Men's Fertility

By: Teah Thies & Jacob Mitchell

STAT 473

Introduction

We are interested in which factors can lead to an altered fertility diagnosis. The data set used in this context is a classification problem because our response variable, diagnosis, is binomial. Our predicting variables include season, age, childish diseases, accidents, surgical intervention, high fevers in the last year, frequency of alcohol consumption, smoking habit, and number of hours sitting per day. From these variables, the ones that are in a person's control are alcohol consumption, smoking habit, and season. None of them ended up having any correlation with fertility. Therefore, based on this data set, male fertility is out of a man's control.

Before starting the analysis, we had assumptions that alcohol consumption, smoking habit, and age would be leading variables in predicting altered infertility. We used a few different techniques to study the data set such as correlation, logistic model, LDA model, KNN model, classification tree, and random forest. We found that the only significant predictor in this data set is accidents and the most accurate technique is random forest.

Questions of Interest

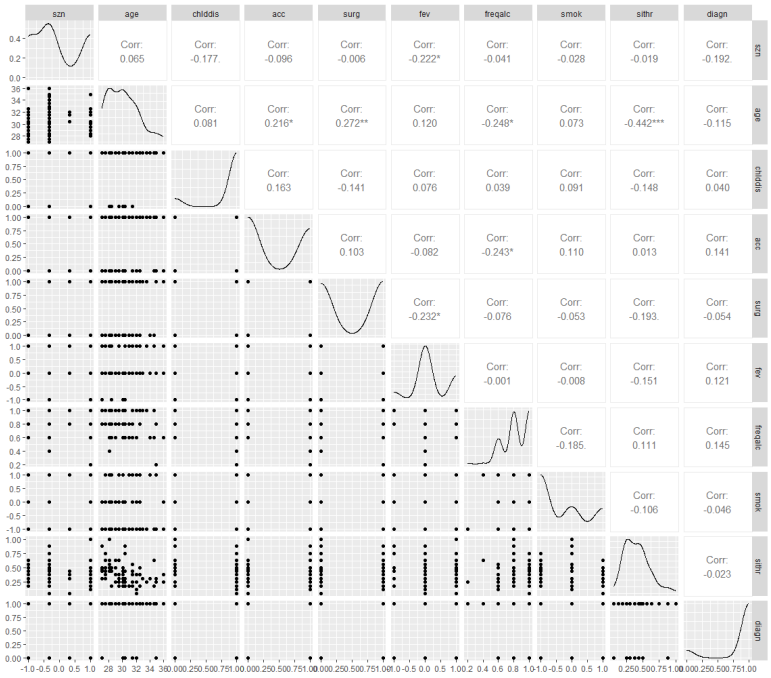
For this project our main questions were as follows:

1. Which predictors are significant in predicting normal or altered fertility?
2. What kinds of correlations are there within the data set?
3. Which variables can a person control, and what are their effects?
4. Which models predict best?

Analysis

Analysis: Data Exploration and Correlations

Before we begin with an in depth analysis of the data, we first must examine how the data all fits together. In an effort to answer our second question we observe any potential correlations within the data generated by ggplot():



From this we can see that a person closer to 36 will have less hours sitting a day, than someone closer to 18. Since these two are our only continuous variables, Age and Hours Sitting having some form of correlation when compared to other variables is to be expected. We also note that Age and having had a Surgery have a positive correlation; something to be expected as general health tends to depreciate over time. We also note a slight correlation between recent surgeries and fevers. This is likely an indicator of recovery from said surgery, as fevers can arise as a side effect of the recovery process.

<u>Analysis: Which Predictors are Significant?</u>

Since we are dealing with a classification problem, one of the main approaches to take is creating a logistic model.

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -11.9203     7.0215  -1.698   0.0896 .
szn           0.6847     0.4462   1.535   0.1249
age           0.3161     0.1938   1.631   0.1029
chlddis       0.3164     0.9653   0.328   0.7431
acc           1.7469     0.8481   2.060   0.0394 *
surg          0.1271     0.7285   0.174   0.8615
fev          -0.9772     0.6844  -1.428   0.1534
freqalc      -2.6816     2.0023  -1.339   0.1805
smok          0.2711     0.4329   0.626   0.5311
sithr         2.6801     2.1946   1.221   0.2220
```

To our surprise, the only significant predictor in this data set was accidents. It is significant at level 0.1. For the data, we see that the coefficient estimate for accidents is approximately 1.75. Therefore, a one-unit increase in accidents with every other variable unchanged is associated with an increase in log odds of altered diagnosis by 1.75 units. The accuracy of this model is found to be 80% using a confusion matrix. The sensitivity is 87%.

Our next thought was to test our only significant predictor, accidents, against diagnosis on its own to see if the results remained.

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.6150     0.5981  -4.372 1.23e-05 ***
acc           0.9620     0.7001   1.374    0.169
```

We can see that accidents are not significant in this model alone. Thus, the other variables have an influence or correlation with accidents.

Analysis: Model Comparison

Having considered the options for a classification problem we arrived at 5 models: the aforementioned Logistic Model, an LDA Model, a Classification Tree, and Forest, and a KNN model for experimentation purposes. During testing we decided on a 50/50 split of training data versus test data. This was mainly due to the data set's size, as an 80/30 split resulted in the LDA model failing to generate predictions. The size of the dataset also led us to use an LDA model as opposed to a QDA Model in an effort to reduce the variance of the classifiers.

Since we already used the Log model to look at correlations, we started with those predictions. We observed:

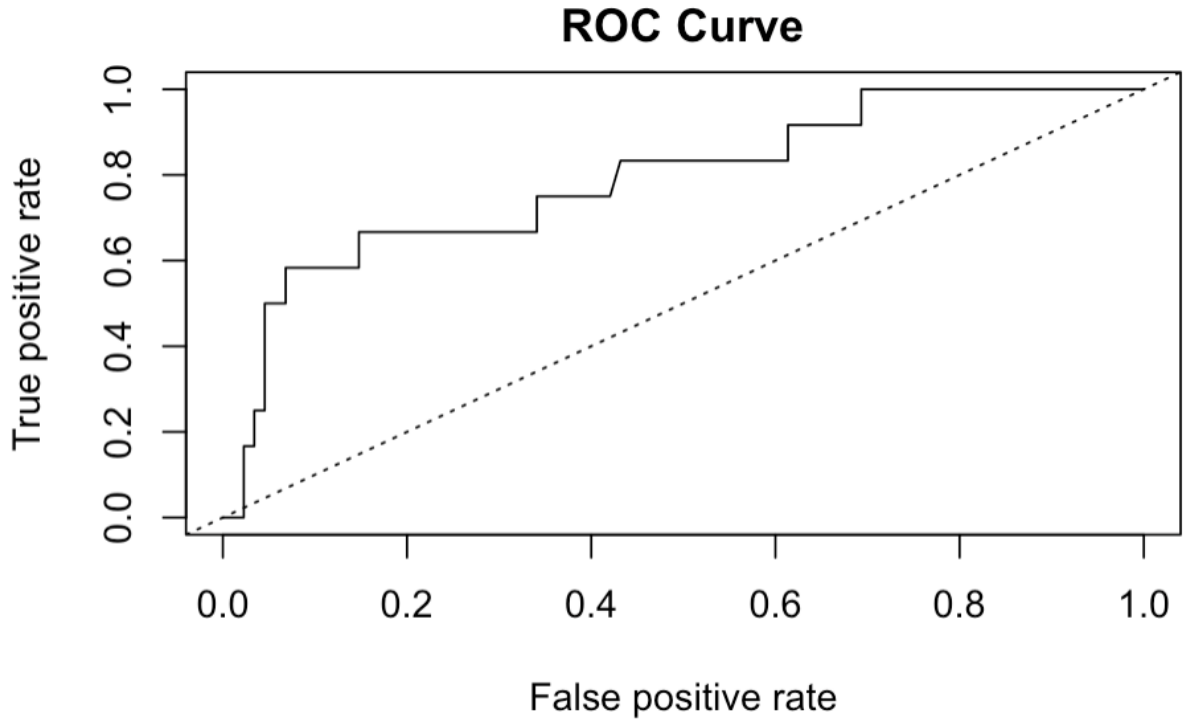
Predicted Status/True Status	**0**	**1**
0	40	4
1	6	0

For a sensitivity of 87% and specificity of 0%. We look at the sensitivity and specificity here as we have a small sample size, and are dealing with medical information. Hence sensitivity and specificity would be a better indicator for our models as they are more readily conveyable in a medical setting.

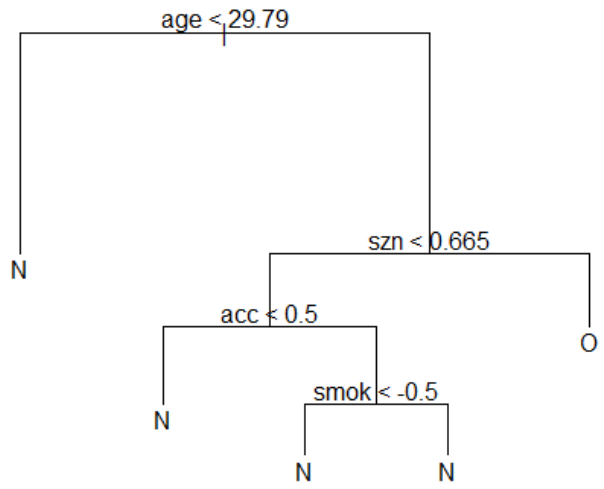
Next, we proceed with an LDA Model where we observed the following classifications results:

Predicted Status/True Status	**0**	**1**
0	87	12
1	0	1

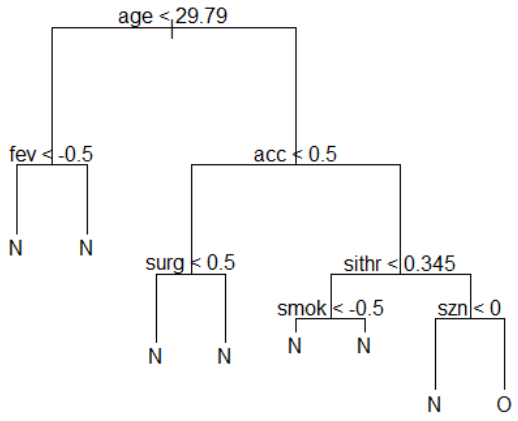
Which gives us an overall sensitivity of 100%, and specificity of ~8%. We also observed the LOC curve an AUC of 0.79:



Our next model attempt was the Classification Tree, which gave us the following graphical representation:



Indicating that age, and season are the most important factors when determining abnormalities. This is also fits with the results we see when we model using the whole data set:



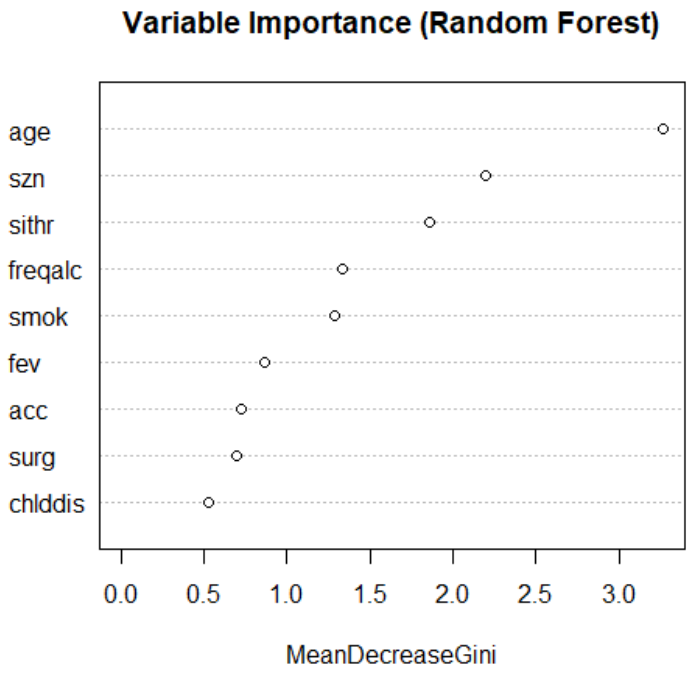
Here we see sit rate and accidents play a role in the classification process, mirroring the correlation we saw in the Logistic Model in the previous section. Our Classification Tree then gave us the following results:

Predicted Status/True Status	N	O
N	36	6
O	9	2

Giving us a sensitivity of 80% and specificity of 25%. Quite an improval of specificity from the LDA model. So, to improve effectivity we used the Random Forest, which gives us:

Predicted Status/True Status	N	O
N	46	3
O	0	1

For a sensitivity of 100% and specificity of 25%, and OOB Error of 22%. Additionally we can visualize each predictors effect in the following chart:



We then proceeded with a KNN model to see how this would interpret the data. While normally a KNN model would not be the best fit for categorical data, given the sample size we wanted to see what would happen. We proceed with K = 3, 5, 7, and 11 assigning a prediction threshold of 0.3 and 0.5 provide a generous estimate and strenuous threshold to see how this threshold would affect the predictions.

For the generous threshold we saw that K = 7 performed the best. While this model had the same specificity as K = 5 (with that being 50%), we saw an increase in sensitivity to 95.7%. Below is the confusion matrix for K = 7:

Predicted Status/True Status	**0**	**1**
0	44	2
1	2	2

As for the strenuous threshold, we saw for K > 5 the model would not assign anything to our false regions, instead only predicting a normal outcome. For K = 5 we had no assignment to the True False region despite a higher sensitivity, and for K = 3 a specificity of 25% and a sensitivity of 97.8%. We chose K = 3 as the best performing since it got one False correct, much better than the 0% sensitivity of K = 5. Below is the confusion matrix for our choice:

Predicted Status/True Status	**0**	**1**
0	42	3
1	4	1

Based on these results we can conclude that, based on this data set, the Random Forest will perform best. Followed by the KNN model for 30%, 50%, LDA and Log Model, and finally our Classification Tree.

Conclusion

We began this study in hopes to find clarification on male fertility. We wanted to find out which factors predict altered infertility as well as which model can most accurately predict altered fertility. Using the logistic model with an accuracy of 80% and sensitivity of 87%, we found that the “accidents and trauma” variable is significant in doing such and the random forest technique

is the most accurate. Since one cannot choose to dodge an accident in their life, we can say that based on the logistic model, male fertility is out of one's control. Although, given the random forest approach to this question, age, sit rate, and season are variables of importance. To be more precise, these predictors highly affected the random forest. In attempts to build the most accurate model for our classification problem, we needed to decide between first between LDA and QDA models. Since we were unable to use QDA because of the small sample size, the LDA model came back with an sensitivity of about 8% and an AUC value of 79%. The AUC value is not too bad, but the sensitivity is very low. We still thought we could find a better technique to answer our questions.

Back to the aforementioned random forest, we found this to be the most effective model at predicting fertility. With an overall accuracy of 94% this was a large increase from the single Classification Tree's accuracy of 70% and also provided the best specificity. We also tested out various KNN fits to get an idea for how this might model the data set. These models proved to be surprisingly accurate, but tended to work best around K= 5 and 7 after creating a decision boundary at 0.3 and 0.5. Overall, as one might expect from a classification problem, the more powerful Random Forest proved the best at classifying male fertility.

Appendix

Data Cleaning

```
## Convert age to actual age
df = as_tibble(df) %>% mutate(age = (age * 18) + 18)

## Convert Diagnosis to
## Normal (N) = 0
## Altered (O) = 1
df = as_tibble(df) %>% mutate(diagn = ifelse(diagn == "O", 1, 0))

## convert accident to "Yes" = 1, "No" = 0
```

```
df = df %>% mutate(acc = ifelse(acc == 1, 0, 1))
```

Log Model

```
log_model = glm(diagn ~ ., family = "binomial", data = df)
summary(log_model)

# log confusion matrix and accuracy
n = nrow(df)
prop = .5
set.seed(1)
train_id = sample(1:n, size = n*prop, replace = FALSE)
test_id = (1:n)[-which(1:n %in% train_id)]
train_set = df[train_id, ]
test_set = df[test_id, ]
train_glm = glm(diagn ~ ., family = "binomial", data = train_set)
glm_pred_class = predict(train_glm, type = "response")
glm_pred_class = ifelse(glm_pred_class > 0.5, "Up", "Down")
cmatrix = table(predict_status = glm_pred_class, true_status=test_set$diagn)
cmatrix
(cmatrix[1, 1] + cmatrix[2, 2])/sum(cmatrix)
```

Significant Log Model

```
log_mod_sig = glm(diagn ~ acc, family = "binomial", data = df)
summary(log_mod_sig)
```

LDA Model

```
lda_fit = lda(diagn ~ ., data = df)
lda_fit

# confusion matrix
```

```
lda_pred_class = predict(lda_fit)$class
con_matrix = table(predict_status = lda_pred_class, true_status=df$diagn)
con_matrix

# accuracy
(con_matrix[1, 1] + con_matrix[2, 2])/sum(con_matrix)

# ROC Curve
lda_pred = predict(lda_fit, df)
lda_pred_post = lda_pred$posterior[,2]
pred = prediction(lda_pred_post, df$diagn)
perf = performance(pred, "tpr", "fpr")
plot(perf, main = "ROC Curve")
abline(0, 1, lty=3)

# auc value
auc = as.numeric(performance(pred, "auc")@y.values)
Auc
```

Helper Function

```
conMatStats = function(T) {
  require(tidyverse)
  acc = diag(T) |> sum()/sum(T)
  sens = T[1] / (T[1] + T[2])
  spec = T[4] / (T[3] + T[4])
  x = c(acc, sens, spec)
  y = c(1 - acc, 1 - sens, 1 - spec)
  z = c('Accuracy', 'Sensitivity (FNR)', 'Specificity (FPR)')
  return(tibble(z, x, y))
}
```

Classification Tree

```
library(tree)
# Create trees
rt = tree(diagn~., train_set)
all = tree(diagn~., df)

# Visualizations
plot(rt)
text(rt, pretty= 0)

plot(all)
text(all, pretty= 0)

# Predictions
pred = predict(rt, test_set, type= "class")

class = table(prediction_status= pred,
              true_status= test_set$diagn)
conMatStats(class)
```

Random Forest

```
# Random forest
library(randomForest)
p = ncol(df) - 1
forest = randomForest(diagn~., train_set,
              mtry= round(sqrt(p)), importance= TRUE)
importance(forest, type= 2)
frpred = predict(forest, spdf$test, type= "class")
forest_pred = table(prediction_status= rfpred,
              true_status= test_set$diagn)
conMatStats(forest_pred)
```

KNN

```
library(caret)
# Functions
# Mutate function
trnsfrm = function(vec, thresh) {
  return(as_tibble(vec) |> mutate(value = ifelse(value > thresh, 1, 0)))
}

# Quickly Make Confusion Matrix
cMat = function(predOutcome, trueOutcome) {
  return (table(prediction_status= predOutcome,
                true_status= trueOutcome))
}

# Create Models
knn3 = knnreg(diagn~., train_set, k= 3)
knn5 = knnreg(diagn~., train_set, k= 5)
knn7 = knnreg(diagn~., train_set, k= 7)
knn11 = knnreg(diagn~., train_set, k= 11)

# Train Models
knn3_pred = predict(knn3, test_set)
knn5_pred = predict(knn5, test_set)
knn7_pred = predict(knn7, test_set)
knn11_pred = predict(knn11, test_set)



# Train Models
knn3_pred = predict(knn3, spdf$test)
knn5_pred = predict(knn5, spdf$test)
knn7_pred = predict(knn7, spdf$test)
```

```
knn11_pred = predict(knn11, spdf$test)

# Create Thresholds
generous_thresh = c(trnsfrm(knn3_pred, 0.3),
                    trnsfrm(knn5_pred, 0.3),
                    trnsfrm(knn7_pred, 0.3),
                    trnsfrm(knn11_pred, 0.3))

stren_thresh = c(trnsfrm(knn3_pred, 0.5),
                 trnsfrm(knn5_pred, 0.5),
                 trnsfrm(knn7_pred, 0.5),
                 trnsfrm(knn11_pred, 0.5))

# Stren Threshold
stren3 = cMat(stren_thresh[1]$value, test_set$diagn)
stren5 = cMat(stren_thresh[2]$value, test_set$diagn)
stren7 = cMat(stren_thresh[3]$value, test_set$diagn)
stren11 = cMat(stren_thresh[4]$value, test_set$diagn)

conMatStats(stren3)
conMatStats(stren5)
conMatStats(stren7)
conMatStats(stren11)

# Generous Threshold
gen3 = cMat(generous_thresh[1]$value, test_set$diagn)
gen5 = cMat(generous_thresh[2]$value, test_set$diagn)
gen7 = cMat(generous_thresh[3]$value, test_set$diagn)
gen11 = cMat(generous_thresh[4]$value, test_set$diagn)

conMatStats(gen3)
```

```
conMatStats(gen5)
conMatStats(gen7)
conMatStats(gen11)
```