

# Fetal Monitoring Can Predict Health Outcomes

HarvardX PH125.9x Capstone

David Vance

2024-06-04

## Background and Introduction

Monitoring the fetus during pregnancy and labor is of great importance, because it directly impacts health outcomes for both the mother and the baby (Urdal, 2021). Among other things, it is useful for assessing the proper development of the fetus, the early detection of any distress, and for helping to guide decisions about labor and delivery. There are several key measurements that are routinely taken throughout the process.

**Baseline Fetal Heart Rate:** This is normally within a range of 110-160 beats per minute. A proper range indicates good fetal oxygen levels and self regulation ability (Pildner von Steinburg, 2013). Deviations from the baseline can signal hypoxia or perhaps caridac abnormalities.

**Accelerations:** This refers to increases in fetal heart rates, which are associated with fetal movement and uterine contractions, among other things. These are positive signs, and may correlate with a lower risk of fetal acidosis (López-Justo, 2021).

**Decelerations:** This refers to decreases in fetal heart rates, which can be benign, but are concerning with increasing length and frequency. They may be caused by umbilical cord compression, but are also quite normal during contractions (López-Justo, 2021).

**Variability:** This refers to the variability in the fetal heart rate, with reduced variability potentially being indicative of a problem (Tarvonen, 2022).

**Histogram Characteristics:** Fetal heart rate can also be characterized in even greater detail, by looking at statistics such as the mean, median, max and min, amongst other things, all of which may be useful measures of fetal well-being (Griffin, 2005).

Large amounts of these types of data, paired with known fetal health outcomes, are a fruitful target for machine learning techniques to help predict the health of future fetuses, and in turn, lead to early interventions to improve the health outcome of the fetus and the mother.

In this report, we will be looking at such a dataset, which contains 2,113 observations of 21 variables each, and seeing if we can build any models that can successfully predict the paired outcome variable, fetal health.

## Data Exploration and Analysis

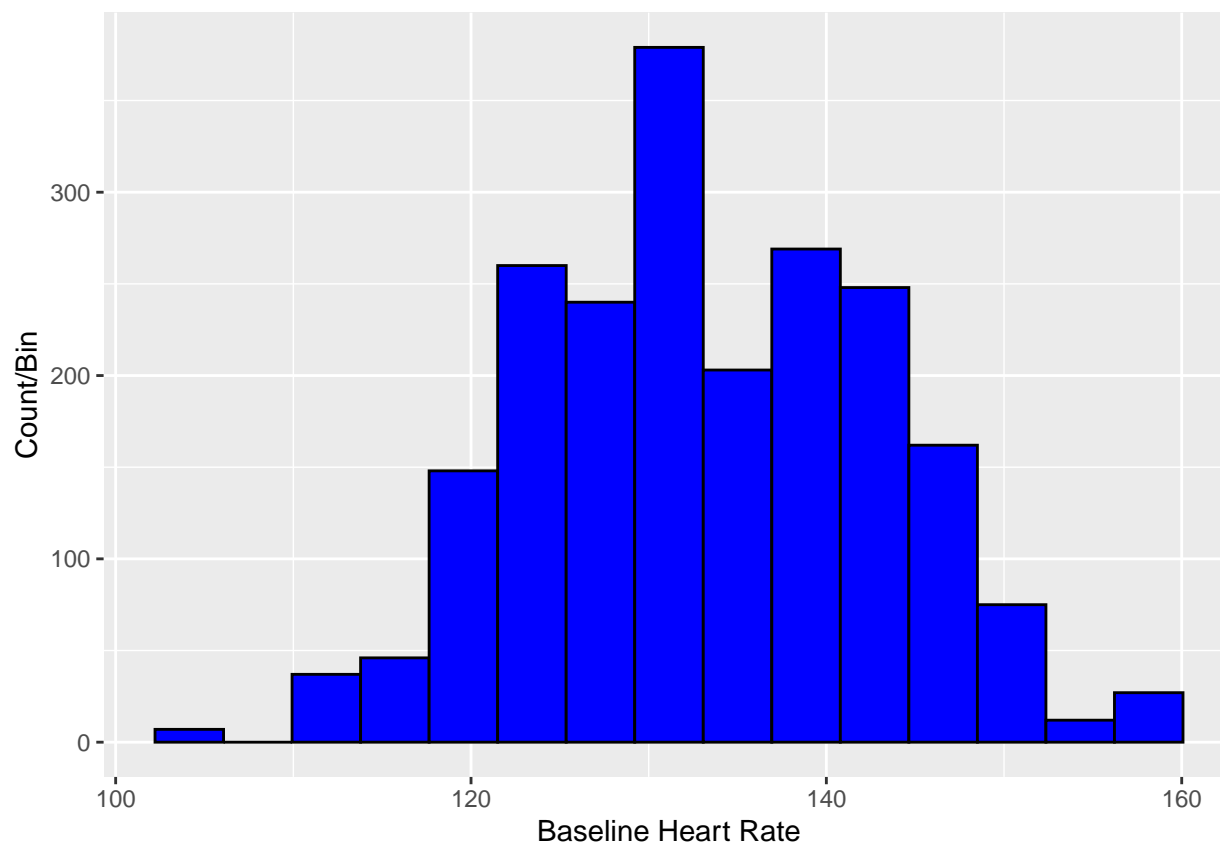
Before we try out any models, lets get a handle on what kind of data we have in the dataset. Let's take a look at the top few entries of the dataset.

##	baseline.value	accelerations	fetal_movement	uterine_contractions
## 1	144	0.002	0.010	0.002
## 2	125	0.005	0.002	0.006

## 3	120	0.013	0.006	0.003
## 4	142	0.000	0.006	0.000
## 5	145	0.000	0.000	0.002
## 6	146	0.000	0.000	0.005
##	light_decelerations	severe_decelerations	prolongued_decelerations	
## 1	0.000	0	0	
## 2	0.009	0	0	
## 3	0.000	0	0	
## 4	0.000	0	0	
## 5	0.000	0	0	
## 6	0.000	0	0	
##	abnormal_short_term_variability	mean_value_of_short_term_variability		
## 1		76		0.6
## 2		60		1.3
## 3		40		1.3
## 4		61		0.7
## 5		49		0.6
## 6		69		0.3
##	percentage_of_time_with_abnormal_long_term_variability			
## 1		0		
## 2		0		
## 3		0		
## 4		52		
## 5		3		
## 6		55		
##	mean_value_of_long_term_variability	histogram_width	histogram_min	
## 1		11.8	54	126
## 2		7.5	96	68
## 3		7.1	89	68
## 4		12.2	79	71
## 5		11.6	27	140
## 6		4.6	11	145
##	histogram_max	histogram_number_of_peaks	histogram_number_of_zeroes	
## 1	180	2	0	
## 2	164	6	1	
## 3	157	5	1	
## 4	150	12	0	
## 5	167	0	0	
## 6	156	0	0	
##	histogram_mode	histogram_mean	histogram_median	histogram_variance
## 1	156	153	155	5
## 2	110	120	120	34
## 3	133	134	136	9
## 4	143	142	144	3
## 5	159	156	158	3
## 6	152	150	152	0
##	histogram_tendency	fetal_health		
## 1	0	2		
## 2	0	1		
## 3	1	1		
## 4	1	2		
## 5	1	2		
## 6	0	2		

As can be seen, the data itself is quite unwieldy. Let's look at the individual observations in more detail.

The first data point is the baseline value of the heart rate. As mentioned earlier, the expected range for this value should be between 110-160. Is this borne out by the data set?



As can be seen, the baseline heart rates do fall in this range, with a minimum of **106**, a maximum of **160**, and an average value of **133.3**.

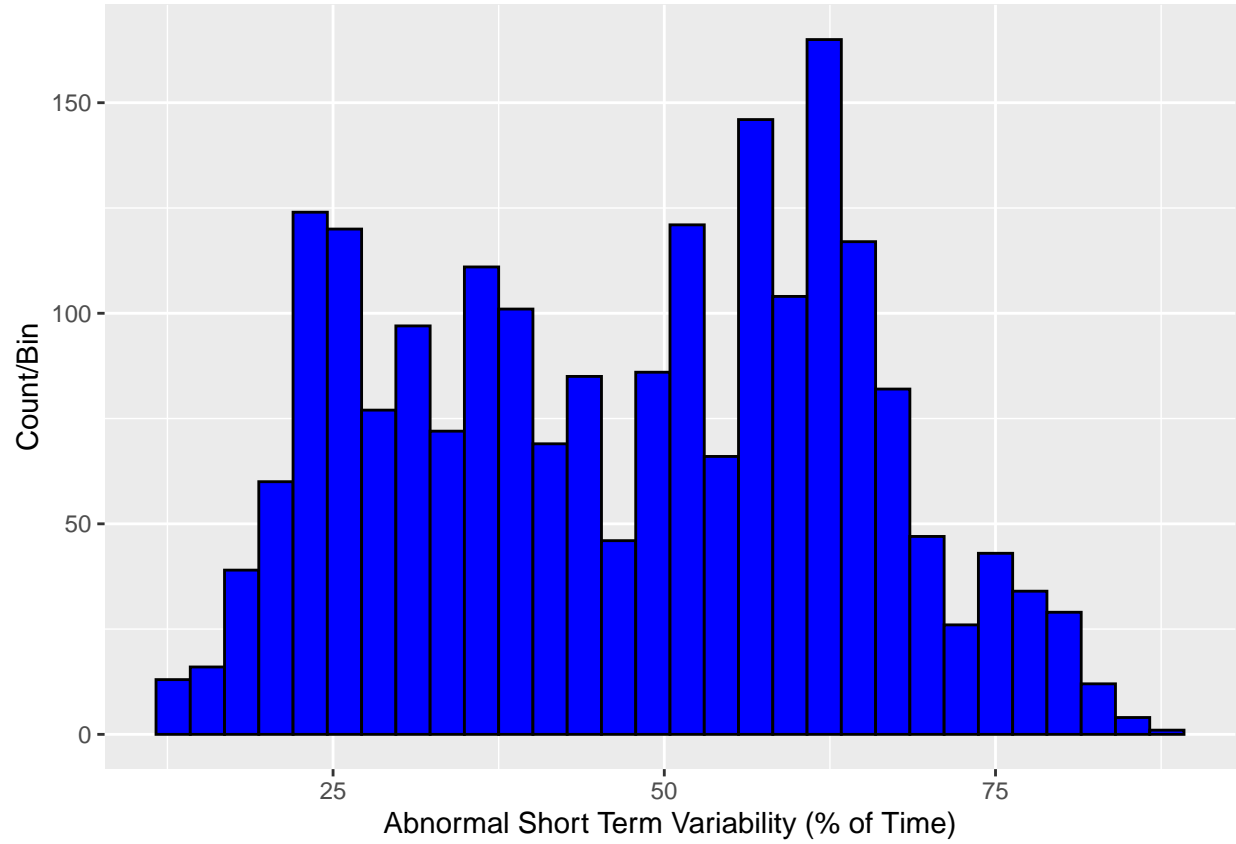
Unlike baseline heart rate, several of the other important variables in this data set have very small values, and do not have a large range of values. Let's take a look at the minimum, maximum, mean, and standard deviation of these variables.

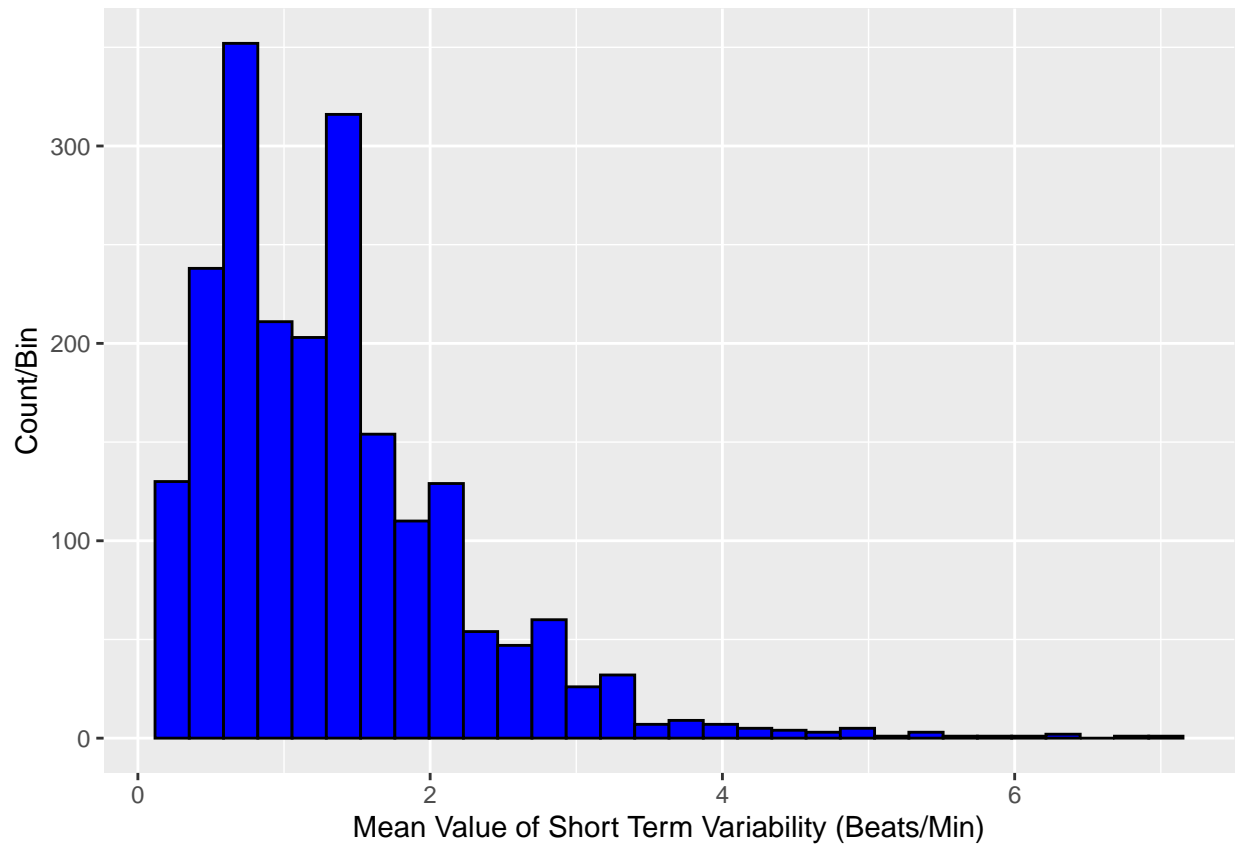
Variable	Min	Max	Mean	StDev
Accelerations	0	0.019	0.0031884	0.0038713
Movement	0	0.481	0.0095173	0.0468038
Uterine Contractions	0	0.015	0.0043871	0.0029406
Light Decelerations	0	0.015	0.0019011	0.0029656
Severe Decelerations	0	0.001	0.0000033	0.0000575
Prolonged Decelerations	0	0.005	0.0001595	0.0005916

Given these small and not too different values amongst the individual variables, it would be difficult for any of them to be predictive on their own. As we progress, we will see whether we can come up with a machine learning model that can make anything of them.

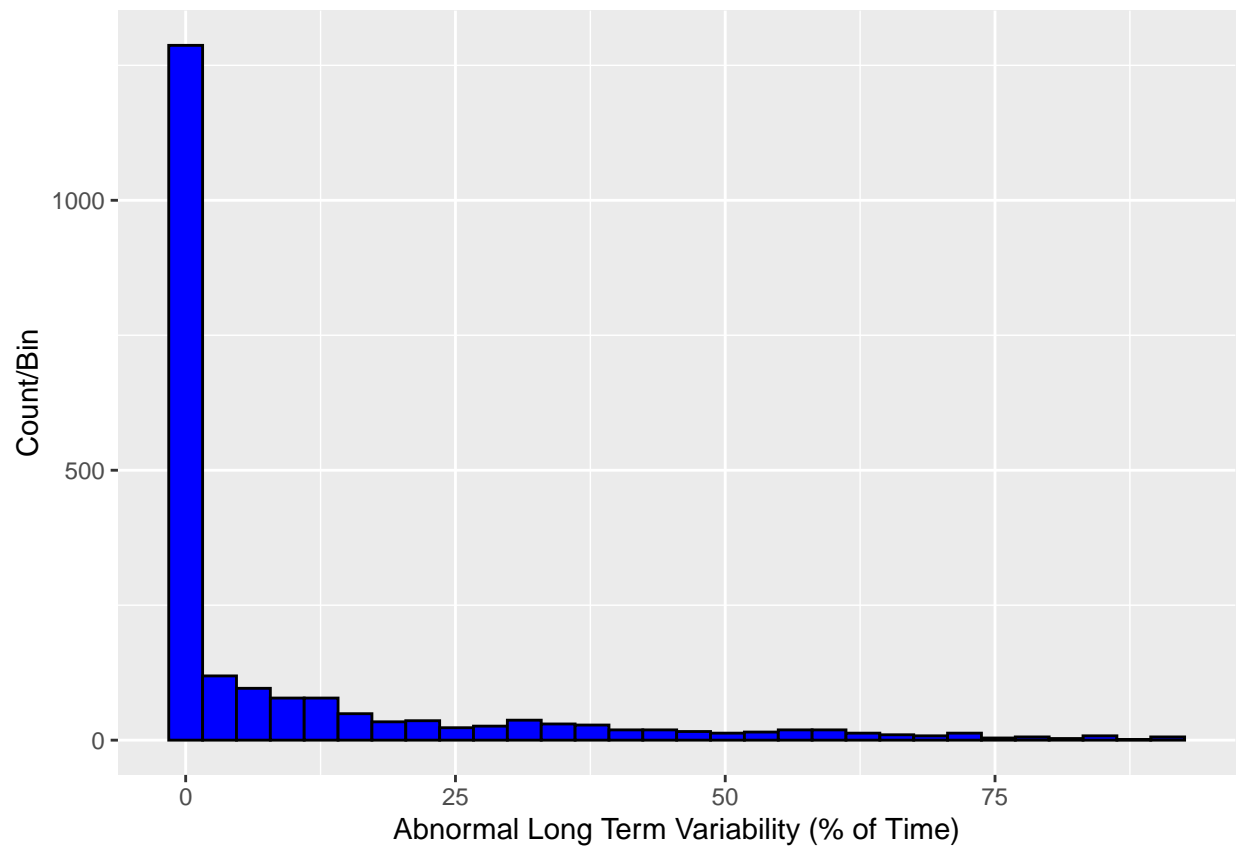
We next come to a group of variables that relate to the variability of the heart rate, and here we do see some big differences in the data. Let's look at the distribution of a few of them. Note: somewhat unintuitively, the term "abnormal" in these data does not necessarily mean that there's anything wrong with the heart

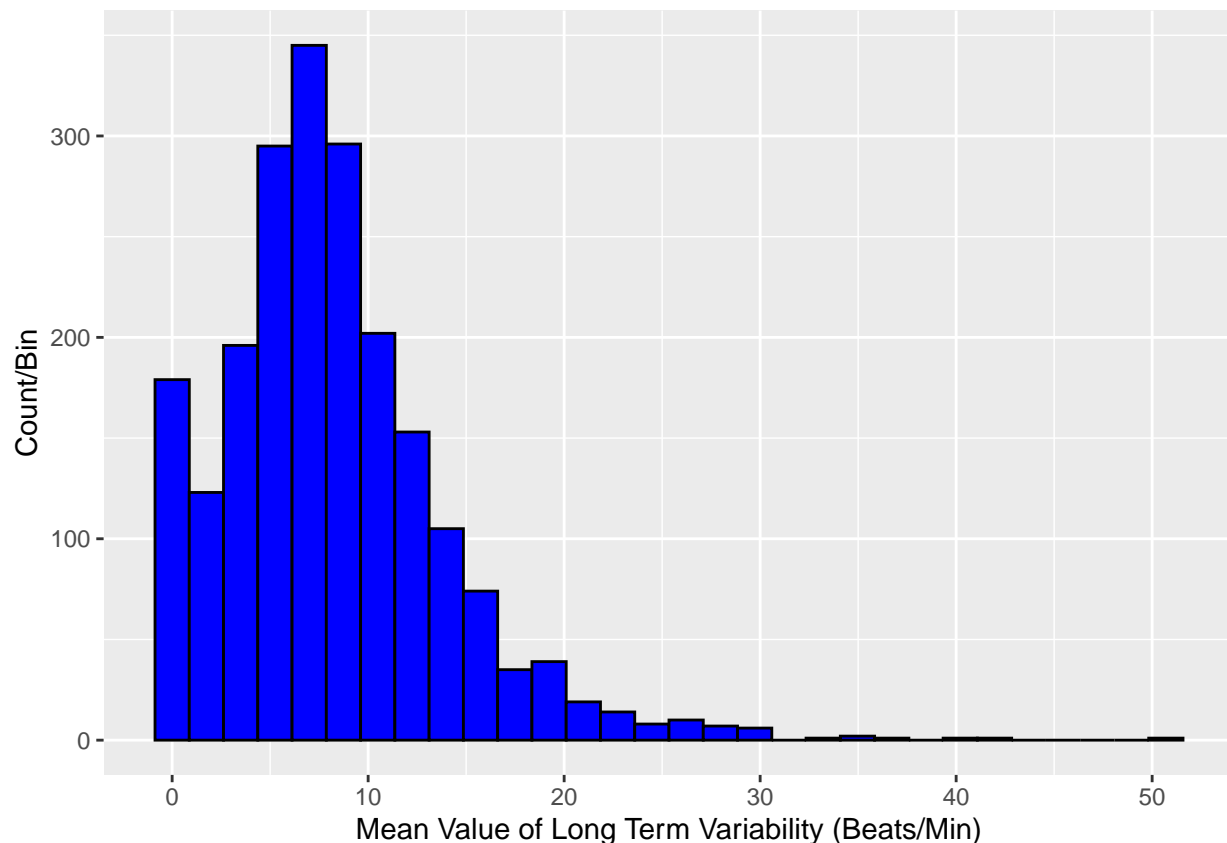
rate, but rather that it differs from the baseline, or normal, heart rate. Indeed, these variations are quite normal, and in fact are a positive sign of fetal viability.





As can be seen, it is not unusual for there to be abnormal short term variability in the fetal heart rate, with up to 80% of the time spent in these short term abnormal periods. Most of this abnormal time is spent within 3 beats/minute of the baseline rate. The data also looks somewhat bimodal, which may prove to have some predictive value.



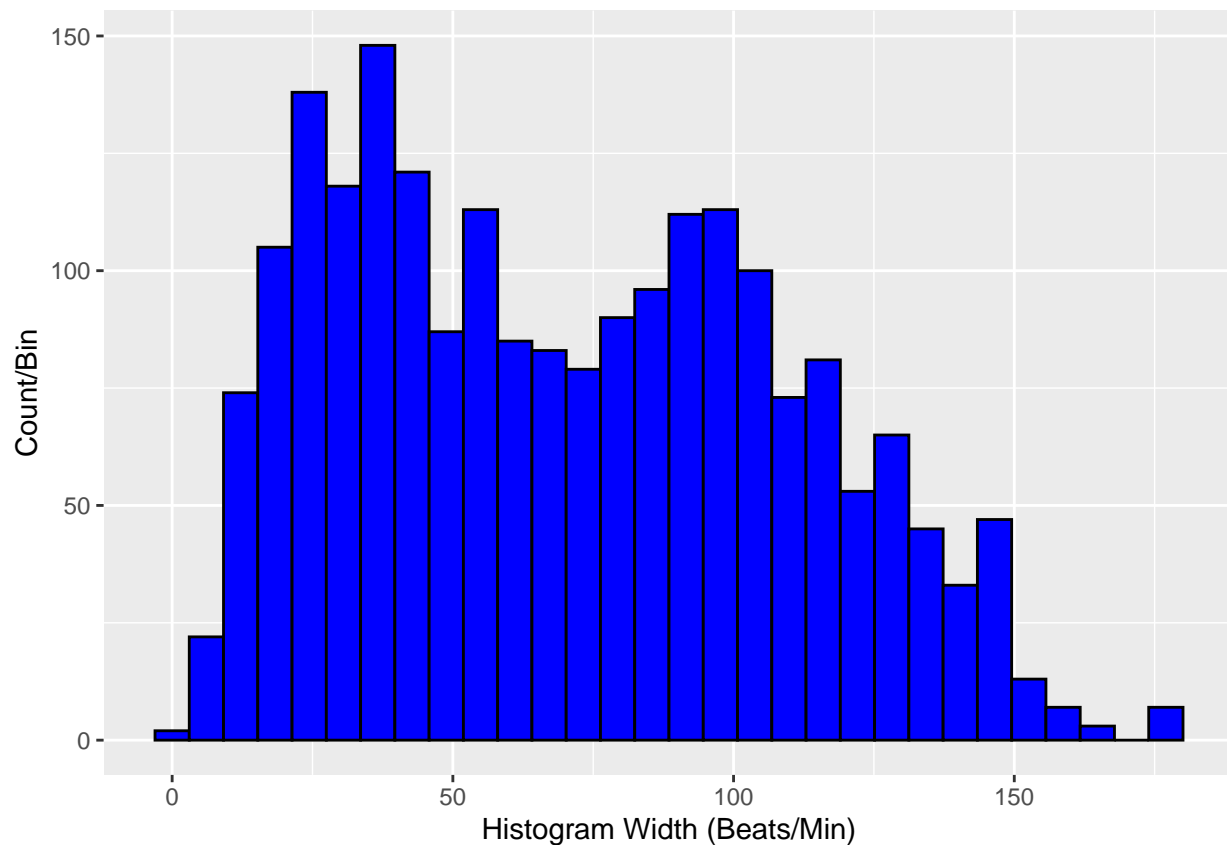


On the other hand, long term variability is much more rare, with a large number of fetuses experiencing it for very short periods, if at all. However, those that do experience it tend to have a much higher mean value of variability, around 8 beats/minute. As noted, variability is normal and positive, though perhaps not for long terms. We shall see if the model finds anything along those lines.

Lastly, there are 10 different descriptors of the histogram signal of fetal heart rate. Some of these are quite variable, while some are not. Let's look at a summary of these statistics.

Variable	Min	Max	Mean	StDev
Width	3	180	70.5352579	39.0077063
Min	50	159	93.5646001	29.5622692
Max	122	238	164.0998580	17.9451755
Number of Peaks	0	18	4.0771415	2.9516644
Number of Zeros	0	10	0.3256034	0.7077710
Mode	60	187	137.4543303	16.4020255
Mean	73	182	134.5996214	15.6104217
Median	77	186	138.0894463	14.4789570
Variance	0	269	18.9072409	29.0387657
Tendency	-1	1	0.3185045	0.6110746

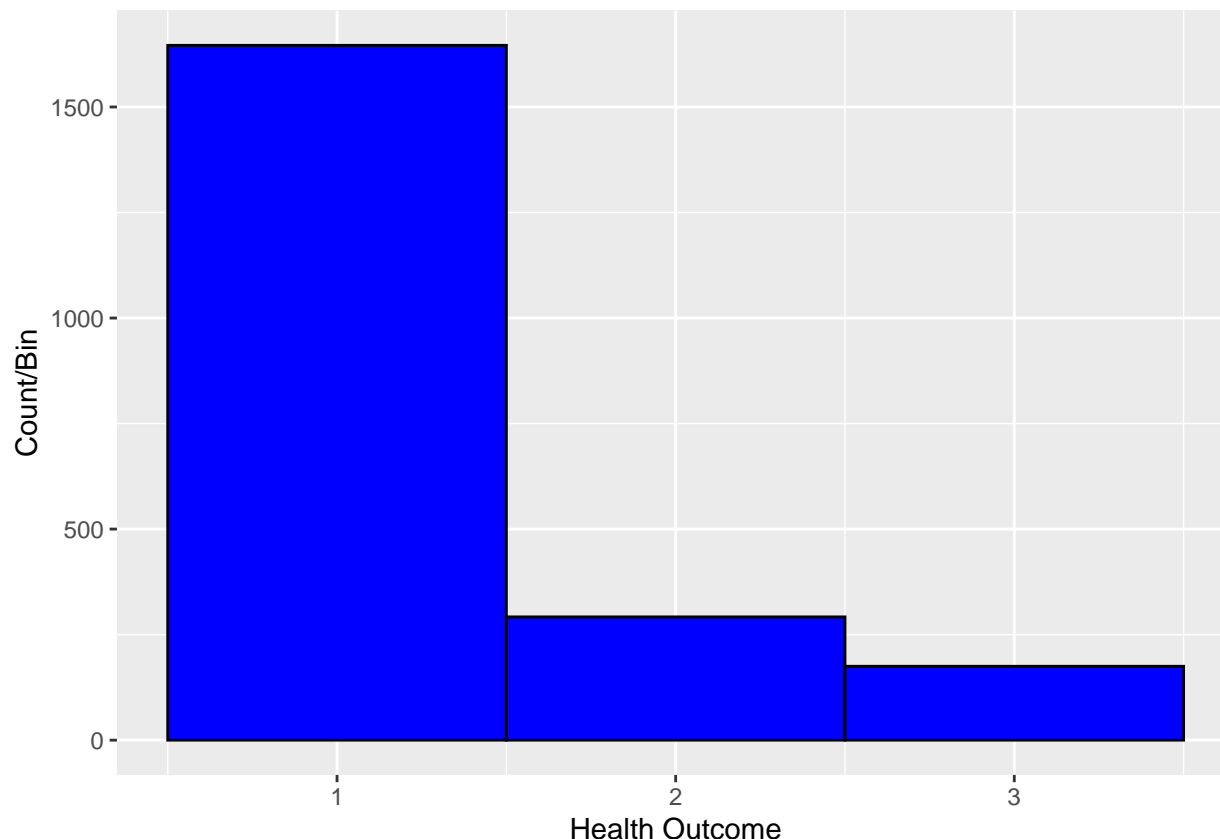
It is not necessary to look beyond the first row of this table to see some real variability in these numbers. Histogram width shows a minimum of just 3 and a maximum of 180, a huge range with a large standard deviation. Let's just take a look at that data quickly to see what that looks like.



Not unexpectedly, given the statistics listed above, the data shows that it is not uncommon to have a histogram width anywhere between 10 and 150, although this data also shows a bit of bimodality.

Last, but not least, the final column represents the known health outcomes for these fetuses. Let's have a look at how they are distributed.





In this data set, a fetal health outcome of 1 means that the fetus was “normal” when born. A health outcome of 2 represents “suspect,” in that there is a suspected, but not yet confirmed, pathology. A health outcome of 3 represents a fetus with a confirmed pathology. As can be seen, thankfully the most likely outcome for fetuses is normal (1), but there is still an uncomfortably high percentage of suspect (2) and pathological (3) outcomes (13.8% and 8.3%, respectively). Being able to identify these suspected or confirmed pathological fetuses early may lead to life saving early intervention. Thus, we will now attempt to use this data set to predict fetal outcome based on all, or some portion of, the provided variables.

## Results

The first thing we must do is to split up our data set into testing and training sets. However, since we will want to test several different models, we will further split up our training set into a `train_train` and `train_test` sets. These two sets will be used to train and test the various models, with the original `test_data` set being reserved to test only the accuracy of the final model, once decided on.

```
train_index <- createDataPartition(fetal_data$fetal_health, times=1, p=0.1, list=FALSE)
train_data <- fetal_data[-train_index, ]
test_data <- fetal_data[train_index, ]

train2_index <- createDataPartition(train_data$fetal_health, times=1, p=0.1, list=FALSE)
train_train_data <- train_data[-train2_index, ]
train_test_data <- train_data[train2_index, ]
```

The full dataset consists of 2113 observations. We have now reserved 10% of the data (`test_data`, 213 observations) to test with the final model. Of the remaining 90% (1900 observations), a further 10% was

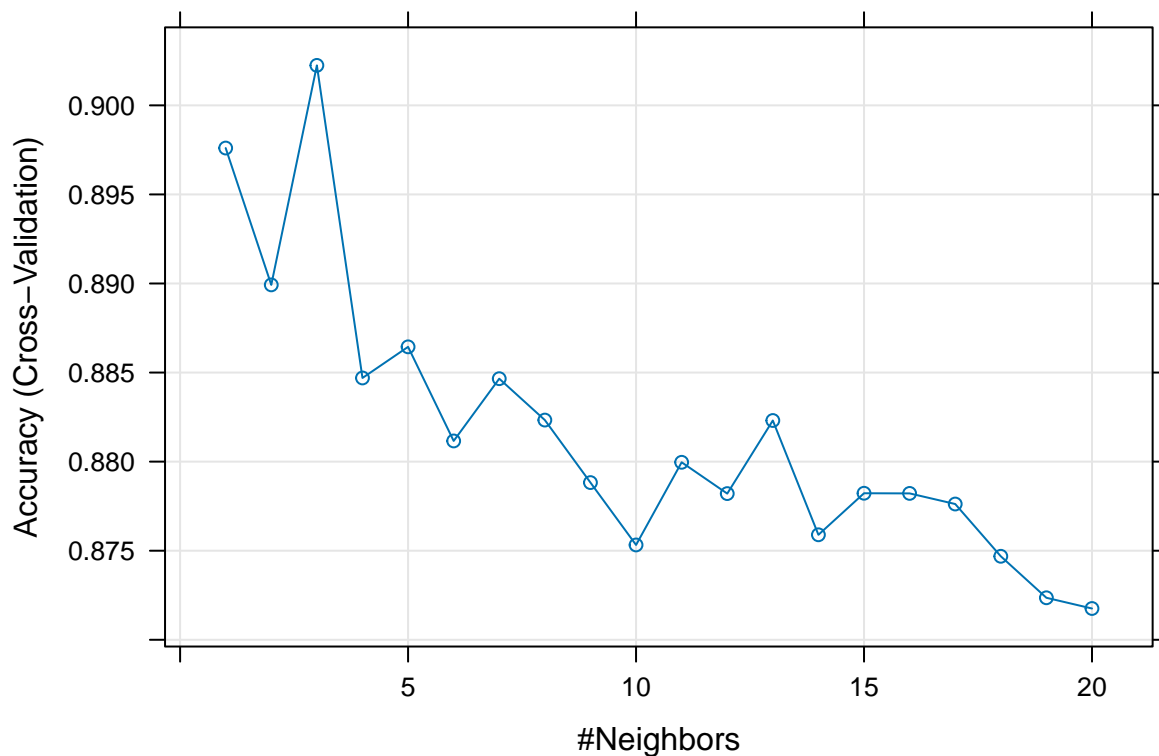
split off to use as a testing set (train\_test\_data, 192 observations) for our actual model training data (train\_train\_data, 1708 observations). We will train three different classification models (K Nearest Neighbors, Regression Tree, Random Forest) with our train\_train data set, and make predictions using those models on the train\_test data set.

**K Nearest Neighbors** As a first model, we will attempt to predict fetal health outcome using all other variables in the data set. First, we will try a number of different values of k to see what the best value is for our model.

```
knn_model <- train(fetal_health ~ ., data=train_train_data, method="knn",  
                  trControl = trainControl(method = "cv"),  
                  tuneGrid = data.frame(k=seq(1,20)))
```

Let's look at a graph of the Accuracy for each potential value of k and see what the best value of the number of nearest neighbors is to use.

```
plot(knn_model)
```



From the graph, it is evident that **3** is the best value to use for the number of nearest neighbors. Now, let's use that value of k to train the model, use the model to make predictions in our train\_test\_data set, and then look at a confusion matrix of how well it has performed.

```
best_model_knn <- knn3(fetal_health ~ ., data=train_train_data,  
                      k = knn_model$bestTune)  
predictions_knn <- predict(best_model_knn, train_test_data, type="class")
```

```
cm_knn <- confusionMatrix(predictions_knn, train_test_data$fetal_health)
cm_knn
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    1    2    3
##              1 145    8    3
##              2    2   18    1
##              3    2    1   12
##
## Overall Statistics
##
##              Accuracy : 0.9115
##              95% CI : (0.862, 0.9476)
##              No Information Rate : 0.776
##              P-Value [Acc > NIR] : 6.332e-07
##
##              Kappa : 0.7453
##
## McNemar's Test P-Value : 0.2839
##
## Statistics by Class:
##
##              Class: 1 Class: 2 Class: 3
## Sensitivity      0.9732  0.66667  0.75000
## Specificity      0.7442  0.98182  0.98295
## Pos Pred Value   0.9295  0.85714  0.80000
## Neg Pred Value    0.8889  0.94737  0.97740
## Prevalence       0.7760  0.14062  0.08333
## Detection Rate   0.7552  0.09375  0.06250
## Detection Prevalence 0.8125  0.10938  0.07812
## Balanced Accuracy 0.8587  0.82424  0.86648
```

Our first model has an accuracy of **91.15%**. That's not too bad for a first model. The good news is that of the 16 fetuses most at risk (those with a known outcome of 3), the model correctly puts all but 3 of them into at least the "suspicious" category (2), which would at least produce more scrutiny on these fetuses.

But can we do better? We saw during data exploration above that there are four basic kinds of data, including baseline heart rate, accelerations/decelerations, variability, and histogram measurements. Let's try training models that use only 1 of these types of variables each. Once we have that data, we can perhaps combine the best ones to get an even better prediction. We will follow the same code as before, but this time during training, instead of using "fetal\_health ~ ." to train against all data, we will replace the "." with the relevant data, for instance "baseline.value." Doing so yields the following results.

Variables_Used	Best_k	Accuracy
Baseline Heart Rate	16	79.17
Accel/Decel	5	80.73
Variability	11	87.50
Histogram	8	86.98

Right off the bat, we can see that no one set of data is as predictive as the combination of all of them, showing off how more data is usually a good thing. Unsurprisingly, using only the baseline heart rate is

not that predictive, since there is such a wide range of acceptable values. Accelerations and decelerations also seem to be less important, which may not be that surprising given the low standard deviations of these values we saw earlier. Conversely, using variability or histogram data alone comes closest to the original accuracy, showing how important they are. Let's try combining the two most accurate metrics, variability and histogram, leaving out the less important metrics, and seeing what our accuracy looks like.

```
knn_model_v_h <- train(fetal_health ~ abnormal_short_term_variability +
  mean_value_of_short_term_variability +
  percentage_of_time_with_abnormal_long_term_variability +
  mean_value_of_long_term_variability + histogram_width +
  histogram_min + histogram_max +
  histogram_number_of_peaks + histogram_number_of_zeroes +
  histogram_mode + histogram_mean + histogram_median +
  histogram_variance + histogram_tendency,
  data=train_train_data, method="knn",
  trControl = trainControl(method = "cv"),
  tuneGrid = data.frame(k=seq(1,20)))
best_model_v_h <- knn3(fetal_health ~ abnormal_short_term_variability +
  mean_value_of_short_term_variability +
  percentage_of_time_with_abnormal_long_term_variability +
  mean_value_of_long_term_variability + histogram_width +
  histogram_min + histogram_max +
  histogram_number_of_peaks + histogram_number_of_zeroes +
  histogram_mode + histogram_mean + histogram_median +
  histogram_variance + histogram_tendency,
  data=train_train_data,
  k = knn_model_v_h$bestTune)
predictions_v_h <- predict(best_model_v_h, train_test_data, type="class")
cm_v_h <- confusionMatrix(predictions_v_h, train_test_data$fetal_health)

cm_v_h
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1    2    3
##           1 145    7    3
##           2   3   18    1
##           3   1    2   12
##
## Overall Statistics
##
##           Accuracy : 0.9115
##           95% CI : (0.862, 0.9476)
##           No Information Rate : 0.776
##           P-Value [Acc > NIR] : 6.332e-07
##
##           Kappa : 0.7477
##
## Mcnemar's Test P-Value : 0.402
##
## Statistics by Class:
##
##           Class: 1 Class: 2 Class: 3
```

```
## Sensitivity      0.9732  0.66667  0.75000
## Specificity      0.7674  0.97576  0.98295
## Pos Pred Value   0.9355  0.81818  0.80000
## Neg Pred Value   0.8919  0.94706  0.97740
## Prevalence       0.7760  0.14062  0.08333
## Detection Rate   0.7552  0.09375  0.06250
## Detection Prevalence 0.8073  0.11458  0.07812
## Balanced Accuracy 0.8703  0.82121  0.86648
```

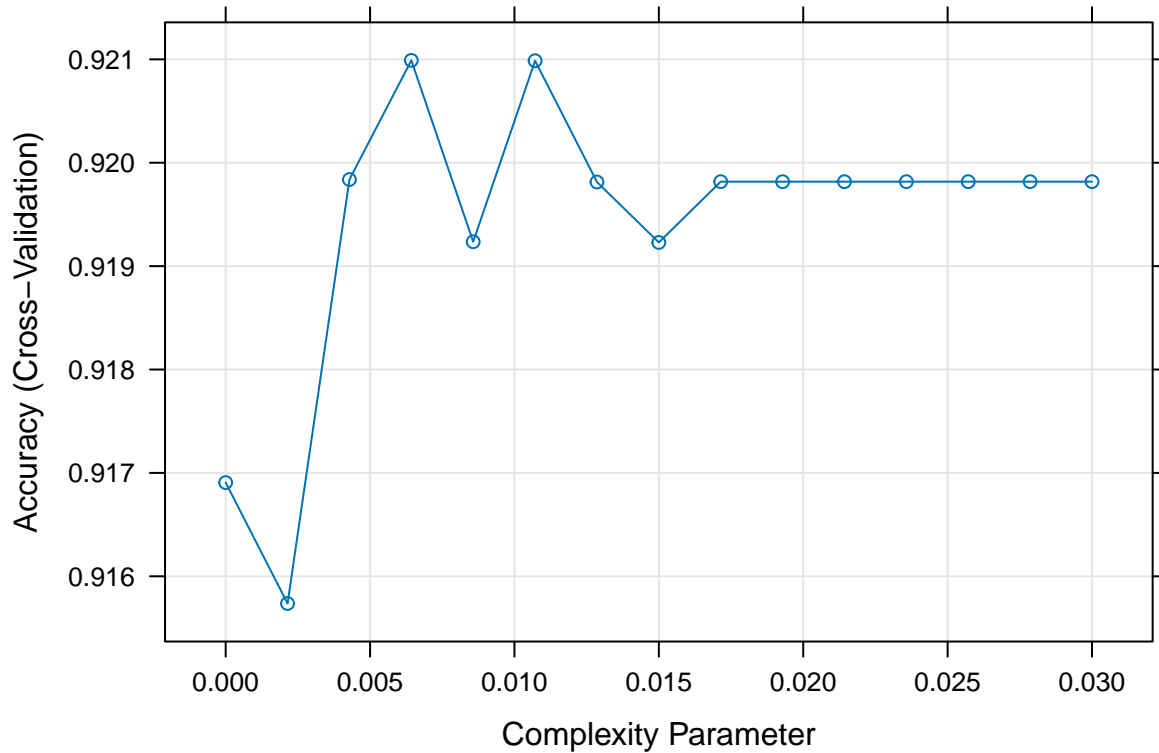
We get the same accuracy as the original model using all of the data, proving the importance of variability and the histogram data. Interestingly, although the accuracy is the same, the predictions are not quite the same, with two fetuses (one each in the health outcome 1 and 2 columns) being changed in their predicted outcome. However, predictions for fetuses in health outcome 3 did not change. For now, since the accuracy is equivalent, we'll stick with using the model with all of the data. Let's start a table of the various models we've tried.

Model	Accuracy
K Nearest Neighbors	91.15

**Regression Tree** Next, let's try a regression tree model. First, we'll tune the complexity parameter using cross validation to see what the proper value is, and plot the resulting graph.

```
rpart <- train(fetal_health ~ ., method = "rpart",
               trControl = trainControl(method = "cv"),
               tuneGrid = data.frame(cp = seq(0.0, 0.03, len = 15)),
               data = train_train_data)

plot(rpart)
```



As can be seen in the graph, the best value of the complexity parameter is **0.006**. Now, let's use that parameter to predict the fetal health outcomes in our train\_test data set, using all data in the dataset.

```
rpart_predict <- predict(rpart, train_test_data)
cm_rpart <- confusionMatrix(rpart_predict, train_test_data$fetal_health)
cm_rpart
```

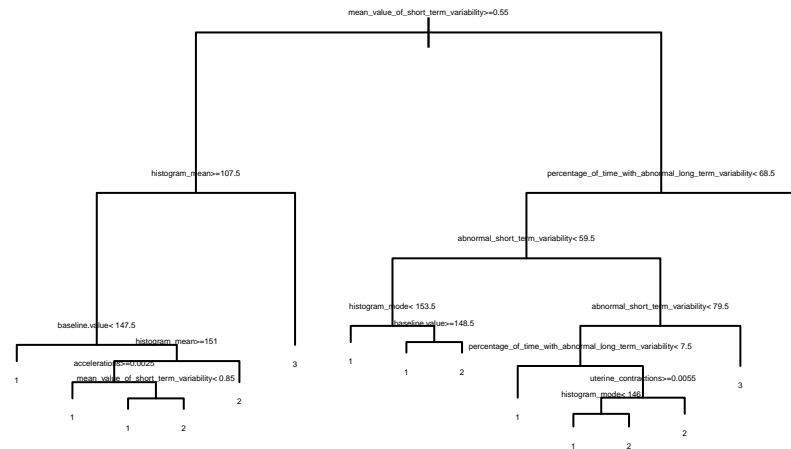
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1    2    3
##           1 144    8    3
##           2   3   18    0
##           3   2    1   13
##
## Overall Statistics
##
##           Accuracy : 0.9115
##           95% CI : (0.862, 0.9476)
##           No Information Rate : 0.776
##           P-Value [Acc > NIR] : 6.332e-07
##
##           Kappa : 0.7479
##
##           McNemar's Test P-Value : 0.3243
##
```

```
## Statistics by Class:
##
##
##          Class: 1 Class: 2 Class: 3
## Sensitivity      0.9664  0.66667  0.81250
## Specificity      0.7442  0.98182  0.98295
## Pos Pred Value   0.9290  0.85714  0.81250
## Neg Pred Value    0.8649  0.94737  0.98295
## Prevalence       0.7760  0.14062  0.08333
## Detection Rate    0.7500  0.09375  0.06771
## Detection Prevalence 0.8073  0.10938  0.08333
## Balanced Accuracy 0.8553  0.82424  0.89773
```

With an accuracy of **91.15%**, this model is just as good as the KNN model. It does do slightly better at classifying the most at-risk fetuses, moving 1 fetus from a predicted outcome of 2 to a predicted outcome of 3. Let's add this value to our results table.

Model	Accuracy
K Nearest Neighbors	91.15
Regression Tree	91.15

Although it is somewhat unwieldy, due to the regrettably lengthy variable names, let's take a look at the decision tree and see what the model thinks are the most important variables, and their important cutoff points.



Interestingly, the regression tree thinks that the mean value of short term variability is the most important predictor. In fact, the model immediately places a fetus in fetal outcome 3 if they have a mean value of

short term variability of  $<0.55$ , and a percentage of time in abnormal long term variability  $> 68.5\%$  of the time, showing both the importance of having high mean short term variability, and a lower amount of time spent in long term variability. There are also several decision points relating to histogram values of mean and mode. This matches quite well with the results of the knn model, which showed that, when tested in isolation, variability and the histogram were the two most predictive types of data.

Also of note, the regression tree uses only 8 of the 21 variables in the dataset to make predictions. Let's try training a regression tree model using only those 8 parameters, to see if that improves the accuracy.

```
rpart8 <- train(fetal_health ~ mean_value_of_short_term_variability +
               percentage_of_time_with_abnormal_long_term_variability +
               abnormal_short_term_variability + histogram_mean +
               histogram_mode + baseline.value + uterine_contractions +
               accelerations, method = "rpart",
               trControl = trainControl(method = "cv"),
               tuneGrid = data.frame(cp = seq(0.0, 0.03, len = 15)),
               data = train_train_data)
rpart_predict8 <- predict(rpart8, train_test_data)
cm_rpart8 <- confusionMatrix(rpart_predict8, train_test_data$fetal_health)
cm_rpart8
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1    2    3
##           1 144    6    3
##           2   3   20    0
##           3    2    1   13
##
## Overall Statistics
##
##           Accuracy : 0.9219
##           95% CI : (0.8744, 0.9556)
##           No Information Rate : 0.776
##           P-Value [Acc > NIR] : 6.26e-08
##
##           Kappa : 0.7817
##
## Mcnemar's Test P-Value : 0.5319
##
## Statistics by Class:
##
##           Class: 1 Class: 2 Class: 3
## Sensitivity      0.9664  0.7407  0.81250
## Specificity      0.7907  0.9818  0.98295
## Pos Pred Value   0.9412  0.8696  0.81250
## Neg Pred Value   0.8718  0.9586  0.98295
## Prevalence       0.7760  0.1406  0.08333
## Detection Rate   0.7500  0.1042  0.06771
## Detection Prevalence 0.7969  0.1198  0.08333
## Balanced Accuracy 0.8786  0.8613  0.89773
```

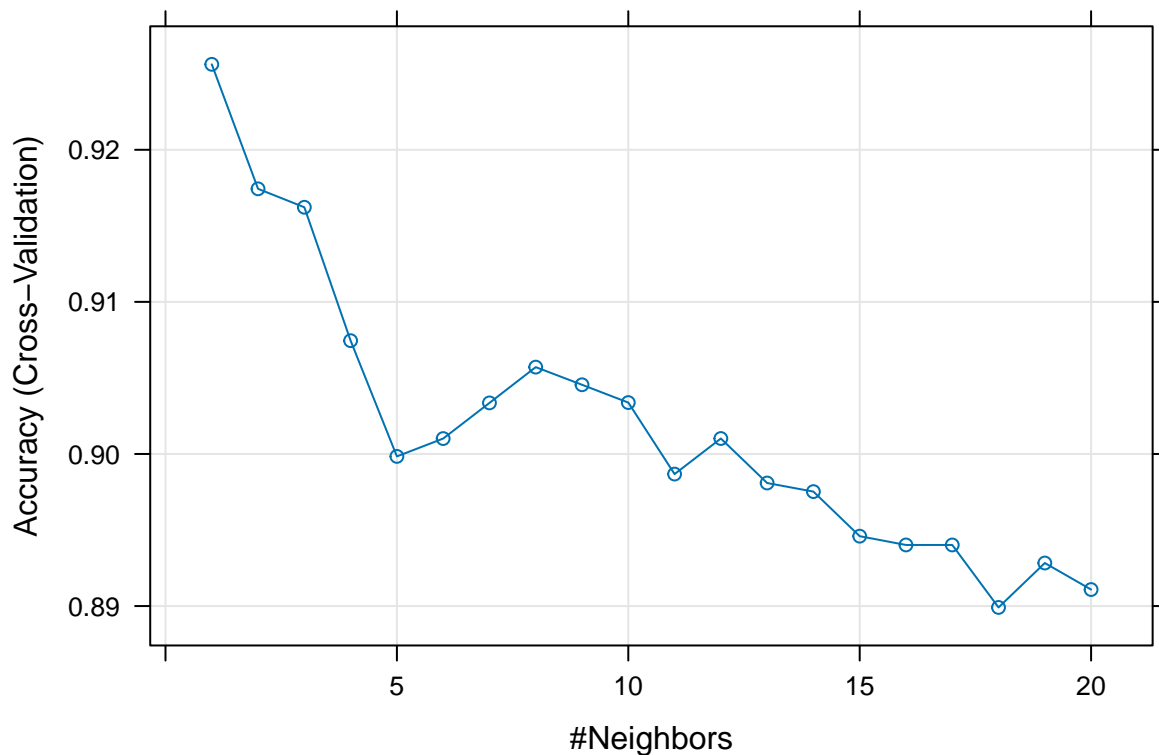
Using only the 8 most important parameters does improve the prediction accuracy to a small extent, up to **92.19%**, proving their predictive ability. The improved accuracy affects mostly fetuses in health outcome 2, correctly moving two fetuses from a prediction of 1 to a prediction of 2. Let's add it to our results table.



Model	Accuracy
K Nearest Neighbors	91.15
Regression Tree	91.15
Regression Tree - best 8 parameters	92.19

**K Nearest Neighbors Redux** The eight variables that the regression tree uses are mean\_value\_of\_short\_term\_variability, percentage\_of\_time\_with\_abnormal\_long\_term\_variability, abnormal\_short\_term\_variability, histogram\_mean, histogram\_mode, baseline.value, uterine\_contractions and accelerations. Let's try training a knn model using only those parameters and see if it performs any better than the knn model using all the data.

```
knn_model_best_params <- train(fetal_health ~ mean_value_of_short_term_variability +
  percentage_of_time_with_abnormal_long_term_variability +
  abnormal_short_term_variability + histogram_mean +
  histogram_mode + baseline.value +
  uterine_contractions + accelerations,
  data=train_train_data, method="knn",
  trControl = trainControl(method = "cv"),
  tuneGrid = data.frame(k=seq(1,20)))
plot(knn_model_best_params)
```



```
best_model_best_params <- knn3(fetal_health ~ mean_value_of_short_term_variability +
  percentage_of_time_with_abnormal_long_term_variability +
  abnormal_short_term_variability + histogram_mean +
```

```

        histogram_mode + baseline.value +
        uterine_contractions + accelerations,
        data=train_train_data,
        k = knn_model_best_params$bestTune)
predictions_best_params <- predict(best_model_best_params,
        train_test_data,
        type="class")
cm_best_params <- confusionMatrix(predictions_best_params,
        train_test_data$fetal_health)
cm_best_params

```

```

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1    2    3
##           1 138    7    2
##           2   9   16    1
##           3   2    4   13
##
## Overall Statistics
##
##           Accuracy : 0.8698
##           95% CI : (0.8138, 0.9139)
##           No Information Rate : 0.776
##           P-Value [Acc > NIR] : 0.0006977
##
##           Kappa : 0.656
##
## Mcnemar's Test P-Value : 0.5620941
##
## Statistics by Class:
##
##           Class: 1 Class: 2 Class: 3
## Sensitivity         0.9262  0.59259  0.81250
## Specificity         0.7907  0.93939  0.96591
## Pos Pred Value      0.9388  0.61538  0.68421
## Neg Pred Value      0.7556  0.93373  0.98266
## Prevalence          0.7760  0.14062  0.08333
## Detection Rate      0.7188  0.08333  0.06771
## Detection Prevalence 0.7656  0.13542  0.09896
## Balanced Accuracy    0.8584  0.76599  0.88920

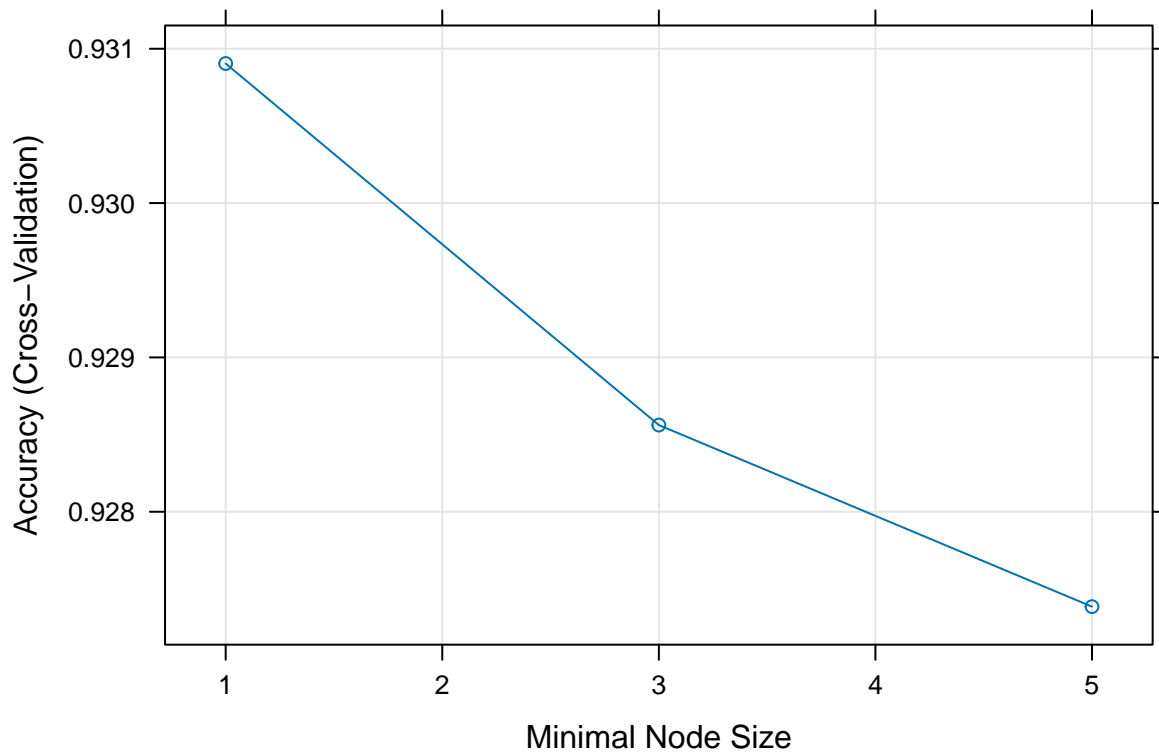
```

Disappointingly, that doesn't seem to have worked quite as well as hoped, with a marked reduction in accuracy (**86.98%**) compared to all other models tested. But it was worth a try! We'll add it to the results table.

Model	Accuracy
K Nearest Neighbors	91.15
Regression Tree	91.15
Regression Tree - best 8 parameters	92.19
K Nearest Neighbors - best 8 parameters	86.98

**Random Forest** Let's now try a random forest model, using cross validation to choose the best node size parameter.

```
rf_model <- train(fetal_health ~ ., method="Rborist",
                  trControl = trainControl(method = "cv"),
                  tuneGrid = data.frame(predFixed = 2,
                                       minNode = c(1,3,5)),
                  data = train_train_data)
plot(rf_model)
```



```
rf_predict <- predict(rf_model, train_test_data)
cm_rf <- confusionMatrix(rf_predict, train_test_data$fetal_health)
cm_rf
```

## Confusion Matrix and Statistics

##

##           Reference

## Prediction   1   2   3

##           1 148   6   4

##           2   1 17   1

##           3   0   4 11

##

## Overall Statistics

##

##                   Accuracy : 0.9167

```
##          95% CI : (0.8682, 0.9516)
##    No Information Rate : 0.776
##    P-Value [Acc > NIR] : 2.057e-07
##
##          Kappa : 0.7556
##
##    McNemar's Test P-Value : 0.02474
##
## Statistics by Class:
##
##          Class: 1 Class: 2 Class: 3
## Sensitivity      0.9933  0.62963  0.68750
## Specificity      0.7674  0.98788  0.97727
## Pos Pred Value   0.9367  0.89474  0.73333
## Neg Pred Value   0.9706  0.94220  0.97175
## Prevalence       0.7760  0.14062  0.08333
## Detection Rate   0.7708  0.08854  0.05729
## Detection Prevalence 0.8229  0.09896  0.07812
## Balanced Accuracy 0.8804  0.80875  0.83239
```

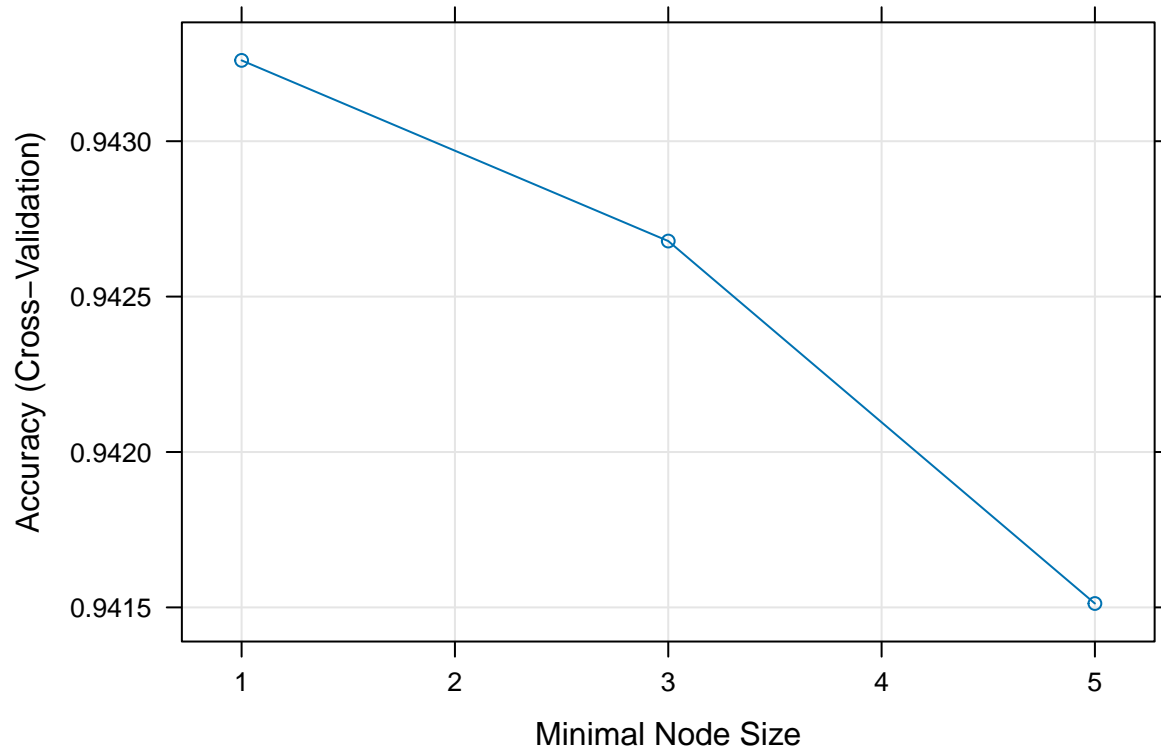
The random forest model results in a very slight gain of accuracy (**91.67%**) over the KNN and Regression Tree models using all of the data, although slightly worse than the regression tree model using only the 8 most important parameters. It also performs worse than all other models, even the relatively inaccurate KNN model using just the 8 parameters, at predicting fetuses in the most important health outcome category (3). That's certainly not ideal! We'll add it to our results table.

Model	Accuracy
K Nearest Neighbors	91.15
Regression Tree	91.15
Regression Tree - best 8 parameters	92.19
K Nearest Neighbors - best 8 parameters	86.98
Random Forest	91.67

For completeness, let's test the random forest model with just the eight most important parameters as well.

```
rf8_model <- train(fetal_health ~ mean_value_of_short_term_variability +
  percentage_of_time_with_abnormal_long_term_variability +
  abnormal_short_term_variability + histogram_mean +
  histogram_mode + baseline.value + uterine_contractions +
  accelerations, method="Rborist",
  trControl = trainControl(method = "cv"),
  tuneGrid = data.frame(predFixed = 2, minNode = c(1,3,5)),
  data = train_train_data)

plot(rf8_model)
```



```
rf8_predict <- predict(rf8_model, train_test_data)
cm_rf8 <- confusionMatrix(rf8_predict, train_test_data$fetal_health)
cm_rf8
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1    2    3
##           1 148    4    3
##           2   1   19    0
##           3   0    4   13
##
## Overall Statistics
##
##           Accuracy : 0.9375
##           95% CI : (0.8934, 0.9673)
##           No Information Rate : 0.776
##           P-Value [Acc > NIR] : 1.144e-09
##
##           Kappa : 0.8222
##
## Mcnemar's Test P-Value : 0.03207
##
## Statistics by Class:
##
```

```
##               Class: 1 Class: 2 Class: 3
## Sensitivity    0.9933  0.70370  0.81250
## Specificity    0.8372  0.99394  0.97727
## Pos Pred Value 0.9548  0.95000  0.76471
## Neg Pred Value 0.9730  0.95349  0.98286
## Prevalence     0.7760  0.14062  0.08333
## Detection Rate 0.7708  0.09896  0.06771
## Detection Prevalence 0.8073  0.10417  0.08854
## Balanced Accuracy 0.9152  0.84882  0.89489
```

We have a winner! Random forest, using only the eight most important parameters, as discovered by the regression tree method, yields an accuracy of **93.75%**, our best model yet. It also does a much better job than the random forest model using all of the data at predicting health outcomes for fetuses in outcome 3. We'll call this our final model, and add the result to our table.

Model	Accuracy
K Nearest Neighbors	91.15
Regression Tree	91.15
Regression Tree - best 8 parameters	92.19
K Nearest Neighbors - best 8 parameters	86.98
Random Forest	91.67
Random Forest - best 8 parameters	93.75

**Final Test** Now that we've identified our best model, let's test it with the original testing data, which we have not touched to this point, and see how well it performs.

```
rf_final_predict <- predict(rf8_model, test_data)
cm_rf_final <- confusionMatrix(rf_final_predict, test_data$fetal_health)
cm_rf_final
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction    1    2    3
##           1 164    8    1
##           2   1   22    1
##           3    0    0   16
##
## Overall Statistics
##
##               Accuracy : 0.9484
##               95% CI : (0.9095, 0.9739)
##       No Information Rate : 0.7746
##       P-Value [Acc > NIR] : 2.917e-12
##
##               Kappa : 0.8519
##
## Mcnemar's Test P-Value : 0.059
##
## Statistics by Class:
##
##               Class: 1 Class: 2 Class: 3
```

## Sensitivity	0.9939	0.7333	0.88889
## Specificity	0.8125	0.9891	1.00000
## Pos Pred Value	0.9480	0.9167	1.00000
## Neg Pred Value	0.9750	0.9577	0.98985
## Prevalence	0.7746	0.1408	0.08451
## Detection Rate	0.7700	0.1033	0.07512
## Detection Prevalence	0.8122	0.1127	0.07512
## Balanced Accuracy	0.9032	0.8612	0.94444

Our final model was able to predict the health outcomes of the test data set with the best accuracy yet - **94.84%**! The model correctly classifies 16/18 fetuses in health outcome 3 correctly, and places one of the other two in outcome 2, a very accurate prediction set for the most at risk fetuses. Let's finalize our results table.

Model	Accuracy
K Nearest Neighbors	91.15
Regression Tree	91.15
Regression Tree - best 8 parameters	92.19
K Nearest Neighbors - best 8 parameters	86.98
Random Forest	91.67
Random Forest - best 8 parameters	93.75
Final Model on Holdout Test Data	94.84

## Conclusions

The health outcome of a fetus during pregnancy and labor/delivery are of paramount importance to parents, doctors, and hospitals. Utilizing a large set of data, we were able to design a model that could predict fetal health outcomes with very high accuracy. Such predictions will allow for health care professionals to take the steps necessary to ensure the best possible care, at the earliest possible time point, for fetuses that need it most.

## References

The data set for this project was originally downloaded from: <https://www.kaggle.com/datasets/yaminh/fetal-health-monitoring-dataset>

Urdal J, Engan K, Eftestøl T, Haaland SH, Kamala B, Mdoe P, Kidanto H, Ersdal H. Fetal heart rate development during labour. *Biomed Eng Online*. 2021 Mar 16;20(1):26. doi: 10.1186/s12938-021-00861-z. PMID: 33726745; PMCID: PMC7962212.

Pildner von Steinburg S, Boulesteix AL, Lederer C, Grunow S, Schiermeier S, Hatzmann W, Schneider KT, Daumer M. What is the "normal" fetal heart rate? *PeerJ*. 2013 Jun 4;1:e82. doi: 10.7717/peerj.82. PMID: 23761161; PMCID: PMC3678114.

López-Justo C, Pliego-Carrillo AC, Ledesma-Ramírez CI, Mendieta-Zerón H, Peña-Castillo MÁ, Echeverría JC, Rodríguez-Arce J, Reyes-Lagos JJ. Differences in the Asymmetry of Beat-to-Beat Fetal Heart Rate Accelerations and Decelerations at Preterm and Term Active Labor. *Sensors (Basel)*. 2021 Dec 10;21(24):8249. doi: 10.3390/s21248249. PMID: 34960343; PMCID: PMC8704786.

Tarvonen MJ, Lear CA, Andersson S, Gunn AJ, Teramo KA. Increased variability of fetal heart rate during labour: a review of preclinical and clinical studies. *BJOG*. 2022 Nov;129(12):2070-2081. doi: 10.1111/1471-0528.17234. Epub 2022 Jun 5. PMID: 35596699; PMCID: PMC9796294.

Griffin MP, Lake DE, Bissonette EA, Harrell FE Jr, O'Shea TM, Moorman JR. Heart rate characteristics: novel physiomarkers to predict neonatal infection and death. *Pediatrics*. 2005 Nov;116(5):1070-4. doi: 10.1542/peds.2004-2461. PMID: 16263991.