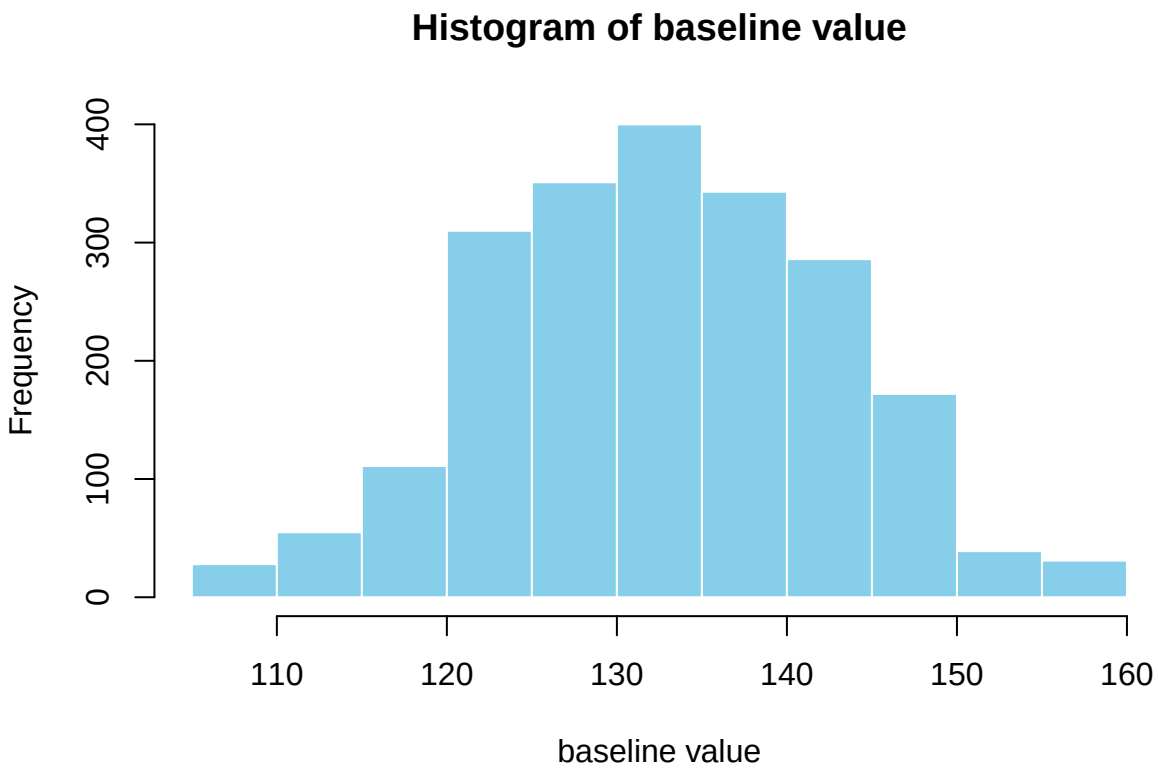
Fetal Health

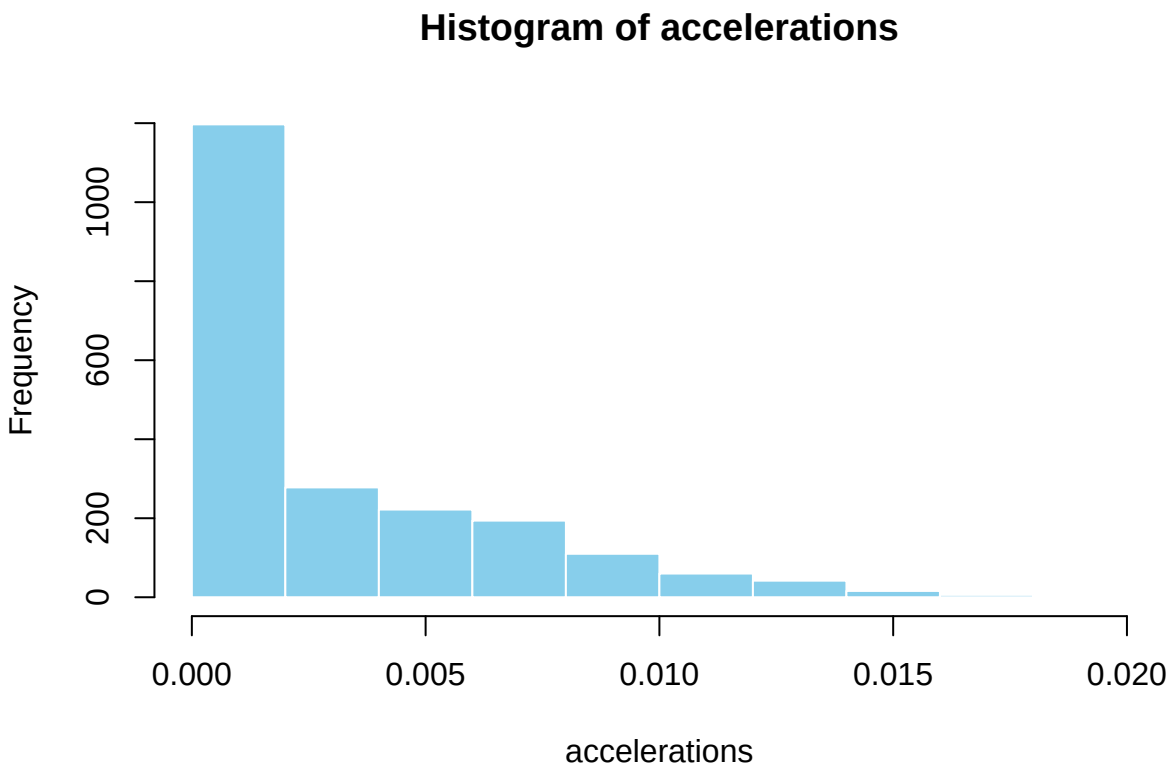
Finn de Lange

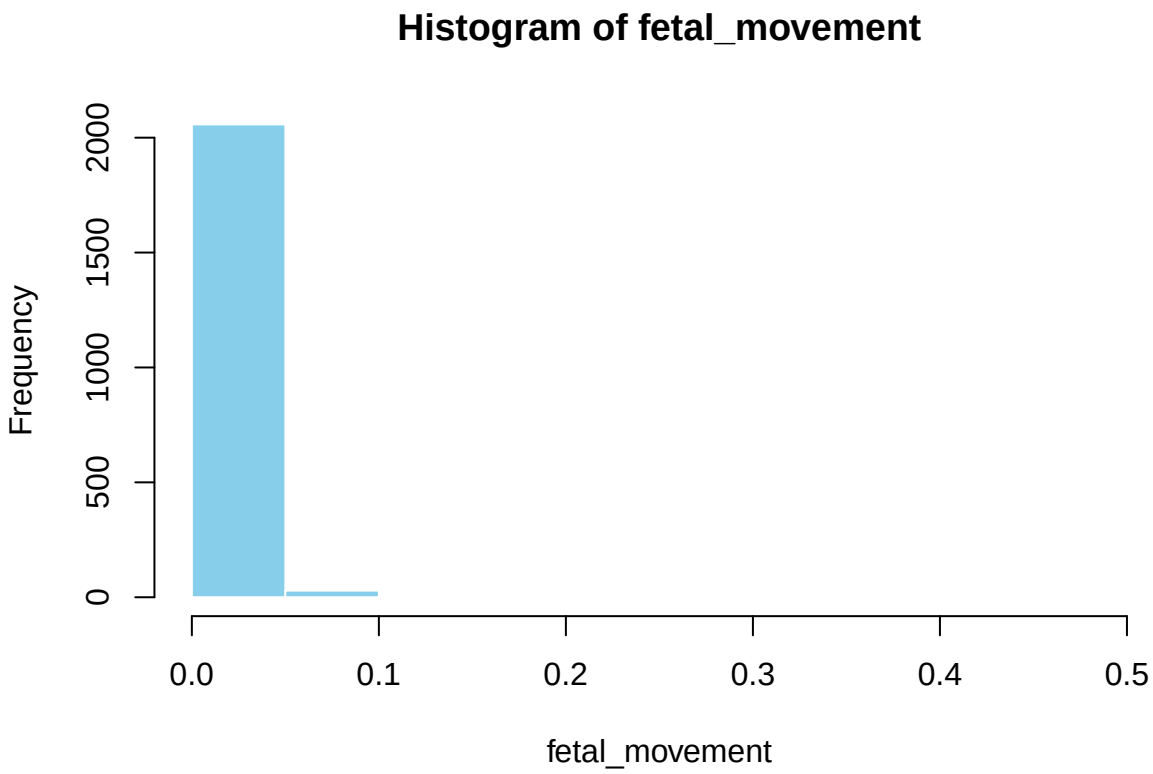
2025-05-24

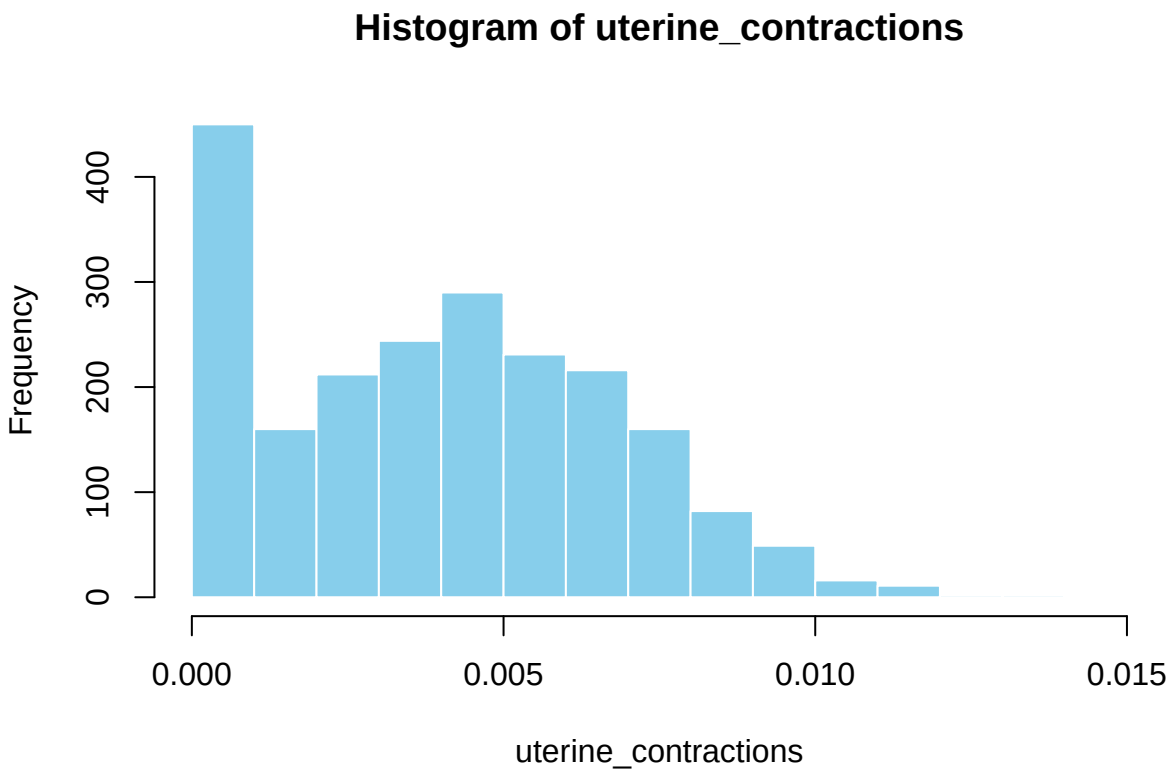
Data import

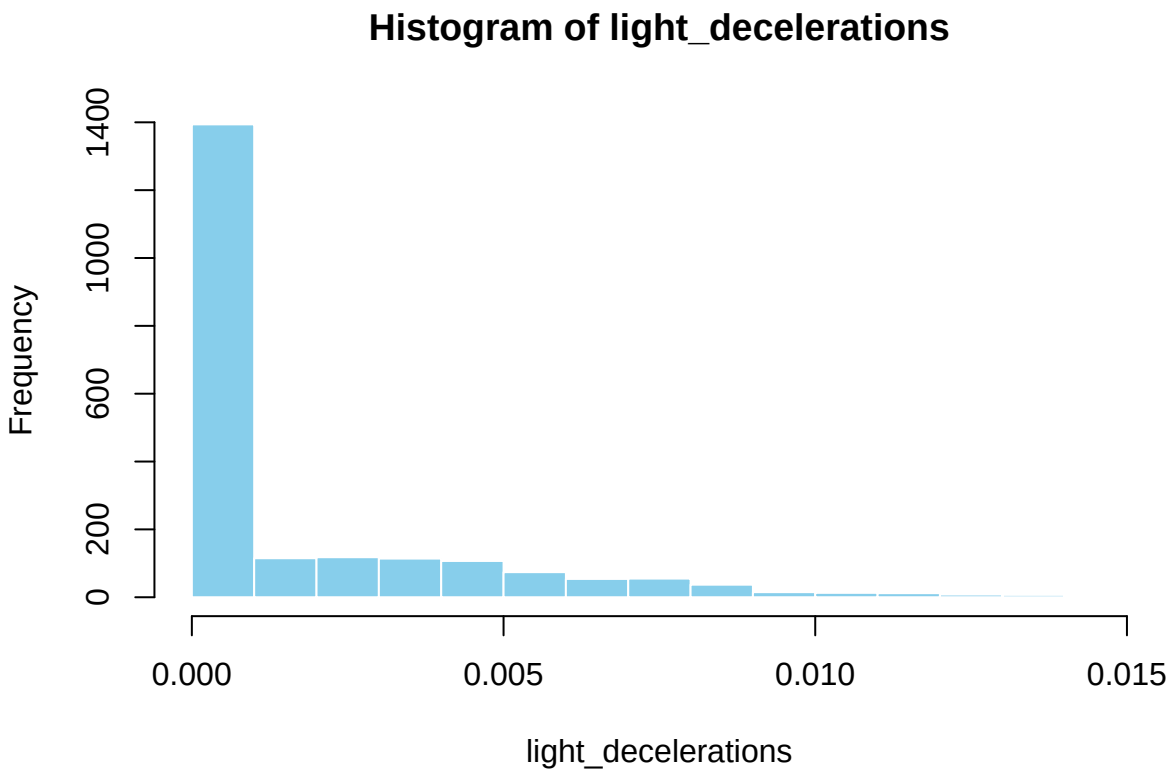
```
data <- read_csv("fetal_health.csv")

for (col_name in names(data)) {
  if (is.numeric(data[[col_name]])) {
    hist(data[[col_name]],
         main = paste("Histogram of", col_name),
         xlab = col_name,
         col = "skyblue",
         border = "white")
  } else {
    print(paste(col_name, "is not numeric and was skipped."))
  }
}
```

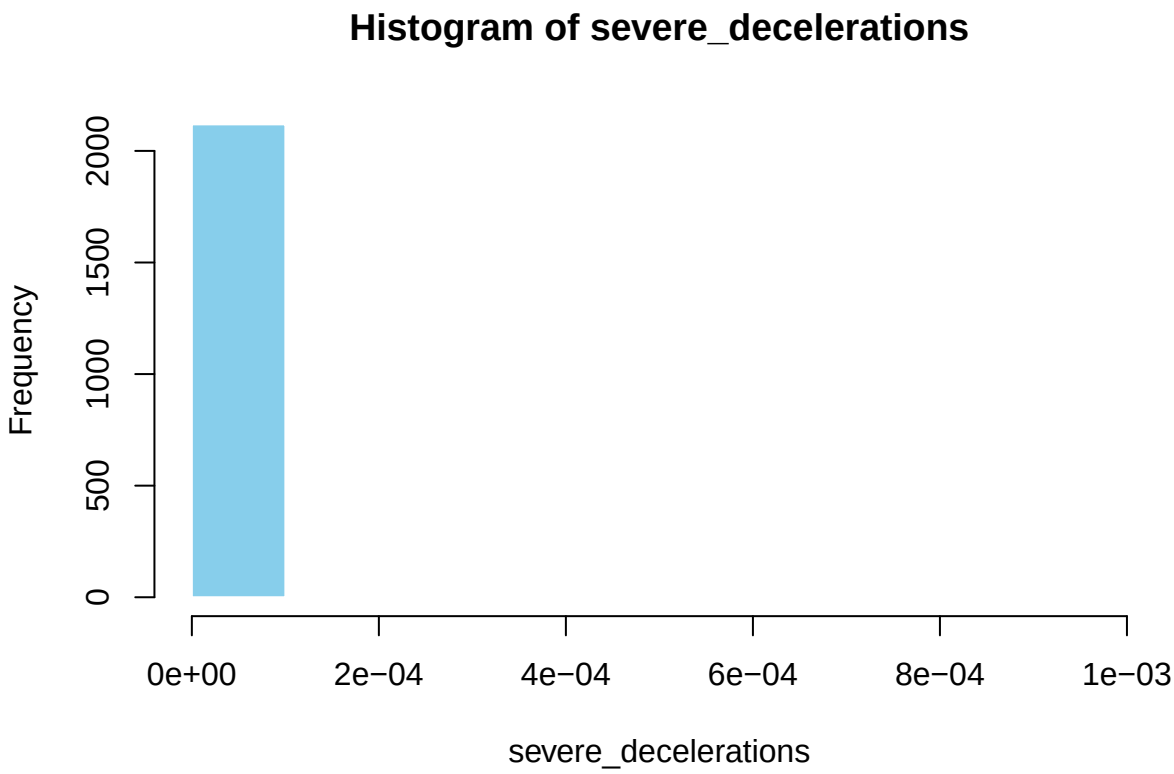

Histogram of baseline value
Frequency
0
100
200
300
400
110
120
130
140
150
160
baseline value

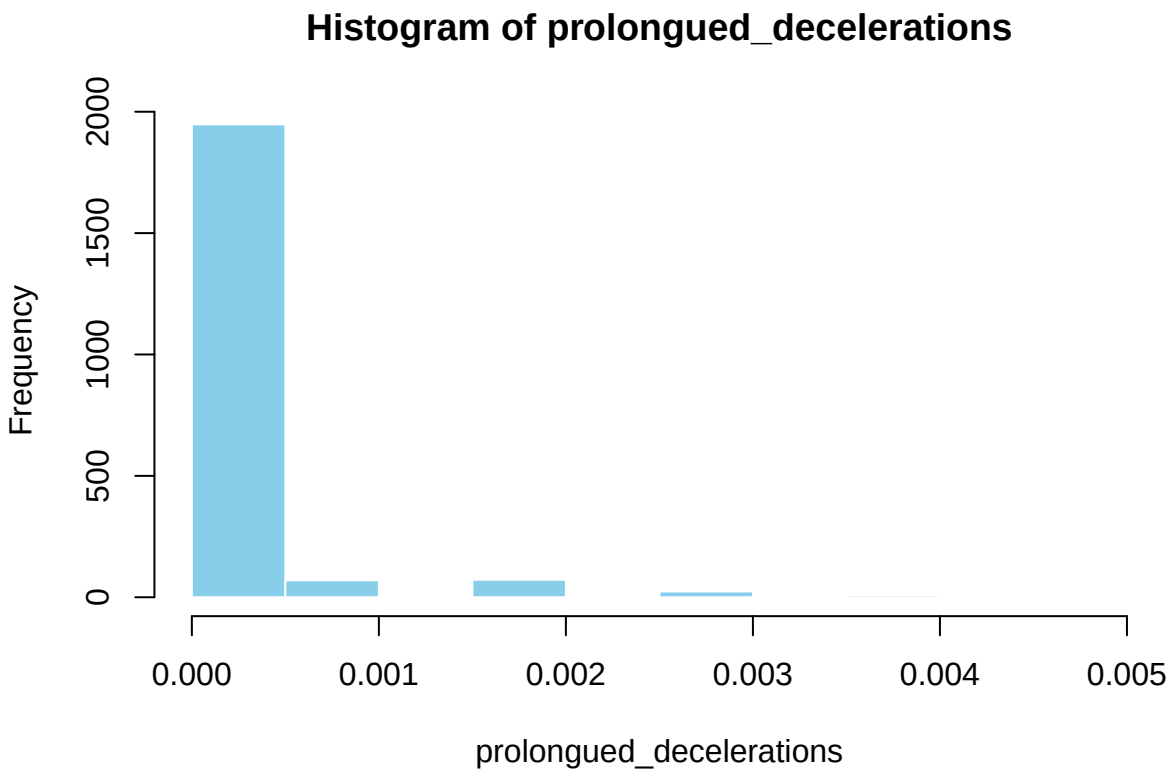

Histogram of accelerations
Frequency
0
200
600
1000
0.000
0.005
0.010
0.015
0.020
accelerations

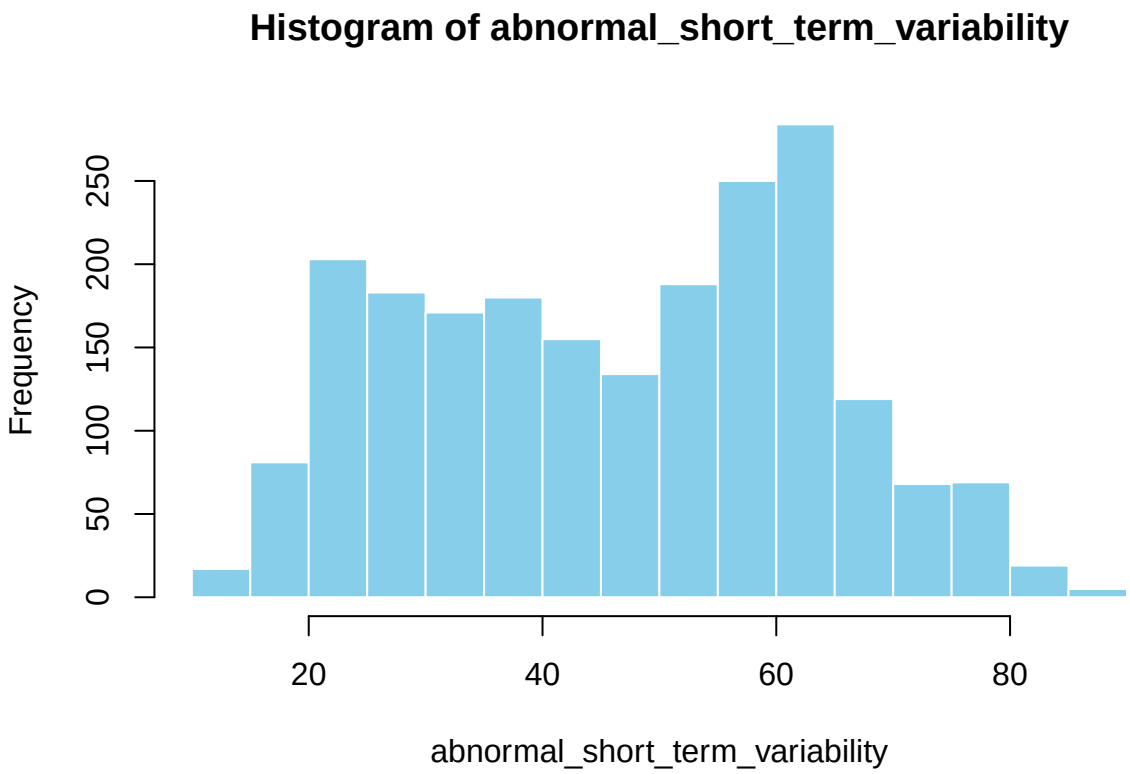

Histogram of fetal_movement
Frequency
0
500
1000
1500
2000
0.0
0.1
0.2
0.3
0.4
0.5
fetal_movement

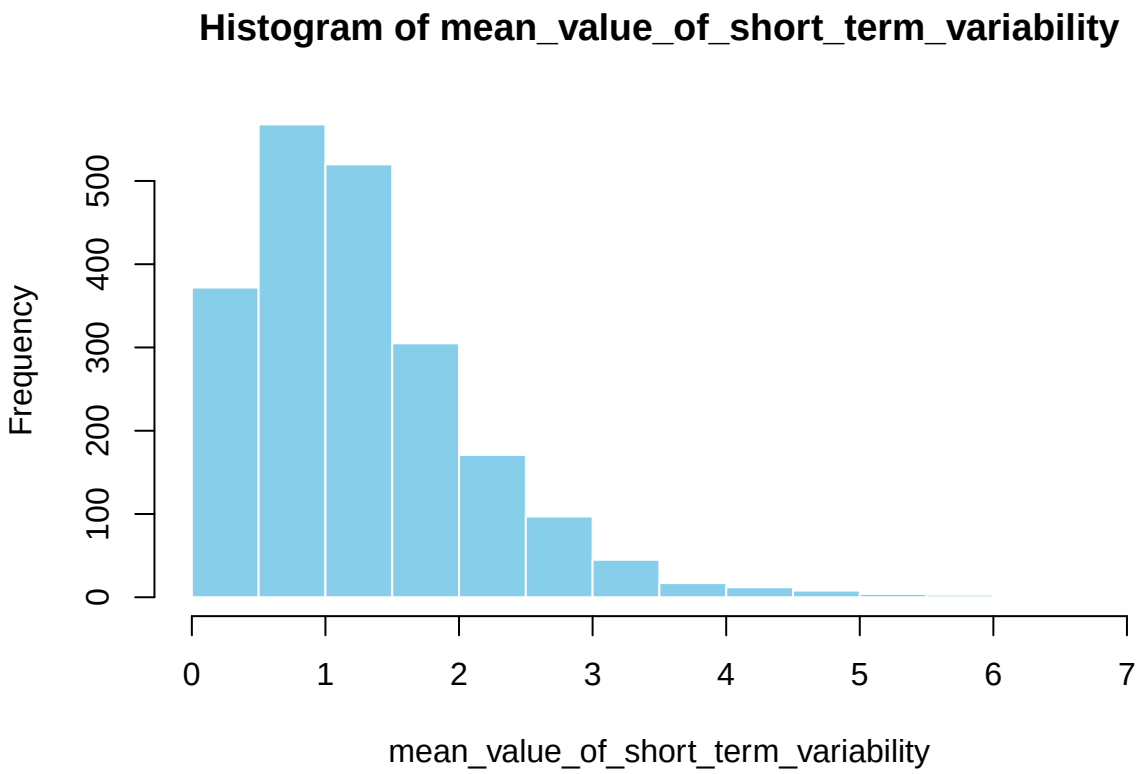

Histogram of uterine_contractions
Frequency
0
100
200
300
400
0.000
0.005
0.010
0.015
uterine_contractions

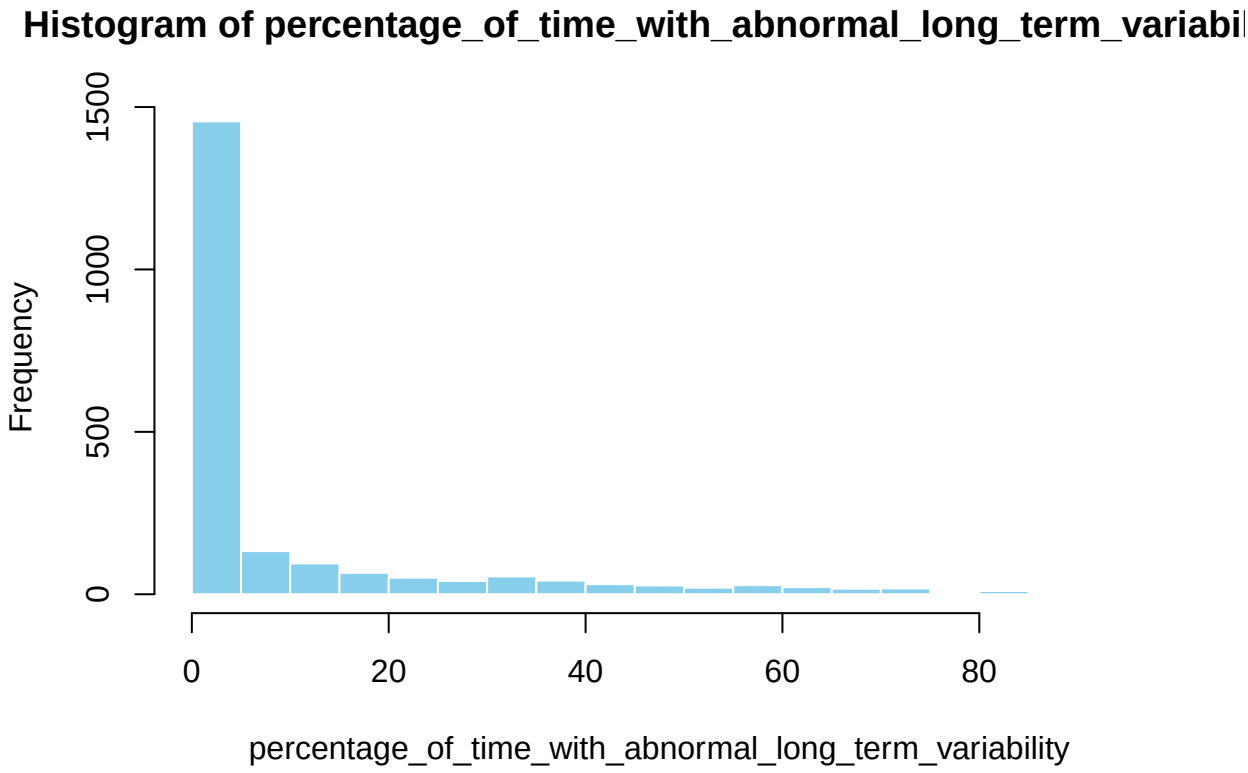

Histogram of light_decelerations
Frequency
0
200
600
1000
1400
0.000
0.005
0.010
0.015
light_decelerations

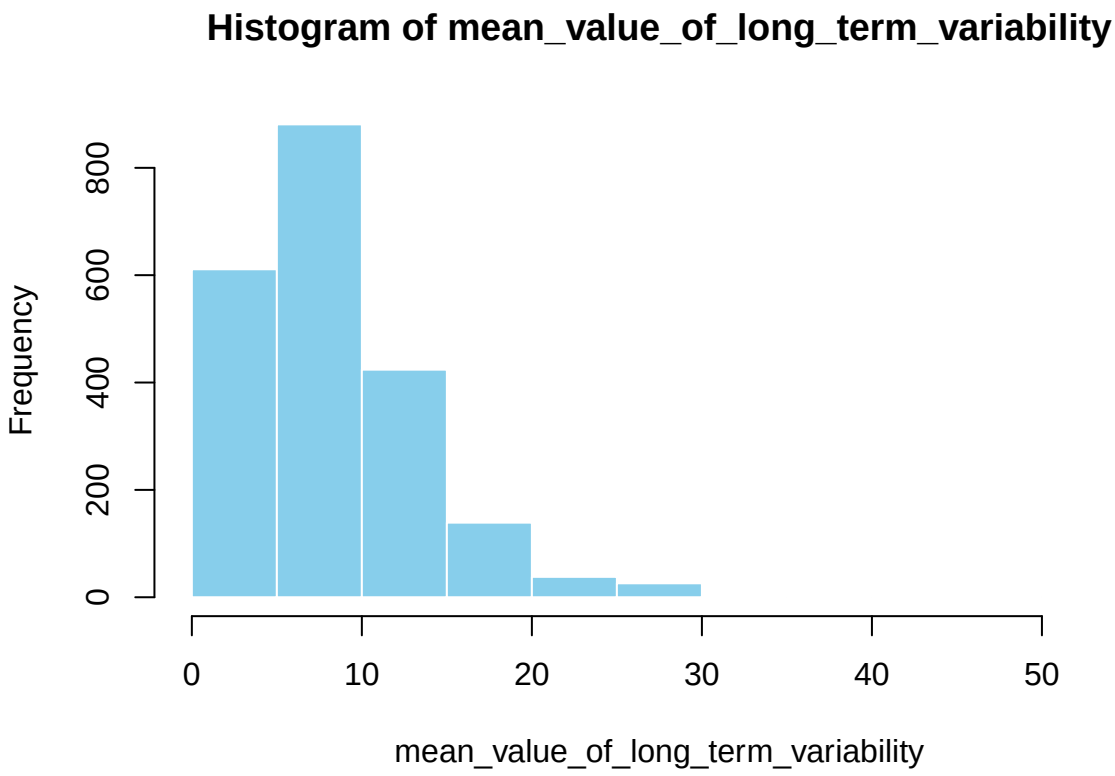

Histogram of severe_decelerations
Frequency
0
500
1000
1500
2000
0e+00
2e−04
4e−04
6e−04
8e−04
1e−03
severe_decelerations

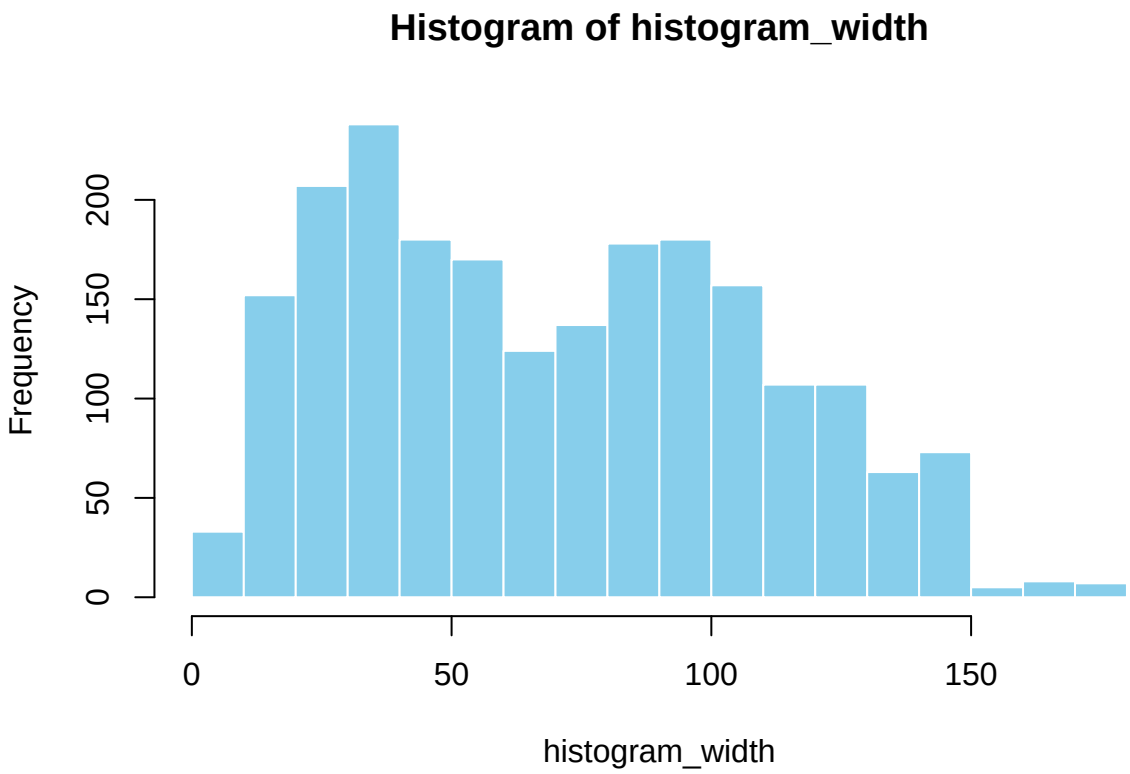

Histogram of prolongued_decelerations
Frequency
0
500
1000
1500
2000
0.000
0.001
0.002
0.003
0.004
0.005
prolongued_decelerations

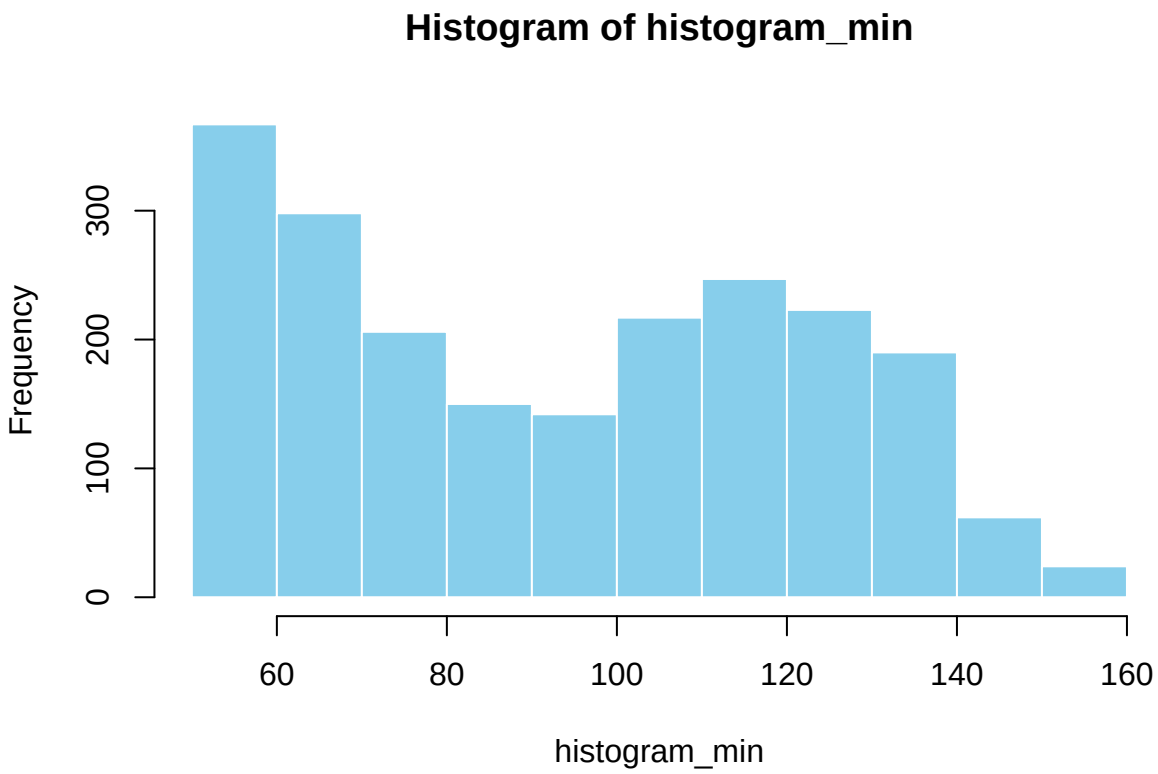

Histogram of abnormal_short_term_variability
Frequency
0
50
100
150
200
250
20
40
60
80
abnormal_short_term_variability

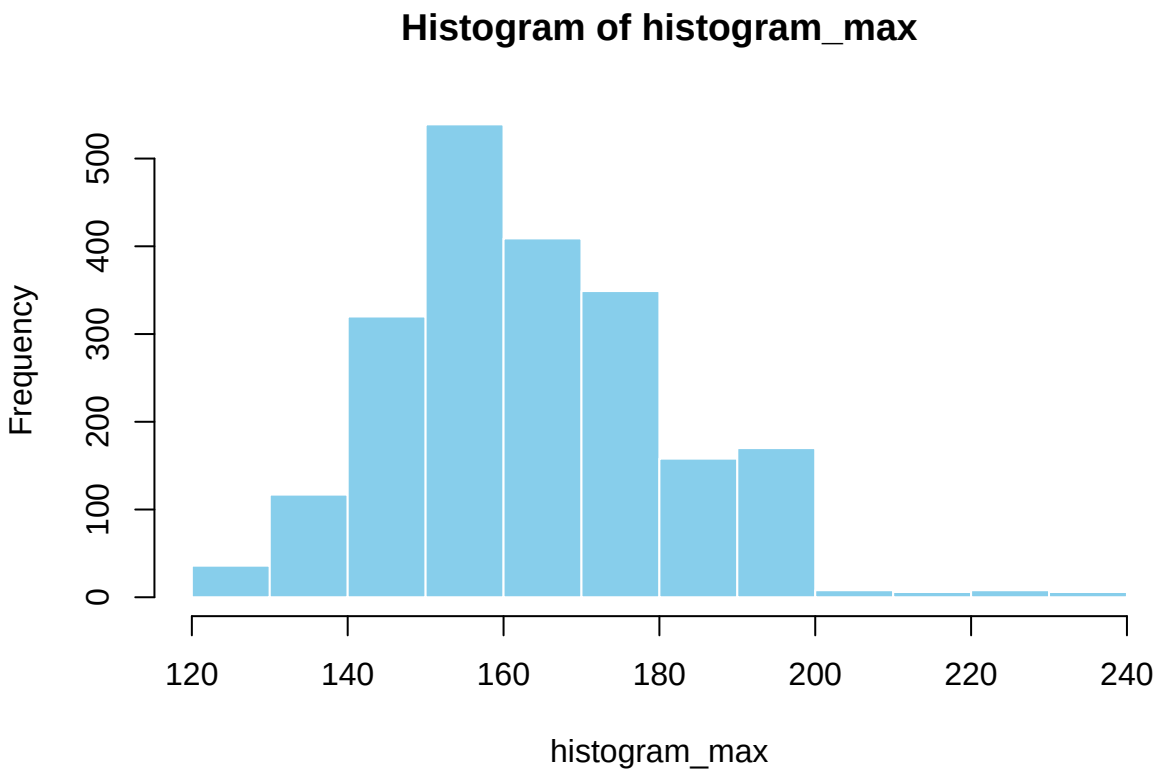

Histogram of mean_value_of_short_term_variability
Frequency
0
100
200
300
400
500
0
1
2
3
4
5
6
7
mean_value_of_short_term_variability

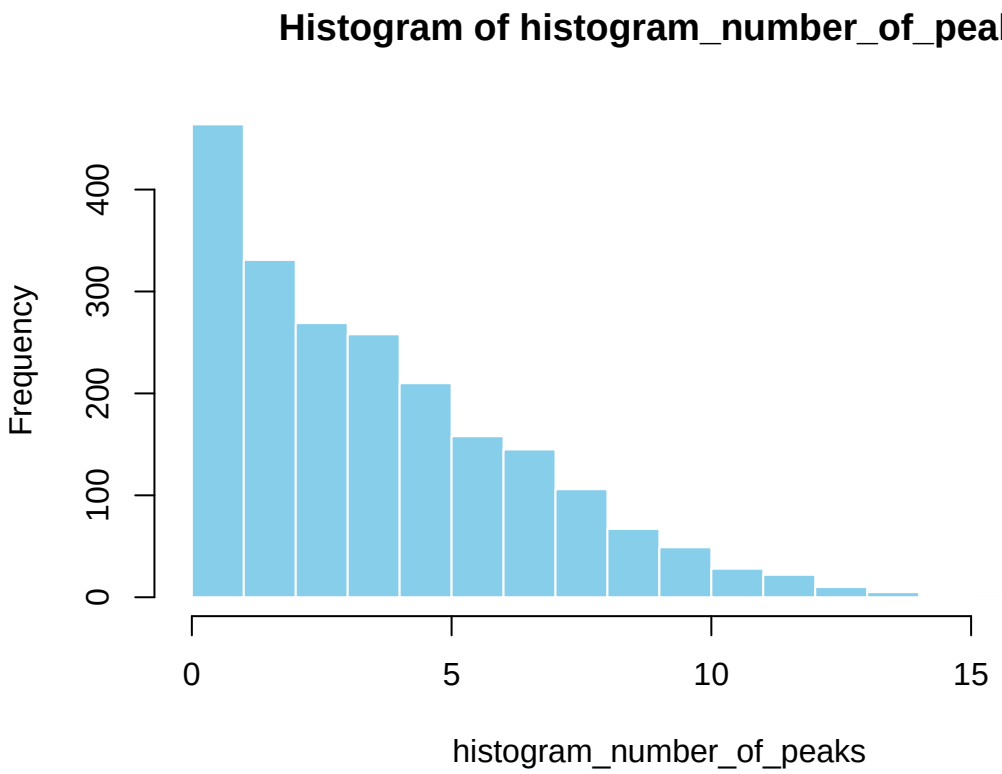

Histogram of percentage_of_time_with_abnormal_long_term_variabi
Frequency
0
500
1000
1500
0
20
40
60
80
percentage_of_time_with_abnormal_long_term_variability

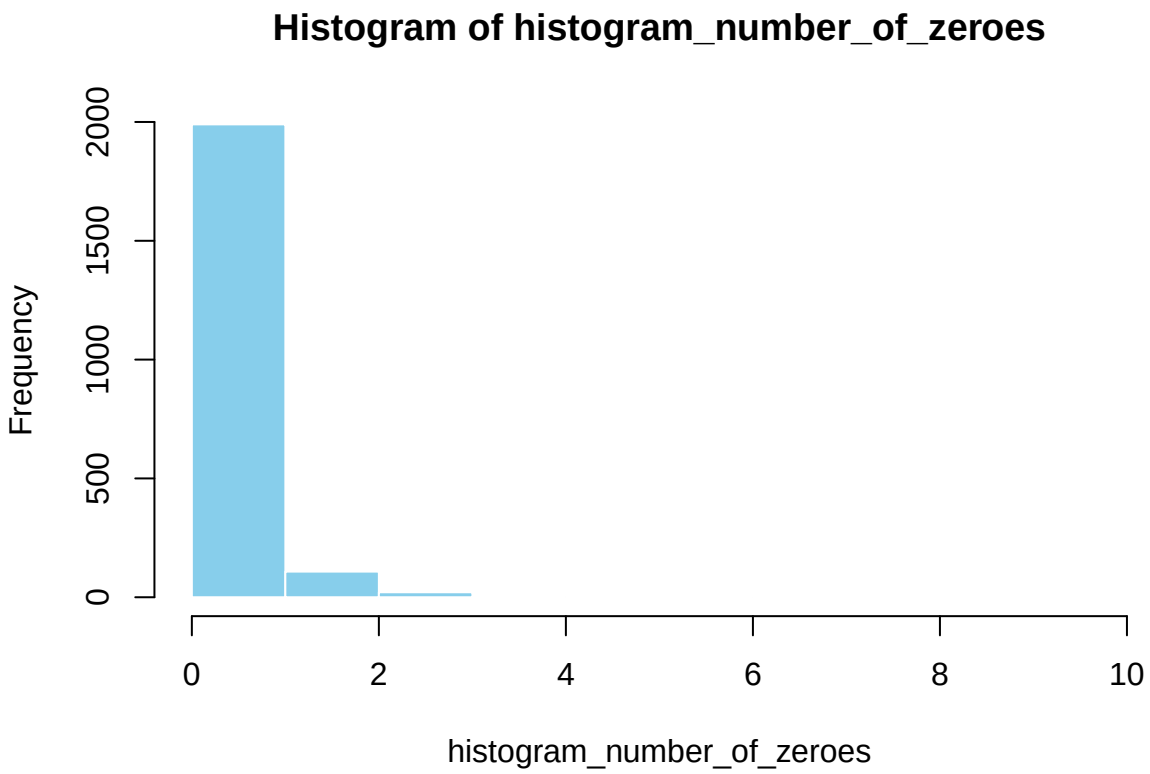

Histogram of mean_value_of_long_term_variability
Frequency
0
200
400
600
800
0
10
20
30
40
50
mean_value_of_long_term_variability

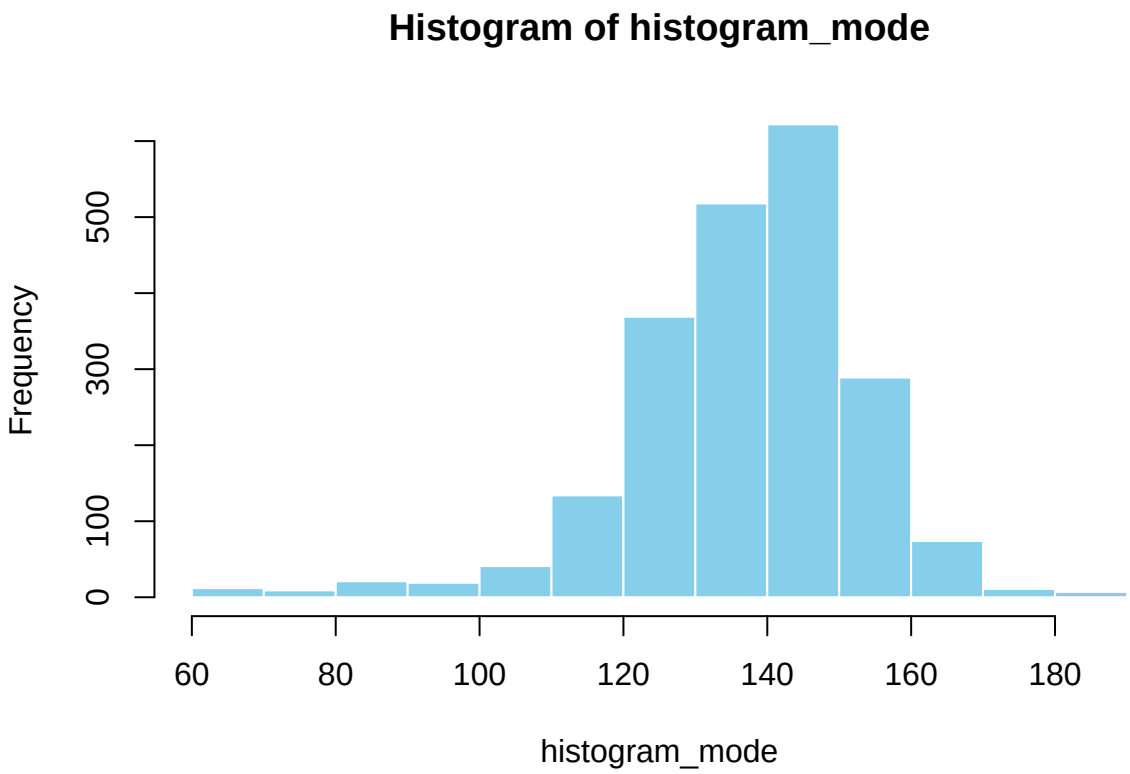

Histogram of histogram_width
Frequency
0
50
100
150
200
0
50
100
150
histogram_width

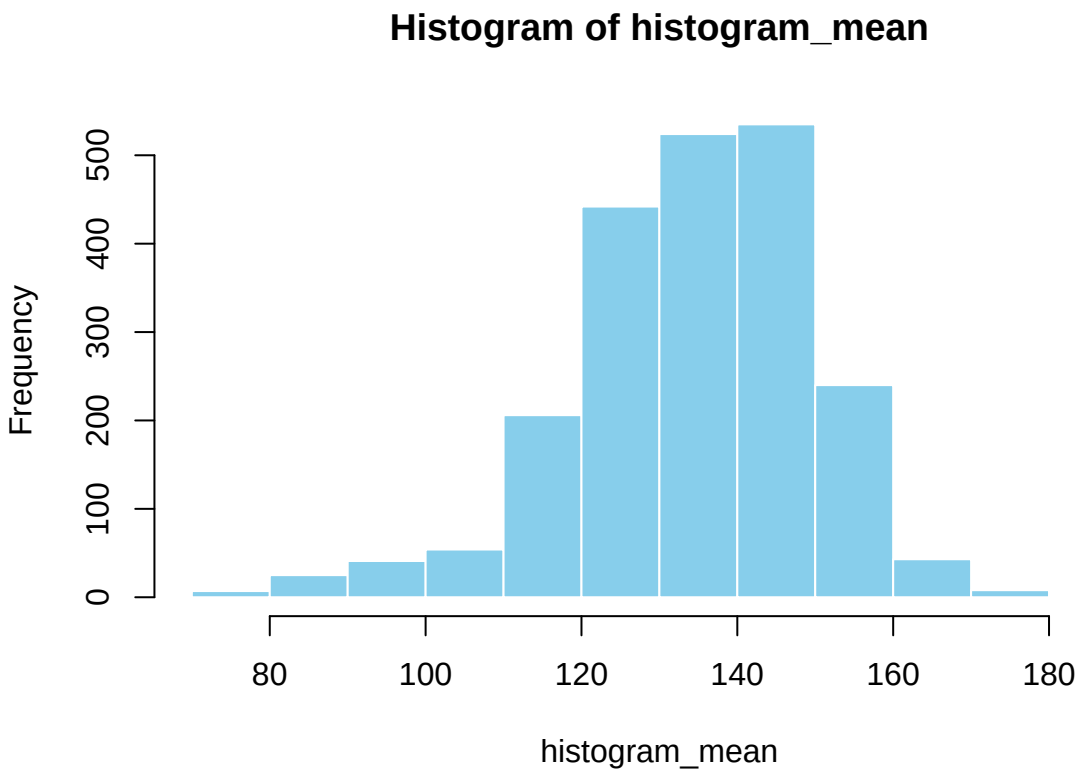

Histogram of histogram_min
Frequency
0
100
200
300
60
80
100
120
140
160
histogram_min

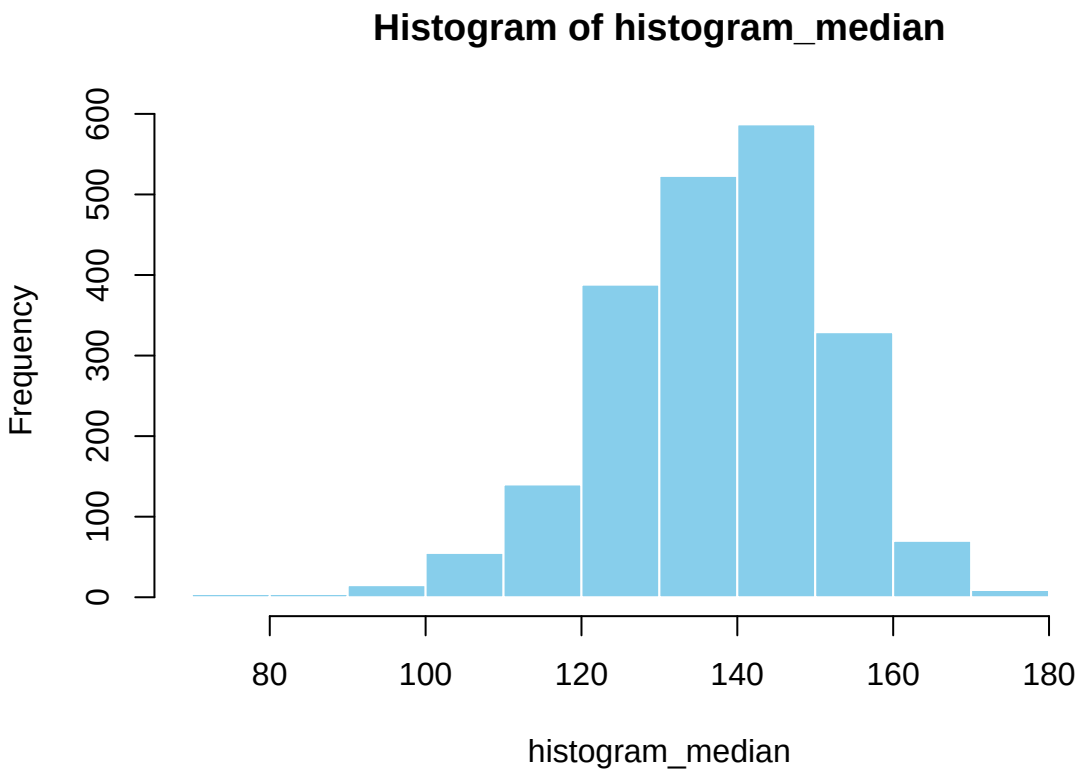

Histogram of histogram_max
Frequency
0
100
200
300
400
500
120
140
160
180
200
220
240
histogram_max

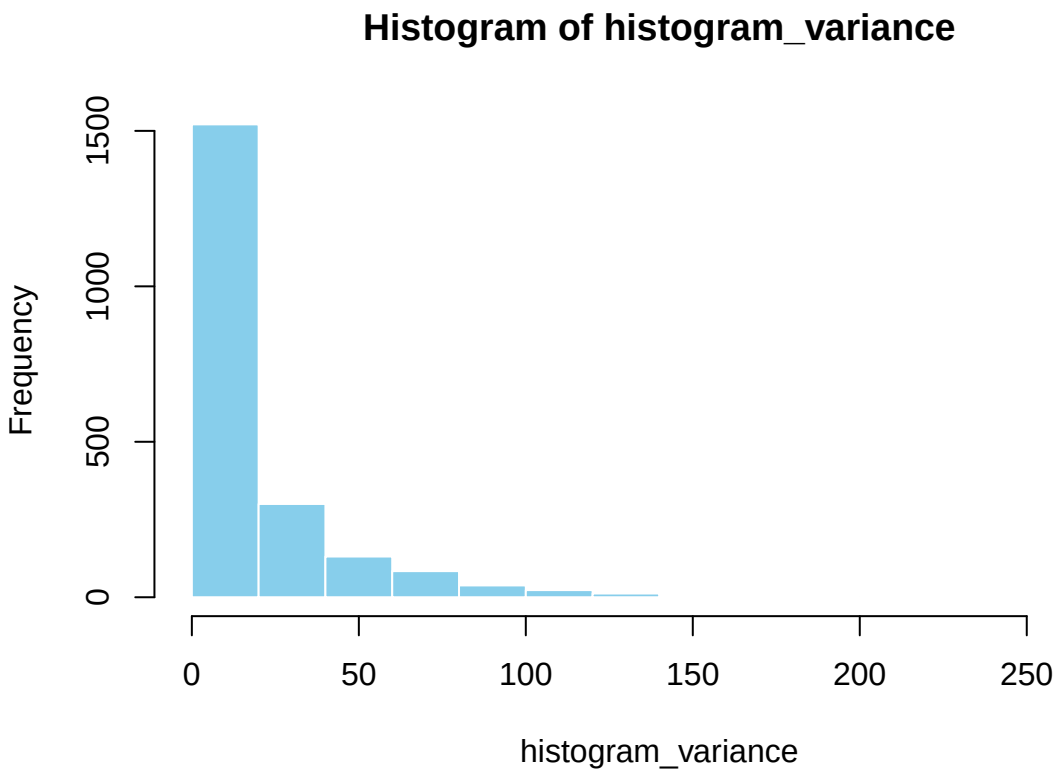

Histogram of histogram_number_of_peaks
Frequency
0
100
200
300
400
0
5
10
15
histogram_number_of_peaks

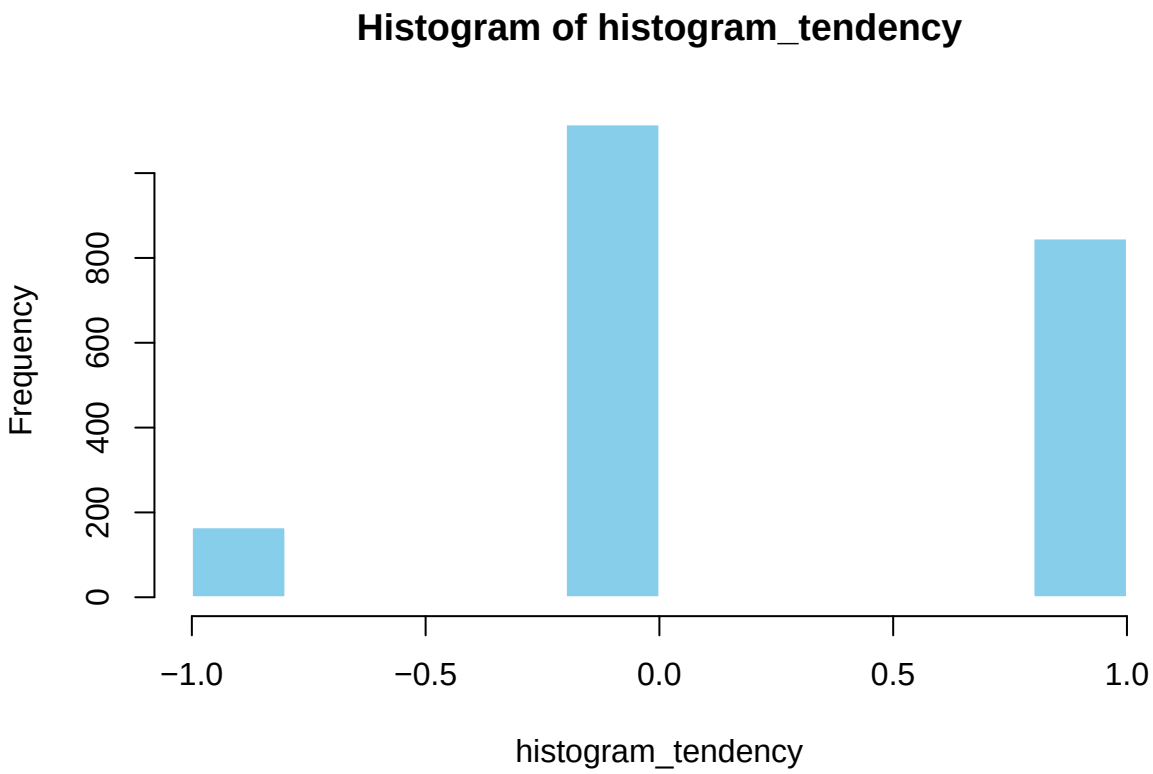

Histogram of histogram_number_of_zeroes
Frequency
0
500
1000
1500
2000
0
2
4
6
8
10
histogram_number_of_zeroes

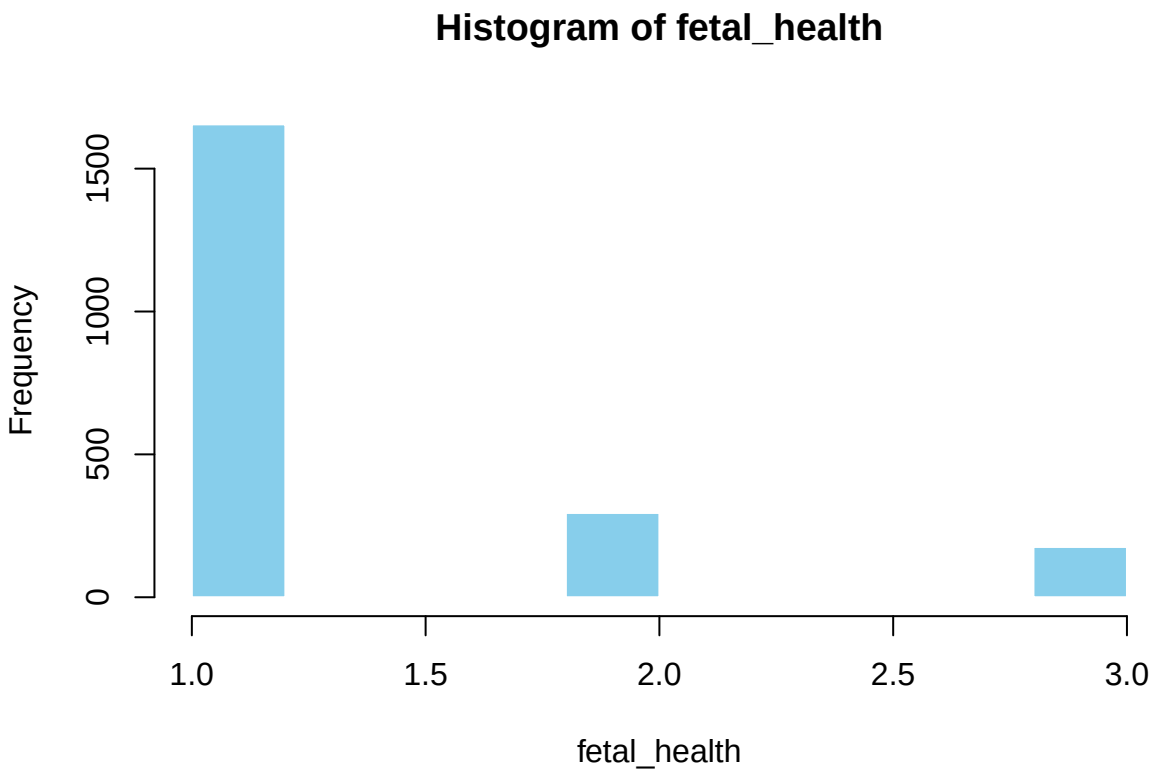

Histogram of histogram_mode
Frequency
0
100
300
500
60
80
100
120
140
160
180
histogram_mode


Histogram of histogram_mean
Frequency
0
100
200
300
400
500
80
100
120
140
160
180
histogram_mean


Histogram of histogram_median
Frequency
0
100
200
300
400
500
600
80
100
120
140
160
180
histogram_median


Histogram of histogram_variance
Frequency
0
500
1000
1500
0
50
100
150
200
250
histogram_variance


Histogram of histogram_tendency
Frequency
0
200
400
600
800
−1.0
−0.5
0.0
0.5
1.0
histogram_tendency



It is fairly clear that many of these variables do not follow multivariate normality. We can further test this using a Mardia test.

```
library(MVN)
MVN::mvn(data = data, mvnTest = "mardia")[2]
```

```
## $univariateNormality
##                Test                                                 Variable
## 1  Anderson-Darling                                    baseline value
## 2  Anderson-Darling                                    accelerations
## 3  Anderson-Darling                                    fetal_movement
## 4  Anderson-Darling                                  uterine_contractions
## 5  Anderson-Darling                                  light_decelerations
## 6  Anderson-Darling                                  severe_decelerations
## 7  Anderson-Darling                                prolongued_decelerations
## 8  Anderson-Darling                             abnormal_short_term_variability
## 9  Anderson-Darling                           mean_value_of_short_term_variability
## 10 Anderson-Darling percentage_of_time_with_abnormal_long_term_variability
## 11 Anderson-Darling           mean_value_of_long_term_variability
## 12 Anderson-Darling                                    histogram_width
## 13 Anderson-Darling                                    histogram_min
## 14 Anderson-Darling                                    histogram_max
## 15 Anderson-Darling                                histogram_number_of_peaks
## 16 Anderson-Darling                                histogram_number_of_zeroes
## 17 Anderson-Darling                                    histogram_mode
## 18 Anderson-Darling                                    histogram_mean
```

```
## 19 Anderson-Darling                    histogram_median
## 20 Anderson-Darling                   histogram_variance
## 21 Anderson-Darling                   histogram_tendency
## 22 Anderson-Darling                      fetal_health
##    Statistic   p value Normality
## 1     3.6904  <0.001      NO
## 2   142.9413  <0.001      NO
## 3   628.6448  <0.001      NO
## 4    24.6105  <0.001      NO
## 5   258.4877  <0.001      NO
## 6   818.9458  <0.001      NO
## 7   672.0443  <0.001      NO
## 8    23.2882  <0.001      NO
## 9    49.2204  <0.001      NO
## 10  341.2727  <0.001      NO
## 11   25.0805  <0.001      NO
## 12   23.7914  <0.001      NO
## 13   41.1214  <0.001      NO
## 14    9.8115  <0.001      NO
## 15   46.4252  <0.001      NO
## 16  414.3285  <0.001      NO
## 17   18.5713  <0.001      NO
## 18   10.1837  <0.001      NO
## 19    5.9694  <0.001      NO
## 20  225.0260  <0.001      NO
## 21  242.5789  <0.001      NO
## 22  471.9679  <0.001      NO
```

Based on these outputs, we can clearly see that our data is not multivariate normal. I considered transforming some of the data using logarithms, but I decided to try fitting a model without transformation first. If we can properly transform the data, using LDA or QDA might be a viable choice for classification instead.

Basic summary statistics

```
data %>%
  summary()
```

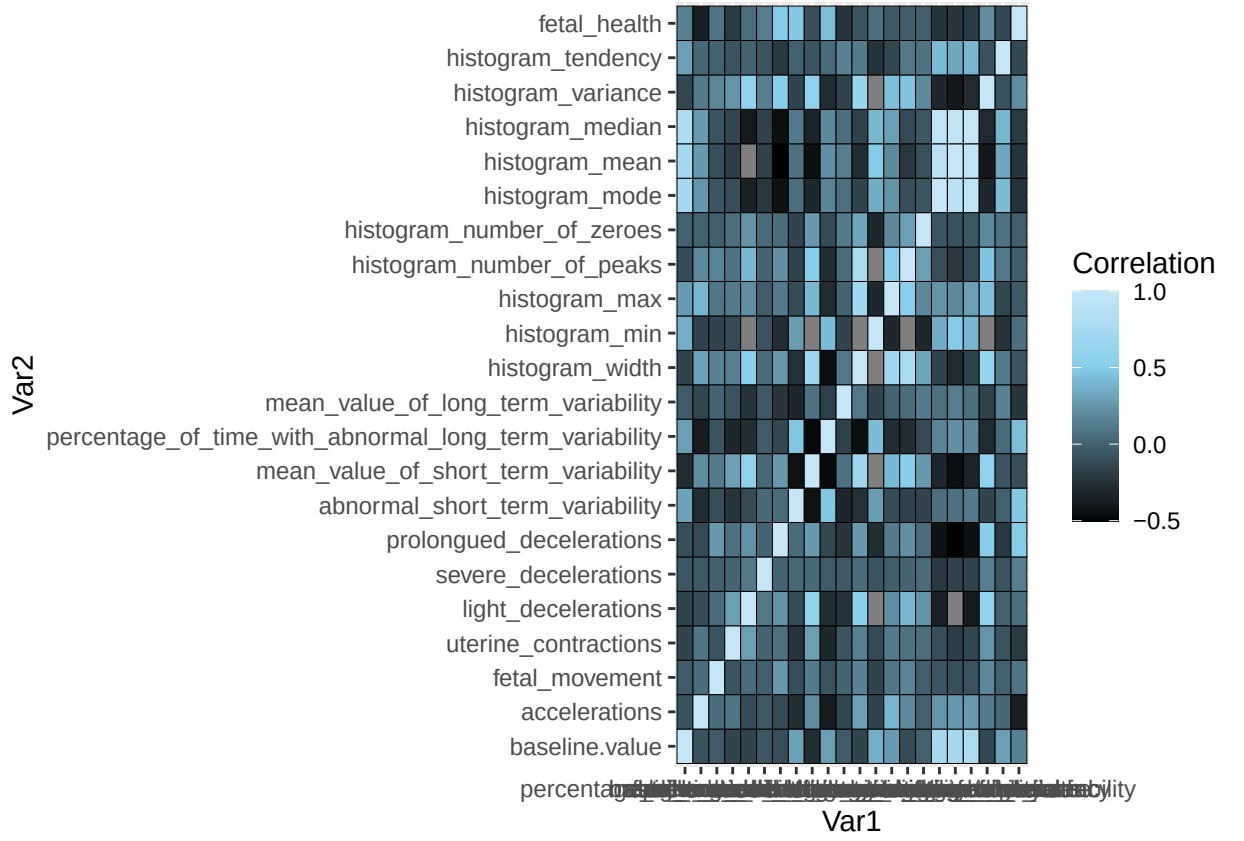
```
##  baseline value  accelerations      fetal_movement     uterine_contractions
##  Min.   :106.0   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:126.0   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.002000
##  Median :133.0   Median :0.002000   Median :0.000000   Median :0.004000
##  Mean   :133.3   Mean   :0.003178   Mean   :0.009481   Mean   :0.004366
##  3rd Qu.:140.0   3rd Qu.:0.006000   3rd Qu.:0.003000   3rd Qu.:0.007000
##  Max.   :160.0   Max.   :0.019000   Max.   :0.481000   Max.   :0.015000
##  light_decelerations severe_decelerations prolongued_decelerations
##  Min.   :0.000000    Min.   :0.000e+00    Min.   :0.0000000
##  1st Qu.:0.000000    1st Qu.:0.000e+00    1st Qu.:0.0000000
##  Median :0.000000    Median :0.000e+00    Median :0.0000000
##  Mean   :0.001889    Mean   :3.293e-06    Mean   :0.0001585
##  3rd Qu.:0.003000    3rd Qu.:0.000e+00    3rd Qu.:0.0000000
##  Max.   :0.015000    Max.   :1.000e-03    Max.   :0.0050000
##  abnormal_short_term_variability mean_value_of_short_term_variability
##  Min.   :12.00                   Min.   :0.200
##  1st Qu.:32.00                   1st Qu.:0.700
```

```
##  Median :49.00                   Median :1.200
##  Mean   :46.99                   Mean   :1.333
##  3rd Qu.:61.00                   3rd Qu.:1.700
##  Max.   :87.00                   Max.   :7.000
##  percentage_of_time_with_abnormal_long_term_variability
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 0.000
##  Mean   : 9.847
##  3rd Qu.:11.000
##  Max.   :91.000
##  mean_value_of_long_term_variability histogram_width  histogram_min
##  Min.   : 0.000                      Min.   :  3.00   Min.   : 50.00
##  1st Qu.: 4.600                      1st Qu.: 37.00   1st Qu.: 67.00
##  Median : 7.400                      Median : 67.50   Median : 93.00
##  Mean   : 8.188                      Mean   : 70.45   Mean   : 93.58
##  3rd Qu.:10.800                      3rd Qu.:100.00   3rd Qu.:120.00
##  Max.   :50.700                      Max.   :180.00   Max.   :159.00
##  histogram_max histogram_number_of_peaks histogram_number_of_zeroes
##  Min.   :122   Min.   : 0.000            Min.   : 0.0000
##  1st Qu.:152   1st Qu.: 2.000            1st Qu.: 0.0000
##  Median :162   Median : 3.000            Median : 0.0000
##  Mean   :164   Mean   : 4.068            Mean   : 0.3236
##  3rd Qu.:174   3rd Qu.: 6.000            3rd Qu.: 0.0000
##  Max.   :238   Max.   :18.000            Max.   :10.0000
##  histogram_mode  histogram_mean  histogram_median histogram_variance
##  Min.   : 60.0   Min.   : 73.0   Min.   : 77.0    Min.   :  0.00
##  1st Qu.:129.0   1st Qu.:125.0   1st Qu.:129.0    1st Qu.:  2.00
##  Median :139.0   Median :136.0   Median :139.0    Median :  7.00
##  Mean   :137.5   Mean   :134.6   Mean   :138.1    Mean   : 18.81
##  3rd Qu.:148.0   3rd Qu.:145.0   3rd Qu.:148.0    3rd Qu.: 24.00
##  Max.   :187.0   Max.   :182.0   Max.   :186.0    Max.   :269.00
##  histogram_tendency  fetal_health
##  Min.   :-1.0000    Min.   :1.000
##  1st Qu.: 0.0000    1st Qu.:1.000
##  Median : 0.0000    Median :1.000
##  Mean   : 0.3203    Mean   :1.304
##  3rd Qu.: 1.0000    3rd Qu.:1.000
##  Max.   : 1.0000    Max.   :3.000
```

Investigating correlations

```r
library(reshape2)
library(RColorBrewer)

df_numeric <- as.data.frame(lapply(data, as.numeric))

# Compute correlation matrix
cor_matrix <- cor(df_numeric, use = "pairwise.complete.obs")

# Melt the correlation matrix to long format
cor_melted <- melt(cor_matrix)

# Create heatmap
```

```
ggplot(cor_melted, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "black") +
  scale_fill_gradient2(
    low = "black", mid = "skyblue", high = "white",
    midpoint = 0.5, limit = c(-0.5, 1),
    name = "Correlation"
  )
```



There do not appear to be any particularly stand out values that correlate, aside from summary statistics (mean, median and mode of the histogram correlate highly).

Given that the fetal health column is recorded and can tell us if the infant was classified as normal, suspect or pathological, we should be able to create model that can classify into two groups - normal or of concern.

We can first create a new column by collapsing the suspect and pathological groups into a single group, and re-code the system to use 0 for normal and 1 for of concern.

```
data <- data %>%
  mutate(
    recoded = ifelse(fetal_health == 1, 0, 1)
  ) %>%
  dplyr::select(-fetal_health)

plot_data <- data
colnames(plot_data) <- paste0(seq_along(data))

cor_matrix <- cor(plot_data[, 1:21], use = "complete.obs")
```

```r
cor_matrix[abs(cor_matrix) < 0.7] <- NA
round(cor_matrix, 2)
```

```
##       1  2  3  4  5  6  7  8  9 10 11    12   13 14   15 16   17   18   19 20
## 1  1.00 NA NA NA NA NA NA NA NA NA NA    NA   NA NA   NA NA 0.71 0.72 0.79 NA
## 2    NA  1 NA NA NA NA NA NA NA NA NA    NA   NA NA   NA NA   NA   NA   NA NA
## 3    NA NA  1 NA NA NA NA NA NA NA NA    NA   NA NA   NA NA   NA   NA   NA NA
## 4    NA NA NA  1 NA NA NA NA NA NA NA    NA   NA NA   NA NA   NA   NA   NA NA
## 5    NA NA NA NA  1 NA NA NA NA NA NA    NA   NA NA   NA NA   NA   NA   NA NA
## 6    NA NA NA NA NA  1 NA NA NA NA NA    NA   NA NA   NA NA   NA   NA   NA NA
## 7    NA NA NA NA NA NA  1 NA NA NA NA    NA   NA NA   NA NA   NA   NA   NA NA
## 8    NA NA NA NA NA NA NA  1 NA NA NA    NA   NA NA   NA NA   NA   NA   NA NA
## 9    NA NA NA NA NA NA NA NA  1 NA NA    NA   NA NA   NA NA   NA   NA   NA NA
## 10   NA NA NA NA NA NA NA NA NA  1 NA    NA   NA NA   NA NA   NA   NA   NA NA
## 11   NA NA NA NA NA NA NA NA NA NA  1    NA   NA NA   NA NA   NA   NA   NA NA
## 12   NA NA NA NA NA NA NA NA NA NA NA  1.00 -0.9 NA 0.75 NA   NA   NA   NA NA
## 13   NA NA NA NA NA NA NA NA NA NA NA -0.90  1.0 NA   NA NA   NA   NA   NA NA
## 14   NA NA NA NA NA NA NA NA NA NA NA    NA   NA  1   NA NA   NA   NA   NA NA
## 15   NA NA NA NA NA NA NA NA NA NA NA  0.75   NA NA 1.00 NA   NA   NA   NA NA
## 16   NA NA NA NA NA NA NA NA NA NA NA    NA   NA NA   NA  1   NA   NA   NA NA
## 17 0.71 NA NA NA NA NA NA NA NA NA NA    NA   NA NA   NA NA 1.00 0.89 0.93 NA
## 18 0.72 NA NA NA NA NA NA NA NA NA NA    NA   NA NA   NA NA 0.89 1.00 0.95 NA
## 19 0.79 NA NA NA NA NA NA NA NA NA NA    NA   NA NA   NA NA 0.93 0.95 1.00 NA
## 20   NA NA NA NA NA NA NA NA NA NA NA    NA   NA NA   NA NA   NA   NA   NA  1
## 21   NA NA NA NA NA NA NA NA NA NA NA    NA   NA NA   NA NA   NA   NA   NA NA
##    21
## 1  NA
## 2  NA
## 3  NA
## 4  NA
## 5  NA
## 6  NA
## 7  NA
## 8  NA
## 9  NA
## 10 NA
## 11 NA
## 12 NA
## 13 NA
## 14 NA
## 15 NA
## 16 NA
## 17 NA
## 18 NA
## 19 NA
## 20 NA
## 21  1
```

```r
colnames(data[,c(12, 13, 15)])
```

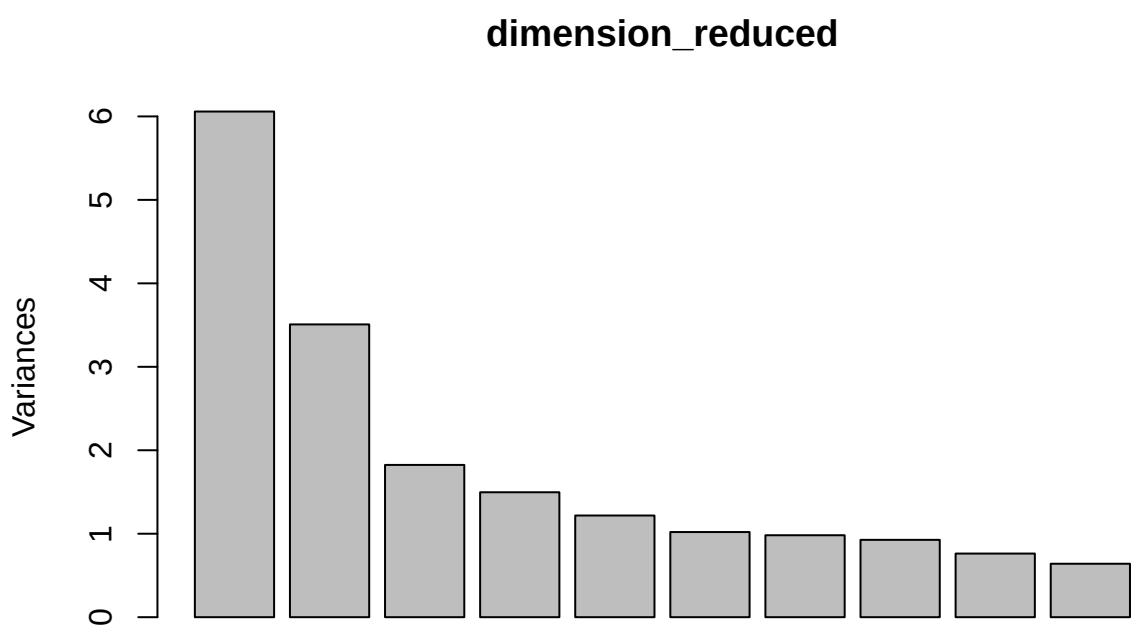
```
## [1] "histogram_width"           "histogram_min"
## [3] "histogram_number_of_peaks"
```

Rudimentary model fitting

From the variance inflation factor analysis above, we can see that there are a few variables that correlate very highly. In particular, we can see that 3x3 block in the lower right of the matrix. These are the mode, mean and median of the histogram, so a high level of multicolinearity is expected. Similarly, the other variables in high association with each other are histogram width, minimum and number of peaks, which we also expect.

For now, we shall leave these terms in our data. Since we are using principal components analysis, the resultant components constructed are orthogonal.

```
dimension_reduced <- prcomp(data[,1:21], scale = TRUE, center = TRUE)

screeplot(dimension_reduced)
```




```
sum(dimension_reduced$sdev[1:5]^2)/sum(dimension_reduced$sdev^2)
```

```
## [1] 0.671692
```

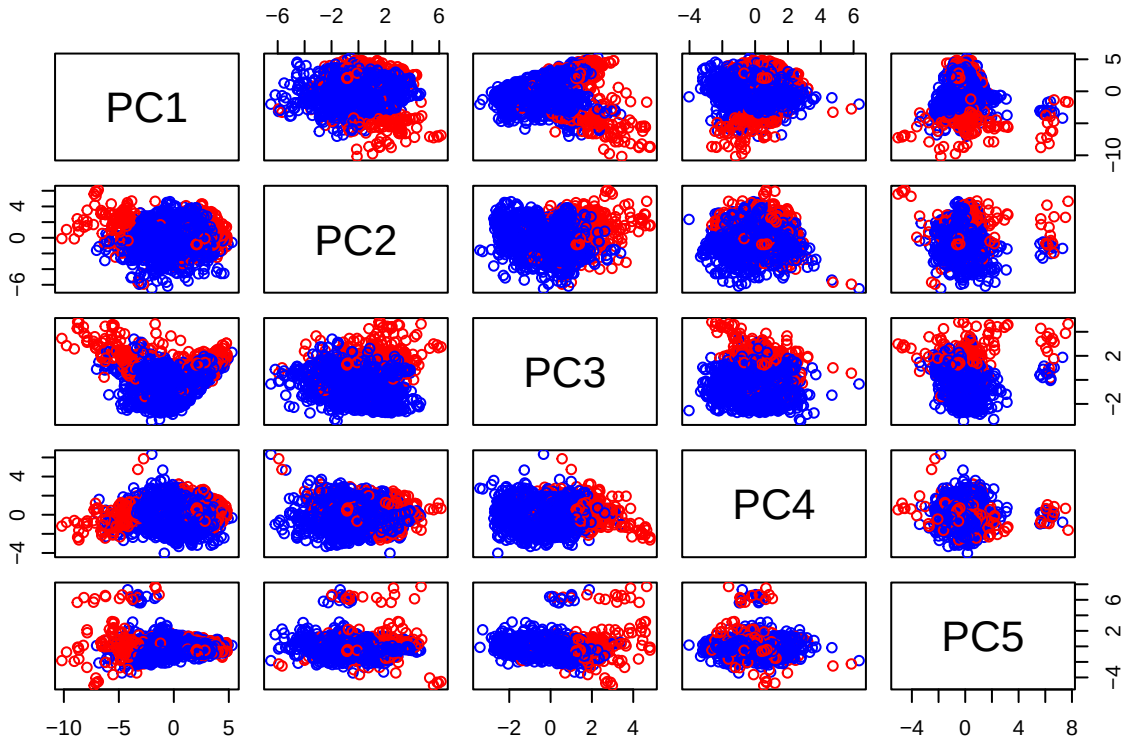
```
round(dimension_reduced$sdev^2 / sum(dimension_reduced$sdev^2), 2)
```

```
##  [1] 0.29 0.17 0.09 0.07 0.06 0.05 0.05 0.04 0.04 0.03 0.03 0.02 0.02 0.02 0.01
## [16] 0.01 0.01 0.01 0.00 0.00 0.00
```

Based on the screeplot and proportion of variance explained, using four of five principal components seems to be a viable option. Given this, we can now begin trying to fit a logistic model.

Let us now consider if we can find any meaningful separation using the principal components.

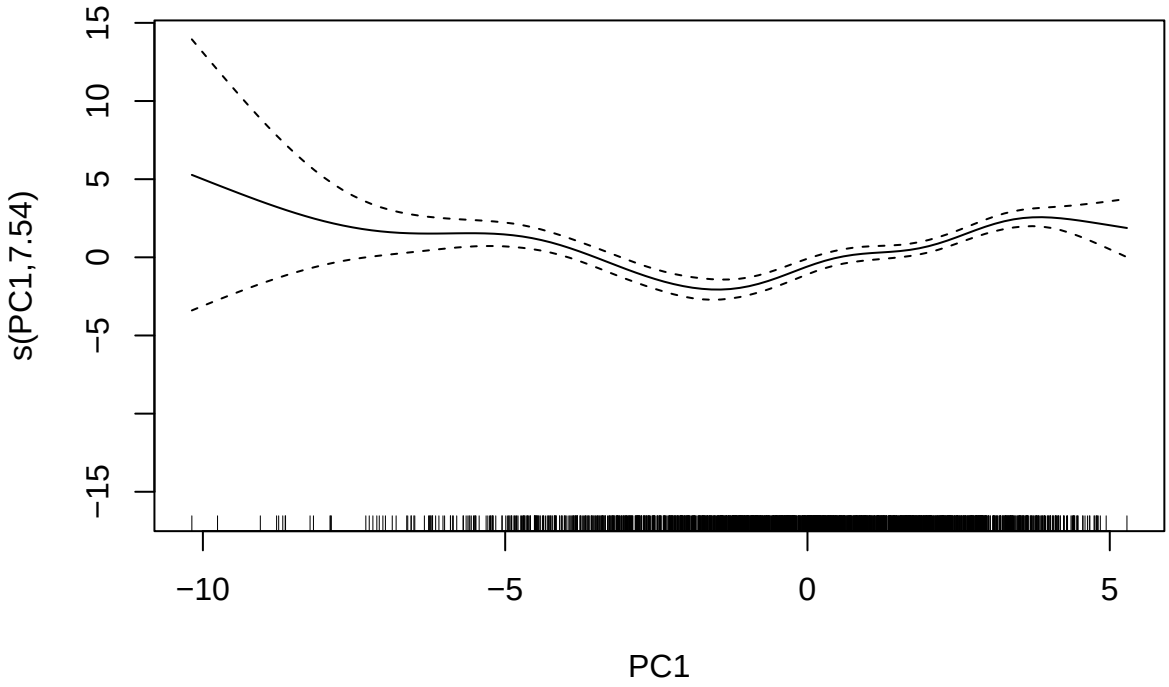
```
colours <- ifelse(data$recoded == 1, 'red', 'blue')
pairs(dimension_reduced$x[,1:5], col=colours)
```

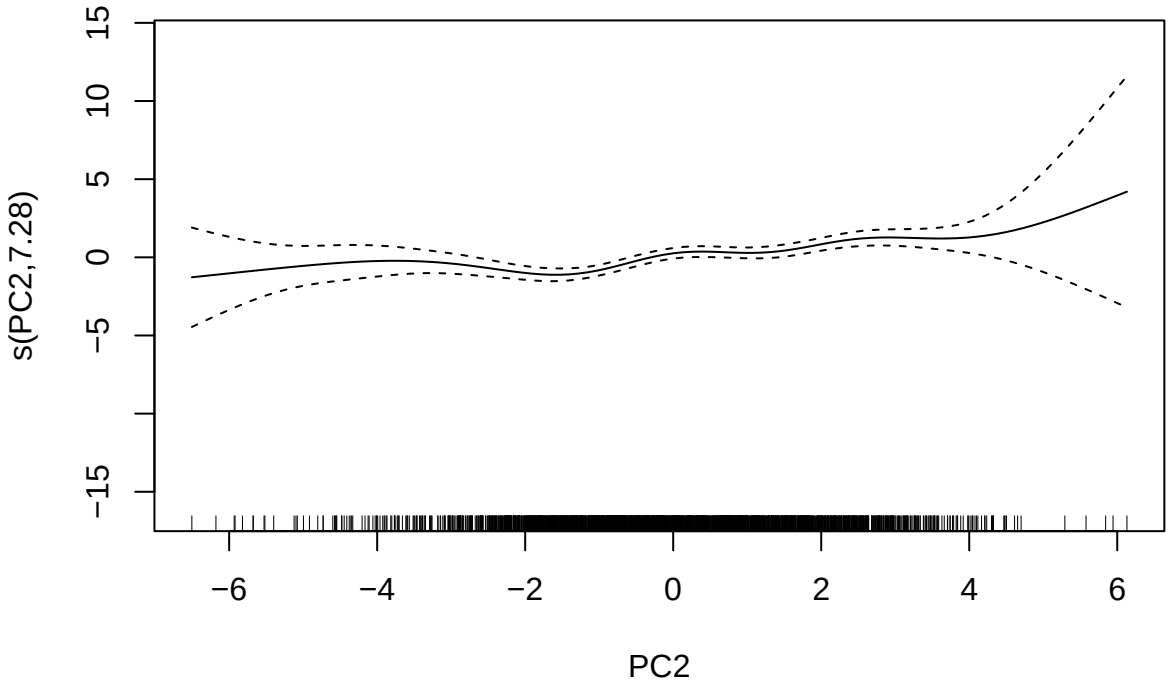


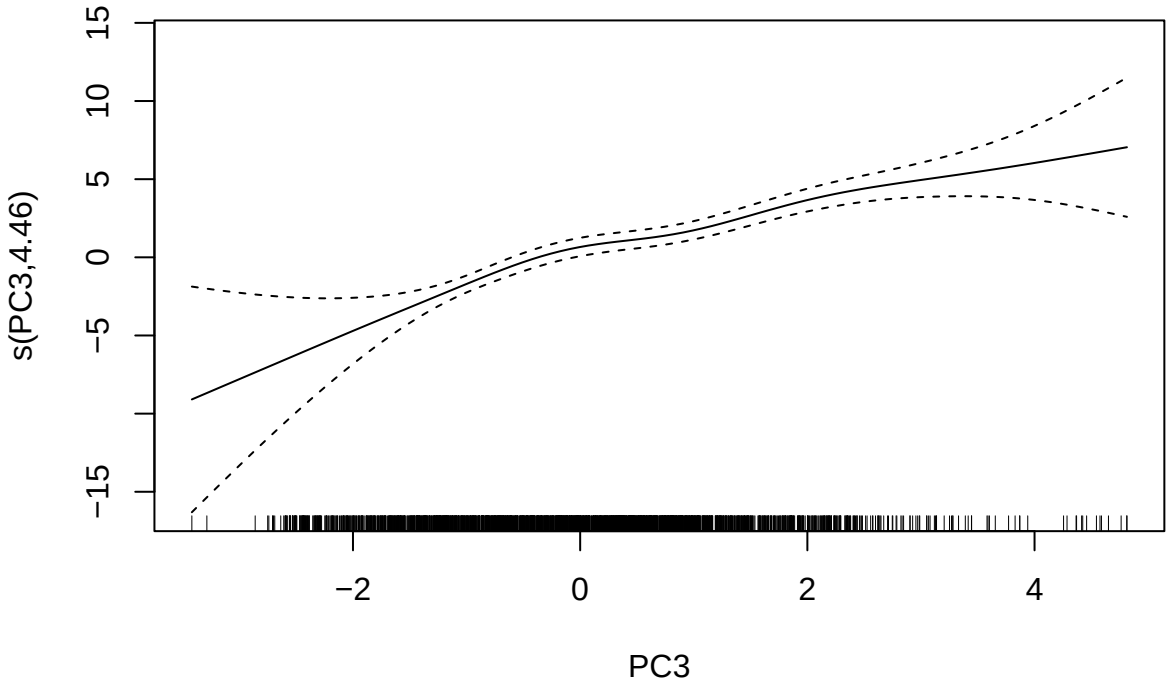
Based on the plot, we can see that there is significant overlap in many of the principal component pairings, However, there do appear to be a few pairings when there is better separtion than others, such as all of the PC3 groupings. Additionally, all of the PC5 groupings show a smaller cluster that appears to be a mix of normal and of concern infants. This may be worth further investigation at another time, but unfortunately does not aim in solving our problem.

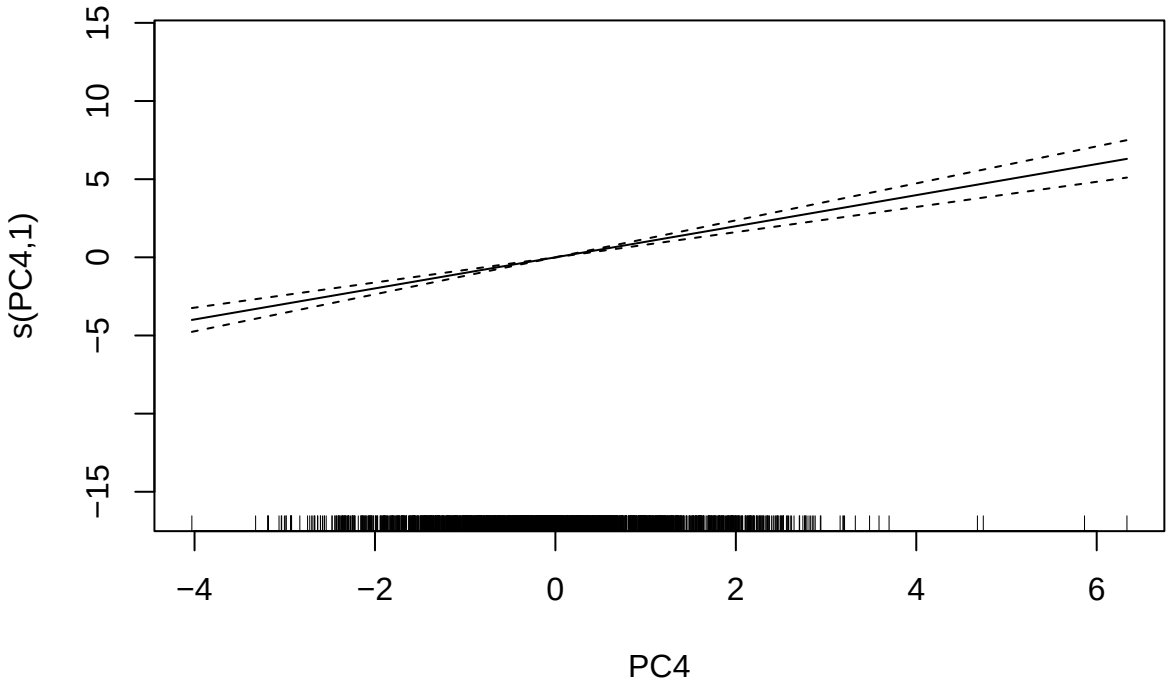
In our next step we will attempt to fit the logistic model, which will involve dredging over all the possible combinations. I have chosen to use AICc as my dredging criteria. We will first consider if any of our derived variables require transformation.

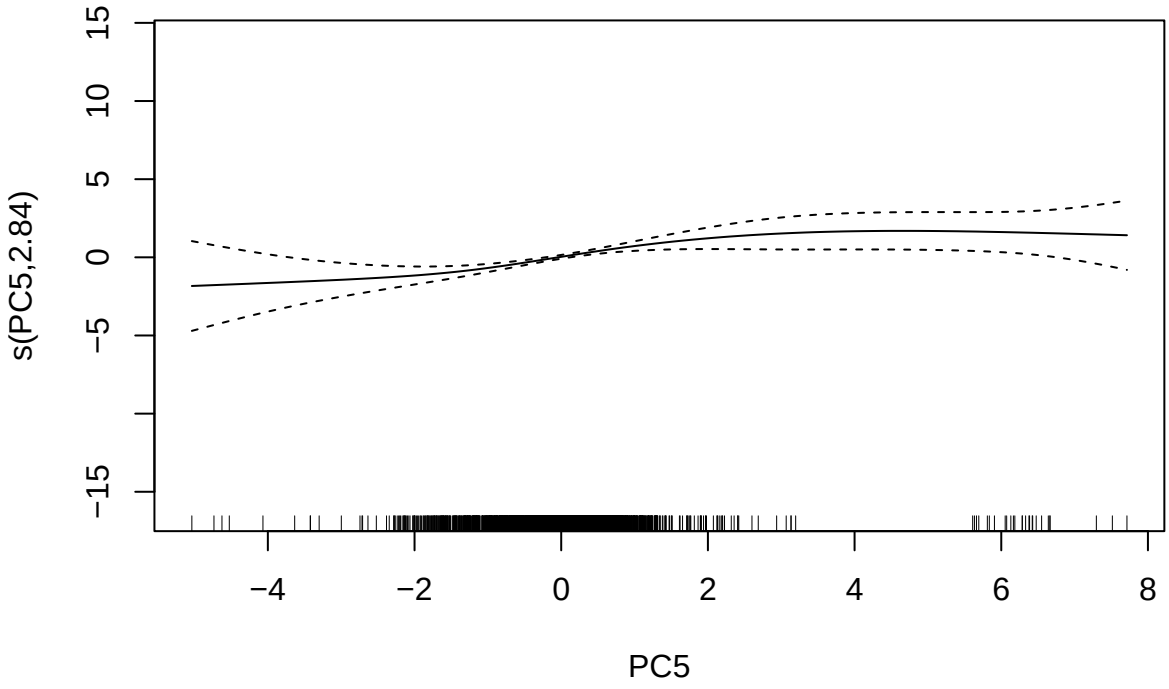
```
library(mgcv)

derived_data <- as.data.frame(dimension_reduced$x[,1:5])
derived_data$code <- data$recoded

gam.fit <- gam(code ~ s(PC1) + s(PC2) + s(PC3) + s(PC4) + s(PC5), family = "binomial", data = derived_da

plot(gam.fit)
```


s(PC1,7.54)
15
10
5
0
−5
−15
−10
−5
0
5
PC1


s(PC2,7.28)
15
10
5
0
−5
−15
−6
−4
−2
0
2
4
6
PC2


s(PC3,4.46)
15
10
5
0
−5
−15
−2
0
2
4
PC3


s(PC4,1)
15
10
5
0
−5
−15
−4
−2
0
2
4
6
PC4



It does not appear like our data requires any transformations, although the first principal component suggests possibly transforming with to a 7th or 8th degree, which I have chosen not to do, as visually it does not appear especially strong. Now we can move on to dredging our model. I intend to avoid any interaction terms with more than two variables, as it makes interpretation of the model more difficult.

```
library(MuMIn)

anova(glm(code ~  PC1 * PC2 * PC3 * PC4 * PC5, data = derived_data, family = binomial), test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: code
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                 2125    2248.69
## PC1                  1    36.84      2124    2211.85 1.284e-09 ***
## PC2                  1   127.26      2123    2084.59 < 2.2e-16 ***
## PC3                  1   940.01      2122    1144.57 < 2.2e-16 ***
## PC4                  1    83.86      2121    1060.71 < 2.2e-16 ***
## PC5                  1    36.95      2120    1023.77 1.214e-09 ***
## PC1:PC2              1    52.47      2119     971.29 4.365e-13 ***
```

```
## PC1:PC3              1    24.86      2118     946.43 6.160e-07 ***
## PC2:PC3              1    13.69      2117     932.74 0.0002152 ***
## PC1:PC4              1     8.15      2116     924.59 0.0043026 **
## PC2:PC4              1     0.86      2115     923.73 0.3543528
## PC3:PC4              1     0.46      2114     923.27 0.4999181
## PC1:PC5              1     7.01      2113     916.26 0.0080988 **
## PC2:PC5              1    14.17      2112     902.10 0.0001672 ***
## PC3:PC5              1     3.11      2111     898.99 0.0779665 .
## PC4:PC5              1     5.33      2110     893.66 0.0209644 *
## PC1:PC2:PC3          1     1.61      2109     892.05 0.2046779
## PC1:PC2:PC4          1     1.59      2108     890.46 0.2068701
## PC1:PC3:PC4          1     1.91      2107     888.54 0.1664591
## PC2:PC3:PC4          1     7.57      2106     880.98 0.0059439 **
## PC1:PC2:PC5          1    13.47      2105     867.50 0.0002422 ***
## PC1:PC3:PC5          1     8.62      2104     858.89 0.0033313 **
## PC2:PC3:PC5          1    19.76      2103     839.12 8.763e-06 ***
## PC1:PC4:PC5          1     5.14      2102     833.98 0.0233177 *
## PC2:PC4:PC5          1     2.95      2101     831.03 0.0857014 .
## PC3:PC4:PC5          1     9.87      2100     821.16 0.0016824 **
## PC1:PC2:PC3:PC4      1     6.43      2099     814.73 0.0112045 *
## PC1:PC2:PC3:PC5      1     3.08      2098     811.64 0.0791515 .
## PC1:PC2:PC4:PC5      1     1.38      2097     810.26 0.2395219
## PC1:PC3:PC4:PC5      1     0.49      2096     809.77 0.4825395
## PC2:PC3:PC4:PC5      1     0.09      2095     809.67 0.7593811
## PC1:PC2:PC3:PC4:PC5  1     2.61      2094     807.07 0.1065096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

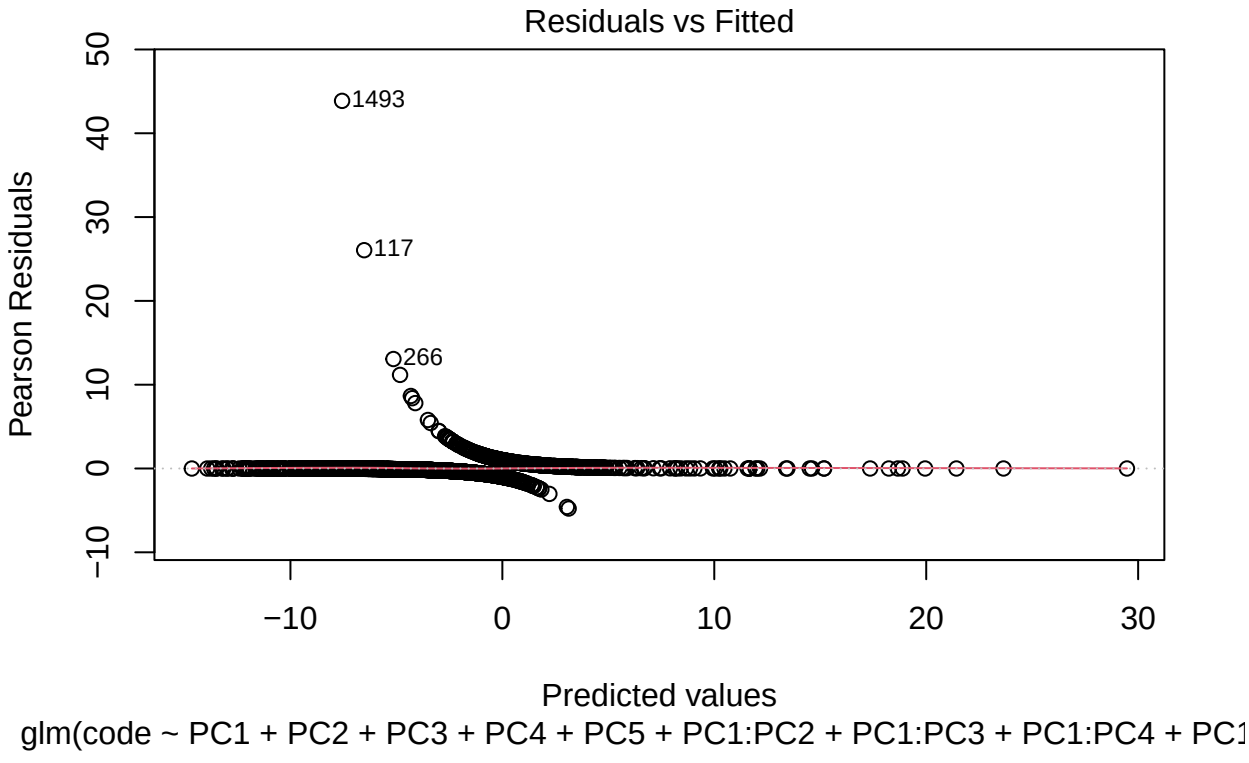
```
full_model <- glm(code ~ PC1 * PC2 + PC1 * PC3 + PC2 * PC3 + PC1 * PC4 + PC1 * PC5 + PC2 * PC5 + PC4 * P

options(na.action = "na.fail")

dredged_model <- dredge(full_model)

chosen_model <- get.models(dredged_model, 2)[[1]]

summary(chosen_model)
```

```
##
## Call:
## glm(formula = code ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC1:PC2 +
##     PC1:PC3 + PC1:PC4 + PC1:PC5 + PC2:PC3 + PC2:PC5 + 1, family = binomial,
##     data = derived_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.17047    0.19293 -16.433  < 2e-16 ***
## PC1          0.67182    0.08958   7.499 6.41e-14 ***
## PC2          0.35880    0.07367   4.870 1.11e-06 ***
## PC3          2.16145    0.13011  16.613  < 2e-16 ***
## PC4          1.00190    0.10895   9.196  < 2e-16 ***
## PC5          0.89417    0.11302   7.911 2.55e-15 ***
## PC1:PC2     -0.16490    0.02475  -6.663 2.68e-11 ***
## PC1:PC3     -0.24797    0.05639  -4.398 1.09e-05 ***
```

```
## PC1:PC4     -0.10730    0.03866  -2.776 0.005505 **
## PC1:PC5      0.07436    0.03359   2.214 0.026851 *
## PC2:PC3      0.30506    0.06472   4.714 2.43e-06 ***
## PC2:PC5      0.22810    0.05916   3.856 0.000115 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2248.69  on 2125  degrees of freedom
## Residual deviance:  902.26  on 2114  degrees of freedom
## AIC: 926.26
##
## Number of Fisher Scoring iterations: 8
```

I ended up choosing the second model provided by dredge, as it had a comparable AICc to the first model, but the first model also included a non-significant interaction term while the second model does not.

```
deviance <- deviance(chosen_model)
df <- df.residual(chosen_model)

plot(chosen_model, which=1)
```

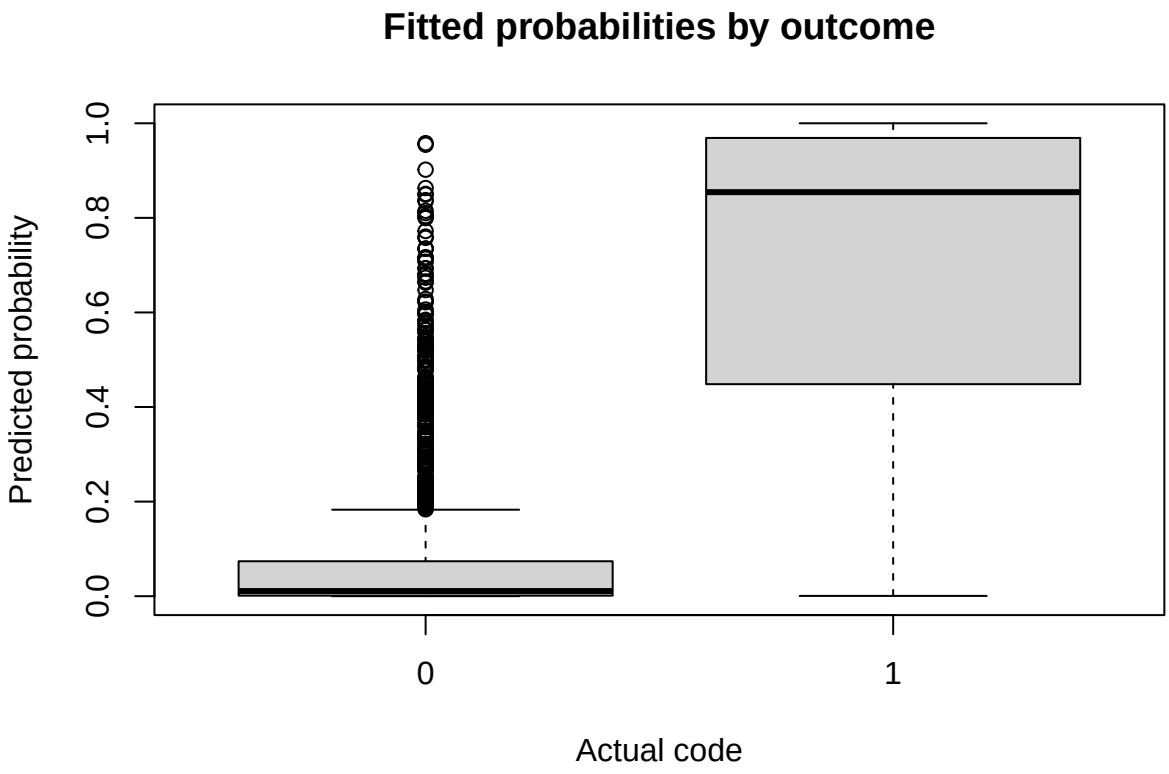



```
# p-value for goodness-of-fit test
1 - pchisq(deviance, df)
```

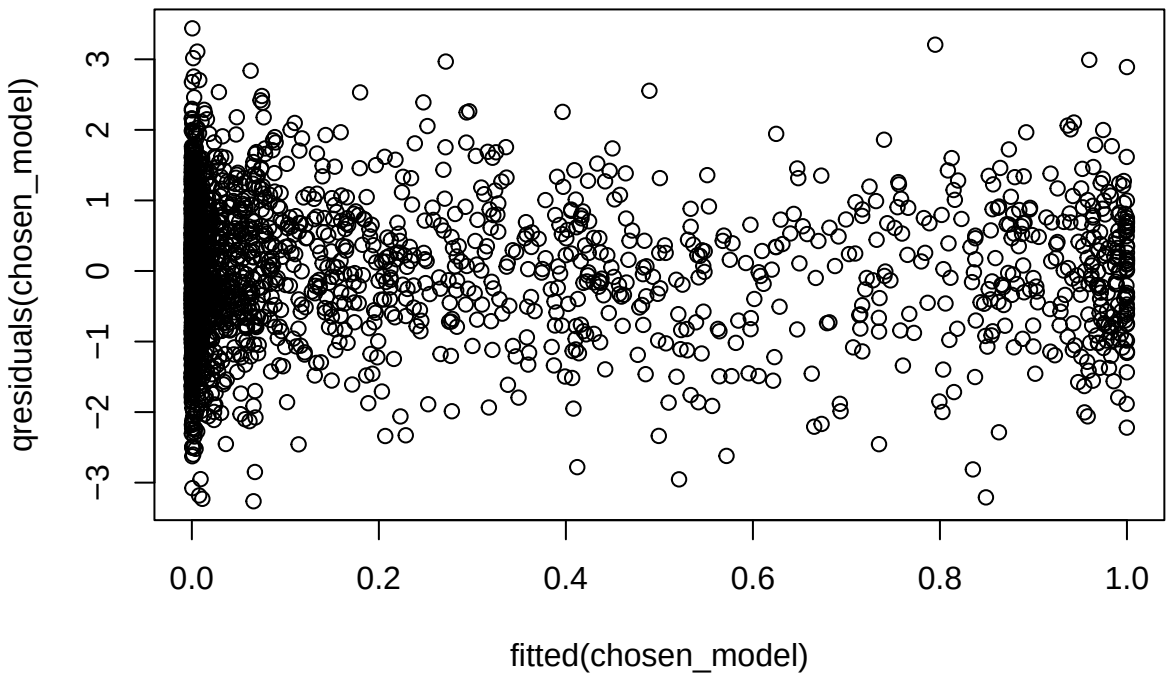
```
## [1] 1
```

Since we are using principal components, this goodness-of-fit test does not tell us much, but we can still check the randomised quantile residuals.

```
boxplot(fitted(chosen_model) ~ derived_data$code,
        main = "Fitted probabilities by outcome",
        xlab = "Actual code", ylab = "Predicted probability")
```




```
library(statmod)
plot(fitted(chosen_model), qresiduals(chosen_model))
```



Based on the outputs here, we can see that our residuals are mostly random noise, and that there is a clear separation between the groups or normal and of concern infants.

Lastly, let us consider our cross-validation to determine how well our model can predict the health of the infants.

```
library(crossval)

data <- as.data.frame(data)

predfun.lm <- function(train.x, train.y, test.x, test.y) {
  glm.fit <- glm(train.y ~ ., data = train.x, family = binomial)
  ynew <- predict(glm.fit, test.x, type = 'response')
  mean((ynew - test.y)^2)
}

y.fit1 <- "code"
cv.out = crossval(predfun.lm, X = derived_data[,c('PC1', 'PC2', 'PC3', 'PC4', 'PC5')], Y = derived_data
MSPE_1 <- cv.out$stat
MSPE_1se <- cv.out$stat.se

MSPE_1
```

```
## [1] 0.07500412
```

```
MSPE_1se
```

```
## [1] 0.00449518
```

So our model fits extremely well. To put our derived variables back into a more interpretable form, let us consider what each principal component may be representing, but investigating their loadings. It is especially of interest to see the loadings of PC3 as it has the greatest estimated coefficient.

```
dimension_reduced$rotation[, 1:5]%*%diag(dimension_reduced$sd[1:5]) -> new_scale
class(new_scale)<-"loadings"
new_scale
```

```
##
## Loadings:
##                                                    [,1]   [,2]   [,3]
## baseline value                                      0.552 -0.519  0.459
## accelerations                                      -0.107 -0.522 -0.252
## fetal_movement                                     -0.198         0.137
## uterine_contractions                               -0.314        -0.163
## light_decelerations                                -0.686         0.271
## severe_decelerations                               -0.142  0.147
## prolongued_decelerations                           -0.499  0.213  0.455
## abnormal_short_term_variability                     0.358  0.229  0.664
## mean_value_of_short_term_variability               -0.779 -0.253 -0.101
## percentage_of_time_with_abnormal_long_term_variability  0.534  0.264  0.433
## mean_value_of_long_term_variability                       -0.181 -0.584
## histogram_width                                    -0.772 -0.530
## histogram_min                                       0.817  0.267
## histogram_max                                      -0.329 -0.711  0.252
## histogram_number_of_peaks                          -0.618 -0.449  0.145
## histogram_number_of_zeroes                         -0.303 -0.238
## histogram_mode                                      0.646 -0.688
## histogram_mean                                      0.760 -0.610
## histogram_median                                    0.679 -0.705
## histogram_variance                                 -0.715 -0.188  0.359
## histogram_tendency                                  0.173 -0.423
##                                                    [,4]   [,5]
## baseline value
## accelerations                                      -0.533  0.116
## fetal_movement                                             0.700
## uterine_contractions                               -0.364 -0.509
## light_decelerations                                       -0.391
## severe_decelerations                                      -0.221
## prolongued_decelerations                           -0.118  0.312
## abnormal_short_term_variability                     0.113
## mean_value_of_short_term_variability
## percentage_of_time_with_abnormal_long_term_variability  0.259
## mean_value_of_long_term_variability                 0.463  0.211
## histogram_width                                     0.129
## histogram_min                                      -0.346
## histogram_max                                      -0.290
## histogram_number_of_peaks                           0.192  0.109
```

```
## histogram_number_of_zeroes                                  0.362 -0.224
## histogram_mode
## histogram_mean
## histogram_median
## histogram_variance                                         -0.117
## histogram_tendency                                          0.585 -0.126
##
##                 [,1]  [,2]  [,3]  [,4]  [,5]
## SS loadings    6.058 3.508 1.824 1.497 1.218
## Proportion Var 0.288 0.167 0.087 0.071 0.058
## Cumulative Var 0.288 0.456 0.542 0.614 0.672
```

PC1 - Possibly represents overall fetal heart rate level and distribution. Higher scores tend to correlate with lower variation, fewer decelerations, and higher histogram averages (mean, media, mode. - Could benefit from being mirrored from interpretability. - Loads lowly on mean value of long term variability, *only*.

PC2 - Appears to focus more on histogram shape and overall heart rate dispersion, loading very low on the histogram averages that PC1 loaded highly on. - Could be used as a constrast to PC1.

PC3 - Loads highest on variabilities and movements. - Does not have any especially large loadings.

PC4 - Loads highest on accelerations, histogram tendency and variability. - Does not load highly on decelerations at all. - Hardly loads on the histogram values.

PC5 - Loads very highly on fetal movement, followed by uterine contractions, although the latter is negatively loaded. - Loads against most decelerations except for prolongued decelerations. - Barely loads on any summary statistics, with almost none of the average/max/min/variability variables being loaded on.

Overall it is very difficult to definitively claim what any of these derived variables represent, especially given my limited knowledge of neo-natal care, and cardiology. However, what is clear from this analysis is that we can accurately classify children into of concern and healthy groupings, which may prove beneficial to health professionals. That said, there is a significant issue in the fact that the mathematics used to get to this conclusion has arrived at a fairly complex model based on derived components. Because of this, the it is not simple equation nor rule of thumb that a practitioner may use, and would in all likelihood require a computer with the model pre-loaded, and then require the inputting of all measurements, significantly delaying the process.

However, one useful piece of information revealed in this analysis is that the baseline value is important in the loadings of the first three principal components, accounting for much of the variation, but also that many of the histogram averages are only loaded by the first two prinicpal components, in opposit directions. This suggests that a less complicated model should be possible, perhaps based on this variables, but likely requiring more as well.

I noticed that in the PCA pairs plots, the third principal component seems to do a fairly good job at separation, so I decided to investigate and noticed that the highest loading on this component was abnormal short term viability. We will now fit a simple logistic regression model using just this predictor to see how well it performs alone.

```
abnormal_logistics <- glm(recoded ~ abnormal_short_term_variability, family=binomial, data = data)
summary(abnormal_logistics)
```

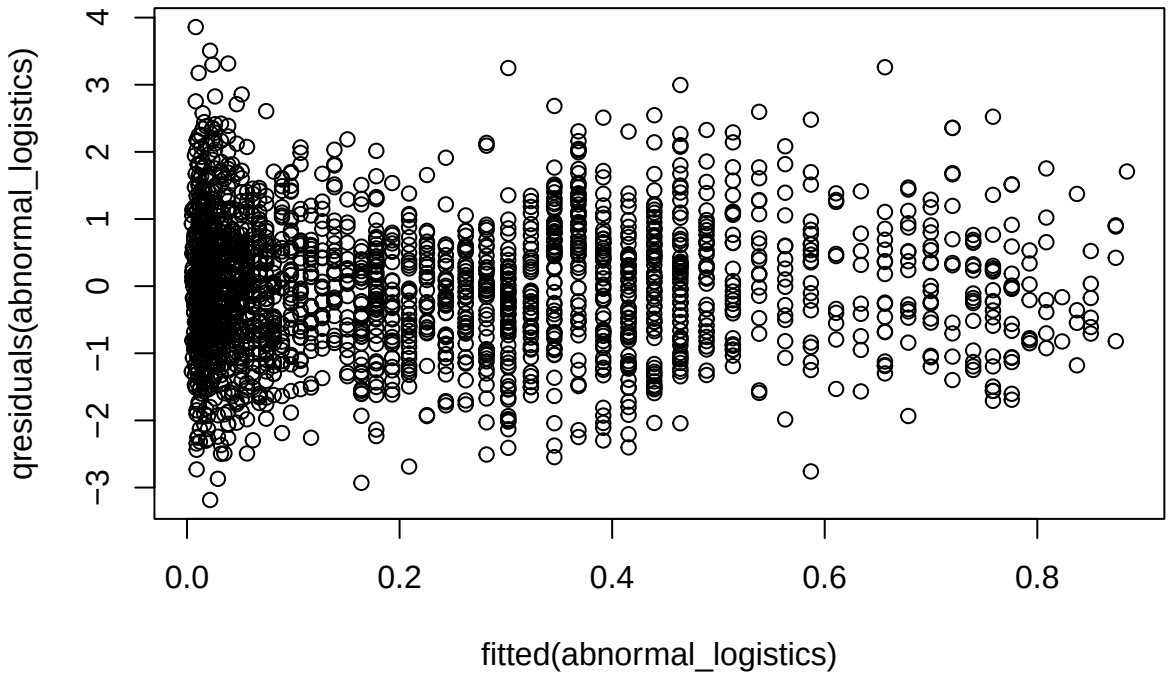
```
##
## Call:
## glm(formula = recoded ~ abnormal_short_term_variability, family = binomial,
##     data = data)
##
## Coefficients:
```

```
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -6.577585   0.314157  -20.94   <2e-16 ***
## abnormal_short_term_variability  0.098986   0.005267   18.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2248.7  on 2125  degrees of freedom
## Residual deviance: 1640.6  on 2124  degrees of freedom
## AIC: 1644.6
##
## Number of Fisher Scoring iterations: 6
```

```
deviance <- deviance(abnormal_logistics)
df <- df.residual(abnormal_logistics)
1 - pchisq(deviance, df) # fits very well, check the randomised quantile data
```

```
## [1] 1
```

```
plot(fitted(abnormal_logistics), qresiduals(abnormal_logistics)) # mostly white noise
```



The model appears fairly consistent with our checks, and also has a much lower AIC than our principal components based model, so let us now consider it's 10 fold cross validation outputs.

```
x.fit1 <- names(abnormal_logistics$coefficients)[-1]
y.fit1 <- "recoded"
cv.out = crossval(predfun.lm, X = as.data.frame(data[,"abnormal_short_term_variability"]), Y = data[, y
MSPE_1 <- cv.out$stat
MSPE_1se <- cv.out$stat.se

MSPE_1
```

```
## [1] 0.1205401
```

```
MSPE_1se
```

```
## [1] 0.002961143
```

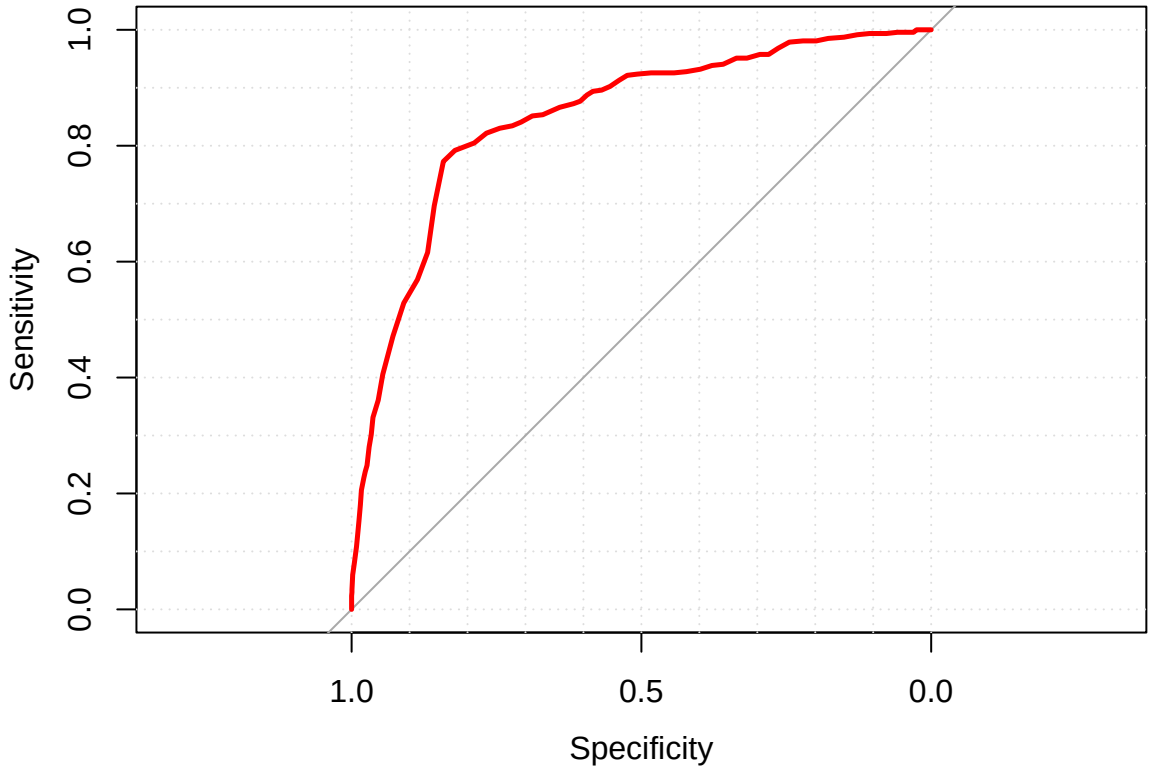
Next steps

- Logging some of the variables
- Removing some of the variables
- Using an ROC

```
library(pROC)

fetal_roc <- roc(response = data$recoded, predictor = fitted.values(abnormal_logistics))

coords(fetal_roc, "best", inut="threshold")
```

```
##   threshold specificity sensitivity
## 1 0.3345881   0.8416918   0.7728238
```

```
plot(fetal_roc, col="red", grid=TRUE, lwd=2.5)
```




```
auc(fetal_roc)
```

```
## Area under the curve: 0.8477
```

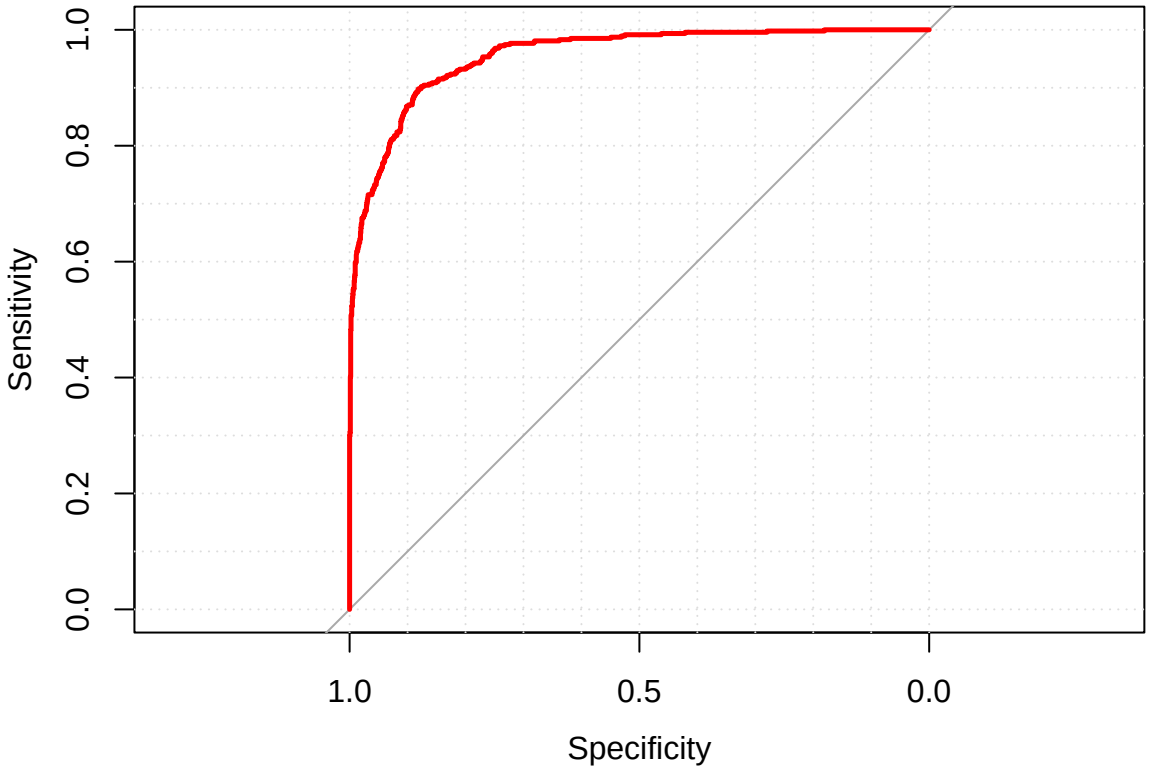
From the further ROC analysis, we can see that at best the second, simpler model can achieve a specificity of 0.84 and sensitivity of 0.77, which is not bad, but a bit low for medical test results.

As such, let us now perform the same analysis on our prior, more complicated model.

```
fetal_roc_PC <- roc(response = data$recoded, predictor = fitted.values(chosen_model))

coords(fetal_roc_PC, "best", input="threshold")
```

```
##   threshold specificity sensitivity
## 1 0.2342552    0.881571   0.8980892
```

```
plot(fetal_roc_PC, col="red", grid=TRUE, lwd=2.5)
```




```
auc(fetal_roc_PC)
```

```
## Area under the curve: 0.9555
```

The second model does a significantly better job discriminating, with both the sensitivty and specificity at similar values, both of which pushing on the edge of 90%. While this is a notable improvement from the simpler model, it would still give false test results a little over 10% of the time.

In conclusion, the simpler model has a significantly lower sensitivity than the complex model, and a slightly lower specificity, which suggests that the more complex second model is more suitable. However, as the second model still under performs compared to the regulator preferences of 95%-99%+ accuracy, so I would not recommend widespread use of either model.

While I would prefer to increase the sensitivity, even at risk of decreasing the specificity, I cannot find an efficient way to do this short of classing every infant as at risk, which would defeat the point of this model entirely.

The only way I can find that would provide a decent attempt at finding this is to trawl through the possible indices for the sensitivities to find one with an accuracy in the desired range.

```
chosen_threshold <- 1100

fetal_roc_PC$thresholds[1291]
```

```
## [1] 0.08648535
```

```
fetal_roc_PC$sensitivities[chosen_threshold]
```

```
## [1] 0.9808917
```

```
fetal_roc_PC$specificities[chosen_threshold]
```

```
## [1] 0.6610272
```

By essentially guessing and checking I was able to find a threshold that gives a sensitivity of 98% in return for a 66% specificity. this means only 2% of at risk infants would be missed, and that roughly 34% of of healthy infants would be class as being at risk.

I believe that this is a suitable trade off, as the risks of an at risk infant not being detected are significantly greater than the risks of further tests or possible treatment.

If we seek to reach the lower bound of this, that being a 95% sensitivty, we find that the specificity increases to 77%. While this is good, as we now only mis-classify 23% of health infants as at-risk, an 11% reduction, we have now more than doubled the number of infants we are declaring healthy despite being at risk. Although 5% in total is fairly low, the decision of which threshold to use should be left to medical practitioners.

```
chosen_threshold <- 1291

fetal_roc_PC$sensitivities[chosen_threshold]
```

```
## [1] 0.9511677
```

```
fetal_roc_PC$specificities[chosen_threshold]
```

```
## [1] 0.7703927
```

Ultimately, it is possible for us to develop a model that will provide a suitable level of accuracy in the identification of at risk infants, with trade offs for the number of false positives we are willing to deal with in exchange for higher successful identification of true positives. The decision regarding the threshold should be left to practitioners, who can decide if the costs, risks, and additional requires of further testing or possible treatment on a healthy infant are worth the higher successful identification of at risk children.

This will also largely depend on the proportion of children we expect to be at risk.

```
round(sum(data$recoded) / nrow(data), 4)
```

```
## [1] 0.2215
```

```
0.2215 * nrow(data)
```

```
## [1] 470.909
```

```
0.23 * (nrow(data) - sum(data$recoded))
```

```
## [1] 380.65
```

As seen, in our cases only 22.15% of the infants in our data were actually at risk, meaning that if we used to 95% model we would have further classed a further 381 infants as at risk when they were actually healthy. However, the majority of cancer tests return false positives, with very few positive mamograms actually leading to breast cancer diagnoses, in exchange for patients recieving peace of mind, and the practitioners knowing they are more likely to catch cancer cases than not.

It would be my naive recommendation to use the 95% sensitive model, as I believe that while successfully classing as many at risk infants as possible correctly, the extra load on the system from testing healthy infants may result in significant delays in an already backlogged health system. As stated many times, the actual decision should ultimately lie with those who best understand our health system, namely the doctors and nurses who conduct much of the testing and treatment.

Finally, I would like to touch on the fact that our model is very complicated. Even though we do have a high level of sensitivity, and a relatively good level of specificity, the fact is that our model is not something that can be applied easily, and will definitely need a computer to be effectively used. As such, I would recommend the automation of test result interpretation by loading results directly into a program with the model pre-loaded.

Another alternative would be to try to develop a model for this data using multivariate techniques such as LDA or QDA. Firstly,there is sufficient data to make these claims, but we need to further investigate the MVN and equal covariance assumptions, which we can do using Kolmogorov-Smirnov and Multivariate Levene's tests respectively.

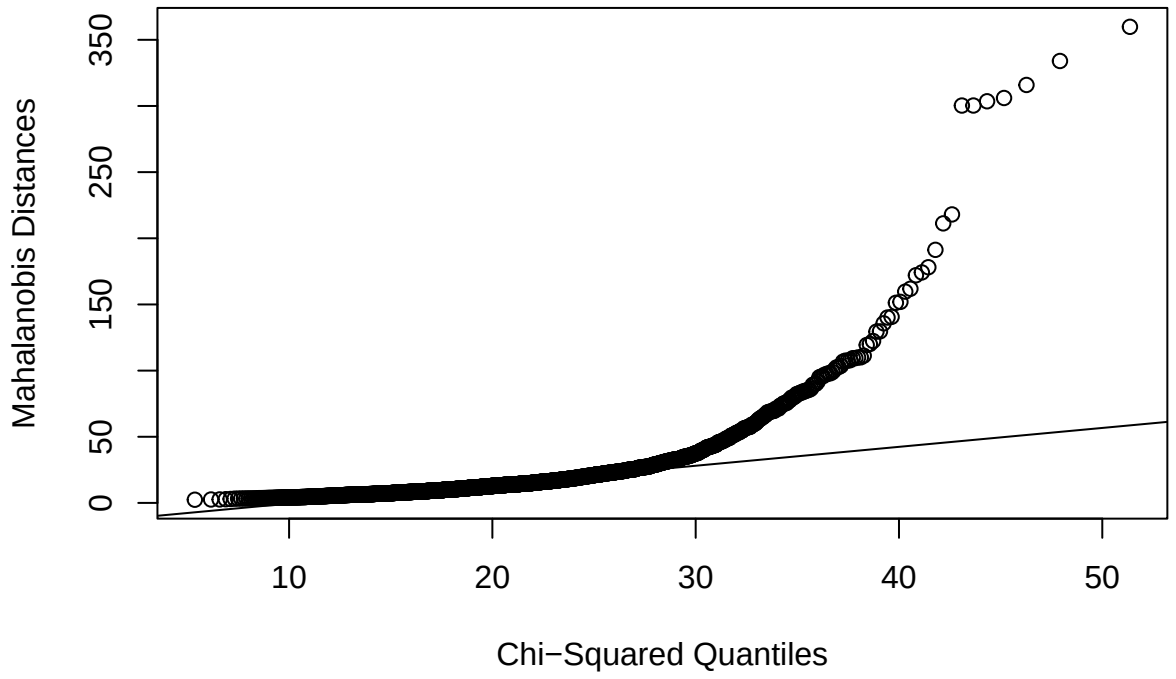
One advantage to this is that we do not have to collapse the groups into two, so we can work with three groups as per the original data.

```
library(mvabund)
data <- read_csv("fetal_health.csv")
data_matrix <- as.matrix(data[,-22])

dimesion_manova <- manova(data_matrix ~ data$fetal_health)
abs_resids <- abs(dimesion_manova$resid)

maonva_permutations <- manylm(abs_resids ~ data$fetal_health, cor.type = "R", test = "F")
anova(maonva_permutations, resamp = 'perm.resid')
```

```
## Analysis of Variance Table
##
## Model: manylm(formula = abs_resids ~ data$fetal_health, test = "F",
## Model:     cor.type = "R")
##
## Overall test for all response variables
## Test statistics:
##                   Res.Df Df.diff val(F) Pr(>F)
## (Intercept)         2125
## data$fetal_health   2124       1   4837  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Arguments:
##  Test statistics calculated assuming unconstrained correlation response
##  P-value calculated using 999 iterations via residual (without replacement) resampling.
```

Based on the highly significant P-value output, we can conclude that the null hypothesis of equal covariance must be rejected. Even if this fails, QDA is still possible provided MVN holds, although based on the plots, it does seem unlikely.

```r
resid_mat <- residuals(dimesion_manova)
Sigma <- cov(resid_mat)

maha <- mahalanobis(resid_mat, center = colMeans(resid_mat), cov = Sigma)

qqplot(qchisq(ppoints(nrow(data)), df = ncol(resid_mat)), y = maha,
       xlab = "Chi-Squared Quantiles", ylab = "Mahalanobis Distances")
qqline(maha, distribution = function(p) qchisq(p, df = ncol(resid_mat)))
```



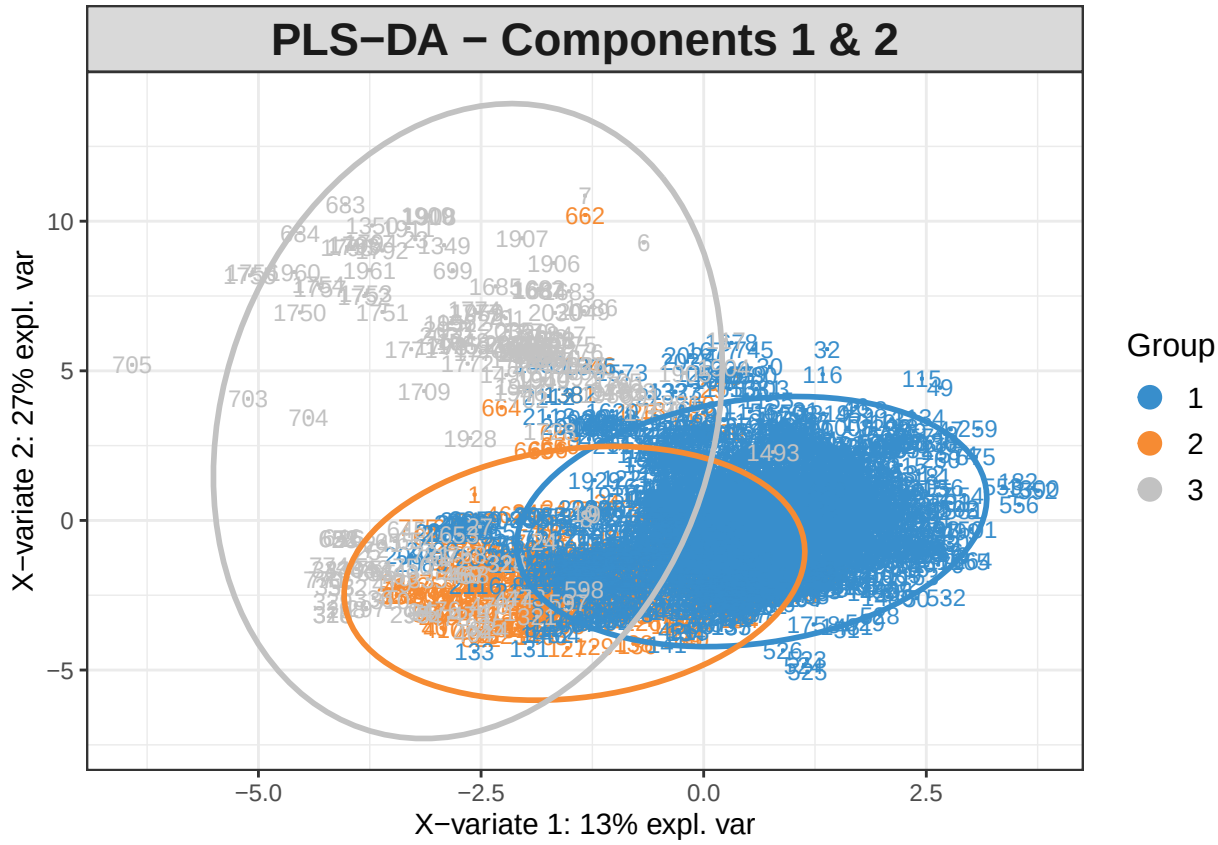

```r
ks.test(maha, "pchisq", df = ncol(resid_mat))
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  maha
## D = 0.42082, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

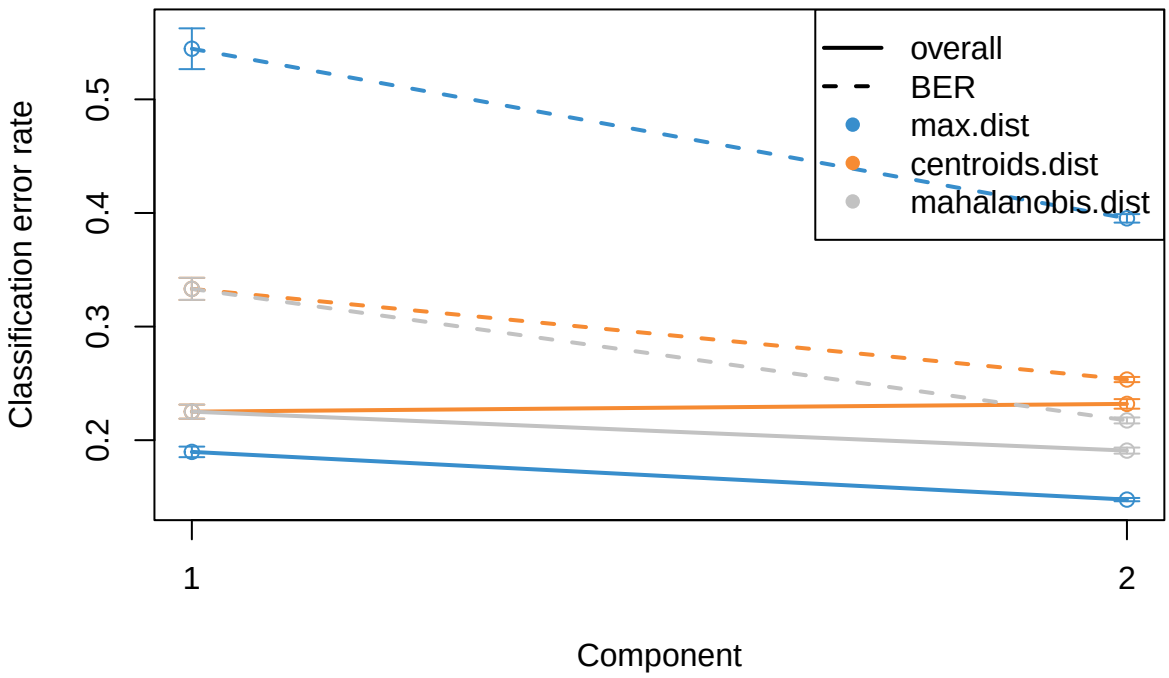
So the data is definitely not MVN. This means that neither LDA nor QDA are good choices, but we can use PLS-DA.

```r
library(mixOmics)

X <- data_matrix
```

```r
Y <- data$fetal_health

plsda_model <- plsda(X, Y, ncomp=2)

plotIndiv(plsda_model,
          comp = c(1, 2),
          group = Y,
          ellipse = TRUE,
          legend = TRUE,
          legend.title = "Group",
          title = "PLS-DA - Components 1 & 2")
```

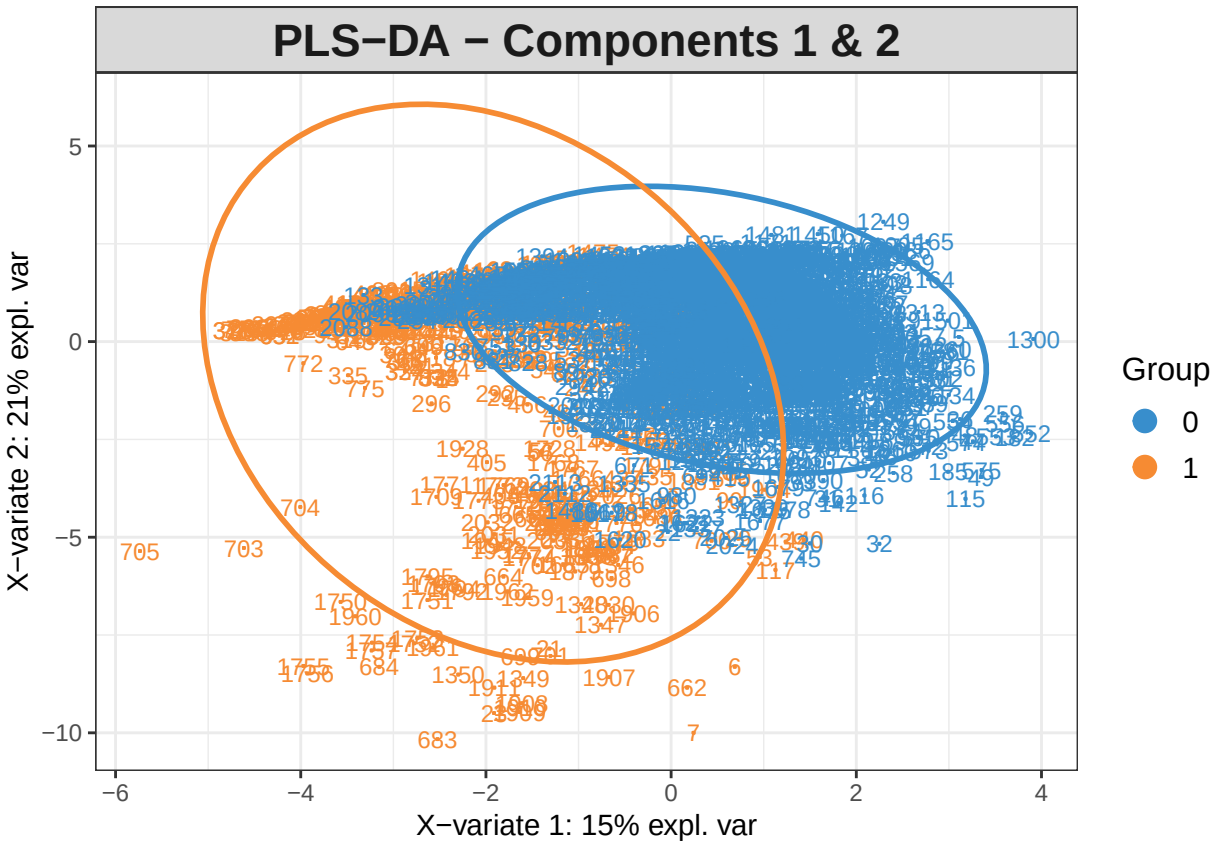



```r
set.seed(123)
perf <- perf(plsda_model, validation = "Mfold", folds = 5,  nrepeat = 10)

plot(perf, criterion = "error.rate")
```



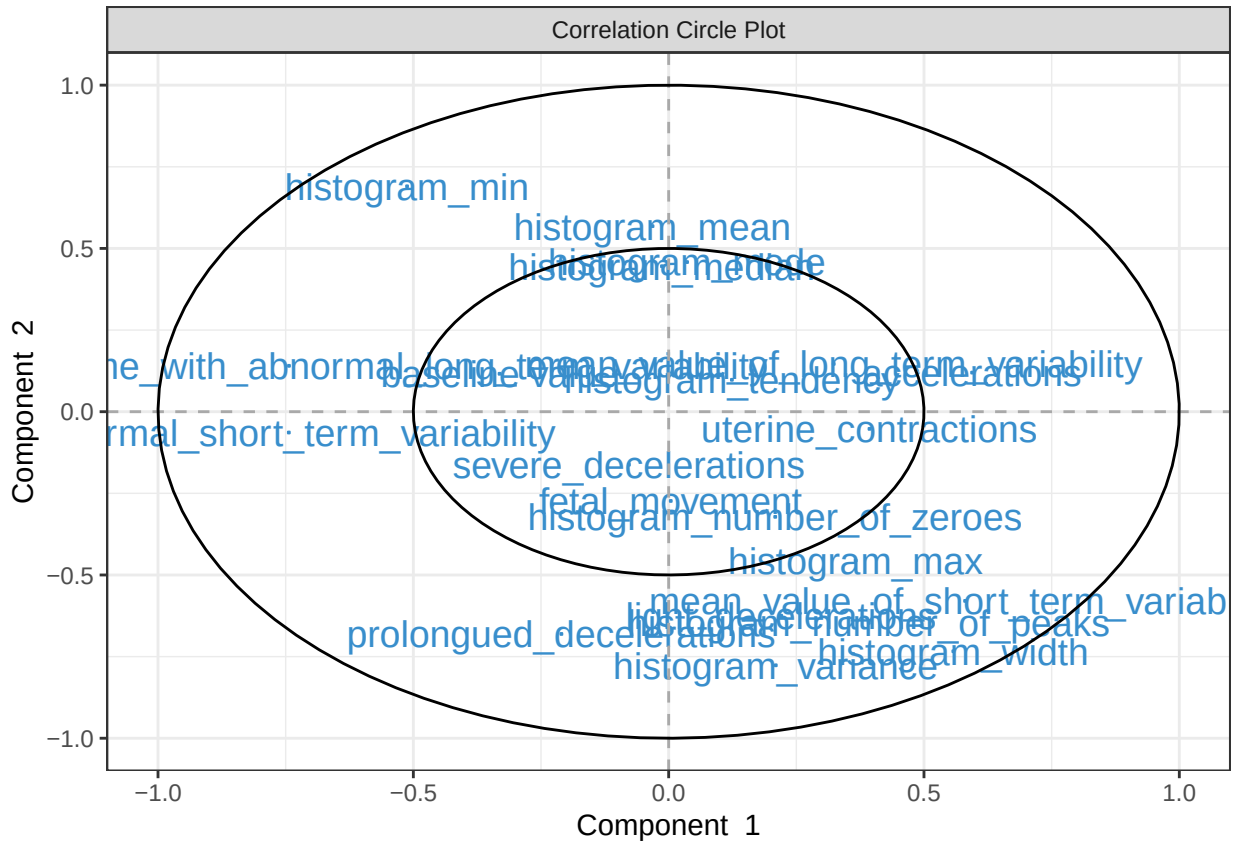
The results from this are not especially useful, as we can see that the second group does not inhabit an especially unique area, and significantly overlaps with the other groups. Let us now return to the collapsed group problem and repeat this step.

```
data <- data %>%
  mutate(
    recoded = ifelse(fetal_health == 1, 0, 1)
  ) %>%
  dplyr::select(-fetal_health)

data_matrix <- as.matrix(data[,-22])

X <- data_matrix
Y <- data$recoded

plsda_model <- plsda(X, Y, ncomp=2)

plotIndiv(plsda_model,
          comp = c(1, 2),
          group = Y,
          ellipse = TRUE,
          legend = TRUE,
          legend.title = "Group",
          title = "PLS-DA - Components 1 & 2")
```

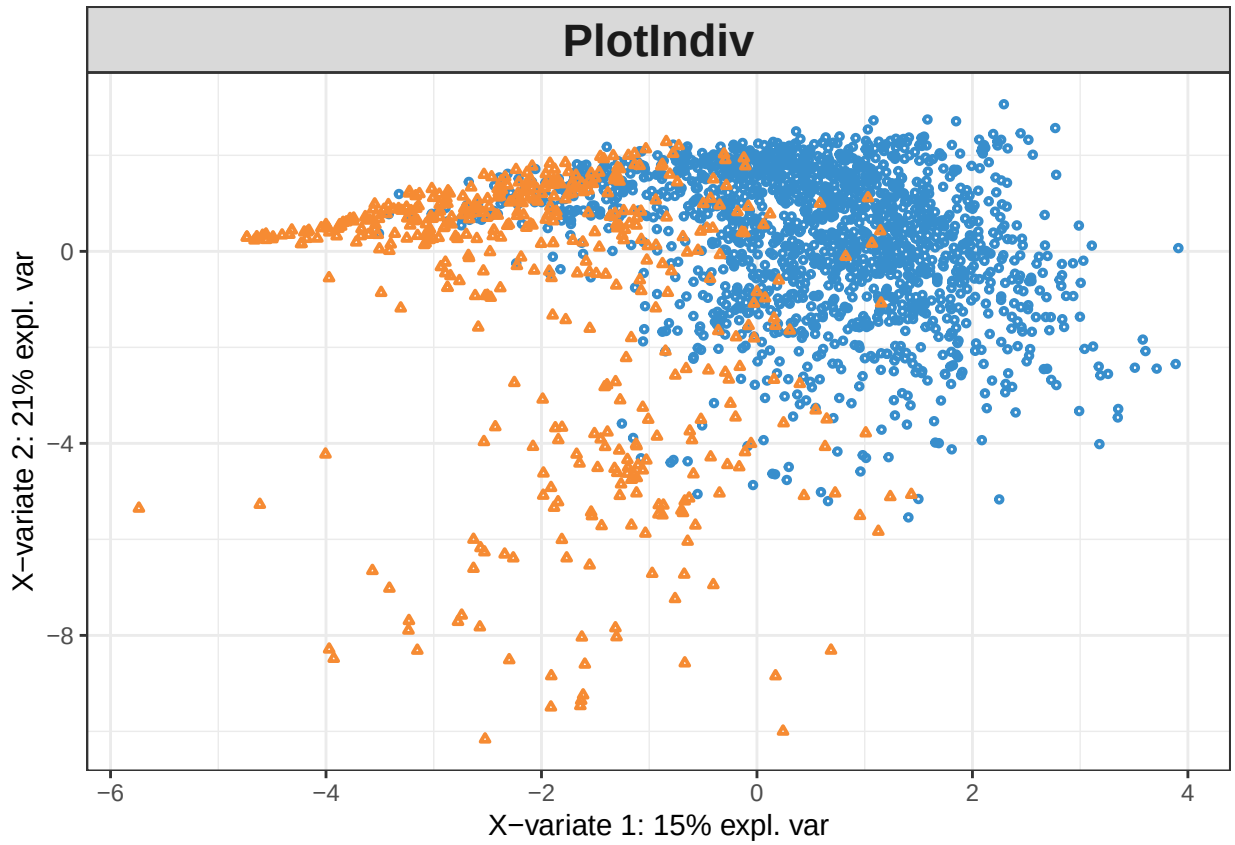


We can see that this solution is actually much worse than the logistic regression solution, as there is significant overlap between these groups. Not only this, but it is clear from both PLS-DA analyses that the proportions of variance explained are far too low in both solutions.

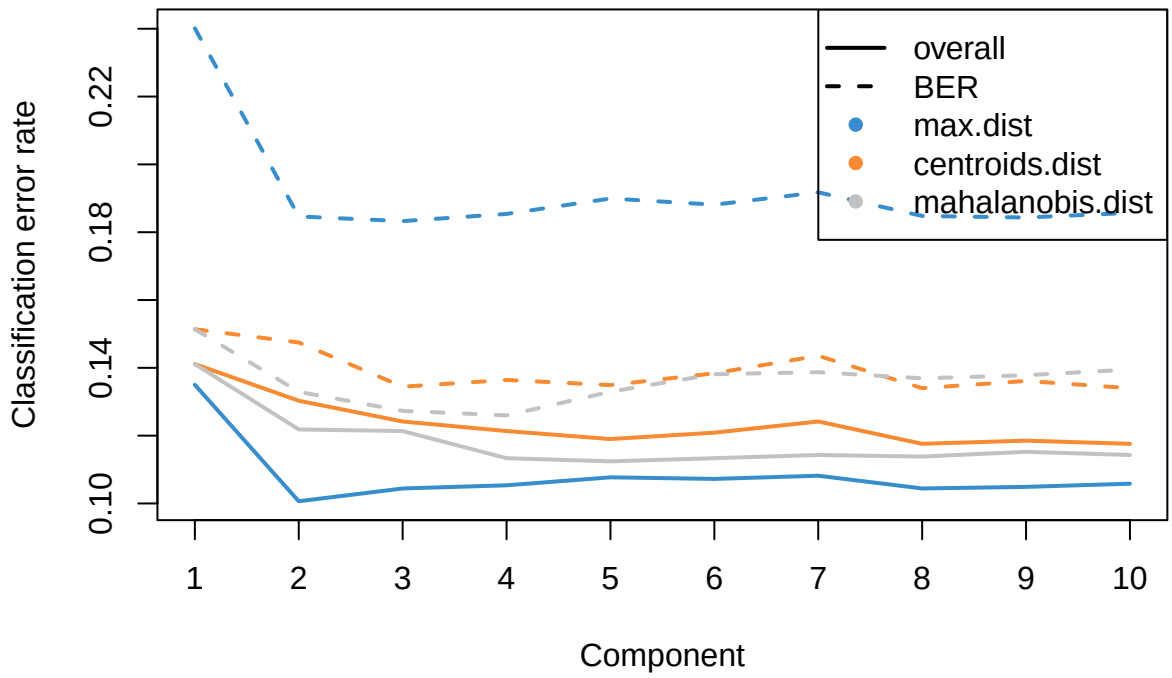
```
plsdaout <- plsda(X, Y, ncomp=10)
plotVar(plsdaout)
```



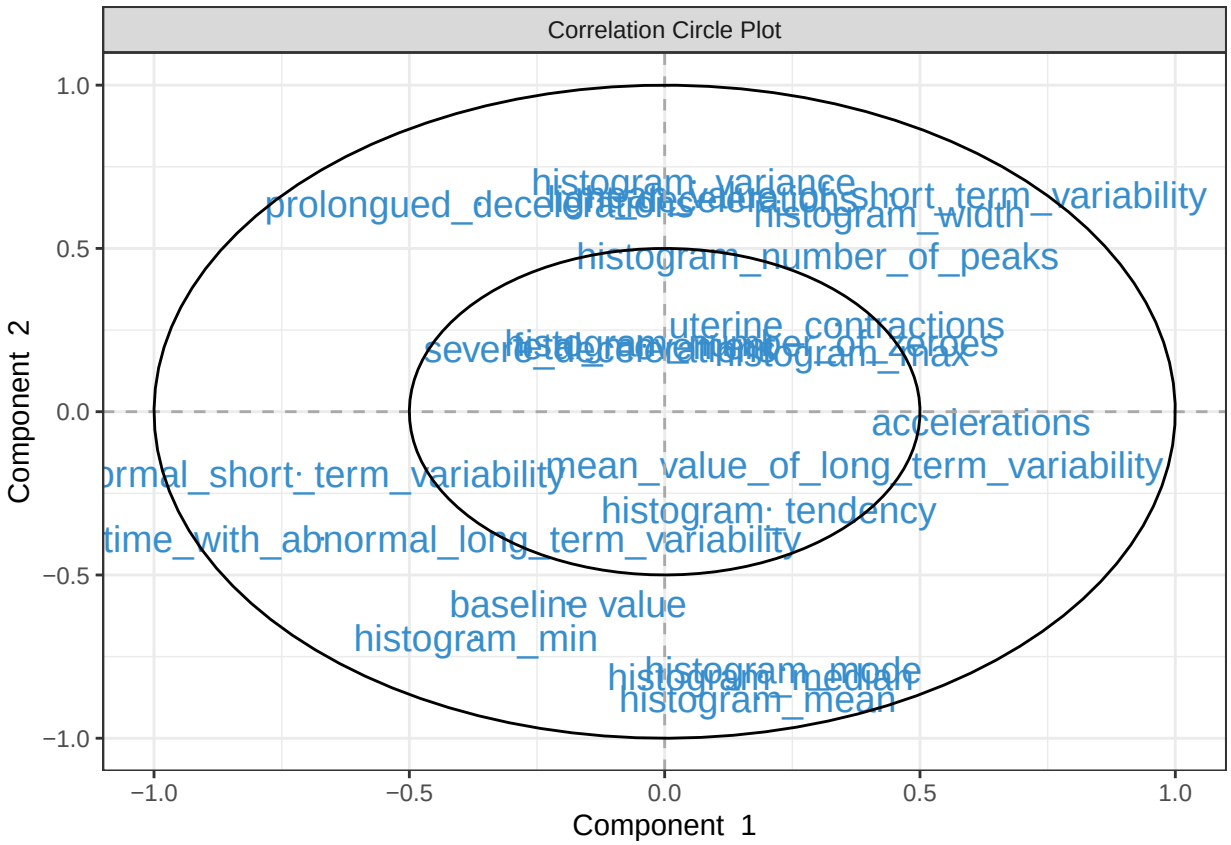

```
plotIndiv(plsdaout, comp=1:2, ind.names=FALSE, cex=0.5)
```



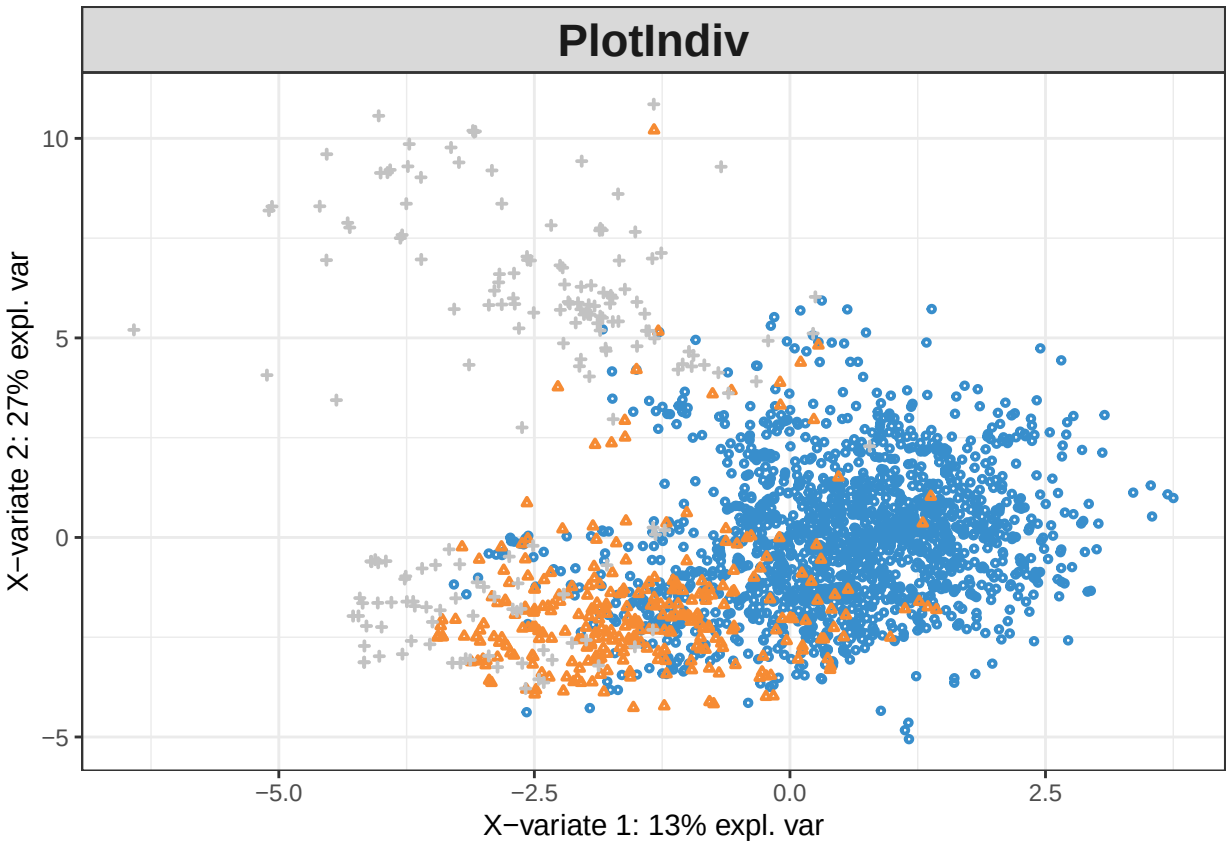

```
health.perf <- perf(plsdaout)
plot(health.perf)
```




```
data <- read_csv("fetal_health.csv")
data_matrix <- as.matrix(data[,-22])

X <- data_matrix
Y <- data$fetal_health

plsdaout <- plsda(X, Y, ncomp=10)
plotVar(plsdaout)
```




```
plotIndiv(plsdaout, comp=1:2, ind.names=FALSE, cex=0.5)
```


PlotIndiv
10
5
0
−5
X−variate 2: 27% expl. var
−5.0
−2.5
0.0
2.5
X−variate 1: 13% expl. var