In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
df=pd.read_csv('train.csv')
df.head()
```

Out[4]:

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_tern
0	142.0	0.000	0.000	0.007	0.000	0.0	0.0	
1	122.0	0.000	0.000	0.006	0.002	0.0	0.0	
2	129.0	0.005	0.003	0.001	0.000	0.0	0.0	
3	136.0	0.006	0.000	0.008	0.000	0.0	0.0	
4	144.0	0.000	0.000	0.006	0.000	0.0	0.0	

5 rows × 22 columns

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1700 entries, 0 to 1699
Data columns (total 22 columns):
 #   Column                                                  Non-Null Count  Dtype  
---  ------                                                  --------------  -----  
 0   baseline value                                          1700 non-null   float64
 1   accelerations                                           1700 non-null   float64
 2   fetal_movement                                          1700 non-null   float64
 3   uterine_contractions                                    1700 non-null   float64
 4   light_decelerations                                     1700 non-null   float64
 5   severe_decelerations                                    1700 non-null   float64
 6   prolongued_decelerations                                1700 non-null   float64
 7   abnormal_short_term_variability                         1700 non-null   float64
 8   mean_value_of_short_term_variability                    1700 non-null   float64
 9   percentage_of_time_with_abnormal_long_term_variability  1700 non-null   float64
 10  mean_value_of_long_term_variability                     1700 non-null   float64
 11  histogram_width                                         1700 non-null   float64
 12  histogram_min                                           1700 non-null   float64
 13  histogram_max                                           1700 non-null   float64
 14  histogram_number_of_peaks                               1700 non-null   float64
 15  histogram_number_of_zeroes                              1700 non-null   float64
 16  histogram_mode                                          1700 non-null   float64
 17  histogram_mean                                          1700 non-null   float64
 18  histogram_median                                        1700 non-null   float64
 19  histogram_variance                                      1700 non-null   float64
 20  histogram_tendency                                      1700 non-null   float64
 21  fetal_health                                            1700 non-null   float64
dtypes: float64(22)
memory usage: 292.3 KB
```

In [6]:
```
df.shape
```

Out[6]:
```
(1700, 22)
```

In [7]:
```
df.isna().sum()
```

Out[7]:

```
baseline value                                            0
accelerations                                             0
fetal_movement                                            0
uterine_contractions                                      0
light_decelerations                                       0
severe_decelerations                                      0
prolongued_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 [8]:

```
df.describe()
```

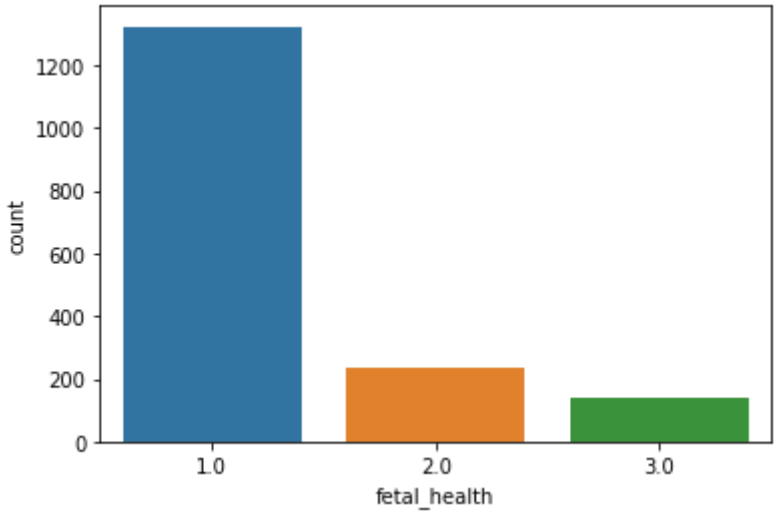
Out[8]:

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_sh
count	1700.000000	1700.000000	1700.000000	1700.000000	1700.000000	1700.000000	1700.000000	
mean	133.213529	0.003212	0.010211	0.004356	0.001899	0.000004	0.000158	
std	9.873344	0.003888	0.050124	0.002943	0.002976	0.000059	0.000587	
min	106.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	126.000000	0.000000	0.000000	0.002000	0.000000	0.000000	0.000000	
50%	133.000000	0.002000	0.000000	0.004000	0.000000	0.000000	0.000000	
75%	140.000000	0.006000	0.003000	0.006000	0.003000	0.000000	0.000000	
max	159.000000	0.019000	0.481000	0.015000	0.015000	0.001000	0.005000	

8 rows × 22 columns

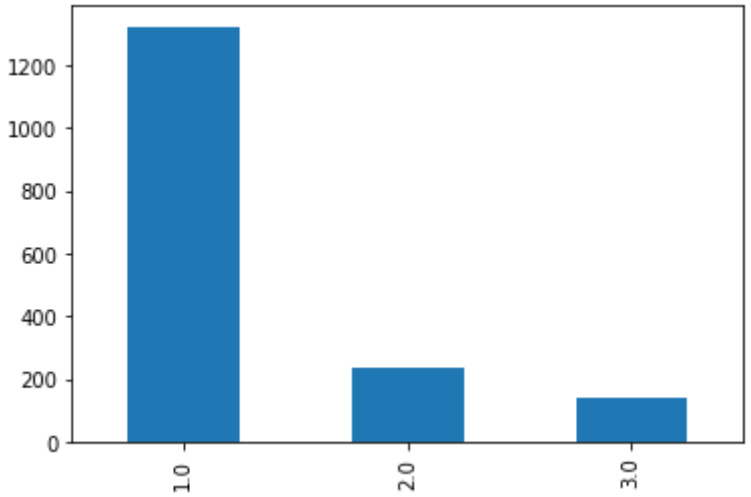
In [9]:
```
#To understand the different types in our target variable (fatal_health)
sns.countplot(df.fetal_health)
```

Out[9]:
```
<AxesSubplot:xlabel='fetal_health', ylabel='count'>
```



In [10]:
```
df['fetal_health'].value_counts().plot(kind='bar')
```

Out[10]:
```
<AxesSubplot:>
```

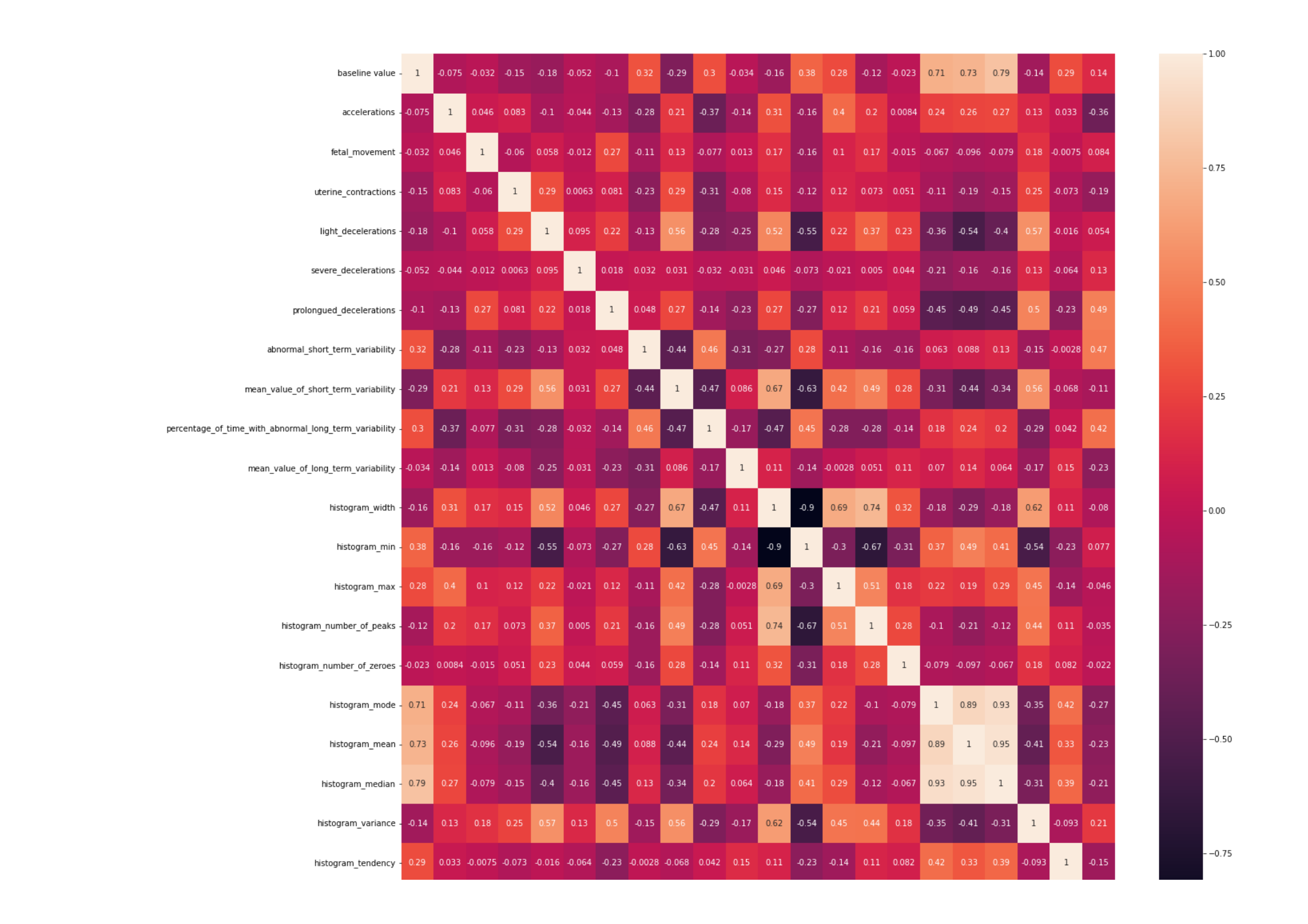


In [12]:
```
#to find all datatypes in our file
cats = list(df.select_dtypes(include=['object','bool']) )
nums = list(df.select_dtypes(include=['int64','float64']))
print(cats)
print(nums)
[]
['baseline value', 'accelerations', 'fetal_movement',
'uterine_contractions', 'light_decelerations', 'severe_decelerations',
'prolongued_decelerations', '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',
'fetal_health']
#splitting the dataset X,y
X=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=21)
```

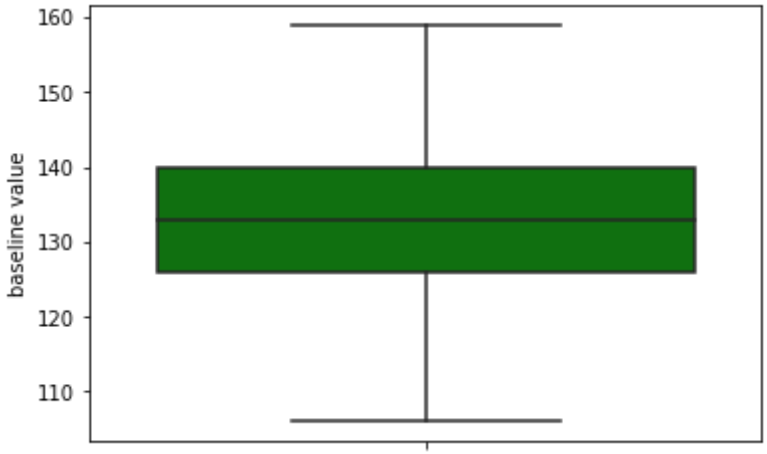
```
[]
['baseline value', 'accelerations', 'fetal_movement', 'uterine_contractions', 'light_decelerations', 'severe_decelerations', 'pr
olongued_decelerations', 'abnormal_short_term_variability', 'mean_value_of_short_term_variability', 'percentage_of_time_with_abn
ormal_long_term_variability', 'mean_value_of_long_term_variability', 'histogram_width', 'histogram_min', 'histogram_max', 'histo
gram_number_of_peaks', 'histogram_number_of_zeroes', 'histogram_mode', 'histogram_mean', 'histogram_median', 'histogram_varianc
e', 'histogram_tendency', 'fetal_health']
```

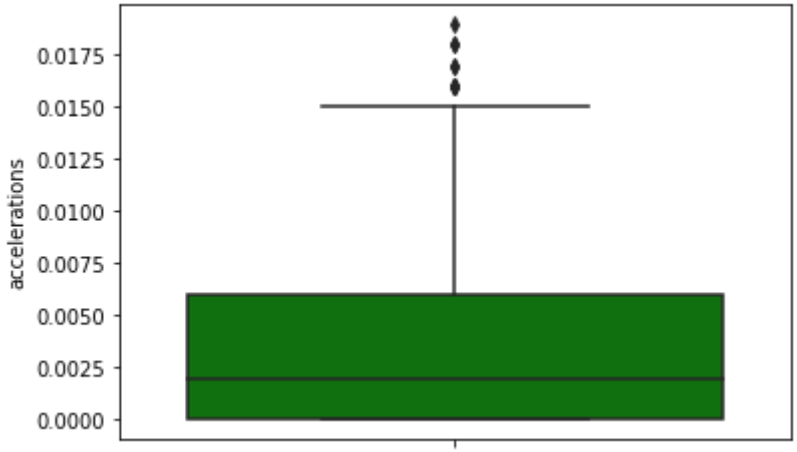
In [13]:
```
#Finding the correlation of df
corr=df.corr()
plt.figure(figsize=(20,20))
sns.heatmap(corr,annot=True)
```

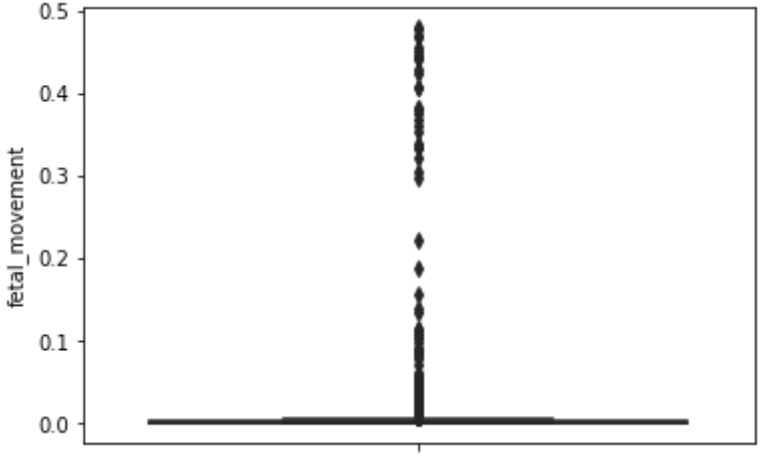
Out[13]:
```
<AxesSubplot:>
```

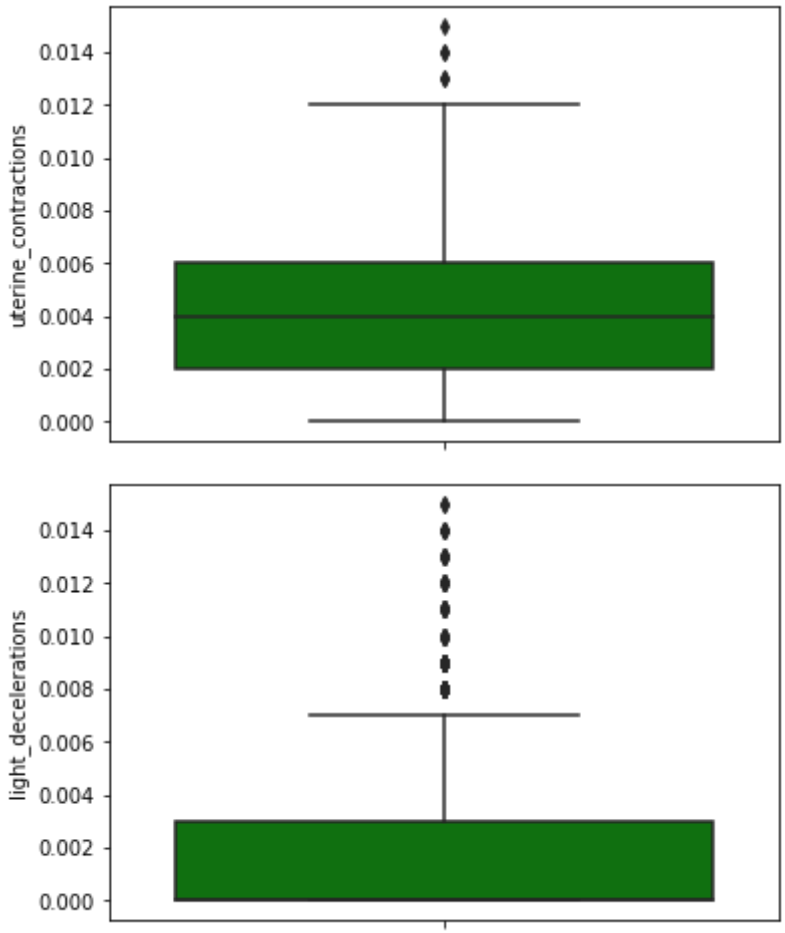

baseline value 1 -0.075 -0.032 -0.15 -0.18 -0.052 -0.1 0.32 -0.29 0.3 -0.034 -0.16 0.38 0.28 -0.12 -0.023 0.71 0.73 0.79 -0.14 0.29 0.14
accelerations -0.075 1 0.046 0.083 -0.1 -0.044 -0.13 -0.28 0.21 -0.37 -0.14 0.31 -0.16 0.4 0.2 0.0084 0.24 0.26 0.27 0.13 0.033 -0.36
fetal_movement -0.032 0.046 1 -0.06 0.058 -0.012 0.27 -0.11 0.13 -0.077 0.013 0.17 -0.16 0.1 0.17 -0.015 -0.067 -0.096 -0.079 0.18 -0.0075 0.084
uterine_contractions -0.15 0.083 -0.06 1 0.29 0.0063 0.081 -0.23 0.29 -0.31 -0.08 0.15 -0.12 0.12 0.073 0.051 -0.11 -0.19 -0.15 0.25 -0.073 -0.19
light_decelerations -0.18 -0.1 0.058 0.29 1 0.095 0.22 -0.13 0.56 -0.28 -0.25 0.52 -0.55 0.22 0.37 0.23 -0.36 -0.54 -0.4 0.57 -0.016 0.054
severe_decelerations -0.052 -0.044 -0.012 0.0063 0.095 1 0.018 0.032 0.031 -0.032 -0.031 0.046 -0.073 -0.021 0.005 0.044 -0.21 -0.16 -0.16 0.13 -0.064 0.13
prolongued_decelerations -0.1 -0.13 0.27 0.081 0.22 0.018 1 0.048 0.27 -0.14 -0.23 0.27 -0.27 0.12 0.21 0.059 -0.45 -0.49 -0.45 0.5 -0.23 0.49
abnormal_short_term_variability 0.32 -0.28 -0.11 -0.23 -0.13 0.032 0.048 1 -0.44 0.46 -0.31 -0.27 0.28 -0.11 -0.16 -0.16 0.063 0.088 0.13 -0.15 -0.0028 0.47
mean_value_of_short_term_variability -0.29 0.21 0.13 0.29 0.56 0.031 0.27 -0.44 1 -0.47 0.086 0.67 -0.63 0.42 0.49 0.28 -0.31 -0.44 -0.34 0.56 -0.068 -0.11
percentage_of_time_with_abnormal_long_term_variability 0.3 -0.37 -0.077 -0.31 -0.28 -0.032 -0.14 0.46 -0.47 1 -0.17 -0.47 0.45 -0.28 -0.28 -0.14 0.18 0.24 0.2 -0.29 0.042 0.42
mean_value_of_long_term_variability -0.034 -0.14 0.013 -0.08 -0.25 -0.031 -0.23 -0.31 0.086 -0.17 1 0.11 -0.14 -0.0028 0.051 0.11 0.07 0.14 0.064 -0.17 0.15 -0.23
histogram_width -0.16 0.31 0.17 0.15 0.52 0.046 0.27 -0.27 0.67 -0.47 0.11 1 -0.9 0.69 0.74 0.32 -0.18 -0.29 -0.18 0.62 0.11 -0.08
histogram_min 0.38 -0.16 -0.16 -0.12 -0.55 -0.073 -0.27 0.28 -0.63 0.45 -0.14 -0.9 1 -0.3 -0.67 -0.31 0.37 0.49 0.41 -0.54 -0.23 0.077
histogram_max 0.28 0.4 0.1 0.12 0.22 -0.021 0.12 -0.11 0.42 -0.28 -0.0028 0.69 -0.3 1 0.51 0.18 0.22 0.19 0.29 0.45 -0.14 -0.046
histogram_number_of_peaks -0.12 0.2 0.17 0.073 0.37 0.005 0.21 -0.16 0.49 -0.28 0.051 0.74 -0.67 0.51 1 0.28 -0.1 -0.21 -0.12 0.44 0.11 -0.035
histogram_number_of_zeroes -0.023 0.0084 -0.015 0.051 0.23 0.044 0.059 -0.16 0.28 -0.14 0.11 0.32 -0.31 0.18 0.28 1 -0.079 -0.097 -0.067 0.18 0.082 -0.022
histogram_mode 0.71 0.24 -0.067 -0.11 -0.36 -0.21 -0.45 0.063 -0.31 0.18 0.07 -0.18 0.37 0.22 -0.1 -0.079 1 0.89 0.93 -0.35 0.42 -0.27
histogram_mean 0.73 0.26 -0.096 -0.19 -0.54 -0.16 -0.49 0.088 -0.44 0.24 0.14 -0.29 0.49 0.19 -0.21 -0.097 0.89 1 0.95 -0.41 0.33 -0.23
histogram_median 0.79 0.27 -0.079 -0.15 -0.4 -0.16 -0.45 0.13 -0.34 0.2 0.064 -0.18 0.41 0.29 -0.12 -0.067 0.93 0.95 1 -0.31 0.39 -0.21
histogram_variance -0.14 0.13 0.18 0.25 0.57 0.13 0.5 -0.15 0.56 -0.29 -0.17 0.62 -0.54 0.45 0.44 0.18 -0.35 -0.41 -0.31 1 -0.093 0.21
histogram_tendency 0.29 0.033 -0.0075 -0.073 -0.016 -0.064 -0.23 -0.0028 -0.068 0.042 0.15 0.11 -0.23 -0.14 0.11 0.082 0.42 0.33 0.39 -0.093 1 -0.15
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

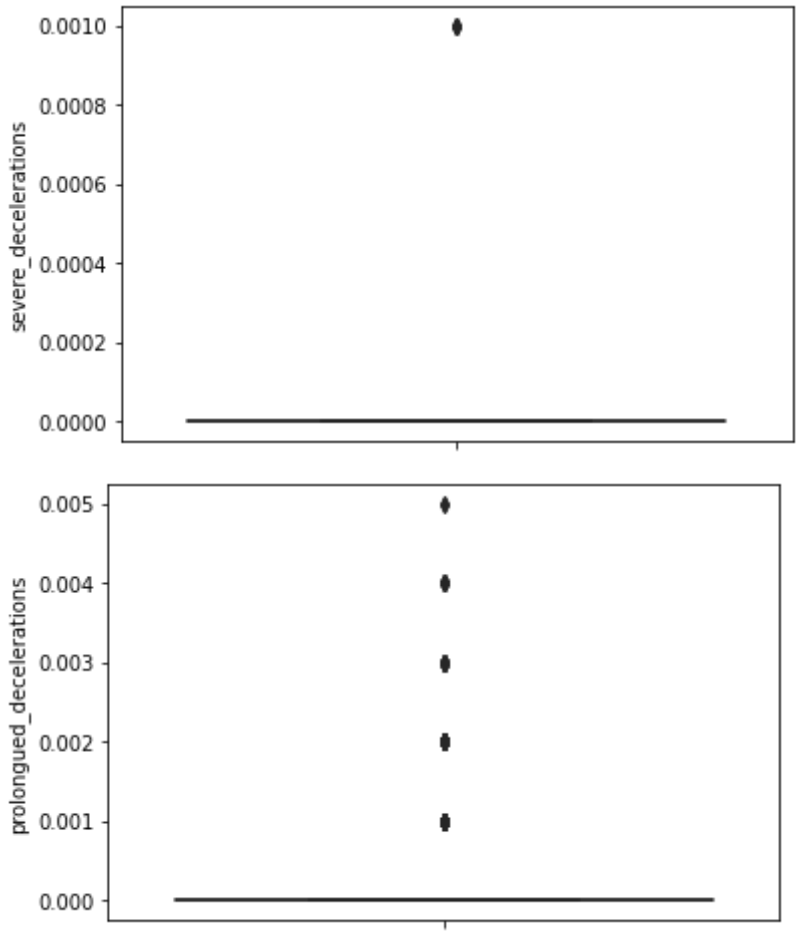
In [14]:
```
#Finding the outlier using boxplot
for i in range(0, len(nums )):
 sns.boxplot(y=df[nums[i]],color='green',orient='v')
 plt.show()
```

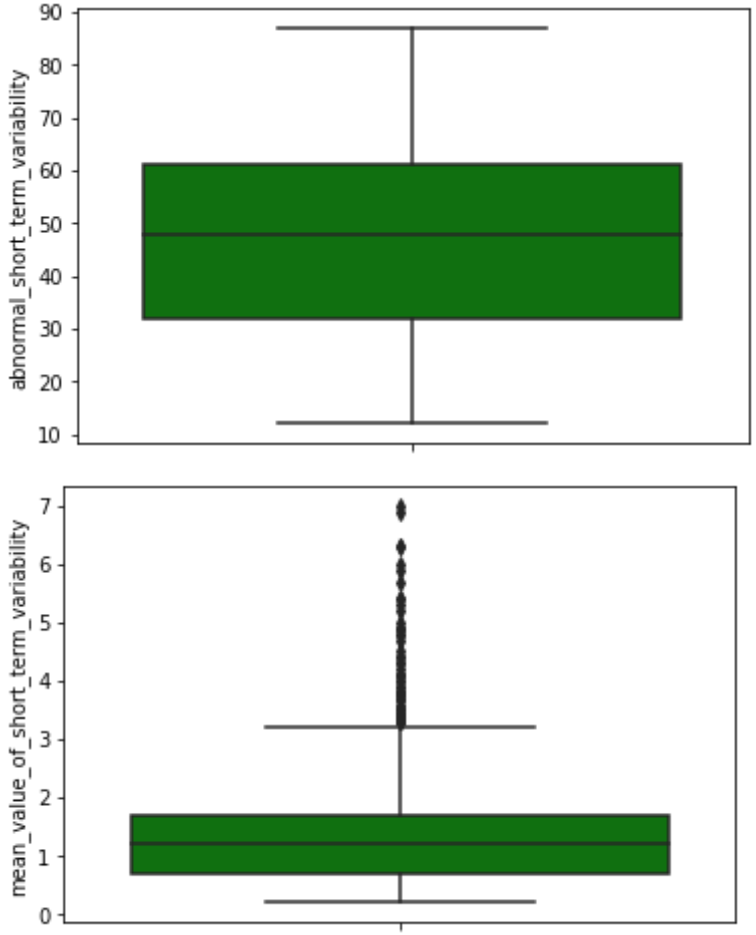


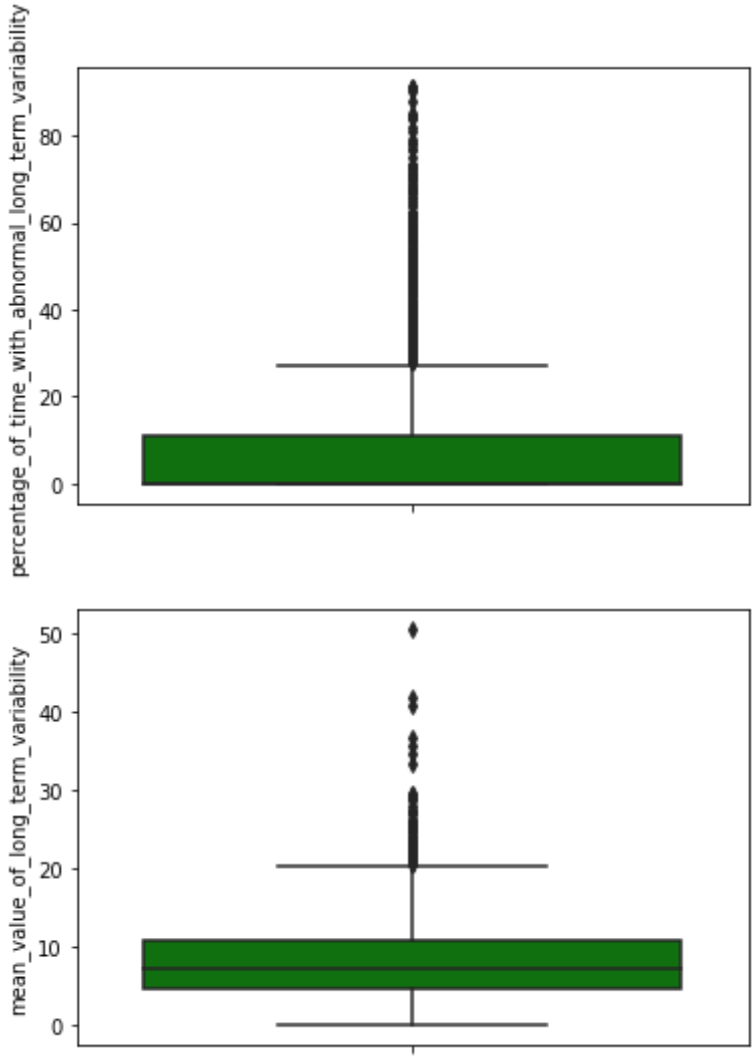

accelerations
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000

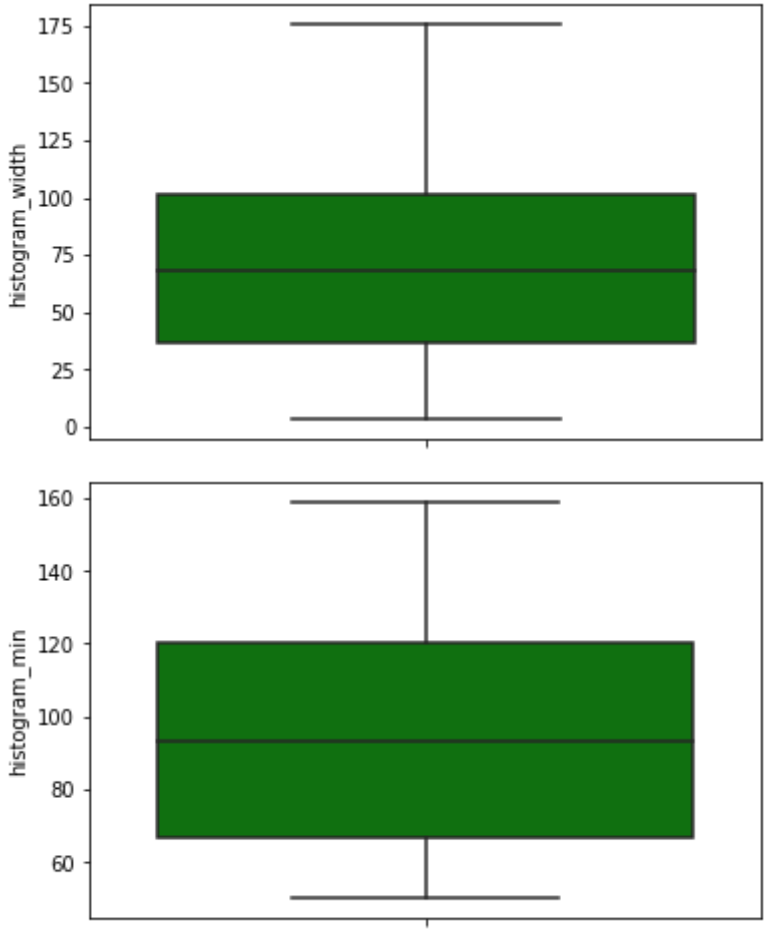

fetal_movement
0.5
0.4
0.3
0.2
0.1
0.0

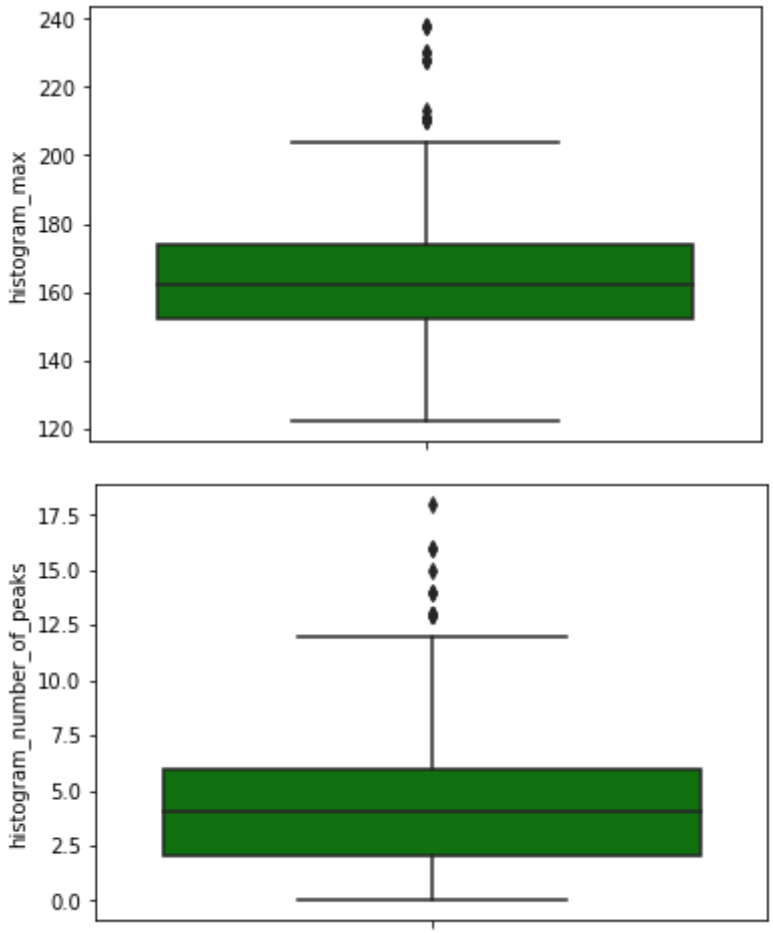

uterine_contractions
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
light_decelerations
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000

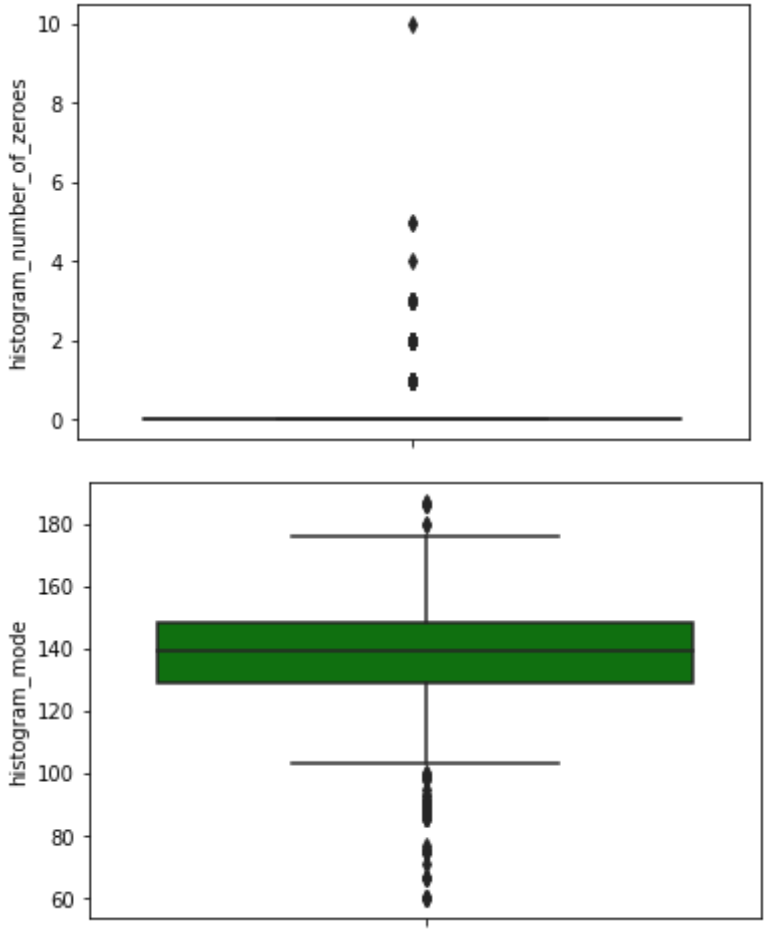

severe_decelerations
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
prolongued_decelerations
0.005
0.004
0.003
0.002
0.001
0.000

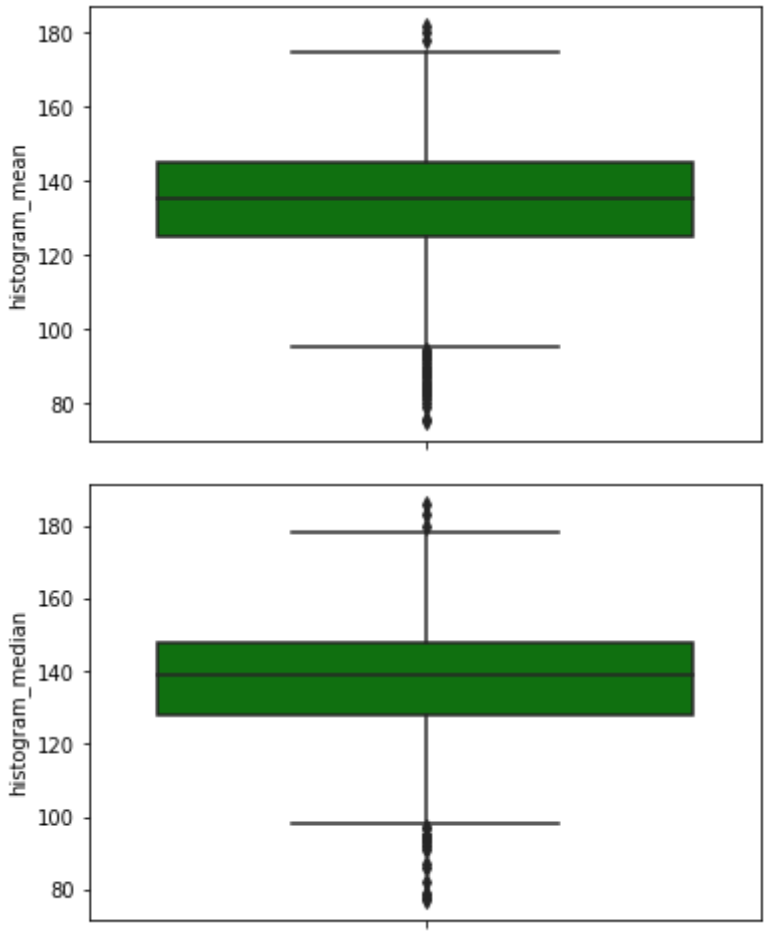

abnormal_short_term_variability
90
80
70
60
50
40
30
20
10
mean_value_of_short_term_variability
7
6
5
4
3
2
1
0

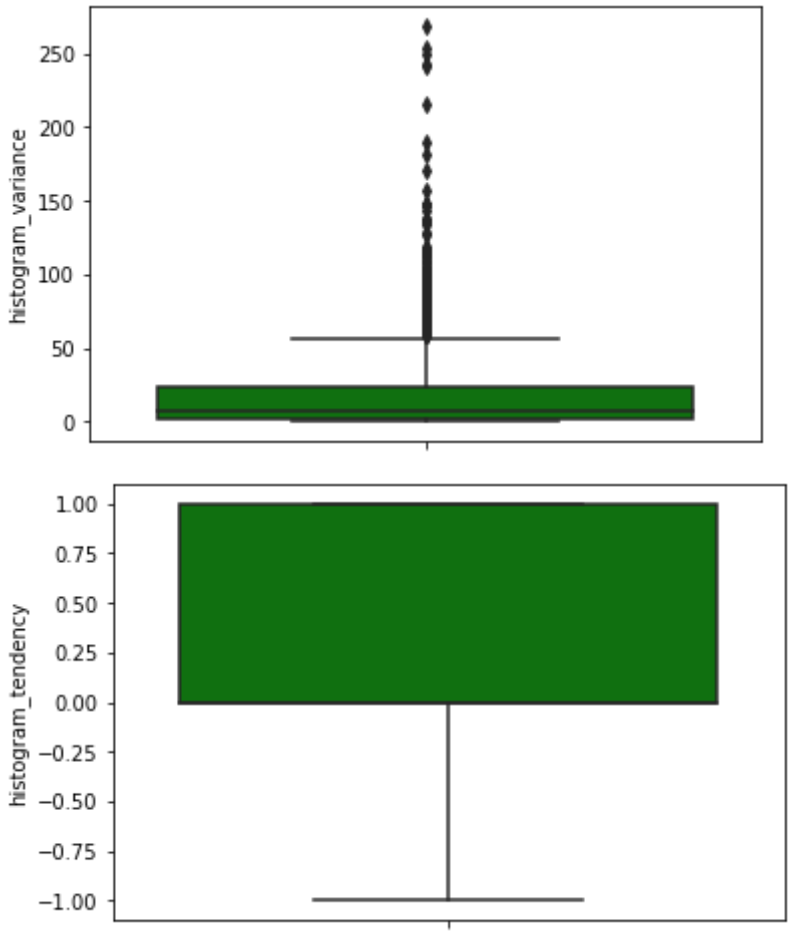

percentage_of_time_with_abnormal_long_term_variability
0
20
40
60
80
mean_value_of_long_term_variability
0
10
20
30
40
50

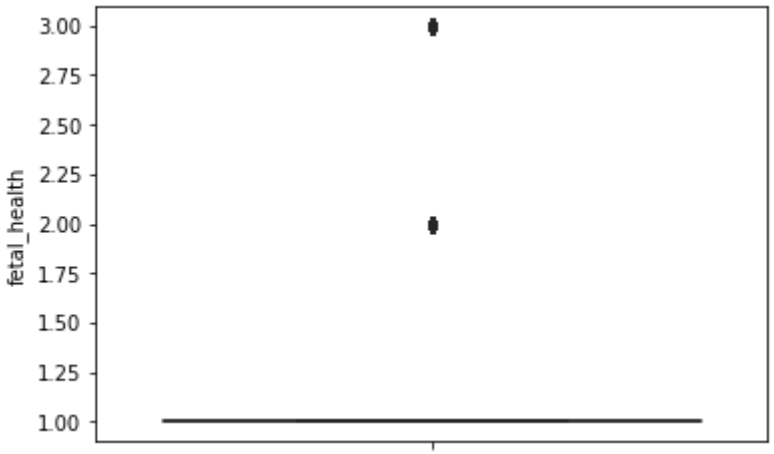

histogram_width
175
150
125
100
75
50
25
0
histogram_min
160
140
120
100
80
60


histogram_max
240
220
200
180
160
140
120
histogram_number_of_peaks
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0

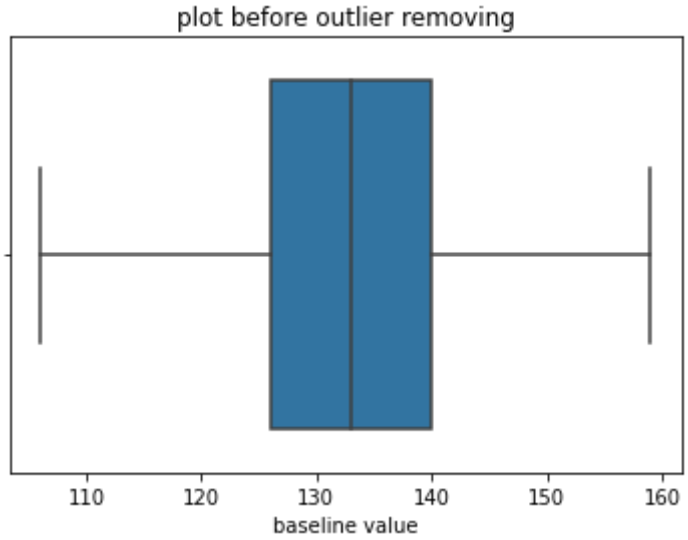

histogram_number_of_zeroes
10
8
6
4
2
0
histogram_mode
180
160
140
120
100
80
60

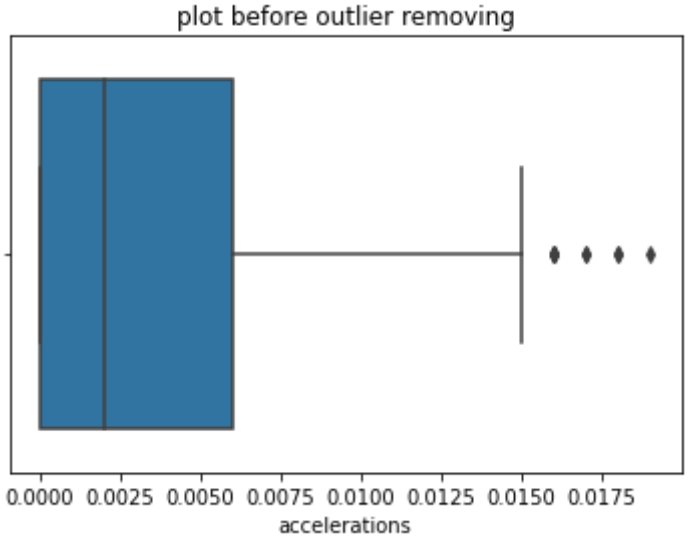

histogram_mean
180
160
140
120
100
80
histogram_median
180
160
140
120
100
80

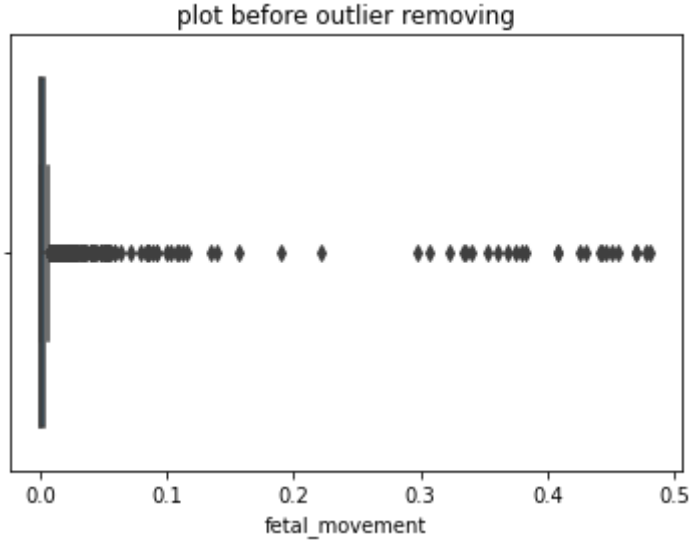

histogram_variance
250
200
150
100
50
0
histogram_tendency
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

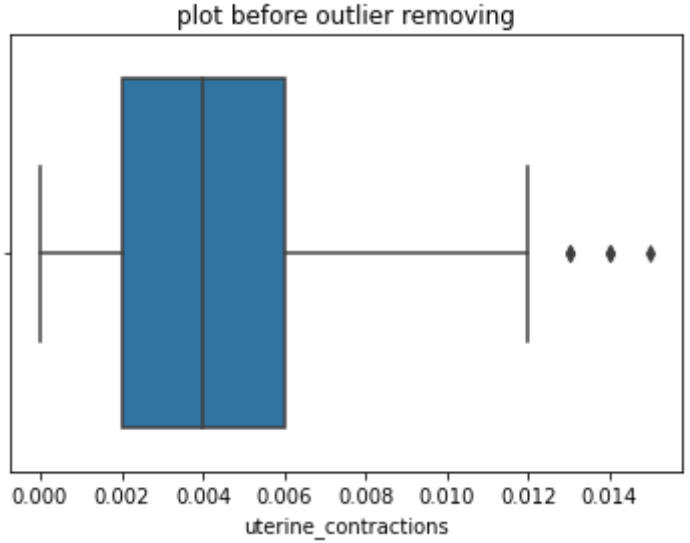


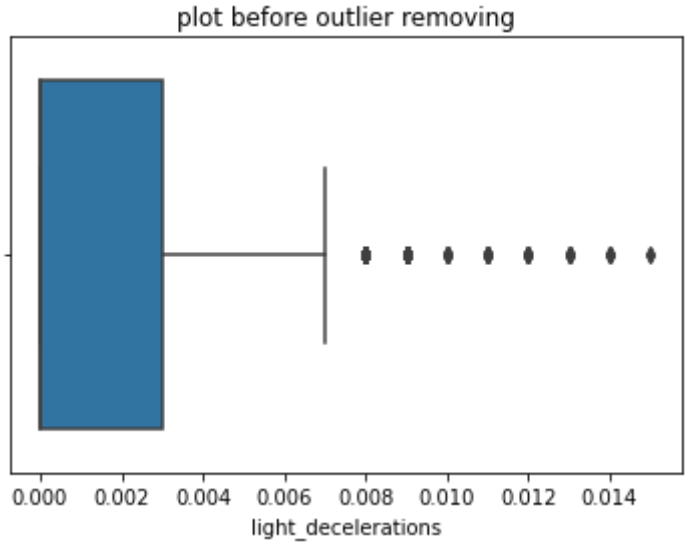
In [15]:
```
#Removing the outliers
for i in range(len(nums)):
    sns.boxplot(df[nums[i]])
    plt.title(nums[i])
    plt.title("plot before outlier removing")
    plt.show()
    def drop_outliers(df, field_name):
        iqr = 1.5 * (np.percentile(df[field_name], 75) - np.percentile(df[field_name], 25))
        df.drop(df[df[field_name] > (iqr + np.percentile(df[field_name], 75))].index, inplace=True)
        df.drop(df[df[field_name] < (np.percentile(df[field_name], 25)- iqr)].index, inplace=True)
        iqr = 1.5 * (np.percentile(df[field_name], 75) - np.percentile(df[field_name], 25))
        df.drop(df[df[field_name] > (iqr + np.percentile(df[field_name], 75))].index, inplace=True)
        df.drop(df[df[field_name] < (np.percentile(df[field_name], 25)- iqr)].index, inplace=True)
        drop_outliers(df, nums[i])
        sns.boxplot(df[nums[i]])
        plt.title("plot after outlier removing")
        plt.show()
```

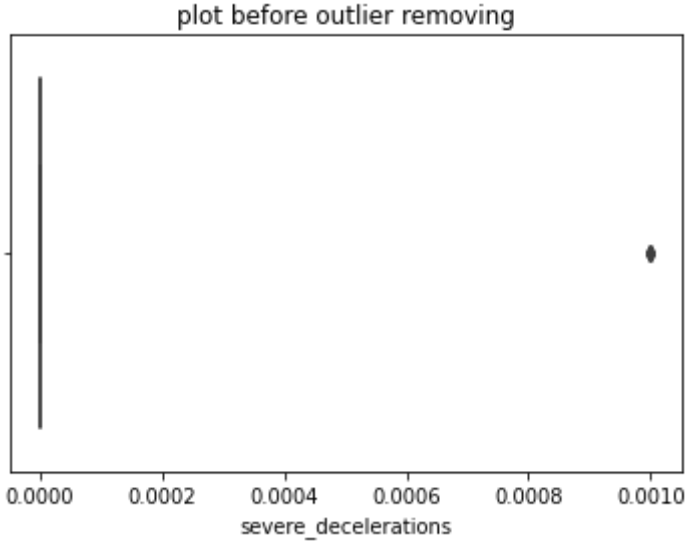

plot before outlier removing
110
120
130
140
150
160
baseline value

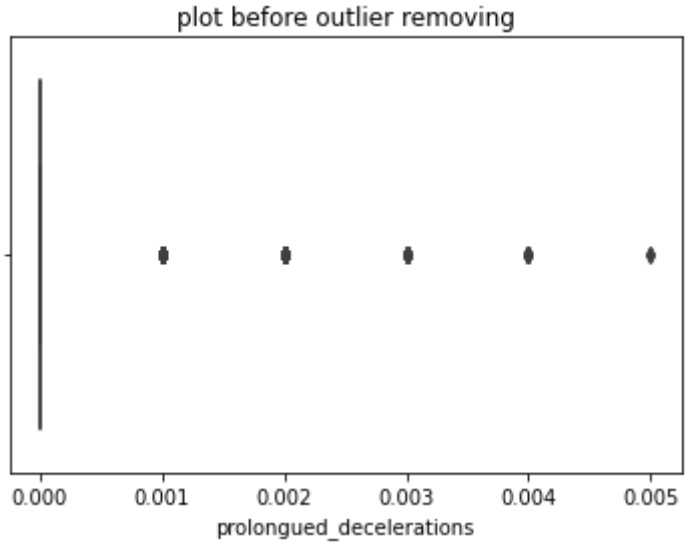

plot before outlier removing
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
accelerations

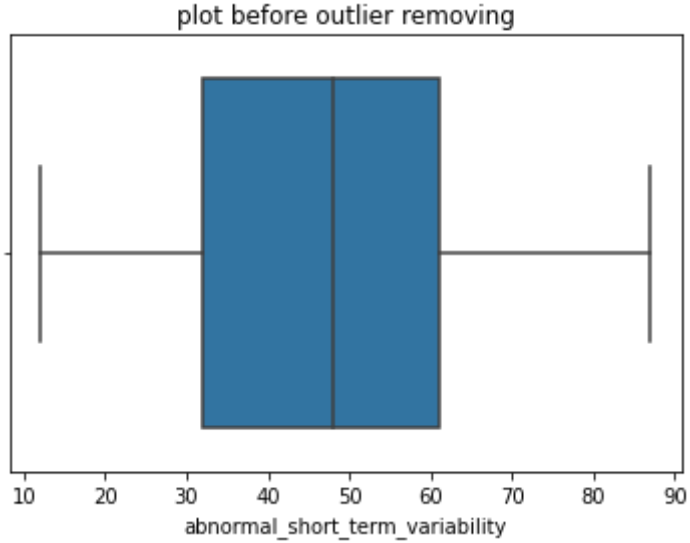

plot before outlier removing
0.0
0.1
0.2
0.3
0.4
0.5
fetal_movement

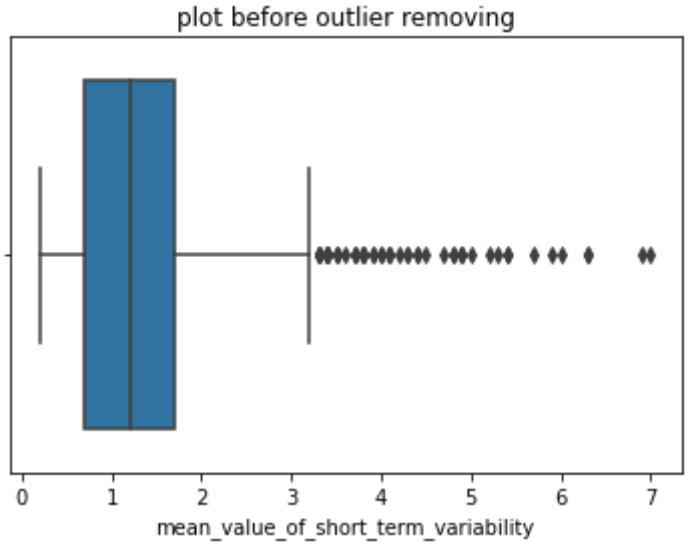

plot before outlier removing
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
uterine_contractions

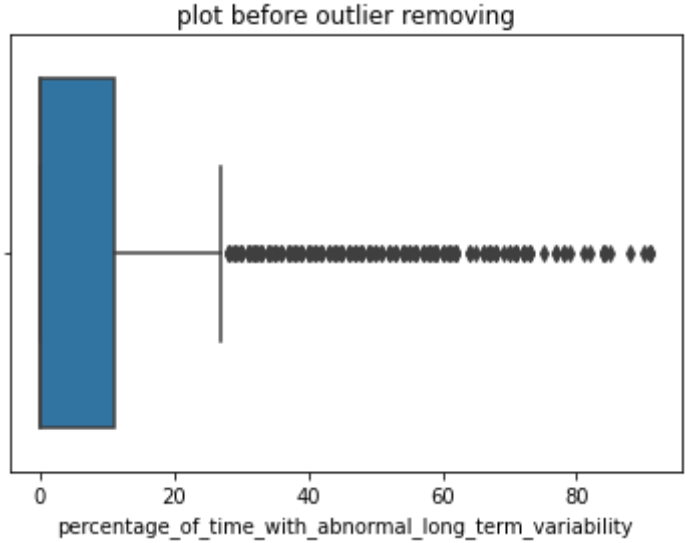

plot before outlier removing
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
light_decelerations

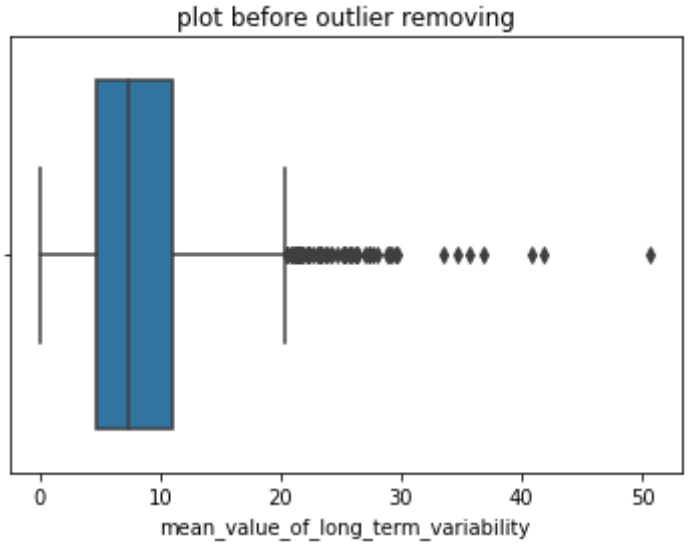

plot before outlier removing
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
severe_decelerations

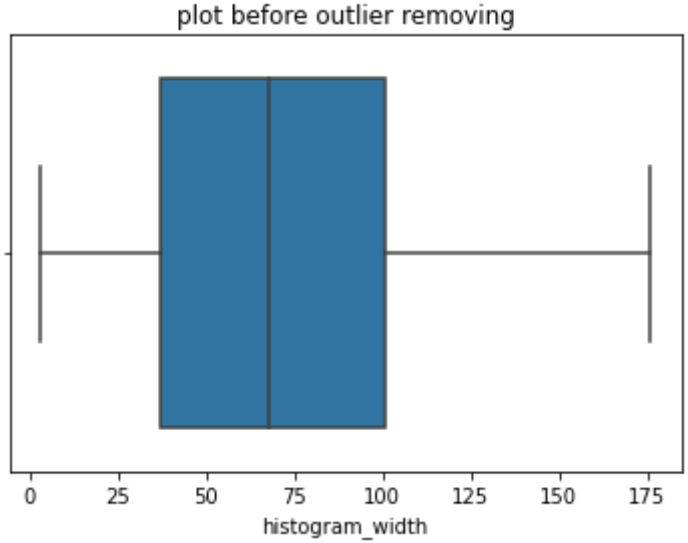

plot before outlier removing
0.000
0.001
0.002
0.003
0.004
0.005
prolongued_decelerations

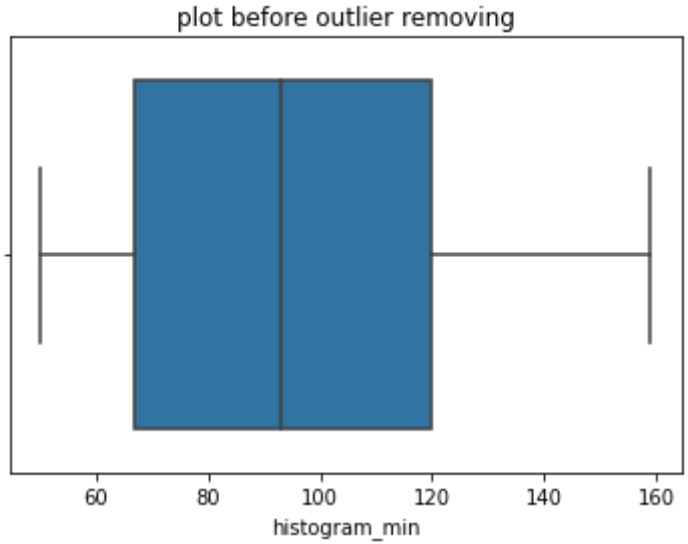

plot before outlier removing
10
20
30
40
50
60
70
80
90
abnormal_short_term_variability

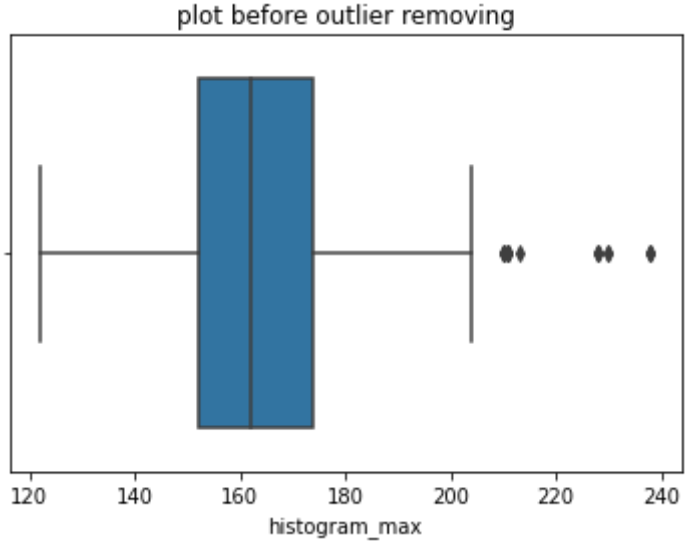

plot before outlier removing
0
1
2
3
4
5
6
7
mean_value_of_short_term_variability

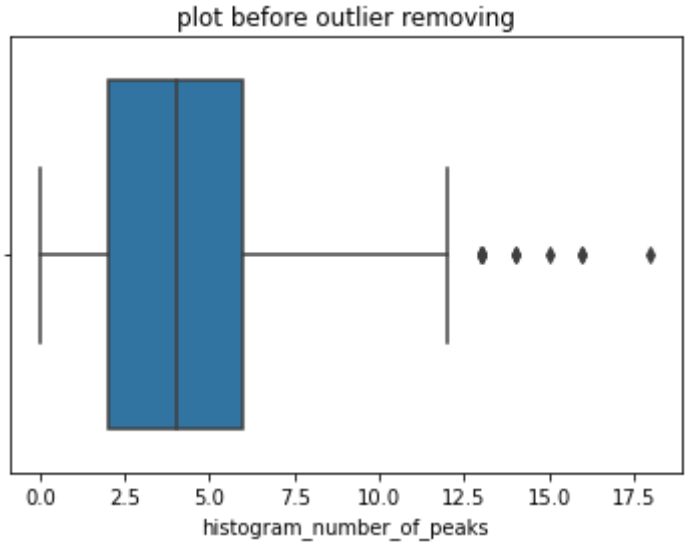

plot before outlier removing
0
20
40
60
80
percentage_of_time_with_abnormal_long_term_variability

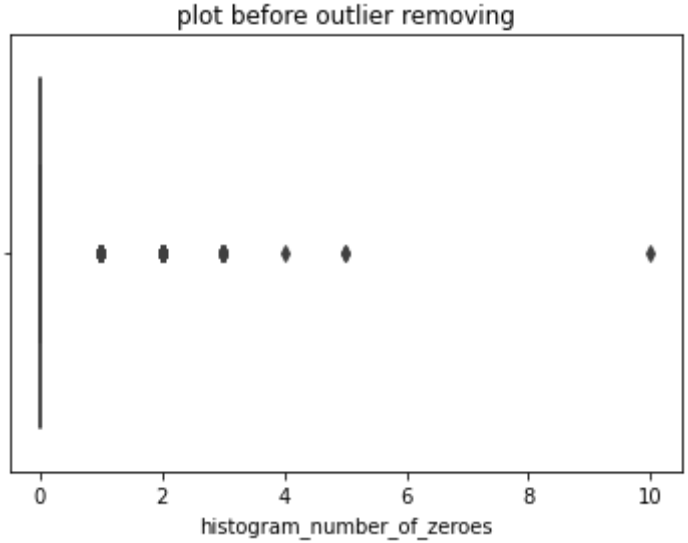

plot before outlier removing
0
10
20
30
40
50
mean_value_of_long_term_variability

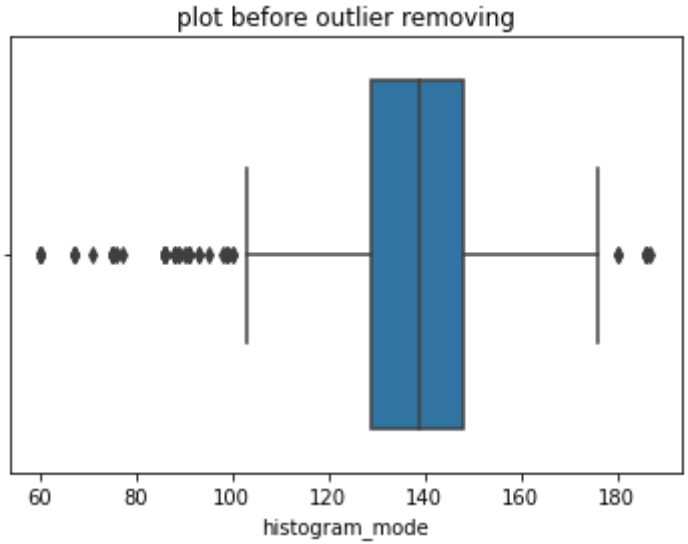

plot before outlier removing
0
25
50
75
100
125
150
175
histogram_width

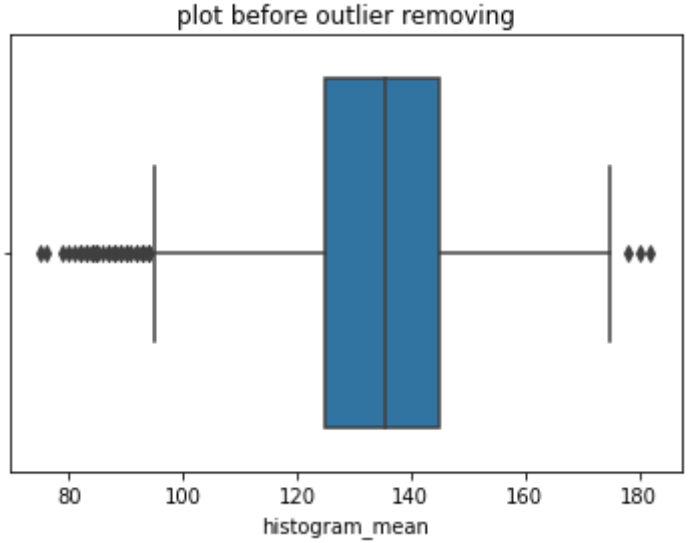

plot before outlier removing
60
80
100
120
140
160
histogram_min

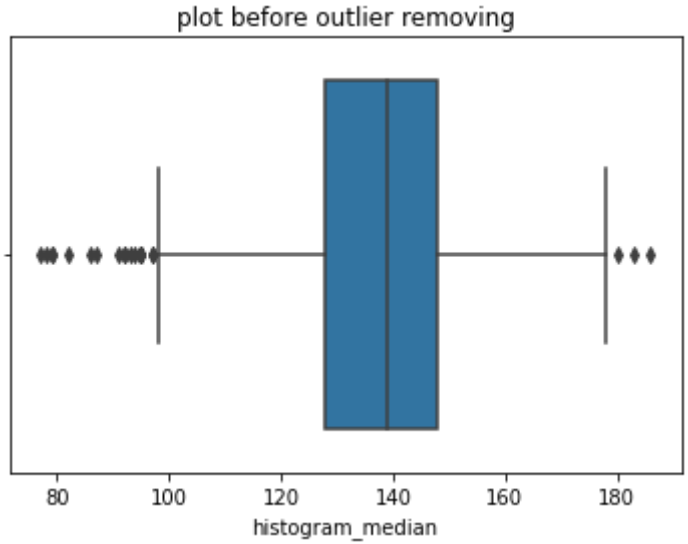

plot before outlier removing
120
140
160
180
200
220
240
histogram_max

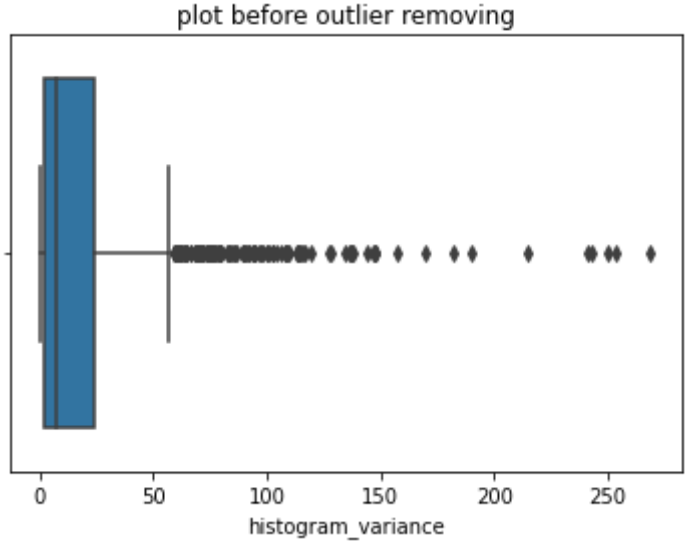

plot before outlier removing
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
histogram_number_of_peaks

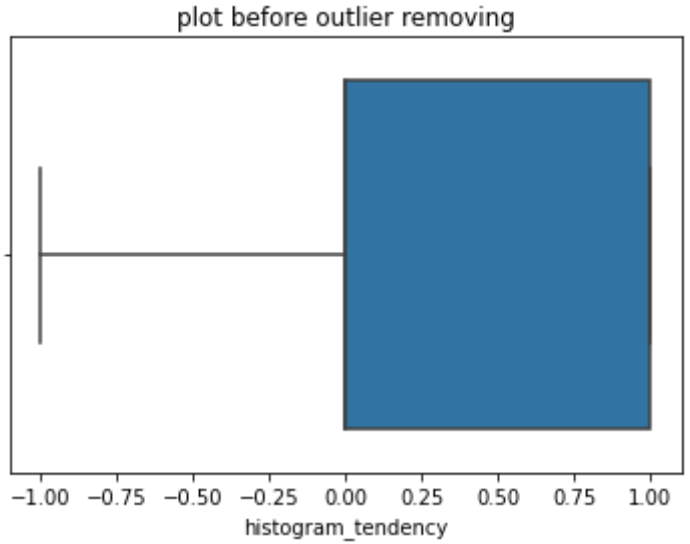

plot before outlier removing
0
2
4
6
8
10
histogram_number_of_zeroes

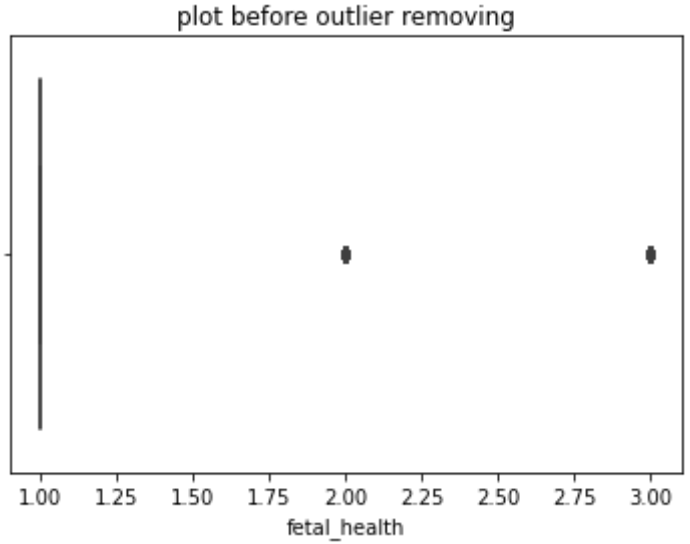

plot before outlier removing
60
80
100
120
140
160
180
histogram_mode


plot before outlier removing
80
100
120
140
160
180
histogram_mean


plot before outlier removing
80
100
120
140
160
180
histogram_median


plot before outlier removing
0
50
100
150
200
250
histogram_variance





In [16]:
```
#Feature scaling
from sklearn.preprocessing import StandardScaler
SC=StandardScaler()
X_train=SC.fit_transform(X_train)
X_test=SC.fit_transform(X_test)
print(X_train)
```

```
[[-1.46513509 -0.83485626 -0.20820521 ... -1.18281773 -0.60623421
  -2.14821249]
 [ 0.16393359 -0.83485626 -0.0746156  ... -0.08025974 -0.53911612
   1.13129086]
 [-1.2615015   2.2251405  -0.00782079 ... -0.42480911 -0.37132092
  -2.14821249]
 ...
 [ 1.28391831  1.71514104 -0.20820521 ...  1.160118   -0.50555708
  -0.50846081]
 [ 1.08028472 -0.83485626  0.6823922  ...  0.74665876 -0.06928955
   1.13129086]
 [ 0.97846793 -0.3248568   0.94957142 ...  0.74665876  1.00459976
  -0.50846081]]
```

In [17]:
```
print(X_test)
```

```
[[ 0.92243937 -0.80741306  0.01293697 ...  0.39981121 -0.48257105
   1.06361291]
 [-1.28386413 -0.80741306 -0.20203067 ... -2.47496427  1.27487407
  -0.59523291]
 [-1.08329109 -0.80741306 -0.20203067 ... -0.83223542 -0.37919193
   1.06361291]
 ...
 [-1.18357761  0.24506147 -0.20203067 ... -0.76378839  0.34446195
   1.06361291]
 [-0.78243152  1.297536   -0.08627887 ... -0.010871    1.58501145
   1.06361291]
 [ 0.01986066 -0.80741306 -0.1854947  ... -0.010871   -0.65486959
  -0.59523291]]
```

In [18]:
```
#Building the Model
from sklearn.neighbors import KNeighborsClassifier
classifier=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
classifier.fit(X_train,y_train)
KNeighborsClassifier()
y_pred=classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1),y_test.reshape(len
(y_test),1)),1))
```

```
[[1. 1.]
 [3. 3.]
 [1. 1.]
 ...
 [1. 1.]
 [1. 1.]
 [2. 1.]]
```

In [19]:
```
#Cross validation
parameters ={ 'n_neighbors' : [5,7,9,11,13,15],
 'weights' : ['uniform','distance'],
 'metric' : ['minkowski','euclidean','manhattan']}
from sklearn.model_selection import RandomizedSearchCV
cv = RandomizedSearchCV(classifier,parameters ,cv=5)
cv.fit(X_train,y_train)
RandomizedSearchCV(cv=5, estimator=KNeighborsClassifier(),param_distributions={'metric': ['minkowski','euclidean','manhattan'],'
y_pred = cv.predict(X_test)
```

In [21]:
```
from sklearn.metrics import accuracy_score
print('\n Hyperparametric tuned knn accuracy:',accuracy_score(y_pred,y_test))
```

```
 Hyperparametric tuned knn accuracy: 0.8960784313725491
```

In [22]:
```
test_set=pd.read_csv('test.csv')
y_pre =classifier.predict(test_set)
print(y_pre )
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

In [25]:

```
#Saving the file in csv format
y = pd.DataFrame(y_pred).astype(int)
y.to_csv('Result.csv')
```

In [ ]: