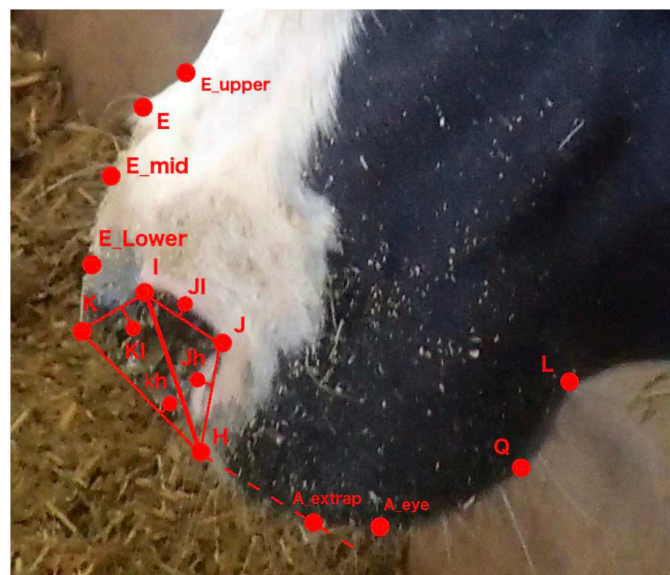
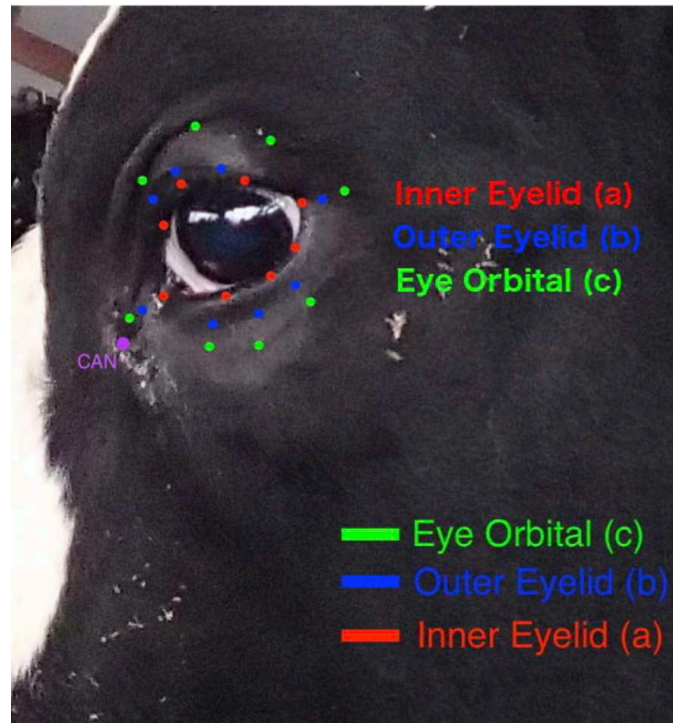


APPENDIX

Appendix A: Specification anatomical reference points and full mathematical derivations of geometric bovine facial biometrics







Canthus Depth Proportion - Depth

$$CDPD = \frac{\|CAN_{int}, CAN\|}{\|Z_{int}, Z\|}$$

$$V1 = \{aa, ba, da\} \rightarrow \{W, X, Z\}$$

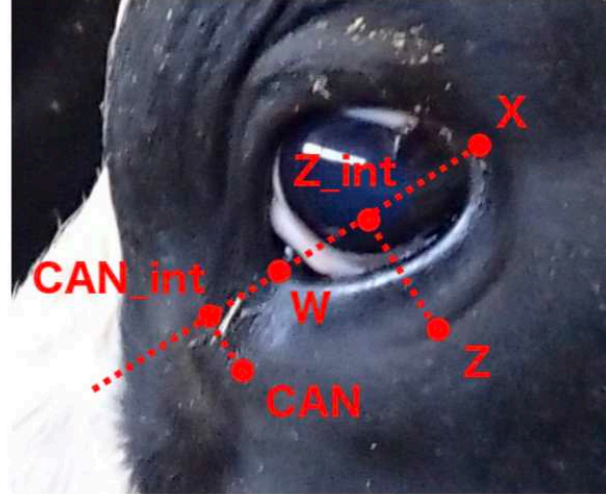
$$V2 = \{aa, ba, db\} \rightarrow \{W, X, Z\}$$

$$V3 = \{aa, ba, dc\} \rightarrow \{W, X, Z\}$$

$$V4 = \{ab, bb, da\} \rightarrow \{W, X, Z\}$$

$$V5 = \{ab, bb, db\} \rightarrow \{W, X, Z\}$$

$$V6 = \{ab, bb, dc\} \rightarrow \{W, X, Z\}$$



Canthus Depth Proportion - Length

$$CDPL = \frac{\|CAN_{int}, CAN\|}{\|W, X\|}$$

$$V1 = \{aa, ba\} \rightarrow \{W, X\}$$

$$V2 = \{ab, bb\} \rightarrow \{W, X\}$$

$$V3 = \{ac, bc\} \rightarrow \{W, X\}$$

$$V4 = \{aa, bb\} \rightarrow \{W, X\}$$

$$V5 = \{ab, ba\} \rightarrow \{W, X\}$$



Canthus Length Proportion

$$CLP = \frac{\|CAN_{int}, W\|}{\|W, X\|}$$

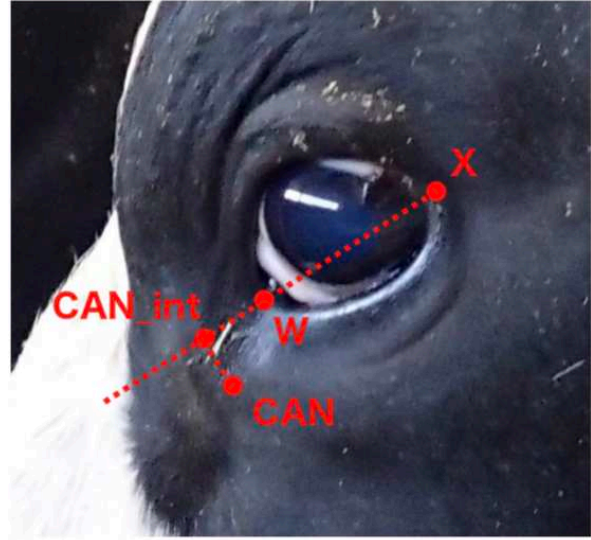
$$V1 = \{aa, ba\} \rightarrow \{W, X\}$$

$$V2 = \{ab, bb\} \rightarrow \{W, X\}$$

$$V3 = \{ac, bc\} \rightarrow \{W, X\}$$

$$V4 = \{aa, bb\} \rightarrow \{W, X\}$$

$$V5 = \{ab, ba\} \rightarrow \{W, X\}$$



Canthus Width-to-Height Ratio

$$CWTHR = \frac{\|CAN, CAN_{int}\|}{\|CAN_{int}, W\|}$$

$$V1 = \{aa, ba\} \rightarrow \{W, X\}$$

$$V2 = \{ab, bb\} \rightarrow \{W, X\}$$

$$V3 = \{ac, bc\} \rightarrow \{W, X\}$$

$$V4 = \{aa, bb\} \rightarrow \{W, X\}$$

$$V5 = \{ab, ba\} \rightarrow \{W, X\}$$



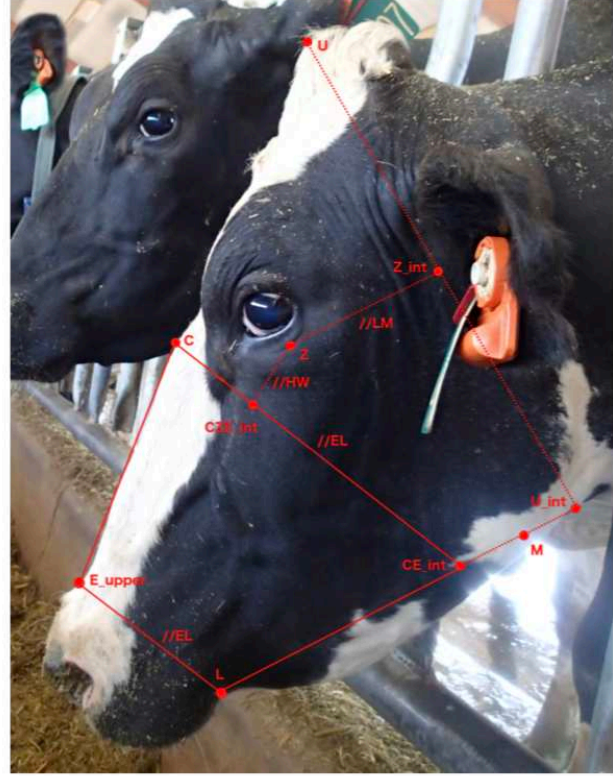
Cheek-Nose Size Proportion

$$CNSP = \frac{A(Z, CZE_{int}CE_{int}U_{int}Y_{int})}{A(LCE_{int}CE_{upper})}$$

$$\begin{aligned} V1 &= \{C_{extrap}, da, L\} \rightarrow \{C, Z, L\} \\ V2 &= \{C_{extrap}, da, L_{full}\} \rightarrow \{C, Z, L\} \\ V3 &= \{C_{extrap}, db, L\} \rightarrow \{C, Z, L\} \\ V4 &= \{C_{extrap}, db, L_{full}\} \rightarrow \{C, Z, L\} \\ V5 &= \{C_{extrap}, dc, L\} \rightarrow \{C, Z, L\} \\ V6 &= \{C_{extrap}, dc, L_{full}\} \rightarrow \{C, Z, L\} \end{aligned}$$

$$\begin{aligned} V7 &= \{C_{eye}, da, L\} \rightarrow \{C, Z, L\} \\ V8 &= \{C_{eye}, da, L_{full}\} \rightarrow \{C, Z, L\} \\ V9 &= \{C_{eye}, db, L\} \rightarrow \{C, Z, L\} \\ V10 &= \{C_{eye}, db, L_{full}\} \rightarrow \{C, Z, L\} \\ V11 &= \{C_{eye}, dc, L\} \rightarrow \{C, Z, L\} \\ V12 &= \{C_{eye}, dc, L_{full}\} \rightarrow \{C, Z, L\} \end{aligned}$$

$$\begin{aligned} V13 &= \{D, da, L\} \rightarrow \{C, Z, L\} \\ V14 &= \{D, da, L_{full}\} \rightarrow \{C, Z, L\} \\ V15 &= \{D, db, L\} \rightarrow \{C, Z, L\} \\ V16 &= \{D, db, L_{full}\} \rightarrow \{C, Z, L\} \\ V17 &= \{D, dc, L\} \rightarrow \{C, Z, L\} \\ V18 &= \{D, dc, L_{full}\} \rightarrow \{C, Z, L\} \end{aligned}$$



Chin Length Proportion

$$CLP = \frac{\|L, A_{int}\|}{\|L, U_{int}\|}$$

$$V1 = \{A_{extrap}, L\} \rightarrow \{A, L\}$$

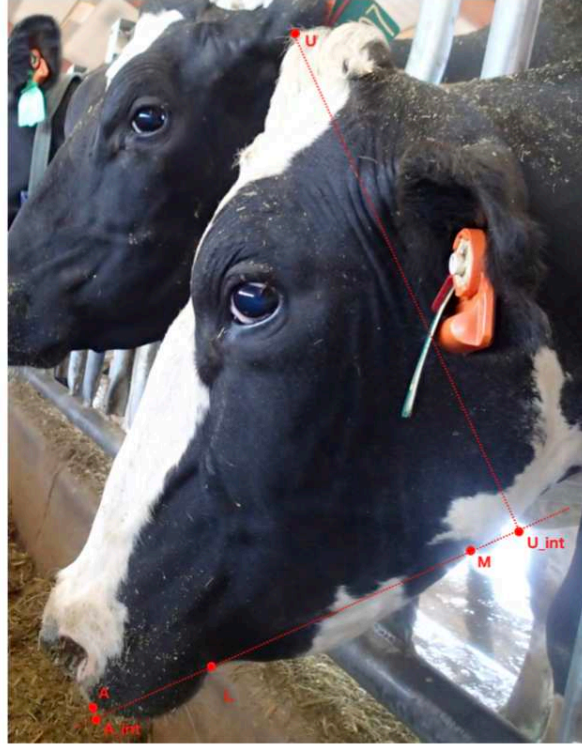
$$V2 = \{A_{eye}, L\} \rightarrow \{A, L\}$$

$$V3 = \{B, L\} \rightarrow \{A, L\}$$

$$V4 = \{A_{extrap}, L_{full}\} \rightarrow \{A, L\}$$

$$V5 = \{A_{eye}, L_{full}\} \rightarrow \{A, L\}$$

$$V6 = \{B, L_{full}\} \rightarrow \{A, L\}$$



Chin Thickness Point Proportion

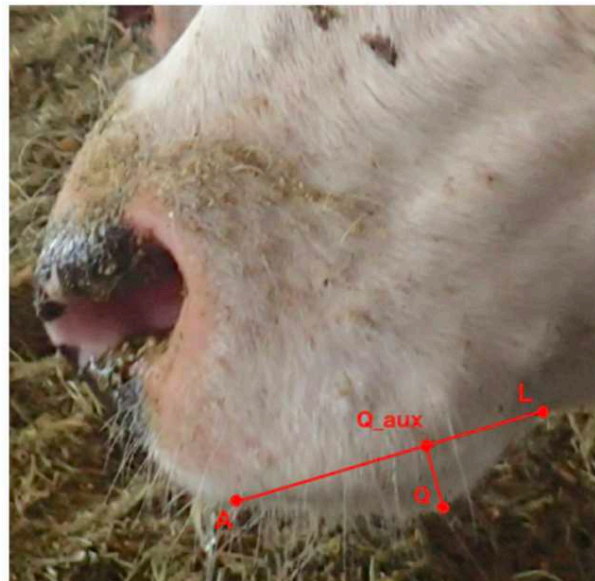
$$CTPP = \frac{\|A, Q_{aux1}\|}{\|A, L\|}$$

$$V1 = \{A_{extrap}, L\} \rightarrow \{A, L\}$$

$$V2 = \{A_{extrap}, L_{full}\} \rightarrow \{A, L\}$$

$$V3 = \{A_{eye}, L\} \rightarrow \{A, L\}$$

$$V4 = \{A_{eye}, L_{full}\} \rightarrow \{A, L\}$$



Chin Thickness Proportion

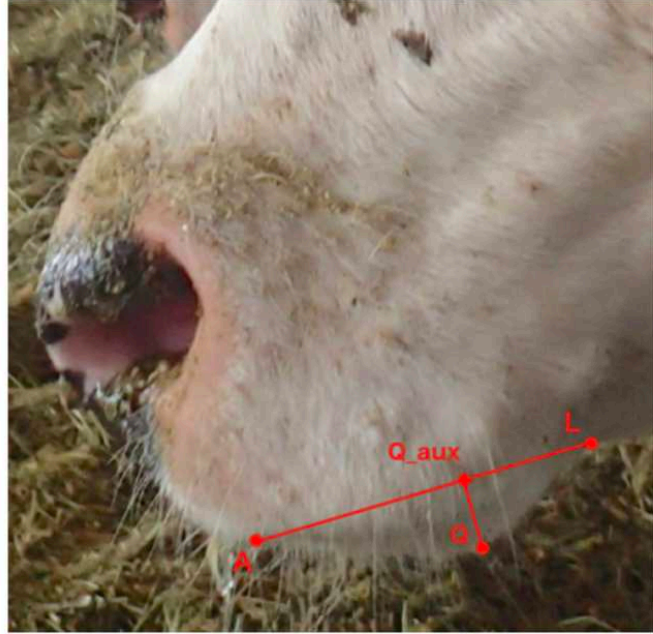
$$CTP = \frac{\|Q, Q_{aux1}\|}{\|A, L\|}$$

$$V1 = \{A_{extrap}, L\} \rightarrow \{A, L\}$$

$$V2 = \{A_{extrap}, L_{full}\} \rightarrow \{A, L\}$$

$$V3 = \{A_{eye}, L\} \rightarrow \{A, L\}$$

$$V4 = \{A_{eye}, L_{full}\} \rightarrow \{A, L\}$$

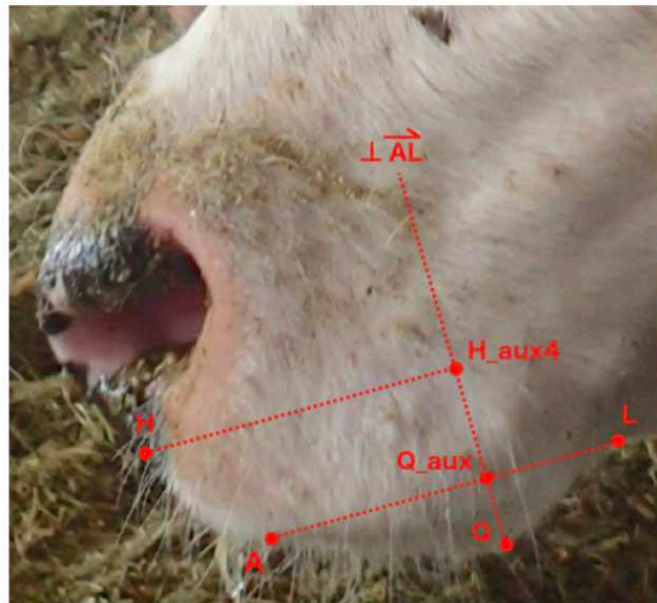


Chin-to-Lip Thickness Ratio

$$CLTR = \frac{\|Q, Q_{aux3}\|}{\|Q, H_{aux4}\|}$$

$$V1 = \{A_{extrap}, L\} \rightarrow \{A, L\}$$

$$V2 = \{A_{eye}, L_{full}\} \rightarrow \{A, L\}$$



Cranio-Topline Length Ratio

$$CTLR = \frac{\|S, T_{int}\|}{\|W, H\|}$$

$$V1 = \{S_{extrap}, T_{int1}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V2 = \{S_{extrap}, T_{int1}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V3 = \{S_{extrap}, T_{int1}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V4 = \{S_{extrap}, T_{int2}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V5 = \{S_{extrap}, T_{int2}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V6 = \{S_{extrap}, T_{int2}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V7 = \{S_{eye}, T_{int1}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

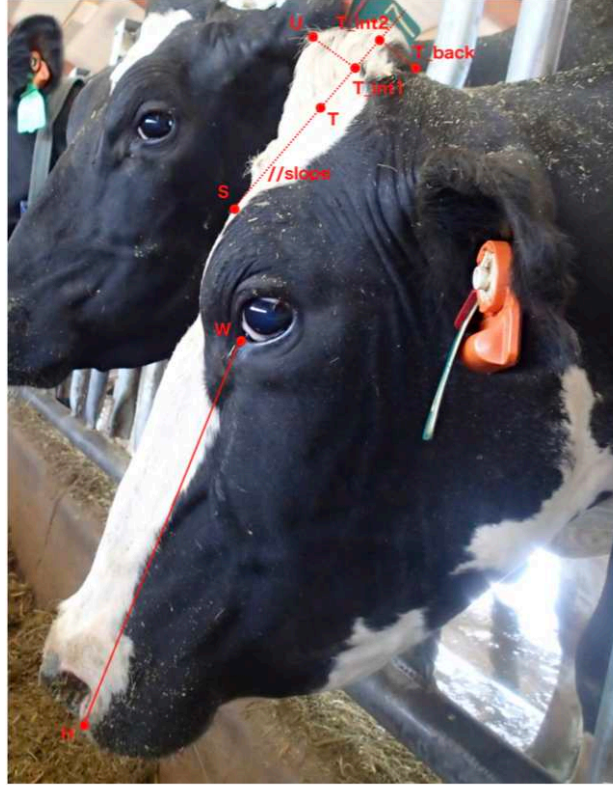
$$V8 = \{S_{eye}, T_{int1}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V9 = \{S_{eye}, T_{int1}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V10 = \{S_{eye}, T_{int2}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V11 = \{S_{eye}, T_{int2}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V12 = \{S_{eye}, T_{int2}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$



Eye Depth Point Proportion

$$EDPP = \frac{\|ay\|}{\|ab\|}$$

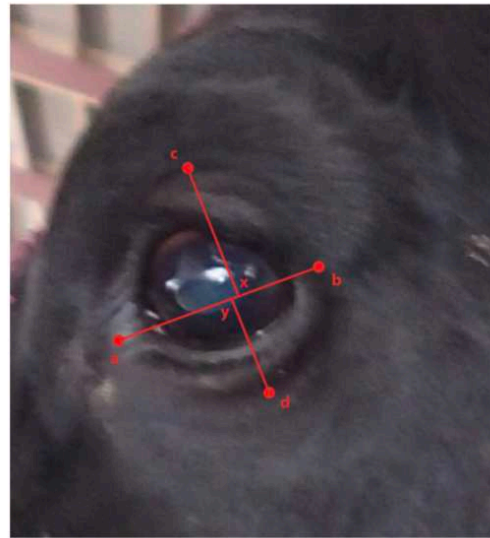
$$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$



Eye-Cranio Size Ratio_Poly

$$ECSR_P = \frac{A(WYXZ)}{A(WST_{poll}Z_{int3}Z)}$$

$$V1 = \{aa, ba, ca, da, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V2 = \{aa, ba, cb, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V3 = \{aa, ba, cc, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V4 = \{aa, ba, cb, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V5 = \{aa, ba, cc, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V6 = \{ab, bb, ca, da, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V7 = \{ab, bb, cb, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V8 = \{ab, bb, cc, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V9 = \{ab, bb, cb, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V10 = \{ab, bb, cc, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V11 = \{ac, bc, ca, da, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V12 = \{ac, bc, cb, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V13 = \{ac, bc, cc, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V14 = \{ac, bc, cb, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V15 = \{ac, bc, cc, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V16 = \{aa, ba, ca, da, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V17 = \{aa, ba, cb, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V18 = \{aa, ba, cc, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V19 = \{aa, ba, cb, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V20 = \{aa, ba, cc, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V21 = \{ab, bb, ca, da, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V22 = \{ab, bb, cb, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V23 = \{ab, bb, cc, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V24 = \{ab, bb, cb, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V25 = \{ab, bb, cc, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

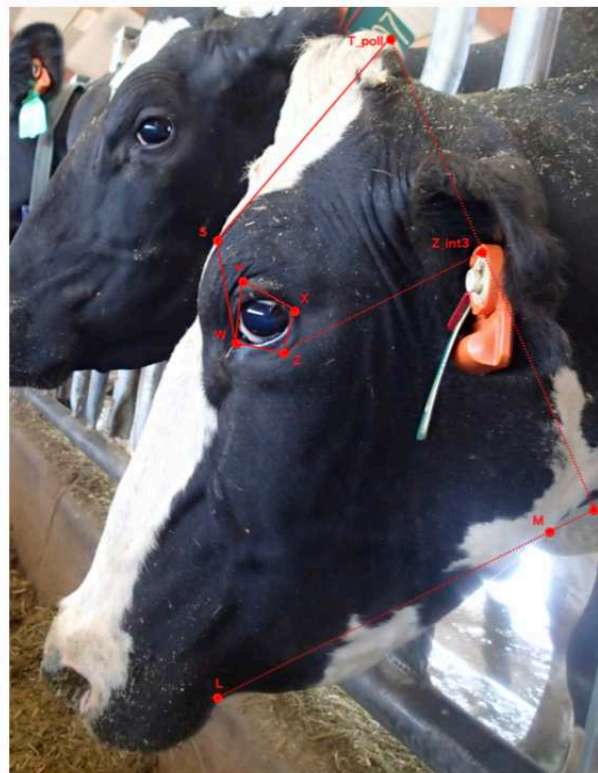
$$V26 = \{ac, bc, ca, da, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V27 = \{ac, bc, cb, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V28 = \{ac, bc, cc, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V29 = \{ac, bc, cb, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V30 = \{ac, bc, cc, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$



Eye Depth Proportion – Front Length

$$EDP = \frac{\|dy\|}{\|ay\|}$$

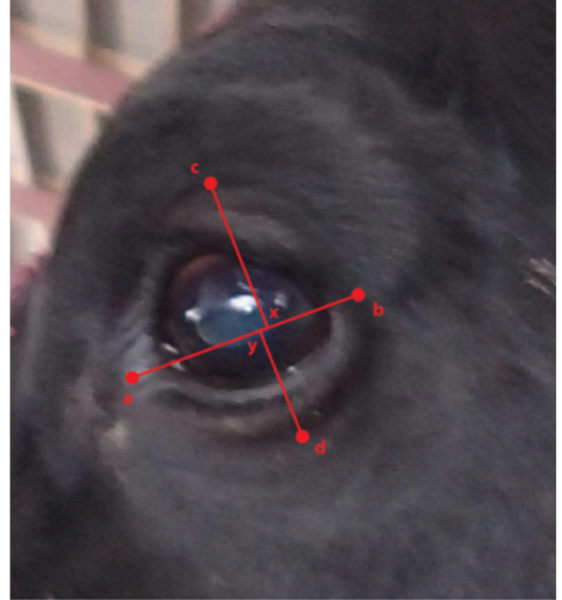
$$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$



Eye Depth Proportion – Full Length

$$EDP = \frac{\|dy\|}{\|ab\|}$$

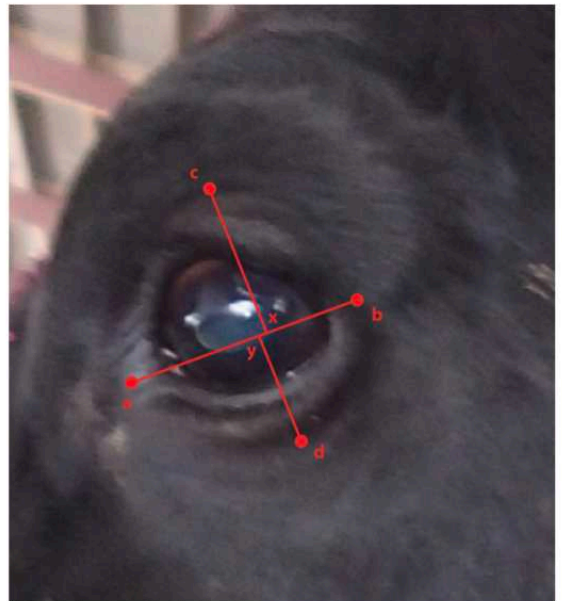
$$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

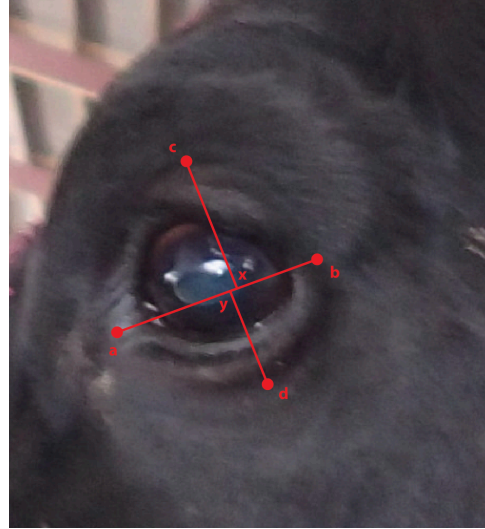


Eye Displacement Proportion

$$EDP = CF * \frac{\|xy\|}{\|ab\|}$$

$$CF = \frac{\|ay\| - \|ax\|}{\|ay\| + \|ax\|}$$

$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Forehead Size Ratio Linear

$$EFSR = \frac{\|W, X\|}{\|W, WX_{int}\|}$$

$V1 = \{aa, ba\} \rightarrow \{W, X\}$
 $V2 = \{ab, bb\} \rightarrow \{W, X\}$
 $V3 = \{ac, bc\} \rightarrow \{W, X\}$
 $V4 = \{aa, bb\} \rightarrow \{W, X\}$
 $V5 = \{ab, ba\} \rightarrow \{W, X\}$



Eye Forehead Size Ratio Poly

$$EFSR = \frac{A(WYXZ)}{A(WSTT_{int}Z_{int2}Z)}$$

$$V1 = \{aa, ba, ca, da, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V2 = \{aa, ba, cb, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V3 = \{aa, ba, cc, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V4 = \{aa, ba, cb, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V5 = \{aa, ba, cc, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V6 = \{ab, bb, ca, da, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V7 = \{ab, bb, cb, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V8 = \{ab, bb, cc, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V9 = \{ab, bb, cb, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V10 = \{ab, bb, cc, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V11 = \{ac, bc, ca, da, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V12 = \{ac, bc, cb, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V13 = \{ac, bc, cc, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V14 = \{ac, bc, cb, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V15 = \{ac, bc, cc, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V16 = \{aa, ba, ca, da, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V17 = \{aa, ba, cb, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V18 = \{aa, ba, cc, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V19 = \{aa, ba, cb, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V20 = \{aa, ba, cc, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V21 = \{ab, bb, ca, da, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V22 = \{ab, bb, cb, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V23 = \{ab, bb, cc, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V24 = \{ab, bb, cb, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V25 = \{ab, bb, cc, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V26 = \{ac, bc, ca, da, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V27 = \{ac, bc, cb, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V28 = \{ac, bc, cc, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V29 = \{ac, bc, cb, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

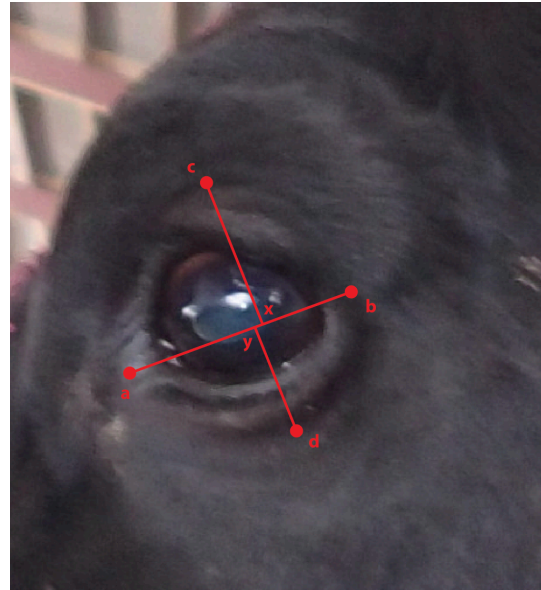
$$V30 = \{ac, bc, cc, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$



Eye Height Point Proportion

$$EHPP = \frac{\|ax\|}{\|ab\|}$$

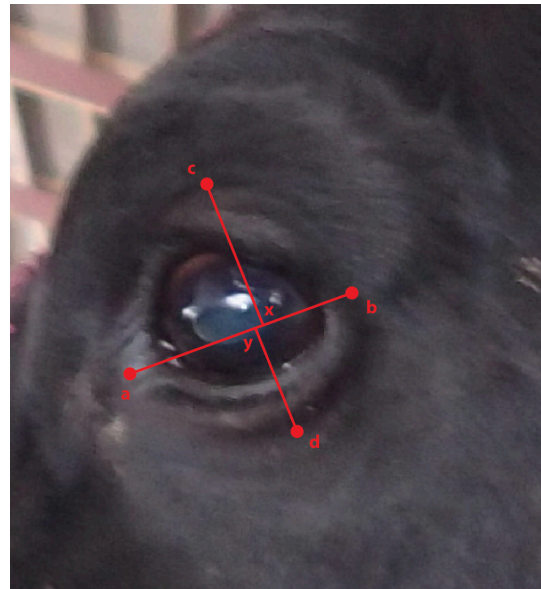
$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Height Proportion – Front Length

$$EHP = \frac{\|cx\|}{\|ax\|}$$

$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Height Proportion – Full Length

$$EHP = \frac{\|cx\|}{\|ab\|}$$

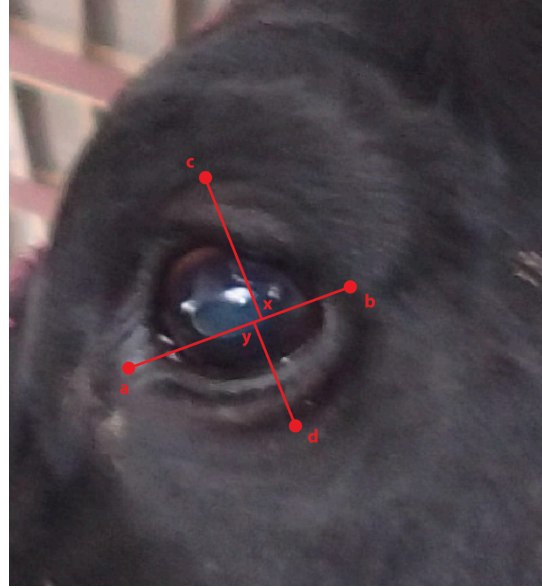
$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Length Proportion - Height

$$ELP = \frac{\min (\|bx\| + \|by\|)}{\|cx\| + \|dy\|}$$

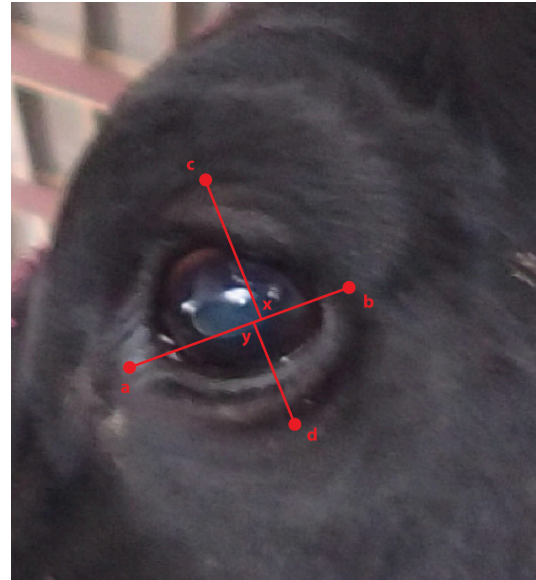
$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Length Proportion - Length

$$ELP = \frac{\min (\|bx\| + \|by\|)}{\|ab\|}$$

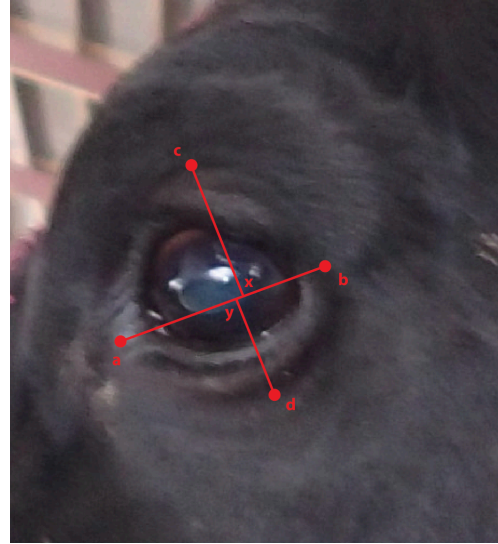
$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$

$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Orbital Height-to-Length Ratio

$$EOHLR = \frac{\|S, S_{int}\|}{\|D, S_{int}\|}$$

$$V1 = \{S_{extrap}, \overline{HW}\} \rightarrow \{S, //slope\}$$

$$V2 = \{S_{extrap}, \overline{S_{eye}T_{slope}}\} \rightarrow \{S, //slope\}$$

$$V3 = \{S_{extrap}, \overline{S_{eye}T_{poll}}\} \rightarrow \{S, //slope\}$$

$$V4 = \{S_{extrap}, \overline{S_{eye}T_{top}}\} \rightarrow \{S, //slope\}$$

$$V5 = \{S_{extrap}, \overline{S_{extrap}T_{slope}}\} \rightarrow \{S, //slope\}$$

$$V6 = \{S_{extrap}, \overline{S_{extrap}T_{poll}}\} \rightarrow \{S, //slope\}$$

$$V7 = \{S_{extrap}, \overline{S_{extrap}T_{top}}\} \rightarrow \{S, //slope\}$$

$$V8 = \{S_{eye}, \overline{HW}\} \rightarrow \{S, //slope\}$$

$$V9 = \{S_{eye}, \overline{S_{eye}T_{slope}}\} \rightarrow \{S, //slope\}$$

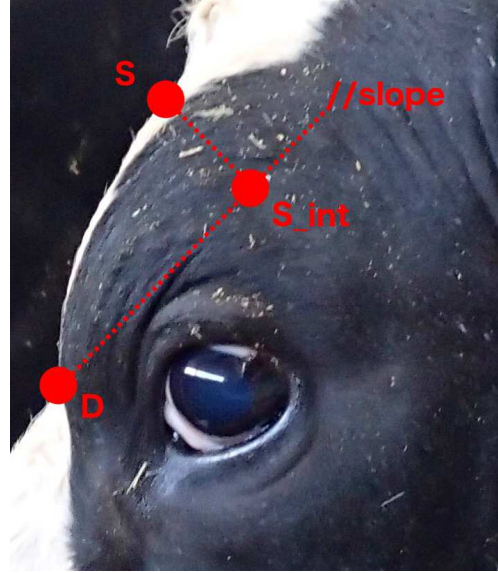
$$V10 = \{S_{eye}, \overline{S_{eye}T_{poll}}\} \rightarrow \{S, //slope\}$$

$$V11 = \{S_{eye}, \overline{S_{eye}T_{top}}\} \rightarrow \{S, //slope\}$$

$$V12 = \{S_{eye}, \overline{S_{extrap}T_{slope}}\} \rightarrow \{S, //slope\}$$

$$V13 = \{S_{eye}, \overline{S_{extrap}T_{poll}}\} \rightarrow \{S, //slope\}$$

$$V14 = \{S_{eye}, \overline{S_{extrap}T_{top}}\} \rightarrow \{S, //slope\}$$



Eye Orbital Projection Proportion

$$EOPP = \frac{\|Y, S_{extrap}\|}{\|Y, Z\|}$$

$$V1 = \{ca, da\} \rightarrow \{Y, Z\}$$

$$V2 = \{ca, db\} \rightarrow \{Y, Z\}$$

$$V3 = \{ca, dc\} \rightarrow \{Y, Z\}$$

$$V4 = \{cb, da\} \rightarrow \{Y, Z\}$$

$$V5 = \{cb, db\} \rightarrow \{Y, Z\}$$

$$V6 = \{cb, dc\} \rightarrow \{Y, Z\}$$

$$V7 = \{cc, da\} \rightarrow \{Y, Z\}$$

$$V8 = \{cc, db\} \rightarrow \{Y, Z\}$$

$$V9 = \{cc, dc\} \rightarrow \{Y, Z\}$$

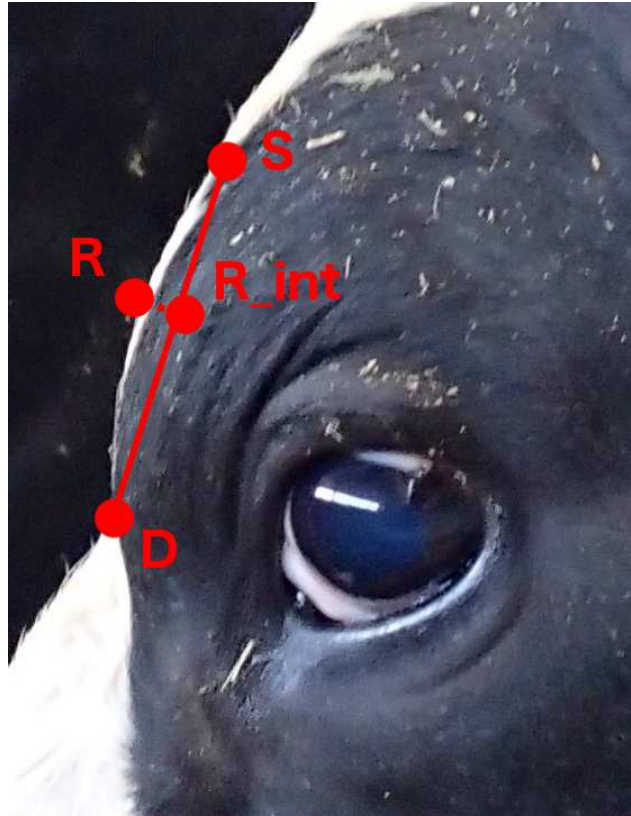


Eye Orbital Roundness Point Proportion

$$EORPP = \frac{\|D, R_{int}\|}{\|D, S\|}$$

$$V1 = \{S_{extrap}\} \rightarrow \{S\}$$

$$V2 = \{S_{eye}\} \rightarrow \{S\}$$



Eye Orbital Roundness Proportion

$$EORP = \frac{\|R, R_{int}\|}{\|G, S\|}$$

$$V1 = \{S_{extrap}\} \rightarrow \{S\}$$

$$V2 = \{S_{eye}\} \rightarrow \{S\}$$



Eye Orbital Thickness Proportion_Poly

$$EOTPP = \frac{A(DRSY)}{A(WYXZ)}$$

$$V1 = \{aa, ba, ca, da, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V2 = \{ab, bb, cb, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V3 = \{ac, bc, cc, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V4 = \{aa, ba, cb, db, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V5 = \{aa, ba, cc, dc, S_{extrap}\} \rightarrow \{W, X, Y, Z, S\}$$

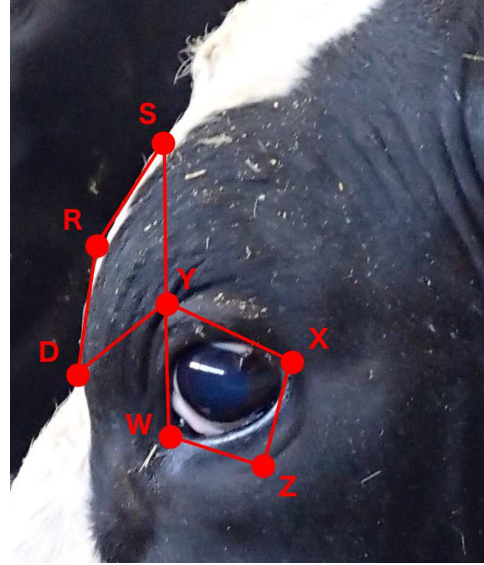
$$V6 = \{aa, ba, ca, da, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V7 = \{ab, bb, cb, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V8 = \{ac, bc, cc, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V9 = \{aa, ba, cb, db, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$

$$V10 = \{aa, ba, cc, dc, S_{eye}\} \rightarrow \{W, X, Y, Z, S\}$$



Eye Orbital-Eye Height Ratio

$$EOEHR = \frac{\|S, S_{int}\|}{\|Z, Z_{int}\|}$$

- $V1 = \{da, S_{eye}, \overline{HW}\} \rightarrow \{Z, S, //slope\}$
 $V2 = \{da, S_{eye}, \overline{S_{eye}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V3 = \{da, S_{eye}, \overline{S_{eye}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V4 = \{da, S_{eye}, \overline{S_{eye}T_{top}}\} \rightarrow \{Z, S, //slope\}$
 $V5 = \{da, S_{eye}, \overline{S_{extrap}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V6 = \{da, S_{eye}, \overline{S_{extrap}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V7 = \{da, S_{eye}, \overline{S_{extrap}T_{top}}\} \rightarrow \{Z, S, //slope\}$

 $V8 = \{da, S_{extrap}, \overline{HW}\} \rightarrow \{Z, S, //slope\}$
 $V9 = \{da, S_{extrap}, \overline{S_{eye}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V10 = \{da, S_{extrap}, \overline{S_{eye}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V11 = \{da, S_{extrap}, \overline{S_{eye}T_{top}}\} \rightarrow \{Z, S, //slope\}$
 $V12 = \{da, S_{extrap}, \overline{S_{extrap}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V13 = \{da, S_{extrap}, \overline{S_{extrap}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V14 = \{da, S_{extrap}, \overline{S_{extrap}T_{top}}\} \rightarrow \{Z, S, //slope\}$

 $V15 = \{db, S_{eye}, \overline{HW}\} \rightarrow \{Z, S, //slope\}$
 $V16 = \{db, S_{eye}, \overline{S_{eye}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V17 = \{db, S_{eye}, \overline{S_{eye}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V18 = \{db, S_{eye}, \overline{S_{eye}T_{top}}\} \rightarrow \{Z, S, //slope\}$
 $V19 = \{db, S_{eye}, \overline{S_{extrap}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V20 = \{db, S_{eye}, \overline{S_{extrap}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V21 = \{db, S_{eye}, \overline{S_{extrap}T_{top}}\} \rightarrow \{Z, S, //slope\}$

 $V22 = \{db, S_{extrap}, \overline{HW}\} \rightarrow \{Z, S, //slope\}$
 $V23 = \{db, S_{extrap}, \overline{S_{eye}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V24 = \{db, S_{extrap}, \overline{S_{eye}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V25 = \{db, S_{extrap}, \overline{S_{eye}T_{top}}\} \rightarrow \{Z, S, //slope\}$
 $V26 = \{db, S_{extrap}, \overline{S_{extrap}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V27 = \{db, S_{extrap}, \overline{S_{extrap}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V28 = \{db, S_{extrap}, \overline{S_{extrap}T_{top}}\} \rightarrow \{Z, S, //slope\}$



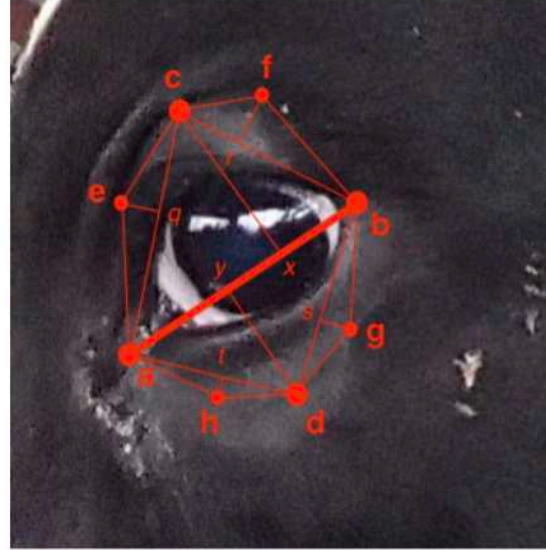
- $V29 = \{dc, S_{eye}, \overline{HW}\} \rightarrow \{Z, S, //slope\}$
 $V30 = \{dc, S_{eye}, \overline{S_{eye}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V31 = \{dc, S_{eye}, \overline{S_{eye}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V32 = \{dc, S_{eye}, \overline{S_{eye}T_{top}}\} \rightarrow \{Z, S, //slope\}$
 $V33 = \{dc, S_{eye}, \overline{S_{extrap}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V34 = \{dc, S_{eye}, \overline{S_{extrap}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V35 = \{dc, S_{eye}, \overline{S_{extrap}T_{top}}\} \rightarrow \{Z, S, //slope\}$

 $V36 = \{dc, S_{extrap}, \overline{HW}\} \rightarrow \{Z, S, //slope\}$
 $V37 = \{dc, S_{extrap}, \overline{S_{eye}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V38 = \{dc, S_{extrap}, \overline{S_{eye}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V39 = \{dc, S_{extrap}, \overline{S_{eye}T_{top}}\} \rightarrow \{Z, S, //slope\}$
 $V40 = \{dc, S_{extrap}, \overline{S_{extrap}T_{slope}}\} \rightarrow \{Z, S, //slope\}$
 $V41 = \{dc, S_{extrap}, \overline{S_{extrap}T_{poll}}\} \rightarrow \{Z, S, //slope\}$
 $V42 = \{dc, S_{extrap}, \overline{S_{extrap}T_{top}}\} \rightarrow \{Z, S, //slope\}$

Eye Roundness Proportion – Lower Front

$$ERPUB = \frac{\|ht\|}{\|ad\|}$$

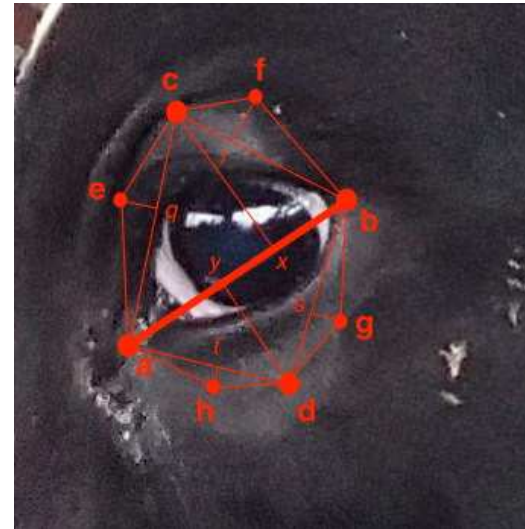
$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Roundness Proportion – Lower Back

$$ERPLB = \frac{\|gs\|}{\|bd\|}$$

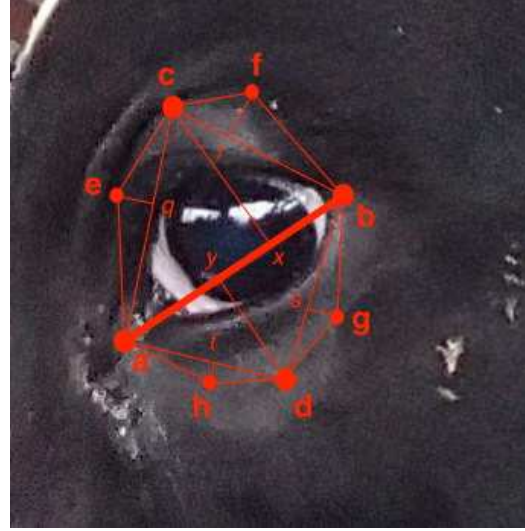
$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Roundness Proportion – Upper Back

$$ERPUB = \frac{\|fr\|}{\|cb\|}$$

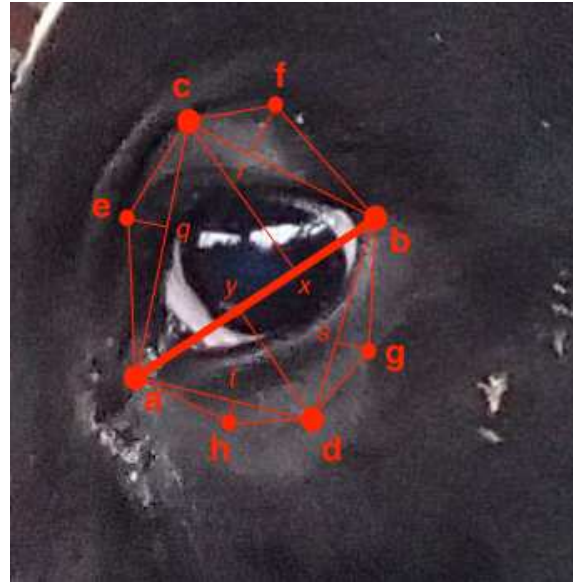
- $V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Roundness Proportion – Upper Front

$$ERPUF = \frac{\|eq\|}{\|ac\|}$$

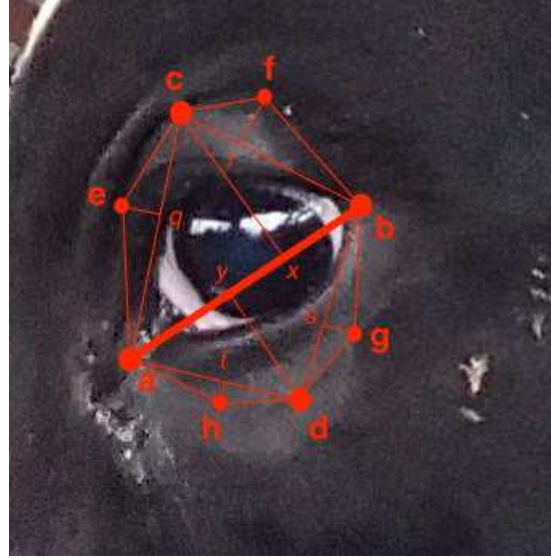
- $V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Roundness Proportion – Lower Back

$$ERPLB = \frac{A(bdg)}{A(bdy)}$$

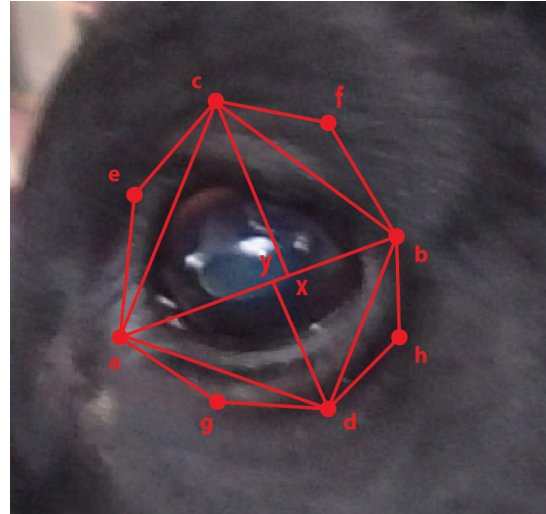
$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Roundness Proportion – Lower Front

$$ERPUB = \frac{A(adg)}{A(ady)}$$

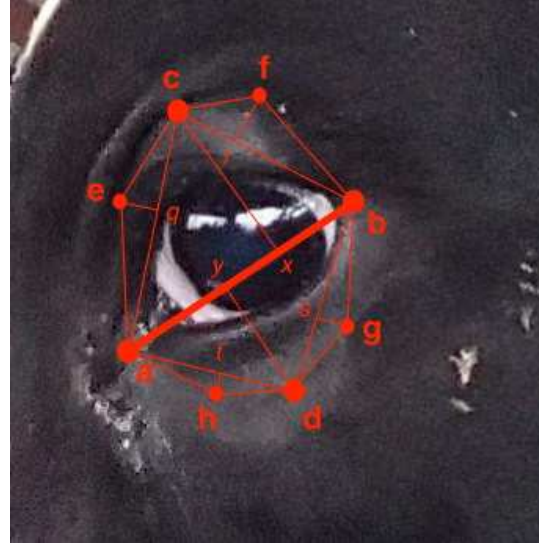
$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Roundness Proportion – Upper Back

$$ERPUB = \frac{A(cfb)}{A(cbx)}$$

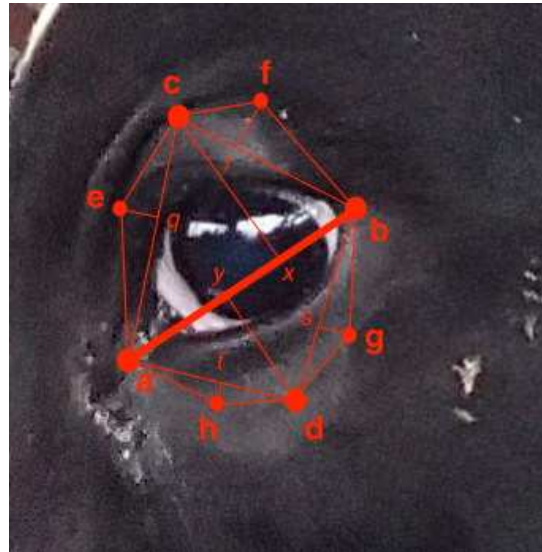
$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Roundness Proportion – Upper Front

$$ERPUF = \frac{A(aec)}{A(acx)}$$

$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$
 $V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$



Eye Roundness Proportion – Total

$$ERPT = \frac{A(ace) + A(cbf) + A(ahd) + A(bdg)}{A(abcd)}$$

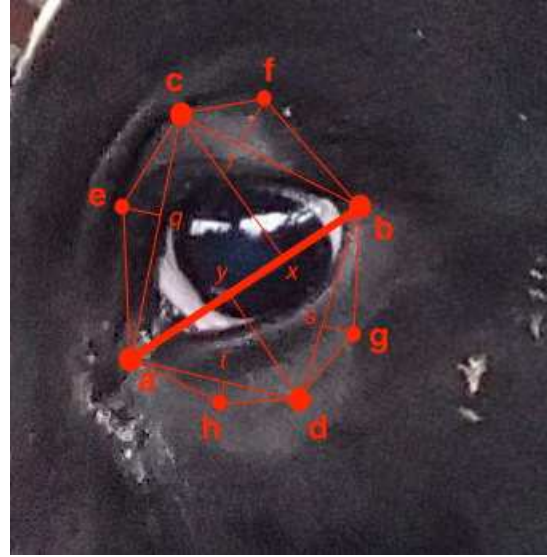
$$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$



Eye Roundness Ratio – Front to Back

$$ERR = \frac{A(ace) + A(ahd)}{A(cbf) + A(bdg)}$$

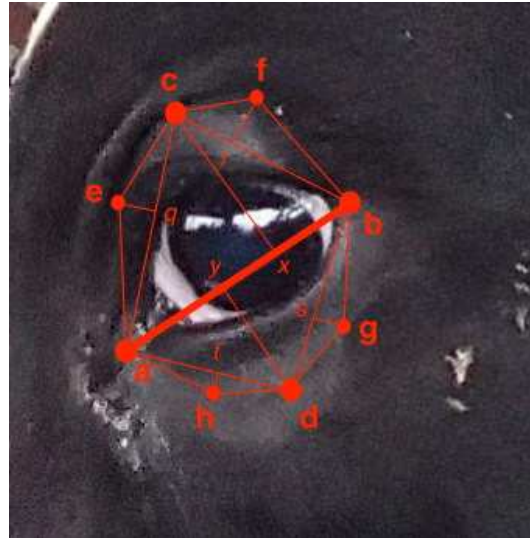
$$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$



Eye Roundness Ratio – Top to Bottom

$$ERR = \frac{A(ace) + A(cbf)}{A(ahd) + A(bdg)}$$

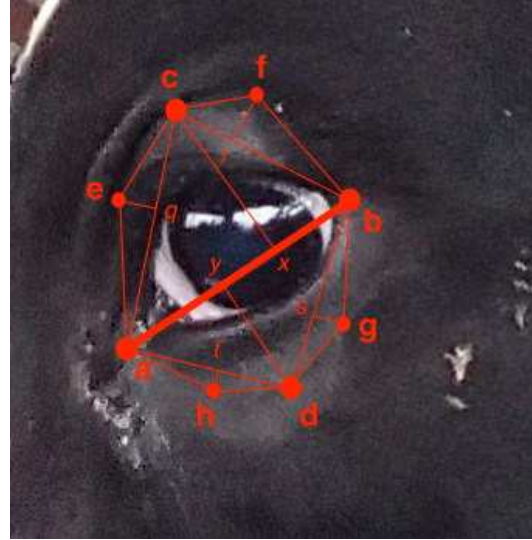
$$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$



Eye Topline Size Ratio - Linear

$$ETSRL = \frac{\|W, X\|}{\|W, H\|}$$

$$V1 = \{aa, ba\} \rightarrow \{W, X\}$$

$$V2 = \{ab, bb\} \rightarrow \{W, X\}$$

$$V3 = \{ac, bc\} \rightarrow \{W, X\}$$

$$V4 = \{aa, bb\} \rightarrow \{W, X\}$$

$$V5 = \{ab, ba\} \rightarrow \{W, X\}$$



Eye Topline Size Ratio - Poly

$$ETSRP = \frac{A(WYXZ)}{A(E_{upper}HZWD)}$$

$$V1 = \{aa, ba, ca, da\} \rightarrow \{W, X, Y, Z\}$$

$$V2 = \{aa, ba, cb, db\} \rightarrow \{W, X, Y, Z\}$$

$$V3 = \{aa, ba, cc, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V4 = \{aa, ba, cb, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V5 = \{aa, ba, cc, db\} \rightarrow \{W, X, Y, Z\}$$

$$V6 = \{ab, bb, ca, da\} \rightarrow \{W, X, Y, Z\}$$

$$V7 = \{ab, bb, cb, db\} \rightarrow \{W, X, Y, Z\}$$

$$V8 = \{ab, bb, cc, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V9 = \{ab, bb, cb, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V10 = \{ab, bb, cc, db\} \rightarrow \{W, X, Y, Z\}$$

$$V11 = \{ac, bc, ca, da\} \rightarrow \{W, X, Y, Z\}$$

$$V12 = \{ac, bc, cb, db\} \rightarrow \{W, X, Y, Z\}$$

$$V13 = \{ac, bc, cc, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V14 = \{ac, bc, cb, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V15 = \{ac, bc, cc, db\} \rightarrow \{W, X, Y, Z\}$$



Eye Width-to-Height Ratio

$$EWHR = \frac{\|cx\| + \|dy\|}{\|ab\|}$$

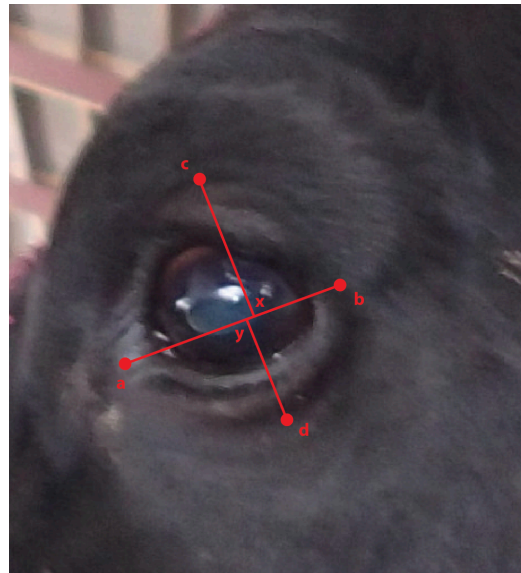
$$V1 = \{aa, ba, ca, da, ea, fa, ga, ha\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V2 = \{ab, bb, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V3 = \{ac, bc, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V4 = \{aa, ba, cb, db, eb, fb, gb, hb\} \rightarrow \{a, b, c, d, e, f, g, h\}$$

$$V5 = \{ab, bb, cc, dc, ec, fc, gc, hc\} \rightarrow \{a, b, c, d, e, f, g, h\}$$



Eye-Sinus Size Ratio_Linear

$$ESSRL = \frac{\|W, X\|}{\|C_{int}, X\|}$$

$$V1 = \{aa, ba, C_{extrap}\} \rightarrow \{W, X, C\}$$

$$V2 = \{ab, bb, C_{extrap}\} \rightarrow \{W, X, C\}$$

$$V3 = \{ac, bc, C_{extrap}\} \rightarrow \{W, X, C\}$$

$$V4 = \{aa, bb, C_{extrap}\} \rightarrow \{W, X, C\}$$

$$V5 = \{ab, ba, C_{extrap}\} \rightarrow \{W, X, C\}$$

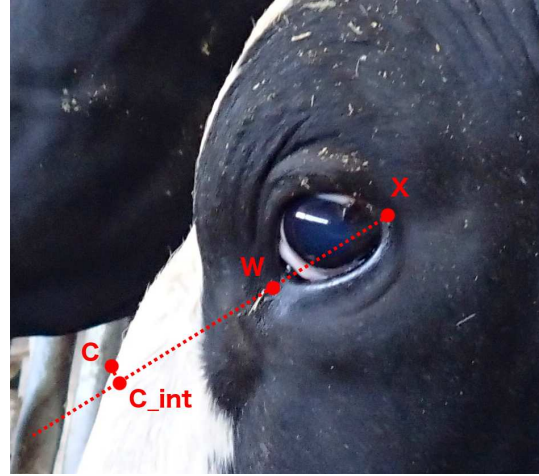
$$V6 = \{aa, ba, C_{eye}\} \rightarrow \{W, X, C\}$$

$$V7 = \{ab, bb, C_{eye}\} \rightarrow \{W, X, C\}$$

$$V8 = \{ac, bc, C_{eye}\} \rightarrow \{W, X, C\}$$

$$V9 = \{aa, bb, C_{eye}\} \rightarrow \{W, X, C\}$$

$$V10 = \{ab, ba, C_{eye}\} \rightarrow \{W, X, C\}$$



Eye-Sinus Size Ratio Poly

$$ESSRP = \frac{A(CDW)}{A(WYXZ)}$$

$$V1 = \{aa, ba, ca, da, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V2 = \{aa, ba, cb, db, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V3 = \{aa, ba, cc, dc, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V4 = \{aa, ba, cb, dc, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V5 = \{aa, ba, cc, db, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V6 = \{ab, bb, ca, da, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V7 = \{ab, bb, cb, db, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V8 = \{ab, bb, cc, dc, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V9 = \{ab, bb, cb, dc, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V10 = \{ab, bb, cc, db, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V11 = \{ac, bc, ca, da, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V12 = \{ac, bc, cb, db, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V13 = \{ac, bc, cc, dc, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V14 = \{ac, bc, cb, dc, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V15 = \{ac, bc, cc, db, C_{extrap}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V16 = \{aa, ba, ca, da, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V17 = \{aa, ba, cb, db, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V18 = \{aa, ba, cc, dc, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V19 = \{aa, ba, cb, dc, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V20 = \{aa, ba, cc, db, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V21 = \{ab, bb, ca, da, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V22 = \{ab, bb, cb, db, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V23 = \{ab, bb, cc, dc, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V24 = \{ab, bb, cb, dc, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V25 = \{ab, bb, cc, db, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

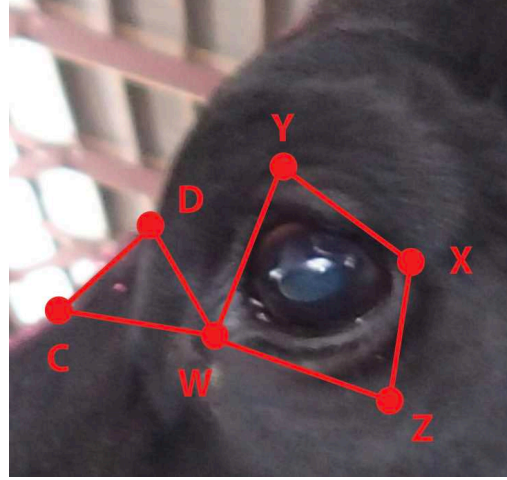
$$V26 = \{ac, bc, ca, da, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V27 = \{ac, bc, cb, db, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V28 = \{ac, bc, cc, dc, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V29 = \{ac, bc, cb, dc, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$

$$V30 = \{ac, bc, cc, db, C_{eye}\} \rightarrow \{W, X, Y, Z, C\}$$



Forehead-Eye Angle_Slope

$$FEAS = CF * \frac{\|T, T_{int}\|}{\|S, T\|}$$

$$CF = \frac{\|T_{int}, T_{aux}\| - \|T, T_{aux}\|}{\|\|T_{int}, T_{aux}\| - \|T, T_{aux}\|\|}$$

$$V1 = \{\overline{aaba}, S_{extrap}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V2 = \{\overline{aaba}, S_{extrap}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V3 = \{\overline{aaba}, S_{extrap}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V4 = \{\overline{aaba}, S_{eye}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V5 = \{\overline{aaba}, S_{eye}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V6 = \{\overline{aaba}, S_{eye}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V7 = \{\overline{abbb}, S_{extrap}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V8 = \{\overline{abbb}, S_{extrap}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V9 = \{\overline{abbb}, S_{extrap}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V10 = \{\overline{abbb}, S_{eye}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V11 = \{\overline{abbb}, S_{eye}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V12 = \{\overline{abbb}, S_{eye}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V13 = \{\overline{aabb}, S_{extrap}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V14 = \{\overline{aabb}, S_{extrap}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V15 = \{\overline{aabb}, S_{extrap}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V16 = \{\overline{aabb}, S_{eye}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V17 = \{\overline{aabb}, S_{eye}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V18 = \{\overline{aabb}, S_{eye}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V19 = \{\overline{abba}, S_{extrap}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

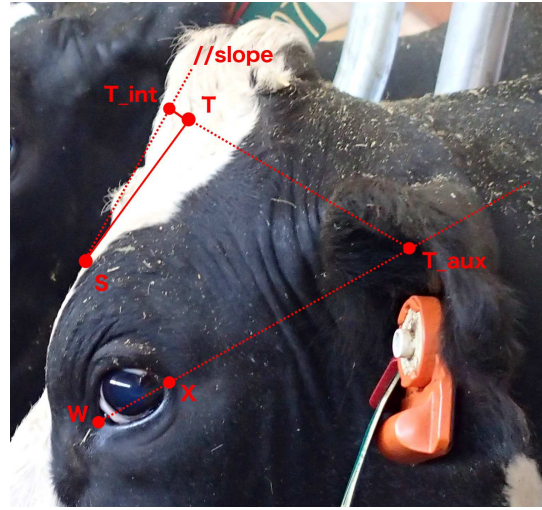
$$V20 = \{\overline{abba}, S_{extrap}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V21 = \{\overline{abba}, S_{extrap}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V22 = \{\overline{abba}, S_{eye}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V23 = \{\overline{abba}, S_{eve}, T_{ton}\} \rightarrow \{//slope, S, T\}$$

$$V24 = \{\overline{abba}, S_{eye}, T_{poll}\} \rightarrow \{//slope, S, T\}$$



Forehead-Jaw Angle_Slope

$$FJAS = CF * \frac{\|T, T_{int}\|}{\|S, T\|}$$

$$C_F = \frac{\|T_{int}, T_{aux}\| - \|T, T_{aux}\|}{\|\|T_{int}, T_{aux}\| - \|T, T_{aux}\|\|}$$

$$V1 = \{\overline{LM}, S_{extrap}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V2 = \{\overline{LM}, S_{extrap}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V3 = \{\overline{LM}, S_{extrap}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V4 = \{\overline{LM}, S_{eye}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V5 = \{LM, S_{eye}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V6 = \{\overline{LM}, S_{eye}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V7 = \{\overline{L_{full}M}, S_{extrap}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

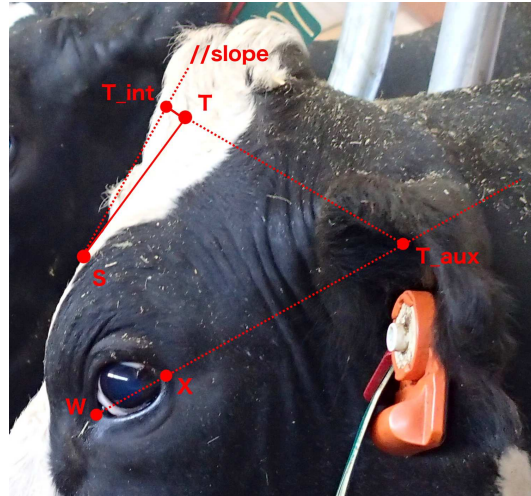
$$V8 = \{\overline{L_{full}M}, S_{extrap}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V9 = \{\overline{L_{full}M}, S_{extrap}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V10 = \{\overline{L_{full}M}, S_{eye}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V11 = \{L_{full}M, S_{eye}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V12 = \{\overline{L_{full}M}, S_{eye}, T_{poll}\} \rightarrow \{//slope, S, T\}$$



Forehead Temple Ratio

$$FTR = \frac{\|S, T_{int}\|}{\|X, X_{int}\|}$$

$$V1 = \{S_{extrap}, T_{poll}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V2 = \{S_{extrap}, T_{poll}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V3 = \{S_{extrap}, T_{poll}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V4 = \{S_{extrap}, T_{poll}, aa, bb\} \rightarrow \{S, T, W, X\}$$

$$V5 = \{S_{extrap}, T_{poll}, ab, ba\} \rightarrow \{S, T, W, X\}$$

$$V6 = \{S_{extrap}, T_{slope}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V7 = \{S_{extrap}, T_{slope}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V8 = \{S_{extrap}, T_{slope}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V9 = \{S_{extrap}, T_{slope}, aa, bb\} \rightarrow \{S, T, W, X\}$$

$$V10 = \{S_{extrap}, T_{slope}, ab, ba\} \rightarrow \{S, T, W, X\}$$

$$V11 = \{S_{extrap}, T_{top}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V12 = \{S_{extrap}, T_{top}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V13 = \{S_{extrap}, T_{top}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V14 = \{S_{extrap}, T_{top}, aa, bb\} \rightarrow \{S, T, W, X\}$$

$$V15 = \{S_{extrap}, T_{top}, ab, ba\} \rightarrow \{S, T, W, X\}$$

$$V16 = \{S_{eye}, T_{poll}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V17 = \{S_{eye}, T_{poll}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V18 = \{S_{eye}, T_{poll}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V19 = \{S_{eye}, T_{poll}, aa, bb\} \rightarrow \{S, T, W, X\}$$

$$V20 = \{S_{eye}, T_{poll}, ab, ba\} \rightarrow \{S, T, W, X\}$$

$$V21 = \{S_{eye}, T_{slope}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V22 = \{S_{eye}, T_{slope}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V23 = \{S_{eye}, T_{slope}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V24 = \{S_{eye}, T_{slope}, aa, bb\} \rightarrow \{S, T, W, X\}$$

$$V25 = \{S_{eye}, T_{slope}, ab, ba\} \rightarrow \{S, T, W, X\}$$

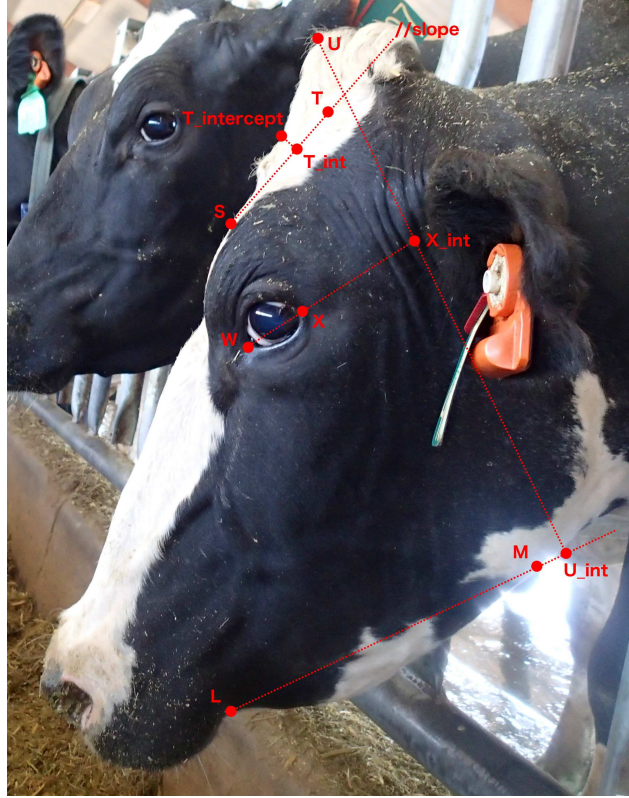
$$V26 = \{S_{eye}, T_{top}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V27 = \{S_{eye}, T_{top}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V28 = \{S_{eye}, T_{top}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V29 = \{S_{eye}, T_{top}, aa, bb\} \rightarrow \{S, T, W, X\}$$

$$V30 = \{S_{eye}, T_{top}, ab, ba\} \rightarrow \{S, T, W, X\}$$



Forehead-Topline Angle_Slope

$$FTAS = CF * \frac{\|T, T_{int}\|}{\|S, T\|}$$

$$CF = \frac{\|T_{int}, T_{aux}\| - \|T, T_{aux}\|}{\|\|T_{int}, T_{aux}\| - \|T, T_{aux}\|\|}$$

$$V1 = \{\overline{WH}, S_{extrap}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

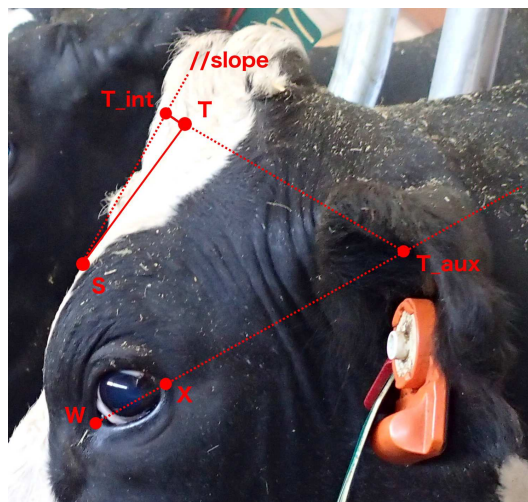
$$V2 = \{\overline{WH}, S_{extrap}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V3 = \{\overline{WH}, S_{extrap}, T_{poll}\} \rightarrow \{//slope, S, T\}$$

$$V4 = \{\overline{WH}, S_{eye}, T_{slope}\} \rightarrow \{//slope, S, T\}$$

$$V5 = \{\overline{WH}, S_{eye}, T_{top}\} \rightarrow \{//slope, S, T\}$$

$$V6 = \{\overline{WH}, S_{eye}, T_{poll}\} \rightarrow \{//slope, S, T\}$$



Forehead Topline Length Ratio

$$FTLR = \frac{\|S, T_{int}\|}{\|W, H\|}$$

$$V1 = \{S_{extrap}, T_{slope}\} \rightarrow \{S, T\}$$

$$V2 = \{S_{extrap}, T_{top}\} \rightarrow \{S, T\}$$

$$V3 = \{S_{extrap}, T_{poll}\} \rightarrow \{S, T\}$$

$$V4 = \{S_{eye}, T_{slope}\} \rightarrow \{S, T\}$$

$$V5 = \{S_{eye}, T_{top}\} \rightarrow \{S, T\}$$

$$V6 = \{S_{eye}, T_{poll}\} \rightarrow \{S, T\}$$



Forehead Width-to-Length Proportion

$$FWLP = \frac{\|X, U_{int}\|}{\|U_{int}, T_{int}\|}$$

$$V1 = \{S_{extrap}, T_{slope}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V2 = \{S_{extrap}, T_{top}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V3 = \{S_{extrap}, T_{poll}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V4 = \{S_{eye}, T_{slope}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V5 = \{S_{eye}, T_{top}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V6 = \{S_{eye}, T_{poll}, aa, ba\} \rightarrow \{S, T, W, X\}$$

$$V7 = \{S_{extrap}, T_{slope}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V8 = \{S_{extrap}, T_{top}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V9 = \{S_{extrap}, T_{poll}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V10 = \{S_{eye}, T_{slope}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V11 = \{S_{eye}, T_{top}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V12 = \{S_{eye}, T_{poll}, ab, bb\} \rightarrow \{S, T, W, X\}$$

$$V13 = \{S_{extrap}, T_{slope}, ac, bc\} \rightarrow \{S, T, W, X\}$$

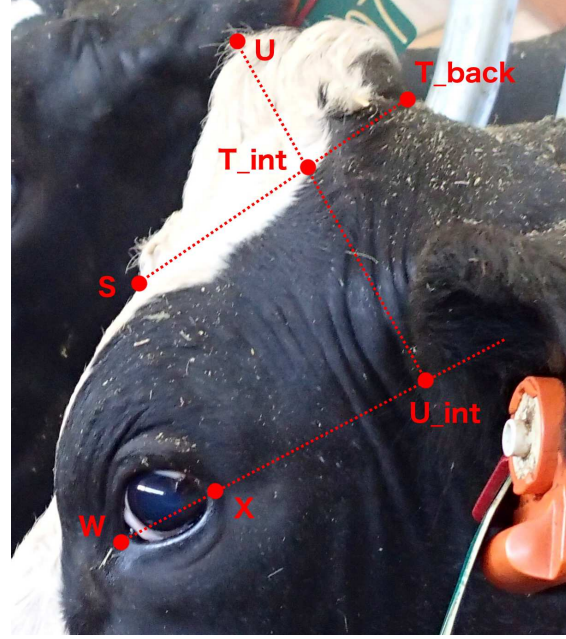
$$V14 = \{S_{extrap}, T_{top}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V15 = \{S_{extrap}, T_{poll}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V16 = \{S_{eye}, T_{slope}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V17 = \{S_{eye}, T_{top}, ac, bc\} \rightarrow \{S, T, W, X\}$$

$$V18 = \{S_{eye}, T_{poll}, ac, bc\} \rightarrow \{S, T, W, X\}$$



Forehead Zygomatic Angle

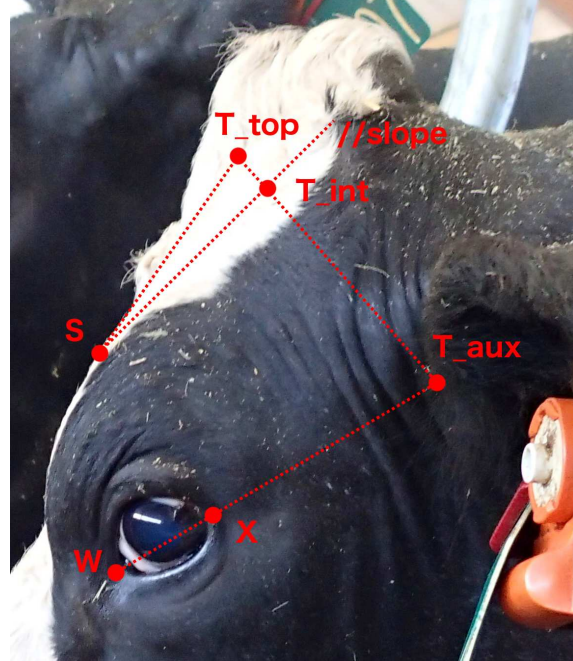
$$FZA = CF * \frac{\|T_{top}, T_{int}\|}{\|S, T_{int}\|}$$

$$CF = \frac{\|T_{top}, T_{aux}\| - \|T_{int}, T_{aux}\|}{\|T_{top}, T_{aux}\| + \|T_{int}, T_{aux}\|}$$

$$\begin{aligned} V1 &= \{S_{extrap}, T_{top}, aa, ba\} \rightarrow \{//slope, W, X\} \\ V2 &= \{S_{extrap}, T_{poll}, aa, ba\} \rightarrow \{//slope, W, X\} \\ V3 &= \{S_{eye}, T_{slope}, aa, ba\} \rightarrow \{//slope, W, X\} \\ V4 &= \{S_{eye}, T_{poll}, aa, ba\} \rightarrow \{//slope, W, X\} \end{aligned}$$

$$\begin{aligned} V5 &= \{S_{extrap}, T_{slope}, ab, bb\} \rightarrow \{//slope, W, X\} \\ V6 &= \{S_{extrap}, T_{poll}, ab, bb\} \rightarrow \{//slope, W, X\} \\ V7 &= \{S_{eye}, T_{slope}, ab, bb\} \rightarrow \{//slope, W, X\} \\ V8 &= \{S_{eye}, T_{poll}, ab, bb\} \rightarrow \{//slope, W, X\} \end{aligned}$$

$$\begin{aligned} V9 &= \{S_{extrap}, T_{slope}, ac, bc\} \rightarrow \{//slope, W, X\} \\ V10 &= \{S_{extrap}, T_{poll}, ac, bc\} \rightarrow \{//slope, W, X\} \\ V11 &= \{S_{eye}, T_{slope}, ac, bc\} \rightarrow \{//slope, W, X\} \\ V12 &= \{S_{eye}, T_{poll}, ac, bc\} \rightarrow \{//slope, W, X\} \end{aligned}$$



Forehead-Poll Length Ratio

$$FPLR = \frac{\|S, T_{mid}\|}{\|S, T_{int}\|}$$

$$V1 = \{S_{extrap}, T_{int1}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V2 = \{S_{extrap}, T_{int1}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V3 = \{S_{extrap}, T_{int1}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V4 = \{S_{extrap}, T_{int2}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V5 = \{S_{extrap}, T_{int2}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V6 = \{S_{extrap}, T_{int2}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V7 = \{S_{eye}, T_{int1}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

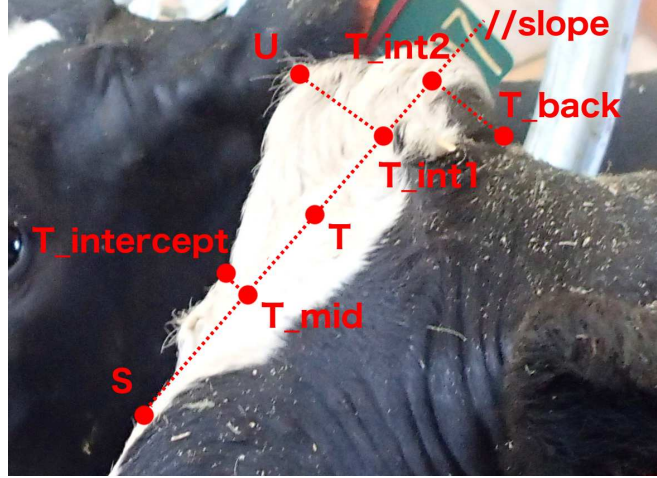
$$V8 = \{S_{eye}, T_{int1}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V9 = \{S_{eye}, T_{int1}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V10 = \{S_{eye}, T_{int2}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V11 = \{S_{eye}, T_{int2}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V12 = \{S_{eye}, T_{int2}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$



Jaw Angle Slope

$$JAS = \frac{\|L, L_{int}\| - \|M, M_{int}\|}{\|L_{int}, M_{int}\|}$$

$$V1 = \{aa, L\} \rightarrow \{W, L\}$$

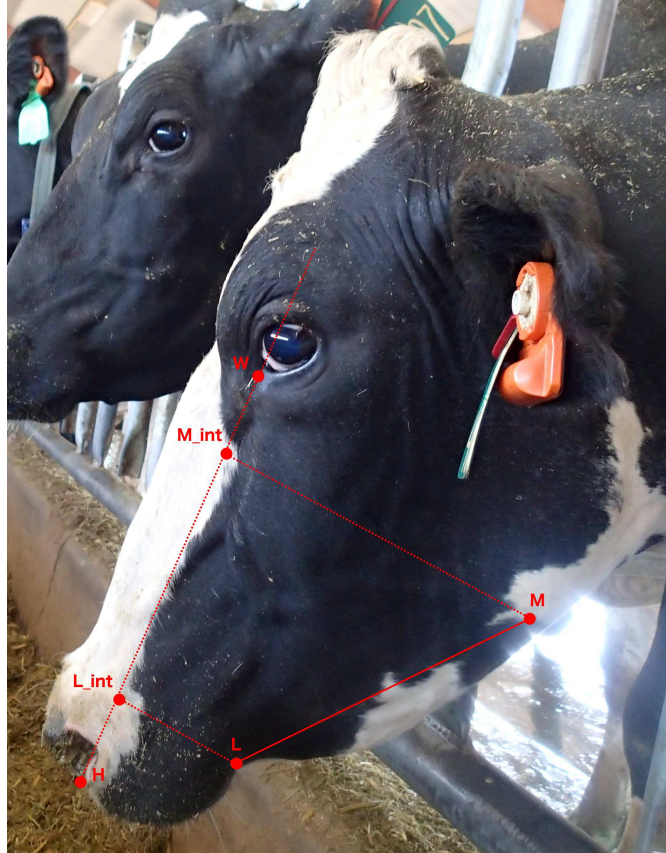
$$V2 = \{ab, L\} \rightarrow \{W, L\}$$

$$V3 = \{ac, L\} \rightarrow \{W, L\}$$

$$V4 = \{aa, L_{full}\} \rightarrow \{W, L\}$$

$$V5 = \{ab, L_{full}\} \rightarrow \{W, L\}$$

$$V6 = \{ac, L_{full}\} \rightarrow \{W, L\}$$



Jaw-Midface Size Ratio

$$JMSR = \frac{A(CEyz)}{A(xzLM)}$$

$$V1 = \{up_{int1}, low_1, L\} \rightarrow \{up_{int}, low, L\}$$

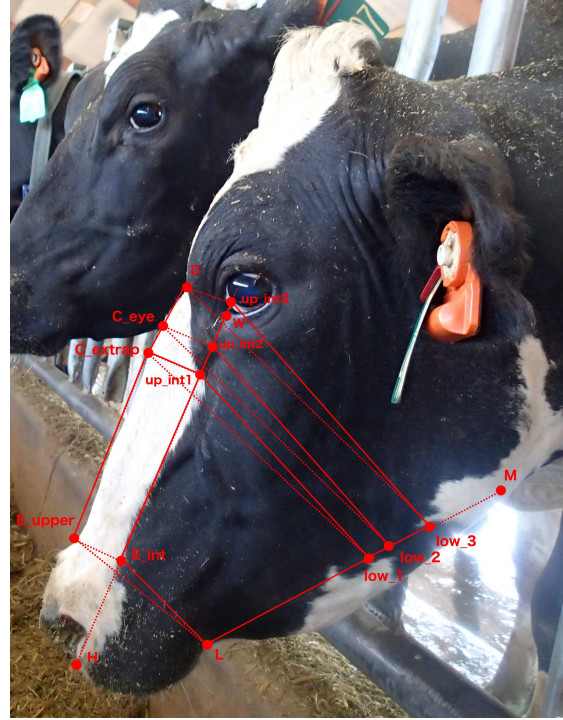
$$V2 = \{up_{int1}, low_1, L_{full}\} \rightarrow \{up_{int}, low, L\}$$

$$V3 = \{up_{int2}, low_2, L\} \rightarrow \{up_{int}, low, L\}$$

$$V4 = \{up_{int2}, low_2, L_{full}\} \rightarrow \{up_{int}, low, L\}$$

$$V5 = \{up_{int3}, low_3, L\} \rightarrow \{up_{int}, low, L\}$$

$$V6 = \{up_{int3}, low_3, L_{full}\} \rightarrow \{up_{int}, low, L\}$$



Jowel-Jaw Length Proportion

$$JJLP = \frac{\|C_{int}, U_{int}\|}{\|L, C_{int}\|}$$

$$V1 = \{C_{extrap}, U_{int1}, L\} \rightarrow \{C, U_{int}, L\}$$

$$V2 = \{C_{extrap}, U_{int2}, L\} \rightarrow \{C, U_{int}, L\}$$

$$V3 = \{C_{eye}, U_{int1}, L\} \rightarrow \{C, U_{int}, L\}$$

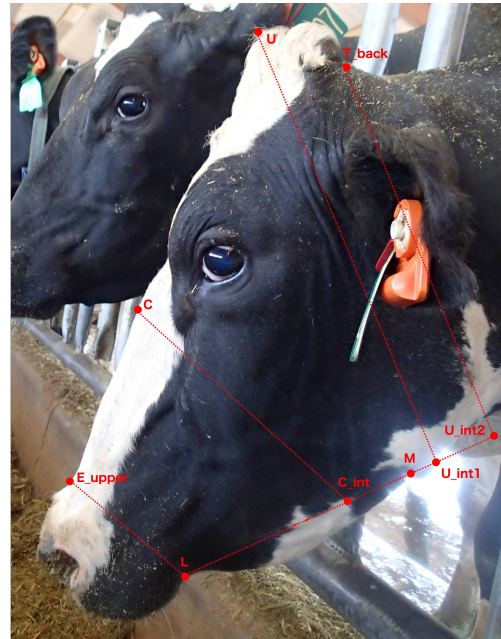
$$V4 = \{C_{eye}, U_{int2}, L\} \rightarrow \{C, U_{int}, L\}$$

$$V5 = \{C_{extrap}, U_{int1}, L_{full}\} \rightarrow \{C, U_{int}, L\}$$

$$V6 = \{C_{extrap}, U_{int2}, L_{full}\} \rightarrow \{C, U_{int}, L\}$$

$$V7 = \{C_{eye}, U_{int1}, L_{full}\} \rightarrow \{C, U_{int}, L\}$$

$$V8 = \{C_{eye}, U_{int2}, L_{full}\} \rightarrow \{C, U_{int}, L\}$$



Jowl Length Proportion

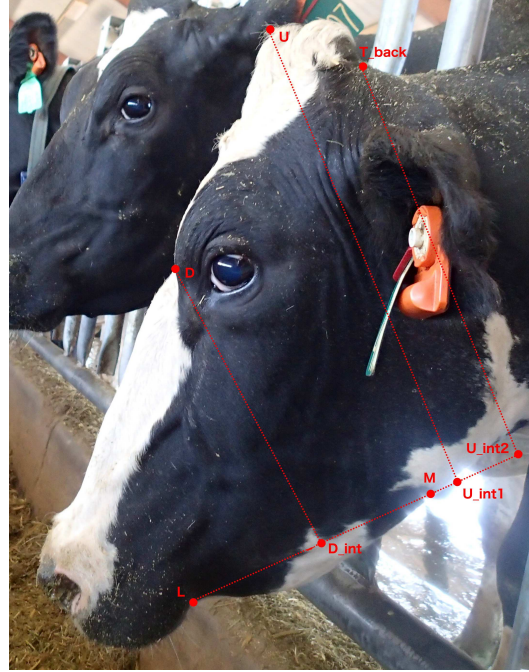
$$JLP = \frac{\|D_{int}, U_{int}\|}{\|L, D_{int}\|}$$

$$V1 = \{U_{int1}, L\} \rightarrow \{U_{int}, L\}$$

$$V2 = \{U_{int2}, L\} \rightarrow \{U_{int}, L\}$$

$$V3 = \{U_{int1}, L_{full}\} \rightarrow \{U_{int}, L\}$$

$$V4 = \{U_{int2}, L_{full}\} \rightarrow \{U_{int}, L\}$$



Midface Divergence Proportion

$$MDP = CF * \frac{\|F_{int2}, F\|}{\|C, F\|}$$

$$CF = \frac{\|F, F_{aux2}\| - \|F_{int2}, F_{aux2}\|}{\|F, F_{aux2}\| - \|F_{int2}, F_{aux2}\|}$$

$$V1 = \{C_{extrap}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V2 = \{C_{eye}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V3 = \{C_{extrap}, F_{eye}\} \rightarrow \{C, F\}$$

$$V4 = \{C_{eye}, F_{eye}\} \rightarrow \{C, F\}$$



Midface Inflection Point Proportion

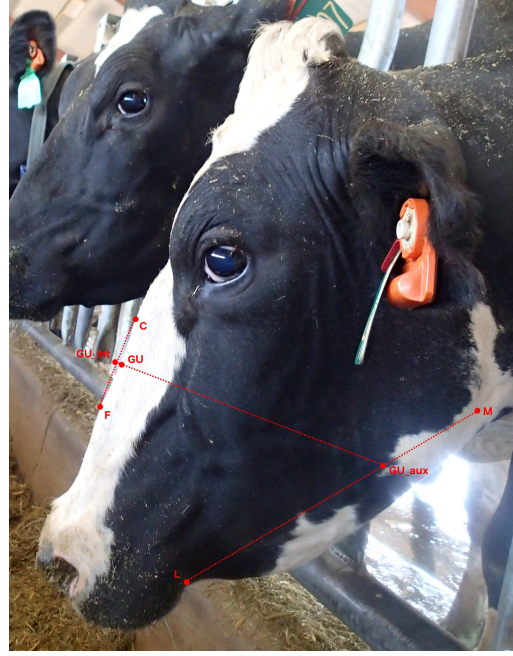
$$MIPP = \frac{\|C, GU_{int}\|}{\|F, C\|}$$

$$V1 = \{C_{extrap}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V2 = \{C_{eye}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V3 = \{C_{extrap}, F_{eye}\} \rightarrow \{C, F\}$$

$$V4 = \{C_{eye}, F_{eye}\} \rightarrow \{C, F\}$$



Midface Inflection Point Proportion

$$MIP = CF * \frac{\|GU, GU_{int}\|}{\|F, C\|}$$

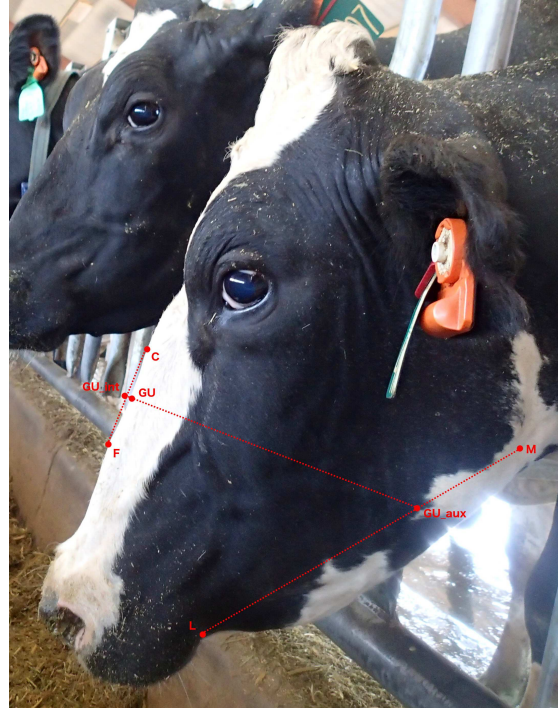
$$CF = \frac{\|GU, GU_{aux}\| - \|GU_{int}, GU_{aux}\|}{\|GU, GU_{aux}\|}$$

$$V1 = \{C_{extrap}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V2 = \{C_{eye}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V3 = \{C_{extrap}, F_{eye}\} \rightarrow \{C, F\}$$

$$V4 = \{C_{eye}, F_{eye}\} \rightarrow \{C, F\}$$



Midface Thickness Proportion

$$MTP = \frac{\|C, C_{int2}\|}{\|C_{int2}, E_{int5}\|}$$

$$V1 = \{C_{extrap}\} \rightarrow \{C\}$$

$$V2 = \{C_{eye}\} \rightarrow \{C\}$$



Midface-Nose Length Proportion

$$MNLP = \frac{\|C_{len}, F_{len}\|}{\|F_{len}, E_{len}\|}$$

$$V1 = \{C_{extrap}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V2 = \{C_{eye}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V3 = \{C_{extrap}, F_{eye}\} \rightarrow \{C, F\}$$

$$V4 = \{C_{eye}, F_{eye}\} \rightarrow \{C, F\}$$



Midface-Nose Roundness Proportion

$$MNRP = CF * \frac{\|E_{int2}, E_{upper}\|}{\|E_{upper}, F\|}$$

$$CF = \frac{\|E_{int2}, E_{aux2}\| - \|E_{upper}, E_{aux2}\|}{\|E_{int2}, E_{aux2}\| - \|E_{upper}, E_{aux2}\|}$$

$$V1 = \{C_{extrap}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V2 = \{C_{eye}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V3 = \{C_{extrap}, F_{eye}\} \rightarrow \{C, F\}$$

$$V4 = \{C_{eye}, F_{eye}\} \rightarrow \{C, F\}$$



Midface-Topline Length Proportion

$$MTLP = \frac{\|C_{len}, F_{len}\|}{\|E_{L_{len}}, D_{len}\|}$$

$$V1 = \{C_{extrap}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V2 = \{C_{eye}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V3 = \{C_{extrap}, F_{eye}\} \rightarrow \{C, F\}$$

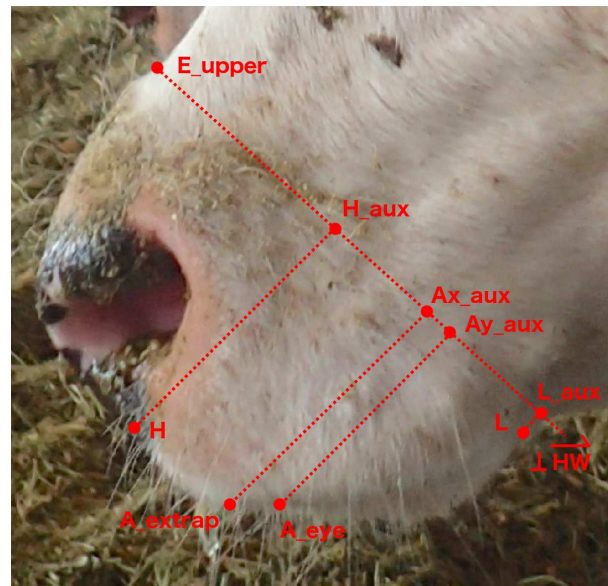
$$V4 = \{C_{eye}, F_{eye}\} \rightarrow \{C, F\}$$



Mouth Eye-to-Extrap Offset_Height

$$MEEOH = CF * \frac{\|Ax_{aux}, Ay_{aux}\|}{\|H_{aux}, L_{aux}\|}$$

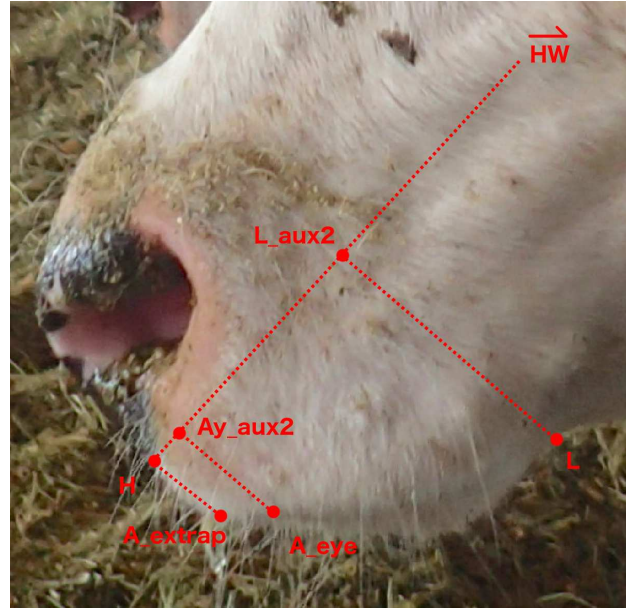
$$CF = \frac{\|H_{aux}, Ay_{aux}\| - \|H_{aux}, Ax_{aux}\|}{\|H_{aux}, Ay_{aux}\| - \|H_{aux}, Ax_{aux}\|}$$



Mouth Eye-to-Extrap Offset Length

$$MEEOL = CF * \frac{\|H, Ay_{aux2}\|}{\|H, L_{aux2}\|}$$

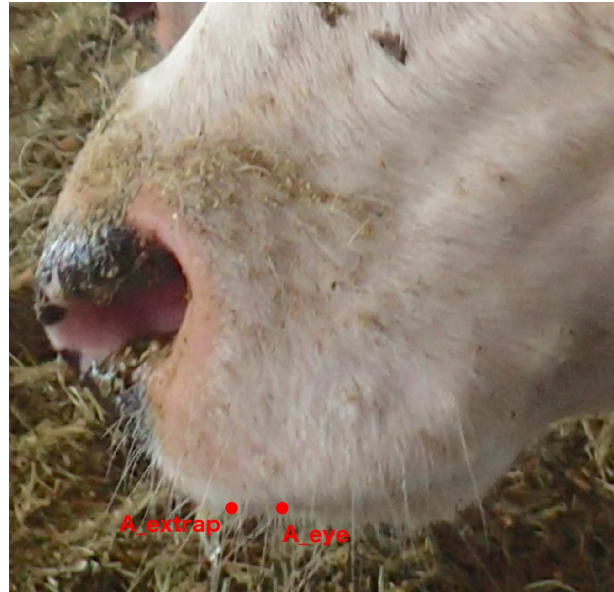
$$CF = \frac{\|Ay_{aux2}, L_{aux2}\| - \|H, L_{aux2}\|}{\text{abs}(\|Ay_{aux2}, L_{aux2}\| - \|H, L_{aux2}\|)}$$



Mouth Eye-to-Extrap Offset Slope

$$MEEOSlope = \frac{(extrap_y - eye_y)}{(extrap_x - eye_x)}$$

$$MEEOSlope = \frac{(extrap_y - eye_y)}{\|A_{extrap} - A_{eye}\|}$$

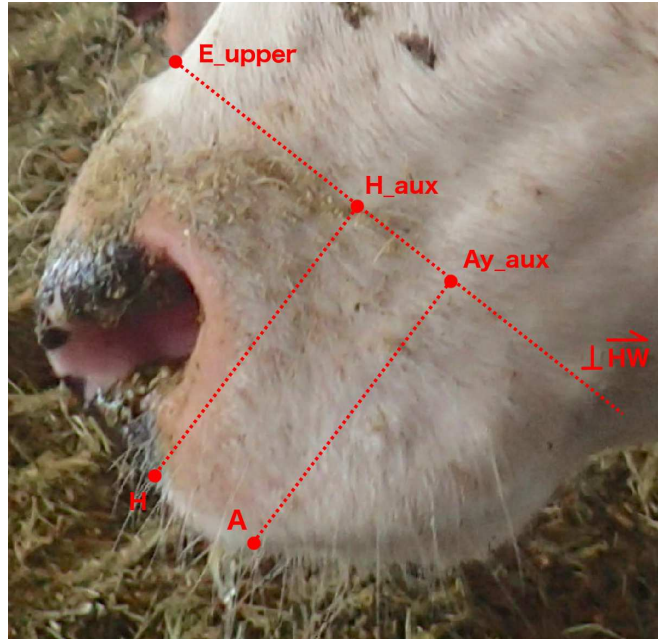


Mouth Thickness Proportion

$$MTP = \frac{\|H_{aux}, Ax_{aux}\|}{\|Ax_{aux}, E_{upper}\|}$$

$$V1 = \{A_{extrap}, L\} \rightarrow \{A, L\}$$

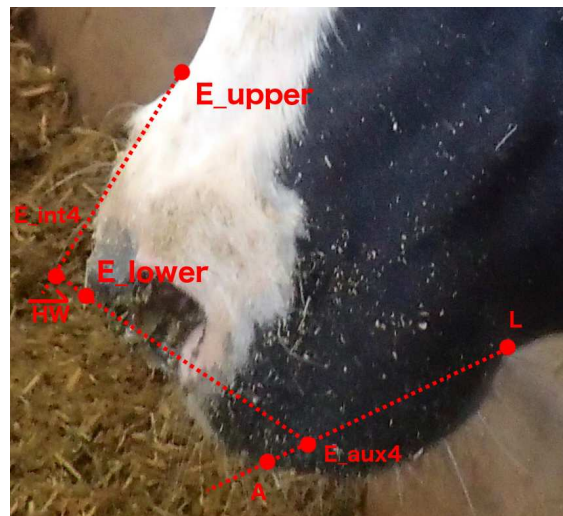
$$V2 = \{A_{eye}, L_{full}\} \rightarrow \{A, L\}$$



Nares Divergence Proportion_V1

$$NaDP = CF * \frac{\|E_{lower}, E_{int4}\|}{\|E_{upper}, E_{lower}\|}$$

$$CF = \frac{\|E_{lower}, E_{aux4}\| - \|E_{int4}, E_{aux4}\|}{\|E_{lower}, E_{aux4}\| - \|E_{int4}, E_{aux4}\|}$$



Muzzle Size Proportion

$$MSP = \frac{A(ALE_{upper}E_{lower})}{A(LCE_{int}CE_{upper})}$$

$$V1 = \{C_{extrap}, A_{extrap}, L\} \rightarrow \{C, A, L\}$$

$$V2 = \{C_{extrap}, A_{extrap}, L_{full}\} \rightarrow \{C, A, L\}$$

$$V3 = \{C_{extrap}, A_{eye}, L\} \rightarrow \{C, A, L\}$$

$$V4 = \{C_{extrap}, A_{eye}, L_{full}\} \rightarrow \{C, A, L\}$$

$$V5 = \{C_{extrap}, A_{extrap}, L\} \rightarrow \{C, A, L\}$$

$$V6 = \{C_{extrap}, A_{extrap}, L_{full}\} \rightarrow \{C, A, L\}$$

$$V7 = \{C_{extrap}, A_{eye}, L\} \rightarrow \{C, A, L\}$$

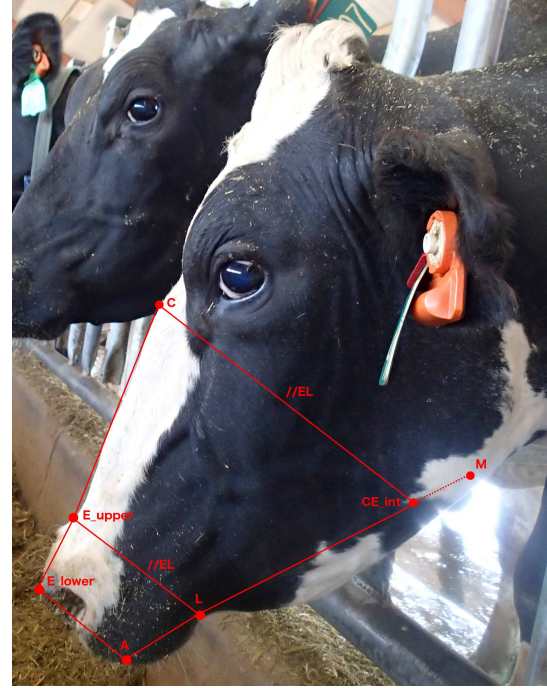
$$V8 = \{C_{extrap}, A_{eye}, L_{full}\} \rightarrow \{C, A, L\}$$

$$V9 = \{D, A_{extrap}, L\} \rightarrow \{C, A, L\}$$

$$V10 = \{D, A_{extrap}, L_{full}\} \rightarrow \{C, A, L\}$$

$$V11 = \{D, A_{eye}, L\} \rightarrow \{C, A, L\}$$

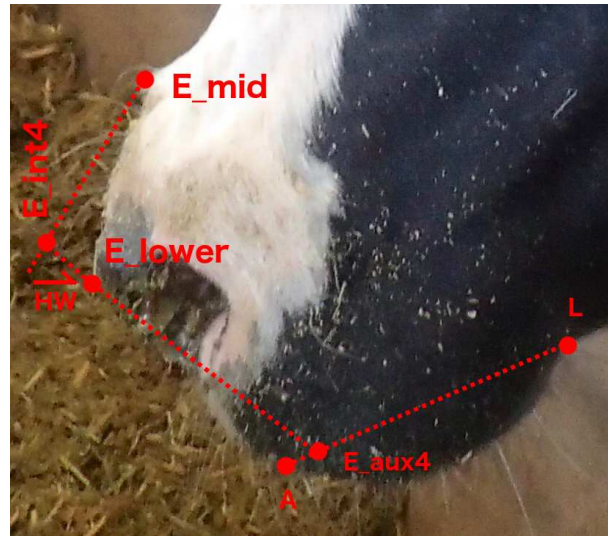
$$V12 = \{D, A_{eye}, L_{full}\} \rightarrow \{C, A, L\}$$



Nares Divergence Proportion V2

$$NaDP = CF * \frac{\|E_{lower}, E_{int4}\|}{\|E_{mid}, E_{int4}\|}$$

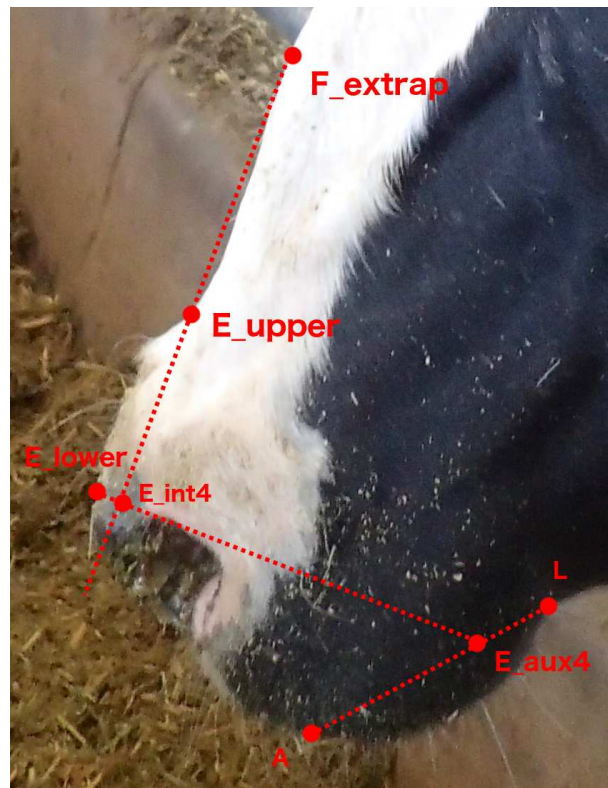
$$CF = \frac{\|E_{lower}, E_{aux4}\| - \|E_{int4}, E_{aux4}\|}{\|E_{lower}, E_{aux4}\| - \|E_{int4}, E_{aux4}\|}$$



Nares Divergence Proportion V3

$$NaDP = CF * \frac{\|E_{lower}, E_{int4}\|}{\|E_{upper}, E_{int4}\|}$$

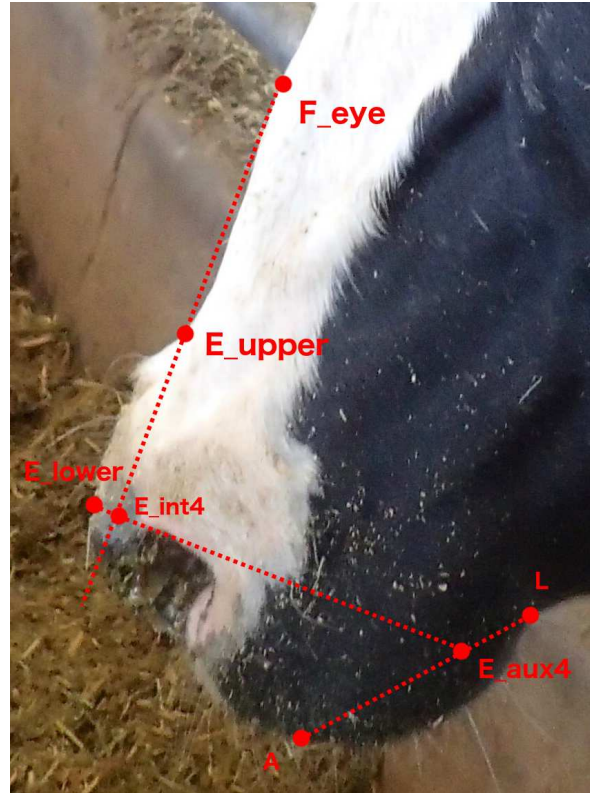
$$CF = \frac{\|E_{lower}, E_{aux4}\| - \|E_{int4}, E_{aux4}\|}{\|E_{lower}, E_{aux4}\| - \|E_{int4}, E_{aux4}\|}$$



Nares Divergence Proportion V4

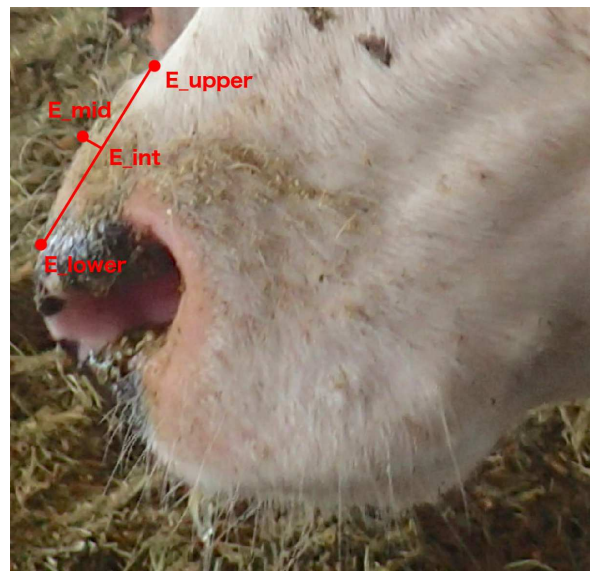
$$NaDP = CF * \frac{\|E_{lower}, E_{int4}\|}{\|E_{upper}, E_{int4}\|}$$

$$CF = \frac{\|E_{lower}, E_{aux4}\| - \|E_{int4}, E_{aux4}\|}{\|E_{lower}, E_{aux4}\| - \|E_{int4}, E_{aux4}\|}$$



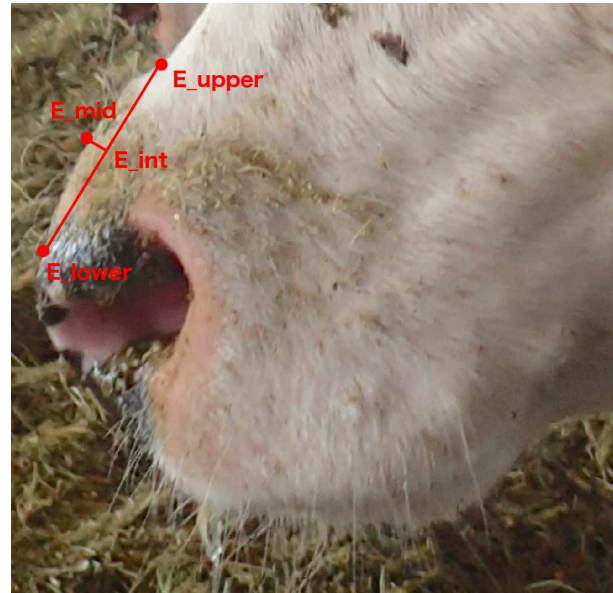
Nares Roundness Point Proportion

$$NRPP = \frac{\|E_{upper}, E_{int}\|}{\|E_{upper}, E_{lower}\|}$$



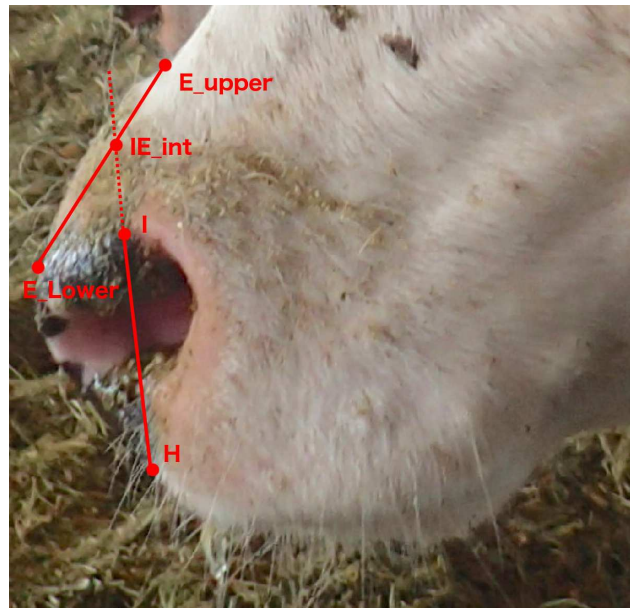
Nares Roundness Proportion

$$NRP = \frac{\|E_{mid}, E_{int}\|}{\|E_{upper}, E_{lower}\|}$$



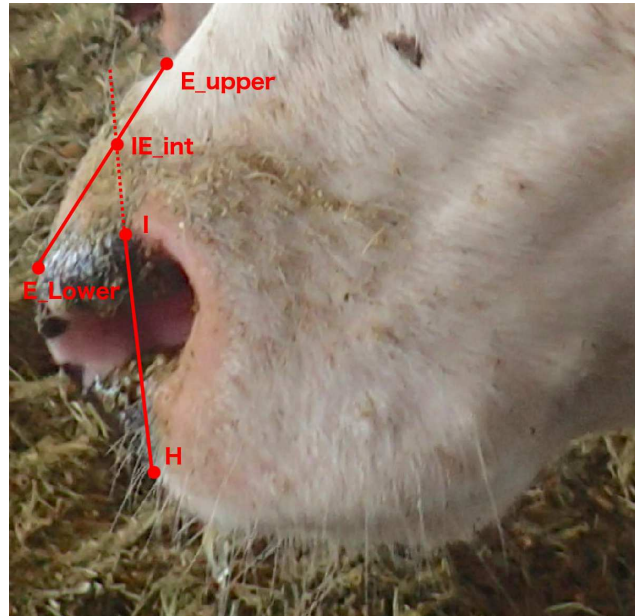
Nares Thickness Proportion V1

$$NTP = \frac{\|I, IE_{int}\|}{\|E_{lower}, E_{upper}\|}$$



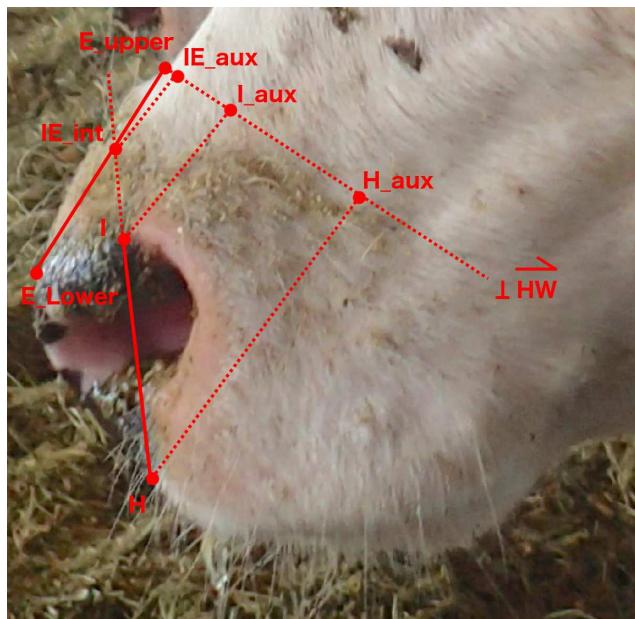
Nares Thickness Proportion_V1

$$NTP = \frac{\|I, IE_{int}\|}{\|I, H\|}$$



Nares Thickness Proportion_V3

$$NTP = \frac{\|I_{aux}, IE_{aux}\|}{\|IE_{aux}, H_{aux}\|}$$



Nares-Nose Length Proportion

$$NNLP = \frac{\|EL_{len}, E_{len}\|}{\|E_{len}, F_{len}\|}$$

$$V1 = \{F_{extrap}\} \rightarrow \{F\}$$

$$V2 = \{F_{eye}\} \rightarrow \{F\}$$



Nares-Topline Length Proportion

$$NaTLP = \frac{\|F_{eye}, GL_{int}\|}{\|F_{eye}, E_{upper}\|}$$



Nasion Thickness Proportion

$$NsTP = \frac{\|G, G_{int}\|}{\|Y, X\|}$$

$$V1 = \{ca, aa\} \rightarrow \{Y, W\}$$

$$V2 = \{ca, ab\} \rightarrow \{Y, W\}$$

$$V3 = \{ca, ac\} \rightarrow \{Y, W\}$$

$$V4 = \{cb, aa\} \rightarrow \{Y, W\}$$

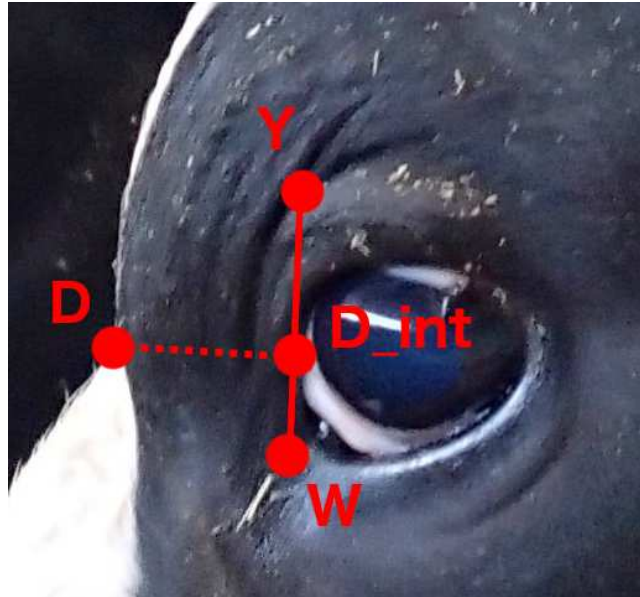
$$V5 = \{cb, ab\} \rightarrow \{Y, W\}$$

$$V6 = \{cb, ac\} \rightarrow \{Y, W\}$$

$$V7 = \{cc, aa\} \rightarrow \{Y, W\}$$

$$V8 = \{cc, ab\} \rightarrow \{Y, W\}$$

$$V9 = \{cc, ac\} \rightarrow \{Y, W\}$$



Nose Divergence Proportion

$$NDP = CF * \frac{\|E_{upper}, E_{int3}\|}{\|E_{upper}, F_{extrap}\|}$$

$$CF = \frac{\|E_{upper}, E_{aux3}\| - \|E_{int3}, E_{aux3}\|}{\|E_{upper}, E_{aux3}\| - \|E_{int3}, E_{aux3}\|}$$

$$V1 = \{F_{extrap}\} \rightarrow \{F\}$$

$$V2 = \{F_{eye}\} \rightarrow \{F\}$$



Nose Inflection Point Proportion

$$NIPP = \frac{\|F, GL_{int}\|}{\|F, E_{upper}\|}$$

$$V1 = \{F_{extrap}\} \rightarrow \{F\}$$

$$V2 = \{F_{eye}\} \rightarrow \{F\}$$



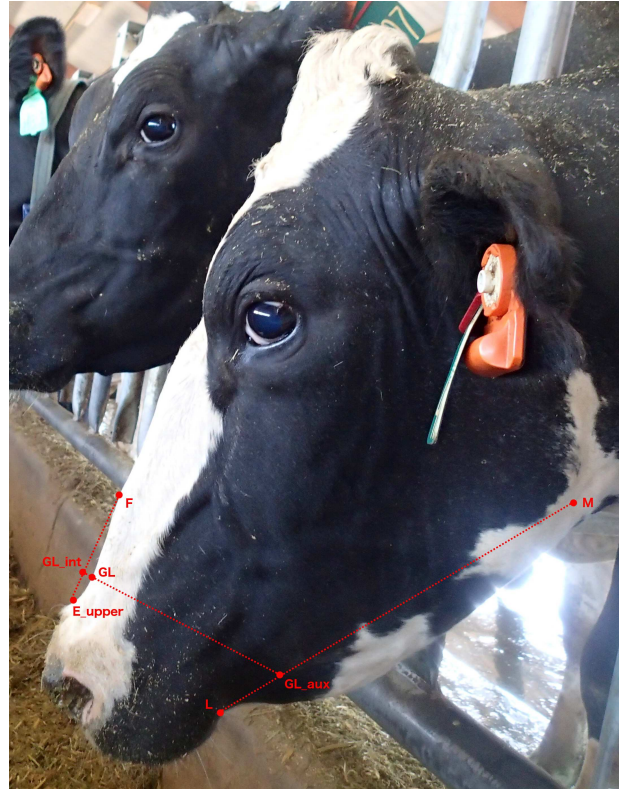
Nose Inflection Proportion

$$NIP = CF * \frac{\|GL, GL_{int}\|}{\|F, E_{upper}\|}$$

$$CF = \frac{\|GL, GL_{aux}\| - \|GL_{int}, GL_{aux}\|}{\|GL, GL_{aux}\| - \|GL_{int}, GL_{aux}\|}$$

$$V1 = \{F_{extrap}\} \rightarrow \{F\}$$

$$V2 = \{F_{eye}\} \rightarrow \{F\}$$



Nose-Topline Length Proportion

$$NTLP = \frac{\|E_{len}, F_{len}\|}{\|EL_{len}, D_{len}\|}$$

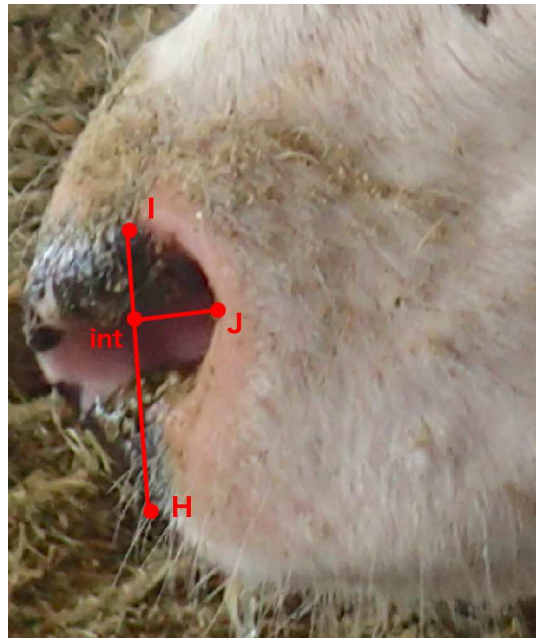
$$V1 = \{F_{extrap}\} \rightarrow \{F\}$$

$$V2 = \{F_{eye}\} \rightarrow \{F\}$$



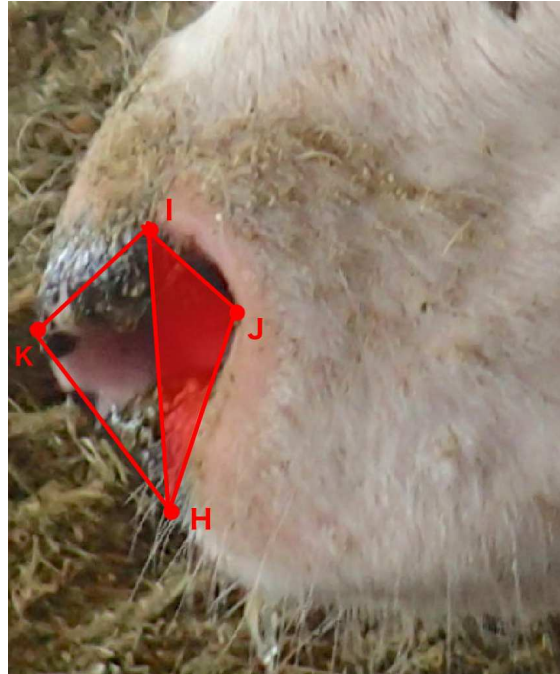
Nostril Depth Point Proportion

$$NDPP = \frac{\|I, j_{int}\|}{\overline{HI}}$$



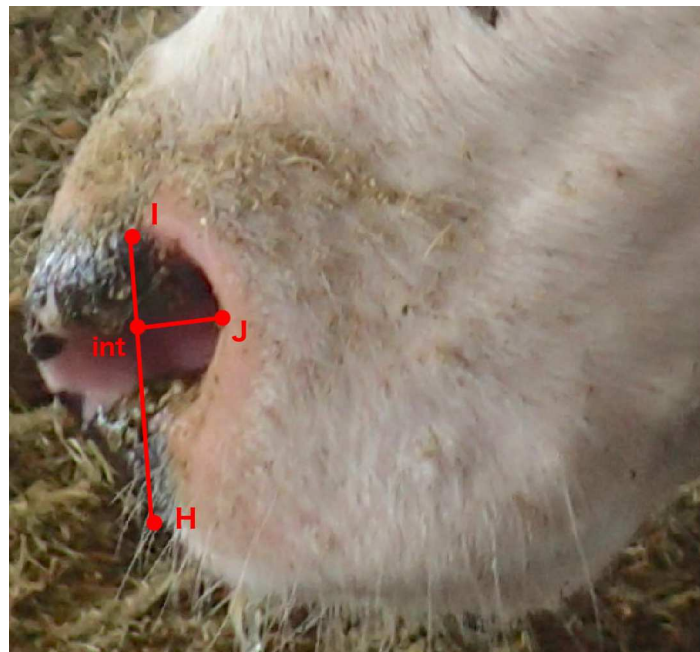
Nostril Depth Proportion Area

$$NDP = \frac{A(I, H, J)}{A(I, J, H, K)}$$



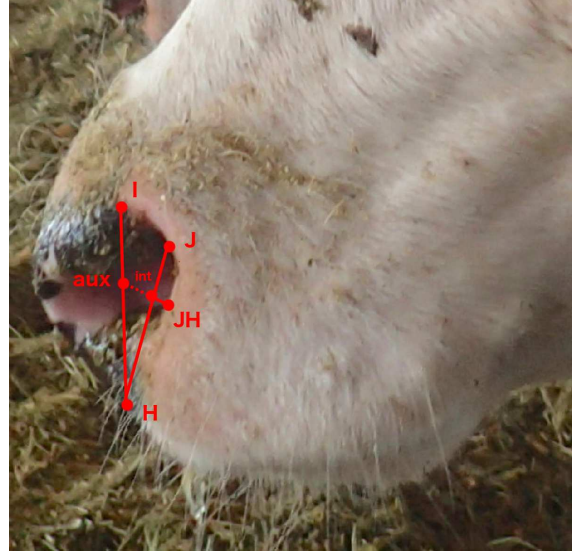
Nostril Depth Proportion Linear

$$NDP = \frac{\|J, J_{int}\|}{\|H, I\|}$$



Nostril Flare Point Proportion – Lower Back

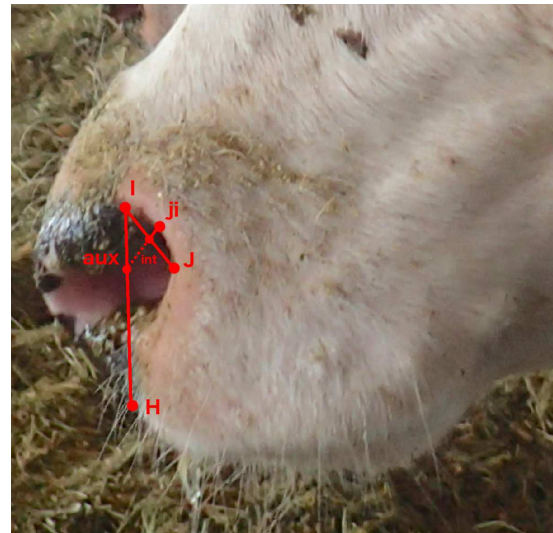
$$NFPP_LB = \frac{\|jh_int, H\|}{\|J, H\|}$$



Nostril Flare Proportion – Upper Back

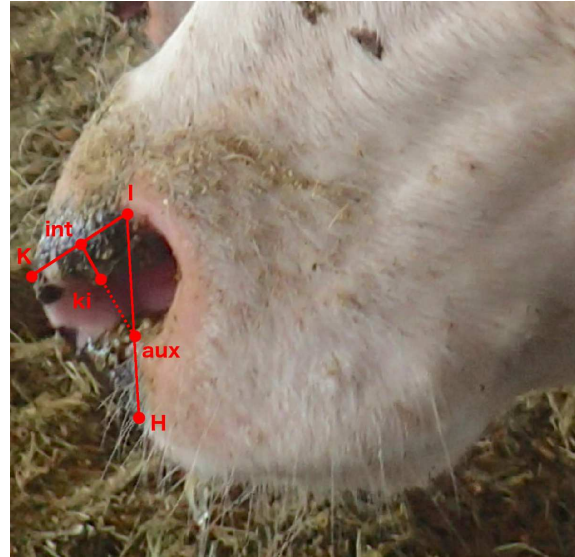
$$NFP_UB = CF * \frac{\|ji, ji_int\|}{\|J, I\|}$$

$$CF = \frac{\|ji_aux, ji\| - \|ji_aux, ji_int\|}{\| \|ji_aux, ji\| - \|ji_aux, ji_int\| \|}$$



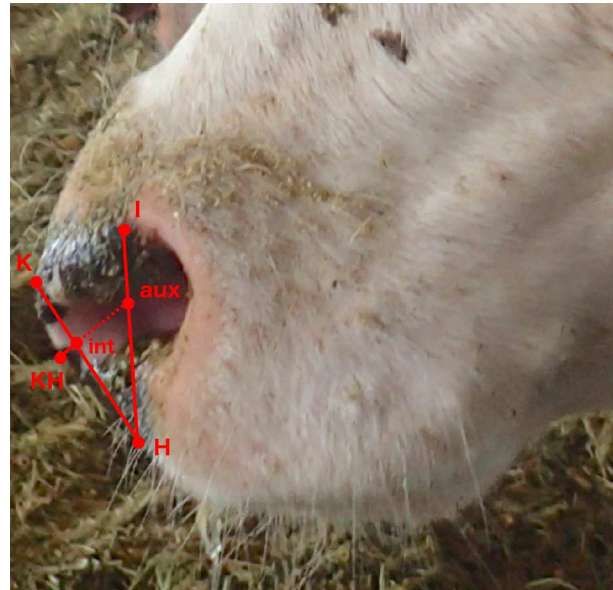
Nostril Flare Point Proportion – Upper Front

$$NFPP_{UF} = \frac{\|I, ki_int\|}{\|K, I\|}$$



Nostril Flare Point Proportion – Lower Front

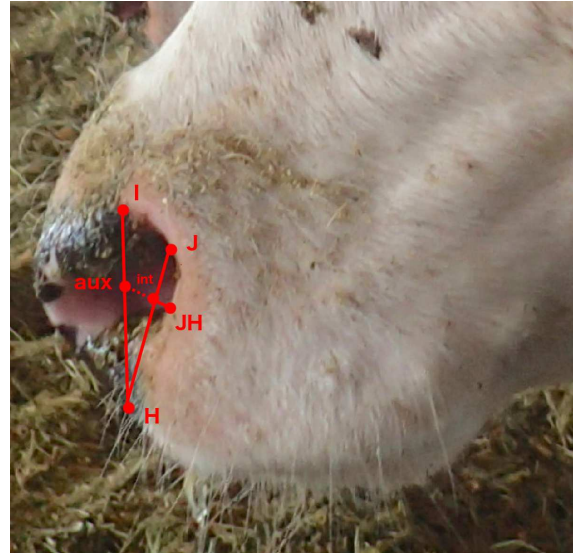
$$NFPP_{LF} = \frac{\|H, kh_int\|}{\|K, H\|}$$



Nostril Flare Proportion – Lower Back

$$NFP_{LB} = CF * \frac{\|jh, jh_{int}\|}{\|J, H\|}$$

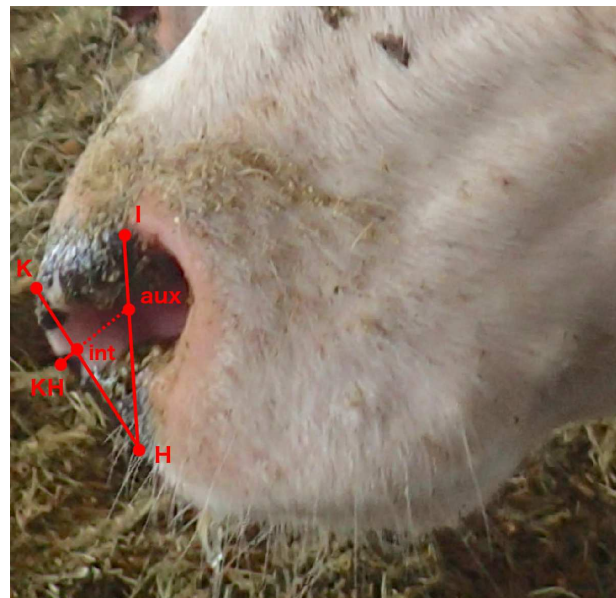
$$CF = \frac{\|jh_{aux}, jh\| - \|jh_{aux}, jh_{int}\|}{\|jh_{aux}, jh\| - \|jh_{aux}, jh_{int}\|}$$



Nostril Flare Proportion – Lower Front

$$NFP_{LF} = CF * \frac{\|kh, kh_{int}\|}{\|K, H\|}$$

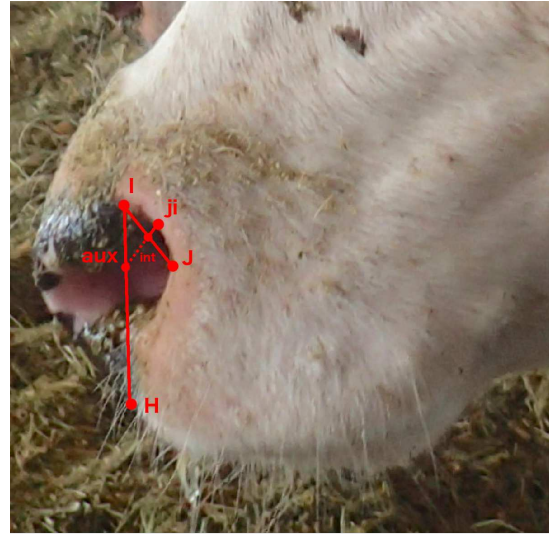
$$CF = \frac{\|kh_{aux}, kh_{int}\| - \|kh_{aux}, kh\|}{\|kh_{aux}, kh_{int}\| - \|kh_{aux}, kh\|}$$



Nostril Flare Proportion – Upper Back

$$NFP_{UB} = CF * \frac{\|ji, ji_{int}\|}{\|J, I\|}$$

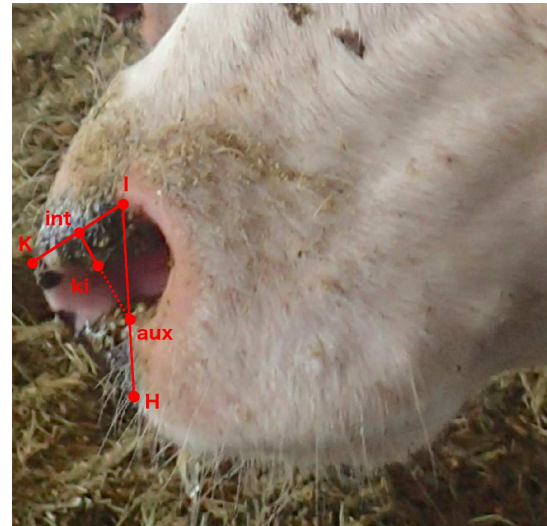
$$CF = \frac{\|ji_{aux}, ji\| - \|ji_{aux}, ji_{int}\|}{\| \|ji_{aux}, ji\| - \|ji_{aux}, ji_{int}\| \|}$$



Nostril Flare Proportion – Upper Front

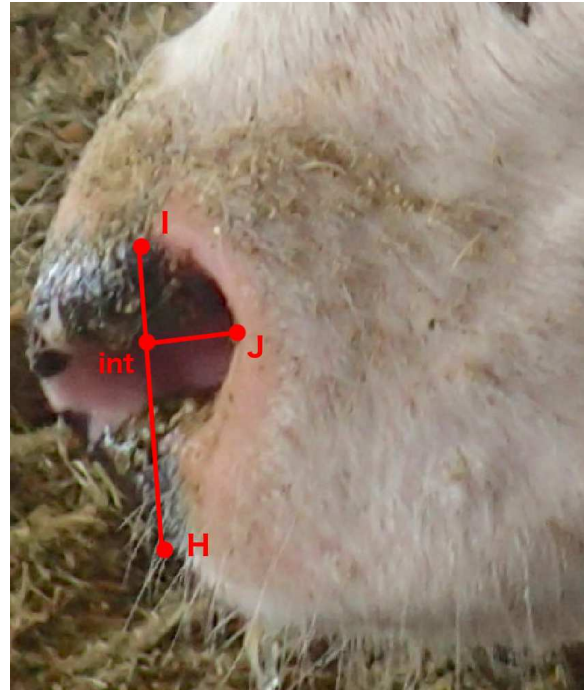
$$NFP_{UF} = CF * \frac{\|ki, ki_{int}\|}{\|K, I\|}$$

$$CF = \frac{\|ki_{aux}, ki_{int}\| - \|ki, ki_{aux}\|}{\| \|ki_{aux}, ki_{int}\| - \|ki, ki_{aux}\| \|}$$



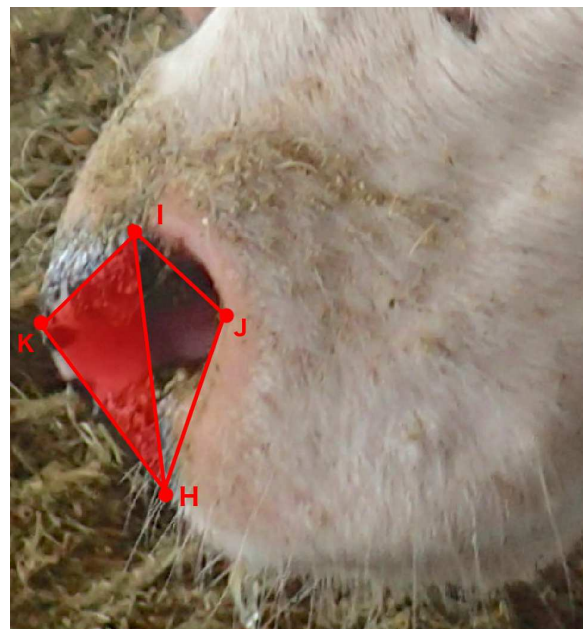
Nostril Depth Point Proportion

$$NDPP = \frac{\|I, j_{int}\|}{\overline{HI}}$$



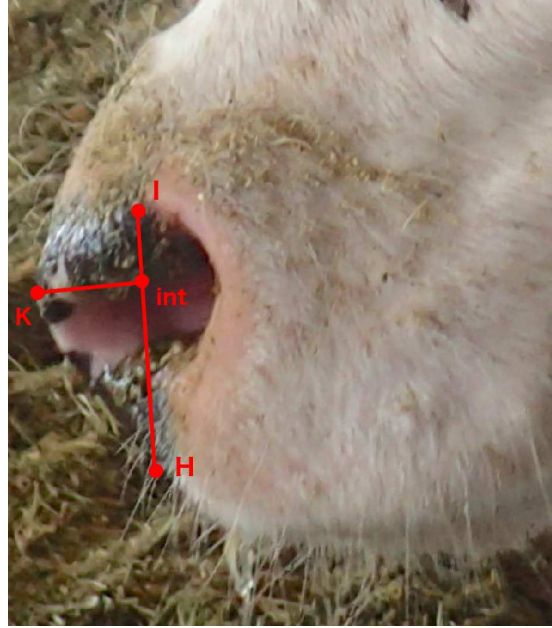
Nostril Height Proportion Area

$$NHP = \frac{A(K, H, I)}{A(I, J, H, K)}$$



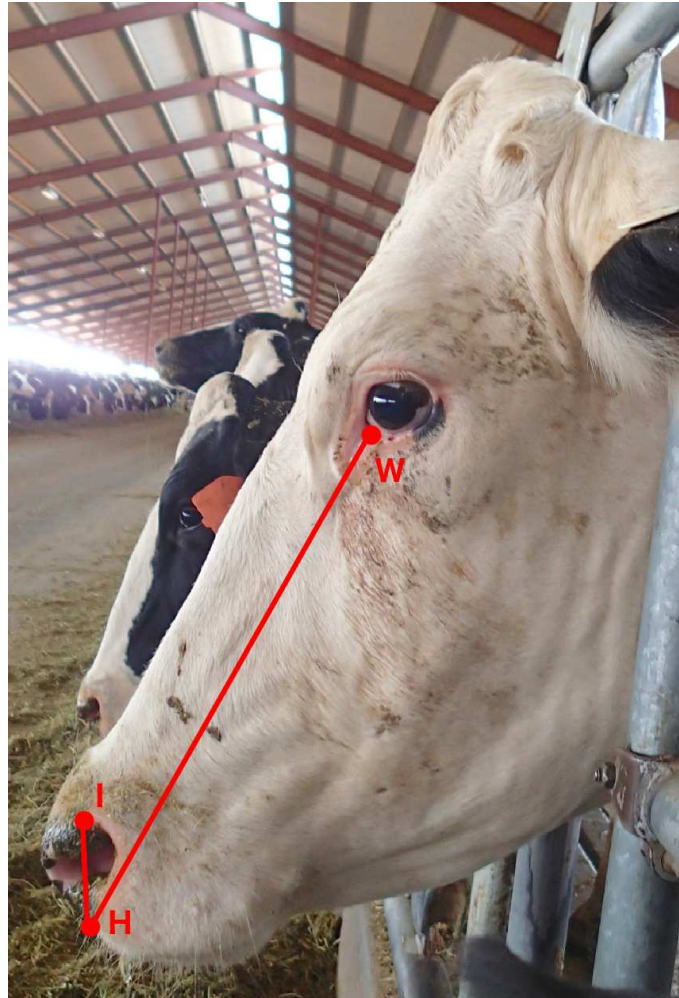
Nostril Height Proportion_Linear

$$NHP = \frac{\|k, k_{int}\|}{\overline{HI}}$$



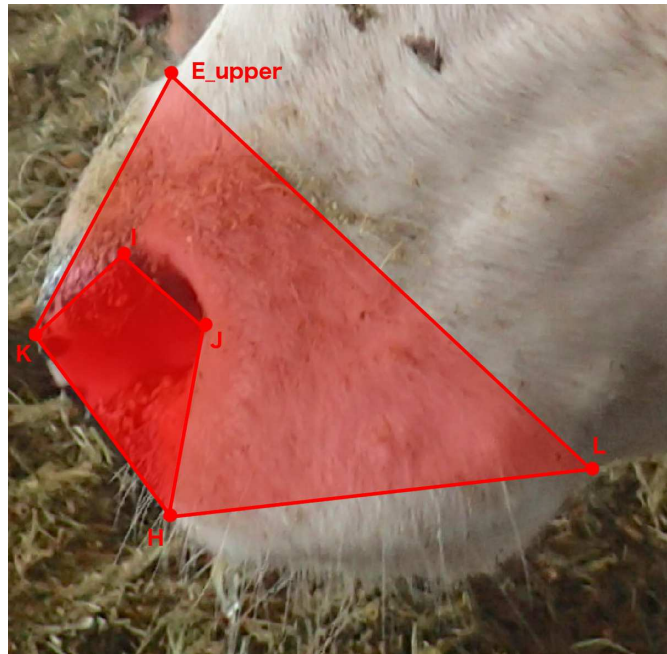
Nostril Position Angle

$$NPA = \angle IHW$$



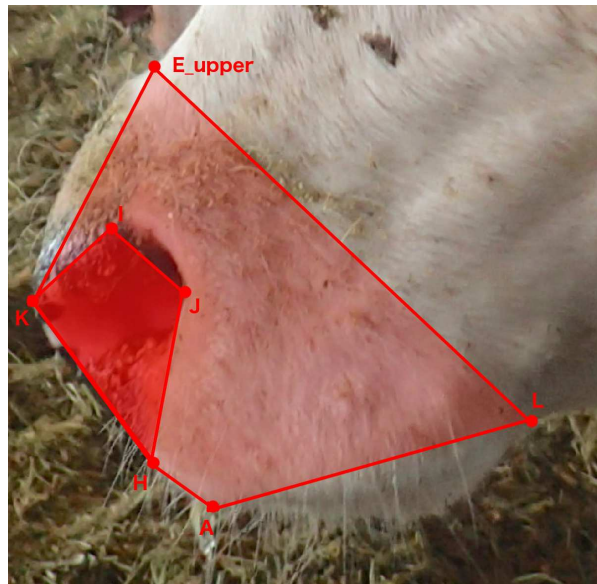
Nostril-Muzzle Ratio_Area_V1

$$NPA = \frac{A(KHJI)}{A(E_{upper}, K, H, L)}$$



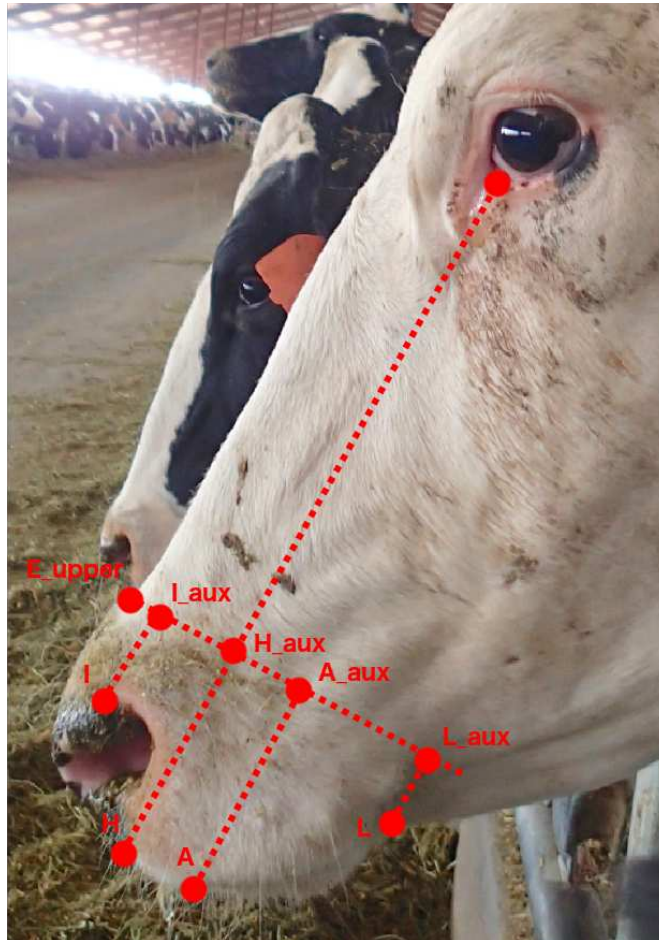
Nostril-Muzzle Ratio_Area_V2

$$NPA = \frac{A(KHJI)}{A(E_{upper}, K, H, A, L)}$$



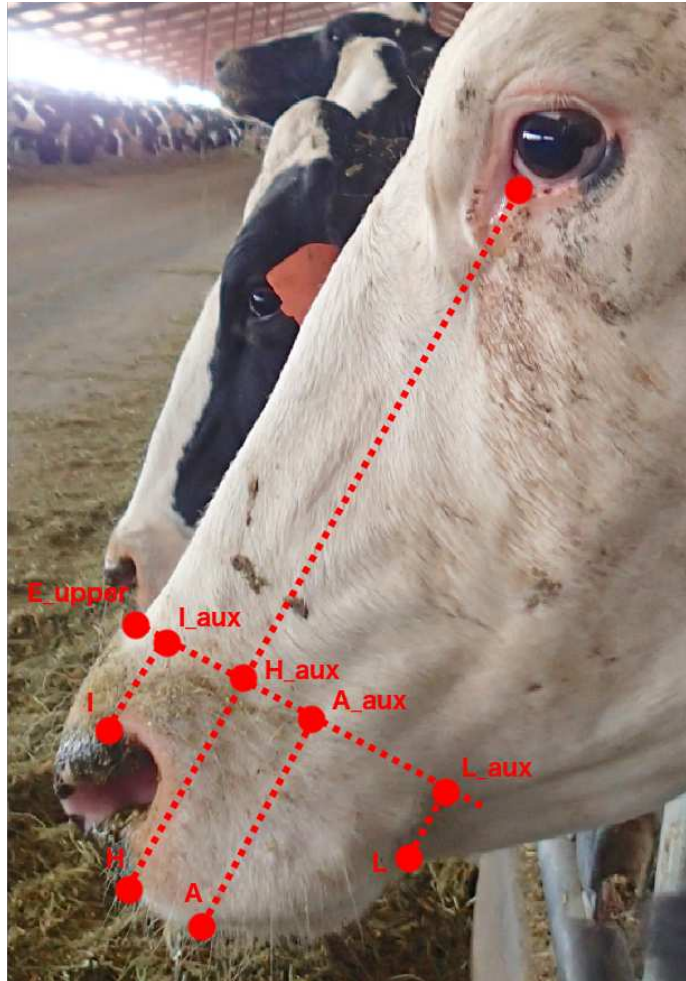
Nostril-Muzzle Ratio_Height_V1

$$NMRH = \frac{\|I_{aux}, H_{aux}\|}{\|E_{upper}, A_{aux}\|}$$



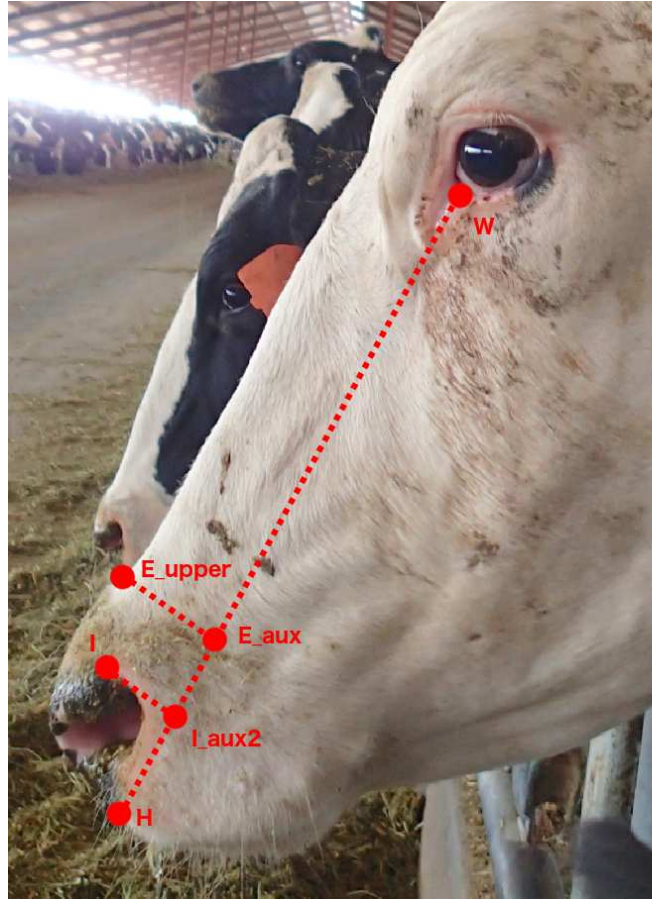
Nostril-Muzzle Ratio_Height_V2

$$NMRH = \frac{\|I_{aux}, H_{aux}\|}{\|E_{upper}, L_{aux}\|}$$



Nostril-Muzzle Ratio_Length

$$NMRL = \frac{\|H, I_{aux2}\|}{\|H, E_{aux}\|}$$



Overall Eye Angle_Angle

$$OEAA = \angle WXW_{int}$$

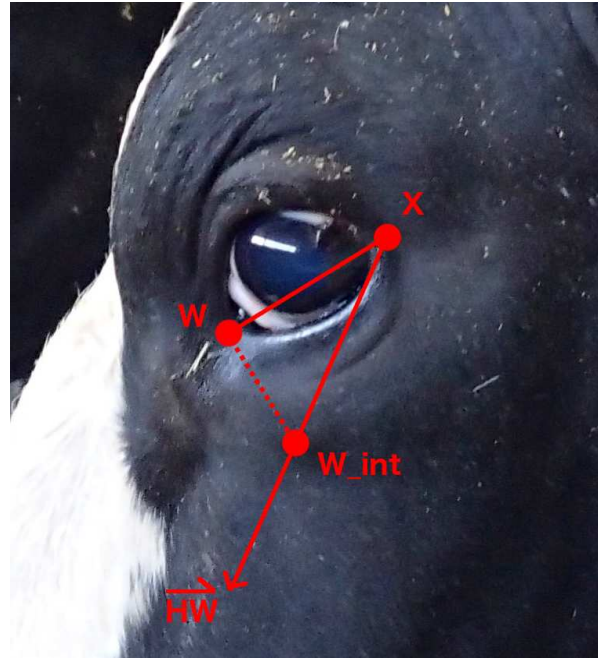
$$V1 = \{aa, ba\} \rightarrow \{W, X\}$$

$$V2 = \{ab, bb\} \rightarrow \{W, X\}$$

$$V3 = \{ac, bc\} \rightarrow \{W, X\}$$

$$V4 = \{aa, bb\} \rightarrow \{W, X\}$$

$$V5 = \{ab, ba\} \rightarrow \{W, X\}$$



Overall Eye Angle_Slope

$$OEAS = \frac{\|W, W_{int}\|}{\|W, X\|}$$

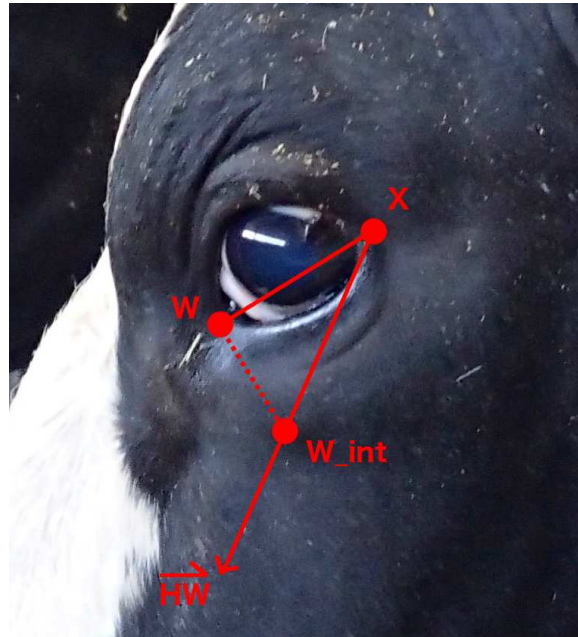
$$V1 = \{aa, ba\} \rightarrow \{W, X\}$$

$$V2 = \{ab, bb\} \rightarrow \{W, X\}$$

$$V3 = \{ac, bc\} \rightarrow \{W, X\}$$

$$V4 = \{aa, bb\} \rightarrow \{W, X\}$$

$$V5 = \{ab, ba\} \rightarrow \{W, X\}$$



Overall Eye Size – Poly

$$OES = \frac{A(WYXZ)}{A(LU_{int}T_{int}T_{fore}E_{upper})}$$

$$V1 = \{aa, ba, ca, da\} \rightarrow \{W, X, Y, Z\}$$

$$V2 = \{aa, ba, cb, db\} \rightarrow \{W, X, Y, Z\}$$

$$V3 = \{aa, ba, cc, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V4 = \{aa, ba, cb, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V5 = \{aa, ba, cc, db\} \rightarrow \{W, X, Y, Z\}$$

$$V6 = \{ab, bb, ca, da\} \rightarrow \{W, X, Y, Z\}$$

$$V7 = \{ab, bb, cb, db\} \rightarrow \{W, X, Y, Z\}$$

$$V8 = \{ab, bb, cc, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V9 = \{ab, bb, cb, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V10 = \{ab, bb, cc, db\} \rightarrow \{W, X, Y, Z\}$$

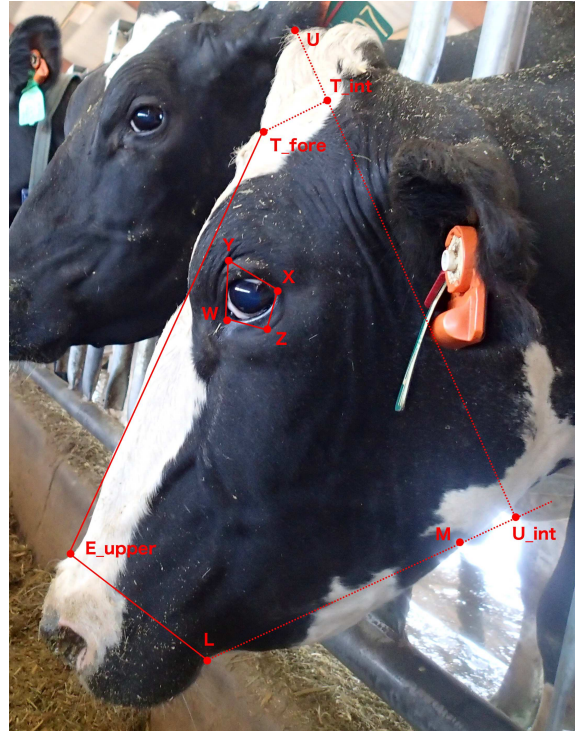
$$V11 = \{ac, bc, ca, da\} \rightarrow \{W, X, Y, Z\}$$

$$V12 = \{ac, bc, cb, db\} \rightarrow \{W, X, Y, Z\}$$

$$V13 = \{ac, bc, cc, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V14 = \{ac, bc, cb, dc\} \rightarrow \{W, X, Y, Z\}$$

$$V15 = \{ac, bc, cc, db\} \rightarrow \{W, X, Y, Z\}$$



Poll Depth Proportion Height

$$PDPH = CF * \frac{\|T_{back}, T_{int}\|}{\|U, T_{mid}\|}$$

$$CF = \frac{\|T_{int}, T_{aux}\| - \|T_{back}, T_{aux}\|}{\| \|T_{int}, T_{aux}\| - \|T_{back}, T_{aux}\| \|}$$

$$V1 = \{S_{extrap}, T_{slope}\} \rightarrow \{S, T\}$$

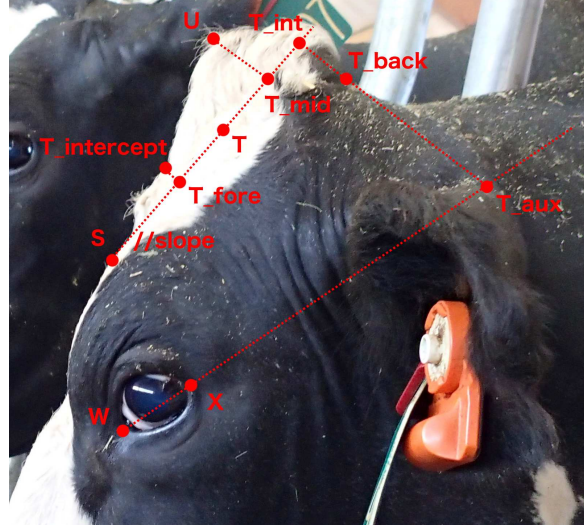
$$V2 = \{S_{extrap}, T_{top}\} \rightarrow \{S, T\}$$

$$V3 = \{S_{extrap}, T_{poll}\} \rightarrow \{S, T\}$$

$$V4 = \{S_{eye}, T_{slope}\} \rightarrow \{S, T\}$$

$$V5 = \{S_{eye}, T_{top}\} \rightarrow \{S, T\}$$

$$V6 = \{S_{eye}, T_{poll}\} \rightarrow \{S, T\}$$



Poll Depth Proportion Length

$$PDPL = CF * \frac{\|T_{back}, T_{int}\|}{\|T_{fore}, T_{int}\|}$$

$$CF = \frac{\|T_{int}, T_{aux}\| - \|T_{back}, T_{aux}\|}{\| \|T_{int}, T_{aux}\| - \|T_{back}, T_{aux}\| \|}$$

$$V1 = \{S_{extrap}, T_{slope}\} \rightarrow \{S, T\}$$

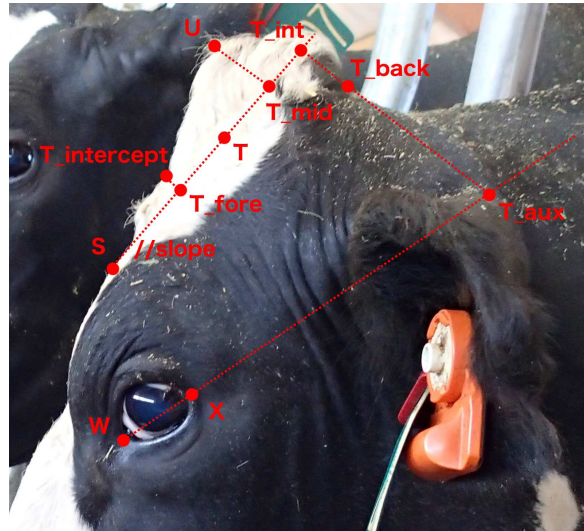
$$V2 = \{S_{extrap}, T_{top}\} \rightarrow \{S, T\}$$

$$V3 = \{S_{extrap}, T_{poll}\} \rightarrow \{S, T\}$$

$$V4 = \{S_{eye}, T_{slope}\} \rightarrow \{S, T\}$$

$$V5 = \{S_{eye}, T_{top}\} \rightarrow \{S, T\}$$

$$V6 = \{S_{eye}, T_{poll}\} \rightarrow \{S, T\}$$



Poll Height Point Proportion

$$PHPP = \frac{\|T_{fore}, T_{mid}\|}{\|T_{fore}, T_{int}\|}$$

$$V1 = \{S_{extrap}, T_{int1}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V2 = \{S_{extrap}, T_{int1}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V3 = \{S_{extrap}, T_{int1}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V4 = \{S_{extrap}, T_{int2}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V5 = \{S_{extrap}, T_{int2}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V6 = \{S_{extrap}, T_{int2}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V7 = \{S_{eye}, T_{int1}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

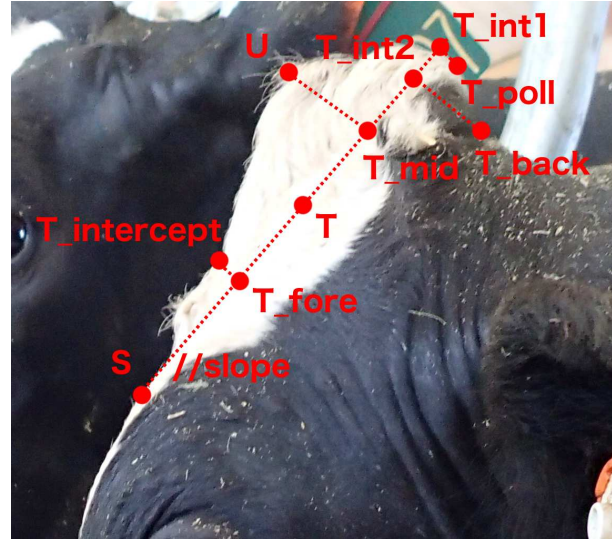
$$V8 = \{S_{eye}, T_{int1}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V9 = \{S_{eye}, T_{int1}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V10 = \{S_{eye}, T_{int2}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V11 = \{S_{eye}, T_{int2}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V12 = \{S_{eye}, T_{int2}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$



Poll Height Proportion

$$PHP = \frac{\|U, T_{mid}\|}{\|T_{fore}, T_{int}\|}$$

$$V1 = \{S_{extrap}, T_{int1}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V2 = \{S_{extrap}, T_{int1}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V3 = \{S_{extrap}, T_{int1}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V4 = \{S_{extrap}, T_{int2}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

$$V5 = \{S_{extrap}, T_{int2}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V6 = \{S_{extrap}, T_{int2}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V7 = \{S_{eye}, T_{int1}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

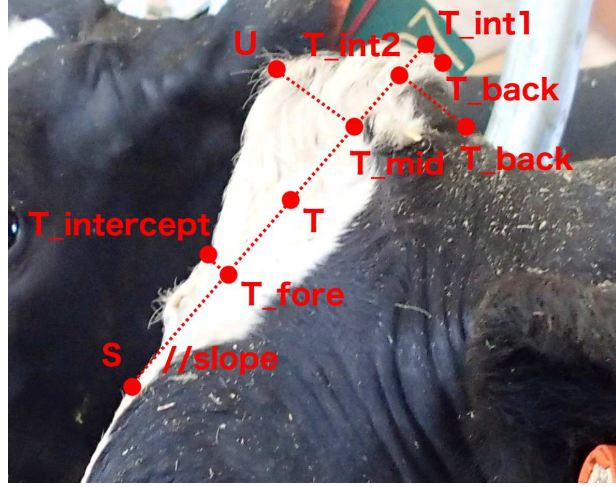
$$V8 = \{S_{eye}, T_{int1}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V9 = \{S_{eye}, T_{int1}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$

$$V10 = \{S_{eye}, T_{int2}, T_{slope}\} \rightarrow \{S, T_{int}, T\}$$

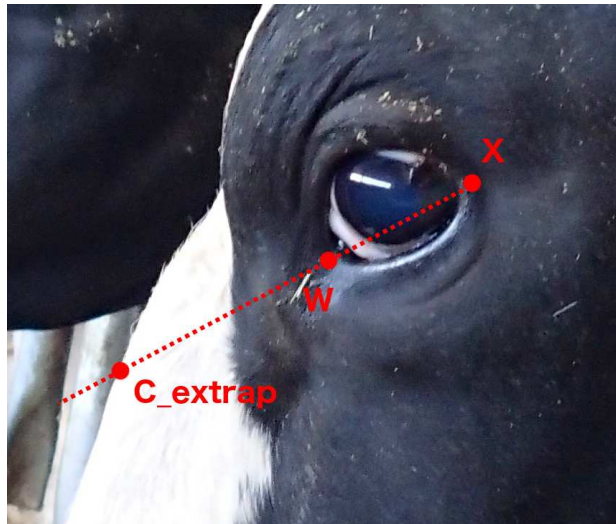
$$V11 = \{S_{eye}, T_{int2}, T_{top}\} \rightarrow \{S, T_{int}, T\}$$

$$V12 = \{S_{eye}, T_{int2}, T_{poll}\} \rightarrow \{S, T_{int}, T\}$$



Sinus Projection Proportion

$$SPP = \frac{\|C_{extrap}, W\|}{\|W, X\|}$$



Sinus-Midface Length Proportion

$$SMLP = \frac{\|C_{len}, F_{len}\|}{\|C_{len}, D_{len}\|}$$

$$V1 = \{C_{extrap}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V2 = \{C_{eye}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V3 = \{C_{extrap}, F_{eye}\} \rightarrow \{C, F\}$$

$$V4 = \{C_{eye}, F_{eye}\} \rightarrow \{C, F\}$$



Sinus-Midface Rounding Proportion

$$SMRP = CF * \frac{\|D_{int}, D\|}{\|C_{extrap}, D_{int}\|}$$

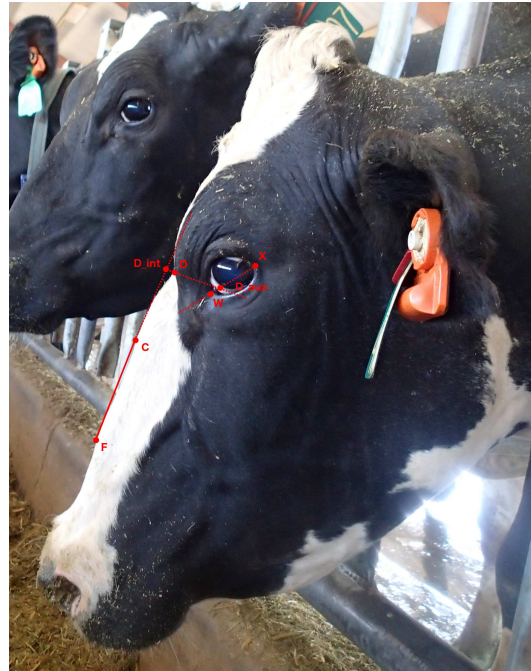
$$CF = \frac{\|D_{int}, D_{aux}\| - \|D, D_{aux}\|}{\|D_{int}, D_{aux}\| - \|D, D_{aux}\|}$$

$$V1 = \{C_{extrap}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V2 = \{C_{eye}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V3 = \{C_{extrap}, F_{eye}\} \rightarrow \{C, F\}$$

$$V4 = \{C_{eye}, F_{eye}\} \rightarrow \{C, F\}$$



Sinus-Topline Length Proportion

$$STLP = \frac{\|C_{len}, D_{len}\|}{\|EL_{len}, D_{len}\|}$$

$$V1 = \{C_{extrap}, F_{extrap}\} \rightarrow \{C, F\}$$

$$V2 = \{C_{eye}, F_{eye}\} \rightarrow \{C, F\}$$

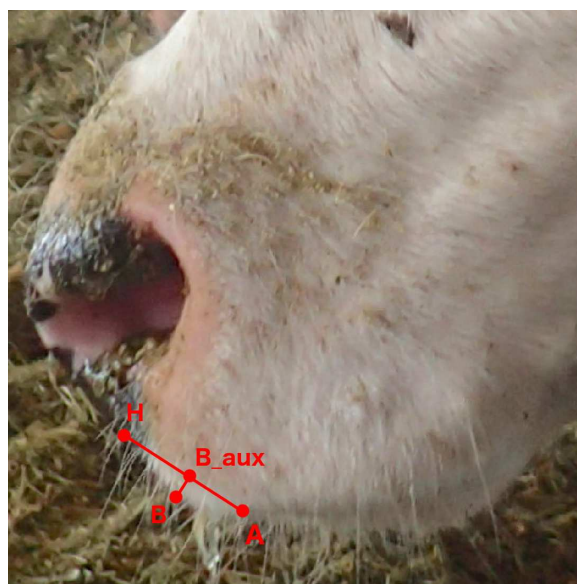


Upper Lip Roundness Point Proportion

$$ULRPP = \frac{\|A, B_{aux}\|}{\|H, A\|}$$

$$V1 = \{A_{extrap}\} \rightarrow \{A\}$$

$$V2 = \{A_{eye}\} \rightarrow \{A\}$$

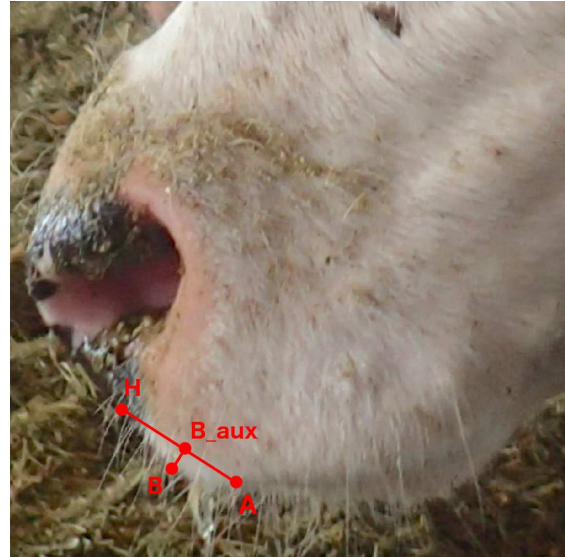


Upper Lip Roundness Proportion

$$ULRP = \frac{\|B, B_{aux}\|}{\|H, A\|}$$

$$V1 = \{A_{extrap}\} \rightarrow \{A\}$$

$$V2 = \{A_{eye}\} \rightarrow \{A\}$$



Upper-Lower Topline Length Proportion

$$ULTLP = \frac{\|EL_{len}, F_{len}\|}{\|F_{len}, D_{len}\|}$$

$$V1 = \{F_{extrap}\} \rightarrow \{F\}$$

$$V2 = \{F_{eye}\} \rightarrow \{F\}$$



Appendix B: Representative codes for measurement system analyses performed in the R programming environment for Chapter 1

```
##### Read in Data #####
##Load Geometric Biometric Data

biom <- read.csv('CSVs/Topline_Results_Master.csv', header=TRUE) biom <- biom[,1:86] # get
rid of nonsense columns
# review of biometric distributions revealed two corrupted files to delete biom[1104, start:done]
<- NA
biom[427, start:done] <- NA

biom_names <- names(biom) start <- 30
done <- ncol(biom)

biom_R1 <- biom[1:(nrow(biom)/2),]
biom_R2 <- biom[((nrow(biom)/2)+1):nrow(biom),]

biom_avg <- cbind(biom[1:(nrow(biom)/2),1:(start-1)],
(biom_R1[,start:done]+biom_R2[,start:done])/2) # average biometric value across the two reps

## load normalized distance measures

normdist <- read.csv('CSVs/ToplineRawDist_All.csv', header=TRUE) normdist [1104,
6:ncol(normdist)] <- NA # negate corrupted files normdist [427, 6:ncol(normdist)] <- NA

normdist_R1 <- normdist[1:(nrow(normdist)/2),]
normdist_R2 <- normdist[((nrow(normdist)/2)+1):nrow(normdist),]

normdist_avg <- cbind(normdist[1:(nrow(normdist)/2), 1:5],
(normdist_R1[,6:ncol(normdist)]+normdist_R2[,6:ncol(normdist)])/2) # average biometric value
across the two reps

#####
# ##### Visualization - Histogram of Geometric Biometrics #####

labels <- read.csv('CSVs/labels_topline.csv', header=F)

r=1
for (i in start:done){ # all metrics together

a <- labels[r,1]
r <- r + 1
b <- names(biom)[i] jpeg(paste('Visualizations/ToplineResults/BiometricDist_', b, '.jpg' , sep =
```



```

")) hist(as.numeric(biom[,i]), main=a , xlab =")
dev.off()

}
##### Visualization - Histogram of Normalized Distances ##### for (i in 6:ncol(normdist)){ # all
metrics together

b <- names(normdist)[i]
a <- paste('Distance',b) jpeg(paste('Visualizations/NormDistDistribution_', b, '.jpg' , sep = ""))
hist(as.numeric(normdist[,i]), main=a , xlab =")
dev.off()

}
##### Checking Outliers ##### which(-10 > biom$SMRP_V1) #[1] 427
biom$Cow.ID[427]
#[1] 31904
which(0.2 < biom$MDP_V1) #[1] 427
biom[427,1:5]
#Cow.ID Day Side Missing Rep #427 31904 D1 R 0 1
# definitely a corrupted file

which(1 < biom$MTLP_V2) #[1] 1104
which(1 < biom$NaDP_V2) #[1] 1104

which(1.5 < biom$NNLP_V1 ) # [1] 1104
which(0.6 < biom$STLP_V2 ) # [1] 1104

biom[1104,1:5]
# Cow.ID Day Side Missing Rep #1104 33513 D3 L 0 2
# also definitely a corrupted file

##### Visualization - Comparing Skew and Kurtosis #####

library('moments') library('ggplot2')

skew.norm <- c()
for(i in 6:ncol(normdist)){

skew.norm <- c(skew.norm, skewness(normdist[,i], na.rm = T))

}
qplot(skew.norm, geom="histogram", xlab = 'Skewness', ylab = "", bins = 20, main = 'Skewness
of Normalized Length Measures - Topline', col=I("black"), fill=I("red"), alpha=I(.5))
mean(skew.norm, na.rm=T)
# [1] 0.03559772

```

```

skew.biom <- c() for (i in start:done){

skew.biom <- c(skew.biom, skewness(biom[,i], na.rm = T)) }

qplot(skew.biom, geom="histogram", xlab = 'Skewness', ylab = "", bins = 20, main = 'Skewness
of Geometric Biometric Measures - Topline', col=I("black"), fill=I("blue"), alpha=I(.5))
mean(skew.biom, na.rm=T)
# [1] 0.1882498

kurt.norm <- c()
for(i in 6:ncol(normdist)){

kurt.norm <- c(kurt.norm, kurtosis(normdist[,i], na.rm = T)) }

qplot(kurt.norm, geom="histogram", xlab = 'Kurtosis', ylab = "", main = 'Kurtosis of Normalized
Length Measures - Topline', col=I("black"), fill=I("red"), alpha=I(.5))
mean(kurt.norm, na.rm=T)
#[1] 3.351645

kurt.biom <- c()
for (i in start:done){

kurt.biom <- c(kurt.biom, kurtosis(biom[,i], na.rm = T)) }

qplot(kurt.biom, geom="histogram", xlab = 'Kurtosis', ylab = "", main = 'Kurtosis of Geometric
Biometric Measures - Topline', col=I("black"), fill=I("blue"), alpha=I(.5))
mean(kurt.biom, na.rm=T)
#[1] 4.191718

#####
# ##### Properly Nested Repeatability Model - Single Click #####

dat<- biom

var_cow<- c() var_side <- c() var_day <- c() var_rep <- c()

err_day <- c() err_day_low <- c() err_day_high <- c()

err_rep <- c() err_rep_low <- c() err_rep_high <- c()

repeat_os <- c() repeat_os_low <- c()

repeat_os_high <- c()

for (i in start:done){
rm(temp)

```

```

rm(temp2)
temp <- data.frame(as.factor(dat$Cow.ID)) names(temp) <- 'Cow.ID'

temp$Day <- dat$Day temp$Side <- dat$Side temp$biometric<-dat[,i]

mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
nested within cow sigma <- as.data.frame(VarCorr(mix_out))

var_cow[i] <- sigma$vcov[3] var_side[i]<- sigma$vcov[2] var_day[i]<- sigma$vcov[1]
var_rep[i]<- sigma$vcov[4]

errDay <- function(.) { # percent of total error attributed to variance between days sigma <-
as.data.frame(VarCorr(.))
cow <- sigma$vcov[3]
side<- sigma$vcov[2]

day<- sigma$vcov[1]
rep<- sigma$vcov[4] return(day/sum(cow,side,day,rep))

}

bootday <- bootMer(mix_out,errDay, nsim = 1000, seed = 12, parallel = "multicore") err_day[i]
<- bootday$t0
temp2 <-boot::boot.ci(bootday, index=1, type= "basic")
err_day_low[i] <- temp2$basic[4]

err_day_high[i] <- temp2$basic[5]

errRep <- function(.) { # percent of total error attributed to rep sigma <-
as.data.frame(VarCorr(.))
cow <- sigma$vcov[3]
side<- sigma$vcov[2]

day<- sigma$vcov[1]
rep<- sigma$vcov[4] return(rep/sum(cow,side,day,rep))

}

bootrep <- bootMer(mix_out,errRep, nsim = 1000, seed = 12, parallel = "multicore") err_rep[i]
<- bootrep$t0
temp2 <-boot::boot.ci(bootrep, index=1, type="basic")
err_rep_low[i] <- temp2$basic[4]

err_rep_high[i] <- temp2$basic[5]

```

```

RepeatOneShot <- function(.) { # repeatability of one-shot biometric sigma <-
as.data.frame(VarCorr(.))
cow <- sigma$vcov[3]
side<- sigma$vcov[2]

day<- sigma$vcov[1] rep<- sigma$vcov[4]

return(sum(cow,side)/sum(cow,side,day,rep)) }

bootros <- bootMer(mix_out,RepeatOneShot, nsim = 1000, seed = 12, parallel = "multicore")
repeat_os[i] <- bootros$t0
temp2 <-boot::boot.ci(bootros, index=1, type="basic")
repeat_os_low[i] <- temp2$basic[4]

repeat_os_high[i] <- temp2$basic[5] }

ReapData_OneShot <- data.frame(names(biom)[start:done],var_cow[start:done],
var_side[start:done], var_day[start:done], var_rep[start:done], err_day_low[start:done],
err_day[start:done],err_day_high[start:done], err_rep_low[start:done],
err_rep[start:done],err_rep_high[start:done],
repeat_os_low[start:done],repeat_os[start:done],repeat_os_high[start:done])

names(ReapData_OneShot) <- c('Biometric', 'Var_Cow', 'Var_Side', 'Var_Day', 'Var_Rep',
'ErrorDay_low', 'ErrorDay','ErrorDay_high','ErrorRep_low',
'ErrorRep','ErrorRep_high','RepeatabilityOS_low','RepeatabilityOS','RepeatabilityOS_high')
write.csv(ReapData_OneShot, file = 'Results_Temp.csv')

##### Properly Nested Repeatability Model - Click Average #####

dat<- biom_avg

var_cow<- c() var_side <- c() var_day <- c() var_rep <- c()

repeat_avg <- c() repeat_avg_low <- c() repeat_avg_high <- c()

for (i in start:done){
rm(temp)
rm(temp2)
temp <- data.frame(as.factor(dat$Cow.ID)) names(temp) <- 'Cow.ID'

temp$Day <- dat$Day temp$Side <- dat$Side temp$biometric<-dat[,i]

mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side), data=temp) # day nested within side nested
within cow sigma <- as.data.frame(VarCorr(mix_out))

var_cow[i] <- sigma$vcov[2] var_side[i]<- sigma$vcov[1] var_day[i]<- sigma$vcov[3]

```



```

RepeatAvg <- function(.) { # repeatability of one-shot biometric sigma <-
as.data.frame(VarCorr(.))
cow <- sigma$vcov[2]
side<- sigma$vcov[1]

day<- sigma$vcov[3]

return(sum(cow,side)/sum(cow,side,day)) }

bootrav <- bootMer(mix_out,RepeatAvg , nsim = 1000, seed = 12, parallel = "multicore")
repeat_avg[i] <- bootrav$t0
temp2 <-boot::boot.ci(bootrav, index=1, type="basic")
repeat_avg_low[i] <- temp2$basic[4]

repeat_avg_high[i] <- temp2$basic[5] }

ReapData_Avg <- data.frame(names(biom)[start:done],var_cow[start:done],
var_side[start:done], var_day[start:done],
repeat_avg_low[start:done],repeat_avg[start:done],repeat_avg_high[start:done])
names(ReapData_Avg) <- c('Biometric', 'Var_Cow', 'Var_Side',
'Var_Day','RepeatabilityAvg_low','RepeatabilityAvg','RepeatabilityAvg_high')

write.csv(ReapData_Avg, file = 'Results_Temp.csv')

##### Comparing Geometric and Normalized Length Repeatability - Between Days #####

dat <- normdist_avg # for repeatability across days, all reps repeat_dist_day <- c()

for (i in 6:ncol(dat)){

rm(temp)
temp <- data.frame(as.factor(dat$Cow.ID)) names(temp) <- 'Cow.ID'
temp$Day <- dat$Day
temp$Side <- dat$Side temp$biometric<-dat[,i]

mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side), data=temp) # day nested within side nested
within cow sigma <- as.data.frame(VarCorr(mix_out))

RepeatAvg <- function(.) { # repeatability of one-shot biometric sigma <-
as.data.frame(VarCorr(.))
cow <- sigma$vcov[2]
side<- sigma$vcov[1]

day<- sigma$vcov[3]

return(sum(cow,side)/sum(cow,side,day)) }

```

```

repeat_dist_day[i] <- RepeatAvg(mix_out) }

repeat_dist_day <- repeat_dist_day[6:ncol(dat)] # remove spare columns names(repeat_dist_day)
<- names(normdist_avg[,6:ncol(dat)])

mean(repeat_dist_day) # overall average repeatability
# [1] 0.3952134
hist(repeat_dist_day, xlim = c(0,1), breaks = 15, main = 'Repeatability of Normalized Length
Between Days', xlab = 'Repeatability')

qplot(repeat_dist_day, geom="histogram", bins = 20, xlab = 'Repeatability of Biometrics
Averaged Over Coordinate Extraction Reps', ylab = "", main = 'Repeatability of Normalized
Length Between Days - Topline', col=I("black"), fill=I("red"), alpha=I(.5))

# Results for Geometric Biometrics

mean(ReapData_Avg$RepeatabilityAvg) # overall average repeatability
# [1] 0.5234241
hist(ReapData_Avg$RepeatabilityAvg, xlim = c(0,1), breaks = 15, main = 'Repeatability of
Geometric Biometrics Between Days', xlab = 'Repeatability')
qplot(ReapData_Avg$RepeatabilityAvg, bins = 20, geom="histogram", xlab = 'Repeatability of
Biometrics Averaged Over Coordinate Extraction Reps', ylab = "", main = 'Repeatability of
Geometric Biometrics Between Days - Topline', col=I("black"), fill=I("blue"), alpha=I(.5))

#### Comparing Geometric and Normalized Length Repeatability - Between Reps ####

dat <- normdist # for repeatability across days, all reps repeat_dist_rep <- c()

for (i in 6:ncol(dat)){

  rm(temp)
  temp <- data.frame(as.factor(dat$Cow.ID)) names(temp) <- 'Cow.ID'
  temp$Day <- dat$Day
  temp$Side <- dat$Side temp$biometric<-dat[,i]

  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
  nested within cow sigma <- as.data.frame(VarCorr(mix_out))

  RepeatOneShot <- function(.) { # repeatability of one-shot biometric sigma <-
as.data.frame(VarCorr(.))
  cow <- sigma$vcov[3]
  side<- sigma$vcov[2]

  day<- sigma$vcov[1]
  rep<- sigma$vcov[4] return(sum(cow,side)/sum(cow,side,day,rep))

```

```

}

repeat_dist_rep[i] <- RepeatOneShot(mix_out) }

repeat_dist_rep <- repeat_dist_rep[6:ncol(dat)] # remove spare columns names(repeat_dist_rep)
<- names(normdist[,6:ncol(dat)])

mean(repeat_dist_rep) # overall average repeatability
# [1] 0.3009558
hist(repeat_dist_rep, xlim = c(0,1), breaks = 15, main = 'Repeatability of Normalized Length
Between Reps', xlab = 'Repeatability')

qplot(repeat_dist_rep, geom="histogram", bins = 20, xlab = 'Repeatability of Biometrics
Between Coordinate Extraction Reps', ylab = "", main = 'Repeatability of Normalized Length
Within Photo - Topline', col=I("black"), fill=I("red"), alpha=I(.5))

# Results for Geometric Biometrics

mean(ReapData_OneShot$RepeatabilityOS) # overall average repeatability
#[1] 0.4345016
hist(ReapData_OneShot$RepeatabilityOS, xlim = c(0,1), breaks = 15, main = 'Repeatability of
Geometric Biometrics Between Reps', xlab = 'Repeatability')

qplot(ReapData_OneShot$RepeatabilityOS, geom="histogram", bins = 20, xlab = 'Repeatability
of Biometrics Between Coordinate Extraction Reps', ylab = "", main = 'Repeatability of
Geometric Biometric Within Photo - Topline', col=I("black"), fill=I("blue"), alpha=I(.5))

##### Comparing Correlation in Errors #####

## biometrics

dat<- biom_avg[complete.cases(biom_avg),] resid_biom <- data.frame(dat[,1:(start-1)])

for (i in start:done){
  rm(temp)
  rm(temp2)
  temp <- data.frame(as.factor(dat$Cow.ID)) names(temp) <- 'Cow.ID'

  temp$Day <- dat$Day temp$Side <- dat$Side temp$biometric<-dat[,i]

  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side), data=temp) # day nested within side nested
  within cow temp2 <- summary(mix_out)
  resid_biom <- cbind(resid_biom , temp2$residuals)

}
names(resid_biom)[(start):ncol(resid_biom)] <- names(biom_avg[,start:done])

```

```

dat <- resid_biom[,start:done]
dat <- dat[,biom_keepers]
dat$Cow.ID <- as.factor(resid_biom$Cow.ID) dat$Side <- as.factor(resid_biom$Side) temp<-
statsBy(dat, group = c('Cow.ID','Side'), cor = T) errcor_biom <- as.matrix(temp$rbg)
temp <- names( resid_biom[,start:done])
temp <- temp[biom_keepers]
colnames(errcor_biom) <- temp rownames(errcor_biom) <- temp

temp1 <- as.vector(errcor_biom[lower.tri(errcor_biom, diag = FALSE)]) cor_resid_biom <-
temp1^2

errcor_biom_TF <- errcor_biom^2 > 0.7
## norm dist dat<-normdist_avg[complete.cases(normdist_avg),] resid_normdist <-
data.frame(dat[1:5])

for (i in 6:ncol(dat)){
  rm(temp)
  rm(temp2)
  temp <- data.frame(as.factor(dat$Cow.ID)) names(temp) <- 'Cow.ID'

  temp$Day <- dat$Day temp$Side <- dat$Side temp$biometric<-dat[,i]

  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side), data=temp) # day nested within side nested
  within cow temp2 <- summary(mix_out)
  resid_normdist <- cbind(resid_normdist , temp2$residuals)

}
names(resid_normdist)[6:ncol(resid_normdist)] <- names(dat[,6:ncol(dat)])

dat <- resid_normdist[,6:ncol(resid_normdist)]
dat$Cow.ID <- as.factor(resid_normdist$Cow.ID)
dat$Side <- as.factor(resid_normdist$Side)
temp<-statsBy(dat, group = c('Cow.ID','Side'), cor = T)
errcor_normdist <- as.matrix(temp$rbg)
colnames(errcor_normdist) <- names( normdist_avg[,6:ncol(normdist)] )
rownames(errcor_normdist) <- names( normdist_avg[,6:ncol(normdist)] )

temp1 <- as.vector(errcor_normdist[lower.tri(errcor_normdist, diag = FALSE)])
cor_resid_normdist <- temp1^2

## Compare residual dist

hist(cor_resid_biom, main = 'Residual Correlations Between Geometric Biometrics', xlab =
'R^2', xlim = c(0,1), breaks = 15)
qplot(cor_resid_biom, geom="histogram", xlab = 'R^2 Error', ylab = "", main = 'Correlations

```



```
Between Between-Day Error for Geometric Biometrics - Topline', col=I("black"), fill=I("blue"),
alpha=I(.5))
```

```
hist(cor_resid_normdist, main = 'Residual Correlations Between Normalized Lengths', xlab =
'R^2', xlim = c(0,1), breaks = 15)
qplot(cor_resid_normdist, geom="histogram", xlab = 'R^2 Error', ylab = "", main = 'Correlations
Between Between- Day Error for Geometric Biometrics - Topline', col=I("black"), fill=I("red"),
alpha=I(.5))
```

```
mean(cor_resid_biom) #[1] 0.08252397 mean(cor_resid_normdist) #[1] 0.04435983
```

```
#####
# ##### Correlations Between Biometrics #####
```

```
dat <- biom_avg[,start:done] names(dat)[biom_keepers]
dat <- dat[,biom_keepers]
dat$Cow.ID <- as.factor(biom_avg$Cow.ID)
dat$Side <- as.factor(biom_avg$Side) temp<-statsBy(dat, group = c('Cow.ID','Side'), cor = T)
cor_biom <- as.matrix(temp$rbg)
```

```
temp <- names( biom_avg[,start:done])
temp <- temp[biom_keepers]
colnames(cor_biom) <- temp
rownames(cor_biom) <- temp
write.csv(cor_biom, 'Visualizations/correlations_biometrics.csv')
```

```
#### Correlations Between Normalized Lengths ####
```

```
dat <- normdist_avg[,6:ncol(normdist)]
dat$Cow.ID <- as.factor(normdist_avg$Cow.ID)
dat$Side <- as.factor(normdist_avg$Side)
temp<-statsBy(dat, group = c('Cow.ID','Side'), cor = T)
cor_normdist <- as.matrix(temp$rbg)
colnames(cor_normdist) <- names( normdist_avg[,6:ncol(normdist)] ) rownames(cor_normdist)
<- names( normdist_avg[,6:ncol(normdist)] )
```

```
#### Compare Correlation Structures between geometric and norm length ####
```

```
temp1 <- as.vector(cor_biom[lower.tri(cor_biom, diag = FALSE)]) temp1 <- temp1 ^2 # convert
to r^2
mean(temp1) # average correlation of biometrics
#[1] 0.08010048
```

```
hist(cor_biom, main = 'Correlations Between Geometric Biometrics', xlab = 'r^2', breaks = 30,
xlim = c(0,1)) qplot(temp1, geom="histogram", bins = 20, xlab = 'R^2', ylab = "", main =
```

```

'Correlations Between Geometric Biometrics - Topline', col=I("black"), fill=I("blue"),
alpha=I(.5))

temp2 <- as.vector(cor_normdist[lower.tri(cor_normdist, diag = FALSE)]) temp2 <- temp2 ^2 #
convert to r^2
mean(temp2) # average correlation of normalized distances
#[1] 0.06512792

hist(cor_normdist, main = 'Correlations Between Normalized Distances', xlab = 'r^2', breaks =
30, xlim = c(0,1)) qplot(temp2, geom="histogram", bins = 20, xlab = 'R^2', ylab = "", main =
'Correlations Between Normalized Distances - Topline', col=I("black"), fill=I("red"),
alpha=I(.5))

#####
##### Geometric Corrections - Image Attribute Metrics #####

## Creating Dataset of Chage in Biometrics between all pairwise sets of days for each cow x side

cows <-unique(biom$Cow.ID)
cordat <- data.frame(rep(cows, each=6)) names(cordat)<-'CowID'
cordat$Side <- rep(c('R','R','R','L','L','L'), times=length(cows)) cordat$Day_ref <- rep( c(1,1,2) ,
times=(2*length(cows))) cordat$Day_test <- rep( c(2,3,3) , times=(2*length(cows)))

refcols <-start:done # the column numbers we'll be pulling computed biometrics out of biom data
set dumpcols <-seq(from=ncol(cordat)+1, by=2,length.out = length(refcols)) # the first column
that each metric corrdinate pair will be placed into

dat <- biom_avg

for (i in 1:length(refcols)){ # parsing through columns of biometric results incol = refcols[i] #
where the input data is coming from biom
outcol1 = dumpcols[i] # where the input data is being inserted into cordat outcol2 =
dumpcols[i]+1

for (j in seq(from=1, to=nrow(cordat), by=6)) { # right side measures
cordat[j,outcol1]=dat[j,incol] cordat[j,outcol2]=dat[j+2,incol] cordat[j+1,outcol1]=dat[j,incol]
cordat[j+1,outcol2]=dat[j+4,incol] cordat[j+2,outcol1]=dat[j+2,incol]
cordat[j+2,outcol2]=dat[j+4,incol]

#left side measures cordat[j+3,outcol1]=dat[j+1,incol] cordat[j+3,outcol2]=dat[j+3,incol]
cordat[j+4,outcol1]=dat[j+1,incol] cordat[j+4,outcol2]=dat[j+5,incol]
cordat[j+5,outcol1]=dat[j+3,incol] cordat[j+5,outcol2]=dat[j+5,incol]

} }

names(cordat)[5:ncol(cordat)] <- rep(names(biom)[start:done],each=2) cordat_biom <- cordat

```

```

errordat_biom <- cordat[,1:4]
for (i in seq(5,ncol(cordat),by=2)){

temp <- cordat[,i] - cordat[,i+1] errordat_biom <- cbind(errordat_biom, temp)

}
names(errordat_biom)[5:ncol(errordat_biom)] <- names(biom)[start:done]

## Creating Dataset of Chage in Biometrics between all pairwise sets of days for each cow x side

cows <-unique(biom$Cow.ID)
cordat <- data.frame(rep(cows, each=6)) names(cordat)<-'CowID'
cordat$Side <- rep(c('R','R','R','L','L','L'), times=length(cows)) cordat$Day_ref <- rep( c(1,1,2) ,
times=(2*length(cows))) cordat$Day_test <- rep( c(2,3,3) , times=(2*length(cows)))

refcols <- 6 : ncol(normdist) # the column numbers we'll be pulling computed biometrics out of
biom data set dumpcols <-seq(from=ncol(cordat)+1, by=2,length.out = length(refcols)) # the first
column that each metric corrdinate pair will be placed into

dat <- normdist_avg

for (i in 1:length(refcols)){ # parsing through columns of biometric results incol = refcols[i] #
where the input data is coming from biom
outcol1 = dumpcols[i] # where the input data is being inserted into cordat outcol2 =
dumpcols[i]+1

for (j in seq(from=1, to=nrow(cordat), by=6)) { # right side measures
cordat[j,outcol1]=dat[j,incol] cordat[j,outcol2]=dat[j+2,incol] cordat[j+1,outcol1]=dat[j,incol]
cordat[j+1,outcol2]=dat[j+4,incol] cordat[j+2,outcol1]=dat[j+2,incol]
cordat[j+2,outcol2]=dat[j+4,incol]

#left side measures cordat[j+3,outcol1]=dat[j+1,incol] cordat[j+3,outcol2]=dat[j+3,incol]
cordat[j+4,outcol1]=dat[j+1,incol] cordat[j+4,outcol2]=dat[j+5,incol]
cordat[j+5,outcol1]=dat[j+3,incol] cordat[j+5,outcol2]=dat[j+5,incol]

} }

names(cordat)[5:ncol(cordat)] <- rep(names(normdist)[6 : ncol(normdist)],each=2)

cordat_normist <- cordat errordat_normdist <- cordat[,1:4]

for (i in seq(5,ncol(cordat),by=2)){

temp <- cordat[,i] - cordat[,i+1]
errordat_normdist <- cbind(errordat_normdist, temp)

```

```

}
names(errordat_normdist)[5:ncol(errordat_normdist)] <- names(normdist)[6 : ncol(normdist)]

## Load in the image attributes data set
imatrib <- read.csv('CSVs/ImageAttributes_R2.csv', header=TRUE)

imatrib <- imatrib[, c(1,2,3,4,26,27,28,29,30,31,32,33,34,35,36,37,38)] # just keep the indexing
info and image attribute metrics

## Creating Dataset of Chage in Biometrics between all pairwise sets of days for each cow x side

cows <-unique(biom$Cow.ID)
cordat <- data.frame(rep(cows, each=6))
names(cordat) <-'CowID'
cordat$Side <- rep(c('R','R','R','L','L','L'), times=length(cows)) cordat$Day_ref <- rep( c(1,1,2) ,
times=(2*length(cows))) cordat$Day_test <- rep( c(2,3,3) , times=(2*length(cows)))

refcols <- 5 : ncol(imatrib) # the column numbers we'll be pulling computed biometrics out of
biom data set dumpcols <-seq(from=ncol(cordat)+1, by=2,length.out = length(refcols)) # the first
column that each metric corrdinate pair will be placed into

dat <- imatrib

for (i in 1:length(refcols)){ # parsing through columns of biometric results incol = refcols[i] #
where the input data is coming from biom
outcol1 = dumpcols[i] # where the input data is being inserted into cordat outcol2 =
dumpcols[i]+1

for (j in seq(from=1, to=nrow(cordat), by=6)) { # right side measures
cordat[j,outcol1]=dat[j,incol] cordat[j,outcol2]=dat[j+2,incol] cordat[j+1,outcol1]=dat[j,incol]
cordat[j+1,outcol2]=dat[j+4,incol] cordat[j+2,outcol1]=dat[j+2,incol]
cordat[j+2,outcol2]=dat[j+4,incol]

#left side measures cordat[j+3,outcol1]=dat[j+1,incol] cordat[j+3,outcol2]=dat[j+3,incol]
cordat[j+4,outcol1]=dat[j+1,incol] cordat[j+4,outcol2]=dat[j+5,incol]
cordat[j+5,outcol1]=dat[j+3,incol] cordat[j+5,outcol2]=dat[j+5,incol]

} }

names(cordat)[5:ncol(cordat)] <- rep(names(imatrib)[5 : ncol(imatrib )],each=2)

cordat_imatrib <- cordat errordat_imatrib <- cordat[,1:4]

for (i in seq(5,ncol(cordat),by=2)){

```



```

temp <- cordat[,i] - cordat[,i+1]
errordat_imatrib <- cbind(errordat_imatrib, temp)

}
names(errordat_imatrib)[5:ncol(errordat_imatrib)] <- names(imatrib)[5 : ncol(imatrib)]

x1 <- errordat_imatrib$raw_head2frameratio # frame2face ratio x2 <- x1^2
x3 <- errordat_imatrib$OverallFaceAngle # overall face angle

x4 <- errordat_imatrib$eyecenteroffset_vertical # eye-center offset - vertical
x5 <- errordat_imatrib$eyecenteroffset_horizontal # eye-center offset - horizontal

## Correlations image quality metrics library('MASS')

# correlation to geometric biometric
rsq_biomerror <- c()
for (i in 5:ncol(errordat_biom)){ # fit regression to all biometric errors

fit <- lm(errordat_biom[,i] ~ x1 + x2 + x3 + x1*x3 + x2*x3)
step <- stepAIC(fit, direction="backward") # stepwise backward regression, save results to list
object temp <- summary(step)
rsq_biomerror[i] <- temp$r.squared

}

mean(rsq_biomerror, na.rm = T)
#[1] 0.01195944
hist(rsq_biomerror , main = 'Proportion of Between-Day Error in Geometric Biometrics
Attributed to Image Quality', xlab = 'R^2' , xlim = c(0,1))
qplot(rsq_biomerror , geom="histogram", bins = 20, xlab = 'R^2', ylab = "", main = 'Proportion of
Between-Day Error in Geometric Biometrics Attributed to Image Quality - Topline',
col=I("black"), fill=I("blue"), alpha=I(.5))

# correlations to geometric biometrics
rsq_rawdistorerror <- c()
for (i in 5:ncol(errordat_normdist)){ # fit regression to all biometric errors

fit <- lm(errordat_normdist[,i] ~ x1 + x2 + x3 + x1*x3 + x2*x3)
step <- stepAIC(fit, direction="backward") # stepwise backward regression, save results to list
object temp <- summary(step)
rsq_rawdistorerror[i] <- temp$r.squared

}

mean(rsq_rawdistorerror, na.rm = T)
# [1]0.009693945

```

```

hist(rsq_rawdisterror, main = 'Proportion of Between-Day Error in Normalize Lengths Attributed
to Image Quality', xlab = 'R^2' , xlim = c(0,1))
qplot(rsq_rawdisterror , geom="histogram", bins = 20, xlab = 'R^2', ylab = "", main = 'Proportion
of Between-Day Error in Normalize Lengths Attributed to Image Quality - Topline',
col=I("black"), fill=I("red"), alpha=I(.5))

##### Geometric Corrections - Camera Position Correction #####

# checking for correlations between camera position and geometric biometric error

rsq_biomerror <- c()
for (i in 5:ncol(errordat_biom)){ # fit regression to all biometric errors

fit <- lm(errordat_biom[,i] ~ x4 + x5 + I(x4^2) + I(x5^2) + x1*x4 + x1*x5 + x2*x4 + x2*x5)
step <- stepAIC(fit, direction="backward") # stepwise backward regression, save results to list
object temp <- summary(step)
rsq_biomerror[i] <- temp$r.squared

}

mean(rsq_biomerror, na.rm = T) #[1] 0.01672386

hist(rsq_biomerror , main = 'Proportion of Between-Day Error in Geometric Biometrics
Attributed to Camera Position', xlab = 'R^2' , xlim = c(0,1))
names(errordat_biom)[which(rsq_biomerror>0.1)]
qplot(rsq_biomerror, geom="histogram", bins = 20, xlab = 'R^2', ylab = "", main = 'Proportion of
Between-Day Error in Geometric Biometrics Attributed to Camera Position - Topline',
col=I("black"), fill=I("blue"), alpha=I(.5))

# checking for correlations between raw eye dist and image attributes

rsq_rawdisterror <- c()
for (i in 5:ncol(errordat_normdist)){ # fit regression to all biometric errors

fit <- lm(errordat_normdist[,i] ~ x4 + x5 + I(x4^2) + I(x5^2) + x1*x4 + x1*x5 + x2*x4 + x2*x5)
step <- stepAIC(fit, direction="backward") # stepwise backward regression, save results to list
object temp <- summary(step)
rsq_rawdisterror[i] <- temp$r.squared

}

mean(rsq_rawdisterror, na.rm = T)
#[1] [1] 0.02119845
hist(rsq_rawdisterror, main = 'Proportion of Between-Day Error in Normalized Lengths
Attributed to Camera Position', xlab = 'R^2' , xlim = c(0,1), breaks = 15)
qplot(rsq_rawdisterror, geom="histogram", bins = 20, xlab = 'R^2', ylab = "", main = 'Proportion

```

of Between-Day Error in Normalize Lengths Attributed to Camera Position - Topline',
col=I("black"), fill=I("red"), alpha=I(.5))

Appendix C: Representative codes for predictive performed in the R programming environment for Chapter 2, as well as supplemental tables

#Data Wrangling

Getting the biometric and genomic data sets read in and merged.

```
biometric_master <- read.csv("~/Documents/Research/Dairy_Behavior/Projects/NFH/Data/Biometric  
Results/Biometric_Master.csv", header=TRUE)  
biometric_master <- biometric_master[complete.cases(biometric_master),]
```

```
biometric_R1 <- subset(biometric_master, Rep == 1)  
biometric_R2 <- subset(biometric_master, Rep == 2)  
biometric_avg <- cbind(biometric_R1[,1], (biometric_R1[,3:ncol(biometric_R1)] +  
biometric_R2[,3:ncol(biometric_R2)])/2) # take average of two reps for each cow  
names(biometric_avg)[1] <- 'CowID'
```

```
genomic_master <- read.csv("~/Documents/Research/Dairy_Behavior/Projects/NFH/Data/Genotype  
Data.csv", header=TRUE)  
names(genomic_master)[1] <- 'CowID'
```

```
cowlist_master <- read.csv("~/Documents/Research/Dairy_Behavior/Projects/NFH/Data/RFID  
Numbers/Cow_Match_Results.csv", header=TRUE)  
cowlist_master <- cowlist_master[,2:3]  
names(cowlist_master) <- c('CowID', 'RFID')
```

```
health_master <- read.csv("~/Documents/Research/Dairy_Behavior/Projects/NFH/Data/Health Data.csv",  
header=TRUE)  
names(health_master)[1] <- 'CowID'  
health_master <- health_master[complete.cases(health_master), ]
```

```
cowdat <- merge(cowlist_master, genomic_master, by = 'CowID')  
cowdat <- merge(cowdat, biometric_avg, by = 'CowID', all.y = TRUE)  
paste('Missing Cows')  
unique(cowdat$CowID[is.na(cowdat$RFID)]) #cows with biometric data but no genomic data  
miscow <- data.frame('CowID' = unique(cowdat$CowID[is.na(cowdat$RFID)]))  
miscow <- merge(miscow, cowlist_master, all.x = TRUE)
```

```
[1] "Missing Cows"  
[1] 20180 21379 23157 23508 27189 27410 27603 28175 28431 28664 29034 29185 29336 33565  
[15] 36993 37056 37332 37447 37534 39703 40316
```

```
cowdat <- cowdat[!is.na(cowdat$RFID),]  
paste('Final Dataset Dimension')  
dim(cowdat)
```

```
[1] "Final Dataset Dimension"
```

```
[1] 573 146
```

```
cowdat.health <- merge(cowdat, health_master, by = 'CowID', all.y = T)
cowdat.health <- cowdat.health[!is.na(cowdat.health$RFID),]
```

```
#####
```

```
#Predictor Characterization
```

First is just to get a sense of the distributions to look for outliers. Any distributions with strong outliers I'll treat with a log transform.

```
cowdat_biom <- cowdat[, 87:ncol(cowdat)]
#names(cowdat_biom)
ncol(cowdat_biom)
par(mfrow=c(2, 2))
```

```
for (i in 1:ncol(cowdat_biom)){
```

```
  hist(cowdat_biom[,i], main = names(cowdat_biom)[i], xlab = 'Biometric Values')
```

```
}
```

```
loglist <- c(which(names(cowdat_biom) == 'Z3_front'), which(names(cowdat_biom) == 'Z9_poly'),
which(names(cowdat_biom) == 'Z10_poly'), which(names(cowdat_biom) == 'ESSRP_V1'),
which(names(cowdat_biom) == 'EOEHR_V1'))
```

```
for (i in 1:ncol(cowdat_biom)){
```

```
  if (i %in% loglist){
    cowdat_biom[,i] <- scale(log(cowdat_biom[,i]))
```

```
    #hist(cowdat_biom[,i])
```

```
  } else{
```

```
    cowdat_biom[,i] <- scale(cowdat_biom[,i])
```

```
  }
```

```
}
```

Checking that no cow has extreme leverage based on combination of biometric values.

```
X = as.matrix(cowdat_biom)
```

```
H = X %*% solve(t(X) %*% X) %*% t(X)
```

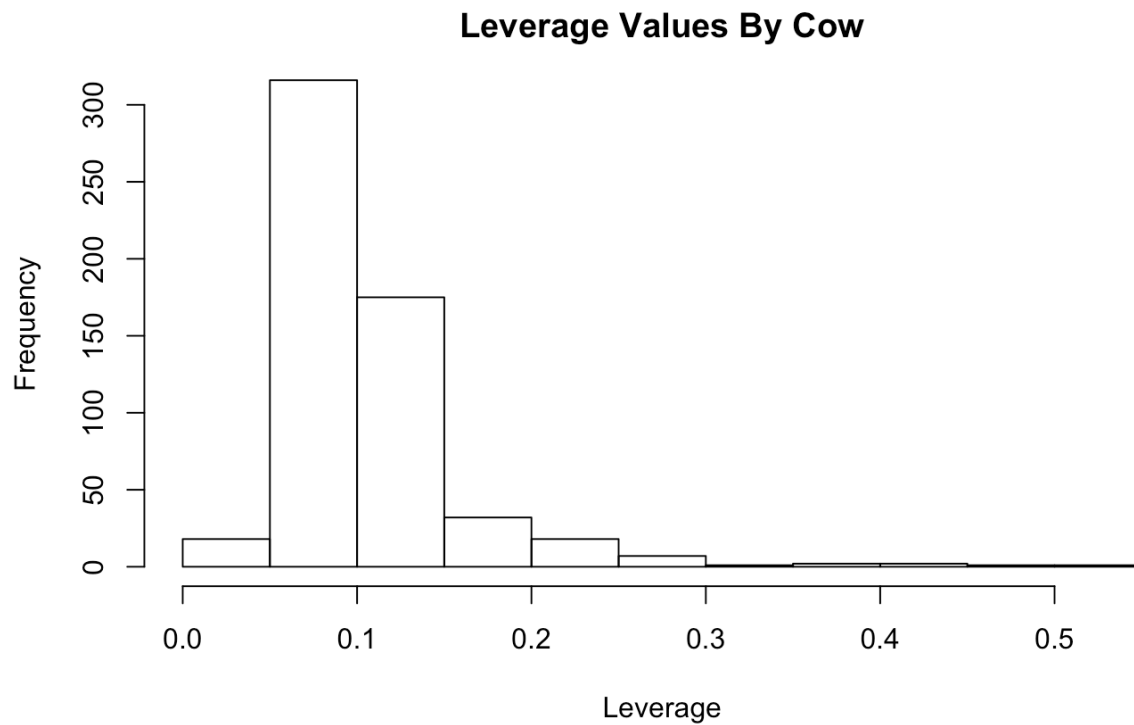
```
hist(diag(H), main = 'Leverage Values By Cow', xlab = 'Leverage')
```

```
sum(diag(H)>0.3)
```

```
7
```

```
sum(diag(H)>0.4)
```

```
4
```

None of these look excessive to me, so I don't suspect any erroneous values, so I'm not inclined to drop any cows.

```
#####
```

Checking the distributions for clear outliers.

```
par(mfrow=c(2, 2))
```

```
cowdat_type <- cbind(cowdat$PTA.Type, cowdat$UDC, cowdat$FLC, cowdat[,45:63])
names(cowdat_type)[1:3] <- c('PTA.Type', 'UDC', 'FLC')
ncol(cowdat_type)
```

```
for (i in 1:ncol(cowdat_type)){
```

```

hist(cowdat_type[,i], main = names(cowdat_type)[i], xlab = 'Conformation Values')
}

for (i in 1:ncol(cowdat_type)){

  cowdat_type[,i] <- scale(cowdat_type[,i])

}

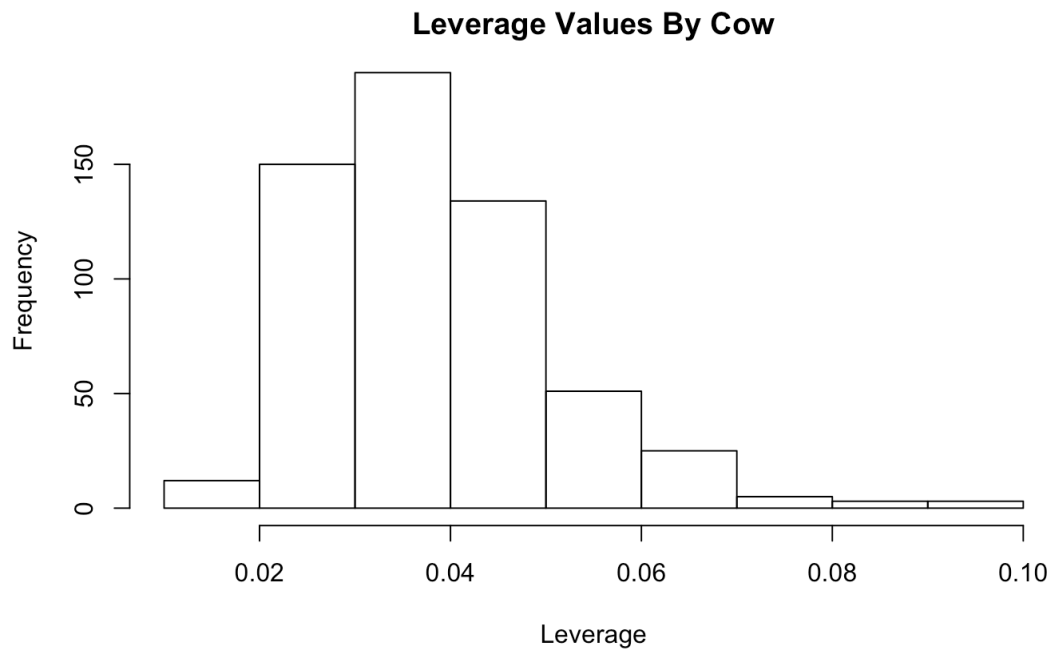
```

Checking that no cow has extreme leverage based on combination of biometric values.

```

X = as.matrix(cowdat_type)
H = X %*% solve(t(X) %*% X) %*% t(X)
hist(diag(H), main = 'Leverage Values By Cow', xlab = 'Leverage')

```



Definitely no leverage animals in terms of conformation.

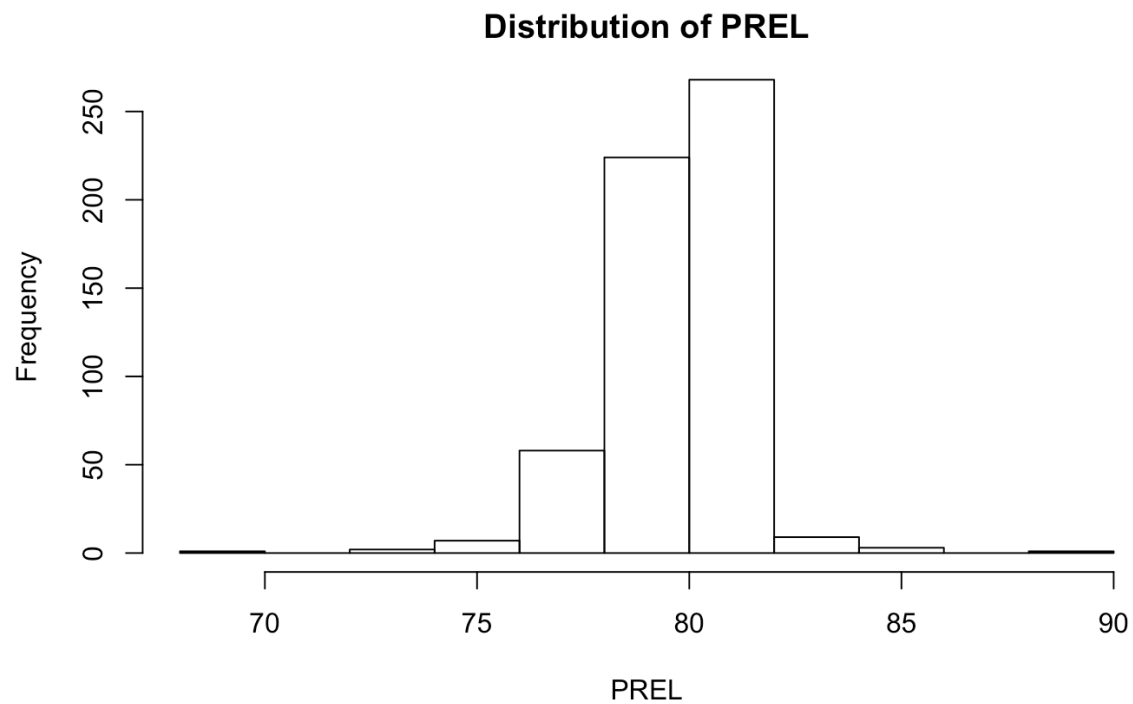
#####

##Reliability Values

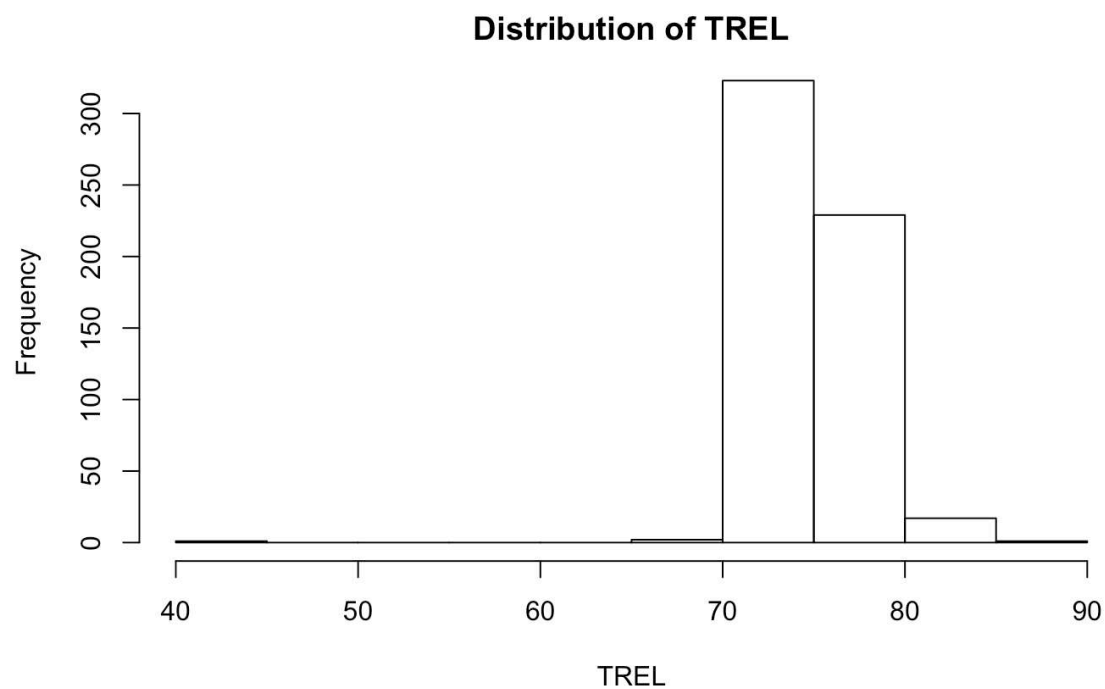
```

hist(cowdat$Prel, main = 'Distribution of PREL', xlab = 'PREL')

```



```
hist(cowdat$Trel, main = 'Distribution of TREL', xlab = 'TREL')
```



```
cowdat_rel <- data.frame(PREL = cowdat$PreI, TREL = cowdat$Trel)
cor(cowdat_rel)
      PREL    TREL
PREL 1.000000 0.5035476
TREL 0.5035476 1.0000000
```

```
#####
```

```
#LASSO Models - Linear
```

From the PTA dataset I'm going to set aside 103 animals for my testing set, and I'll leave the remaining 470 for model development

```
library('glmnet')  
library("plotmo")
```

```
set.seed(1212)
```

```
testid <- sample(nrow(cowdat), 103, replace = FALSE)
```

```
cowdat_train <- cowdat[-testid,]  
cowdat_test <- cowdat[testid,]
```

```
cowdat_biom_train <- cowdat_biom[-testid,]  
cowdat_biom_test <- cowdat_biom[testid,]
```

```
cowdat_type_train <- cowdat_type[-testid,]  
cowdat_type_test <- cowdat_type[testid,]
```

```
cowdat_rel_train <- cowdat_rel[-testid,]  
cowdat_rel_test <- cowdat_rel[testid,]
```

```
cowdat_x_train <- cbind(cowdat_type_train, cowdat_biom_train)  
cowdat_x_test <- cbind(cowdat_type_test, cowdat_biom_test)
```

From the health dataset I'll set aside 54 animals and leave the remaining 290 for training the models

```
set.seed(1212)  
testidH <- sample(nrow(cowdat.health), 54, replace = FALSE)
```

```
cowdat.health.biom <- cowdat.health[, 87:146]
```

```
cowdat.health.type <- cbind(cowdat.health$PTA.Type, cowdat.health$UDC, cowdat.health$FLC,  
cowdat.health[,45:63])  
names(cowdat.health.type)[1:3] <- c('PTA.Type', 'UDC', 'FLC')
```

```
cowdat.health.rel <- data.frame(PREL = cowdat.health$Prel, TREL = cowdat.health$Trel)
```

```
cowdat.health.train <- cowdat.health[-testidH,]  
cowdat.health.test <- cowdat.health[testidH,]
```

```
cowdat.health.biom.train <- cowdat.health.biom[-testidH,]  
cowdat.health.biom.test <- cowdat.health.biom[testidH,]
```



```

cowdat.health.type.train <- cowdat.health.type[-testidH,]
cowdat.health.type.test <- cowdat.health.type[testidH,]

cowdat.health.rel.train <- cowdat.health.rel[-testidH,]
cowdat.health.rel.test <- cowdat.health.rel[testidH,]

cowdat.health.x.train <- cbind(cowdat.health.type.train, cowdat.health.biom.train)
cowdat.health.x.test <- cbind(cowdat.health.type.test, cowdat.health.biom.test)

```

Initializing result vectors:

```

out.r.full <- c()
out.r.type <- c()
out.r.biom <- c()
out.r.test <- c()
out.r.biomgain.training <- c()
out.r.biomgain.test <- c()

out.coef <- c()

```

##TPI - Total Performance Index

```

set.seed(12)
index = 1

```

```

hist(cowdat$CTPI, main = 'CTPI', xlab = 'PTA Value')

```

```

x_train <- model.matrix( ~ .-1, cowdat_x_train)
x_test <- model.matrix( ~ .-1, cowdat_x_test)

```

```

x_type_train <- model.matrix( ~ .-1, cowdat_type_train)
x_type_test <- model.matrix( ~ .-1, cowdat_type_test)

```

```

x_biom_train <- model.matrix( ~ .-1, cowdat_biom_train)
x_biom_test <- model.matrix( ~ .-1, cowdat_biom_test)

```

```

cv.out <- cv.glmnet(x_train, cowdat_train$CTPI , alpha=1, nlambda = 10000, weights =
cowdat_rel_train$PREL, nfolds = 10)
bestlam <- cv.out$lambda.min

```

```

lasso.mod <- glmnet(x_train, cowdat_train$CTPI, alpha=1, lambda=bestlam, intercept = T, weights =
cowdat_rel_train$PREL)
coef(lasso.mod)
out.coef <- matrix(coef(lasso.mod))
colnames(out.coef) <- 'CTPI'

```

```

cv.out <- cv.glmnet(x_type_train, cowdat_train$CTPI , alpha=1, nlambda = 10000, weights =
cowdat_rel_train$PREL, nfolds = 10)

```

```

bestlam <- cv.out$lambda.min

lasso.mod.type <- glmnet(x_type_train, cowdat_train$CTPI, alpha=1, lambda=bestlam, intercept = T,
weights = cowdat_rel_train$PREL)

cv.out <- cv.glmnet(x_biom_train, cowdat_train$CTPI, alpha=1, nlambda = 10000, weights =
cowdat_rel_train$PREL, nfolds = 10)
bestlam <- cv.out$lambda.min

lasso.mod.biom <- glmnet(x_biom_train, cowdat_train$CTPI, alpha=1, lambda=bestlam, intercept = T,
weights = cowdat_rel_train$PREL)

ytrain <- cowdat_train$CTPI
ytest <- cowdat_test$CTPI

paste('R^2 Full Training Model')

yhat <- predict(lasso.mod, newx = x_train)
y <- ytrain
rsq <- ifelse(is.na(cor(yhat, y)^2), 0, cor(yhat, y)^2 )
rsq
out.r.full[index] <- rsq

paste('R^2 Type Training Model')

yhat <- predict(lasso.mod.type, newx = x_type_train)
y <- ytrain
rsq2 <- ifelse(is.na(cor(yhat, y)^2 ), 0, cor(yhat, y)^2 )
rsq2
out.r.type[index] <- rsq2

paste('Gain in Training R^2 Biom above Type')
rsq - rsq2
out.r.biomgain.training[index] <- rsq - rsq2

paste('R^2 Biom Training Model')

yhat <- predict(lasso.mod.biom, newx = x_biom_train)
y <- ytrain
rsq2 <- ifelse(is.na(cor(yhat, y)^2 ), 0, cor(yhat, y)^2 )
rsq2
out.r.biom[index] <- rsq2

paste('Gain in Training R^2 Type above Biom')
rsq - rsq2

paste('R^2 for Validation Data')
yhat <- predict(lasso.mod, newx = x_test)
y <- ytest
rsq <- ifelse(is.na(cor(yhat, y)^2 ), 0, cor(yhat, y)^2 )
rsq

```

```

out.r.test[index] <- rsq

paste('Gain in Test R^2 Biom above Type')
yhat <- predict(lasso.mod.type, newx = x_type_test)
y <- ytest
rsq.type <- ifelse(is.na(cor(yhat, y)^2 ), 0, cor(yhat, y)^2 )
rsq - rsq.type
out.r.biomgain.test[index] <- rsq - rsq.type

##Z_LAME – Lameness

set.seed(1212)
index = 15

hist(cowdat.health$Z_LAME, main = 'Lameness', xlab = 'PTA Value')

x_train <- model.matrix( ~ .-1, cowdat.health.x.train)
x_test <- model.matrix( ~ .-1, cowdat.health.x.test)

x_type_train <- model.matrix( ~ .-1, cowdat.health.type.train)
x_type_test <- model.matrix( ~ .-1, cowdat.health.type.test)

x_biom_train <- model.matrix( ~ .-1, cowdat.health.biom.train)
x_biom_test <- model.matrix( ~ .-1, cowdat.health.biom.test)

cv.out <- cv.glmnet(x_train, cowdat.health.train$Z_LAME , alpha=1, nlambda = 10000, weights =
cowdat.health.rel.train$TREL, nfolds = 5)
bestlam <- cv.out$lambda.min

lasso.mod <- glmnet(x_train, cowdat.health.train$Z_LAME, alpha=1, lambda=bestlam, intercept = T,
weights = cowdat.health.rel.train$TREL)
coef(lasso.mod)
out.coef <- cbind(out.coef,matrix(coef(lasso.mod)))
colnames(out.coef)[index] <- 'Lameness'

cv.out <- cv.glmnet(x_type_train, cowdat.health.train$Z_LAME , alpha=1, nlambda = 10000, weights =
cowdat.health.rel.train$TREL, nfolds = 5)
bestlam <- cv.out$lambda.min

lasso.mod.type <- glmnet(x_type_train, cowdat.health.train$Z_LAME, alpha=1, lambda=bestlam,
intercept = T, weights = cowdat.health.rel.train$TREL)

cv.out <- cv.glmnet(x_biom_train, cowdat.health.train$Z_LAME , alpha=1, nlambda = 10000, weights =
cowdat.health.rel.train$TREL, nfolds = 5)
bestlam <- cv.out$lambda.min

lasso.mod.biom <- glmnet(x_biom_train, cowdat.health.train$Z_LAME, alpha=1, lambda=bestlam,
intercept = T, weights = cowdat.health.rel.train$TREL)

ytrain <- cowdat.health.train$Z_LAME

```

```

ytest <- cowdat.health.test$Z_LAME

paste('R^2 Full Training Model')

yhat <- predict(lasso.mod, newx = x_train)
y <- ytrain
rsq <- ifelse(is.na(cor(yhat, y)^2), 0, cor(yhat, y)^2 )
rsq
out.r.full[index] <- rsq

paste('R^2 Type Training Model')

yhat <- predict(lasso.mod.type, newx = x_type_train)
y <- ytrain
rsq2 <- ifelse(is.na(cor(yhat, y)^2 ), 0, cor(yhat, y)^2 )
rsq2
out.r.type[index] <- rsq2

paste('Gain in Training R^2 Biom above Type')
rsq - rsq2
out.r.biomgain.training[index] <- rsq - rsq2

paste('R^2 Biom Training Model')

yhat <- predict(lasso.mod.biom, newx = x_biom_train)
y <- ytrain
rsq2 <- ifelse(is.na(cor(yhat, y)^2 ), 0, cor(yhat, y)^2 )
rsq2
out.r.biom[index] <- rsq2

paste('Gain in Training R^2 Type above Biom')
rsq - rsq2

paste('R^2 for Validation Data')
yhat <- predict(lasso.mod, newx = x_test)
y <- ytest
rsq <- ifelse(is.na(cor(yhat, y)^2 ), 0, cor(yhat, y)^2 )
rsq
out.r.test[index] <- rsq

paste('Gain in Test R^2 Biom above Type')
yhat <- predict(lasso.mod.type, newx = x_type_test)
y <- ytest
rsq.type <- ifelse(is.na(cor(yhat, y)^2 ), 0, cor(yhat, y)^2 )
rsq - rsq.type
out.r.biomgain.test[index] <- rsq - rsq.type

```

Formatting the Linear LASSO

```
library('knitr')
```

```

linear.lasso.results <- as.matrix(data.frame('Full' = out.r.full, 'Type' = out.r.type, 'Biom' = out.r.biom,
'BiomGainTrain' = out.r.biomgain.training, 'Test' = out.r.test, 'BiomGainTest' = out.r.biomgain.test))

ylist <- c('CTPI', 'NM', 'PTAM', 'PTAF', 'PTAP', 'FeedEff', 'PL', 'HCR', 'CCR', 'SCS', 'Fert.Index', 'CE',
'SB', "Z_MAST", "Z_LAME", "Z_MET", "Z_RP", "Z_KET", "Z_DA", "Z_Calf_LIV", "Z_Calf_Scours",
"Z_Calf_Resp")
rownames(linear.lasso.results) <- c('CTPI', 'NM', 'PTAM', 'PTAF', 'PTAP', 'FeedEff', 'PL', 'HCR', 'CCR',
'SCS', 'Fert.Index', 'CE', 'SB', "Z_MAST", "Z_LAME", "Z_MET", "Z_RP", "Z_KET", "Z_DA",
"Z_Calf_LIV", "Z_Calf_Scours", "Z_Calf_Resp")

colnames(linear.lasso.results) <- c('R^2 Full Training Model', 'R^2 Type Training Model', 'R^2 Biom
Training Model', 'R^2 Biom Gain Training', 'R^2 Full Test Model', 'R^2 Biom Gain Test')

kable(linear.lasso.results, caption = 'Results of Linear LASSO', digits = 3)
write.csv(round(linear.lasso.results,3), file = "Results_LinearLASSO.csv")

rownames(out.coef) <- rownames(coef(lasso.mod))
out.coef <- cbind(out.coef, apply(abs(out.coef)>0, 1, FUN = sum))
colnames(out.coef)[ncol(out.coef)] <- 'Total Significant Terms'
write.csv(round(out.coef,3), file = "Results_LinearLASSO_Coef.csv")

#####

#Spline Models for Nonlinearity

##TPI - Total Performance Index

library('mgcv')
set.seed(12)

# setup datasets

x_type_train <- model.matrix(~.-1, cowdat_type)
x_biom_train <- model.matrix(~.-1, cowdat_biom)
w.train <- cowdat_rel$PREL

y.train <- cowdat$CTPI

# start by fitting optimal base model of type traits using linear lasso

cv.out <- cv.glmnet(x_type_train, y.train, alpha=1, nlambda = 10000, weights = w.train, nfolds = 10)
bestlam <- cv.out$lambda.min

lasso.mod.type <- glmnet(x_type_train, y.train, alpha=1, lambda=bestlam, intercept = T, weights =
w.train)

```



```

#temp.keep <- which(abs(coef(lasso.mod.type))>0)[-1] - 1
#x_type_train <- x_type_train[,temp.keep] # keep only sig type traits
#x_type_test <- x_type_test[,temp.keep]

# create function statement for this base model

temp.form <- 'y.train ~ '

for (i in which(abs(coef(lasso.mod.type))>0)[-1]){

  temp.form <- paste(temp.form, ' + ', rownames(coef(lasso.mod.type))[i])

}

temp.form.base <- as.formula(temp.form)

temp.form <- paste(temp.form, ' + ', 's(biom)')

temp.form.full <- as.formula(temp.form)

# add a smoothing spline for each biometric on top of base type trait model

result.df <- c()
result.pv <- c()
result.r2gain.train <- c()
result.r2gain.test <- c()

for (i in 1:ncol(x_biom_train)){

  #x.temp = data.frame(x_type_train)
  #gam.out = gam(y.train ~ . + s(x_biom_train[,1]), data = x.temp)

  x.temp = data.frame(cbind(y.train, x_type_train, x_biom_train[,i]))
  names(x.temp)[ncol(x.temp)] <- 'biom'
  gam.out.full = gam( temp.form.full, data = x.temp, weights = w.train)
  temp <- summary(gam.out.full)

  result.df[i] <- temp$edf # estimated degrees of freedom
  result.pv[i] <- temp$s.pv # pvalue of significance of spline fit

  gam.out.base = gam( temp.form.base, data = x.temp, weights = w.train)
  temp2 <- summary(gam.out.base)

  result.r2gain.train[i] <- temp$r.sq - temp2$r.sq

}

results.spline.tpi <- as.matrix(data.frame(df = result.df, p = result.pv, rtrain = result.r2gain.train))

```

```

rownames(results.spline.tpi) <- colnames(x_biom_train)
colnames(results.spline.tpi) <- c('DF Spline','Spline P-Value','R^2 Gain Training')

write.csv(round(results.spline.tpi, 3), file = 'Results_SplineModels/CTPI.csv')
kable(results.spline.tpi, caption = 'Biom Spline Results for TPI')

out.spline.pvals <- result.pv

##Z_LAME – Lameness

library('mgcv')
set.seed(12)

# setup datasets

x_type_train <- model.matrix(~.-1, cowdat.health.type)
x_biom_train <- model.matrix(~.-1, cowdat.health.biom)

y.train <- cowdat.health$Z_LAME
w.train <- cowdat.health.rel$PREL

# start by fitting optimal base model of type traits using linear lasso

cv.out <- cv.glmnet(x_type_train, y.train, alpha=1, nlambda = 10000, weights = w.train, nfolds = 5)
bestlam <- cv.out$lambda.min

lasso.mod.type <- glmnet(x_type_train, y.train, alpha=1, lambda=bestlam, intercept = T, weights =
w.train)

#temp.keep <- which(abs(coef(lasso.mod.type))>0)[-1] - 1
#x_type_train <- x_type_train[,temp.keep] # keep only sig type traits
#x_type_test <- x_type_test[,temp.keep]

# create function statement for this base model

temp.form <- 'y.train ~ '

for (i in which(abs(coef(lasso.mod.type))>0)[-1]){

  temp.form <- paste(temp.form, ' + ', rownames(coef(lasso.mod.type))[i])

}

temp.form.base <- as.formula(temp.form)

temp.form <- paste(temp.form, ' + ', 's(biom)')

temp.form.full <- as.formula(temp.form)

```

```

# add a smoothing spline for each biometric on top of base type trait model

result.df <- c()
result.pv <- c()
result.r2gain.train <- c()
result.r2gain.test <- c()

for (i in 1:ncol(x_biom_train)){

  #x.temp = data.frame(x_type_train)
  #gam.out = gam(y.train ~ . + s(x_biom_train[,1]), data = x.temp)

  x.temp = data.frame(cbind(y.train, x_type_train, x_biom_train[,i]))
  names(x.temp)[ncol(x.temp)] <- 'biom'
  gam.out.full = gam( temp.form.full, data = x.temp, weights = w.train)
  temp <- summary(gam.out.full)

  result.df[i] <- temp$edf # pvalue of significance of spline fit
  result.pv[i] <- temp$ss.pv # smoothing term

  gam.out.base = gam( temp.form.base, data = x.temp, weights = w.train)
  temp2 <- summary(gam.out.base)

  result.r2gain.train[i] <- temp$r.sq - temp2$r.sq

}

results.spline.zlam <- as.matrix(data.frame(df = result.df, p = result.pv, rtrain = result.r2gain.train))

rownames(results.spline.zlam) <- colnames(x_biom_train)
colnames(results.spline.zlam) <- c('DF Spline','Spline P-Value','R^2 Gain Training')

write.csv(round(results.spline.zlam, 3), file = 'Results_SplineModels/ZLAME.csv')
kable(results.spline.zlam, caption = 'Biom Spline Results for Lameness')

out.spline.pvals <- cbind(out.spline.pvals, result.pv)

## Summarizing Spline Results

rownames(out.spline.pvals) <- colnames(x_biom_train)
colnames(out.spline.pvals) <- c('CTPI', 'NM', 'PTAM', 'PTAF', 'PTAP', 'FeedEff', 'PL', 'HCR', 'CCR',
'SCS', 'Fert.Index', 'CE', 'SB', "Z_MAST", "Z_LAME", "Z_MET", "Z_RP", "Z_KET", "Z_DA",
"Z_Calf_LIV", "Z_Calf_Scours", "Z_Calf_Resp")

out.spline.pvals <- cbind(out.spline.pvals, apply(abs(out.spline.pvals)<=0.05, 1, FUN = sum))
colnames(out.spline.pvals)[ncol(out.spline.pvals)] <- 'Total Significant Terms'

```

```
write.csv(round(out.spline.pvals,3), 'Results_SplineSummary.csv')
kable(out.spline.pvals[,ncol(out.spline.pvals)], digits = 3, caption = 'Summary of Spline Results')
```

```
#####
```

```
#Regression Trees
```

```
rf.gain <- c()
rf.gain.test <- c()
bag.biom.complex <- c()
bag.biom.depth <- c()
bag.type.complex <- c()
bag.type.depth <- c()
bag.biom.r <- c()
bag.gain <- c()
bag.biom.r.test <- c()
bag.gain.test <- c()

x_train <- data.frame(model.matrix( ~ .-1, cowdat_x_train))
bag.coefcount <- rep(0, ncol(x_train))
names(bag.coefcount) <- names(x_train)
```

```
##TPI - Total Performance Index
```

```
set.seed(1212)
```

```
library('rpart')
library('ggplot2')
```

```
index = 1
```

```
x_train <- data.frame(model.matrix( ~ .-1, cowdat_x_train))
x_test <- data.frame(model.matrix( ~ .-1, cowdat_x_test))

x_type_train <- data.frame(model.matrix( ~ .-1, cowdat_type_train))
x_type_test <- data.frame(model.matrix( ~ .-1, cowdat_type_test))
```

```
w.train <- cowdat_rel_train$PREL
w.test <- cowdat_rel_test$PREL
```

```
y.train <- cowdat_train$CTPI
y.test <- cowdat_test$CTPI
# Bagging - full
```

```
k = 10
```

```
library('gbm')
```

```
set.seed(61916)
```

```

bag.out.full.1 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=1)

set.seed(61916)
bag.out.full.2 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=2)

set.seed(61916)
bag.out.full.3 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=3)

set.seed(61916)
bag.out.full.4 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=4)

set.seed(61916)
bag.out.full.5 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=5)

temp <- cbind(bag.out.full.1$cv.error, bag.out.full.2$cv.error, bag.out.full.3$cv.error,
bag.out.full.4$cv.error, bag.out.full.5$cv.error)

bag.cv.grid.base <- c()
temp.mse <- c()

for (j in 1:5){

  temp.min <- min(temp[,j])
  bag.cv.grid.base[j] <- round(min(which(temp[,j] < (temp.min + 0.01*temp.min))), -2)
  if (bag.cv.grid.base[j] == 0){
    bag.cv.grid.base[j] = 100 # minimum tree depth of 100
  }
  temp.mse[j] <- temp[bag.cv.grid.base[j],j]

}

temp.min <- min(temp.mse)
g <- min(which(temp.mse < (temp.min + 0.01 * temp.min))) # best tree complexity
B <- bag.cv.grid.base[g] # best tree depth

bag.biom.complex[index] <- g
bag.biom.depth[index] <- B

set.seed(61916)
bag.out.full.final <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage
= 0.001, n.trees=B, cv.folds = k, interaction.depth=g)

temp <- summary(bag.out.full.final)
temp <- data.frame(var = temp$var, varimp = temp$rel.inf)
temp <- temp[order(temp$varimp, decreasing = T),]

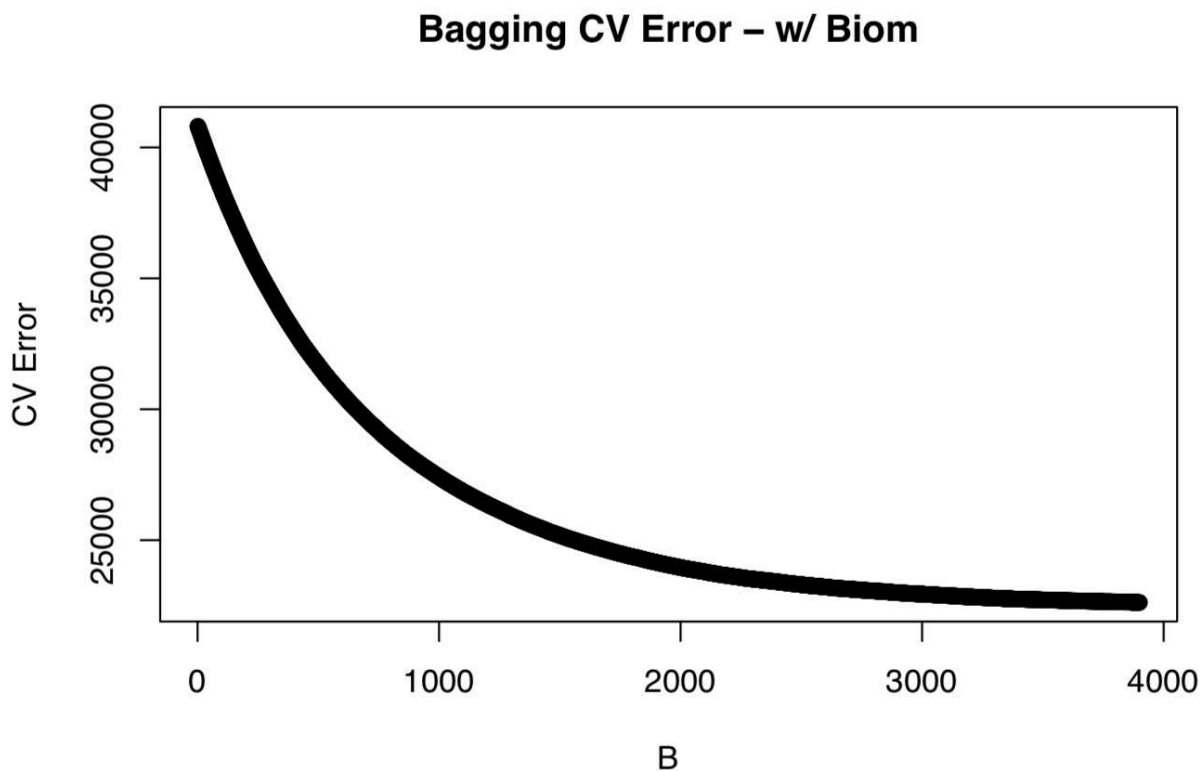
```



```
temp <- temp[1:15,]
ggplot(data=temp, aes(x=var, y=varimp)) + geom_bar(stat="identity" , fill = 'slateblue4') + coord_flip() +
xlab('Variable') + ylab('Variable Importance') + ggtitle('Boosted Tree Variable Importance')
```

```
for(i in 1:nrow(temp)){
  # keep track of how many times each coef is included in top 15 across all response models
  indextemp <- which(temp$var[i] == names(bag.coefcount))
  bag.coefcount[indextemp] <- bag.coefcount[indextemp] + 1
}
```

```
plot(1:B, bag.out.full.final$cv.error, xlab = 'B', ylab = 'CV Error', main = 'Bagging CV Error - w/ Biom')
```



```
# Bagging - base
```

```
set.seed(61916)
bag.out.base.1 <- gbm(y.train ~ ., data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=1)
```

```
set.seed(61916)
bag.out.base.2 <- gbm(y.train ~ ., data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=2)
```

```
set.seed(61916)
```

```

bag.out.base.3 <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=3)

set.seed(61916)
bag.out.base.4 <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=4)

set.seed(61916)
bag.out.base.5 <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=5)

temp <- cbind(bag.out.base.1$cv.error, bag.out.base.2$cv.error, bag.out.base.3$cv.error,
bag.out.base.4$cv.error, bag.out.base.5$cv.error)

bag.cv.grid.base <- c()
temp.mse <- c()

for (j in 1:5){

  temp.min <- min(temp[,j])
  bag.cv.grid.base[j] <- round(min(which(temp[,j] < (temp.min + 0.01*temp.min))), -2)
  if (bag.cv.grid.base[j] == 0){
    bag.cv.grid.base[j] = 100 # minimum tree depth of 100
  }
  temp.mse[j] <- temp[bag.cv.grid.base[j],j]

}

temp.min <- min(temp.mse)
g <- min(which(temp.mse < (temp.min + 0.01 * temp.min))) # best tree complexity
B <- bag.cv.grid.base[g] # best tree depth

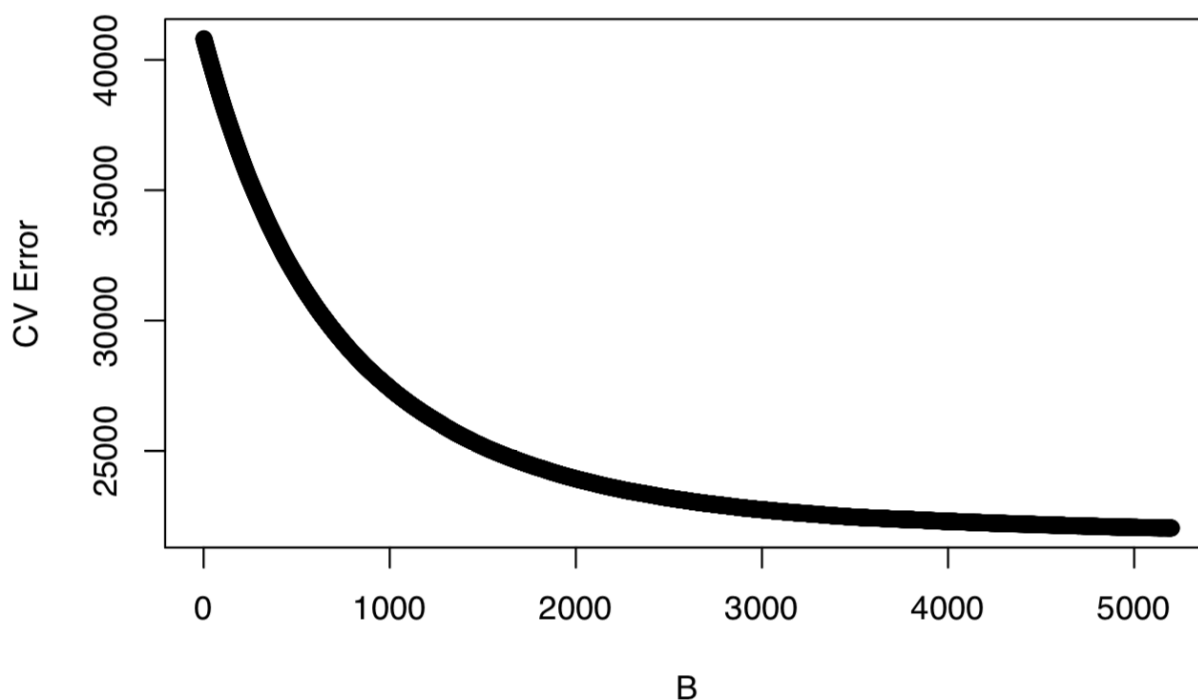
bag.type.complex[index] <- g
bag.type.depth[index] <- B

set.seed(61916)
bag.out.base.final <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=B, cv.folds = k, interaction.depth=g)

plot(1:B, bag.out.base.final$cv.error, xlab = 'B', ylab = 'CV Error', main = 'Bagging CV Error - Base')

```

Bagging CV Error – Base



```
bag.biom.r[index] <- cor(y.train, bag.out.full.final$fit)^2
bag.gain[index] <- cor(y.train, bag.out.full.final$fit)^2 - cor(y.train, bag.out.base.final$fit)^2
```

```
yhat_full <- predict(bag.out.full.final, newdata = x_test)
yhat_base <- predict(bag.out.base.final, newdata = x_test)
```

```
bag.biom.r.test[index] <- cor(yhat_full, y.test)^2
bag.gain.test[index] <- cor(yhat_full, y.test)^2 - cor(yhat_base, y.test)^2
```

```
##Z_LAME – Lameness
```

```
set.seed(1212)
```

```
library('rpart')
library('ggplot2')
```

```
index = 15
```

```
x_type_train <- data.frame(model.matrix(~.-1, cowdat.health.type))
x_biom_train <- data.frame(model.matrix(~.-1, cowdat.health.biom))
```

```
x_train <- cbind(x_type_train, x_biom_train)
```

```

x_train <- data.frame(x_train)

y.train <- cowdat.health$Z_LAME
w.train <- cowdat.health.rel$TREL

# Bagging - full

k = 5

library('gbm')

set.seed(61916)
bag.out.full.1 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=1)

set.seed(61916)
bag.out.full.2 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=2)

set.seed(61916)
bag.out.full.3 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=3)

set.seed(61916)
bag.out.full.4 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=4)

set.seed(61916)
bag.out.full.5 <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage =
0.001, n.trees=7000, cv.folds = k, interaction.depth=5)

temp <- cbind(bag.out.full.1$cv.error, bag.out.full.2$cv.error, bag.out.full.3$cv.error,
bag.out.full.4$cv.error, bag.out.full.5$cv.error)

bag.cv.grid.base <- c()
temp.mse <- c()

for (j in 1:5){

  temp.min <- min(temp[,j])
  bag.cv.grid.base[j] <- round(min(which(temp[,j] < (temp.min + 0.01*temp.min))), -2)
  if (bag.cv.grid.base[j] == 0){
    bag.cv.grid.base[j] = 100 # minimum tree depth of 100
  }
  temp.mse[j] <- temp[bag.cv.grid.base[j],j]
  # find index within 1% of the absolute minima, and then round to nearest 100

}

temp.min <- min(temp.mse)
g <- min(which(temp.mse < (temp.min + 0.01 * temp.min))) # best tree complexity

```

```

B <- bag.cv.grid.base[g] # best tree depth

bag.biom.complex[index] <- g
bag.biom.depth[index] <- B

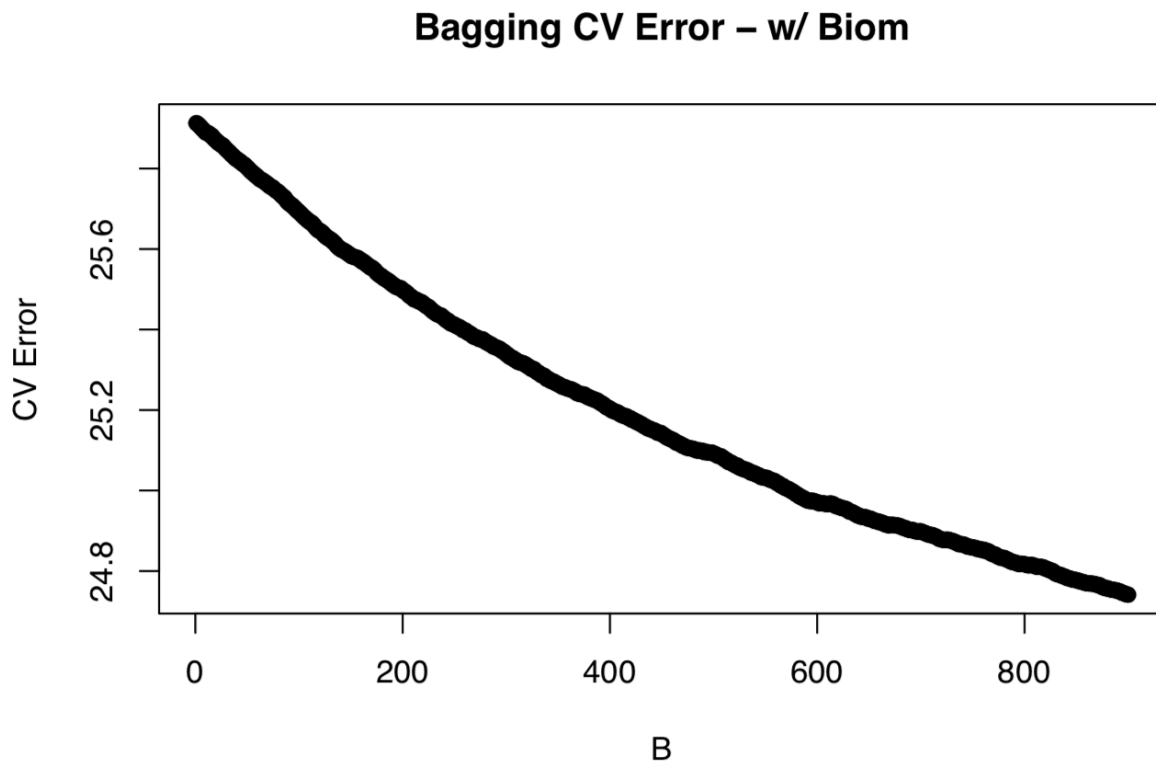
set.seed(61916)
bag.out.full.final <- gbm(y.train ~. , data = x_train, weights = w.train, distribution="gaussian", shrinkage
= 0.001, n.trees=B, cv.folds = k, interaction.depth=g)

temp <- summary(bag.out.full.final)
temp <- data.frame(var = temp$var, varimp = temp$rel.inf)
temp <- temp[order(temp$varimp, decreasing = T),]
temp <- temp[1:20,]
ggplot(data=temp, aes(x=var, y=varimp)) + geom_bar(stat="identity" , fill = 'slateblue4') + coord_flip() +
xlab('Variable') + ylab('Variable Importance') + ggtitle('Boosted Tree Variable Importance')

for(i in 1:nrow(temp)){
  # keep track of how many times each coef is included in top 15 across all response models
  indextemp <- which(temp$var[i] == names(bag.coefcount))
  bag.coefcount[indextemp] <- bag.coefcount[indextemp] + 1
}

plot(1:B, bag.out.full.final$cv.error, xlab = 'B', ylab = 'CV Error', main = 'Bagging CV Error - w/ Biom')

```



Bagging - base

```
set.seed(61916)
```



```

bag.out.base.1 <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=1)

set.seed(61916)
bag.out.base.2 <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=2)

set.seed(61916)
bag.out.base.3 <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=3)

set.seed(61916)
bag.out.base.4 <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=4)

set.seed(61916)
bag.out.base.5 <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=7000, cv.folds = k, interaction.depth=5)

temp <- cbind(bag.out.base.1$cv.error, bag.out.base.2$cv.error, bag.out.base.3$cv.error,
bag.out.base.4$cv.error, bag.out.base.5$cv.error)

bag.cv.grid.base <- c()
temp.mse <- c()

for (j in 1:5){

  temp.min <- min(temp[,j])
  bag.cv.grid.base[j] <- round(min(which(temp[,j] < (temp.min + 0.01*temp.min))), -2)
  if (bag.cv.grid.base[j] == 0){
    bag.cv.grid.base[j] = 100 # minimum tree depth of 100
  }
  temp.mse[j] <- temp[bag.cv.grid.base[j],j]
  # find index within 1% of the absolute minima, and then round to nearest 100

}

temp.min <- min(temp.mse)
g <- min(which(temp.mse < (temp.min + 0.01 * temp.min))) # best tree complexity
B <- bag.cv.grid.base[g] # best tree depth

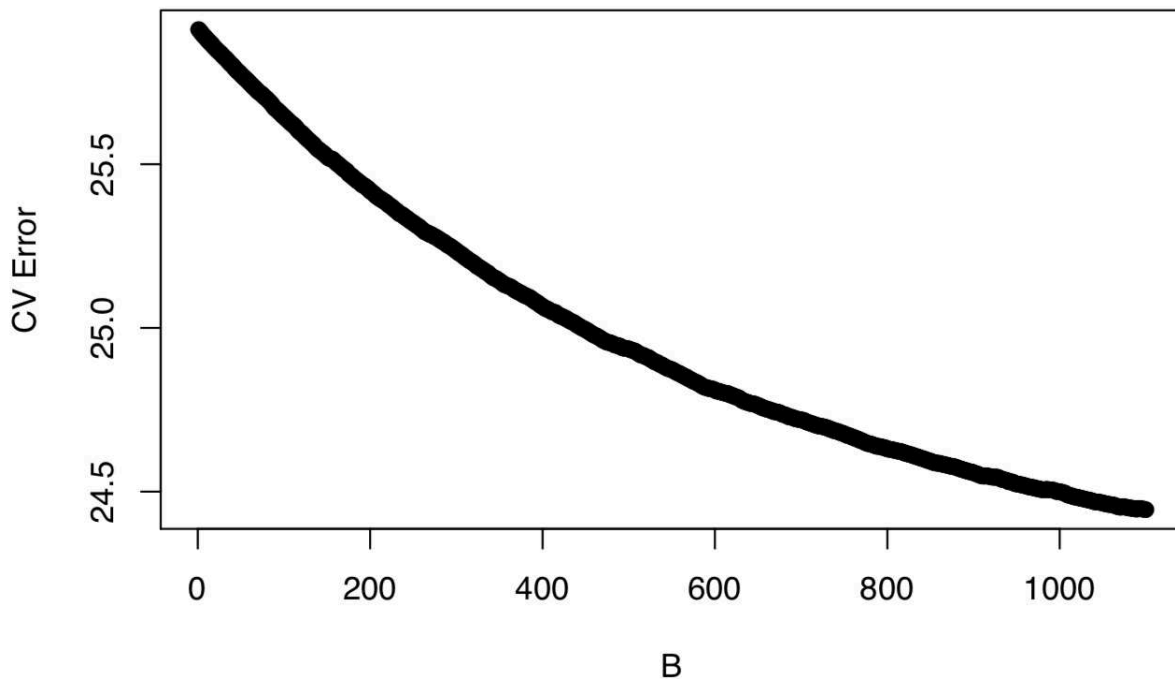
bag.type.complex[index] <- g
bag.type.depth[index] <- B

set.seed(61916)
bag.out.base.final <- gbm(y.train ~. , data = x_type_train, weights = w.train, distribution="gaussian",
shrinkage = 0.001, n.trees=B, cv.folds = k, interaction.depth=g)

plot(1:B, bag.out.base.final$cv.error, xlab = 'B', ylab = 'CV Error', main = 'Bagging CV Error - Base')

```

Bagging CV Error – Base



```

bag.biom.r[index] <- cor(y.train, bag.out.full.final$fit)^2
bag.gain[index] <- cor(y.train, bag.out.full.final$fit)^2 - cor(y.train, bag.out.base.final$fit)^2

bag.gain.test[index] <- NA
bag.biom.r.test[index] <- NA

## Summarizing Regression Tree Results

library('knitr')

tree.results <- as.matrix(cbind(bag.biom.complex, bag.biom.depth, bag.type.complex, bag.type.depth,
bag.biom.r, bag.gain, bag.biom.r.test, bag.gain.test))

ylist <- c('CTPI', 'NM', 'PTAM', 'PTAF', 'PTAP', 'FeedEff', 'PL', 'HCR', 'CCR', 'SCS', 'Fert.Index', 'CE',
'SB', "Z_MAST", "Z_LAME", "Z_MET", "Z_RP", "Z_KET", "Z_DA", "Z_Calf_LIV", "Z_Calf_Scours",
"Z_Calf_Resp")
rownames(tree.results) <- c('CTPI', 'NM', 'PTAM', 'PTAF', 'PTAP', 'FeedEff', 'PL', 'HCR', 'CCR', 'SCS',
'Fert.Index', 'CE', 'SB', "Z_MAST", "Z_LAME", "Z_MET", "Z_RP", "Z_KET", "Z_DA", "Z_Calf_LIV",
"Z_Calf_Scours", "Z_Calf_Resp")

colnames(tree.results) <- c('Depth Biometric Tree', 'N Trees Biometric', 'Depth Base Tree', 'N Trees Base',
'R^2 Full Model', 'R^2 Gain Training', 'R^2 Test', 'R^2 Gain Test')

kable(tree.results, caption = 'Results of Regression Trees')

```

```
bag.coef.count <- data.frame(bag.coefcount)
colnames(bag.coef.count) <- 'Model Inclusion Count'
kable(bag.coef.count, caption = 'Counts of Significant Coeffs for Boosted Tree')

write.csv(tree.results, 'Results_TreeSummary.csv')
write.csv(bag.coef.count, 'Results_TreeCoef.csv')
```

Supplemental Table: LASSO Coefficient Results for all PTA & Health Models

	CTPI	NM	PTAM	PTAF	PTAP	FeedEff	PL	HCR	CCR	SCS	Fertility Index	CE	SB	Mastitis	Lame	Metritis	RP	Ketosis	DA	Calf Liv.	Calf Scours	Calf Resp.	Total Significant Terms
PTA.Type	91.047	58.358	56.853	0.589	6.356	17.747	0.497	-0.009	0	0	0	0	0	0	0	0	0	0	0	0	0.435	0	9
UDC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FLC	0	0	0	0	0	0	0	0	0	-0.002	0	0	0	0	0	0	0	0	0	0	0	0	1
GPTA.STA	-22.742	-50.928	0	-4.507	0	-14.543	-0.527	0.207	0	0	0.079	0	0	-0.398	0	0	0	0	-0.118	0	0.316	-0.348	11
GPTA.STR	154.442	132.535	486.231	6.186	16.824	30.331	0.75	0	0	0	0.068	0.033	0	0	0	0	0	0	0	0	0	0	9
GPTA.BDE	-163.855	-131.08	-554.732	0	-19.776	-30.495	-0.952	0	-0.21	0	-0.036	0.084	0	0	-0.369	0	0	0	-0.929	0	0.176	0	12
GPTA.DRM	70.181	84.586	438.625	8.687	12.123	33.377	0	-0.406	-0.415	0.014	-0.537	0	0	-1.412	-0.163	0	0	-0.45	-0.472	-0.228	0.081	0	16
GPTA.RPA	23.345	23.208	35.442	0.711	1.342	4.413	0.275	0.004	0.172	0	0.1	0	0	0	0	0	0	0.052	0	0	0	0	11
GPTA.TRW	-22.816	-30.134	-16.727	-1.776	-1.308	-7.963	-0.25	0.067	0	0	0	0.049	0	0	0	0.449	0	0	0	0	0	0	10
GPTA.RLS	21.366	19.822	26.255	0.015	1.362	4.214	0.224	0.167	0.021	0	0.064	0.077	0.004	0	0	0	0	-0.071	-0.076	0	0	0	14
GPTA.RLR	6.089	5.196	0	0	0	0	0	0.146	0	0	0	0.013	0	0	0	0	0	0	0	0	0	0	4
GPTA.FTA	29.522	23.649	46.298	3.255	3.001	12.873	-0.015	-0.356	0	0	0.039	0.015	0	0	0	0	0	0	0	0	0	0	10
GPTA.FLS	17.389	5.391	0	0	-0.747	-5.221	0.371	0.352	0.108	0	0.08	0	0	0	0	0	0	0	0	0	0.916	0	9
GPTA.FUA	-5.286	-7.974	-50.49	0	-2.361	-10.187	0.262	-0.112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
GPTA.RUH	27.202	13.611	135.2	2.987	2.243	6.528	-0.115	0.103	0	0	0	0	0	0	-0.026	0	-0.81	0	-0.258	0	0	0	11
PTA.RUW	3.941	2.865	0	0	0.296	0.436	0	0.001	0	0	0	0	0	0	-0.009	0	-0.009	0	-0.002	0	0	0	8
GPTA.UCL	-0.301	0	-30.947	0	-1.353	-4.028	0.095	-0.045	0	0	0	0	0	0	0	0	0	0	0	0	0.103	0	7
GPTA.UDP	-29.354	-13.69	-140.183	-4.557	-5.441	-8.793	0.111	0	0.034	-0.027	0	0	0.284	0	0	0	0.548	0	0	0	0	0	11
GPTA.FTP	0	0	-43.094	0.162	-0.741	0.447	0	0.031	0.053	0	0.065	-0.044	0	0	0	0	0	0	0	0	0	0	8
GPTA.RTP	-1.237	-4.571	0	0	0	-0.249	0.059	0	0	0	0	-0.053	0	0	0	0	0	0	-0.091	0	0	0	6
GPTA.TLG	-18.793	-17.369	0	-1.326	-0.604	-2.724	-0.128	-0.255	-0.087	0	-0.12	0.112	0.046	0	0	0	0	-0.176	-0.119	0	0	0	13
GPTA.BSC	0	0	0	0	0	0	0	0	0	0	0	0.039	0	0	0	0	0	0	0	0	0	-0.333	2
Z1_front	0	0	4.174	0	0.698	0	0	-0.087	0	0	-0.038	0	0	0	0	0	0	0	0	0	0	0	4
Z2	1.438	0.842	-23.991	0	-0.087	0	0	0.046	0	0.032	0	-0.02	0	0	0	0	0	0	0	0	0	0	7
Z3_front	14.418	13.632	8.586	1.7	1.049	3.607	0.084	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
Z4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z7_length	0	-1.09	-6.038	0	-0.463	-0.762	0	0	0	0	0.003	0.014	0	0	0	0	0	0	0	0	0	0	6
Z9_poly	9.09	5.348	0	0.964	-0.056	0	0	0.07	0.071	-0.01	0.048	0	0	0	0	0	0	0	0	0	0	0	8
Z10_poly	4.759	4.626	0	0.105	0	0.473	0.136	0	0.001	0	-0.02	-0.026	0	2.091	0	-0.517	0	0	0	0	0	0	10
Z11_poly	-3.583	-0.679	0	-0.364	0	-1.087	0.06	-0.022	0	0	0	0	0	0	0	0	1.529	0	0	0	0	0	7
Z11_linear	4.771	2.921	0	0.466	0.351	3.452	0	0.005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
NFP_LF	-3.918	-1.373	-20.131	0	-0.766	-0.921	0	0	0	0	0	0	0	0	0	0	4.785	0	0	0	0	0	6
NFPF_LF	7.818	6.322	8.451	0	0.149	0.215	0.072	0.065	0	0	0.012	-0.029	0	0	0	0	-1.092	0	0	0	0	0	10
NPA	9.374	8.05	0	0.464	0.256	1.896	0.066	0.027	0	0	0.007	0	-0.03	0	0	0	0	0	0	0	0	0	9
ULRP_V1	0.917	1.811	0	0	0	0	0.012	0	0	0	0	0	0	0	0	2.663	0	0.965	0	0	0	0	5
MTP_V1	0	-1.656	0	-1.914	0	0	0.005	0	0	0	0	0	0	0	0	0	0	6.375	0	0	0	0	4
CTP_V1	-18.012	-13.069	-44.328	-0.468	-0.949	-2.938	-0.018	-0.11	-0.138	0	-0.089	0	0.007	0	-3.855	0	0	0	0	0	0	0	12
CLTR_V1	0	0.019	0	0	0	1.692	-0.078	0	-0.033	0	-0.064	0	0	0	0	0	0	0	0	0	0	0	5
NRP	6.326	9.976	18.087	2.583	0.349	3.551	0.018	-0.005	-0.045	0	-0.037	0	0.008	0	0	0	-0.173	0	0	0	0	0	12
SMRP_V1	1.066	0	-2.826	0	-0.013	0	0	0	0	0	0	0	0	0	-5.862	0	0	0	0	0	0	0	4
MNRP_V1	0.24	0.771	-2.462	0	-0.361	0	0.044	0	0	0	0.008	0	0	0	0	0	0	0	0	0	0	0	6
MDP_V1	1.904	0	0	0	0	-0.51	0	0.047	0.056	0	0.061	0	-0.061	0	0	0	0	0	0	0	4.638	7	
NDP_V1	0	0	0	-0.164	0	0	0.014	0	0	0	-0.003	-0.054	0	0	-14.423	0	0	0	0	0	0	5	
NaDP_V1	-5.774	-4.031	7.258	0	-0.284	-2.062	-0.007	0	-0.059	0	0	0.03	0	0	0	0	0	0	0	0	0	0	8
MIP_V1	-13.287	-9.722	-32.84	-0.583	-0.708	-2.028	-0.066	-0.075	-0.051	0	-0.025	0	0.013	0	10.724	0	0	0	0	0	0	0	12
MIPP_V1	0	-0.004	-12.011	-0.209	0	-0.35	0	-0.023	0	0	0	0	0	0	0	0	0	0.052	0	0	0	0	6
NIP_V1	5.957	4.319	0	0.227	0	0.627	0.052	0.021	0	0	0	0.006	0	0	0	0	0	0	0	0	4.86	8	
NaTLP	-6.204	0	-4.846	0	-0.963	-2.309	0.001	0	0	0	-0.002	0	0	0	0	0	0	0	0	0	0.364	7	
MTLP_V1	2.736	4.836	7.378	0	0.039	0.646	0	-0.025	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
STLP_V1	0	0	0	-2.572	-5.583	0	0.15	0	0	0	0	0	0	0	0	0	-1.842	0	0	0	0	0	4
ULTLP_V1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.931	0	0	0	0	0	1
SMLP_V1	10.37	5.373	0	0.565	0	0	0.229	0	0	0	0	0	0	0	0	0.387	0	0	0	0	0	0	5
MNLP_V1	-0.97	0	0	0	0	0	0.009	-0.096	0	0	-0.006	0	0	0	0	0	0	0	0	0	0	0	4
EFSPR_V1	-3.053	0	-17.798	0	0	0	0	0	0	0	0	-0.015	0	0	0	0	0	0	0	0	0	0	3
EF SRL_V1	0	-0.006	0	0	0	-0.034	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
ECSR_V1	0	0	0.236	0	0.44	2.746	0.006	-0.248	-0.061	0	-0.076	0	0	0	0	0	0	0	0	0	0	0	7
ETSR_V1	8.889	11.26	16.348	1.625	0	0	0.024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
ET SRL_V1	0	0	0	0	0	-1.358	0	0	0	0	0.031	0	0	1.941	0	0	-2.71	0	0	0	0	0	4
ESSRP_V1	0	0	0	0	0	-0.007	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ESSRL_V1	-8.32	-8.487	0	0	0	-0.012	0	-0.107	-0.059	0	-0.099	0	0	0	0	0	0	0	0	0	2.974	7	
MTP_V1.1	1.9	0	0	0	2.869	6.518	0	0	0	0	0	0	0	0	0	0	-1.951	0	0	0	0	0	4
OES_V1	0	0	47.237	0	0	0	0	0.167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
OEAA_V1	0	0	0	0	-0.316	0	-0.069	-0.02	0	0	0	-0.046	0	0	0	0	0	0	0	0.009	0	0	5
MSP_V9	0	0	0	0	0	0	0	0	0	0	-0.009	0	0	0	0	0	0	0	0	0	0	0	1
CNSP_V14	0	0	0	0	-0.201	0	0	0	0	0	0	0.074	0	0	0	0	0	0	0	0	0	0	2
CLP_V6	-10.854	-13.099	-65.036	-1.355	-1.295	-4.217	0	0	0	0	0	0	-0.019	0	0	0	0	0	0	0	0	0	7
JLPL_V7	-1.358	-2.414	0	0	0	-0.681	0	-0.016	0	0	-0.003	0	0.01	0	0	0	0	0	0	0	0	0	6
JAS_V1	0	0	-4.826	0	0	0	0.007	0.086	0	0	-0.019	0	0	0	0	0	1.678	0	0	0	0	0	5
JMSR_V																							

Supplemental Tables: Full Results of Smoothing Spline Optimizations

Calf Livability				Calf Respiratory				Calf Scours			
	DF Spline	Spline P-Value	R ² Gain Training		DF Spline	Spline P-Value	R ² Gain Training		DF Spline	Spline P-Value	R ² Gain Training
Z1_front	1.0	0.35	0.000	Z1_front	1.0	0.13	0.006	Z1_front	1.0	0.63	-0.002
Z2	1.0	0.40	-0.001	Z2	1.0	0.48	0.001	Z2	1.0	0.74	-0.002
Z3_front	1.0	0.30	0.000	Z3_front	1.0	0.65	0.000	Z3_front	1.0	0.28	0.000
Z4	2.5	0.54	0.004	Z4	1.0	0.80	0.000	Z4	1.4	0.59	0.000
Z7_length	1.0	0.23	0.001	Z7_length	5.4	0.38	0.014	Z7_length	1.0	0.74	-0.002
Z9_poly	4.2	0.11	0.020	Z9_poly	6.9	0.70	0.003	Z9_poly	1.0	0.45	-0.001
Z10_poly	1.0	0.62	-0.002	Z10_poly	1.0	0.98	0.000	Z10_poly	1.1	0.40	0.000
Z11_poly	1.4	0.66	0.000	Z11_poly	1.0	0.02	0.016	Z11_poly	3.5	0.49	0.005
Z11_linear	1.0	0.62	-0.002	Z11_linear	1.0	0.55	0.001	Z11_linear	1.0	0.18	0.002
NFP_LF	1.9	0.10	0.013	NFP_LF	1.6	0.61	0.001	NFP_LF	1.5	0.46	0.002
NFPP_LF	1.0	0.51	-0.002	NFPP_LF	1.6	0.32	0.004	NFPP_LF	1.0	0.70	-0.002
NPA	1.0	0.93	-0.003	NPA	1.0	0.28	0.003	NPA	1.0	0.18	0.002
ULRP_V1	2.6	0.58	0.003	ULRP_V1	1.7	0.53	0.002	ULRP_V1	5.5	0.28	0.014
MTP_V1	1.1	0.29	0.002	MTP_V1	1.0	0.37	-0.001	MTP_V1	1.3	0.34	0.002
CTP_V1	1.0	0.91	-0.003	CTP_V1	1.0	0.51	0.001	CTP_V1	1.0	0.65	-0.002
CLTR_V1	1.9	0.29	0.007	CLTR_V1	1.0	0.52	0.001	CLTR_V1	1.0	0.66	-0.002
NRP	1.0	0.05	0.008	NRP	7.9	0.04	0.037	NRP	1.0	0.02	0.011
SMRP_V1	2.4	0.22	0.011	SMRP_V1	1.0	0.04	0.012	SMRP_V1	1.0	0.47	-0.001
MNRP_V1	2.7	0.38	0.008	MNRP_V1	1.0	0.26	0.003	MNRP_V1	1.0	0.25	0.001
MDP_V1	1.0	0.71	-0.002	MDP_V1	1.0	0.06	0.010	MDP_V1	1.0	0.87	-0.003
NDP_V1	2.4	0.51	0.005	NDP_V1	1.0	0.79	0.000	NDP_V1	1.0	0.29	0.000
NaDP_V1	1.0	0.46	-0.001	NaDP_V1	2.3	0.09	0.017	NaDP_V1	1.3	0.77	-0.001
MIP_V1	1.0	0.77	-0.003	MIP_V1	1.0	0.69	0.000	MIP_V1	1.0	0.33	0.000
MIPP_V1	1.0	1.00	-0.003	MIPP_V1	1.0	0.87	0.000	MIPP_V1	4.3	0.18	0.017
NIP_V1	1.0	0.60	-0.002	NIP_V1	1.9	0.13	0.012	NIP_V1	1.0	0.52	-0.002
NaTLP	1.0	0.72	-0.002	NaTLP	1.0	0.16	0.005	NaTLP	2.0	0.18	0.009
MTLP_V1	1.0	0.81	-0.003	MTLP_V1	1.0	0.68	-0.002	MTLP_V1	1.0	0.96	-0.003
STLP_V1	2.7	0.56	0.004	STLP_V1	7.8	0.10	0.028	STLP_V1	1.0	0.41	-0.001
ULTLP_V1	1.0	0.15	0.003	ULTLP_V1	3.0	0.35	0.009	ULTLP_V1	1.0	0.18	0.002
SMLP_V1	1.0	0.53	-0.002	SMLP_V1	6.3	0.40	0.014	SMLP_V1	1.5	0.67	0.000
MNLP_V1	1.0	0.32	0.000	MNLP_V1	1.0	0.44	0.001	MNLP_V1	1.0	0.96	-0.003
EFSRP_V1	1.0	0.06	0.007	EFSRP_V1	1.0	0.39	0.002	EFSRP_V1	6.3	0.31	0.014
EFSRL_V1	1.0	0.06	0.007	EFSRL_V1	3.4	0.35	0.009	EFSRL_V1	1.0	0.89	-0.003
ECSRP_V1	4.3	0.20	0.016	ECSRP_V1	1.0	0.52	0.001	ECSRP_V1	3.1	0.44	0.007
ETSRP_V1	1.0	0.12	0.004	ETSRP_V1	8.0	0.23	0.020	ETSRP_V1	1.0	0.27	0.001
ET SRL_V1	1.0	0.35	0.000	ET SRL_V1	2.5	0.48	0.005	ET SRL_V1	2.8	0.65	0.002
ESSRP_V1	1.0	0.40	-0.001	ESSRP_V1	7.3	0.08	0.029	ESSRP_V1	1.0	0.77	-0.002
ESSRL_V1	1.0	0.12	0.004	ESSRL_V1	1.0	0.14	0.006	ESSRL_V1	1.0	0.69	-0.002
MTP_V1.1	1.0	0.76	-0.003	MTP_V1.1	2.0	0.48	0.004	MTP_V1.1	1.0	0.42	-0.001
OES_V1	1.0	0.82	-0.003	OES_V1	1.5	0.74	-0.001	OES_V1	2.8	0.62	0.003
OEAA_V1	1.7	0.27	0.006	OEAA_V1	1.0	0.50	-0.002	OEAA_V1	1.0	0.12	0.004
MSP_V9	1.0	0.53	-0.002	MSP_V9	1.0	0.11	0.007	MSP_V9	7.0	0.05	0.031
CNSP_V14	1.0	0.08	0.006	CNSP_V14	1.4	0.11	0.009	CNSP_V14	6.1	0.27	0.016
CLP_V6	1.0	0.17	0.003	CLP_V6	1.0	0.86	0.000	CLP_V6	1.0	0.51	-0.002
JJLP_V7	1.0	0.19	0.002	JJLP_V7	1.0	0.49	-0.001	JJLP_V7	1.0	0.89	-0.003
JAS_V1	1.0	0.46	-0.001	JAS_V1	1.5	0.54	0.001	JAS_V1	2.8	0.53	0.004
JMSR_V1	1.0	0.58	-0.002	JMSR_V1	2.6	0.49	0.005	JMSR_V1	1.7	0.11	0.010
EOPP_V1	1.1	0.51	-0.001	EOPP_V1	7.5	0.45	0.010	EOPP_V1	1.0	0.15	0.003
NsTP_V1	1.0	0.50	-0.002	NsTP_V1	2.4	0.13	0.015	NsTP_V1	1.0	0.02	0.011
EOEHR_V1	1.4	0.77	-0.001	EOEHR_V1	3.1	0.20	0.014	EOEHR_V1	1.0	0.41	-0.001
EOTPP_V1	1.8	0.41	0.004	EOTPP_V1	2.2	0.20	0.011	EOTPP_V1	1.2	0.07	0.008
FTR_V1	1.0	0.90	-0.003	FTR_V1	1.0	0.92	0.000	FTR_V1	1.9	0.04	0.018
FEAS_V3	1.0	0.76	-0.003	FEAS_V3	1.0	0.86	-0.003	FEAS_V3	1.8	0.29	0.005
FTAS_V3	1.9	0.34	0.005	FTAS_V3	1.0	0.27	0.003	FTAS_V3	1.0	0.56	-0.002
FJAS_V3	1.0	0.76	-0.003	FJAS_V3	7.1	0.07	0.031	FJAS_V3	1.0	0.41	-0.001
FTLR_V3	2.0	0.03	0.022	FTLR_V3	1.9	0.19	0.009	FTLR_V3	1.8	0.03	0.019
CTLR_V3	1.0	0.00	0.023	CTLR_V3	1.0	0.15	0.006	CTLR_V3	1.0	0.52	-0.002
FPLP_V3	1.0	0.65	-0.002	FPLP_V3	7.1	0.15	0.023	FPLP_V3	1.1	0.00	0.030
PHP_V4	1.0	0.08	0.006	PHP_V4	1.0	0.26	0.001	PHP_V4	1.1	0.22	0.002
FWLP_V3	1.2	0.85	-0.002	FWLP_V3	1.0	0.47	-0.001	FWLP_V3	3.8	0.27	0.012

Cow Conception Rate

	DF Spline	Spline P-Value	R^2 Gain Training
Z1_front	1.0	0.23	0.001
Z2	1.0	0.17	0.001
Z3_front	4.7	0.33	0.007
Z4	1.0	0.29	0.000
Z7_length	1.0	0.22	0.001
Z9_poly	1.0	0.19	0.001
Z10_poly	1.0	0.17	0.001
Z11_poly	1.3	0.64	0.000
Z11_linear	1.0	0.65	-0.001
NFP_LF	1.0	0.64	-0.001
NFPP_LF	6.2	0.45	0.006
NPA	1.0	0.09	0.003
ULRP_V1	6.4	0.55	0.005
MTP_V1	1.0	0.39	0.000
CTP_V1	1.0	0.02	0.007
CLTR_V1	1.0	0.15	0.002
NRP	1.0	0.15	0.002
SMRP_V1	1.0	0.69	-0.001
MNRP_V1	1.5	0.59	0.000
MDP_V1	1.0	0.41	0.000
NDP_V1	5.9	0.45	0.005
NaDP_V1	1.0	0.10	0.002
MIP_V1	3.0	0.11	0.009
MIPP_V1	1.0	0.90	-0.001
NIP_V1	1.0	0.14	0.002
NaTLP	1.0	0.53	-0.001
MTLP_V1	2.4	0.26	0.005
STLP_V1	6.2	0.09	0.014
ULTLP_V1	1.7	0.27	0.003
SMLP_V1	2.1	0.36	0.003
MNLP_V1	6.4	0.41	0.007
EFSRP_V1	1.0	0.70	-0.001
EFSRL_V1	1.0	0.99	-0.001
ECSR_V1	1.0	0.15	0.002
ETSR_V1	8.1	0.19	0.011
ETSRL_V1	1.1	0.86	-0.001
ESSRP_V1	1.0	0.25	0.000
ESSRL_V1	1.0	0.06	0.004
MTP_V1.1	1.0	0.16	0.001
OES_V1	1.0	0.65	-0.001
OEAA_V1	1.0	0.50	-0.001
MSP_V9	1.0	0.37	0.000
CNSP_V14	1.0	0.08	0.003
CLP_V6	1.0	0.20	0.001
JJLP_V7	7.5	0.25	0.009
JAS_V1	7.6	0.07	0.016
JMSR_V1	1.0	0.21	0.001
EOPP_V1	1.0	0.96	-0.001
NsTP_V1	1.0	0.87	-0.001
EOEHR_V1	6.2	0.07	0.015
EOTPP_V1	1.0	0.93	-0.001
FTR_V1	1.0	0.42	-0.001
FEAS_V3	1.0	0.93	-0.001
FTAS_V3	3.2	0.34	0.005
FJAS_V3	1.0	0.26	0.000
FTLR_V3	1.0	0.53	-0.001
CTLR_V3	1.6	0.58	0.000
FPLP_V3	1.0	0.55	-0.001
PHP_V4	2.4	0.04	0.012
FWLP_V3	1.0	0.59	-0.001

Calving Ease

	DF Spline	Spline P-Value	R^2 Gain Training
Z1_front	1.0	0.99	-0.002
Z2	1.0	0.21	0.001
Z3_front	1.0	0.12	0.002
Z4	1.9	0.46	0.002
Z7_length	1.0	0.35	0.000
Z9_poly	1.0	0.98	-0.002
Z10_poly	1.0	0.00	0.011
Z11_poly	1.0	0.68	-0.001
Z11_linear	2.3	0.22	0.005
NFP_LF	1.0	0.25	0.000
NFPP_LF	1.0	0.25	0.001
NPA	1.0	0.15	0.002
ULRP_V1	2.1	0.54	0.002
MTP_V1	1.4	0.18	0.003
CTP_V1	1.0	0.34	0.000
CLTR_V1	1.0	0.32	0.000
NRP	1.1	0.67	-0.001
SMRP_V1	1.0	0.83	-0.001
MNRP_V1	1.0	0.60	-0.001
MDP_V1	1.0	0.66	-0.001
NDP_V1	1.0	0.85	-0.001
NaDP_V1	1.0	0.00	0.012
MIP_V1	1.7	0.46	0.002
MIPP_V1	1.0	0.80	-0.001
NIP_V1	1.0	0.62	-0.001
NaTLP	1.1	0.71	-0.001
MTLP_V1	1.0	0.59	-0.001
STLP_V1	1.0	0.41	0.000
ULTLP_V1	1.0	0.43	-0.001
SMLP_V1	1.0	0.52	-0.001
MNLP_V1	3.1	0.43	0.004
EFSRP_V1	2.9	0.41	0.004
EFSRL_V1	2.4	0.56	0.001
ECSR_V1	1.0	0.34	0.000
ETSR_V1	3.7	0.05	0.014
ETSRL_V1	1.0	0.20	0.001
ESSRP_V1	1.4	0.17	0.004
ESSRL_V1	1.4	0.30	0.002
MTP_V1.1	7.7	0.38	0.008
OES_V1	2.7	0.35	0.004
OEAA_V1	1.0	0.08	0.003
MSP_V9	1.4	0.52	0.000
CNSP_V14	1.0	0.02	0.007
CLP_V6	3.7	0.28	0.007
JJLP_V7	2.5	0.23	0.005
JAS_V1	1.0	0.69	-0.001
JMSR_V1	1.0	0.05	0.005
EOPP_V1	3.2	0.17	0.008
NsTP_V1	3.1	0.08	0.011
EOEHR_V1	1.0	0.03	0.006
EOTPP_V1	2.7	0.40	0.003
FTR_V1	1.0	0.42	-0.001
FEAS_V3	1.0	0.00	0.020
FTAS_V3	1.2	0.01	0.012
FJAS_V3	1.0	0.05	0.005
FTLR_V3	1.0	0.61	-0.001
CTLR_V3	1.0	0.30	0.000
FPLP_V3	1.0	0.89	-0.001
PHP_V4	1.0	0.89	-0.001
FWLP_V3	1.2	0.00	0.017

Cow Total Productivity Index

	DF Spline	Spline P-Value	R^2 Gain Training
Z1_front	1.1	0.83	-0.001
Z2	1.0	0.69	0.000
Z3_front	1.0	0.08	0.002
Z4	1.0	0.60	0.000
Z7_length	1.0	0.79	0.000
Z9_poly	1.0	0.45	0.000
Z10_poly	1.0	0.25	0.001
Z11_poly	1.0	0.96	-0.001
Z11_linear	1.0	0.76	-0.001
NFP_LF	1.0	0.08	0.002
NFPP_LF	7.7	0.06	0.010
NPA	1.2	0.02	0.005
ULRP_V1	6.5	0.39	0.004
MTP_V1	1.0	0.09	0.002
CTP_V1	1.7	0.11	0.003
CLTR_V1	1.0	0.60	0.000
NRP	1.0	0.07	0.002
SMRP_V1	1.0	0.60	0.000
MNRP_V1	3.1	0.23	0.004
MDP_V1	1.0	0.72	0.000
NDP_V1	1.0	0.73	0.000
NaDP_V1	1.3	0.21	0.001
MIP_V1	1.0	0.11	0.002
MIPP_V1	1.0	0.77	-0.001
NIP_V1	1.0	0.03	0.004
NaTLP	1.0	0.24	0.001
MTLP_V1	1.4	0.70	0.000
STLP_V1	1.7	0.50	0.001
ULTLP_V1	1.7	0.50	0.001
SMLP_V1	4.9	0.13	0.006
MNLP_V1	6.4	0.23	0.006
EFSRP_V1	1.4	0.35	0.001
EFSRL_V1	2.6	0.35	0.002
ECSR_V1	1.0	0.29	0.001
ETSR_V1	1.0	0.36	0.000
ETSRL_V1	1.4	0.55	0.000
ESSRP_V1	1.2	0.64	0.000
ESSRL_V1	1.9	0.17	0.003
MTP_V1.1	5.7	0.41	0.003
OES_V1	1.0	0.90	0.000
OEAA_V1	1.0	0.41	0.000
MSP_V9	1.0	0.24	0.001
CNSP_V14	1.0	0.40	0.000
CLP_V6	1.7	0.41	0.001
JJLP_V7	7.3	0.30	0.004
JAS_V1	7.3	0.48	0.003
JMSR_V1	1.0	0.07	0.002
EOPP_V1	5.4	0.46	0.003
NsTP_V1	3.2	0.61	0.001
EOEHR_V1	1.0	0.84	0.000
EOTPP_V1	1.2	0.59	0.000
FTR_V1	1.0	0.89	0.000
FEAS_V3	1.1	0.38	0.000
FTAS_V3	1.0	0.81	0.000
FJAS_V3	1.0	0.87	0.000
FTLR_V3	1.0	0.45	0.000
CTLR_V3	2.1	0.42	0.001
FPLP_V3	3.4	0.41	0.003
PHP_V4	1.9	0.22	0.002
FWLP_V3	1.0	0.34	0.000

Feed Efficiency				Fertility Index				Heifer Conception Rate			
	DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training
Z1_front	1.0	0.68	0.000	Z1_front	1.7	0.34	0.002	Z1_front	2.0	0.22	0.005
Z2	1.0	0.79	-0.001	Z2	1.1	0.29	0.000	Z2	1.0	0.46	-0.001
Z3_front	1.0	0.01	0.006	Z3_front	4.1	0.60	0.003	Z3_front	1.0	0.99	0.000
Z4	1.0	0.14	0.002	Z4	1.0	0.38	0.000	Z4	1.0	0.47	0.001
Z7_length	1.0	0.72	0.000	Z7_length	1.3	0.47	0.000	Z7_length	5.0	0.14	0.012
Z9_poly	1.0	0.74	0.000	Z9_poly	1.0	0.11	0.002	Z9_poly	1.0	0.12	0.004
Z10_poly	5.9	0.34	0.005	Z10_poly	1.0	0.16	0.001	Z10_poly	1.5	0.36	0.002
Z11_poly	1.0	0.93	-0.001	Z11_poly	1.6	0.58	0.000	Z11_poly	2.1	0.27	0.005
Z11_linear	1.0	0.89	-0.001	Z11_linear	2.6	0.72	0.001	Z11_linear	1.0	0.93	-0.002
NFP_LF	3.8	0.22	0.005	NFP_LF	1.0	0.72	-0.001	NFP_LF	1.0	0.99	0.000
NFPP_LF	1.0	0.63	-0.001	NFPP_LF	6.7	0.44	0.005	NFPP_LF	1.0	0.28	0.002
NPA	1.0	0.02	0.004	NPA	1.0	0.09	0.003	NPA	1.0	0.05	0.005
ULRP_V1	1.0	0.46	0.000	ULRP_V1	6.4	0.32	0.008	ULRP_V1	5.9	0.25	0.010
MTP_V1	1.0	0.04	0.004	MTP_V1	1.0	0.42	0.000	MTP_V1	8.1	0.28	0.010
CTP_V1	1.4	0.58	0.000	CTP_V1	1.0	0.02	0.006	CTP_V1	1.0	0.03	0.008
CLTR_V1	1.0	0.55	-0.001	CLTR_V1	1.0	0.18	0.001	CLTR_V1	1.0	0.47	-0.001
NRP	5.0	0.02	0.012	NRP	1.0	0.16	0.001	NRP	1.0	0.82	0.000
SMRP_V1	1.0	0.16	0.002	SMRP_V1	1.7	0.68	0.000	SMRP_V1	1.0	0.14	0.002
MNRP_V1	2.5	0.36	0.002	MNRP_V1	1.5	0.43	0.001	MNRP_V1	3.4	0.23	0.007
MDP_V1	1.8	0.77	0.000	MDP_V1	1.0	0.31	0.000	MDP_V1	1.0	0.23	0.001
NDP_V1	1.0	0.49	0.000	NDP_V1	6.1	0.34	0.007	NDP_V1	5.9	0.28	0.009
NaDP_V1	1.5	0.27	0.002	NaDP_V1	1.0	0.23	0.001	NaDP_V1	1.0	0.15	0.003
MIP_V1	1.0	0.43	0.000	MIP_V1	3.3	0.10	0.010	MIP_V1	2.4	0.15	0.008
MIPP_V1	3.9	0.31	0.004	MIPP_V1	1.0	1.00	-0.001	MIPP_V1	2.4	0.11	0.009
NIP_V1	1.0	0.30	0.000	NIP_V1	1.0	0.13	0.002	NIP_V1	1.0	0.11	0.002
NaTLP	1.0	0.16	0.002	NaTLP	1.0	0.25	0.000	NaTLP	1.0	0.30	0.002
MTLP_V1	3.8	0.60	0.002	MTLP_V1	2.4	0.27	0.005	MTLP_V1	2.4	0.08	0.011
STLP_V1	5.8	0.15	0.007	STLP_V1	2.2	0.11	0.007	STLP_V1	2.6	0.00	0.022
ULTLP_V1	1.0	0.83	0.000	ULTLP_V1	2.1	0.11	0.007	ULTLP_V1	2.5	0.08	0.011
SMLP_V1	5.5	0.01	0.015	SMLP_V1	2.1	0.28	0.004	SMLP_V1	3.7	0.04	0.016
MNLP_V1	2.5	0.68	0.000	MNLP_V1	6.0	0.59	0.004	MNLP_V1	1.0	0.31	0.002
EFSRP_V1	1.0	0.36	0.000	EFSRP_V1	1.0	0.58	-0.001	EFSRP_V1	1.0	0.29	0.002
EFSRL_V1	2.7	0.44	0.002	EFSRL_V1	1.0	0.85	-0.001	EFSRL_V1	1.0	0.58	0.000
ECSR_V1	2.3	0.37	0.002	ECSR_V1	1.0	0.10	0.002	ECSR_V1	1.0	0.04	0.007
ETSRP_V1	4.8	0.46	0.003	ETSRP_V1	8.4	0.06	0.016	ETSRP_V1	8.5	0.01	0.027
ETSRL_V1	1.0	0.41	0.000	ETSRL_V1	2.5	0.58	0.001	ETSRL_V1	1.6	0.41	0.002
ESSRP_V1	1.0	0.70	-0.001	ESSRP_V1	1.4	0.33	0.002	ESSRP_V1	2.4	0.10	0.009
ESSRL_V1	1.7	0.48	0.001	ESSRL_V1	1.0	0.04	0.005	ESSRL_V1	2.1	0.09	0.009
MTP_V1.1	7.0	0.07	0.011	MTP_V1.1	1.0	0.21	0.001	MTP_V1.1	1.7	0.31	0.003
OES_V1	1.0	0.97	-0.001	OES_V1	1.0	0.94	-0.001	OES_V1	1.0	0.85	0.000
OEAA_V1	1.0	0.10	0.002	OEAA_V1	1.0	0.29	0.000	OEAA_V1	1.0	0.36	0.001
MSP_V9	1.0	0.22	0.000	MSP_V9	1.0	0.16	0.001	MSP_V9	1.0	0.21	0.002
CNSP_V14	2.8	0.49	0.002	CNSP_V14	1.0	0.08	0.003	CNSP_V14	1.0	0.06	0.004
CLP_V6	1.9	0.19	0.003	CLP_V6	1.0	0.22	0.001	CLP_V6	1.0	0.24	0.002
JJLP_V7	1.0	0.75	-0.001	JJLP_V7	6.8	0.53	0.004	JJLP_V7	2.9	0.18	0.007
JAS_V1	1.0	0.99	0.000	JAS_V1	7.3	0.10	0.014	JAS_V1	1.7	0.57	0.001
JMSR_V1	1.6	0.17	0.003	JMSR_V1	1.0	0.27	0.000	JMSR_V1	1.0	0.31	0.002
EOPP_V1	5.0	0.31	0.005	EOPP_V1	1.0	0.78	-0.001	EOPP_V1	1.0	0.47	0.001
NsTP_V1	3.5	0.56	0.002	NsTP_V1	1.0	0.66	-0.001	NsTP_V1	1.0	0.93	0.000
EOEHR_V1	1.0	0.97	-0.001	EOEHR_V1	6.2	0.12	0.013	EOEHR_V1	1.2	0.76	-0.001
EOTPP_V1	1.4	0.74	0.000	EOTPP_V1	1.0	0.69	-0.001	EOTPP_V1	1.0	0.96	0.000
FTR_V1	1.0	0.93	0.000	FTR_V1	1.0	0.38	0.000	FTR_V1	1.0	0.27	0.000
FEAS_V3	1.5	0.20	0.002	FEAS_V3	1.0	0.69	-0.001	FEAS_V3	1.0	0.38	0.001
FTAS_V3	1.0	0.95	0.000	FTAS_V3	4.0	0.18	0.009	FTAS_V3	3.4	0.20	0.009
FJAS_V3	1.0	0.94	0.000	FJAS_V3	1.0	0.31	0.000	FJAS_V3	1.0	0.75	0.000
FTLR_V3	1.0	0.51	0.000	FTLR_V3	1.0	0.47	-0.001	FTLR_V3	5.6	0.30	0.008
CTLR_V3	1.0	0.57	-0.001	CTLR_V3	1.3	0.54	0.000	CTLR_V3	1.8	0.44	0.002
FPLP_V3	2.7	0.49	0.001	FPLP_V3	1.0	0.66	-0.001	FPLP_V3	1.0	0.13	0.004
PHP_V4	1.0	0.17	0.002	PHP_V4	2.4	0.08	0.010	PHP_V4	2.2	0.16	0.007
FWLP_V3	1.0	0.06	0.003	FWLP_V3	1.0	0.25	0.000	FWLP_V3	1.0	0.56	0.001

Net Merit				Productive Life				Still Birth			
	DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training
Z1_front	1.0	0.70	0.000	Z1_front	1.0	0.43	0.000	Z1_front	1.0	0.17	0.001
Z2	1.0	0.84	-0.001	Z2	1.5	0.62	0.000	Z2	1.2	0.03	0.009
Z3_front	1.0	0.07	0.003	Z3_front	1.0	0.19	0.001	Z3_front	1.0	0.28	0.000
Z4	1.0	0.48	0.000	Z4	1.0	0.53	-0.001	Z4	1.2	0.89	-0.001
Z7_length	1.0	0.94	-0.001	Z7_length	1.0	0.74	-0.001	Z7_length	1.6	0.04	0.009
Z9_poly	1.0	0.77	-0.001	Z9_poly	6.8	0.31	0.006	Z9_poly	1.0	0.68	-0.001
Z10_poly	1.0	0.38	0.000	Z10_poly	1.0	0.26	0.000	Z10_poly	3.6	0.02	0.018
Z11_poly	1.0	0.84	-0.001	Z11_poly	1.0	0.17	0.001	Z11_poly	1.0	0.34	0.000
Z11_linear	1.0	0.70	-0.001	Z11_linear	2.1	0.35	0.002	Z11_linear	1.1	0.04	0.006
NFP_LF	1.0	0.15	0.002	NFP_LF	1.0	0.58	-0.001	NFP_LF	1.0	0.93	-0.002
NFPF_LF	7.4	0.06	0.013	NFPF_LF	6.7	0.05	0.014	NFPF_LF	1.0	0.47	-0.001
NPA	1.0	0.02	0.005	NPA	1.0	0.11	0.002	NPA	1.7	0.01	0.013
ULRP_V1	6.2	0.64	0.002	ULRP_V1	1.0	0.99	-0.001	ULRP_V1	5.8	0.41	0.007
MTP_V1	1.0	0.20	0.001	MTP_V1	8.4	0.32	0.006	MTP_V1	2.2	0.05	0.011
CTP_V1	1.8	0.25	0.003	CTP_V1	2.3	0.40	0.002	CTP_V1	1.0	0.06	0.004
CLTR_V1	1.0	0.65	-0.001	CLTR_V1	1.6	0.26	0.002	CLTR_V1	2.0	0.52	0.002
NRP	4.1	0.13	0.007	NRP	1.0	0.46	-0.001	NRP	1.1	0.56	-0.001
SMRP_V1	1.0	0.46	0.000	SMRP_V1	1.0	0.91	-0.001	SMRP_V1	1.0	0.23	0.001
MNRP_V1	2.8	0.26	0.004	MNRP_V1	1.3	0.62	0.000	MNRP_V1	1.0	0.69	-0.001
MDP_V1	1.0	0.96	-0.001	MDP_V1	1.0	0.96	-0.001	MDP_V1	1.0	0.01	0.010
NDP_V1	1.0	0.86	-0.001	NDP_V1	4.2	0.60	0.002	NDP_V1	1.0	0.01	0.009
NaDP_V1	1.0	0.26	0.000	NaDP_V1	1.0	0.37	0.000	NaDP_V1	1.4	0.38	0.001
MIP_V1	1.0	0.21	0.001	MIP_V1	1.0	0.53	-0.001	MIP_V1	1.0	0.29	0.000
MIPP_V1	4.9	0.24	0.006	MIPP_V1	1.0	1.00	-0.001	MIPP_V1	1.0	0.45	-0.001
NIP_V1	1.0	0.04	0.003	NIP_V1	1.0	0.02	0.005	NIP_V1	1.0	0.31	0.000
NaTLP	1.0	0.52	0.000	NaTLP	1.0	0.31	0.000	NaTLP	1.2	0.82	-0.001
MTLP_V1	1.0	0.84	-0.001	MTLP_V1	6.9	0.36	0.006	MTLP_V1	1.0	0.04	0.005
STLP_V1	5.6	0.33	0.005	STLP_V1	6.8	0.09	0.011	STLP_V1	1.0	0.04	0.005
ULTLP_V1	1.5	0.71	0.000	ULTLP_V1	1.4	0.74	0.000	ULTLP_V1	1.0	0.16	0.002
SMLP_V1	5.1	0.09	0.010	SMLP_V1	1.0	0.84	-0.001	SMLP_V1	1.0	0.04	0.005
MNLP_V1	5.9	0.27	0.006	MNLP_V1	6.1	0.25	0.007	MNLP_V1	1.0	0.79	-0.001
EFSRP_V1	1.0	0.27	0.000	EFSRP_V1	1.9	0.30	0.002	EFSRP_V1	2.0	0.26	0.004
EFSRL_V1	1.8	0.31	0.002	EFSRL_V1	1.4	0.34	0.001	EFSRL_V1	2.2	0.14	0.007
ECSR_V1	1.0	0.60	-0.001	ECSR_V1	1.0	0.45	0.000	ECSR_V1	1.0	0.58	-0.001
ETSRP_V1	1.0	0.64	0.000	ETSRP_V1	1.3	0.84	-0.001	ETSRP_V1	1.0	0.15	0.002
ETSRL_V1	1.4	0.63	0.000	ETSRL_V1	2.0	0.47	0.002	ETSRL_V1	1.6	0.21	0.004
ESSRP_V1	1.0	0.62	-0.001	ESSRP_V1	1.0	0.86	-0.001	ESSRP_V1	1.0	0.28	0.000
ESSRL_V1	1.9	0.33	0.002	ESSRL_V1	1.0	0.25	0.000	ESSRL_V1	1.0	0.29	0.000
MTP_V1.1	6.0	0.32	0.006	MTP_V1.1	1.0	0.40	0.000	MTP_V1.1	1.0	0.43	-0.001
OES_V1	1.0	0.91	-0.001	OES_V1	1.9	0.49	0.001	OES_V1	1.9	0.30	0.003
OEAA_V1	1.0	0.31	0.001	OEAA_V1	1.0	0.96	-0.001	OEAA_V1	1.0	0.00	0.012
MSP_V9	1.0	0.48	0.000	MSP_V9	1.0	0.54	-0.001	MSP_V9	1.3	0.81	-0.001
CNSP_V14	1.0	0.77	0.000	CNSP_V14	1.0	0.82	-0.001	CNSP_V14	2.0	0.24	0.005
CLP_V6	1.8	0.22	0.003	CLP_V6	1.7	0.57	0.001	CLP_V6	1.0	0.02	0.008
JJLP_V7	7.1	0.35	0.005	JJLP_V7	7.9	0.15	0.010	JJLP_V7	1.4	0.05	0.008
JAS_V1	1.0	0.67	0.000	JAS_V1	1.0	0.55	-0.001	JAS_V1	1.0	0.13	0.002
JMSR_V1	1.0	0.10	0.002	JMSR_V1	1.0	0.37	0.000	JMSR_V1	1.0	0.84	-0.002
EOPP_V1	6.2	0.29	0.006	EOPP_V1	5.7	0.45	0.004	EOPP_V1	1.0	0.95	-0.002
NsTP_V1	1.6	0.72	0.000	NsTP_V1	1.3	0.74	0.000	NsTP_V1	1.0	0.61	-0.001
EOEHR_V1	1.0	0.87	0.000	EOEHR_V1	1.0	0.90	-0.001	EOEHR_V1	1.0	0.56	-0.001
EOTPP_V1	1.6	0.57	0.000	EOTPP_V1	1.1	0.35	0.000	EOTPP_V1	1.5	0.11	0.006
FTR_V1	1.0	0.46	0.000	FTR_V1	1.0	0.47	-0.001	FTR_V1	2.8	0.40	0.004
FEAS_V3	1.9	0.57	0.001	FEAS_V3	4.1	0.21	0.006	FEAS_V3	1.0	0.02	0.007
FTAS_V3	1.0	0.87	0.000	FTAS_V3	6.2	0.29	0.007	FTAS_V3	1.0	0.54	-0.001
FJAS_V3	1.0	0.65	0.000	FJAS_V3	1.0	0.47	-0.001	FJAS_V3	1.3	0.62	0.000
FTLR_V3	1.0	0.70	-0.001	FTLR_V3	1.0	0.65	-0.001	FTLR_V3	1.9	0.34	0.003
CTLR_V3	1.3	0.73	-0.001	CTLR_V3	1.0	0.63	-0.001	CTLR_V3	1.9	0.29	0.004
FPLP_V3	3.0	0.53	0.002	FPLP_V3	1.0	0.80	-0.001	FPLP_V3	1.9	0.48	0.002
PHP_V4	1.7	0.39	0.001	PHP_V4	2.5	0.31	0.003	PHP_V4	1.0	0.35	0.000
FWLP_V3	1.0	0.25	0.001	FWLP_V3	1.0	0.97	-0.001	FWLP_V3	2.6	0.08	0.009

PTA Milk				PTA Fat				PTA Protein			
	DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training
Z1_front	1.0	0.09	0.002	Z1_front	1.0	0.81	-0.001	Z1_front	1.0	0.14	0.001
Z2	1.1	0.08	0.002	Z2	1.0	0.58	-0.001	Z2	1.7	0.43	0.001
Z3_front	1.0	0.94	-0.001	Z3_front	1.0	0.15	0.001	Z3_front	1.0	0.02	0.003
Z4	1.0	0.76	0.000	Z4	1.0	0.64	-0.001	Z4	1.0	0.13	0.001
Z7_length	1.0	0.52	0.000	Z7_length	1.0	0.94	-0.001	Z7_length	1.0	0.54	0.000
Z9_poly	1.0	0.77	0.000	Z9_poly	1.4	0.60	0.000	Z9_poly	1.0	0.42	0.000
Z10_poly	1.0	0.26	0.001	Z10_poly	1.0	0.45	-0.001	Z10_poly	7.6	0.04	0.010
Z11_poly	1.0	0.66	-0.001	Z11_poly	1.0	0.40	0.000	Z11_poly	1.0	0.29	0.000
Z11_linear	1.0	0.23	0.000	Z11_linear	1.0	0.71	-0.001	Z11_linear	7.7	0.73	0.001
NFP_LF	7.5	0.46	0.004	NFP_LF	3.9	0.30	0.005	NFP_LF	1.0	0.03	0.003
NFPP_LF	1.0	0.67	-0.001	NFPP_LF	4.3	0.48	0.003	NFPP_LF	1.0	0.51	0.000
NPA	1.0	0.97	-0.001	NPA	1.0	0.05	0.003	NPA	1.4	0.22	0.001
ULRP_V1	2.9	0.41	0.002	ULRP_V1	1.0	0.62	-0.001	ULRP_V1	1.0	0.38	0.000
MTP_V1	1.0	0.84	0.000	MTP_V1	1.0	0.08	0.002	MTP_V1	1.1	0.18	0.001
CTP_V1	2.7	0.12	0.005	CTP_V1	1.0	0.51	-0.001	CTP_V1	2.2	0.19	0.003
CLTR_V1	5.6	0.62	0.002	CLTR_V1	1.0	0.43	0.000	CLTR_V1	1.0	0.75	-0.001
NRP	2.3	0.10	0.005	NRP	1.0	0.00	0.012	NRP	1.9	0.14	0.003
SMRP_V1	1.0	0.26	0.001	SMRP_V1	1.0	0.20	0.001	SMRP_V1	1.0	0.76	-0.001
MNRP_V1	1.0	0.44	0.000	MNRP_V1	2.1	0.46	0.002	MNRP_V1	1.1	0.09	0.002
MDP_V1	1.0	0.92	-0.001	MDP_V1	1.0	0.84	-0.001	MDP_V1	1.9	0.55	0.001
NDP_V1	1.0	0.47	0.000	NDP_V1	1.0	0.77	-0.001	NDP_V1	1.0	0.11	0.001
NaDP_V1	4.0	0.41	0.003	NaDP_V1	1.2	0.82	-0.001	NaDP_V1	2.6	0.18	0.003
MIP_V1	1.0	0.04	0.004	MIP_V1	1.0	0.53	-0.001	MIP_V1	4.2	0.36	0.003
MIPP_V1	5.6	0.48	0.004	MIPP_V1	4.0	0.18	0.007	MIPP_V1	1.0	0.87	-0.001
NIP_V1	1.3	0.75	0.000	NIP_V1	1.0	0.24	0.000	NIP_V1	1.0	0.54	0.000
NaTLP	2.4	0.61	0.001	NaTLP	1.0	0.30	0.000	NaTLP	1.0	0.30	0.000
MTLP_V1	1.0	0.98	0.000	MTLP_V1	4.5	0.45	0.004	MTLP_V1	1.0	0.36	0.000
STLP_V1	1.0	0.98	-0.001	STLP_V1	6.3	0.06	0.014	STLP_V1	1.0	0.52	0.000
ULTLP_V1	1.0	0.94	0.000	ULTLP_V1	3.0	0.56	0.002	ULTLP_V1	1.0	0.89	-0.001
SMLP_V1	1.0	0.75	0.000	SMLP_V1	5.5	0.01	0.021	SMLP_V1	1.0	0.51	0.000
MNLP_V1	1.0	0.79	0.000	MNLP_V1	1.0	0.99	-0.001	MNLP_V1	1.0	0.85	-0.001
EFSRP_V1	1.9	0.36	0.002	EFSRP_V1	1.0	0.98	-0.001	EFSRP_V1	1.0	0.09	0.001
EFSRL_V1	5.0	0.24	0.006	EFSRL_V1	3.7	0.41	0.004	EFSRL_V1	1.8	0.25	0.002
ECSR_V1	1.0	0.87	0.000	ECSR_V1	2.2	0.29	0.004	ECSR_V1	1.0	0.20	0.000
ETSRP_V1	1.0	0.37	0.001	ETSRP_V1	7.0	0.29	0.007	ETSRP_V1	3.8	0.51	0.002
ETSRL_V1	5.7	0.17	0.007	ETSRL_V1	1.0	0.79	-0.001	ETSRL_V1	5.6	0.47	0.003
ESSRP_V1	1.0	0.76	-0.001	ESSRP_V1	4.0	0.50	0.003	ESSRP_V1	1.0	0.59	-0.001
ESSRL_V1	6.6	0.11	0.008	ESSRL_V1	3.7	0.30	0.005	ESSRL_V1	1.5	0.32	0.001
MTP_V1.1	1.0	0.84	-0.001	MTP_V1.1	7.0	0.13	0.011	MTP_V1.1	1.8	0.37	0.001
OES_V1	1.1	0.87	-0.001	OES_V1	1.0	0.60	-0.001	OES_V1	1.0	0.72	-0.001
OEAA_V1	1.0	0.53	0.000	OEAA_V1	7.1	0.06	0.014	OEAA_V1	1.0	0.44	0.000
MSP_V9	5.3	0.17	0.007	MSP_V9	1.0	0.22	0.001	MSP_V9	1.0	0.49	0.000
CNSP_V14	3.1	0.34	0.003	CNSP_V14	1.0	0.74	-0.001	CNSP_V14	3.2	0.29	0.003
CLP_V6	3.6	0.06	0.008	CLP_V6	1.8	0.17	0.004	CLP_V6	2.4	0.22	0.003
JJLP_V7	1.0	0.11	0.002	JJLP_V7	1.0	0.90	-0.001	JJLP_V7	1.0	0.76	-0.001
JAS_V1	1.0	0.65	0.000	JAS_V1	1.0	0.91	-0.001	JAS_V1	1.0	0.76	-0.001
JMSR_V1	1.0	0.70	0.000	JMSR_V1	1.0	0.32	0.000	JMSR_V1	1.0	0.09	0.001
EOPP_V1	1.0	0.66	0.000	EOPP_V1	1.0	0.60	-0.001	EOPP_V1	4.7	0.35	0.004
NsTP_V1	1.2	0.82	-0.001	NsTP_V1	4.4	0.13	0.009	NsTP_V1	1.4	0.31	0.001
EOEHR_V1	1.0	0.49	0.000	EOEHR_V1	4.3	0.63	0.003	EOEHR_V1	1.0	0.72	-0.001
EOTPP_V1	1.0	0.42	0.000	EOTPP_V1	1.8	0.57	0.001	EOTPP_V1	1.0	0.17	0.001
FTR_V1	2.6	0.36	0.002	FTR_V1	1.0	0.35	0.000	FTR_V1	1.0	0.80	-0.001
FEAS_V3	1.0	0.27	0.001	FEAS_V3	2.3	0.17	0.005	FEAS_V3	1.0	0.45	0.000
FTAS_V3	1.0	0.48	0.000	FTAS_V3	1.0	0.83	-0.001	FTAS_V3	1.0	0.96	-0.001
FJAS_V3	1.0	0.37	0.001	FJAS_V3	1.0	0.94	-0.001	FJAS_V3	1.0	0.91	-0.001
FTLR_V3	2.7	0.52	0.001	FTLR_V3	1.0	0.78	-0.001	FTLR_V3	1.0	0.18	0.001
CTLR_V3	1.0	0.50	0.000	CTLR_V3	1.0	1.00	-0.001	CTLR_V3	1.0	0.25	0.000
FPLP_V3	5.0	0.38	0.004	FPLP_V3	2.7	0.59	0.001	FPLP_V3	3.2	0.37	0.003
PHP_V4	1.0	0.08	0.002	PHP_V4	1.0	0.61	-0.001	PHP_V4	1.0	0.02	0.003
FWLP_V3	1.3	0.63	0.000	FWLP_V3	1.0	0.05	0.003	FWLP_V3	1.0	0.46	0.000

Somatic Cell Score				Displaced Abomasum				Ketosis			
	DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training
Z1_front	3.7	0.69	0.002	Z1_front	1.4	0.11	0.008	Z1_front	1.0	0.47	-0.001
Z2	1.4	0.68	0.000	Z2	2.6	0.21	0.010	Z2	1.0	0.72	-0.002
Z3_front	1.0	0.29	0.000	Z3_front	2.0	0.10	0.012	Z3_front	1.0	0.94	-0.003
Z4	1.0	0.03	0.005	Z4	3.0	0.46	0.006	Z4	1.0	0.85	-0.002
Z7_length	1.0	0.08	0.003	Z7_length	1.0	0.85	-0.002	Z7_length	1.0	0.97	-0.002
Z9_poly	2.5	0.00	0.022	Z9_poly	2.5	0.45	0.004	Z9_poly	1.0	0.24	0.001
Z10_poly	1.0	0.72	-0.001	Z10_poly	1.0	0.85	-0.002	Z10_poly	1.0	0.86	-0.003
Z11_poly	1.0	0.39	0.000	Z11_poly	1.0	0.14	0.003	Z11_poly	6.5	0.43	0.011
Z11_linear	1.7	0.47	0.001	Z11_linear	1.0	0.29	0.001	Z11_linear	1.0	0.46	-0.001
NFP_LF	1.0	0.14	0.002	NFP_LF	3.8	0.00	0.038	NFP_LF	1.0	0.83	-0.003
NFPP_LF	1.8	0.20	0.004	NFPP_LF	1.0	0.41	-0.001	NFPP_LF	1.0	0.47	-0.001
NPA	1.0	0.76	-0.001	NPA	1.0	0.43	-0.001	NPA	4.1	0.45	0.008
ULRP_V1	4.8	0.26	0.008	ULRP_V1	1.2	0.01	0.016	ULRP_V1	6.9	0.01	0.043
MTP_V1	1.0	0.26	0.000	MTP_V1	1.0	0.00	0.029	MTP_V1	1.0	0.36	0.000
CTP_V1	1.0	0.37	0.000	CTP_V1	1.0	0.88	-0.002	CTP_V1	1.0	0.34	0.000
CLTR_V1	1.0	0.42	-0.001	CLTR_V1	1.8	0.06	0.013	CLTR_V1	7.0	0.57	0.008
NRP	1.0	0.58	-0.001	NRP	5.3	0.28	0.014	NRP	3.4	0.08	0.021
SMRP_V1	1.0	0.61	-0.001	SMRP_V1	1.0	0.54	-0.002	SMRP_V1	1.0	0.80	-0.002
MNRP_V1	1.0	0.74	-0.001	MNRP_V1	1.0	0.72	-0.002	MNRP_V1	1.0	0.93	-0.002
MDP_V1	1.0	0.52	-0.001	MDP_V1	5.8	0.12	0.021	MDP_V1	1.0	0.66	-0.002
NDP_V1	1.0	0.86	-0.001	NDP_V1	1.0	0.18	0.002	NDP_V1	1.0	0.69	-0.002
NaDP_V1	1.0	0.72	-0.001	NaDP_V1	1.0	0.29	0.001	NaDP_V1	1.0	0.66	-0.002
MIP_V1	1.7	0.56	0.001	MIP_V1	3.3	0.01	0.031	MIP_V1	1.0	0.39	0.000
MIPP_V1	3.5	0.03	0.016	MIPP_V1	1.0	0.54	-0.002	MIPP_V1	2.4	0.02	0.024
NIP_V1	2.3	0.07	0.009	NIP_V1	2.9	0.06	0.019	NIP_V1	1.0	0.15	0.003
NaTLP	2.0	0.24	0.005	NaTLP	6.9	0.48	0.008	NaTLP	7.1	0.14	0.023
MTLP_V1	1.7	0.68	0.000	MTLP_V1	1.0	0.33	0.000	MTLP_V1	1.0	0.24	0.001
STLP_V1	1.7	0.46	0.001	STLP_V1	1.0	0.06	0.006	STLP_V1	1.0	0.04	0.009
ULTLP_V1	1.3	0.64	0.000	ULTLP_V1	1.0	0.02	0.011	ULTLP_V1	1.0	0.02	0.013
SMLP_V1	1.9	0.64	0.001	SMLP_V1	1.0	0.13	0.003	SMLP_V1	1.0	0.08	0.006
MNLP_V1	1.0	0.98	-0.002	MNLP_V1	1.0	0.48	-0.001	MNLP_V1	4.0	0.25	0.014
EFSRP_V1	2.2	0.13	0.008	EFSRP_V1	1.0	0.57	-0.002	EFSRP_V1	1.0	0.37	0.000
EFSRL_V1	1.3	0.74	-0.001	EFSRL_V1	1.0	0.16	0.002	EFSRL_V1	1.0	0.36	0.000
ECSR_V1	1.3	0.67	0.000	ECSR_V1	5.6	0.31	0.013	ECSR_V1	1.0	0.36	0.000
ETSR_V1	1.0	0.48	-0.001	ETSR_V1	1.7	0.04	0.015	ETSR_V1	1.2	0.42	0.000
ETSRL_V1	1.0	0.14	0.002	ETSRL_V1	1.7	0.71	0.000	ETSRL_V1	1.5	0.23	0.006
ESSRP_V1	1.3	0.70	0.000	ESSRP_V1	1.3	0.23	0.004	ESSRP_V1	1.0	0.73	-0.002
ESSRL_V1	1.8	0.22	0.004	ESSRL_V1	2.1	0.05	0.017	ESSRL_V1	3.6	0.23	0.014
MTP_V1.1	1.0	0.66	-0.001	MTP_V1.1	1.7	0.19	0.007	MTP_V1.1	1.6	0.34	0.004
OES_V1	1.1	0.74	-0.001	OES_V1	1.0	0.99	-0.002	OES_V1	1.0	0.13	0.004
OEAA_V1	1.0	0.72	-0.001	OEAA_V1	5.6	0.11	0.022	OEAA_V1	1.0	0.81	-0.002
MSP_V9	1.9	0.36	0.003	MSP_V9	1.0	0.50	-0.001	MSP_V9	1.0	0.68	-0.002
CNSP_V14	1.0	0.75	-0.001	CNSP_V14	1.7	0.44	0.002	CNSP_V14	1.0	0.33	0.000
CLP_V6	1.0	0.86	-0.001	CLP_V6	1.0	0.56	-0.002	CLP_V6	1.5	0.66	0.000
JJLP_V7	2.1	0.28	0.004	JJLP_V7	7.9	0.05	0.028	JJLP_V7	1.0	0.46	-0.001
JAS_V1	1.0	0.29	0.000	JAS_V1	3.1	0.30	0.009	JAS_V1	1.0	0.10	0.005
JMSR_V1	1.0	0.42	-0.001	JMSR_V1	1.0	0.02	0.012	JMSR_V1	1.9	0.35	0.005
EOPP_V1	1.0	0.59	-0.001	EOPP_V1	2.1	0.43	0.004	EOPP_V1	1.0	0.27	0.001
NsTP_V1	1.0	0.82	-0.001	NsTP_V1	4.2	0.11	0.019	NsTP_V1	1.0	0.07	0.006
EOEHR_V1	1.0	0.88	-0.001	EOEHR_V1	1.0	0.87	-0.002	EOEHR_V1	5.2	0.11	0.022
EOTPP_V1	1.0	0.12	0.002	EOTPP_V1	1.0	0.81	-0.002	EOTPP_V1	1.0	0.09	0.005
FTR_V1	1.0	0.13	0.002	FTR_V1	1.0	0.05	0.007	FTR_V1	1.9	0.27	0.007
FEAS_V3	1.0	0.91	-0.001	FEAS_V3	6.7	0.44	0.009	FEAS_V3	1.5	0.80	-0.001
FTAS_V3	5.9	0.45	0.006	FTAS_V3	1.0	0.05	0.007	FTAS_V3	2.0	0.37	0.005
FJAS_V3	4.4	0.60	0.004	FJAS_V3	1.7	0.12	0.009	FJAS_V3	1.0	0.20	0.002
FTLR_V3	1.0	0.11	0.002	FTLR_V3	1.4	0.74	-0.001	FTLR_V3	1.0	0.51	-0.001
CTLR_V3	2.9	0.13	0.009	CTLR_V3	1.0	0.93	-0.002	CTLR_V3	1.0	0.58	-0.002
FPLP_V3	1.0	0.16	0.001	FPLP_V3	2.0	0.32	0.005	FPLP_V3	1.1	0.53	-0.001
PHP_V4	1.0	1.00	-0.002	PHP_V4	4.1	0.50	0.006	PHP_V4	1.5	0.76	-0.001
FWLP_V3	1.8	0.52	0.001	FWLP_V3	5.6	0.30	0.012	FWLP_V3	1.0	0.68	-0.002

Lameness				Mastitis				Metritis			
	DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training		DF Spline	Spline P-Value	R^2 Gain Training
Z1_front	4.4	0.08	0.024	Z1_front	2.2	0.28	0.006	Z1_front	4.3	0.31	0.012
Z2	3.3	0.16	0.016	Z2	1.9	0.54	0.002	Z2	3.5	0.30	0.012
Z3_front	1.0	0.87	-0.001	Z3_front	7.1	0.61	0.003	Z3_front	1.0	0.60	0.000
Z4	1.0	0.57	0.000	Z4	1.0	0.63	-0.002	Z4	4.2	0.17	0.019
Z7_length	1.8	0.70	0.000	Z7_length	1.0	0.35	0.000	Z7_length	3.4	0.19	0.014
Z9_poly	1.0	0.41	0.001	Z9_poly	7.5	0.08	0.023	Z9_poly	4.3	0.02	0.034
Z10_poly	1.0	0.04	0.009	Z10_poly	1.0	0.51	-0.001	Z10_poly	1.0	0.31	0.002
Z11_poly	1.0	0.41	-0.001	Z11_poly	1.8	0.58	0.001	Z11_poly	1.0	0.93	-0.003
Z11_linear	2.9	0.38	0.007	Z11_linear	1.2	0.79	-0.001	Z11_linear	1.5	0.65	0.000
NFP_LF	2.0	0.37	0.005	NFP_LF	1.0	0.28	0.000	NFP_LF	1.0	0.09	0.007
NFPP_LF	1.0	0.15	0.005	NFPP_LF	1.0	0.13	0.003	NFPP_LF	1.0	0.42	-0.001
NPA	1.4	0.59	0.000	NPA	1.0	0.18	0.002	NPA	4.5	0.07	0.026
ULRP_V1	1.0	0.79	-0.001	ULRP_V1	1.0	0.02	0.010	ULRP_V1	1.3	0.04	0.015
MTP_V1	1.0	0.63	0.000	MTP_V1	1.0	0.03	0.009	MTP_V1	1.4	0.11	0.009
CTP_V1	1.0	0.30	0.002	CTP_V1	1.0	0.30	0.000	CTP_V1	1.7	0.22	0.007
CLTR_V1	1.5	0.72	-0.001	CLTR_V1	1.0	0.03	0.008	CLTR_V1	1.5	0.64	0.000
NRP	6.7	0.54	0.009	NRP	1.0	0.09	0.004	NRP	1.5	0.62	0.000
SMRP_V1	1.0	0.02	0.015	SMRP_V1	1.0	0.64	-0.002	SMRP_V1	1.0	0.54	0.000
MNRP_V1	1.0	0.71	-0.002	MNRP_V1	4.6	0.27	0.012	MNRP_V1	3.7	0.32	0.012
MDP_V1	1.0	0.17	0.002	MDP_V1	1.0	0.91	-0.002	MDP_V1	1.0	0.68	-0.001
NDP_V1	1.0	0.52	0.000	NDP_V1	1.6	0.61	0.001	NDP_V1	1.0	0.39	0.001
NaDP_V1	6.6	0.10	0.026	NaDP_V1	1.0	0.93	-0.002	NaDP_V1	1.0	0.45	0.000
MIP_V1	1.0	0.76	-0.001	MIP_V1	1.0	0.87	-0.002	MIP_V1	2.3	0.12	0.015
MIPP_V1	1.0	0.25	0.003	MIPP_V1	1.0	0.61	-0.002	MIPP_V1	1.0	0.18	0.004
NIP_V1	1.0	0.98	-0.001	NIP_V1	1.0	0.31	0.000	NIP_V1	1.0	0.07	0.008
NaTLP	1.0	0.29	0.002	NaTLP	1.0	0.15	0.003	NaTLP	6.6	0.44	0.012
MTLP_V1	1.0	0.85	-0.003	MTLP_V1	1.0	0.40	-0.001	MTLP_V1	1.0	0.46	0.000
STLP_V1	1.4	0.76	-0.001	STLP_V1	1.0	0.43	-0.001	STLP_V1	2.2	0.30	0.008
ULTLP_V1	1.0	0.95	-0.003	ULTLP_V1	1.5	0.67	0.000	ULTLP_V1	2.1	0.17	0.011
SMLP_V1	1.6	0.54	0.001	SMLP_V1	1.3	0.44	0.001	SMLP_V1	1.0	0.24	0.001
MNLP_V1	1.0	0.31	0.002	MNLP_V1	2.8	0.20	0.010	MNLP_V1	4.9	0.23	0.016
EFSRP_V1	1.8	0.59	0.002	EFSRP_V1	1.0	0.94	-0.002	EFSRP_V1	1.0	0.67	-0.001
EFSRL_V1	1.4	0.35	0.003	EFSRL_V1	1.0	0.91	-0.002	EFSRL_V1	2.2	0.42	0.005
ECSR_P_V1	4.5	0.31	0.012	ECSR_P_V1	1.0	0.95	-0.002	ECSR_P_V1	1.0	0.69	-0.001
ETSRP_V1	5.3	0.31	0.013	ETSRP_V1	1.0	0.18	0.002	ETSRP_V1	1.0	0.25	0.002
ETSRL_V1	1.0	0.06	0.008	ETSRL_V1	1.0	0.47	-0.001	ETSRL_V1	4.4	0.30	0.013
ESSRP_V1	5.0	0.34	0.012	ESSRP_V1	1.0	0.13	0.003	ESSRP_V1	1.7	0.55	0.002
ESSRL_V1	4.6	0.37	0.012	ESSRL_V1	1.0	0.27	0.000	ESSRL_V1	6.0	0.33	0.014
MTP_V1.1	1.0	0.27	0.001	MTP_V1.1	1.0	0.18	0.002	MTP_V1.1	6.6	0.20	0.022
OES_V1	1.0	0.34	0.001	OES_V1	1.6	0.68	0.000	OES_V1	1.0	0.66	-0.001
OEAA_V1	1.0	0.69	-0.001	OEAA_V1	1.0	0.11	0.004	OEAA_V1	7.3	0.05	0.033
MSP_V9	1.0	0.15	0.005	MSP_V9	1.7	0.40	0.003	MSP_V9	4.9	0.18	0.018
CNSP_V14	1.0	0.95	-0.001	CNSP_V14	1.2	0.86	-0.002	CNSP_V14	1.6	0.16	0.008
CLP_V6	1.0	0.50	0.000	CLP_V6	1.0	0.32	0.000	CLP_V6	2.3	0.15	0.013
JJLP_V7	7.2	0.13	0.023	JJLP_V7	1.0	0.24	0.001	JJLP_V7	1.9	0.07	0.015
JAS_V1	1.0	0.61	0.000	JAS_V1	1.0	0.46	-0.001	JAS_V1	3.9	0.21	0.015
JMSR_V1	1.0	0.52	0.000	JMSR_V1	1.0	0.07	0.005	JMSR_V1	1.0	0.25	0.003
EOPP_V1	4.5	0.07	0.025	EOPP_V1	2.7	0.04	0.017	EOPP_V1	1.0	0.78	-0.003
NsTP_V1	1.0	0.08	0.007	NsTP_V1	1.0	0.29	0.000	NsTP_V1	3.4	0.15	0.016
EOEHR_V1	5.1	0.43	0.010	EOEHR_V1	1.0	0.26	0.001	EOEHR_V1	1.0	0.14	0.005
EOTPP_V1	1.0	0.16	0.004	EOTPP_V1	1.0	0.23	0.001	EOTPP_V1	1.0	0.97	-0.001
FTR_V1	1.2	0.88	-0.002	FTR_V1	1.0	0.28	0.000	FTR_V1	4.2	0.18	0.018
FEAS_V3	1.0	0.21	0.003	FEAS_V3	1.0	0.45	-0.001	FEAS_V3	2.9	0.33	0.009
FTAS_V3	6.0	0.10	0.025	FTAS_V3	6.5	0.30	0.013	FTAS_V3	1.0	0.16	0.004
FJAS_V3	1.0	0.17	0.004	FJAS_V3	1.0	0.16	0.002	FJAS_V3	1.0	0.99	-0.003
FTLR_V3	1.0	0.69	-0.002	FTLR_V3	3.4	0.51	0.004	FTLR_V3	2.5	0.03	0.024
CTLR_V3	1.0	0.81	-0.001	CTLR_V3	1.0	0.17	0.002	CTLR_V3	2.2	0.41	0.005
FPLP_V3	1.0	0.27	0.001	FPLP_V3	1.0	0.76	-0.002	FPLP_V3	2.4	0.10	0.016
PHP_V4	1.0	0.60	0.000	PHP_V4	5.2	0.11	0.019	PHP_V4	8.2	0.03	0.038
FWLP_V3	1.0	0.23	0.001	FWLP_V3	6.6	0.32	0.011	FWLP_V3	1.9	0.26	0.007

Retained Placenta

	DF Spline	Spline P-Value	R^2 Gain Training
Z1_front	1.0	0.54	0.000
Z2	1.1	0.73	-0.002
Z3_front	1.0	0.07	0.008
Z4	1.0	0.26	0.002
Z7_length	1.0	0.33	0.001
Z9_poly	1.6	0.49	0.002
Z10_poly	2.0	0.04	0.019
Z11_poly	3.2	0.35	0.008
Z11_linear	3.2	0.12	0.018
NFP_LF	1.7	0.21	0.007
NFPP_LF	1.0	0.14	0.005
NPA	2.4	0.12	0.015
ULRP_V1	1.0	0.17	0.004
MTP_V1	1.0	0.10	0.006
CTP_V1	1.4	0.71	-0.001
CLTR_V1	3.3	0.30	0.010
NRP	5.1	0.15	0.020
SMRP_V1	2.4	0.15	0.014
MNRP_V1	1.0	0.04	0.010
MDP_V1	1.0	0.15	0.004
NDP_V1	1.0	0.00	0.027
NaDP_V1	1.0	0.29	0.000
MIP_V1	1.6	0.59	0.001
MIPP_V1	1.0	0.95	-0.001
NIP_V1	1.0	0.26	0.001
NaTLP	1.0	0.92	-0.001
MTLP_V1	1.0	0.98	-0.001
STLP_V1	1.0	0.94	-0.001
ULTLP_V1	1.0	0.86	-0.001
SMLP_V1	1.0	0.88	-0.001
MNLP_V1	1.0	0.85	-0.003
EFSRP_V1	1.0	0.32	0.001
EFSRL_V1	1.0	0.21	0.003
ECSRP_V1	1.7	0.26	0.005
ETSRP_V1	1.0	0.31	0.000
ETSRL_V1	3.6	0.40	0.008
ESSRP_V1	1.0	0.89	-0.003
ESSRL_V1	1.0	0.85	-0.001
MTP_V1.1	6.4	0.29	0.019
OES_V1	1.0	0.71	-0.001
OEAA_V1	1.0	0.35	0.001
MSP_V9	1.0	0.49	0.000
CNSP_V14	1.4	0.75	-0.001
CLP_V6	1.7	0.46	0.003
JJLP_V7	1.0	0.88	-0.001
JAS_V1	1.0	0.21	0.003
JMSR_V1	1.0	0.34	0.001
EOPP_V1	3.1	0.17	0.014
NsTP_V1	1.0	0.29	0.002
EOEHR_V1	6.8	0.54	0.008
EOTPP_V1	1.0	0.29	0.002
FTR_V1	1.7	0.52	0.002
FEAS_V3	1.4	0.72	-0.001
FTAS_V3	1.0	0.39	0.001
FJAS_V3	1.0	0.81	-0.001
FTLR_V3	1.0	0.54	0.000
CTLR_V3	1.0	0.21	0.003
FPLP_V3	1.0	0.69	-0.001
PHP_V4	1.6	0.45	0.003
FWLP_V3	1.4	0.46	0.002

Appendix D: Representative codes for social network analyses in the R programming environment for Chapter 3, as well as supplemental figures

#Data Wrangling

##Getting Complete List of Organilac Cows

```
cowlist <- read.csv('final dataset_MILK YIELD.csv')
names(cowlist)[1] <- "ID"
cowlist <- unique(cowlist$ID)
cowlist <- sort(cowlist[!is.na(cowlist)]) # creates list of all cows potentially present in the organilac herd
length(cowlist)
[1] 203
```

##Reading in the Data:

```
datlist <- read.csv('Milk Order Data/Milking_Infered_Data/CSVList.csv', stringsAsFactors = F, header = F)
orderdata_infered <- list()

for (i in 1:nrow(datlist)){
  milking <- paste('Milk Order Data/Milking_Infered_Data/', datlist[i,1], sep = ")
  temp <- read.csv(milking, stringsAsFactors = F, header = F, blank.lines.skip = F)

  #seperating header and data
  temp2 = temp[1:20,1]
  start <- which("Cow No." == temp2 | "Cow No. " == temp2)
  header <- temp[1:(start-1),]
  dat <- read.csv(milking, stringsAsFactors = F, header = T, skip = (start-1)
  orderdata_infered[[i]] <- list(data = dat, HeaderInfo = header)
}
```

##Pulling out Sheet Attributes:

GroupID <- 34

```
for (i in 1:length(orderdata_infered)){
  temp <- orderdata_infered[[i]]
  headtemp <- temp[[2]]

  timetemp <- gsub(' ', "", headtemp[1,3])
  datestart <- regexpr('Date:', timetemp)[1] + 6
  datestop <- regexpr('Time:', timetemp)[1] - 1
  timestart <- regexpr('Time:', timetemp)[1] + 5
  timestop <- nchar(timetemp)
  #print(substr(timetemp, datestart, datestop))
  #print(substr(timetemp, timestart, timestop))

  orderdata_infered[[i]][['Date']] <- as.Date(substr(timetemp, datestart, datestop), format = "%m/%d/%y",
tz = 'America/Denver')
```

```

if (is.na(orderdata_infered[[i]][['Date']])){
  orderdata_infered[[i]][['Date']] <- as.Date(substr(timetemp, datestart, datestop), format =
"%m.%d.%y", tz = 'America/Denver')
  #print(i)
}
#print(substr(timetemp, datestart, datestop))
#print(orderdata_infered[[i]][['Date']])

timetemp2 <- paste(substr(timetemp, datestart, datestop), substr(timetemp, timestart, timestop))
orderdata_infered[[i]][['Time']] <- strptime(timetemp2, format = "%m/%d/%y %H:%M")
if (is.na(orderdata_infered[[i]][['Time']])){
  orderdata_infered[[i]][['Time']] <- strptime(timetemp2, format = "%m.%d.%y %H:%M")
  #print(i)
}
#print(orderdata_infered[[i]][['Time']])

dattemp <- temp[[1]]
dattemp <- subset(dattemp, Group.No. == 34)
orderdata_infered[[i]][['CowCount']] <- nrow(dattemp)
#print(orderdata_infered[[i]][['CowCount']])

if (sum(as.numeric(dattemp$Avg..Conductivity.Today.2), na.rm = T) == 0){
  orderdata_infered[[i]][['Milking']] <- 1
} else if (sum(as.numeric(dattemp$Avg..Conductivity.Today.3), na.rm = T) == 0){
  orderdata_infered[[i]][['Milking']] <- 2
} else{
  orderdata_infered[[i]][['Milking']] <- 3
}
}

```

##Extraction - Interaction Matrices

```

interactionmatrix <- function(orderdat, cowcol = 1, cowlist, ndown){

  mat <- matrix(data = NA, nrow = length(cowlist), ncol = length(cowlist))
  colnames(mat) <- cowlist
  rownames(mat) <- cowlist

  pattern <- orderdat[,cowcol]

  for (i in 1:length(pattern)){
    currentcow <- pattern[i]

    if(length(which(currentcow == cowlist)) == 0){
      next # if cow is not in organilac list skip to next cow/iteration
    }else{
      cowindex <- which(currentcow == cowlist)
    }
  }
}

```

```

mat[cowindex,] <- 0

subcows <- pattern[(i+1):min((i+ndown), length(pattern))]

for (c in 1:length(subcows)){
  if(length(which(subcows[c] == cowlist)) == 1){
    cowindex2 <- which(subcows[c] == cowlist)
    mat[cowindex, cowindex2] <- (ndown + 1 - c)/ndown
  }
}

}

for (j in 1:nrow(mat)){
  if (is.na(mat[j,j])){
    mat[,j] <- NA
  }
}

return(mat)
}

matchech <- interactionmatrix(orderdata_indexed[[1]]$data, cowcol = 1, cowlist, ndown = 5 )

##Interaction Matrices

datelist_indexed <- c()

for (i in 1:length(orderdata_indexed)){

  orderdata_indexed[[i]][["Interactions_1']] <- interactionmatrix(orderdata_indexed[[i]]$data, cowcol = 1,
cowlist, ndown = 1)
  orderdata_indexed[[i]][["Interactions_3']] <- interactionmatrix(orderdata_indexed[[i]]$data, cowcol = 1,
cowlist, ndown = 3)
  orderdata_indexed[[i]][["Interactions_5']] <- interactionmatrix(orderdata_indexed[[i]]$data, cowcol = 1,
cowlist, ndown = 5)

  datelist_indexed <- c(datelist_indexed, orderdata_indexed[[i]]$Date)
}

for (i in 1:length(orderdata_indexed)){

  if (i == 1){
    vectempl1a <- c(orderdata_indexed[[i]]$Milking, orderdata_indexed[[i]]$CowCount,
as.vector(orderdata_indexed[[i]]$Interactions_1))
    dat.ind.interac1 <- data.frame(vectempl1a)
  }
}

```

```

    vectemp3a <- c(orderdata_indexed[[i]]$Milking, orderdata_indexed[[i]]$CowCount,
as.vector(orderdata_indexed[[i]]$Interactions_3))
    dat.ind.interac3 <- data.frame(vectemp3a)

    vectemp5a <- c(orderdata_indexed[[i]]$Milking, orderdata_indexed[[i]]$CowCount,
as.vector(orderdata_indexed[[i]]$Interactions_5))
    dat.ind.interac5 <- data.frame(vectemp5a)

  }else{

    vectemp1b <- c(orderdata_indexed[[i]]$Milking, orderdata_indexed[[i]]$CowCount,
as.vector(orderdata_indexed[[i]]$Interactions_1))
    dat.ind.interac1 <- cbind(dat.ind.interac1, vectemp1b)

    vectemp3b <- c(orderdata_indexed[[i]]$Milking, orderdata_indexed[[i]]$CowCount,
as.vector(orderdata_indexed[[i]]$Interactions_3))
    dat.ind.interac3 <- cbind(dat.ind.interac3, vectemp3b)

    vectemp5b <- c(orderdata_indexed[[i]]$Milking, orderdata_indexed[[i]]$CowCount,
as.vector(orderdata_indexed[[i]]$Interactions_5))
    dat.ind.interac5 <- cbind(dat.ind.interac5, vectemp5b)

  }
}

colnames(dat.ind.interac1) <- datelist_indexed
colnames(dat.ind.interac3) <- datelist_indexed
colnames(dat.ind.interac5) <- datelist_indexed

dat.ind.interac1 <- dat.ind.interac1[,order(names(dat.ind.interac1))]
dat.ind.interac3 <- dat.ind.interac3[,order(names(dat.ind.interac3))]
dat.ind.interac5 <- dat.ind.interac5[,order(names(dat.ind.interac5))]

rownames(dat.ind.interac1)[1:2] <- c('Milking', 'CowCount')
rownames(dat.ind.interac3)[1:2] <- c('Milking', 'CowCount')
rownames(dat.ind.interac5)[1:2] <- c('Milking', 'CowCount')

##Extraction - Rank and Quantile

#Define Function

makerankvector <- function(orderdat, cowcol = 1, cowlist, quantile = T){

  pattern <- orderdat[,cowcol] # obs cows
  pattern_culled <- pattern[pattern %in% cowlist]

  rankvec <- rep(NA, length(cowlist))
  names(rankvec) <- cowlist

  quantvec <- rep(NA, length(cowlist))

```



```

names(quantvec) <- cowlist

for (i in 1:length(rankvec)){
  if(names(rankvec)[i] %in% pattern_culled){
    rank <- which(pattern_culled == names(rankvec)[i])
    rankvec[i] <- rank
    quantvec[i] <- 1 - (rank - 1)/length(pattern_culled)
  }
}

if (quantile){
  return(quantvec)
}else{
  return(rankvec)
}
}

checkrank <- makerankvector(orderdata_indexed[[1]]$data, cowcol = 1, cowlist, F)
checkquant <- makerankvector(orderdata_indexed[[1]]$data, cowcol = 1, cowlist)

dat.inf.cowquant <- as.data.frame( matrix(data = NA, nrow = length(cowlist), ncol =
length(datelist_infered)))
rownames(dat.inf.cowquant) <- cowlist
colnames(dat.inf.cowquant) <- datelist_infered

for (j in 1:length(datelist_infered)){
  temp <- orderdata_infered[[j]]$QuantileOrder
  for (i in 1:length(cowlist)){
    dat.inf.cowquant[i,j] <- temp[which(names(temp) == cowlist[i])]
  }
}

dat.ind.cowquant <- as.data.frame( matrix(data = NA, nrow = length(cowlist), ncol =
length(datelist_indexed)))
rownames(dat.ind.cowquant) <- cowlist
colnames(dat.ind.cowquant) <- datelist_indexed

for (j in 1:length(datelist_indexed)){
  temp <- orderdata_indexed[[j]]$QuantileOrder
  for (i in 1:length(cowlist)){
    dat.ind.cowquant[i,j] <- temp[which(names(temp) == cowlist[i])]
  }
}

##Data Visualization and Summarization

milkobs_infered <- c()

```

```

for (i in 1:length(orderdata_infered)){
  milkobs_infered[i] <- orderdata_infered[[i]][['Milking']]
}
table(milkobs_infered)
milkobs_infered
  1  2  3
130 16  4

milking = c()
date = c()
cowcount = c()
extracount = c()

for (i in 1:length(orderdata_indexed)){

  milking <- c( milking, orderdata_indexed[[i]][['Milking']] )
  date <- c(date, orderdata_indexed[[i]][['Date']])

  dattemp <- orderdata_indexed[[i]][[1]]
  obscows <- dattemp$Cow.No.
  cowcount <- c(cowcount , length(intersect(obscows, cowlist)))
  orderdata_indexed[[i]][['NumInclude']] <- length(intersect(obscows, cowlist))
  extracount <- c(extracount, length(obscows ) - length(intersect(obscows, cowlist)))
  orderdata_indexed[[i]][['NumExclude']] <- length(obscows ) - length(intersect(obscows, cowlist))

}

for (i in 1:length(orderdata_infered)){

  milking <- c( milking, orderdata_infered[[i]][['Milking']] )
  date <- c(date, orderdata_infered[[i]][['Date']])

  dattemp <- orderdata_infered[[i]][[1]]
  obscows <- dattemp$Cow.No.
  cowcount <- c(cowcount , length(intersect(obscows, cowlist)))
  orderdata_infered[[i]][['NumInclude']] <- length(intersect(obscows, cowlist))
  extracount <- c(extracount, length(obscows ) - length(intersect(obscows, cowlist)))
  orderdata_infered[[i]][['NumExclude']] <- length(obscows ) - length(intersect(obscows, cowlist))

}

library(ggplot2)

qplot(cowcount, geom="histogram", xlab = 'Number of Organilac Cows in Milking', ylab = "", main =
'Distribution of Organilac Cow Inclusion Rate in Milking Observation', col=I("black"), fill=I("blue"),
alpha=I(.5))

qplot(extracount, geom="histogram", xlab = 'Number of Non-Organilac Cows in Milking', ylab = "", main =
'Distribution of Inclusion Rate of Extraneous Cows', col=I("black"), fill=I("blue"), alpha=I(.5))

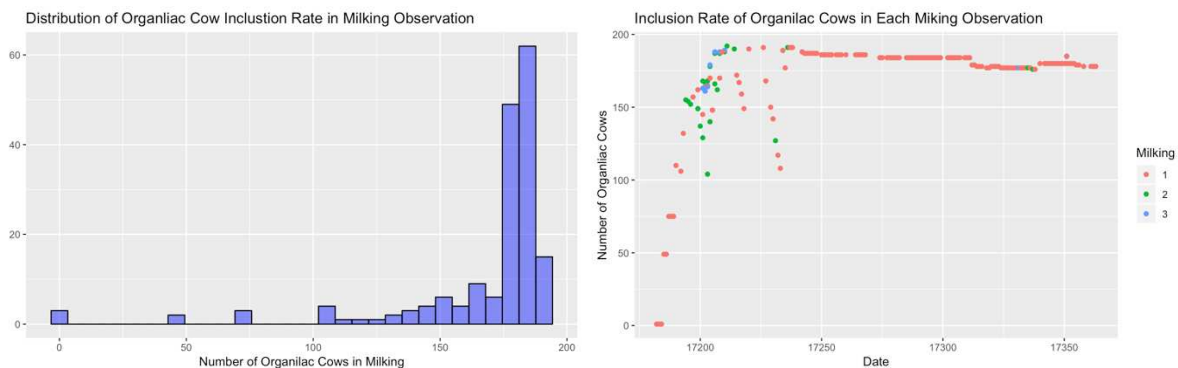
```

```
table(extracount)
sum(extracount>0)
sum(extracount>10)
```

```
print('Number of Milking Obs w/ No Additional Cows')
sum(extracount==0)
```

```
cowcountdata <- data.frame(Milking = as.factor(milking), Date = date, InCows = as.numeric(cowcount),
OutCows = as.numeric(extracount))
```

```
ggplot(cowcountdata, aes(x=Date, y=InCows, color=Milking)) + geom_point() + ggtitle ('Inclusion Rate
of Organilac Cows in Each Miking Observation') + ylab('Number of Organliac Cows')
```



```
#####
#Preliminary Visualizations – Cow Quantile Plots
```

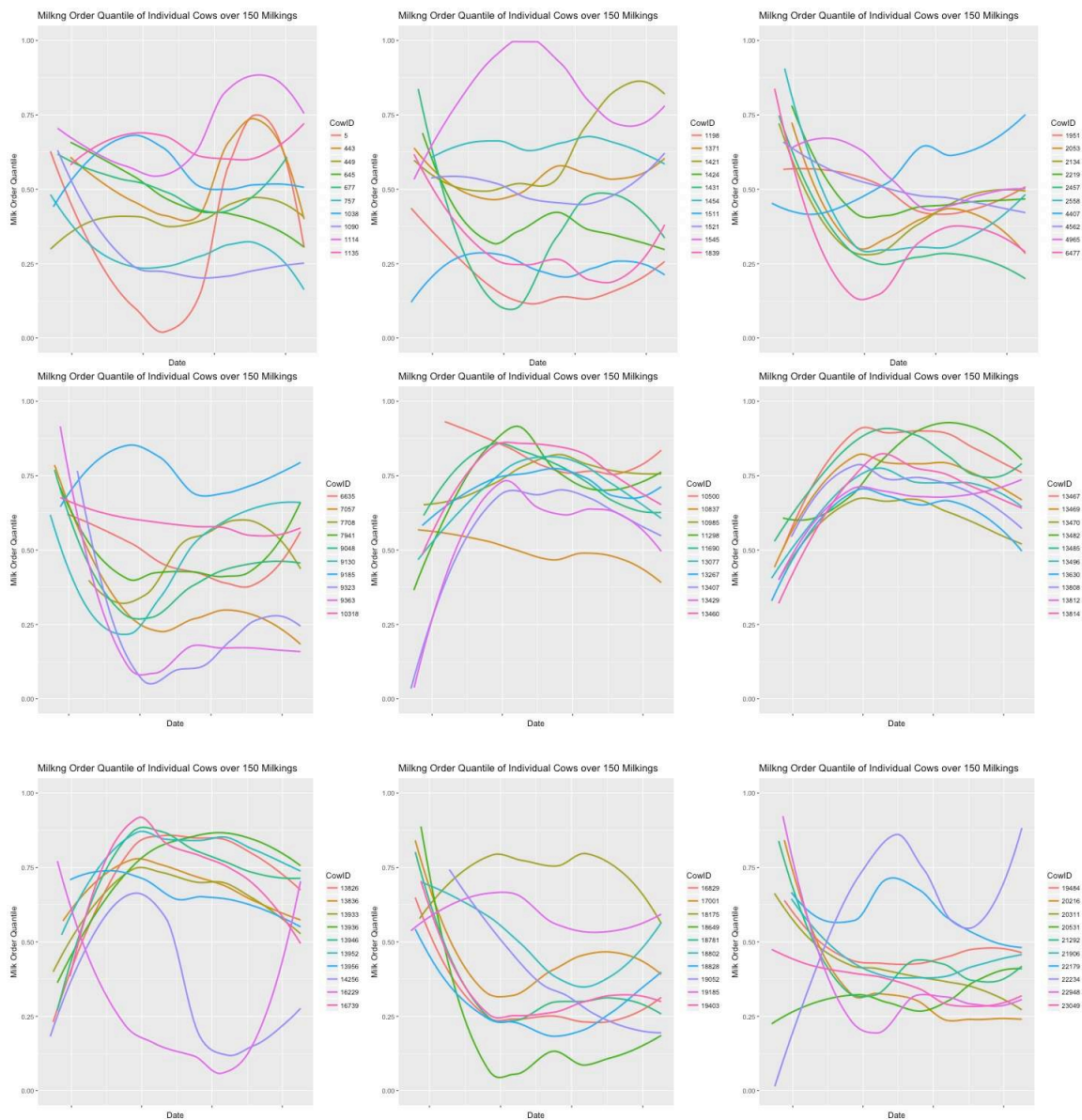
```
quantilestack <- c()
for (i in 1:nrow(dat.inf.cowquant)){
  q <- t(as.vector(dat.inf.cowquant[i,]))

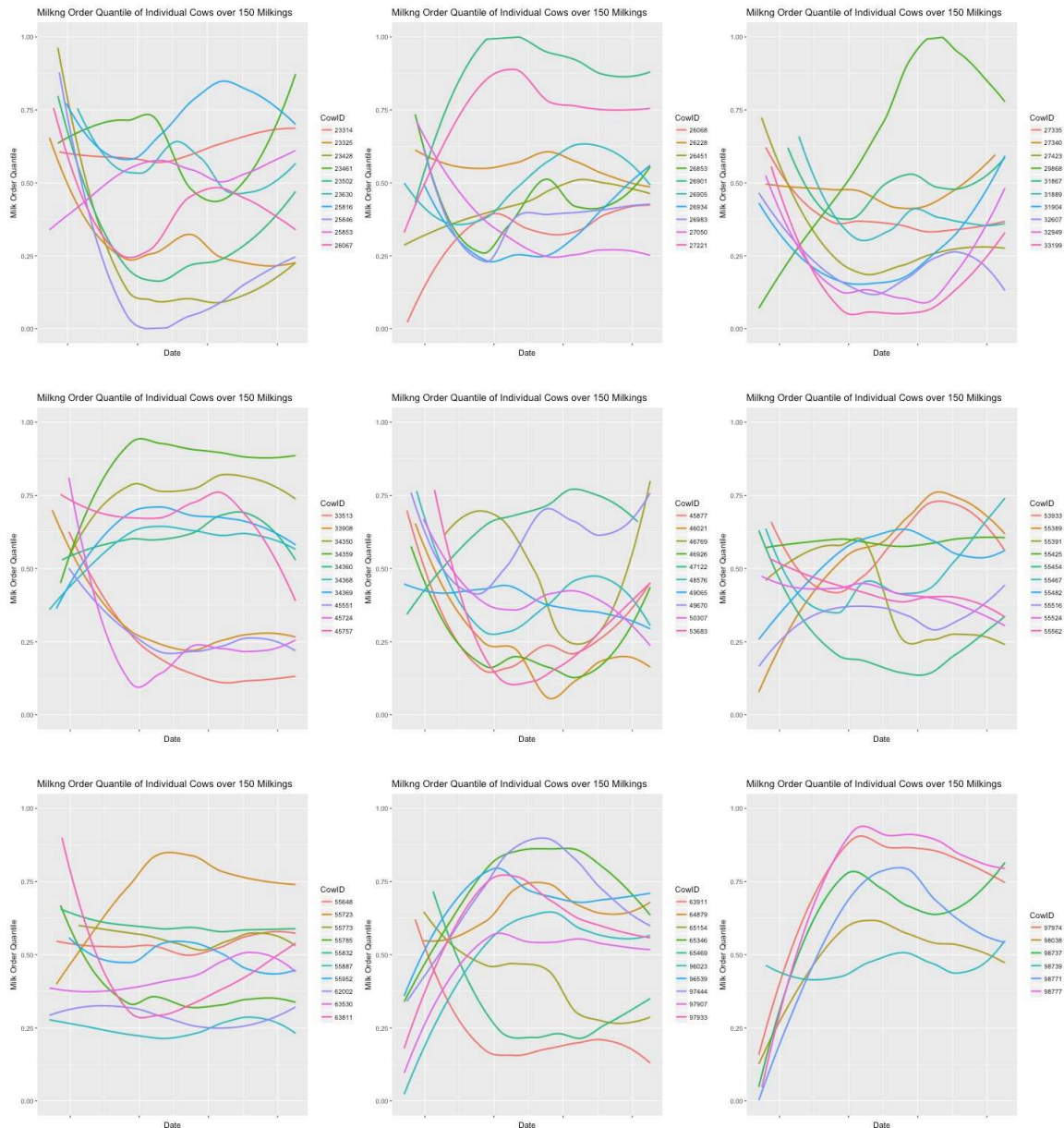
  if (sum(is.na(q))/length(q) < 0.25){
    c <- as.numeric(rep(rownames(dat.inf.cowquant)[i], ncol(dat.inf.cowquant)))
    d <- as.numeric(colnames(dat.inf.cowquant))
    temp <- cbind(d,q,c)
    quantilestack <- rbind(quantilestack, temp)
  }
}

quantilestack<- data.frame(quantilestack)
```

```
names(quantilestack)<- c('Date', 'Quantile', 'CowID')
quantilestack$CowID <- as.factor(quantilestack$CowID)
cowlist_quant <- unique(quantilestack$CowID)
```

```
for (i in seq(1,length(cowlist_quant), by = 10)){
  tempdat <- subset(quantilestack, CowID%in%cowlist_quant[i:(i+9)])
  jpeg(paste('QuantilePlots_Pasture/Plot_', i, '.jpg' , sep = ""))
  print(ggplot(tempdat, aes(x=Date, y=Quantile, colour=CowID)) + geom_smooth(se= F) + ggtitle
('Milkng Order Quantile of Individual Cows over 150 Milkings') + ylab('Milk Order Quantile') +
xlab('Date') + theme(axis.text.x=element_blank()) + ylim(0, 1))
  dev.off()
}
```





#Preliminary Visualizations – PCA Analyses

```
pattern.pasture.cowquant <- data.frame(t(dat.inf.cowquant[1,]))
colnames(pattern.pasture.cowquant)[1] <- rownames(dat.inf.cowquant)[1]
c = 1
```

```
for (i in 2:nrow(dat.inf.cowquant)){
  if (sum(is.na(dat.inf.cowquant[i,]))/length(dat.inf.cowquant[i,]) > 0.15){
    next
  }else{
    pattern.pasture.cowquant <- cbind(pattern.pasture.cowquant,t(dat.inf.cowquant[i,]))
  }
```

```

c = c + 1
colnames(pattern.pasture.cowquant)[c] <- rownames(dat.inf.cowquant)[i]
}
}

datetemp <- datelist_infered[datelist_infered >= (17182+55) & datelist_infered < (17363 - 10) ]
pattern.pasture.cowquant <- pattern.pasture.cowquant[(datelist_infered >= (17182 + 55) &
datelist_infered < (17363 - 10)),] # remove the days that are out of bounds (too early in trial or too late)
pattern.pasture.cowquant <- pattern.pasture.cowquant[order(datetemp),] # reorder rows to be increasing
by day
pattern.pasture.cowquant <- pattern.pasture.cowquant[,-which(colSums(is.na(pattern.pasture.cowquant))
> 0)] # remove cow 1421 as she was

```

```

write.csv(pattern.pasture.cowquant, file = 'CowQuantPattern_Pasture.csv', na = 'NaN', row.names = F)

```

```

library('ggplot2')

```

```

pca.out <- princomp(t(pattern.pasture.cowquant))

```

```

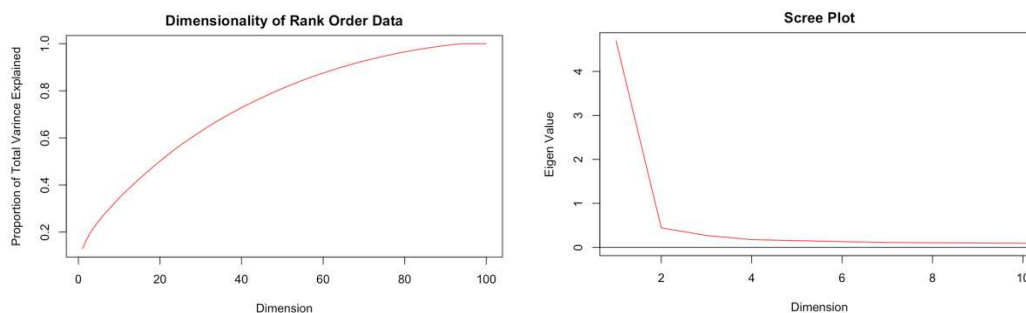
plot(1:100, cumsum(pca.out$sdev/sum(pca.out$sdev)), type = 'l', col = 'red', xlab = 'Dimension', ylab =
'Proportion of Total Varinace Explained', main = 'Dimensionality of Rank Order Data')

```

```

plot(1:100, pca.out$sdev^2, type = 'l', col = 'red', xlab = 'Dimension', ylab = 'Eigen Value', main = 'Scree
Plot', xlim = c(1,10))
abline(h=0)

```



```

temp <- as.data.frame(pca.out$scores[, 1:3]) # keep only the firt 3 dimensions

```

```

ggplot(temp, aes(x=Comp.1, y=Comp.2, col = 'red')) + geom_point() + ggtitle ('Principal
Component Visualization - Social Structure') + ylab('PC2') + xlab('PC1') +
theme(legend.position="none")

```

```

ggplot(temp, aes(x=Comp.1, y=Comp.3, col = 'red')) + geom_point() + ggtitle ('Principal
Component Visualization - Social Structure') + ylab('PC3') + xlab('PC1') +
theme(legend.position="none")

```

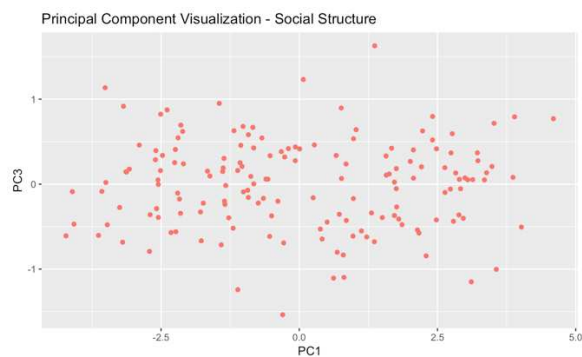
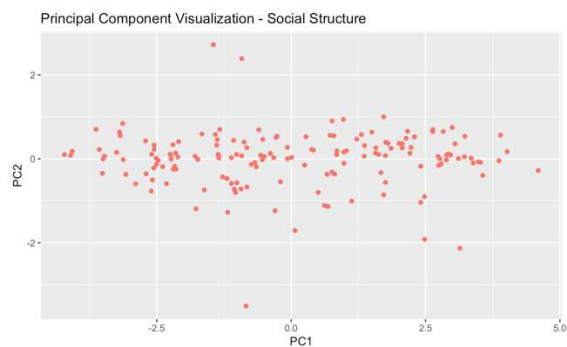


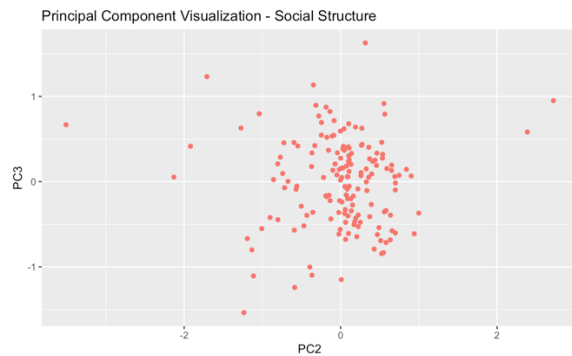
```
ggplot(temp, aes(x=Comp.2, y=Comp.3, col = 'red')) + geom_point() + ggtitle ('Principal
Component Visualization - Social Structure') + ylab('PC3') + xlab('PC2') +
theme(legend.position="none")
```

```
library('plotly')
library('dplyr')
```

```
plot_ly(temp, x = ~Comp.1, y = ~Comp.2, z = ~Comp.3) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'PC1'),
    yaxis = list(title = 'PC2'),
    zaxis = list(title = 'PC3')))
```

```
jpeg('PCAViz.jpg')
ggplot(temp, aes(x=Comp.1, y=Comp.2, col = 'red')) + geom_point() + ggtitle ('Principal
Component Visualization') + ylab('PC2') + xlab('PC1') + theme(legend.position="none")
dev.off()
```

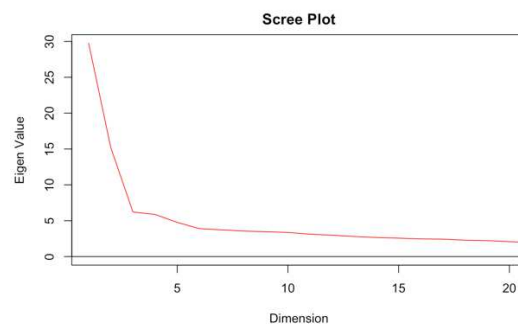
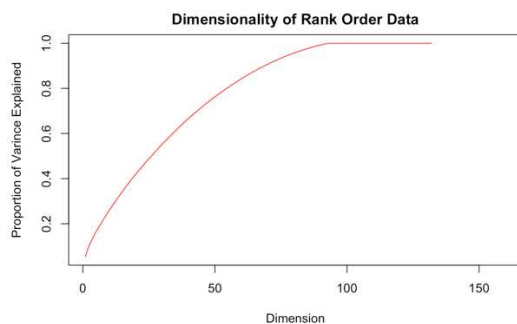




```
X <- scale(pattern.pasture.cowquant)
C <- cov(X)
eig <- eigen(C)
Y <- as.matrix(X) %*% eig$vectors
```

```
plot(1:length(eig$values), cumsum(sqrt(eig$values)/sum(sqrt(eig$values), na.rm = T)), type = 'l',
col = 'red', xlab = 'Dimension', ylab = 'Proportion of Varince Explained', main = 'Dimensionality
of Rank Order Data')
```

```
plot(1:length(eig$values), eig$values, type = 'l', col = 'red', xlab = 'Dimension', ylab = 'Eigen
Value', main = 'Scree Plot', xlim = c(1,20))
abline(h=0)
```



```
temp <- as.data.frame(Y[, 1:3] ) # keep only the first 3 dimensions
names(temp) <- c('Comp.1','Comp.2','Comp.3')
temprow <- which(rownames(pattern.pasture.cowquant) == 'X17280')
tempday <- c(rep('Pen',temprow-1),
rep('Pasture',length(temprow:nrow(pattern.pasture.cowquant)))) )
Environment <- tempday
```

```
ggplot(temp, aes(x=Comp.1, y=Comp.2, col = Environment)) + geom_point() + ggtitle
('Principal Component Visualization - Pasture vs Pen') + ylab('PC2') + xlab('PC1')
```

```
ggplot(temp, aes(x=Comp.1, y=Comp.2, col = datetemp)) + geom_point() + ggtitle ('Principal
Component Visualization - Pasture vs Pen') + ylab('PC2') + xlab('PC1') +
theme(legend.position="none")
```

```
ggplot(temp, aes(x=Comp.1, y=Comp.3, col = Environment)) + geom_point() + ggtitle
('Principal Component Visualization - Pasture vs Pen') + ylab('PC3') + xlab('PC1')
```

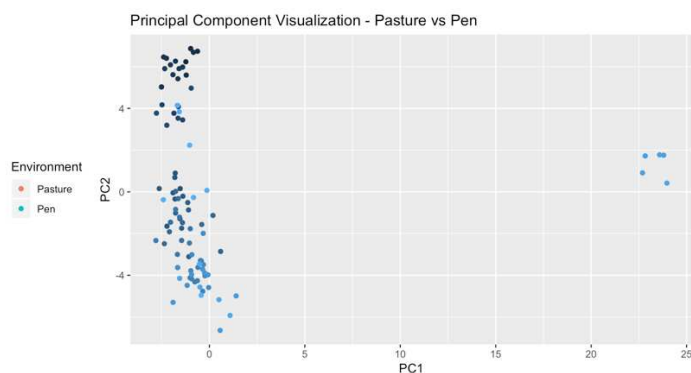
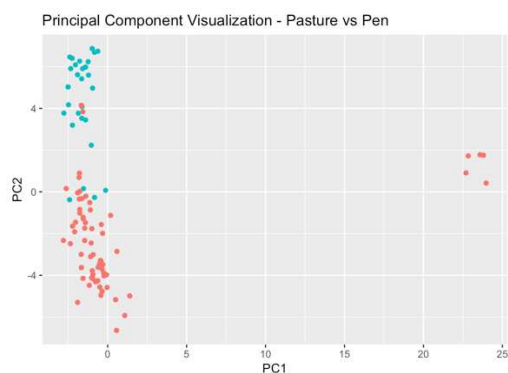
```
ggplot(temp, aes(x=Comp.1, y=Comp.3, col = datetemp)) + geom_point() + ggtitle ('Principal
Component Visualization - Pasture vs Pen') + ylab('PC3') + xlab('PC1') +
theme(legend.position="none")
```

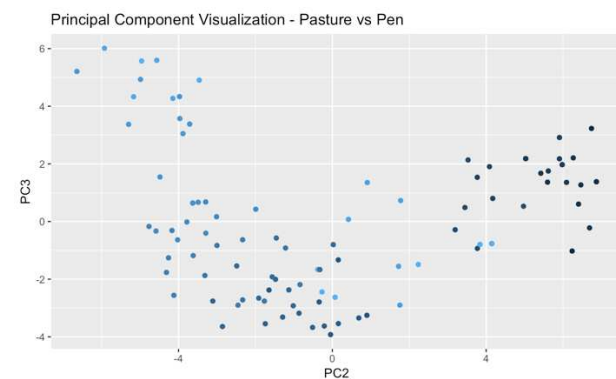
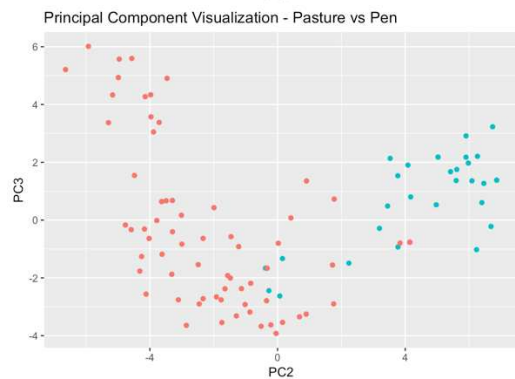
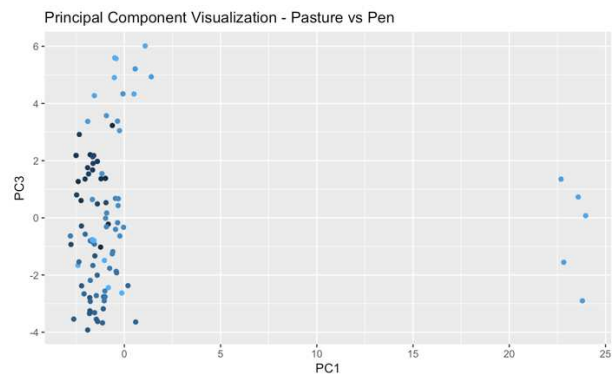
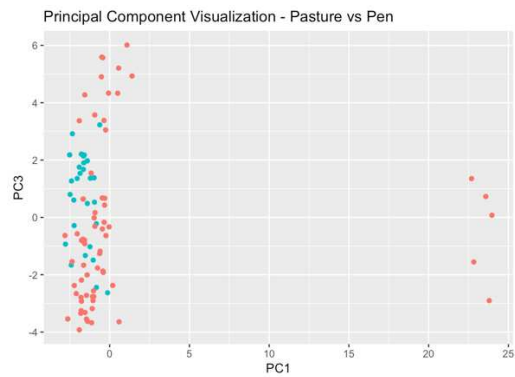
```
ggplot(temp, aes(x=Comp.2, y=Comp.3, col = Environment)) + geom_point() + ggtitle
('Principal Component Visualization - Pasture vs Pen') + ylab('PC3') + xlab('PC2')
```

```
ggplot(temp, aes(x=Comp.2, y=Comp.3, col = datetemp)) + geom_point() + ggtitle ('Principal
Component Visualization - Pasture vs Pen') + ylab('PC3') + xlab('PC2') +
theme(legend.position="none")
```

```
library('plotly')
library('dplyr')
```

```
plot_ly(temp, x = ~Comp.1, y = ~Comp.2, z = ~Comp.3, color = tempday) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'PC1'),
    yaxis = list(title = 'PC2'),
    zaxis = list(title = 'PC3')))
```





```
rownames(pattern.pasture.cowquant)[which(temp$Comp.1>20)]
library(zoo)
as.Date(17342)
as.Date(17343)
as.Date(17344)
as.Date(17345)
as.Date(17346)
```

```
[1] "X17342" "X17343" "X17344" "X17345" "X17346"
[1] "2017-06-25"
[1] "2017-06-26"
[1] "2017-06-27"
[1] "2017-06-28"
[1] "2017-06-29"
```

```
# Making Final Dataset
```

```
##Pasture
```

```
finaldata <- list()
datelist_final <- c()
```

```
r <- 1
```

```

for (i in 1:length(orderdata_infered)){
  if (orderdata_infered[[i]]$Date >= (17280 + 10) & orderdata_infered[[i]]$Date < (17363-10) &
orderdata_infered[[i]]$Milking == 1 ){

    if( orderdata_infered[[i]]$Date %in% c(17342, 17343, 17344, 17345, 17346)){

      print(i)
      next # remove the weird days from the PCA analysis

    }

    finaldata[[r]] <- orderdata_infered[[i]]
    r <- r +1

    datelist_final <- c(datelist_final, orderdata_infered[[i]]$Date )

  }
}

print('Number of Milking Observations:')
length(finaldata)
[1] "Number of Milking Observations:"
[1] 50

###Pen Data

finaldata_pen <- list()
datelist_final_pen <- c()

r <- 1

for (i in 1:length(orderdata_infered)){
  if (orderdata_infered[[i]]$Date >= (17182 + 50) & orderdata_infered[[i]]$Date < (17280 - 5) &
orderdata_infered[[i]]$Milking == 1 ){

    finaldata_pen[[r]] <- orderdata_infered[[i]]
    r <- r +1

    datelist_final_pen <- c(datelist_final_pen, orderdata_infered[[i]]$Date )

  }
}

print('Number of Milking Observations:')
length(finaldata_pen)
[1] "Number of Milking Observations:"
[1] 31

```

```
#Tracking Cow Attendance
```

```
#Pasture Data
```

```
cowattendance.final <- as.data.frame(matrix(NA, nrow = length(cowlist), ncol =  
length(finaldata)))  
rownames(cowattendance.final) <- cowlist  
colnames(cowattendance.final) <- datelist_final
```

```
for (j in 1:ncol(cowattendance.final)){  
  temp <- finaldata[[j]]$data  
  for (i in 1:nrow(cowattendance.final)){  
    cowtemp <- rownames(cowattendance.final)[i]  
    if (cowtemp %in% temp[,1]){  
      cowattendance.final[i,j] <- 1  
    }else{  
      cowattendance.final[i,j] <- 0  
    }  
  }  
}
```

```
cowattendance.final$PercentAttendance <- apply(cowattendance.final,1, FUN =  
sum)/ncol(cowattendance.final)
```

```
print('Number of cows that made it to pasture:')  
length(rownames(cowattendance.final)[cowattendance.final$PercentAttendance>0])
```

```
finalcowlist.pasture <-  
rownames(cowattendance.final)[cowattendance.final$PercentAttendance>0.50]  
print('Number of cows retained in network')  
length(finalcowlist.pasture)  
[1] "Number of cows that made it to pasture:"  
[1] 186  
[1] "Number of cows retained in network"  
[1] 182
```

```
##Pen Data
```

```
cowattendance.final.pen <- as.data.frame(matrix(NA, nrow = length(cowlist), ncol =  
length(finaldata_pen)))  
rownames(cowattendance.final.pen) <- cowlist  
colnames(cowattendance.final.pen) <- datelist_final_pen
```

```
for (j in 1:ncol(cowattendance.final.pen)){  
  temp <- finaldata_pen[[j]]$data  
  for (i in 1:nrow(cowattendance.final.pen)){  
    cowtemp <- rownames(cowattendance.final.pen)[i]  
    if (cowtemp %in% temp[,1]){
```



```

    cowattendance.final.pen[i,j] <- 1
  }else{
    cowattendance.final.pen[i,j] <- 0
  }
}
}

cowattendance.final.pen$PercentAttendance <- apply(cowattendance.final.pen,1, FUN =
sum)/ncol(cowattendance.final.pen)

print('Number of cows that made it past freshening:')
length(rownames(cowattendance.final.pen)[cowattendance.final.pen$PercentAttendance>0])

finalcowlist.pen <-
rownames(cowattendance.final.pen)[cowattendance.final.pen$PercentAttendance>0.50]
print('Number of cows retained in network')
length(finalcowlist.pen)
[1] "Number of cows that made it past freshening:"
[1] 191
[1] "Number of cows retained in network"
[1] 186

#Creating the Adjacency Matrix

##Pasture Data

networkdat.pasture <- matrix(0, nrow=length(finalcowlist.pasture),
ncol=length(finalcowlist.pasture))
rownames(networkdat.pasture) <- finalcowlist.pasture
colnames(networkdat.pasture) <- finalcowlist.pasture

skipcount <- 0

for (j in 1:length(finaldata)){

  temp <- finaldata[[j]]$data[,1]

  if (finaldata[[j]]$Milking != 1){ # checking I only have milking 1 records
    skipcount <- skipcount + 1
    next # skip if this isn't morning data coming in from the pasture
  }

  for (i in 3:(length(temp)-2)){
    if (temp[i] %in% finalcowlist.pasture){ # if obs cow isn't network list, skip
      rowtemp <- which(temp[i] == finalcowlist.pasture)
      if (temp[i+1] %in% finalcowlist.pasture){ # if their opponent isn't in network list, skip
        coltemp <- which(temp[i+1] == finalcowlist.pasture)

```

```

    networkdat.pasture[rowtemp, coltemp] <- networkdat.pasture[rowtemp, coltemp] + 1 # add
one win to count of that dyadic interaction
  }
}

```

```

}
}

```

```

skipcount
table(networkdat.pasture)
temp <- table(networkdat.pasture)
print('Proportion unpopulated pairwise interactions')
temp[1]/sum(temp)

```

```

adj.pasture <- networkdat.pasture
networkdat.pasture
  0  1  2  3  4  5  6  7  8  9 13 29
26447 5162 1157 261 72 12 5 2 2 2 1 1
[1] "Proportion unpopulated pairwise interactions"
0
0.7984241

```

```

##Pen Data

```

```

networkdat.pen <- matrix(0, nrow=length(finalcowlist.pen), ncol=length(finalcowlist.pen))
rownames(networkdat.pen) <- finalcowlist.pen
colnames(networkdat.pen) <- finalcowlist.pen

```

```

skipcount <- 0

```

```

for (j in 1:length(finaldata_pen)){

```

```

  temp <- finaldata_pen[[j]]$data[,1]

```

```

  if (finaldata_pen[[j]]$Milking != 1){ # checking I only have milking 1 records
    skipcount <- skipcount + 1
    next # skip if this isn't morning data coming in from the pasture
  }

```

```

  for (i in 3:(length(temp)-2)){
    if (temp[i] %in% finalcowlist.pen){ # if obs cow isn't network list, skip
      rowtemp <- which(temp[i] == finalcowlist.pen)
      if (temp[i+1] %in% finalcowlist.pen){ # if their opponent isn't in network list, skip
        coltemp <- which(temp[i+1] == finalcowlist.pen)
        networkdat.pen[rowtemp, coltemp] <- networkdat.pen[rowtemp, coltemp] + 1 # add one
win to count of that dyadic interaction
      }
    }
  }
}

```

```

    }

  }
}

skipcount
table(networkdat.pen)
temp <-table(networkdat.pen)
print('Proportion unpopulated pairwise interactions')
temp[1]/sum(temp)

adj.pen <- networkdat.pen
networkdat.pen
  0  1  2  3  4  5  6
31926 2863 722 165 36 7 2
[1] "Proportion unpopulated pairwise interactions"
0
0.89376

#####
#Computing Facial Biometric Values

library('ggplot2')

birthdat <- read.csv('CowBirthDates.csv', stringsAsFactors = F)
birthdat$DOB <- as.Date('01/16/2017', "%m/%d/%Y") - as.Date(birthdat$DOB, "%m/%d/%Y")

qplot(as.vector(birthdat$DOB), geom="histogram", xlab = 'Age in Days', ylab = "", main =
'Distribution of Cow Ages', col=I("black"), fill=I("blue"), alpha=I(.5), bins = 20)

```



```

##Eye Biometrics – Overall

library('lme4')

biom <- read.csv('Biom Data/Eye_Results_Master.csv', header=TRUE)
biom <- biom[,1:179] # getting rid of junk columns

```

```

biom_names <- names(biom)

start <- 54
done <- 179

dat <- biom
#dat_out <- data.frame(CowId = unique(biom$Cow.ID))

for (i in start:done){
  rm(temp)
  temp <- data.frame(as.factor(dat$Cow.ID))
  names(temp) <- 'Cow.ID'
  temp$Day <- dat$Day
  temp$Side <- dat$Side
  temp$biometric<-dat[,i]

  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
  nested within cow

  temp <- coef(mix_out)

  if (i == start){
    dat_out <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
  }

  dat_out <- cbind(dat_out, unlist(temp$Cow.ID))
  colnames(dat_out)[ncol(dat_out)] <- biom_names[i]
}

eyedat_final <- dat_out[, c('Cow.ID', 'Z1_front_6', 'Z2_6', 'Z3_front_6', 'Z4_6', 'Z7_length_6',
'Z9_poly_6', 'Z10_poly_6', 'Z11_poly_6', 'Z11_linear_6')]

##Muzzle Biometrics – Overall

biom <- read.csv('Biom Data/Muzzle_Results_Master.csv', header=TRUE)
biom <- biom[,1:96] # getting rid of junk columns
biom$Cow.ID <- as.factor(biom$Cow.ID)
biom$Side <- as.factor(biom$Side)
biom$Day <- as.factor(biom$Day)
biom$Rep <- as.factor(biom$Rep)

biom <- biom[,-78] # this biometric is clearly having problems dividing by zero

biom_names <- names(biom)

```

```

start <- 58
done <- ncol(biom)

dat<- biom

for (i in start:done){
  rm(temp)
  temp <- data.frame(as.factor(dat$Cow.ID))
  names(temp) <- 'Cow.ID'
  temp$Day <- dat$Day
  temp$Side <- dat$Side
  temp$biometric<-dat[,i]

  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
  nested within cow

  temp <- coef(mix_out)

  if (i == start){
    dat_out <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
  }

  dat_out <- cbind(dat_out, unlist(temp$Cow.ID))
  colnames(dat_out)[ncol(dat_out)] <- biom_names[i]
}

muzzledat_final <- dat_out[, c('Cow.ID', 'NFP_LF', 'NFPP_LF', 'NPA', 'ULRP_V1', 'MTP_V1',
'CTP_V1', 'CLTR_V1')]

##Topline Biometrics – Overall

biom <- read.csv('Biom Data/Topline_Results_Master.csv', header=TRUE)
biom <- biom[,1:86] # get rid of nonsense columns

biom_names <- names(biom)
start <- 30
done <- ncol(biom)

# review of biometric distributions revealed two corrupted files to delete
biom[1104, start:done] <- NA
biom[427, start:done] <- NA

```

```
dat<- biom
```

```
for (i in start:done){
  rm(temp)
  temp <- data.frame(as.factor(dat$Cow.ID))
  names(temp) <- 'Cow.ID'
  temp$Day <- dat$Day
  temp$Side <- dat$Side
  temp$biometric<-dat[,i]

  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
  nested within cow

  temp <- coef(mix_out)

  if (i == start){
    dat_out <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
  }

  dat_out <- cbind(dat_out, unlist(temp$Cow.ID))
  colnames(dat_out)[ncol(dat_out)] <- biom_names[i]

}

topdat_final <- dat_out[, c('Cow.ID', 'NRP', 'SMRP_V1', 'MNRP_V1', 'MDP_V1', 'NDP_V1',
'NaDP_V1', 'MIP_V1', 'MIPP_V1', 'NIP_V1', 'NaTLP', 'MTLP_V1', 'MTLP_V1', 'STLP_V1',
'ULTLP_V1', 'SMLP_V1', 'MNLP_V1')]

##Forehead Biometrics – Overall

biom <- read.csv('Biom Data/Forehead_Results_Master.csv', header=TRUE)
biom$Cow.ID <- as.factor(biom$Cow.ID)
biom$Side <- as.factor(biom$Side)
biom$Day <- as.factor(biom$Day)
biom$Rep <- as.factor(biom$Rep)
biom <- biom[,1:522]

biom_names <- names(biom)
start <- 34
done <- ncol(biom)

biom[1104, start:done] <- NA
biom[427, start:done] <- NA
```



```

biom[49, start:done] <- NA

dat<- biom

for (i in start:done){
  rm(temp)
  temp <- data.frame(as.factor(dat$Cow.ID))
  names(temp) <- 'Cow.ID'
  temp$Day <- dat$Day
  temp$Side <- dat$Side
  temp$biometric<-dat[,i]

  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
  nested within cow

  temp <- coef(mix_out)

  if (i == start){
    dat_out <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
  }

  dat_out <- cbind(dat_out, unlist(temp$Cow.ID))
  colnames(dat_out)[ncol(dat_out)] <- biom_names[i]
}

foreheaddat_final <- dat_out[, c('Cow.ID', 'Eye.Forehead.Size.Ratio_Poly_V1',
'Eye.Forehead.Size.Ratio_Linear_V1', 'Eye.Cranio.Size.Ratio_Poly_V1',
'Eye.Topline.Size.Ratio_Poly_V1', 'Eye.Topline.Size.Ratio_Linear_V1',
'Eye.Sinus.Size.Ratio_Poly_V1', 'Eye.Sinus.Size.Ratio_Linear_V1',
'Midface.Thickness.Proportion_V1', 'Overall.Eye.Size_V1', 'Overall.Eye.Angle_Angle_V1',
'Muzzle.Size.Proportion_V9', 'Cheek.Nose.Size.Proportion_V14', 'Chin.Length.Proportion_V6',
'Jowel.Jaw.Length.Proportion_V1', 'Jaw.Angle_Slope_V1', 'Jaw.Midface.Size.Ratio_V1',
'Eye.Orbital.Projection.Proportion_V1', 'Nasion.Thickness.Proportion_V1',
'Eye.Orbital.Eye.Height.Ratio_V1', 'Eye.Orbital.Thickness.Proportion_Poly_V1',
'Forehead.Temple.Ratio_V1', 'Forehead.Eye.Angle_Slope_V3',
'Forehead.Topline.Angle_Slope_V3', 'Forehead.Jaw.Angle_Slope_V3',
'Forehead.Topline.Length.Ratio_V3', 'Cranio.Topline.Length.Ratio_V3',
'Forehead.Poll.Length.Ratio_V3', 'Poll.Height.Proportion_V4',
'Forehead.Width.to.Length.Ratio_V3')]

colnames(foreheaddat_final) <- c('Cow.ID', 'EFSRP_V1', 'EFSRL_V1', 'ECSR_P_V1',
'ETSRP_V1', 'ETSRL_V1', 'ESSRP_V1', 'ESSRL_V1', 'MTP_V1.1', 'OES_V1', 'OEAA_V1',
'MSP_V9', 'CNSP_V14', 'CLP_V6', 'JJLP_V7', 'JAS_V1', 'JMSR_V1', 'EOPP_V1', 'NsTP_V1',
'EOEHR_V1', 'EOTPP_V1', 'FTR_V1', 'FEAS_V3', 'FTAS_V3', 'FJAS_V3', 'FTLR_V3',
'CTLR_V3', 'FPLP_V3', 'PHP_V4', 'FWLP_V3')

```

```
##Eye Biometrics – Sides
```

```
library('lme4')
```

```
biom <- read.csv('Biom Data/Eye_Results_Master.csv', header=TRUE)
biom <- biom[,1:179] # getting rid of junk columns
```

```
biom_names <- names(biom)
```

```
start <- 54
done <- 179
```

```
rm(dat_out_L)
rm(dat_out_R)
dat <- biom
#dat_out <- data.frame(CowId = unique(biom$Cow.ID))
```

```
for (i in start:done){
  rm(temp)
  temp <- data.frame(as.factor(dat$Cow.ID))
  names(temp) <- 'Cow.ID'
  temp$Day <- dat$Day
  temp$Side <- dat$Side
  temp$biometric<-dat[,i]
```

```
  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
  nested within cow
```

```
  temp <- coef(mix_out)
```

```
  if (i == start){
    dat_out_L <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
    dat_out_R <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
  }
```

```
  templ <- length(unlist(temp$`Side:Cow.ID`))
```

```
  dat_out_L <- cbind(dat_out_L, unlist(temp$`Side:Cow.ID`)[1:(templ/2)])
  colnames(dat_out_L)[ncol(dat_out_L)] <- paste('L_', biom_names[i], sep = "")
```

```
  dat_out_R <- cbind(dat_out_R, unlist(temp$`Side:Cow.ID`)[(templ/2+1):templ])
  colnames(dat_out_R)[ncol(dat_out_R)] <- paste('R_', biom_names[i], sep = "")
```

```
}
```

```
eyedat_final_L <- dat_out_L[, c('Cow.ID', 'L_Z1_front_6', 'L_Z2_6', 'L_Z3_front_6', 'L_Z4_6',
'L_Z7_length_6', 'L_Z9_poly_6', 'L_Z10_poly_6', 'L_Z11_poly_6', 'L_Z11_linear_6')]
```

```
eyedat_final_R <- dat_out_R[, c('Cow.ID', 'R_Z1_front_6', 'R_Z2_6', 'R_Z3_front_6', 'R_Z4_6',
'R_Z7_length_6', 'R_Z9_poly_6', 'R_Z10_poly_6', 'R_Z11_poly_6', 'R_Z11_linear_6')]
```

##Muzzle Biometrics – Sides

```
biom <- read.csv('Biom Data/Muzzle_Results_Master.csv', header=TRUE)
```

```
biom <- biom[,1:96] # getting rid of junk columns
```

```
biom$Cow.ID <- as.factor(biom$Cow.ID)
```

```
biom$Side <- as.factor(biom$Side)
```

```
biom$Day <- as.factor(biom$Day)
```

```
biom$Rep <- as.factor(biom$Rep)
```

```
biom <- biom[,-78] # this biometric is clearly having problems dividing by zero
```

```
biom_names <- names(biom)
```

```
start <- 58
```

```
done <- ncol(biom)
```

```
dat<- biom
```

```
rm(dat_out_L)
```

```
rm(dat_out_R)
```

```
for (i in start:done){
```

```
  rm(temp)
```

```
  temp <- data.frame(as.factor(dat$Cow.ID))
```

```
  names(temp) <- 'Cow.ID'
```

```
  temp$Day <- dat$Day
```

```
  temp$Side <- dat$Side
```

```
  temp$biometric<-dat[,i]
```

```
  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
  nested within cow
```

```
  temp <- coef(mix_out)
```

```
  if (i == start){
```

```
    dat_out_L <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
```

```
    dat_out_R <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
```

```
  }
```

```
templ <- length(unlist(temp$`Side:Cow.ID`))
```

```

dat_out_L <- cbind(dat_out_L, unlist(temp$`Side:Cow.ID`)[1:(templ/2)])
colnames(dat_out_L)[ncol(dat_out_L)] <- paste('L_', biom_names[i], sep = "")

dat_out_R <- cbind(dat_out_R, unlist(temp$`Side:Cow.ID`)[(templ/2+1):templ])
colnames(dat_out_R)[ncol(dat_out_R)] <- paste('R_', biom_names[i], sep = "")

}

muzzledat_final_L <- dat_out_L[, c('Cow.ID', 'L_NFP_LF', 'L_NFPP_LF', 'L_NPA',
'L_ULRP_V1', 'L_MTP_V1', 'L_CTP_V1', 'L_CLTR_V1')]

muzzledat_final_R <- dat_out_R[, c('Cow.ID', 'R_NFP_LF', 'R_NFPP_LF', 'R_NPA',
'R_ULRP_V1', 'R_MTP_V1', 'R_CTP_V1', 'R_CLTR_V1')]

##Topline Biometrics – Sides

biom <- read.csv('Biom Data/Topline_Results_Master.csv', header=TRUE)
biom <- biom[,1:86] # get rid of nonsense columns

biom_names <- names(biom)
start <- 30
done <- ncol(biom)

# review of biometric distributions revealed two corrupted files to delete
biom[1104, start:done] <- NA
biom[427, start:done] <- NA

dat<- biom
rm(dat_out_L)
rm(dat_out_R)

for (i in start:done){
  rm(temp)
  temp <- data.frame(as.factor(dat$Cow.ID))
  names(temp) <- 'Cow.ID'
  temp$Day <- dat$Day
  temp$Side <- dat$Side
  temp$biometric<-dat[,i]

  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
  nested within cow

  temp <- coef(mix_out)

```

```

if (i == start){
  dat_out_L <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
  dat_out_R <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
}

templ <- length(unlist(temp$`Side:Cow.ID`))

dat_out_L <- cbind(dat_out_L, unlist(temp$`Side:Cow.ID`)[1:(templ/2)])
colnames(dat_out_L)[ncol(dat_out_L)] <- paste('L_', biom_names[i], sep = "")

dat_out_R <- cbind(dat_out_R, unlist(temp$`Side:Cow.ID`)[(templ/2+1):templ])
colnames(dat_out_R)[ncol(dat_out_R)] <- paste('R_', biom_names[i], sep = "")

}

topdat_final_L <- dat_out_L[, c('Cow.ID', 'L_NRP', 'L_SMRP_V1', 'L_MNRP_V1',
'L_MDP_V1', 'L_NDP_V1', 'L_NaDP_V1', 'L_MIP_V1', 'L_MIPP_V1', 'L_NIP_V1',
'L_NaTLP', 'L_MTLP_V1', 'L_MTLP_V1', 'L_STLP_V1', 'L_ULTLP_V1', 'L_SMLP_V1',
'L_MNLP_V1')]

topdat_final_R <- dat_out_R[, c('Cow.ID', 'R_NRP', 'R_SMRP_V1', 'R_MNRP_V1',
'R_MDP_V1', 'R_NDP_V1', 'R_NaDP_V1', 'R_MIP_V1', 'R_MIPP_V1', 'R_NIP_V1',
'R_NaTLP', 'R_MTLP_V1', 'R_MTLP_V1', 'R_STLP_V1', 'R_ULTLP_V1', 'R_SMLP_V1',
'R_MNLP_V1')]

##Forehead Biometrics – Sides

biom <- read.csv('Biom Data/Forehead_Results_Master.csv', header=TRUE)
biom$Cow.ID <- as.factor(biom$Cow.ID)
biom$Side <- as.factor(biom$Side)
biom$Day <- as.factor(biom$Day)
biom$Rep <- as.factor(biom$Rep)
biom <- biom[,1:522]

biom_names <- names(biom)

start <- 34
done <- ncol(biom)

biom[1104, start:done] <- NA
biom[427, start:done] <- NA
biom[49, start:done] <- NA

```

```

dat<- biom
rm(dat_out_L)
rm(dat_out_R)

for (i in start:done){
  rm(temp)
  temp <- data.frame(as.factor(dat$Cow.ID))
  names(temp) <- 'Cow.ID'
  temp$Day <- dat$Day
  temp$Side <- dat$Side
  temp$biometric<-dat[,i]

  mix_out<-lmer(biometric ~ 1 + (1 | Cow.ID/Side/Day), data=temp) # day nested within side
  nested within cow

  temp <- coef(mix_out)

  if (i == start){
    dat_out_L <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
    dat_out_R <- data.frame(Cow.ID = row.names(temp$Cow.ID) )
  }

  templ <- length(unlist(temp$`Side:Cow.ID`))

  dat_out_L <- cbind(dat_out_L, unlist(temp$`Side:Cow.ID`)[1:(templ/2)])
  colnames(dat_out_L)[ncol(dat_out_L)] <- biom_names[i]

  dat_out_R <- cbind(dat_out_R, unlist(temp$`Side:Cow.ID`)[(templ/2+1):templ])
  colnames(dat_out_R)[ncol(dat_out_R)] <- biom_names[i]

}

foreheaddat_final_L <- dat_out_L[, c('Cow.ID', 'Eye.Forehead.Size.Ratio_Poly_V1',
'Eye.Forehead.Size.Ratio_Linear_V1', 'Eye.Cranio.Size.Ratio_Poly_V1',
'Eye.Topline.Size.Ratio_Poly_V1', 'Eye.Topline.Size.Ratio_Linear_V1',
'Eye.Sinus.Size.Ratio_Poly_V1', 'Eye.Sinus.Size.Ratio_Linear_V1',
'Midface.Thickness.Proportion_V1', 'Overall.Eye.Size_V1', 'Overall.Eye.Angle_Angle_V1',
'Muzzle.Size.Proportion_V9', 'Cheek.Nose.Size.Proportion_V14', 'Chin.Length.Proportion_V6',
'Jowel.Jaw.Length.Proportion_V1', 'Jaw.Angle_Slope_V1', 'Jaw.Midface.Size.Ratio_V1',
'Eye.Orbital.Projection.Proportion_V1', 'Nasion.Thickness.Proportion_V1',
'Eye.Orbital.Eye.Height.Ratio_V1', 'Eye.Orbital.Thickness.Proportion_Poly_V1',
'Forehead.Temple.Ratio_V1', 'Forehead.Eye.Angle_Slope_V3',
'Forehead.Topline.Angle_Slope_V3', 'Forehead.Jaw.Angle_Slope_V3',
'Forehead.Topline.Length.Ratio_V3', 'Cranio.Topline.Length.Ratio_V3',

```



```
'Forehead.Poll.Length.Ratio_V3', 'Poll.Height.Proportion_V4',
'Forehead.Width.to.Length.Ratio_V3')]
```

```
foreheaddat_final_R <- dat_out_R[, c('Cow.ID', 'Eye.Forehead.Size.Ratio_Poly_V1',
'Eye.Forehead.Size.Ratio_Linear_V1', 'Eye.Cranio.Size.Ratio_Poly_V1',
'Eye.Topline.Size.Ratio_Poly_V1', 'Eye.Topline.Size.Ratio_Linear_V1',
'Eye.Sinus.Size.Ratio_Poly_V1', 'Eye.Sinus.Size.Ratio_Linear_V1',
'Midface.Thickness.Proportion_V1', 'Overall.Eye.Size_V1', 'Overall.Eye.Angle_Angle_V1',
'Muzzle.Size.Proportion_V9', 'Cheek.Nose.Size.Proportion_V14', 'Chin.Length.Proportion_V6',
'Jowel.Jaw.Length.Proportion_V1', 'Jaw.Angle_Slope_V1', 'Jaw.Midface.Size.Ratio_V1',
'Eye.Orbital.Projection.Proportion_V1', 'Nasion.Thickness.Proportion_V1',
'Eye.Orbital.Eye.Height.Ratio_V1', 'Eye.Orbital.Thickness.Proportion_Poly_V1',
'Forehead.Temple.Ratio_V1', 'Forehead.Eye.Angle_Slope_V3',
'Forehead.Topline.Angle_Slope_V3', 'Forehead.Jaw.Angle_Slope_V3',
'Forehead.Topline.Length.Ratio_V3', 'Cranio.Topline.Length.Ratio_V3',
'Forehead.Poll.Length.Ratio_V3', 'Poll.Height.Proportion_V4',
'Forehead.Width.to.Length.Ratio_V3')]
```

```
colnames(foreheaddat_final_L) <- c('Cow.ID', 'L_EFSRP_V1', 'L_EFSRL_V1', 'L_ECSRP_V1',
'L_ETSRP_V1', 'L_ETSRL_V1', 'L_ESSRP_V1', 'L_ESSRL_V1', 'L_MTP_V1.1', 'L_OES_V1',
'L_OEAA_V1', 'L_MSP_V9', 'L_CNSP_V14', 'L_CLP_V6', 'L_JJLP_V7', 'L_JAS_V1',
'L_JMSR_V1', 'L_EOPP_V1', 'L_NsTP_V1', 'L_EOEHR_V1', 'L_EOTPP_V1', 'L_FTR_V1',
'L_FEAS_V3', 'L_FTAS_V3', 'L_FJAS_V3', 'L_FTLR_V3', 'L_CTLR_V3', 'L_FPLP_V3',
'L_PHP_V4', 'L_FWLP_V3')
```

```
colnames(foreheaddat_final_R) <- c('Cow.ID', 'R_EFSRP_V1', 'R_EFSRL_V1', 'R_ECSRP_V1',
'R_ETSRP_V1', 'R_ETSRL_V1', 'R_ESSRP_V1', 'R_ESSRL_V1', 'R_MTP_V1.1', 'R_OES_V1',
'R_OEAA_V1', 'R_MSP_V9', 'R_CNSP_V14', 'R_CLP_V6', 'R_JJLP_V7', 'R_JAS_V1',
'R_JMSR_V1', 'R_EOPP_V1', 'R_NsTP_V1', 'R_EOEHR_V1', 'R_EOTPP_V1', 'R_FTR_V1',
'R_FEAS_V3', 'R_FTAS_V3', 'R_FJAS_V3', 'R_FTLR_V3', 'R_CTLR_V3', 'R_FPLP_V3',
'R_PHP_V4', 'R_FWLP_V3')
```

##Creating Final Biometrics Dataset – Overall

```
biomdat_final <- merge(birthdat, eyedat_final, by = 'Cow.ID')
biomdat_final <- merge(biomdat_final, muzzledat_final, by = 'Cow.ID')
biomdat_final <- merge(biomdat_final, topdat_final, by = 'Cow.ID')
biomdat_final <- merge(biomdat_final, foreheaddat_final, by = 'Cow.ID')
```

```
biom <- read.csv('Biom Data/Forehead_Results_Master.csv', header=TRUE)#forehead runs had
the highest number of dropped cows
biom[1104, start:done] <- NA
biom[427, start:done] <- NA
biom[49, start:done] <- NA
```

```

biom <- subset(biom, biom$Missing==0)
biom <- subset(biom, !is.na(biom[20]))
table(biom$Cow.ID)
sum(table(biom$Cow.ID) < 4)
sum(table(biom$Cow.ID) < 8)/length(table(biom$Cow.ID))
length(table(biom$Cow.ID))
[1] 0
[1] 0.1401869
[1] 107

```

##Creating Final Biometrics Dataset – Sides

```

biomdat_final_L <- merge(birthdat, eyedat_final_L, by = 'Cow.ID')
biomdat_final_L <- merge(biomdat_final_L, muzzledat_final_L, by = 'Cow.ID')
biomdat_final_L <- merge(biomdat_final_L, topdat_final_L, by = 'Cow.ID')
biomdat_final_L <- merge(biomdat_final_L, foreheaddat_final_L, by = 'Cow.ID')

```

```

biomdat_final_R <- merge(birthdat, eyedat_final_R, by = 'Cow.ID')
biomdat_final_R <- merge(biomdat_final_R, muzzledat_final_R, by = 'Cow.ID')
biomdat_final_R <- merge(biomdat_final_R, topdat_final_R, by = 'Cow.ID')
biomdat_final_R <- merge(biomdat_final_R, foreheaddat_final_R, by = 'Cow.ID')

```

```

#####
#Rank Order Analysis - Pasture

```

```

library('Perc')
#load("~/Documents/Research/Dairy_Behavior/Projects/Organilac
Network/OrganilacNetworkWorkspace_PercRuns2.RData")

```

```

pasturedat <- as.conflictmat(networkdat.pasture)
pasturetrans <- transitivity(pasturedat)

```

```

pasturetrans$transitivity
pasturetrans$alpha

```

```

domprob.pasture.2 <- conductance(pasturedat, maxLength = 2)
domprob.pasture.3 <- conductance(pasturedat, maxLength = 3)

```

```

infogain12 <- domprob.pasture.2$imputed.conf - pasturedat
plotConfmat(infogain12, ordering = NA, labels = FALSE)

```

```

infogain23 <- domprob.pasture.3$imputed.conf - domprob.pasture.2$imputed.conf
plotConfmat(infogain23, ordering = NA, labels = FALSE)

```

```

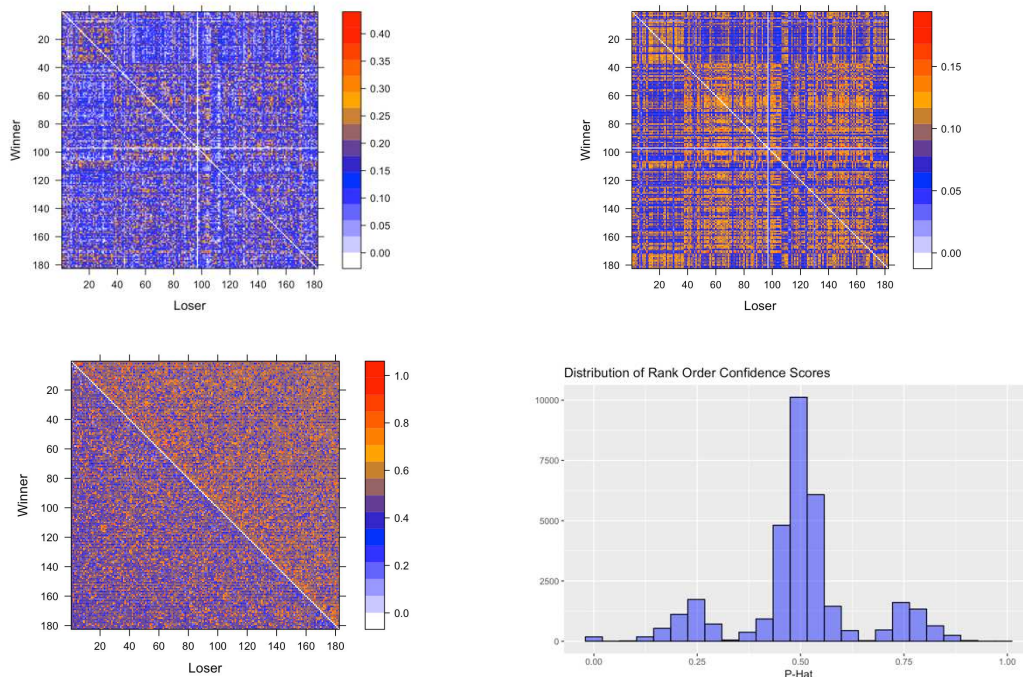
rank.pasture <- simRankOrder(domprob.pasture.3$p.hat, num = 10, kmax = 1000)

```

```
library('ggplot2')
```

```
plotConfmats(domprob.pasture.3$p.hat, ordering = rank.pasture[[1]]$ID, labels = FALSE)
```

```
qplot(as.vector(domprob.pasture.3$p.hat), geom="histogram", xlab = 'P-Hat', ylab = "", main =  
'Distribution of Rank Order Confidence Scores', col="black", fill="blue", alpha=I(.5), bins =  
25)
```



##Rank Order Analysis – Pen

```
library('Perc')
```

```
load("~/Documents/Research/Dairy_Behavior/Projects/Organilac  
Network/OrganilacNetworkWorkspace_PercRuns.RData")
```

```
pendat <- as.conflictmat(networkdat.pen)
```

```
pentrans <- transitivity(pendat)
```

```
pasturetrans$transitivity
```

```
pasturetrans$alpha
```

```
domprob.pen.2 <- conductance(pendat, maxLength = 2)
```

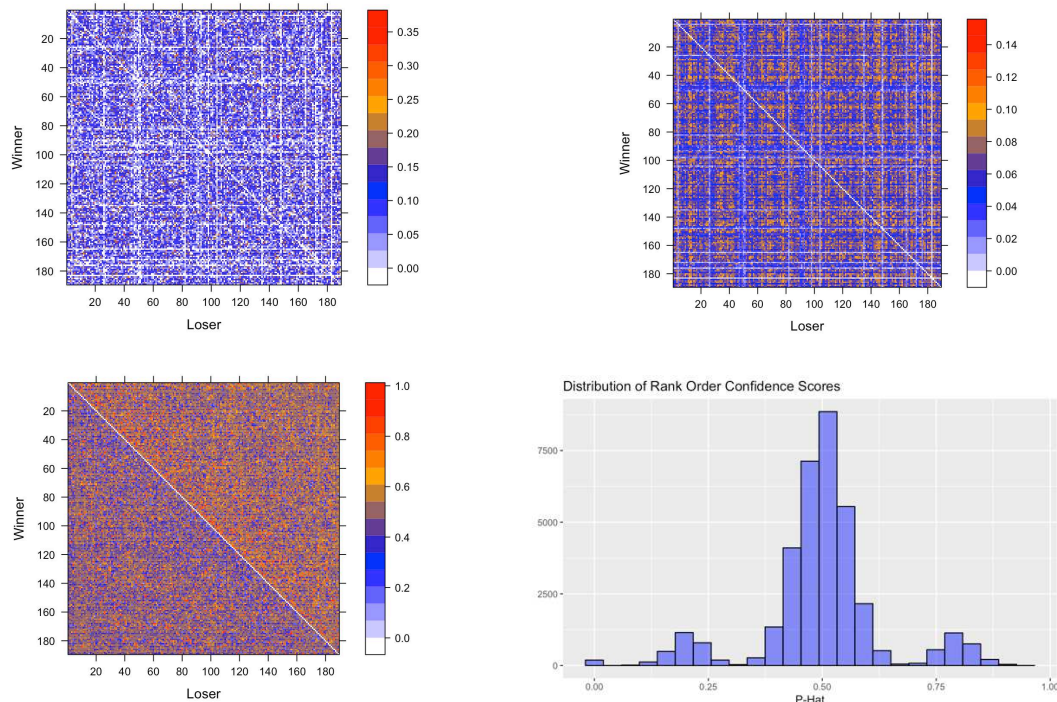
```
domprob.pen.3 <- conductance(pendat, maxLength = 3)
```

```
infogain12 <- domprob.pen.2$imputed.conf - pendat
```

```
plotConfmats(infogain12, ordering = NA, labels = FALSE)
```

```
infogain23 <- domprob.pen.3$imputed.conf - domprob.pen.2$imputed.conf
plotConfmat(infogain23, ordering = NA, labels = FALSE)
```

```
rank.pen <- simRankOrder(domprob.pen.3$p.hat, num = 10, kmax = 1000)
plotConfmat(domprob.pen.3$p.hat, ordering = rank.pen[[1]]$ID, labels = FALSE)
qplot(as.vector(domprob.pen.3$p.hat), geom="histogram", xlab = 'P-Hat', ylab = "", main =
'Distribution of Rank Order Confidence Scores', col=I("black"), fill=I("blue"), alpha=I(.5), bins =
25)
```



##Comparing Rank Order Estimates

```
rank.out.pen <- as.data.frame(rank.pen[[1]])
rank.out.pasture <- as.data.frame(rank.pasture[[1]])

rank.out.pen <- rank.out.pen[rank.out.pen$ID %in% rank.out.pasture$ID ,]
rank.out.pasture <- rank.out.pasture[rank.out.pasture$ID %in% rank.out.pen$ID ,]

rank.out.pen$ranking <- seq(1,nrow(rank.out.pen))
rank.out.pasture$ranking <- seq(1,nrow(rank.out.pasture))

rank.out.all <- merge(rank.out.pen, rank.out.pasture, by = 'ID')
names(rank.out.all)[2:3] <- c('rankpen','rankpasture')
rank.out.all <- rank.out.all[order(rank.out.all$rankpen),]
```

```
cor.test(rank.out.all$rankpen, rank.out.all$rankpasture, alternative = 'greater', method = 'pearson',
conf.level = 0.95)
```

```
cor.test(rank.out.all$rankpen, rank.out.all$rankpasture, alternative = 'greater', method = 'kendall',
conf.level = 0.95)
```

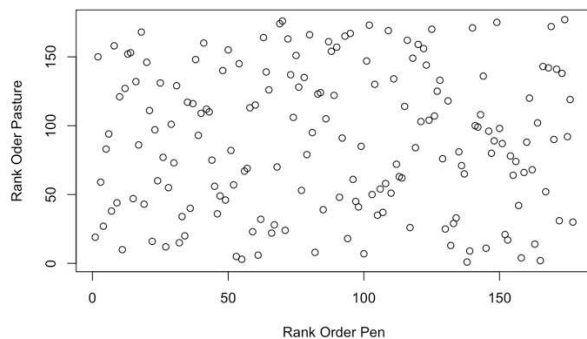
```
plot(rank.out.all$rankpen, rank.out.all$rankpasture, xlab = 'Rank Order Pen', ylab = 'Rank Oder
Pasture')
```

Pearson's product-moment correlation

```
data: rank.out.all$rankpen and rank.out.all$rankpasture
t = 0.05302, df = 175, p-value = 0.4789
alternative hypothesis: true correlation is greater than 0
95 percent confidence interval:
-0.1201055 1.0000000
sample estimates:
cor
0.004007895
```

Kendall's rank correlation tau

```
data: rank.out.all$rankpen and rank.out.all$rankpasture
z = 0.068508, p-value = 0.4727
alternative hypothesis: true tau is greater than 0
sample estimates:
tau
0.003466872
```



Creating Complete Dataset – Overall

```
rankorder <- as.data.frame(rank.pasture[[1]])
names(rankorder)[1] <- 'Cow.ID'
rankorder[,1] <- as.character(rankorder[,1])
rankorder$quantile <- NA
```

```

for (i in 1:nrow(rankorder)){

  rankorder[i,3] <- 1 - (rankorder$ranking[i] - 1)/nrow(rankorder)

}

biomdat_final$Cow.ID <- as.character(biomdat_final$Cow.ID)
rankdat.pasture.full <- merge(biomdat_final, rankorder, by = 'Cow.ID')
rankdat.pasture.full <- rankdat.pasture.full[order(rankdat.pasture.full$ranking),] # sort data in
order of rank

rankorder <- as.data.frame(rank.pen[[1]])
names(rankorder)[1] <- 'Cow.ID'
rankorder[,1] <- as.character(rankorder[,1])
rankorder$quantile <- NA

for (i in 1:nrow(rankorder)){

  rankorder[i,3] <- 1 - (rankorder$ranking[i] - 1)/nrow(rankorder)

}

biomdat_final$Cow.ID <- as.character(biomdat_final$Cow.ID)
rankdat.pen.full <- merge(biomdat_final, rankorder, by = 'Cow.ID')
rankdat.pen.full <- rankdat.pen.full[order(rankdat.pen.full$ranking),] # sort data in order of rank

## Creating Complete Dataset – Left

rankorder <- as.data.frame(rank.pasture[[1]])
names(rankorder)[1] <- 'Cow.ID'
rankorder[,1] <- as.character(rankorder[,1])
rankorder$quantile <- NA

for (i in 1:nrow(rankorder)){

  rankorder[i,3] <- 1 - (rankorder$ranking[i] - 1)/nrow(rankorder)

}

biomdat_final_L$Cow.ID <- as.character(biomdat_final_L$Cow.ID)
rankdat.pasture.L <- merge(biomdat_final_L, rankorder, by = 'Cow.ID')
rankdat.pasture.L <- rankdat.pasture.L[order(rankdat.pasture.L$ranking),] # sort data in order of
rank

```



```

rankorder <- as.data.frame(rank.pen[[1]])
names(rankorder)[1] <- 'Cow.ID'
rankorder[,1] <- as.character(rankorder[,1])
rankorder$quantile <- NA

for (i in 1:nrow(rankorder)){

  rankorder[i,3] <- 1 - (rankorder$ranking[i] - 1)/nrow(rankorder)

}

biomdat_final_L$Cow.ID <- as.character(biomdat_final_L$Cow.ID)
rankdat.pen.L <- merge(biomdat_final_L, rankorder, by = 'Cow.ID')
rankdat.pen.L <- rankdat.pen.L[order(rankdat.pen.L$ranking),] # sort data in order of rank

## Creating Complete Dataset – Right

rankorder <- as.data.frame(rank.pasture[[1]])
names(rankorder)[1] <- 'Cow.ID'
rankorder[,1] <- as.character(rankorder[,1])
rankorder$quantile <- NA

for (i in 1:nrow(rankorder)){

  rankorder[i,3] <- 1 - (rankorder$ranking[i] - 1)/nrow(rankorder)
}
biomdat_final_R$Cow.ID <- as.character(biomdat_final_R$Cow.ID)
rankdat.pasture.R <- merge(biomdat_final_R, rankorder, by = 'Cow.ID')
rankdat.pasture.R <- rankdat.pasture.R[order(rankdat.pasture.R$ranking),] # sort data in order of
rank

rankorder <- as.data.frame(rank.pen[[1]])
names(rankorder)[1] <- 'Cow.ID'
rankorder[,1] <- as.character(rankorder[,1])
rankorder$quantile <- NA

for (i in 1:nrow(rankorder)){

  rankorder[i,3] <- 1 - (rankorder$ranking[i] - 1)/nrow(rankorder)

}

biomdat_final_R$Cow.ID <- as.character(biomdat_final_R$Cow.ID)

```

```

rankdat.pen.R <- merge(biomdat_final_R, rankorder, by = 'Cow.ID')
rankdat.pen.R <- rankdat.pen.R[order(rankdat.pen.R$ranking),] # sort data in order of rank

#####
#Analysis of Biometrics - Kendal Tau by Biometric

tempdat <- rankdat.pasture.full

startbiom <- which(colnames(tempdat) == 'Z1_front_6')
endbiom <- which(colnames(tempdat) == 'FWLP_V3')

kendalldat.all <- data.frame(Biom = c('Age', names(tempdat)[startbiom:endbiom]))
kendalldat.all.r <- data.frame(Biom = c('Age', names(tempdat)[startbiom:endbiom]))

## Overall Pasture

library('knitr')
tempdat <- rankdat.pasture.full

startbiom <- which(colnames(tempdat) == 'Z1_front_6')
endbiom <- which(colnames(tempdat) == 'FWLP_V3')

y <- tempdat$ranking
results.kendal <- c()
results.kendal.r <- c()
x <- as.numeric(tempdat$DOB)
temp <- cor.test(x, y, alternative = 'two.sided', method = 'kendall', conf.level = 0.95)
results.kendal <- c(results.kendal, temp$sp.value)
results.kendal.r <- c(results.kendal.r, temp$estimate)

for (i in startbiom:endbiom){

  x <- tempdat[,i]
  temp <- cor.test(x, y, alternative = 'two.sided', method = 'kendall', conf.level = 0.95)
  results.kendal <- c(results.kendal, temp$sp.value)
  results.kendal.r <- c(results.kendal.r, temp$estimate)

}

kendalldat.all$OverallPasture <- results.kendal
kendalldat.all.r$OverallPasture <- results.kendal.r

```

```

kendal.out <- data.frame(Biometric = c('Age', names(tempdat)[startbiom:endbiom]),
P=results.kendal)
kendal.out <- kendal.out[order(kendal.out$P),]
kable(kendal.out, row.names = FALSE)

## Saving Data:
kendal.dat.all.out <- data.frame(Biometric = kendal.dat.all$Biom)

for (i in 1:nrow(kendal.dat.all)){
  for (j in 2:ncol(kendal.dat.all)){

    kendal.dat.all.out[i,j] <- paste(round(kendal.dat.all.r[i,j],2), ' (p = ', round(kendal.dat.all[i,j],2),
    '), sep = ")
  }
}

names(kendal.dat.all.out) <- names(kendal.dat.all)
write.csv(kendal.dat.all.out, 'KendallTauResultsAll.csv')

#####
#Analysis of Biometrics - Boosted Regression Tree to Rank Order

##Overall Pen

tempdat <- rankdat.pen.full

startbiom <- which(colnames(tempdat) == 'Z1_front_6')
endbiom <- which(colnames(tempdat) == 'FWLP_V3')

x_train <- cbind(as.numeric(tempdat$DOB), tempdat[, startbiom:endbiom])
colnames(x_train)[1] <- 'DOB'
y.train <- tempdat$quantile

k = 5
library('gbm')

set.seed(61916)
bag.out.full.1 <- gbm(y.train ~. , data = x_train, distribution="gaussian", shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=1)

set.seed(61916)
bag.out.full.2 <- gbm(y.train ~. , data = x_train, distribution="gaussian", shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=2)

set.seed(61916)

```

```

bag.out.full.3 <- gbm(y.train ~. , data = x_train, distribution="gaussian", shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=3)

set.seed(61916)
bag.out.full.4 <- gbm(y.train ~. , data = x_train, distribution="gaussian", shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=4)

set.seed(61916)
bag.out.full.5 <- gbm(y.train ~. , data = x_train, distribution="gaussian", shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=5)

temp <- cbind(bag.out.full.1$cv.error, bag.out.full.2$cv.error, bag.out.full.3$cv.error,
bag.out.full.4$cv.error, bag.out.full.5$cv.error)

bag.cv.grid.base <- c()
temp.mse <- c()

for (j in 1:5){

  temp.min <- min(temp[,j])
  bag.cv.grid.base[j] <- min(which(temp[,j] < (temp.min + 0.01*temp.min)))
  if (bag.cv.grid.base[j] < 5){
    bag.cv.grid.base[j] = 5 # minimum tree depth of 5
  }
  temp.mse[j] <- temp[bag.cv.grid.base[j],j]
  # find index within 1% of the absolute minima

}

temp.min <- min(temp.mse)
g <- min(which(temp.mse < (temp.min + 0.01 * temp.min))) # best tree complexity
g
B <- bag.cv.grid.base[g] # best tree depth
B

set.seed(61916)
bag.out.full.final <- gbm(y.train ~. , data = x_train, distribution="gaussian", shrinkage = 0.001,
n.trees=B, cv.folds = k, interaction.depth=g)

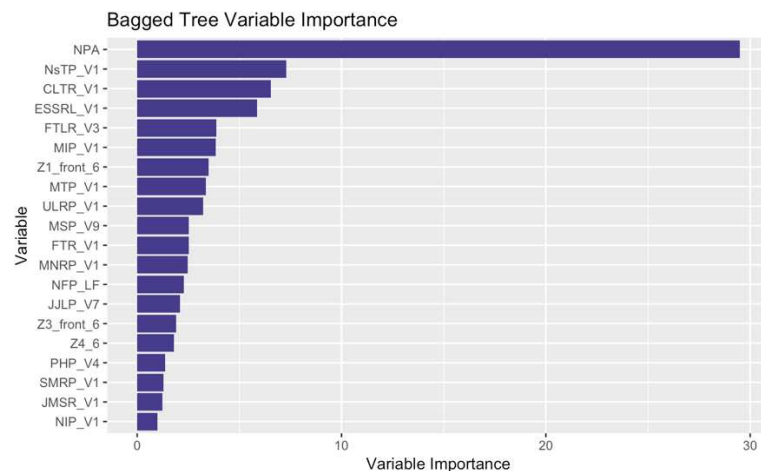
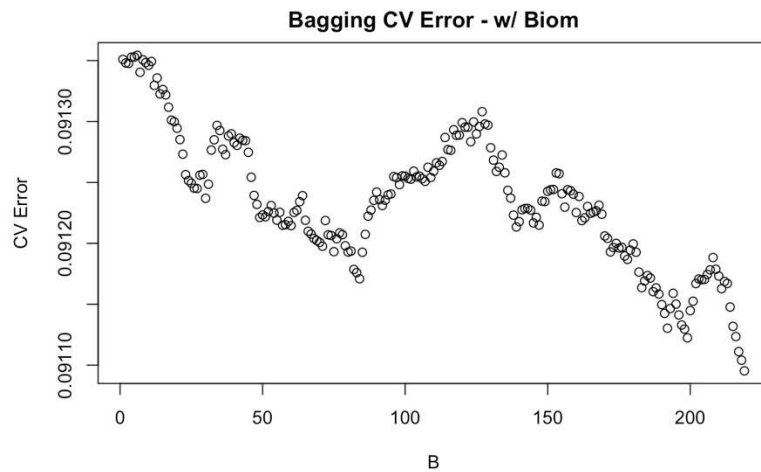
temp <- summary(bag.out.full.final)
temp <- data.frame(var = temp$var, varimp = temp$rel.inf)
temp <- temp[order(temp$varimp, decreasing = T),]
temp <- temp[1:20,]

```

```
ggplot(data=temp, aes(x=reorder(var, varimp), y=varimp)) + geom_bar(stat="identity" , fill =
'slateblue4') + coord_flip() + xlab('Variable') + ylab('Variable Importance') + ggtitle('Boosted
Tree Variable Importance')
```

```
plot(1:B, bag.out.full.final$cv.error, xlab = 'B', ylab = 'CV Error', main = 'Bagging CV Error - w/
Biom')
```

```
rsq.out <- cor(y.train, bag.out.full.final$fit)^2
rsq.out
0.4918441
```



```
#####
```

```
#Network Analyses
```

```
##Construct graph and basic features
```

```
library('igraph')
library('threejs')
```

```

library('knitr')

g.pasture <- graph_from_adjacency_matrix(networkdat.pasture, mode = 'directed', weighted =
TRUE)
is.directed(g.pasture)
is.weighted(g.pasture)
#E(g)
#gorder(g)
#vertex_attr(g)

get_diameter(g.pasture)
farthest_vertices(g.pasture)

reciprocity(g.pasture)

[1] TRUE
[1] TRUE
+ 5/182 vertices, named, from 69e01b9:
[1] 1114 55516 2134 1421 29868
$vertices
+ 2/182 vertices, named, from 69e01b9:
[1] 1114 29868

$distance
[1] 4

[1] 0.3660326

g.pen <- graph_from_adjacency_matrix(networkdat.pen, mode = 'directed', weighted = TRUE)
is.directed(g.pen)
is.weighted(g.pen)
#E(g)
#gorder(g)
#vertex_attr(g)

get_diameter(g.pen)
farthest_vertices(g.pen)

reciprocity(g.pen)

[1] TRUE
[1] TRUE
+ 6/189 vertices, named, from aeb0cb6:
[1] 1424 65346 98737 16739 26901 63106
$vertices
+ 2/189 vertices, named, from aeb0cb6:
[1] 1424 63106

```



```
$distance  
[1] 6
```

```
[1] 0.2740448
```

##Measures of Vertex Importance – Pasture

```
vert.out.between <- betweenness(g.pasture, directed = TRUE)  
qplot(vert.out.between, geom="histogram", xlab = 'Betweenness', ylab = "", main = 'Distribution  
of Betweenness Values', col=I("black"), fill=I("blue"), alpha=I(.5), bins = 25)
```

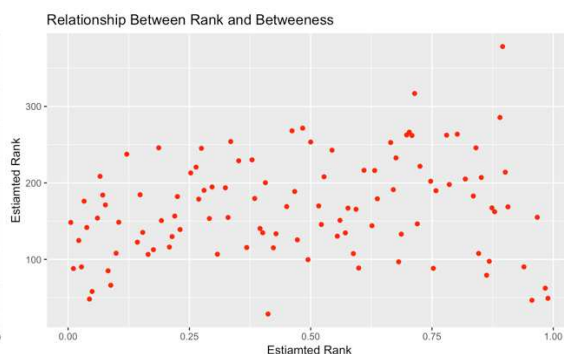
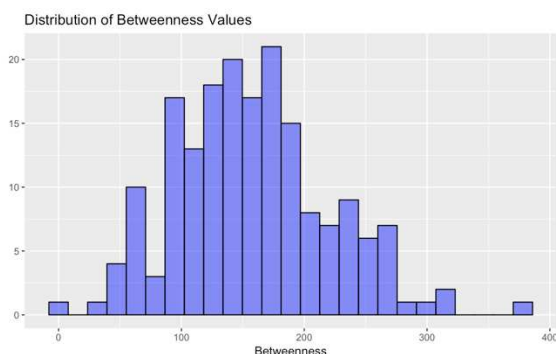
```
tempdat <- data.frame(Cow.ID = names(vert.out.between), betw = vert.out.between)  
networkdat.pasture.full <- merge(rankdat.pasture.full, tempdat, by = 'Cow.ID')  
networkdat.pasture.L <- merge(rankdat.pasture.L, tempdat, by = 'Cow.ID')  
networkdat.pasture.R <- merge(rankdat.pasture.R, tempdat, by = 'Cow.ID')
```

```
ggplot(networkdat.pasture.full, aes(x=quantile, y=betw)) + geom_point(color = 'red') + ggtitle  
( 'Relationship Between Rank and Betweenness') + ylab('Estiamted Rank') + xlab('Estiamted  
Rank')
```

```
vert.out.pagerank <- page_rank(g.pasture, directed = TRUE)  
qplot(vert.out.pagerank$vector , geom="histogram", xlab = 'Page Rank', ylab = "", main =  
'Distribution of Page Rank Values', col=I("black"), fill=I("blue"), alpha=I(.5), bins = 25)
```

```
tempdat <- data.frame(Cow.ID = names(vert.out.pagerank$vector), pr =  
vert.out.pagerank$vector)  
networkdat.pasture.full <- merge(networkdat.pasture.full, tempdat, by = 'Cow.ID')  
networkdat.pasture.L <- merge(networkdat.pasture.L, tempdat, by = 'Cow.ID')  
networkdat.pasture.R <- merge(networkdat.pasture.R, tempdat, by = 'Cow.ID')
```

```
ggplot(networkdat.pasture.full, aes(x=quantile, y=pr)) + geom_point(color = 'red') + ggtitle  
( 'Relationship Between Rank and Page Rank') + ylab('Estiamted Rank') + xlab('Estiamted Rank')
```



```
## Predicting Bilateral Biometrics
```

```
tempdat <- networkdat.pasture.full  
startbiom <- which(colnames(tempdat) == 'Z1_front_6')  
endbiom <- which(colnames(tempdat) == 'FWLP_V3')
```

```
x_train <- cbind(as.numeric(tempdat$DOB), tempdat$quantile, tempdat[, startbiom:endbiom])  
colnames(x_train)[1:2] <- c('DOB', 'Rank')  
y.train <- tempdat$betw
```

```
library('gbm')
```

```
set.seed(61916)  
bag.out.full.1 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,  
n.trees=7000, cv.folds = k, interaction.depth=1)
```

```
set.seed(61916)  
bag.out.full.2 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,  
n.trees=7000, cv.folds = k, interaction.depth=2)
```

```
set.seed(61916)  
bag.out.full.3 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,  
n.trees=7000, cv.folds = k, interaction.depth=3)
```

```
set.seed(61916)  
bag.out.full.4 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,  
n.trees=7000, cv.folds = k, interaction.depth=4)
```

```
set.seed(61916)  
bag.out.full.5 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,  
n.trees=7000, cv.folds = k, interaction.depth=5)
```

```
temp <- cbind(bag.out.full.1$cv.error, bag.out.full.2$cv.error, bag.out.full.3$cv.error,  
bag.out.full.4$cv.error, bag.out.full.5$cv.error)
```

```
bag.cv.grid.base <- c()  
temp.mse <- c()
```

```
for (j in 1:5){
```

```
  temp.min <- min(temp[,j])  
  bag.cv.grid.base[j] <- min(which(temp[,j] < (temp.min + 0.01*temp.min)))  
  if (bag.cv.grid.base[j] < 5){  
    bag.cv.grid.base[j] = 5 # minimum tree depth of 5  
  }
```

```

temp.mse[j] <- temp[bag.cv.grid.base[j],j]
# find index within 1% of the absolute minima

}

temp.min <- min(temp.mse)
g <- min(which(temp.mse < (temp.min + 0.01 * temp.min))) # best tree complexity
g
B <- bag.cv.grid.base[g] # best tree depth
B

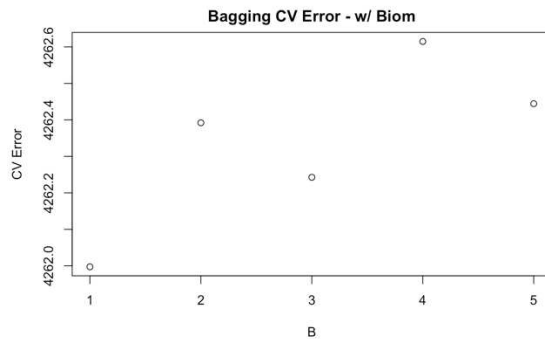
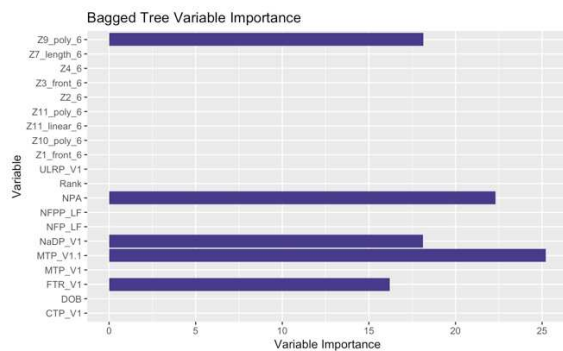
set.seed(61916)
bag.out.full.final <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,
n.trees=B, cv.folds = k, interaction.depth=g)

temp <- summary(bag.out.full.final)
temp <- data.frame(var = temp$var, varimp = temp$rel.inf)
temp <- temp[order(temp$varimp, decreasing = T),]
temp <- temp[1:20,]
ggplot(data=temp, aes(x=var, y=varimp)) + geom_bar(stat="identity" , fill = 'slateblue4') +
coord_flip() + xlab('Variable') + ylab('Variable Importance') + ggtitle('Boosted Tree Variable
Importance')

plot(1:B, bag.out.full.final$cv.error, xlab = 'B', ylab = 'CV Error', main = 'Bagging CV Error - w/
Biom')

rsq.out <- cor(y.train, bag.out.full.final$fit)^2
rsq.out
0.08267378

```



##Measures of Vertex Importance – Pen

```

vert.out.between <- betweenness(g.pen, directed = TRUE)

```

```
qplot(vert.out.between, geom="histogram", xlab = 'Betweenness', ylab = "", main = 'Distribution
of Betweenness Values', col=I("black"), fill=I("blue"), alpha=I(.5), bins = 25)
```

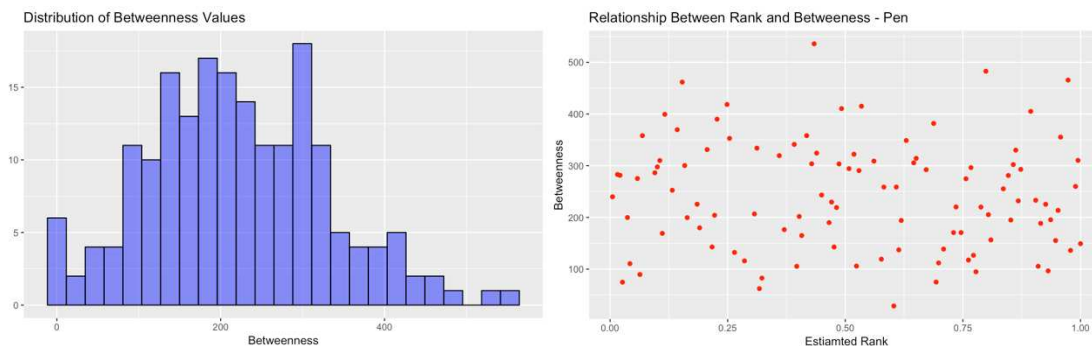
```
tempdat <- data.frame(Cow.ID = names(vert.out.between), betw = vert.out.between)
networkdat.pen.full <- merge(rankdat.pen.full, tempdat, by = 'Cow.ID')
networkdat.pen.L <- merge(rankdat.pen.L, tempdat, by = 'Cow.ID')
networkdat.pen.R <- merge(rankdat.pen.R, tempdat, by = 'Cow.ID')
```

```
ggplot(networkdat.pen.full, aes(x=quantile, y=betw)) + geom_point(color = 'red') + ggtitle
('Relationship Between Rank and Betweenness - Pen') + ylab('Betweenness') + xlab('Estiamted
Rank')
```

```
vert.out.pagerank <- page_rank(g.pen, directed = TRUE)
qplot(vert.out.pagerank$vector, geom="histogram", xlab = 'Page Rank', ylab = "", main =
'Distribution of Page Rank Values', col=I("black"), fill=I("blue"), alpha=I(.5), bins = 25)
```

```
tempdat <- data.frame(Cow.ID = names(vert.out.pagerank$vector), pr =
vert.out.pagerank$vector)
networkdat.pen.full <- merge(networkdat.pen.full, tempdat, by = 'Cow.ID')
networkdat.pen.L <- merge(networkdat.pen.L, tempdat, by = 'Cow.ID')
networkdat.pen.R <- merge(networkdat.pen.R, tempdat, by = 'Cow.ID')
```

```
ggplot(networkdat.pen.full, aes(x=quantile, y=pr)) + geom_point(color = 'red') + ggtitle
('Relationship Between Rank and Page Rank') + ylab('Estiamted Rank') + xlab('Estiamted Rank')
```



Predicting Bilateral Biometrics

```
tempdat <- networkdat.pen.all
startbiom <- which(colnames(tempdat) == 'Z1_front_6')
endbiom <- which(colnames(tempdat) == 'FWLP_V3')
```

```
x_train <- cbind(as.numeric(tempdat$DOB), tempdat$quantile, tempdat[, startbiom:endbiom])
```

```

colnames(x_train)[1:2] <- c('DOB', 'Rank')
y.train <- tempdat$betw

library('gbm')

set.seed(61916)
bag.out.full.1 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=1)

set.seed(61916)
bag.out.full.2 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=2)

set.seed(61916)
bag.out.full.3 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=3)

set.seed(61916)
bag.out.full.4 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=4)

set.seed(61916)
bag.out.full.5 <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,
n.trees=7000, cv.folds = k, interaction.depth=5)

temp <- cbind(bag.out.full.1$cv.error, bag.out.full.2$cv.error, bag.out.full.3$cv.error,
bag.out.full.4$cv.error, bag.out.full.5$cv.error)

bag.cv.grid.base <- c()
temp.mse <- c()

for (j in 1:5){

  temp.min <- min(temp[,j])
  bag.cv.grid.base[j] <- min(which(temp[,j] < (temp.min + 0.01*temp.min)))
  if (bag.cv.grid.base[j] < 5){
    bag.cv.grid.base[j] = 5 # minimum tree depth of 5
  }
  temp.mse[j] <- temp[bag.cv.grid.base[j],j]
  # find index within 1% of the absolute minima

}

temp.min <- min(temp.mse)
g <- min(which(temp.mse < (temp.min + 0.01 * temp.min))) # best tree complexity
g

```

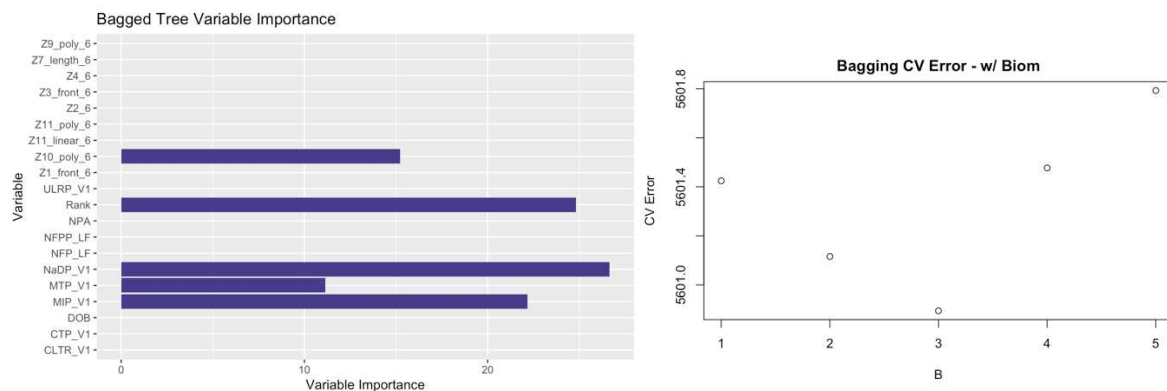
```
B <- bag.cv.grid.base[g] # best tree depth
B
```

```
set.seed(61916)
bag.out.full.final <- gbm(y.train ~. , data = x_train, distribution='gaussian', shrinkage = 0.001,
n.trees=B, cv.folds = k, interaction.depth=g)
```

```
temp <- summary(bag.out.full.final)
temp <- data.frame(var = temp$var, varimp = temp$rel.inf)
temp <- temp[order(temp$varimp, decreasing = T),]
temp <- temp[1:20,]
ggplot(data=temp, aes(x=var, y=varimp)) + geom_bar(stat="identity" , fill = 'slateblue4') +
coord_flip() + xlab('Variable') + ylab('Variable Importance') + ggtitle('Boosted Tree Variable
Importance')
```

```
plot(1:B, bag.out.full.final$cv.error, xlab = 'B', ylab = 'CV Error', main = 'Bagging CV Error - w/
Biom')
```

```
rsq.out <- cor(y.train, bag.out.full.final$fit)^2
rsq.out
0.2161326
```



##Assortativity Analysis – Pasture

```
gsub.pasture <- induced_subgraph(g.pasture, biomdat_final$Cow.ID)
```

```
B = 5000
assort.out <- c()
p.out <- c()
for (i in c(2, 4:ncol(tempdat))) {
```

```
  t <- assortativity(gsub.pasture, biomdat_final[,i], directed = T)
  assort.out <- c(assort.out, t)
```

```

results <- vector('list', B)
for(j in 1:B){
  results[[j]] <- assortativity(gsub.pasture, sample(biomdat_final[,i]), directed = T)
}
results <- abs(unlist(results))
p <- sum(results > abs(t))/length(results)
p.out <- c(p.out, p)
}

assort.out <- data.frame(cow = colnames(biomdat_final)[-c(1,3)] , assort = round(assort.out,3) ,
pval = round(p.out,2), stringsAsFactors = F)
names(assort.out) <- c('Biometric', 'Assortativity', 'P-Value')
assort.out$Biometric[1] <- 'Age'

assort.out <- assort.out[order(assort.out$Assortativity),]
kable(assort.out, row.names = F)
assort.out.pasture <- assort.out

##Assortativity Analysis – Pasture

tempdat <- biomdat_final[-which(!(biomdat_final$Cow.ID %in% names(V(g.pen)))),]
gsub.pen <- induced_subgraph(g.pen, tempdat$Cow.ID) # need to remove a few biom cows that
were excluded from pen data

B = 5000
assort.out <- c()
p.out <- c()
for (i in c(2, 4:ncol(tempdat))) {

  t <- assortativity(gsub.pen, tempdat[,i], directed = T)
  assort.out <- c(assort.out, t)

  results <- vector('list', B)
  for(j in 1:B){
    results[[j]] <- assortativity(gsub.pen, sample(tempdat[,i]), directed = T)
  }
  results <- abs(unlist(results))
  p <- sum(results > abs(t))/length(results)
  p.out <- c(p.out, p)
}

```



```

assort.out <- data.frame(cow = colnames(tempdat)[-c(1,3)] , assort = round(assort.out,3), pval =
round(p.out,2), stringsAsFactors = F)
names(assort.out) <- c('Biometric', 'Assortativity', 'P-Value')
assort.out$Biometric[1] <- 'Age'

assort.out <- assort.out[order(assort.out$Assortativity),]
kable(assort.out, row.names = F)
assort.out.pen <- assort.out

assort.out.all <- merge(assort.out.pasture, assort.out.pen, by = 'Biometric')
names(assort.out.all)[c(3,5)] <- c('P-Value','P-Value')
write.csv(assort.out.all, 'Results_Assortativity.csv')

```