Sensitivity/Specificity Analyses - Canine Leishmaniosis

Marie Ozanne

March 3, 2020

Exploratory Analyses

```
##           DPP
## PCR        Negative Positive
##   Negative      743       16
##   Positive        1       11
```

Models

Angela's paper (Toepp et al., 2019, https://doi.org/10.1371/journal.pntd.0007058) uses logistic regression, with age, sex, and variables that have to do with diagnostic tests as explanatory variables. They are something like this:

Model A 1: $logit(\pi_k) = \beta_0 + \beta_1 Age_k + \beta_2 Sex_k + \beta_3 Y_k$, where Y_k is diagnostically positive (as defined in Model 1 below), but for the mom and π_k is the probability of disease for individual k

Model A 2: $logit(\pi_k) = \beta_0 + \beta_1 Age_k + \beta_2 Sex_k + \beta_3 T_{1k} + \beta_4 T_{2k}$, where T_{jk} is the result for Test j (as defined in Model 1 below), but for the mom and π_k is the probability of disease for individual k

Note, these models were fit with a log link function, presumably so that relative risks could be recovered?

We plan to evaluate similar models for our data and to then incorporate sensitivity and specificity of the tests into these models. Then we will compare the model performance to that of other methods. Hopefully we will see an improvement/some details that we miss when we do not include the sensitivity and specificity for the tests.

In all these models, we will assume that the observations are independent.

Model 1:

Data Model

$$T_{1k}|D_k \sim Bernoulli(\pi_{1k})$$

where $\pi_{1k} = P(T_{1k} = 1|D_k) = D_k \times \underbrace{P(T_{1k} = 1|D_k = 1)}_{sensitivity} + (1 - D_k) \times \underbrace{(1 - P(T_{1k} = 0|D_k = 0))}_{1-specificity}$.

$$T_{2k}|D_k \sim Bernoulli(\pi_{2k})$$

where $\pi_{2k} = P(T_{2k} = 1|D_k) = D_k \times \underbrace{P(T_{2k} = 1|D_k = 1)}_{sensitivity} + (1 - D_k) \times \underbrace{(1 - P(T_{2k} = 0|D_k = 0))}_{1-specificity}$.

Process Model

$$D_k \sim Bernoulli(\delta_k)$$

where $\delta_k = logit^{-1}(logit(\rho) + \mathbf{x}_k^T \boldsymbol{\beta} + \epsilon_k)$, and ϵ_k is a random effect for individual k.

Prior Model

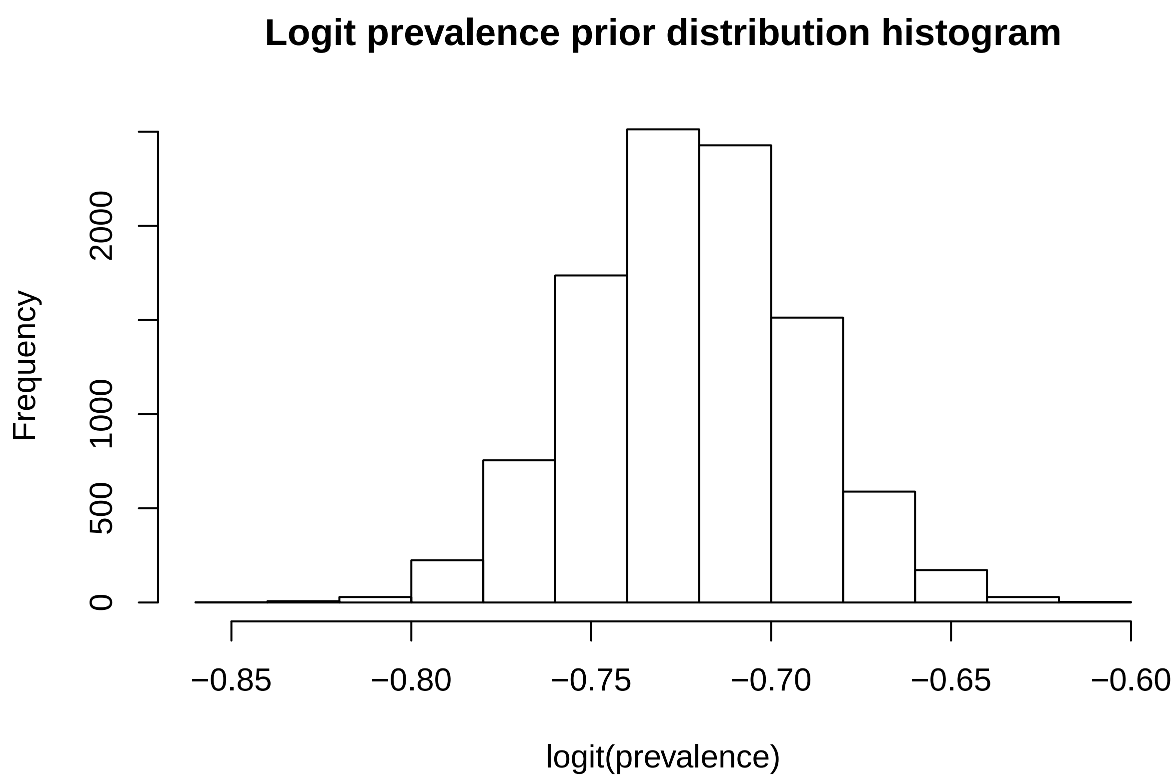
Prevalence

$$logit(\rho) \sim Normal(\mu_\rho, \tau^2)$$

where $\mu_\rho = logit^{-1}(\rho^*)$ and $\tau^2 = \ldots$

We have a range for the prevalence of (0.05, 0.10). This corresponds to a range of (-0.7497, -0.6972) on the logit scale.

```
hist(rnorm(10000, mean = log(0.075)/(1-log(0.075)), sd = 0.03),
     main="Logit prevalence prior distribution histogram",
     xlab="logit(prevalence)")
```



Note, the prevalence in this population may be higher - these are exposed dogs in the United States in our hunting hound population. We may want to change this, but we also want to generalize this to the larger canine population - to Brazil, if possible.

Linear predictors

The regression parameters are $\boldsymbol{\beta} = (\beta_{age}, \beta_{sex}, \beta_{age*sex})^T$;

$$\boldsymbol{\beta} \sim Normal(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$$

where $\boldsymbol{\mu}_\beta = \mathbf{0}$ and $\boldsymbol{\Sigma}_\beta = \mathbf{I}$ in our code.

Random intercept

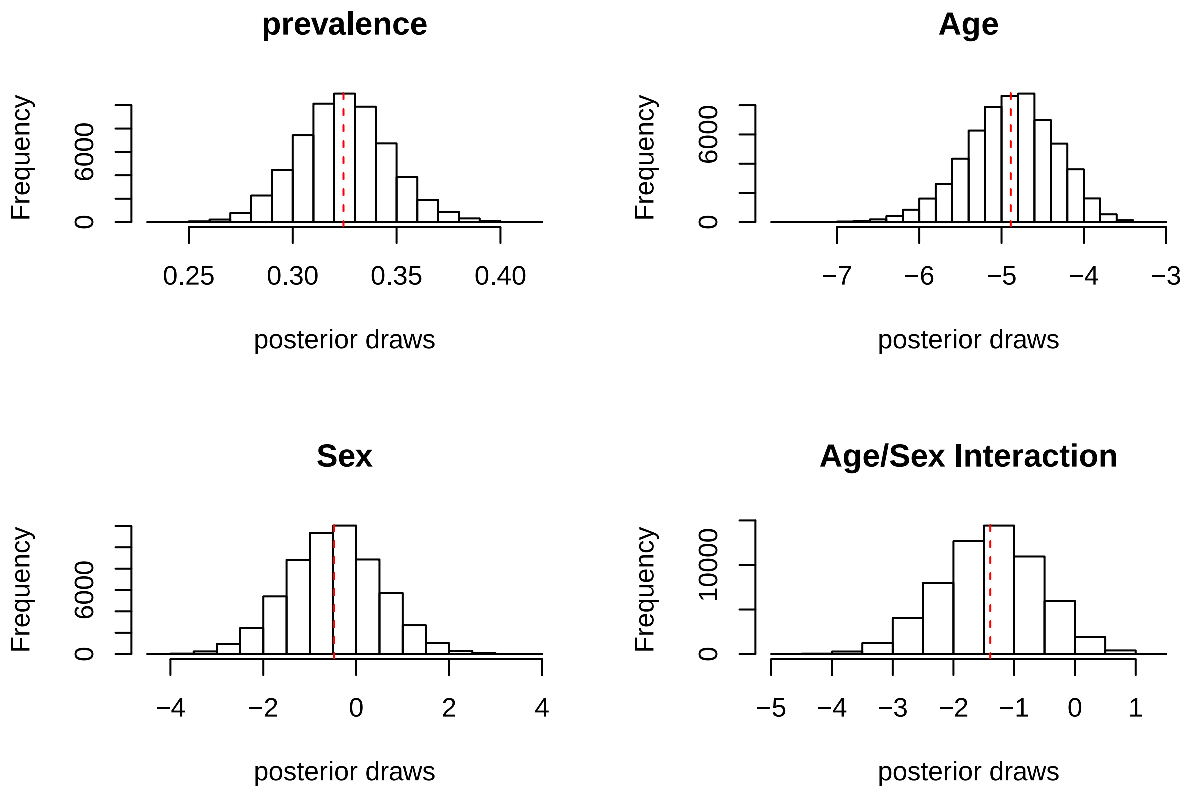
$$\boldsymbol{\epsilon} \sim Normal(0, 5000)$$

```
## ranges of sensitivities and specificities
sens.pcr.range <- c(0.839, 0.990)
sens.dpp.range <- c(0.832, 0.930)
spec.pcr.range <- c(0.871, 0.970)
spec.dpp.range <- c(0.682, 0.951)

sens.pcr <- mean(sens.pcr.range)
sens.dpp <- mean(sens.dpp.range)
spec.pcr <- mean(spec.pcr.range)
spec.dpp <- mean(spec.dpp.range)

## range of prevalence for visceral leishmaniasis
prev.range <- c(0.05,0.10)
prev <- mean(prev.range)
```

OpenBUGS Model 1 Implementation

OpenBUGS Model 1 Posterior Distributions

```
par(mfrow=c(2,2))
## Graphial summaries of posterior distributions
hist(exp(model1_df$lpi)/(1+exp(model1_df$lpi)), main="prevalence", xlab="posterior draws")
abline(v=mean(exp(model1_df$lpi)/(1+exp(model1_df$lpi))), lty="dashed", col="red")

hist(model1_df$b1, main="Age", xlab="posterior draws")
abline(v=mean(model1_df$b1), lty="dashed", col="red")
hist(model1_df$b2, main="Sex", xlab="posterior draws")
abline(v=mean(model1_df$b2), lty="dashed", col="red")
hist(model1_df$b3, main="Age/Sex Interaction", xlab="posterior draws")
abline(v=mean(model1_df$b3), lty="dashed", col="red")
```



OpenBUGS Model 1 Disease State Classification

```
## Set up storage for model results
pred_df_m1 <- data.frame(obs=1:nind,
                         D=rep(NA,nind), ## average estimate
                         SD=rep(NA,nind),
                         LB=rep(NA,nind), ## 2.5th percentile
                         UB=rep(NA,nind), ## 97.5th percentile
                         model_assignment=rep(NA,nind),
                         Clinical_status=ss_data2$ClinicalStatus,
                         Diagnostic_status=ss_data2$Diagnostically_positive)

## Calculate probabilities of compartment membership for each posterior draw
pred_df_m1$D <- apply(model1_df[,grep("D", names(model1_df))], 2, mean)
pred_df_m1$SD <- apply(model1_df[,grep("D", names(model1_df))], 2, sd)
pred_df_m1$LB <- apply(model1_df[,grep("D", names(model1_df))], 2,
                       quantile, probs=0.025)
pred_df_m1$UB <- apply(model1_df[,grep("D", names(model1_df))], 2,
                       quantile, probs=0.975)


summary(pred_df_m1)
```

```
##        obs                D                    SD                  LB
```

```
##  Min.   :  1.0   Min.   :0.000e+00   Min.   :0.000000   Min.   :0
##  1st Qu.:193.5   1st Qu.:0.000e+00   1st Qu.:0.000000   1st Qu.:0
##  Median :386.0   Median :0.000e+00   Median :0.000000   Median :0
##  Mean   :386.0   Mean   :1.358e-03   Mean   :0.010641   Mean   :0
##  3rd Qu.:578.5   3rd Qu.:6.667e-05   3rd Qu.:0.008165   3rd Qu.:0
##  Max.   :771.0   Max.   :2.726e-01   Max.   :0.445283   Max.   :0
##        UB           model_assignment Clinical_status Diagnostic_status
##  Min.   :0.000000   Mode:logical     A:  0           Negative:743
##  1st Qu.:0.000000   NA's:771         N:736           Positive: 28
##  Median :0.000000                    S: 35
##  Mean   :0.009079
##  3rd Qu.:0.000000
##  Max.   :1.000000
```

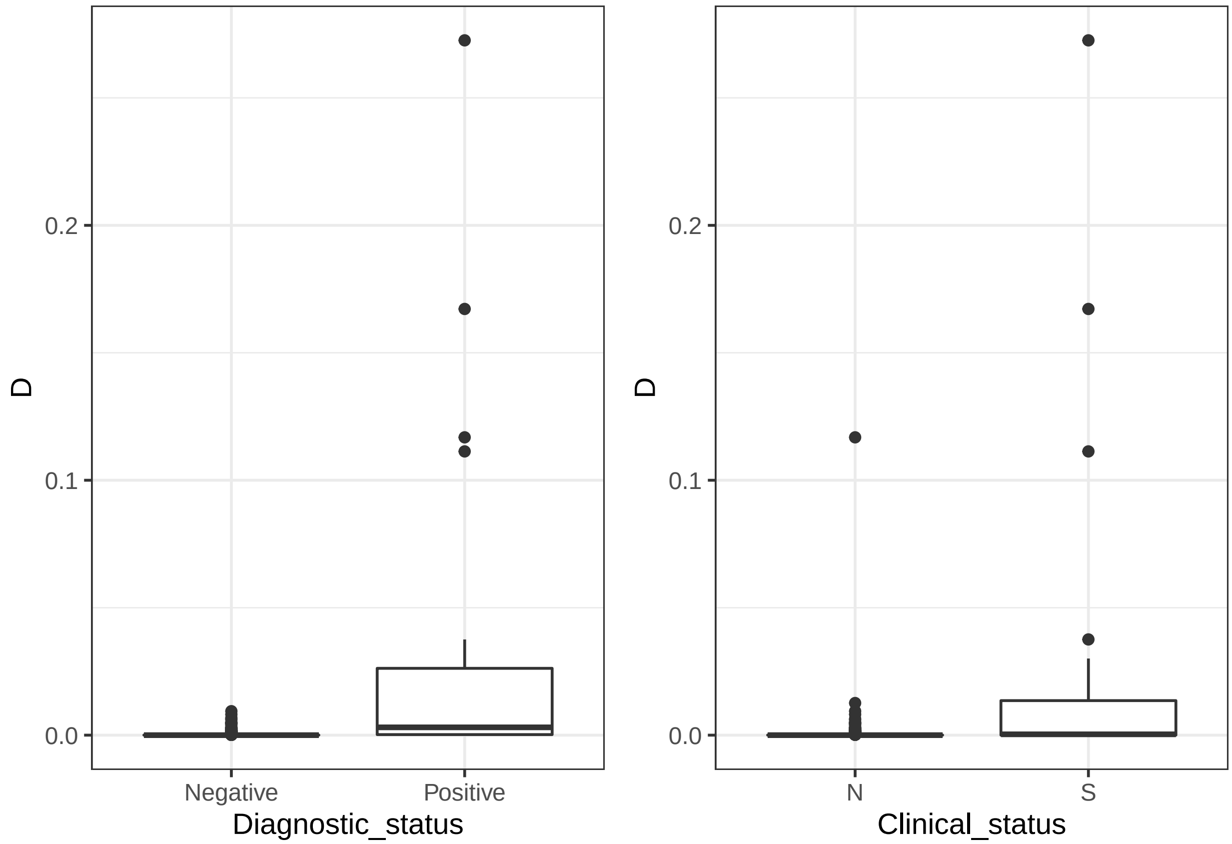
```r
## Apply a cut off of point estimate of 0.5; if pi.D > 0.5, classify as S (symptomatic), otherwise as N
## Summarize in a table (clinical status versus diagnostic status)
table(pred_df_m1[pred_df_m1$D > 0.5,]$Clinical_status,
      pred_df_m1[pred_df_m1$D > 0.5,]$Diagnostic_status)
```

```
##
##     Negative Positive
##   A        0        0
##   N        0        0
##   S        0        0
```

```r
## Print summary table of clinical status versus diagnostic status from the original data
table(pred_df_m1$Clinical_status, pred_df_m1$Diagnostic_status)
```

```
##
##     Negative Positive
##   A        0        0
##   N      728        8
##   S       15       20
```

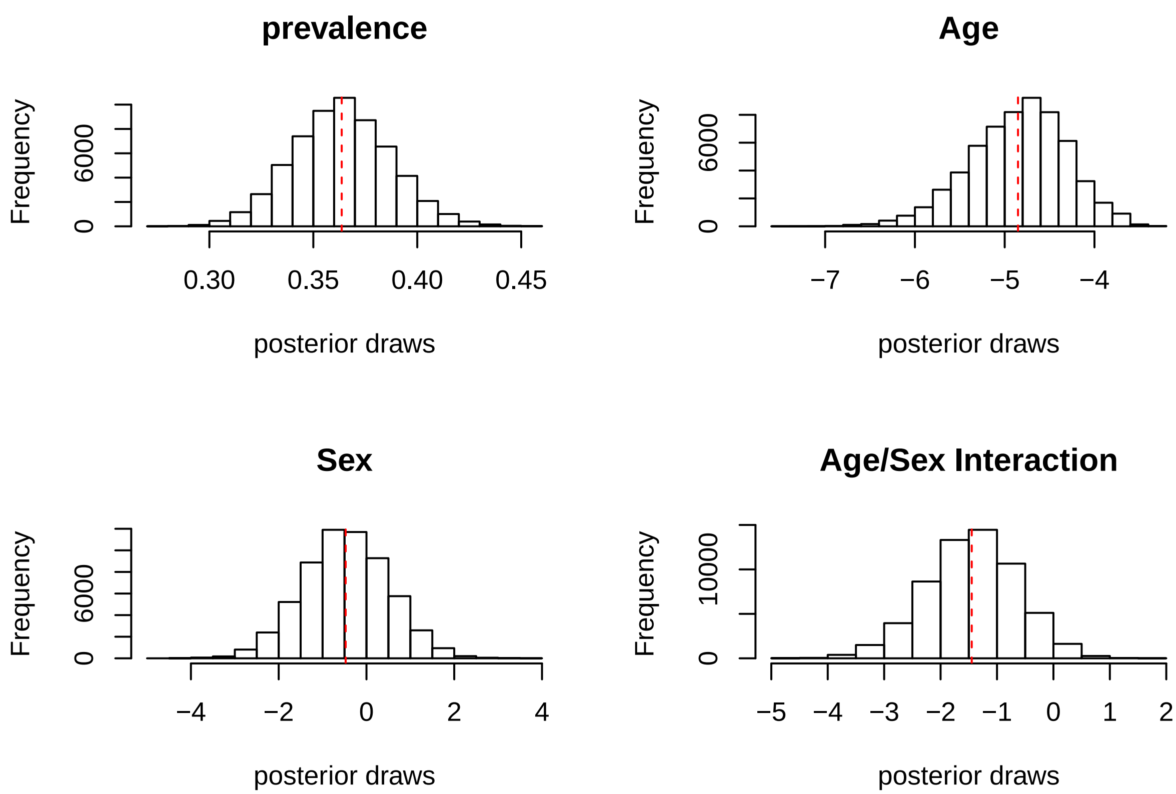
```r
## boxplots
p1 <- (ggplot(data=pred_df_m1, aes(x=Diagnostic_status, y=D))
       + geom_boxplot()
       + theme_bw())
p2 <- (ggplot(data=pred_df_m1, aes(x=Clinical_status, y=D))
       + geom_boxplot()
       + theme_bw())

ggarrange(p1,p2, nrow=1)
```



OpenBUGS Model 1 Implementation - Informative Prior Prevalence

OpenBUGS Model 1 Posterior Distributions - Informative Prior Prevalence

```r
par(mfrow=c(2,2))
## Graphial summaries of posterior distributions
hist(exp(model1b_df$lpi)/(1+exp(model1b_df$lpi)), main="prevalence", xlab="posterior draws")
abline(v=mean(exp(model1b_df$lpi)/(1+exp(model1b_df$lpi))), lty="dashed", col="red")

hist(model1b_df$b1, main="Age", xlab="posterior draws")
abline(v=mean(model1b_df$b1), lty="dashed", col="red")
hist(model1b_df$b2, main="Sex", xlab="posterior draws")
abline(v=mean(model1b_df$b2), lty="dashed", col="red")
hist(model1b_df$b3, main="Age/Sex Interaction", xlab="posterior draws")
abline(v=mean(model1b_df$b3), lty="dashed", col="red")
```



OpenBUGS Model 1 Disease State Classification - Informative Prior Prevalence

```
## Set up storage for model results
pred_df_m1b <- data.frame(obs=1:nind,
                          D=rep(NA,nind), ## average estimate
                          SD=rep(NA,nind),
                          LB=rep(NA,nind), ## 2.5th percentile
                          UB=rep(NA,nind), ## 97.5th percentile
                          model_assignment=rep(NA,nind),
                          Clinical_status=ss_data2$ClinicalStatus,
                          Diagnostic_status=ss_data2$Diagnostically_positive)

## Calculate probabilities of compartment membership for each posterior draw
pred_df_m1b$D <- apply(model1b_df[,grep("D", names(model1b_df))], 2, mean)
pred_df_m1b$SD <- apply(model1b_df[,grep("D", names(model1b_df))], 2, sd)
pred_df_m1b$LB <- apply(model1b_df[,grep("D", names(model1b_df))], 2,
                        quantile, probs=0.025)
pred_df_m1b$UB <- apply(model1b_df[,grep("D", names(model1b_df))], 2,
                        quantile, probs=0.975)


summary(pred_df_m1b)
```

```
##       obs                  D                    SD                   LB
```

```
##  Min.   :  1.0   Min.   :0.0000000   Min.   :0.000000   Min.   :0
##  1st Qu.:193.5   1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0
##  Median :386.0   Median :0.0000000   Median :0.000000   Median :0
##  Mean   :386.0   Mean   :0.0015722   Mean   :0.011201   Mean   :0
##  3rd Qu.:578.5   3rd Qu.:0.0000917   3rd Qu.:0.009564   3rd Qu.:0
##  Max.   :771.0   Max.   :0.3305333   Max.   :0.470409   Max.   :0
##        UB           model_assignment Clinical_status Diagnostic_status
##  Min.   :0.000000   Mode:logical     A:  0           Negative:743
##  1st Qu.:0.000000   NA's:771         N:736           Positive: 28
##  Median :0.000000                    S: 35
##  Mean   :0.009079
##  3rd Qu.:0.000000
##  Max.   :1.000000
```

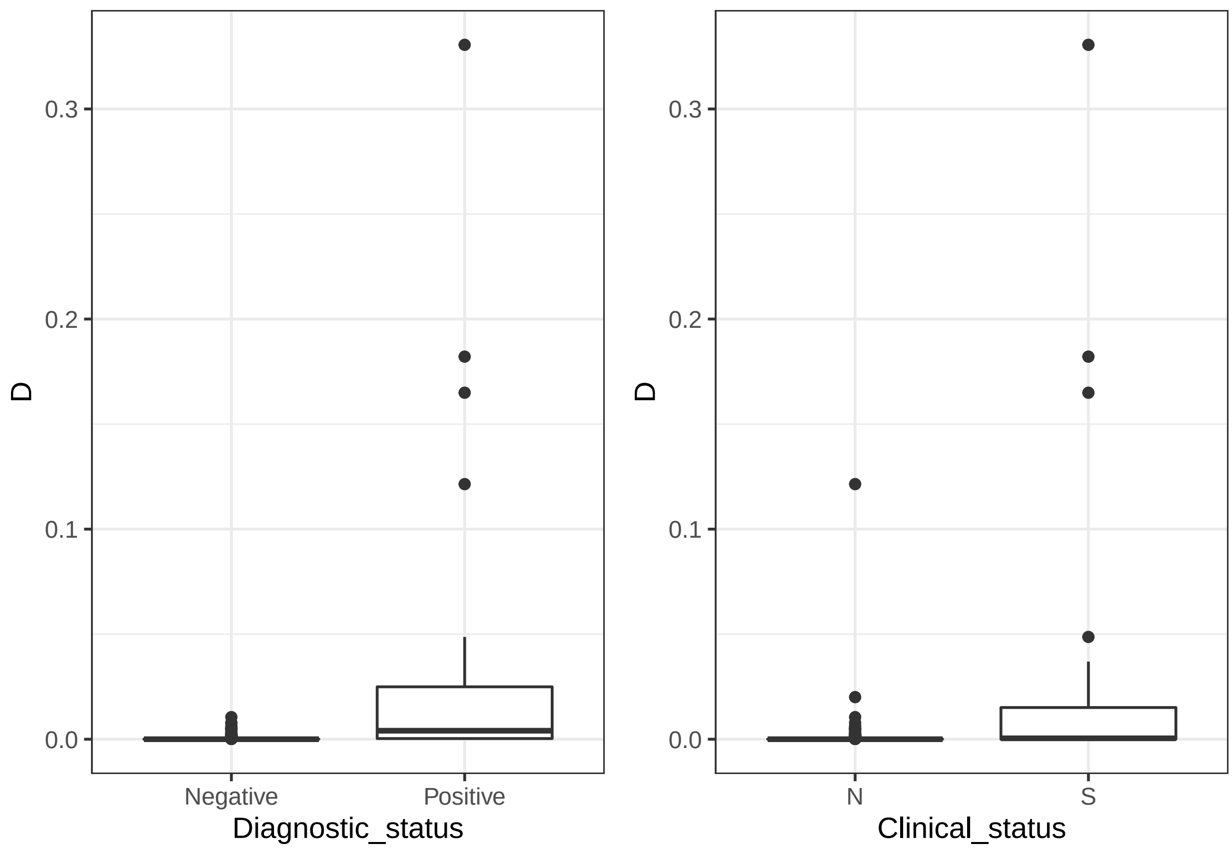
```
## Apply a cut off of point estimate of 0.5; if pi.D > 0.5, classify as S (symptomatic), otherwise as N
## Summarize in a table (clinical status versus diagnostic status)
table(pred_df_m1b[pred_df_m1b$D > 0.5,]$Clinical_status,
      pred_df_m1b[pred_df_m1b$D > 0.5,]$Diagnostic_status)
```

```
##
##     Negative Positive
##   A        0        0
##   N        0        0
##   S        0        0
```

```
## Print summary table of clinical status versus diagnostic status from the original data
table(pred_df_m1b$Clinical_status, pred_df_m1b$Diagnostic_status)
```

```
##
##     Negative Positive
##   A        0        0
##   N      728        8
##   S       15       20
```

```
## boxplots
p1 <- (ggplot(data=pred_df_m1b, aes(x=Diagnostic_status, y=D))
       + geom_boxplot()
       + theme_bw())
p2 <- (ggplot(data=pred_df_m1b, aes(x=Clinical_status, y=D))
       + geom_boxplot()
       + theme_bw())

ggarrange(p1,p2, nrow=1)
```



Model 2:

The data outcome we are using is "diagnostically positive", meaning that an individual tests positive on at least one diagnostic test. This is what we have used in our other papers and seems to be popular in the literature (add some references to this). In this model, we assume that the two diagnostic tests are independent, and that there is some imprecision in the test results, so we include sensitivity and specificity for each test in the model.

Data Model

$$Y_k|T_{1k}, T_{2k}, D_k \sim Bernoulli\left(\pi_k^{DP}\right)$$

We are assuming that the test outcomes are independent. The probability of a diagnostically positive test for individual k, π_k^{DP}, is

$$\begin{aligned}\pi_k^{DP} &= P(Y_k = 1|D_k)\\ &= D_k \times P(Y_k = 1|D_k = 1) + (1 - D_k) \times (1 - P(Y_k = 0|D_k = 0)),\end{aligned}$$

where $P(Y_k = 1|D_k = 1) = P(T_{1k} = 1|D_k = 1) + P(T_{2k} = 1|D_k = 1) - P(T_{1k} = 1|D_k = 1) \times P(T_{2k} = 1|D_k = 1)$ and $P(Y_k = 0|D_k = 0) = P(T_{1k} = 0|D_k = 0) \times P(T_{2k} = 0|D_k = 0)$.

Process Model

$$D_k \sim Bernoulli(\delta_k)$$

where $\delta_k = logit^{-1}(logit(\rho) + \mathbf{x}_k^T \boldsymbol{\beta})$.

Prior Model

Prevalence

$$logit(\rho) \sim Normal(\mu_\rho, \tau^2)$$

where $\mu_\rho = logit^{-1}(\rho^*)$ and $\tau^2 =$...

Linear predictors

The regression parameters are $\boldsymbol{\beta} = (\beta_{age}, \beta_{sex}, \beta_{age*sex})^T$;

$$\boldsymbol{\beta} \sim Normal(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$$

where $\boldsymbol{\mu}_\beta = \mathbf{0}$ and $\boldsymbol{\Sigma}_\beta = \mathbf{I}$ in our code.

Other parameters:

$$\boldsymbol{\beta} \sim Normal(\boldsymbol{\mu}_\beta, \Sigma_\beta)$$

We will assume that the regression coefficients are independent, so Σ_β is a diagonal matrix.

OpenBUGS Model 2 Implementation

Not running this - Model 3 is essentially the same, but with a random intercept term.

```
# par(mfrow=c(2,2))
# ## Graphial summaries of posterior distribtuions
# hist(exp(model2_df$lpi)/(1+exp(model2_df$lpi)), main="prevalence", xlab="posterior draws")
# abline(v=mean(exp(model2_df$lpi)/(1+exp(model2_df$lpi))), lty="dashed", col="red")
#
# hist(model2_df$b1, main="Age", xlab="posterior draws")
# abline(v=mean(model2_df$b1), lty="dashed", col="red")
# hist(model2_df$b2, main="Sex", xlab="posterior draws")
# abline(v=mean(model2_df$b2), lty="dashed", col="red")
# hist(model2_df$b3, main="Age/Sex Interaction", xlab="posterior draws")
# abline(v=mean(model2_df$b3), lty="dashed", col="red")
#
# ## Numeric summaries of posterior distributions
# #boxplot(model2_df[,!(names(model2_df) %in% c("deviance"))])
```

OpenBUGS Model 2 Disease State Prediction

```
## Set up storage for model results
# pred_df_m2 <- data.frame(obs=1:nind,
#                            D=rep(NA,nind), ## average estimate
#                            SD=rep(NA,nind),
#                            LB=rep(NA,nind), ## 2.5th percentile
#                            UB=rep(NA,nind), ## 97.5th percentile
#                            model_assignment=rep(NA,nind),
```

```
#                            Clinical_status=ss_data2$ClinicalStatus,
#                            Diagnostic_status=ss_data2$Diagnostically_positive)
#
# ## Calculate probabilities of compartment membership for each posterior draw
# pred_df_m2$D <- apply(model2_df[,grep("D", names(model2_df))], 2, mean)
# pred_df_m2$SD <- apply(model2_df[,grep("D", names(model2_df))], 2, sd)
# pred_df_m2$LB <- apply(model2_df[,grep("D", names(model2_df))], 2,
#                          quantile, probs=0.025)
# pred_df_m2$UB <- apply(model2_df[,grep("D", names(model2_df))], 2,
#                          quantile, probs=0.975)
#
#
# summary(pred_df_m2)
#
# ## Apply a cut off of point estimate of 0.5; if pi.D > 0.5, classify as S (symptomatic), otherwise as
# ## Summarize in a table (clinical status versus diagnostic status)
# table(pred_df_m2[pred_df_m2$D > 0.5,]$Clinical_status,
#       pred_df_m2[pred_df_m2$D > 0.5,]$Diagnostic_status)
#
# ## Print summary table of clinical status versus diagnostic status from the original data
# table(pred_df_m2$Clinical_status, pred_df_m2$Diagnostic_status)
#
# ## boxplots
# p1 <- (ggplot(data=pred_df_m2, aes(x=Diagnostic_status, y=D))
#        + geom_boxplot()
#        + theme_bw())
# p2 <- (ggplot(data=pred_df_m2, aes(x=Clinical_status, y=D))
#        + geom_boxplot()
#        + theme_bw())
#
# ggarrange(p1,p2, nrow=1)
```

Model 3:

The data outcome we are using is "diagnostically positive", meaning that an individual tests positive on at least one diagnostic test. This is what we have used in our other papers and seems to be popular in the literature (add some references to this). In this model, we assume that the two diagnostic tests are independent, and that there is some imprecision in the test results, so we include sensitivity and specificity for each test in the model. We also include a random intercept term to see how this changes the model.

Data Model

$$Y_k|T_{1k}, T_{2k}, D_k \sim Bernoulli\left(\pi_k^{DP}\right)$$

We are assuming that the test outcomes are independent. The probability of a diagnostically positive test for individual k, π_k^{DP}, is

$$\begin{aligned}\pi_k^{DP} &= P(Y_k = 1|D_k)\\ &= D_k \times P(Y_k = 1|D_k = 1) + (1 - D_k) \times (1 - P(Y_k = 0|D_k = 0)),\end{aligned}$$

where $P(Y_k = 1|D_k = 1) = P(T_{1k} = 1|D_k = 1) + P(T_{2k} = 1|D_k = 1) - P(T_{1k} = 1|D_k = 1) \times P(T_{2k} = 1|D_k = 1)$ and $P(Y_k = 0|D_k = 0) = P(T_{1k} = 0|D_k = 0) \times P(T_{2k} = 0|D_k = 0)$.

Process Model

$$D_k \sim Bernoulli(\delta_k)$$

where $\delta_k = logit^{-1}(logit(\rho) + \mathbf{x}_k^T \boldsymbol{\beta} + \epsilon_k)$.

Prior Model

Prevalence

$$logit(\rho) \sim Normal(\mu_\rho, \tau^2)$$

where $\mu_\rho = logit^{-1}(\rho^*)$ and $\tau^2 = \ldots$

Note, the prevalence in this population may be higher - these are exposed dogs in the United States in our hunting hound population. We may want to change this, but we also want to generalize this to the larger canine population - to Brazil, if possible.

Linear predictors

The regression parameters are $\boldsymbol{\beta} = (\beta_{age}, \beta_{sex}, \beta_{age*sex})^T$;

$$\boldsymbol{\beta} \sim Normal(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$$

where $\boldsymbol{\mu}_\beta = \mathbf{0}$ and $\boldsymbol{\Sigma}_\beta = \mathbf{I}$ in our code.

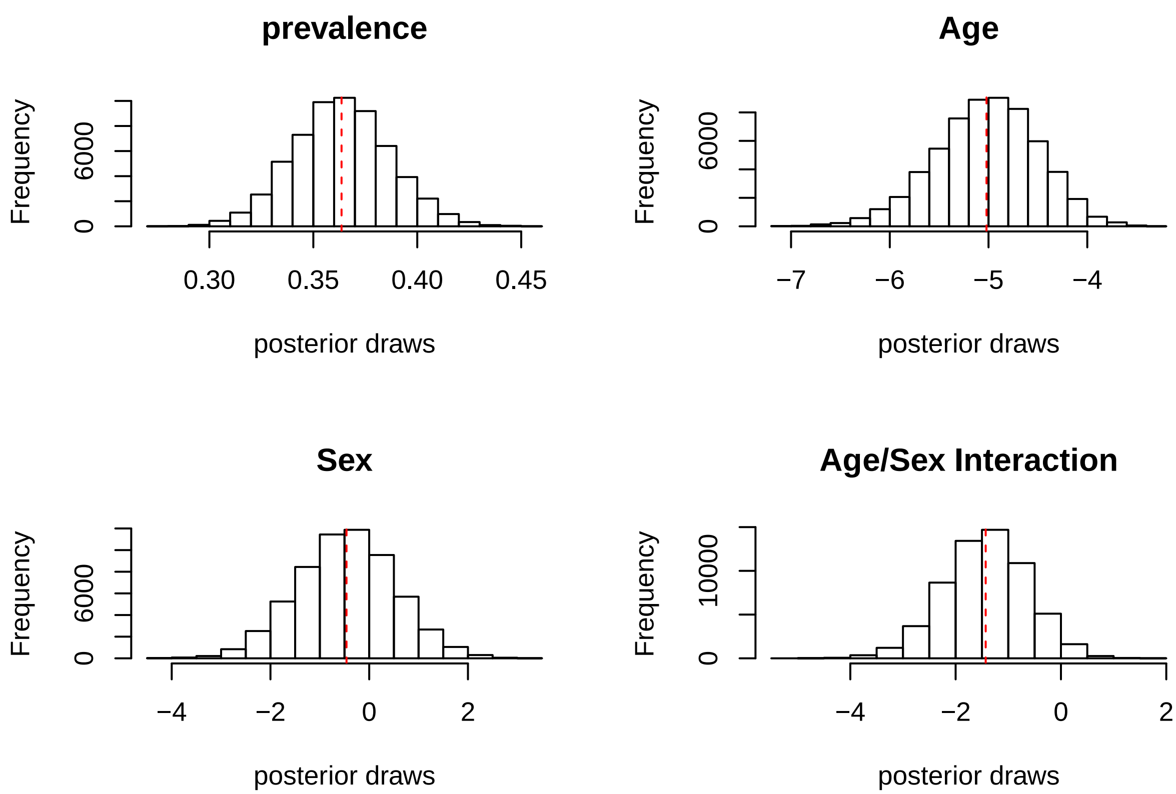
We will assume that the regression coefficients are independent, so Σ_β is a diagonal matrix.

Individual random effect - parameterized using precision.

$$\boldsymbol{\epsilon} \sim Normal(0, 5 \times 10^{-3})$$

OpenBUGS Model 3 Implementation

```r
par(mfrow=c(2,2))
## Graphial summaries of posterior distribtuions
hist(exp(model3_df$lpi)/(1+exp(model3_df$lpi)), main="prevalence", xlab="posterior draws")
abline(v=mean(exp(model3_df$lpi)/(1+exp(model3_df$lpi))), lty="dashed", col="red")

hist(model3_df$b1, main="Age", xlab="posterior draws")
abline(v=mean(model3_df$b1), lty="dashed", col="red")
hist(model3_df$b2, main="Sex", xlab="posterior draws")
abline(v=mean(model3_df$b2), lty="dashed", col="red")
hist(model3_df$b3, main="Age/Sex Interaction", xlab="posterior draws")
abline(v=mean(model3_df$b3), lty="dashed", col="red")
```




```
## Numeric summaries of posterior distributions
#boxplot(model3_df[,!(names(model3_df) %in% c("deviance"))])
```

The mean posterior prevalence is 0.3636 and the 95% credible interval is: (0.3191, 0.4097).

OpenBUGS Model 3 Disease State Prediction

```
## Set up storage for model results
pred_df_m3 <- data.frame(obs=1:nind,
                         D=rep(NA,nind), ## average estimate
                         SD=rep(NA,nind),
                         LB=rep(NA,nind), ## 2.5th percentile
                         UB=rep(NA,nind), ## 97.5th percentile
                         model_assignment=rep(NA,nind),
                         Clinical_status=ss_data2$ClinicalStatus,
                         Diagnostic_status=ss_data2$Diagnostically_positive)

## Calculate probabilities of compartment membership for each posterior draw
pred_df_m3$D <- apply(model3_df[,grep("D", names(model3_df))], 2, mean)
pred_df_m3$SD <- apply(model3_df[,grep("D", names(model3_df))], 2, sd)
pred_df_m3$LB <- apply(model3_df[,grep("D", names(model3_df))], 2,
                       quantile, probs=0.025)
pred_df_m3$UB <- apply(model3_df[,grep("D", names(model3_df))], 2,
                       quantile, probs=0.975)
```

```r
summary(pred_df_m3)
```

```
##       obs              D                   SD                  LB
##  Min.   :  1.0   Min.   :0.0000000   Min.   :0.000000   Min.   :0
##  1st Qu.:193.5   1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0
##  Median :386.0   Median :0.0000000   Median :0.000000   Median :0
##  Mean   :386.0   Mean   :0.0015327   Mean   :0.009780   Mean   :0
##  3rd Qu.:578.5   3rd Qu.:0.0000833   3rd Qu.:0.009128   3rd Qu.:0
##  Max.   :771.0   Max.   :0.5380500   Max.   :0.498554   Max.   :0
##        UB          model_assignment Clinical_status Diagnostic_status
##  Min.   :0.000000   Mode:logical     A:  0           Negative:743
##  1st Qu.:0.000000   NA's:771         N:736           Positive: 28
##  Median :0.000000                    S: 35
##  Mean   :0.005188
##  3rd Qu.:0.000000
##  Max.   :1.000000
```

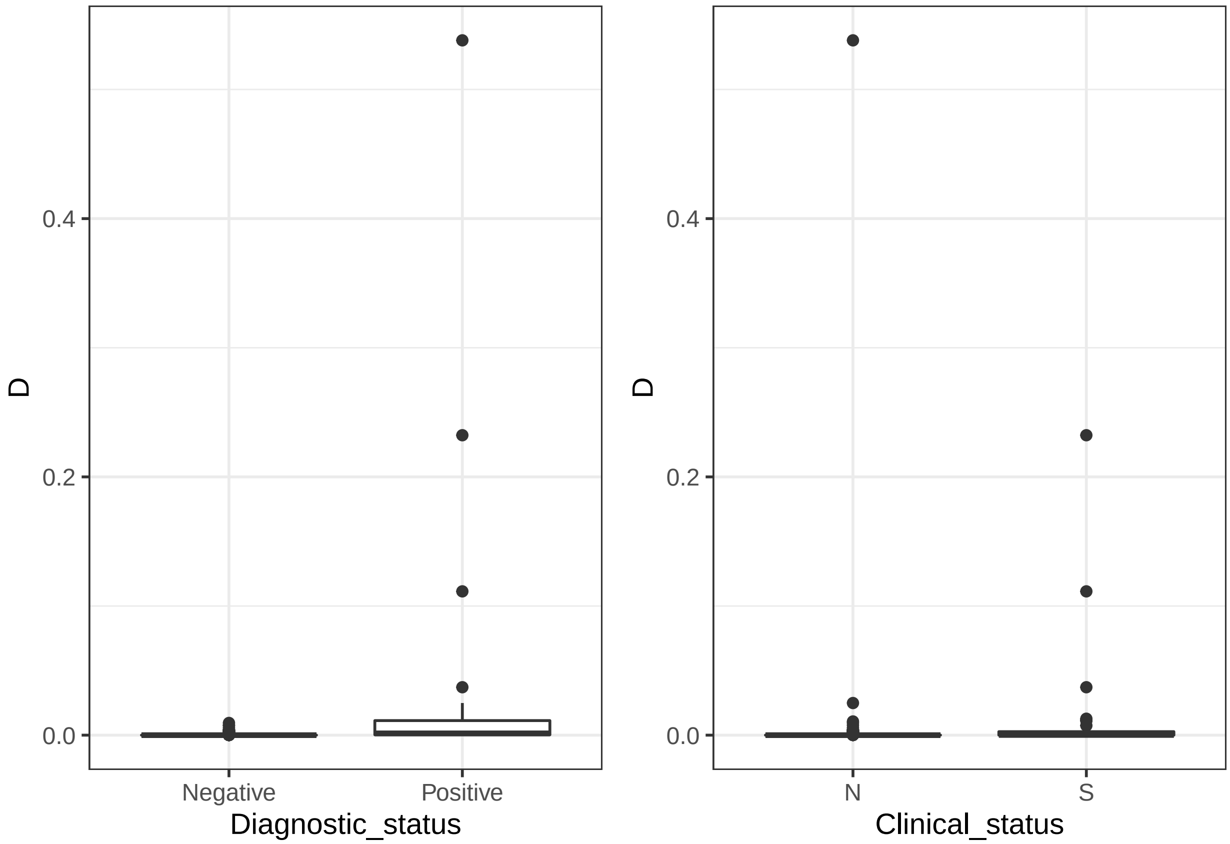
```r
## Apply a cut off of point estimate of 0.5; if pi.D > 0.5, classify as S (symptomatic), otherwise as N
## Summarize in a table (clinical status versus diagnostic status)
table(pred_df_m3[pred_df_m3$D > 0.5,]$Clinical_status,
      pred_df_m3[pred_df_m3$D > 0.5,]$Diagnostic_status)
```

```
##
##     Negative Positive
##   A        0        0
##   N        0        1
##   S        0        0
```

```r
## Print summary table of clinical status versus diagnostic status from the original data
table(pred_df_m3$Clinical_status, pred_df_m3$Diagnostic_status)
```

```
##
##     Negative Positive
##   A        0        0
##   N      728        8
##   S       15       20
```

```r
## boxplots
p1 <- (ggplot(data=pred_df_m3, aes(x=Diagnostic_status, y=D))
       + geom_boxplot()
       + theme_bw())
p2 <- (ggplot(data=pred_df_m3, aes(x=Clinical_status, y=D))
       + geom_boxplot()
       + theme_bw())

ggarrange(p1,p2, nrow=1)
```



0.4
0.2
0.0
D
Negative
Positive
Diagnostic_status
0.4
0.2
0.0
D
N
S
Clinical_status