

Fetal Health Visual Analysis

Anurag Dinesh Karmarkar

-----StartOfCode-----

```
In [2]: #Importing required packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: #Creating The dataset variable
fetal_p = 'fetal_health.csv'
```

```
In [4]: #Reading the dataset and displaying the head
fetal = pd.read_csv(fetal_p)
fetal.head(10)
```

Out[4]:

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_dece
0	120.0	0.000	0.0	0.000	0.000	
1	132.0	0.006	0.0	0.006	0.003	
2	133.0	0.003	0.0	0.008	0.003	
3	134.0	0.003	0.0	0.008	0.003	
4	132.0	0.007	0.0	0.008	0.000	
5	134.0	0.001	0.0	0.010	0.009	
6	134.0	0.001	0.0	0.013	0.008	
7	122.0	0.000	0.0	0.000	0.000	
8	122.0	0.000	0.0	0.002	0.000	
9	122.0	0.000	0.0	0.003	0.000	

10 rows × 22 columns



```
In [5]: # Data Description
# This dataset indicates the various factors instrumental in determining the h
ealth of a fetus.
# The dataset features the Cardiotocograms data for many fetuses. Cardiotocogr
am is a device which monitors
#the fetal health by using various factors. The factors are mentioned in the d
ataset.
# There are also some features related to a histogram, such as histogram mean,
max, etc. which
#indicate the features of the histogram plotted using each record.
```

-----DataCleaning-----

```
In [6]: #A small code to look at the type of data stored in each and every column
print(fetal.shape)
for i in range(22):
    print(type(fetal.iloc[0,i]))
```

```
(2126, 22)
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
<class 'numpy.float64'>
```

```
In [7]: # Because we have all data as numerical hence describing the dataset Transposing for a better view
        fetal.describe().T
```

Out[7]:

	count	mean	std	min	max
baseline_value	2126.0	133.303857	9.840844	106.0	126.0
accelerations	2126.0	0.003178	0.003866	0.0	0.004
fetal_movement	2126.0	0.009481	0.046666	0.0	0.01
uterine_contractions	2126.0	0.004366	0.002946	0.0	0.005
light_decelerations	2126.0	0.001889	0.002960	0.0	0.003
severe_decelerations	2126.0	0.000003	0.000057	0.0	0.000
prolongued_decelerations	2126.0	0.000159	0.000590	0.0	0.001
abnormal_short_term_variability	2126.0	46.990122	17.192814	12.0	32.0
mean_value_of_short_term_variability	2126.0	1.332785	0.883241	0.2	1.4
percentage_of_time_with_abnormal_long_term_variability	2126.0	9.846660	18.396880	0.0	100.0
mean_value_of_long_term_variability	2126.0	8.187629	5.628247	0.0	20.0
histogram_width	2126.0	70.445908	38.955693	3.0	320.0
histogram_min	2126.0	93.579492	29.560212	50.0	60.0
histogram_max	2126.0	164.025400	17.944183	122.0	150.0
histogram_number_of_peaks	2126.0	4.068203	2.949386	0.0	10.0
histogram_number_of_zeroes	2126.0	0.323612	0.706059	0.0	1.0
histogram_mode	2126.0	137.452023	16.381289	60.0	120.0
histogram_mean	2126.0	134.610536	15.593596	73.0	120.0
histogram_median	2126.0	138.090310	14.466589	77.0	120.0
histogram_variance	2126.0	18.808090	28.977636	0.0	100.0
histogram_tendency	2126.0	0.320320	0.610829	-1.0	1.0
fetal_health	2126.0	1.304327	0.614377	1.0	2.0

```
In [8]: #checking dataset for the null values
        fetal.isnull().sum()
```

```
Out[8]: baseline value                                0
        accelerations                                0
        fetal_movement                               0
        uterine_contractions                         0
        light_decelerations                         0
        severe_decelerations                        0
        prolonged_decelerations                    0
        abnormal_short_term_variability             0
        mean_value_of_short_term_variability        0
        percentage_of_time_with_abnormal_long_term_variability 0
        mean_value_of_long_term_variability         0
        histogram_width                             0
        histogram_min                               0
        histogram_max                               0
        histogram_number_of_peaks                   0
        histogram_number_of_zeroes                  0
        histogram_mode                              0
        histogram_mean                              0
        histogram_median                            0
        histogram_variance                          0
        histogram_tendency                          0
        fetal_health                                0
        dtype: int64
```

```
In [9]: #checking dataset for the na values(because its all numeric)
        fetal.isna().sum()
```

```
Out[9]: baseline value                                0
        accelerations                                0
        fetal_movement                               0
        uterine_contractions                         0
        light_decelerations                         0
        severe_decelerations                        0
        prolonged_decelerations                    0
        abnormal_short_term_variability             0
        mean_value_of_short_term_variability        0
        percentage_of_time_with_abnormal_long_term_variability 0
        mean_value_of_long_term_variability         0
        histogram_width                             0
        histogram_min                               0
        histogram_max                               0
        histogram_number_of_peaks                   0
        histogram_number_of_zeroes                  0
        histogram_mode                              0
        histogram_mean                              0
        histogram_median                            0
        histogram_variance                          0
        histogram_tendency                          0
        fetal_health                                0
        dtype: int64
```

```
In [10]: # Getting The type of values in our target variable
fetal['fetal_health'].value_counts()
```

```
Out[10]: 1.0    1655
         2.0     295
         3.0     176
         Name: fetal_health, dtype: int64
```

```
In [11]: #As we are planing to do classification hence changing the target from numeric
         to Labels
fetal['fetal_health'].replace({1.0 : "Normal" , 2.0: "Suspect" , 3.0 : "Pathalo
         gical"},inplace = True)
fetal['fetal_health'].value_counts()
```

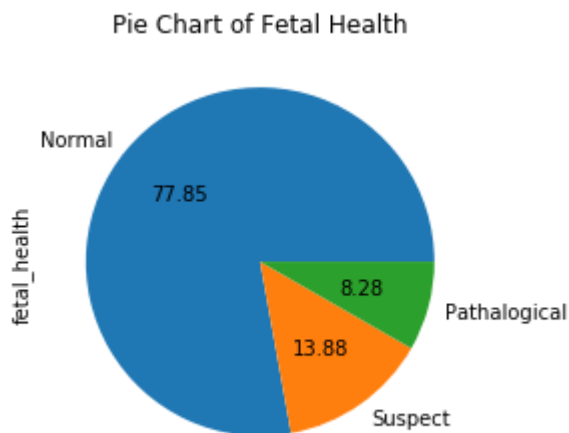
```
Out[11]: Normal          1655
         Suspect          295
         Pathological      176
         Name: fetal_health, dtype: int64
```

-----DataVisulatization-----

```
In [12]: # One variable Plots
```

```
In [13]: #Pie chart representing the spread of fetal Health(target variable) stages
fetal['fetal_health'].value_counts().plot(kind='pie', autopct = '%.2f', title =
         'Pie Chart of Fetal Health')
```

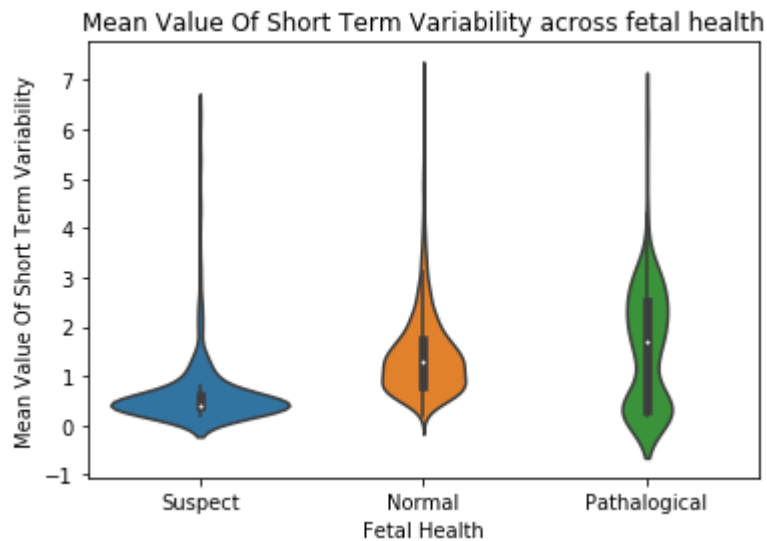
```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x28ab0109fc8>
```



```
In [14]: # We ran the feature seelction process for our dataset in WEKA and selected to
         p three features and thought of plottig their
         # spread across the target variable ie. fetal health
```

```
In [15]: # The first feature is "mean_value_of_short_term_variability" and for showing  
# its spread we have used the  
# violinplot from the seaborn package  
# By looking at this graph we can say that when the value of 'mean_value_of_sh  
# ort_term_variability' is between 0-1 than  
# chances for fetus being suspect are more  
# OTOH then value is around 1 its more possible that fetus is normal  
sns.violinplot(x=fetal['fetal_health'], y=fetal['mean_value_of_short_term_vari  
ability']) .set_title('Mean Value Of Short Term Variability across fetal healt  
h')  
plt.xlabel("Fetal Health")  
plt.ylabel("Mean Value Of Short Term Variability")
```

Out[15]: Text(0, 0.5, 'Mean Value Of Short Term Variability')

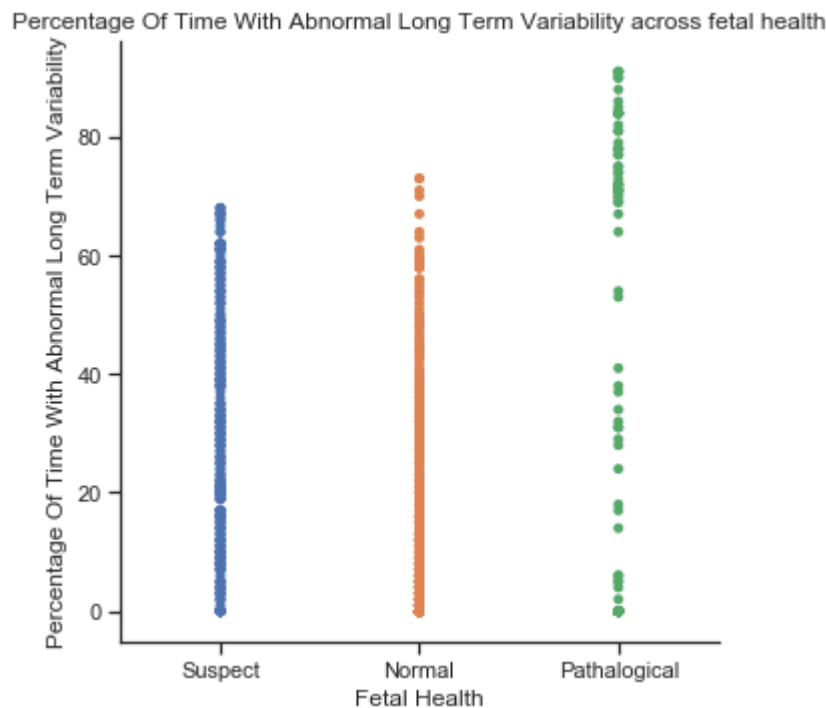


```

In [16]: # The next feature selected was percentage_of_time_with_abnormal_long_term_variability
          # and we have create a catplot
          # from the seaborn package
          # This plot shows the spread of percentage_of_time_with_abnormal_long_term_variability
          # against fetal health stages
          # By looking at the plot we can see that the range of pathological fetus's for
          # percentage_of_time_with_abnormal_long_term_variability
          # is more than the other 2 stages viz. Normal & suspect
          sns.set(style="ticks", color_codes=True)
          sns.catplot(x="fetal_health", y="percentage_of_time_with_abnormal_long_term_variability",
                      jitter=False, data=fetal)
          plt.title("Percentage Of Time With Abnormal Long Term Variability across fetal health")
          plt.xlabel("Fetal Health")
          plt.ylabel("Percentage Of Time With Abnormal Long Term Variability")

```

Out[16]: Text(12.085, 0.5, 'Percentage Of Time With Abnormal Long Term Variability')



```

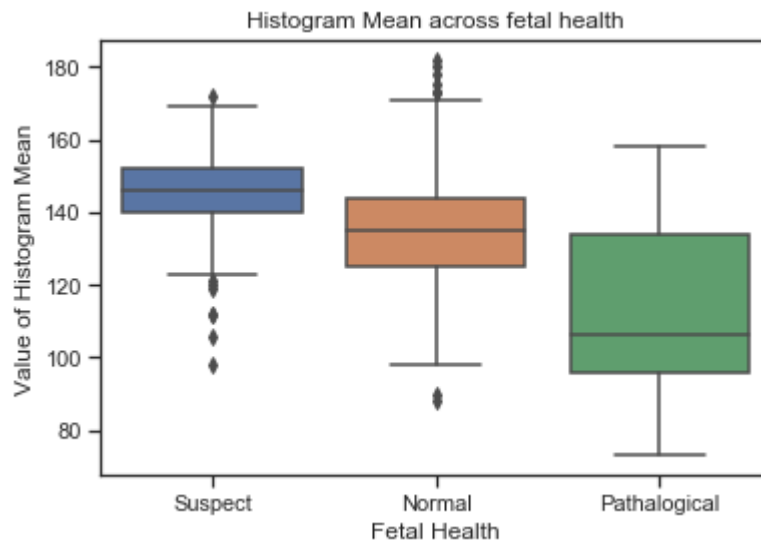
In [17]: # The next selected feature was Histogram Mean and we have repressed it across
          # fetal health using box plot
          # We can say that the average value of Histogram mean for pathological fetus
          # is way less in comparison to the Normal &
          # Suspect fetus
          sns.boxplot( x=fetal['fetal_health'], y=fetal['histogram_mean'] )
          plt.title("Histogram Mean across fetal health")
          plt.xlabel("Fetal Health")
          plt.ylabel("Value of Histogram Mean")

```

```

Out[17]: Text(0, 0.5, 'Value of Histogram Mean')

```



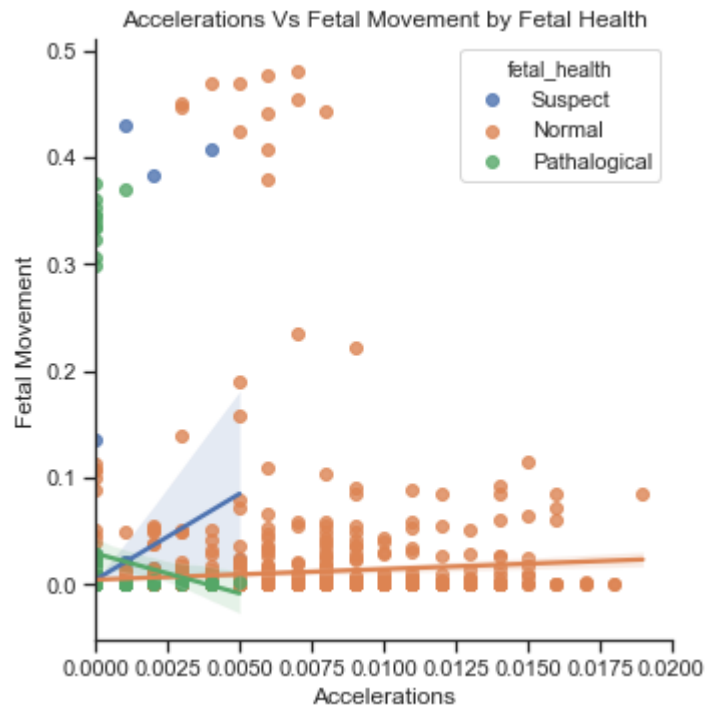
```

In [18]: # Two Variable Plots

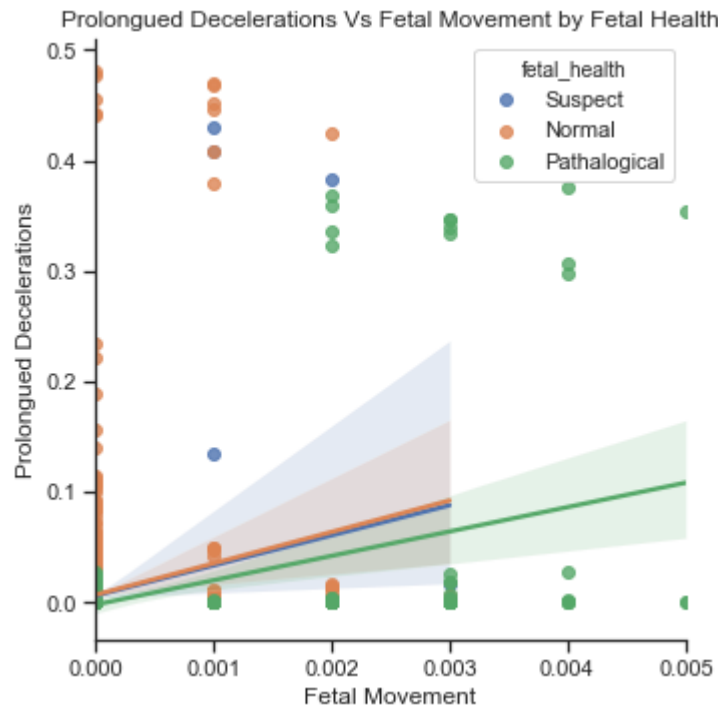
```



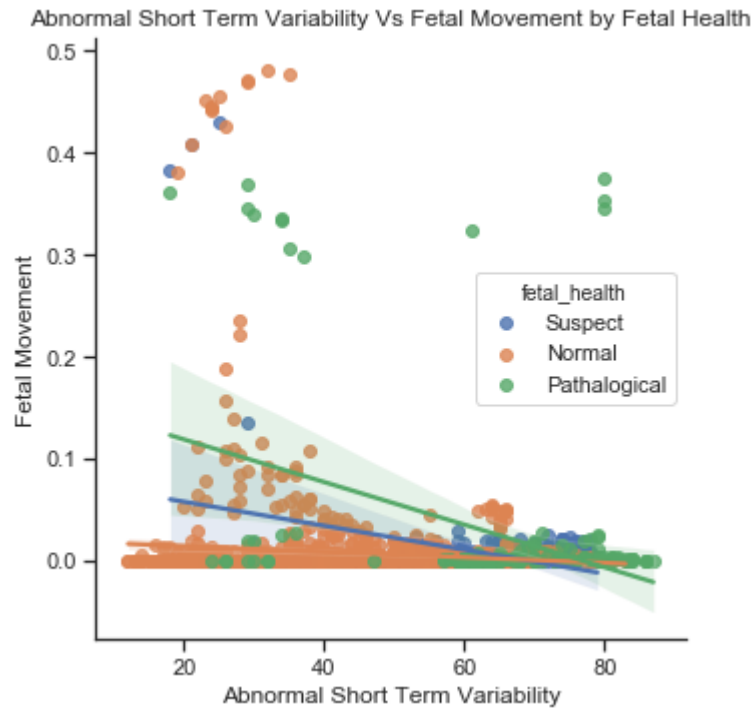
```
In [19]: # In this plot we have plotted acceleration with fetal movement according to fetal health
sns.lmplot(data =fetal,x="accelerations",y="fetal_movement",hue="fetal_health",
legend_out=False)
plt.title('Accelerations Vs Fetal Movement by Fetal Health')
plt.xlabel('Accelerations')
plt.ylabel('Fetal Movement')
plt.xlim([0.0,0.02])
plt.show()
```



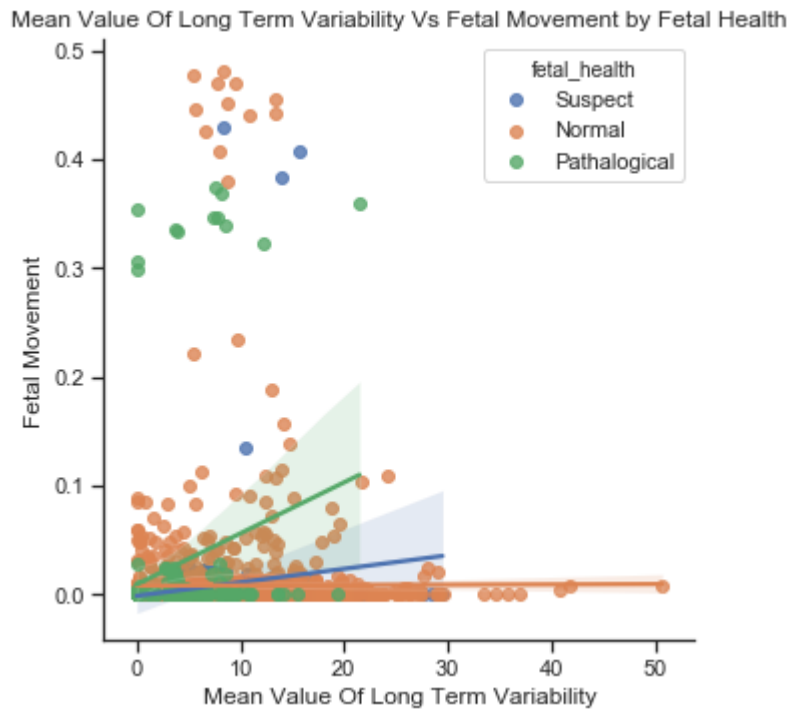
```
In [20]: # In this plot we have Prolonged Decelerations with fetal movement according
         # to fetal health
         sns.lmplot(data =fetal,x="prolonged_decelerations",y="fetal_movement",hue="fe
         tal_health",legend_out=False)
         plt.title('Prolongued Decelerations Vs Fetal Movement by Fetal Health')
         plt.xlabel('Fetal Movement')
         plt.ylabel('Prolongued Decelerations')
         plt.xlim([0.0,0.005])
         plt.show()
```



```
In [21]: # In this plot we have Abnormal Short Term Variability with fetal movement according to fetal health
sns.lmplot(data =fetal,x="abnormal_short_term_variability",y="fetal_movement",
hue="fetal_health",legend_out=False)
plt.title('Abnormal Short Term Variability Vs Fetal Movement by Fetal Health')
plt.xlabel('Abnormal Short Term Variability')
plt.ylabel('Fetal Movement')
plt.show()
```



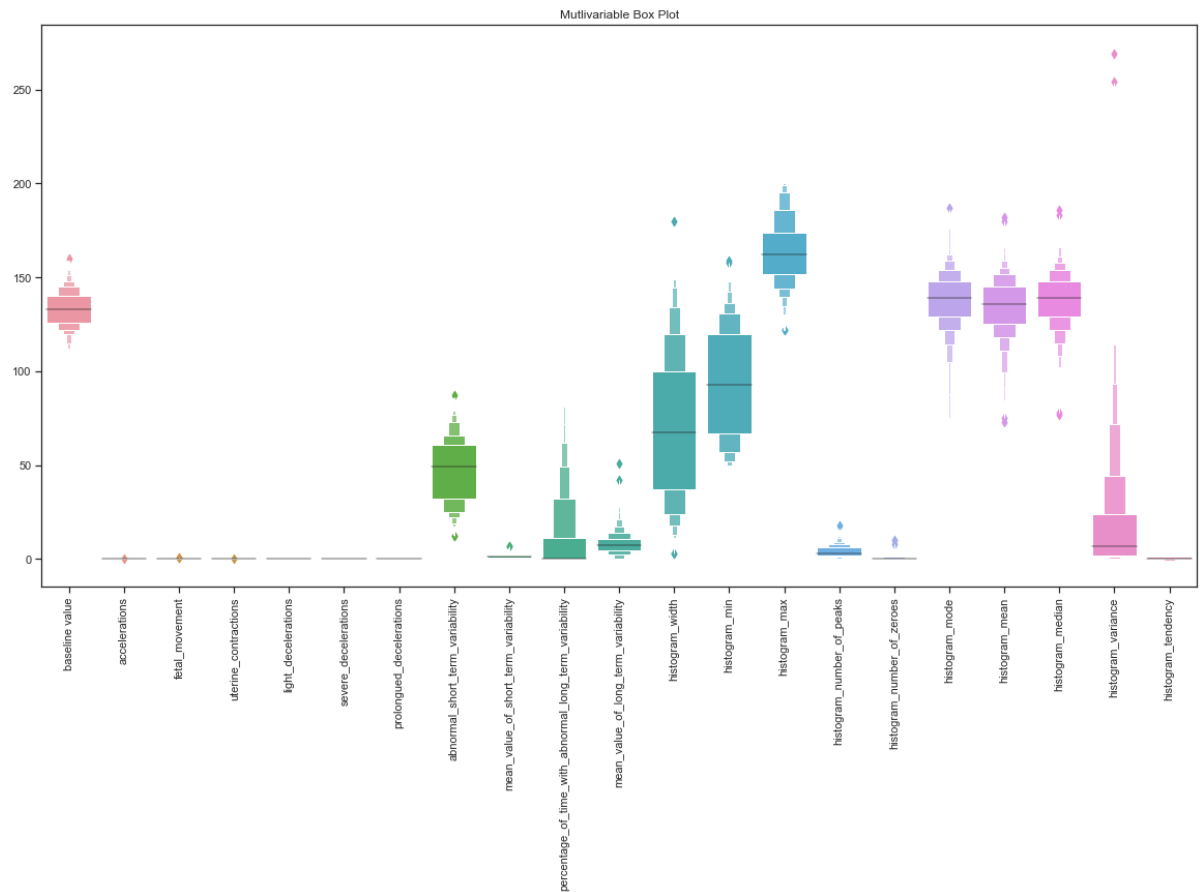
```
In [22]: # In this plot we have Mean Value Of Long Term Variability with fetal movement
         # according to fetal health
         sns.lmplot(data =fetal,x="mean_value_of_long_term_variability",y="fetal_movement",
         hue="fetal_health",legend_out=False)
         plt.title('Mean Value Of Long Term Variability Vs Fetal Movement by Fetal Health')
         plt.xlabel('Mean Value Of Long Term Variability')
         plt.ylabel('Fetal Movement')
         plt.show()
```



```
In [23]: # Three and mutli variable plots
```

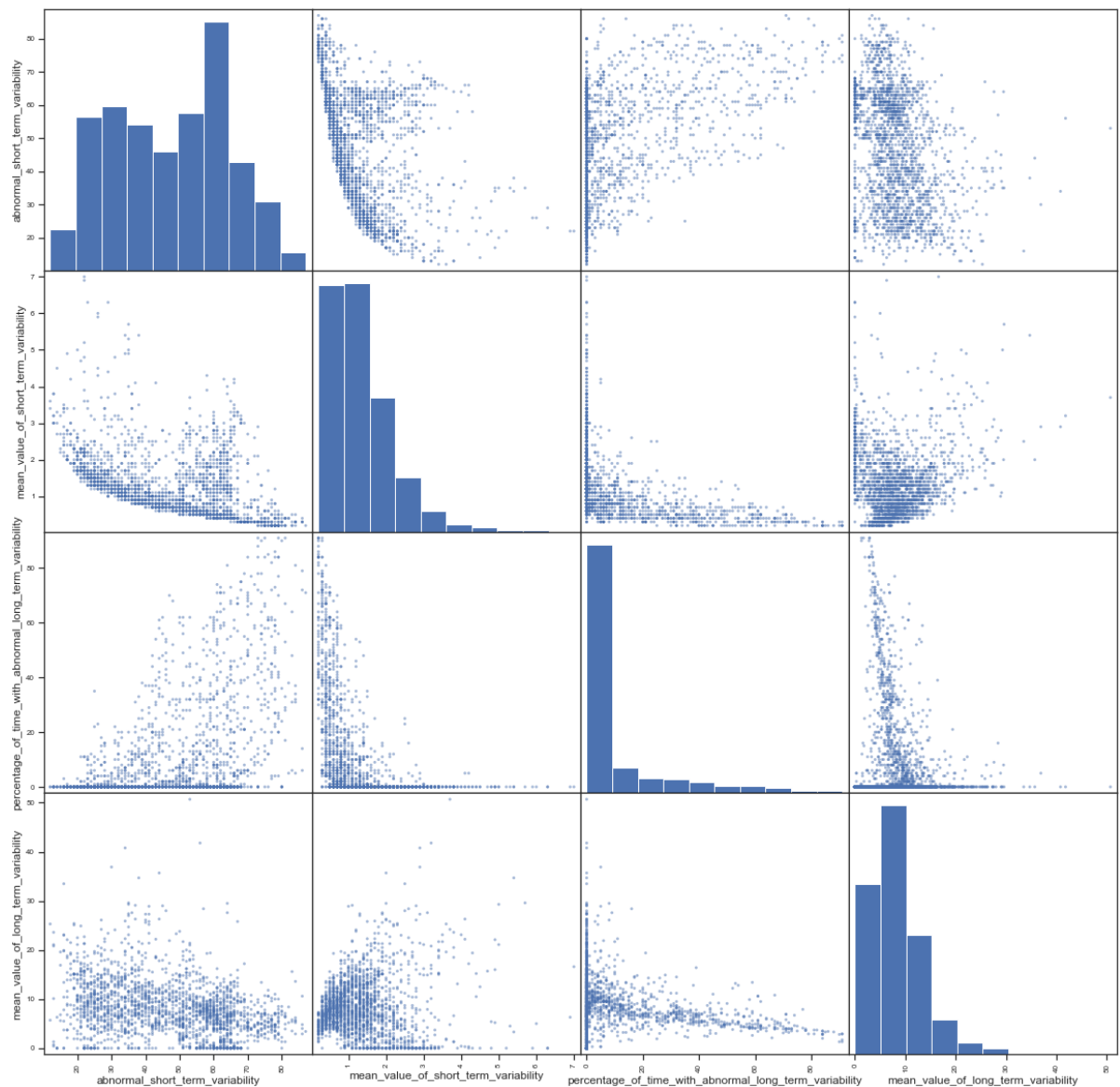
In [24]: *#Multi variable Box Plot*
In the following plot we have tried to show the range of our feature attributes by making a multi variable box plot.
We can see that all the features are in different numeric ranges so for further modeling we have to normalize/ scale
each value in a similar range.

```
plt.figure(figsize=(20,10))
sns.boxenplot(data = fetal)
plt.xticks(rotation=90)
plt.title('Mutlivariable Box Plot')
plt.show()
```

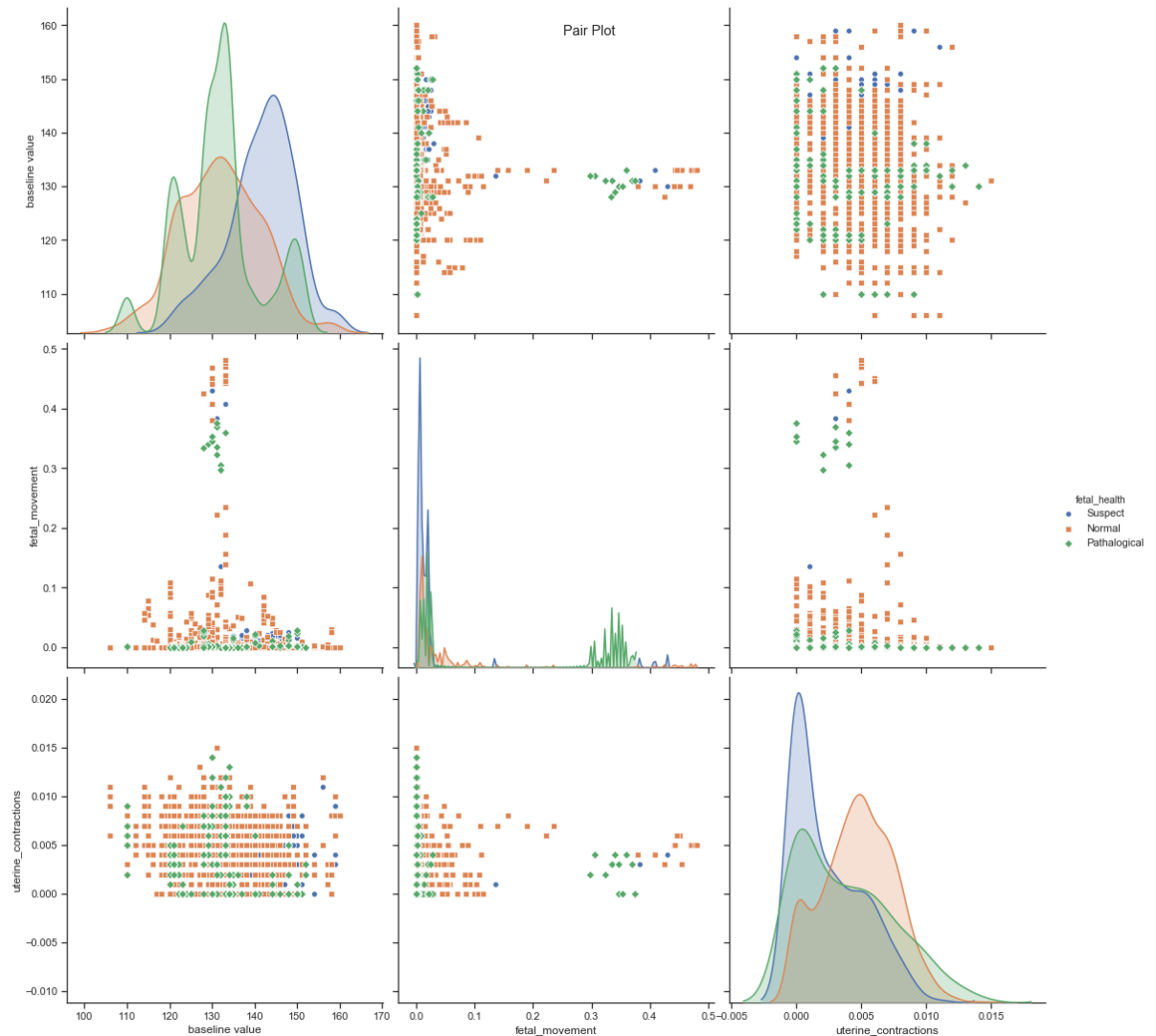


```
In [25]: # We have plotted 4 variables amongst each other in a scatter matrix giving us
12 plots and 4 histograms.
from pandas.plotting import scatter_matrix
scatter_matrix(fetal[['abnormal_short_term_variability', 'mean_value_of_short_t
erm_variability'
                    , 'percentage_of_time_with_abnormal_long_term_variabilit
y',
                    'mean_value_of_long_term_variability']],figsize=(20,20),
diagonal='hist')
plt.suptitle('Scatter Matrix of types of variability')
plt.show()
```

Scatter Matrix of types of variability



```
In [26]: # We have plotted 3 variables amongst each other in a pair plots with a color
#         code according to fetal health
#         giving us 6 plots and 3 kernel density estimate plots.
dataforplot = fetal[['baseline value', 'fetal_movement', 'uterine_contractions',
                    'fetal_health']]
sns.pairplot(dataforplot, kind="scatter", hue = 'fetal_health', height = 5, aspect
            = 1, markers=["o", "s", "D"])
plt.suptitle('Pair Plot')
plt.show()
```



```

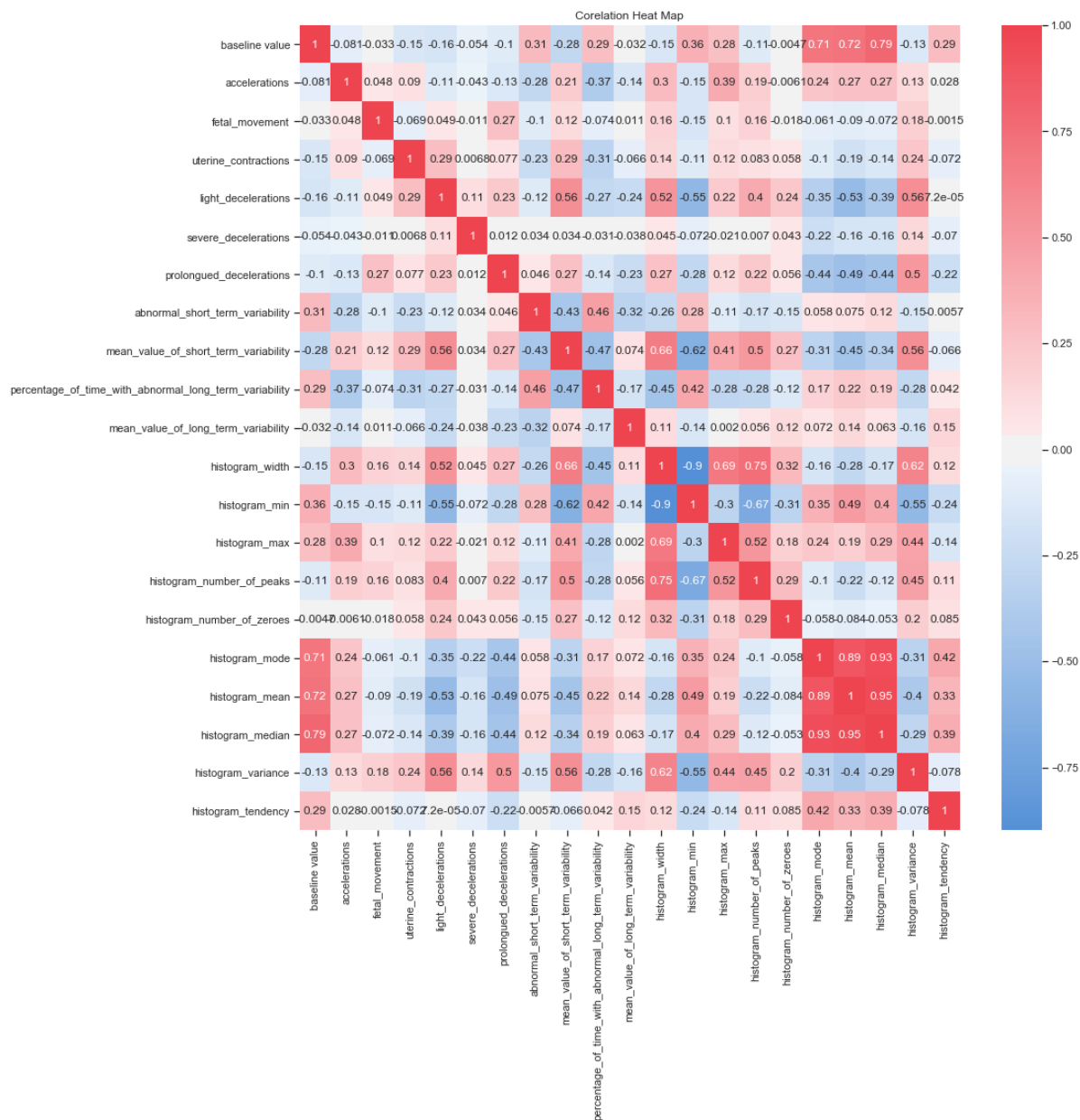
In [27]: #Heat Map
# By looking at the heat map we can clearly say that 'accelerations','prolongu
ed_decelerations',
#'abnormal_short_term_variability','percentage_of_time_with_abnormal_long_term
_variability',
#'mean_value_of_long_term_variability' these features are highly correlated wi
th fetal health
corrmat= fetal.corr()
plt.figure(figsize=(15,15))

cmap = sns.diverging_palette(250, 10, s=80, l=55, n=9, as_cmap=True)

sns.heatmap(corrmat,annot=True, cmap=cmap, center=0)
plt.title('Corelation Heat Map')

```

Out[27]: Text(0.5, 1, 'Corelation Heat Map')




```
In [28]: #References:
# 1. Kaggle.com. 2020. Fetal Health Classification. [online] Available at:
# <https://www.kaggle.com/andrewmvd/fetal-health-classification> [Accessed 10
# April 2021].
# 2. Matplotlib.org. n.d. Matplotlib: Python plotting – Matplotlib 3.4.1 docum
# entation. [online] Available at:
# <https://matplotlib.org/> [Accessed 10 April 2021].
# 3. Seaborn.pydata.org. n.d. seaborn: statistical data visualization – seabor
# n 0.11.1 documentation. [online] Available at:
# <https://seaborn.pydata.org/index.html> [Accessed 10 April 2021]
# 4. Pandas.pydata.org. n.d. pandas - Python Data Analysis Library. [online] A
# vailable at:
# <https://pandas.pydata.org/> [Accessed 10 April 2021].
```

-----EndOfCode-----