```
import pandas as pd
from pandas import read_csv
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import
accuracy_score,classification_report,confusion_matrix,r2_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
import warnings
warnings.filterwarnings("ignore")

data=pd.read_csv(r"C:\Users\Admin\Downloads\parkinsons.csv")
data.head()
```

```
             name  MDVP:Fo(Hz)  MDVP:Fhi(Hz)  MDVP:Flo(Hz)
MDVP:Jitter(%)  \
0  phon_R01_S01_1      119.992       157.302        74.997
0.008
1  phon_R01_S01_2      122.400       148.650       113.819
0.010
2  phon_R01_S01_3      116.682       131.111       111.555
0.011
3  phon_R01_S01_4      116.676       137.871       111.366
0.010
4  phon_R01_S01_5      116.014       141.781       110.655
0.013

   MDVP:Jitter(Abs)  MDVP:RAP  MDVP:PPQ  Jitter:DDP  MDVP:Shimmer  ...
\
0             0.000     0.004     0.006       0.011         0.044  ...

1             0.000     0.005     0.007       0.014         0.061  ...

2             0.000     0.005     0.008       0.016         0.052  ...

3             0.000     0.005     0.007       0.015         0.055  ...

4             0.000     0.007     0.009       0.020         0.064  ...


   Shimmer:DDA   NHR    HNR  status  RPDE   DFA  spread1  spread2
D2   PPE
0        0.065 0.022 21.033       1 0.415 0.815   -4.813    0.266
2.301 0.285
```

```
1        0.094 0.019 19.085       1 0.458 0.820   -4.075    0.336
2.487 0.369
2        0.083 0.013 20.651       1 0.430 0.825   -4.443    0.311
2.342 0.333
3        0.088 0.014 20.644       1 0.435 0.819   -4.118    0.334
2.406 0.369
4        0.105 0.018 19.649       1 0.417 0.823   -3.748    0.235
2.332 0.410

[5 rows x 24 columns]
```

```
data.shape
```

```
(195, 24)
```

```
data.duplicated().sum()
```

```
0
```

```
data.isnull().sum()
```

```
name                0
MDVP:Fo(Hz)         0
MDVP:Fhi(Hz)        0
MDVP:Flo(Hz)        0
MDVP:Jitter(%)      0
MDVP:Jitter(Abs)    0
MDVP:RAP            0
MDVP:PPQ            0
Jitter:DDP          0
MDVP:Shimmer        0
MDVP:Shimmer(dB)    0
Shimmer:APQ3        0
Shimmer:APQ5        0
MDVP:APQ            0
Shimmer:DDA         0
NHR                 0
HNR                 0
status              0
RPDE                0
DFA                 0
spread1             0
spread2             0
D2                  0
PPE                 0
dtype: int64
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
```

```
Data columns (total 24 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   name              195 non-null    object
 1   MDVP:Fo(Hz)       195 non-null    float64
 2   MDVP:Fhi(Hz)      195 non-null    float64
 3   MDVP:Flo(Hz)      195 non-null    float64
 4   MDVP:Jitter(%)    195 non-null    float64
 5   MDVP:Jitter(Abs)  195 non-null    float64
 6   MDVP:RAP          195 non-null    float64
 7   MDVP:PPQ          195 non-null    float64
 8   Jitter:DDP        195 non-null    float64
 9   MDVP:Shimmer      195 non-null    float64
 10  MDVP:Shimmer(dB)  195 non-null    float64
 11  Shimmer:APQ3      195 non-null    float64
 12  Shimmer:APQ5      195 non-null    float64
 13  MDVP:APQ          195 non-null    float64
 14  Shimmer:DDA       195 non-null    float64
 15  NHR               195 non-null    float64
 16  HNR               195 non-null    float64
 17  status            195 non-null    int64
 18  RPDE              195 non-null    float64
 19  DFA               195 non-null    float64
 20  spread1           195 non-null    float64
 21  spread2           195 non-null    float64
 22  D2                195 non-null    float64
 23  PPE               195 non-null    float64
dtypes: float64(22), int64(1), object(1)
memory usage: 36.7+ KB
```

```
data.describe()
```

```
       MDVP:Fo(Hz)  MDVP:Fhi(Hz)  MDVP:Flo(Hz)  MDVP:Jitter(%)  \
count      195.000       195.000       195.000         195.000
mean       154.229       197.105       116.325           0.006
std         41.390        91.492        43.521           0.005
min         88.333       102.145        65.476           0.002
25%        117.572       134.863        84.291           0.003
50%        148.790       175.829       104.315           0.005
75%        182.769       224.206       140.019           0.007
max        260.105       592.030       239.170           0.033

       MDVP:Jitter(Abs)  MDVP:RAP  MDVP:PPQ  Jitter:DDP  MDVP:Shimmer
\
count           195.000   195.000   195.000     195.000       195.000

mean              0.000     0.003     0.003       0.010         0.030

std               0.000     0.003     0.003       0.009         0.019
```

```
min                0.000     0.001     0.001       0.002          0.010

25%                0.000     0.002     0.002       0.005          0.017

50%                0.000     0.003     0.003       0.007          0.023

75%                0.000     0.004     0.004       0.012          0.038

max                0.000     0.021     0.020       0.064          0.119


       MDVP:Shimmer(dB)  ...  Shimmer:DDA     NHR     HNR  status
RPDE  \
count           195.000  ...      195.000 195.000 195.000 195.000
195.000
mean              0.282  ...        0.047   0.025  21.886   0.754
0.499
std               0.195  ...        0.030   0.040   4.426   0.432
0.104
min               0.085  ...        0.014   0.001   8.441   0.000
0.257
25%               0.148  ...        0.025   0.006  19.198   1.000
0.421
50%               0.221  ...        0.038   0.012  22.085   1.000
0.496
75%               0.350  ...        0.061   0.026  25.075   1.000
0.588
max               1.302  ...        0.169   0.315  33.047   1.000
0.685

          DFA  spread1  spread2      D2     PPE
count 195.000  195.000  195.000 195.000 195.000
mean    0.718   -5.684    0.227   2.382   0.207
std     0.055    1.090    0.083   0.383   0.090
min     0.574   -7.965    0.006   1.423   0.045
25%     0.675   -6.450    0.174   2.099   0.137
50%     0.722   -5.721    0.219   2.362   0.194
75%     0.762   -5.046    0.279   2.636   0.253
max     0.825   -2.434    0.450   3.671   0.527

[8 rows x 23 columns]
```

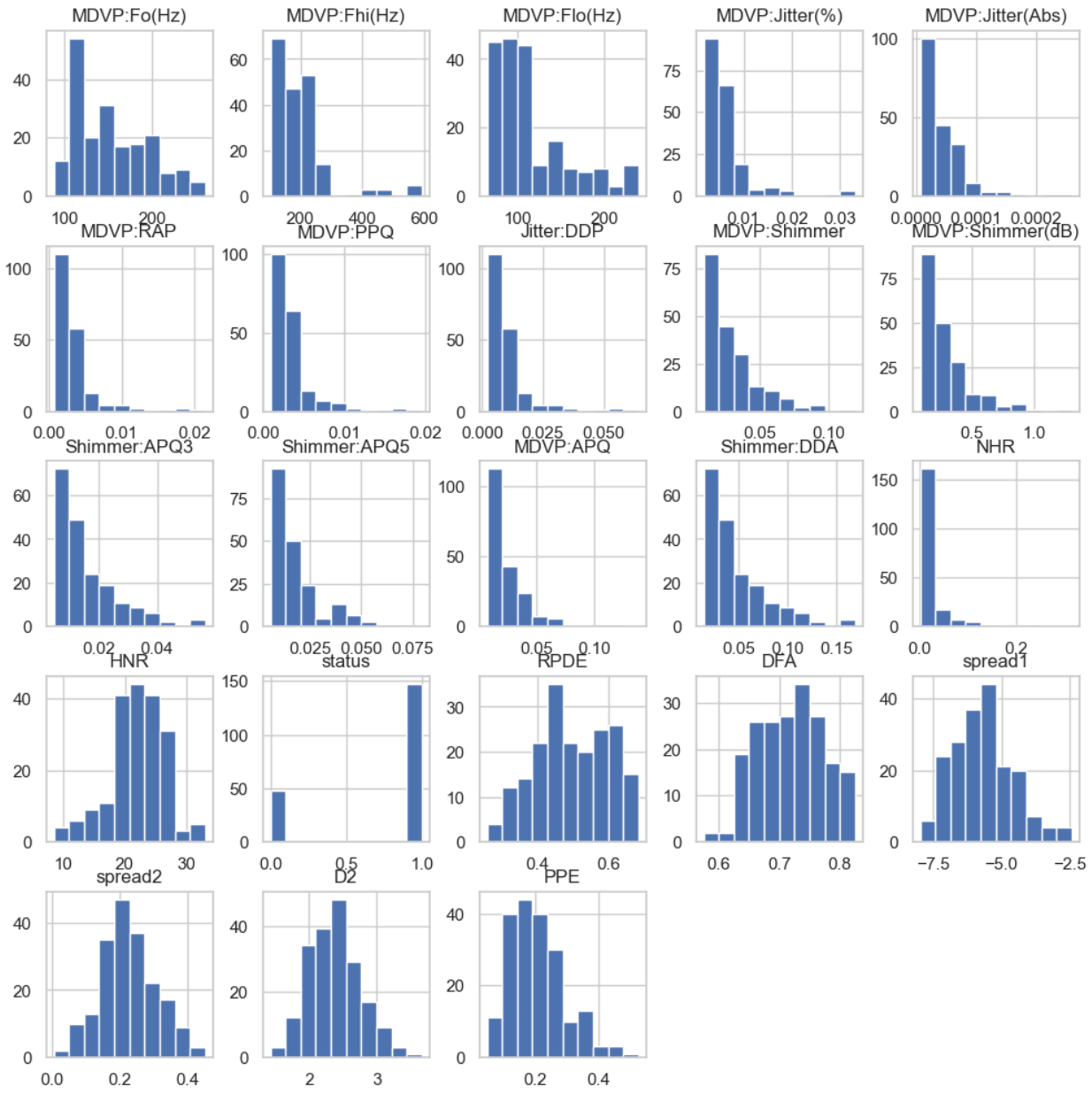
```
pd.set_option('display.float_format',lambda x:'%.3f' %x)
data.describe().transpose()
```

```
                   count    mean    std     min     25%     50%
75%  \
MDVP:Fo(Hz)      195.000 154.229 41.390  88.333 117.572 148.790
182.769
MDVP:Fhi(Hz)     195.000 197.105 91.492 102.145 134.863 175.829
```

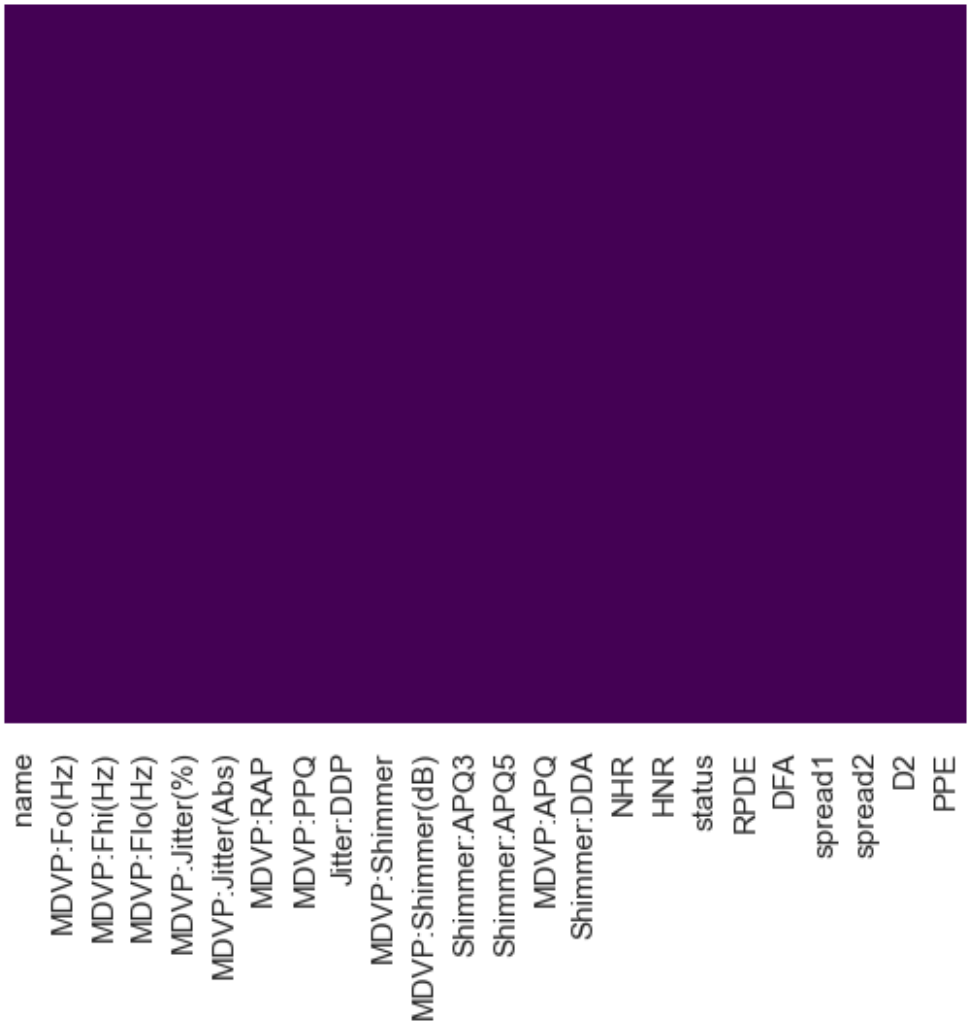
```
224.206
MDVP:Flo(Hz)     195.000 116.325 43.521  65.476  84.291 104.315
140.019
MDVP:Jitter(%)   195.000   0.006  0.005   0.002   0.003   0.005
0.007
MDVP:Jitter(Abs) 195.000   0.000  0.000   0.000   0.000   0.000
0.000
MDVP:RAP         195.000   0.003  0.003   0.001   0.002   0.003
0.004
MDVP:PPQ         195.000   0.003  0.003   0.001   0.002   0.003
0.004
Jitter:DDP       195.000   0.010  0.009   0.002   0.005   0.007
0.012
MDVP:Shimmer     195.000   0.030  0.019   0.010   0.017   0.023
0.038
MDVP:Shimmer(dB) 195.000   0.282  0.195   0.085   0.148   0.221
0.350
Shimmer:APQ3     195.000   0.016  0.010   0.005   0.008   0.013
0.020
Shimmer:APQ5     195.000   0.018  0.012   0.006   0.010   0.013
0.022
MDVP:APQ         195.000   0.024  0.017   0.007   0.013   0.018
0.029
Shimmer:DDA      195.000   0.047  0.030   0.014   0.025   0.038
0.061
NHR              195.000   0.025  0.040   0.001   0.006   0.012
0.026
HNR              195.000  21.886  4.426   8.441  19.198  22.085
25.075
status           195.000   0.754  0.432   0.000   1.000   1.000
1.000
RPDE             195.000   0.499  0.104   0.257   0.421   0.496
0.588
DFA              195.000   0.718  0.055   0.574   0.675   0.722
0.762
spread1          195.000  -5.684  1.090  -7.965  -6.450  -5.721  -
5.046
spread2          195.000   0.227  0.083   0.006   0.174   0.219
0.279
D2               195.000   2.382  0.383   1.423   2.099   2.362
2.636
PPE              195.000   0.207  0.090   0.045   0.137   0.194
0.253

                     max
MDVP:Fo(Hz)      260.105
MDVP:Fhi(Hz)     592.030
MDVP:Flo(Hz)     239.170
MDVP:Jitter(%)     0.033
```

```
MDVP:Jitter(Abs)   0.000
MDVP:RAP           0.021
MDVP:PPQ           0.020
Jitter:DDP         0.064
MDVP:Shimmer       0.119
MDVP:Shimmer(dB)   1.302
Shimmer:APQ3       0.056
Shimmer:APQ5       0.079
MDVP:APQ           0.138
Shimmer:DDA        0.169
NHR                0.315
HNR               33.047
status             1.000
RPDE               0.685
DFA                0.825
spread1           -2.434
spread2            0.450
D2                 3.671
PPE                0.527
```

```
data.hist(figsize=(12,12))
plt.show()
```

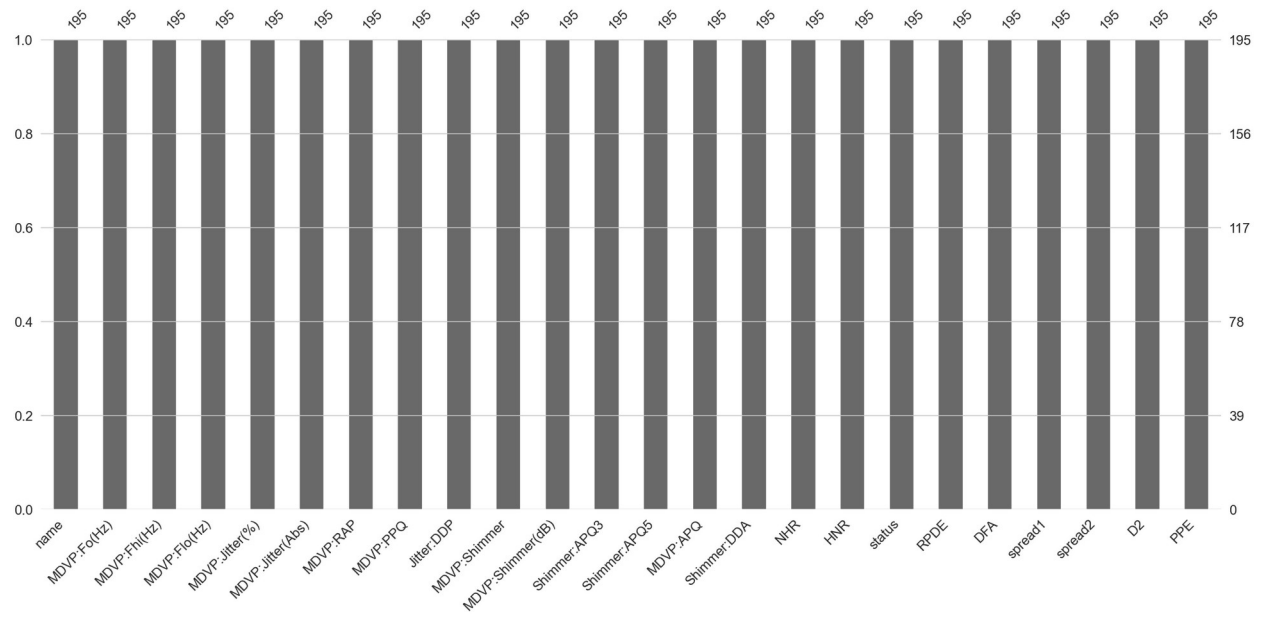



```
sns.heatmap(data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
plt.show()
```




```
import missingno as msno
msno.bar(data)
```

```
<AxesSubplot:>
```




```
data.corr()

                  MDVP:Fo(Hz)  MDVP:Fhi(Hz)  MDVP:Flo(Hz)
MDVP:Jitter(%)  \
MDVP:Fo(Hz)             1.000         0.401         0.597           -
0.118
MDVP:Fhi(Hz)            0.401         1.000         0.085
0.102
MDVP:Flo(Hz)            0.597         0.085         1.000           -
0.140
MDVP:Jitter(%)         -0.118         0.102        -0.140
1.000
MDVP:Jitter(Abs)       -0.382        -0.029        -0.278
0.936
MDVP:RAP               -0.076         0.097        -0.101
0.990
MDVP:PPQ               -0.112         0.091        -0.096
0.974
Jitter:DDP             -0.076         0.097        -0.100
0.990
MDVP:Shimmer           -0.098         0.002        -0.145
0.769
MDVP:Shimmer(dB)       -0.074         0.043        -0.119
0.804
Shimmer:APQ3           -0.095        -0.004        -0.151
0.747
Shimmer:APQ5           -0.071        -0.010        -0.101
0.726
MDVP:APQ               -0.078         0.005        -0.107
0.758
Shimmer:DDA            -0.095        -0.004        -0.151
```

```
0.747
NHR                      -0.022          0.164        -0.109
0.907
HNR                       0.059        -0.025         0.211           -
0.728
status                   -0.384        -0.166        -0.380
0.278
RPDE                     -0.384        -0.112        -0.400
0.361
DFA                      -0.446        -0.343        -0.050
0.099
spread1                  -0.414        -0.077        -0.395
0.694
spread2                  -0.249        -0.003        -0.244
0.385
D2                        0.178         0.176        -0.101
0.433
PPE                      -0.372        -0.070        -0.340
0.722

                  MDVP:Jitter(Abs)  MDVP:RAP  MDVP:PPQ  Jitter:DDP  \
MDVP:Fo(Hz)                -0.382    -0.076    -0.112      -0.076
MDVP:Fhi(Hz)               -0.029     0.097     0.091       0.097
MDVP:Flo(Hz)               -0.278    -0.101    -0.096      -0.100
MDVP:Jitter(%)              0.936     0.990     0.974       0.990
MDVP:Jitter(Abs)            1.000     0.923     0.898       0.923
MDVP:RAP                    0.923     1.000     0.957       1.000
MDVP:PPQ                    0.898     0.957     1.000       0.957
Jitter:DDP                  0.923     1.000     0.957       1.000
MDVP:Shimmer                0.703     0.760     0.798       0.760
MDVP:Shimmer(dB)            0.717     0.791     0.839       0.791
Shimmer:APQ3                0.697     0.745     0.764       0.745
Shimmer:APQ5                0.649     0.710     0.787       0.710
MDVP:APQ                    0.649     0.737     0.804       0.737
Shimmer:DDA                 0.697     0.745     0.764       0.745
NHR                         0.835     0.920     0.845       0.920
HNR                        -0.657    -0.722    -0.732      -0.721
status                      0.339     0.267     0.289       0.267
RPDE                        0.442     0.342     0.333       0.342
DFA                         0.175     0.064     0.196       0.064
spread1                     0.736     0.648     0.716       0.648
spread2                     0.389     0.324     0.408       0.324
D2                          0.311     0.427     0.413       0.427
PPE                         0.748     0.671     0.770       0.671

                  MDVP:Shimmer  MDVP:Shimmer(dB)  ...  Shimmer:DDA
NHR  \
MDVP:Fo(Hz)             -0.098            -0.074  ...       -0.095 -
0.022
```

```
MDVP:Fhi(Hz)              0.002             0.043  ...        -0.004
0.164
MDVP:Flo(Hz)             -0.145            -0.119  ...        -0.151 -
0.109
MDVP:Jitter(%)            0.769             0.804  ...         0.747
0.907
MDVP:Jitter(Abs)          0.703             0.717  ...         0.697
0.835
MDVP:RAP                  0.760             0.791  ...         0.745
0.920
MDVP:PPQ                  0.798             0.839  ...         0.764
0.845
Jitter:DDP                0.760             0.791  ...         0.745
0.920
MDVP:Shimmer              1.000             0.987  ...         0.988
0.722
MDVP:Shimmer(dB)          0.987             1.000  ...         0.963
0.744
Shimmer:APQ3              0.988             0.963  ...         1.000
0.716
Shimmer:APQ5              0.983             0.974  ...         0.960
0.658
MDVP:APQ                  0.950             0.961  ...         0.897
0.694
Shimmer:DDA               0.988             0.963  ...         1.000
0.716
NHR                       0.722             0.744  ...         0.716
1.000
HNR                      -0.835            -0.828  ...        -0.827 -
0.714
status                    0.367             0.351  ...         0.348
0.189
RPDE                      0.447             0.411  ...         0.435
0.371
DFA                       0.160             0.165  ...         0.151 -
0.132
spread1                   0.655             0.653  ...         0.611
0.541
spread2                   0.452             0.454  ...         0.402
0.318
D2                        0.507             0.512  ...         0.467
0.471
PPE                       0.694             0.695  ...         0.645
0.553

                     HNR  status   RPDE    DFA  spread1  spread2     D2
PPE
MDVP:Fo(Hz)        0.059  -0.384 -0.384 -0.446   -0.414   -0.249  0.178
-0.372
```

```
MDVP:Fhi(Hz)     -0.025  -0.166 -0.112 -0.343   -0.077   -0.003  0.176
-0.070
MDVP:Flo(Hz)      0.211  -0.380 -0.400 -0.050   -0.395   -0.244 -0.101
-0.340
MDVP:Jitter(%)   -0.728   0.278  0.361  0.099    0.694    0.385  0.433
0.722
MDVP:Jitter(Abs) -0.657   0.339  0.442  0.175    0.736    0.389  0.311
0.748
MDVP:RAP         -0.722   0.267  0.342  0.064    0.648    0.324  0.427
0.671
MDVP:PPQ         -0.732   0.289  0.333  0.196    0.716    0.408  0.413
0.770
Jitter:DDP       -0.721   0.267  0.342  0.064    0.648    0.324  0.427
0.671
MDVP:Shimmer     -0.835   0.367  0.447  0.160    0.655    0.452  0.507
0.694
MDVP:Shimmer(dB) -0.828   0.351  0.411  0.165    0.653    0.454  0.512
0.695
Shimmer:APQ3     -0.827   0.348  0.435  0.151    0.611    0.402  0.467
0.645
Shimmer:APQ5     -0.814   0.351  0.400  0.214    0.647    0.457  0.502
0.702
MDVP:APQ         -0.800   0.364  0.451  0.157    0.673    0.502  0.537
0.722
Shimmer:DDA      -0.827   0.348  0.435  0.151    0.611    0.402  0.467
0.645
NHR              -0.714   0.189  0.371 -0.132    0.541    0.318  0.471
0.553
HNR               1.000  -0.362 -0.599 -0.009   -0.673   -0.432 -0.601
-0.693
status           -0.362   1.000  0.309  0.232    0.565    0.455  0.340
0.531
RPDE             -0.599   0.309  1.000 -0.111    0.591    0.480  0.237
0.546
DFA              -0.009   0.232 -0.111  1.000    0.196    0.167 -0.165
0.270
spread1          -0.673   0.565  0.591  0.196    1.000    0.652  0.495
0.962
spread2          -0.432   0.455  0.480  0.167    0.652    1.000  0.524
0.645
D2               -0.601   0.340  0.237 -0.165    0.495    0.524  1.000
0.481
PPE              -0.693   0.531  0.546  0.270    0.962    0.645  0.481
1.000

[23 rows x 23 columns]
```

```
data.corr()['spread1']
```

```
MDVP:Fo(Hz)         -0.414
MDVP:Fhi(Hz)        -0.077
MDVP:Flo(Hz)        -0.395
MDVP:Jitter(%)       0.694
MDVP:Jitter(Abs)     0.736
MDVP:RAP             0.648
MDVP:PPQ             0.716
Jitter:DDP           0.648
MDVP:Shimmer         0.655
MDVP:Shimmer(dB)     0.653
Shimmer:APQ3         0.611
Shimmer:APQ5         0.647
MDVP:APQ             0.673
Shimmer:DDA          0.611
NHR                  0.541
HNR                 -0.673
status               0.565
RPDE                 0.591
DFA                  0.196
spread1              1.000
spread2              0.652
D2                   0.495
PPE                  0.962
Name: spread1, dtype: float64
```

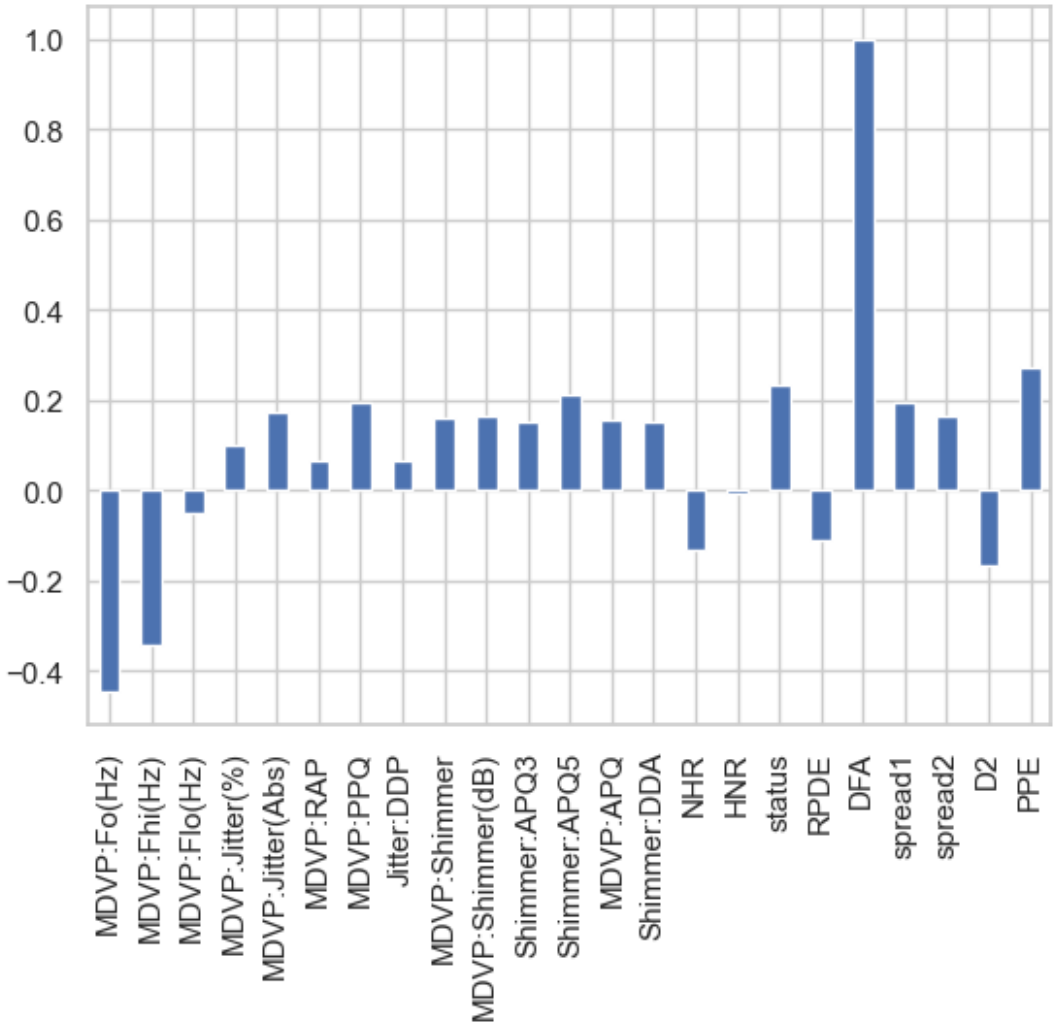
```
data.corr()['PPE']
```

```
MDVP:Fo(Hz)         -0.372
MDVP:Fhi(Hz)        -0.070
MDVP:Flo(Hz)        -0.340
MDVP:Jitter(%)       0.722
MDVP:Jitter(Abs)     0.748
MDVP:RAP             0.671
MDVP:PPQ             0.770
Jitter:DDP           0.671
MDVP:Shimmer         0.694
MDVP:Shimmer(dB)     0.695
Shimmer:APQ3         0.645
Shimmer:APQ5         0.702
MDVP:APQ             0.722
Shimmer:DDA          0.645
NHR                  0.553
HNR                 -0.693
status               0.531
RPDE                 0.546
DFA                  0.270
spread1              0.962
spread2              0.645
D2                   0.481
```

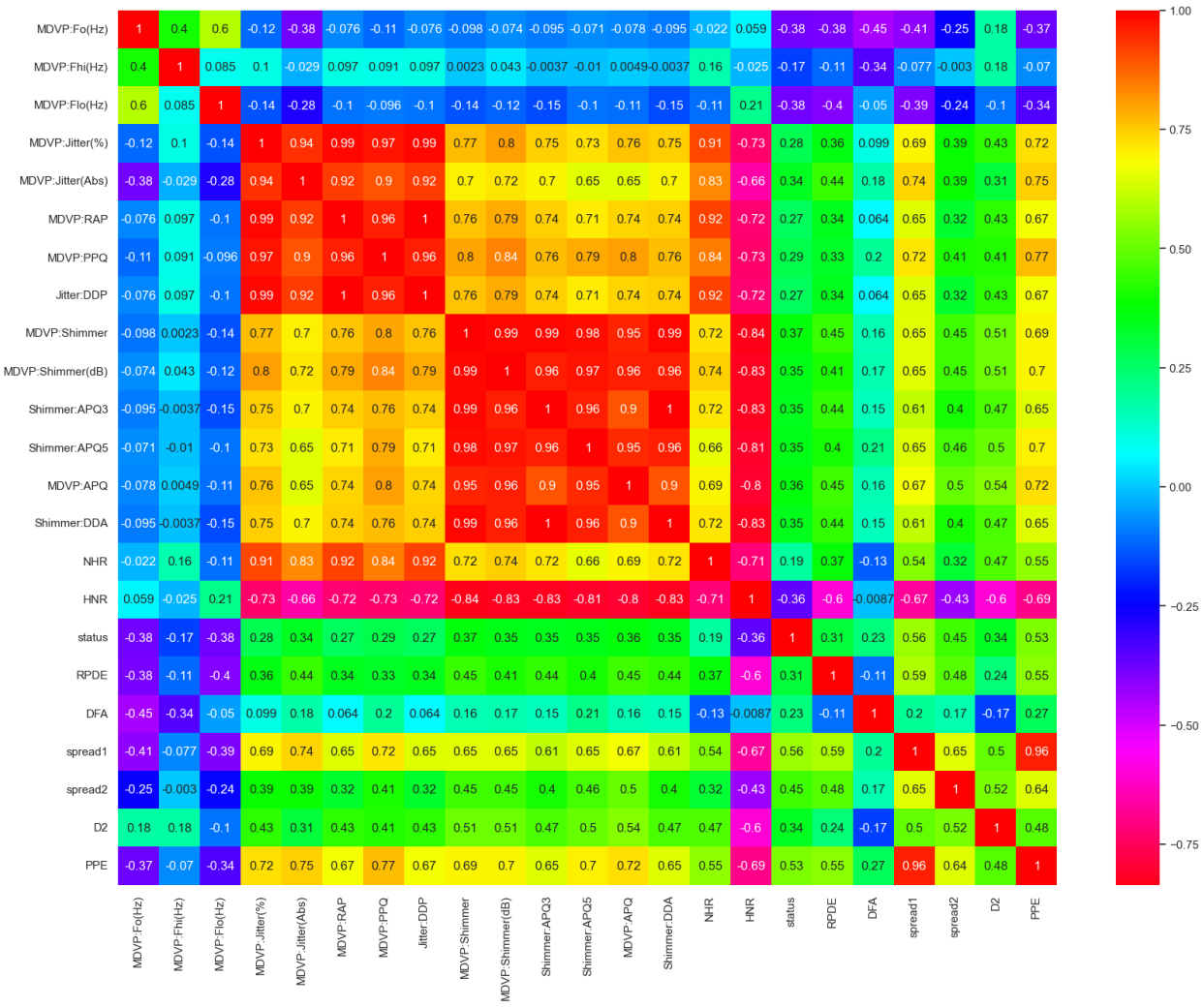
```
PPE                  1.000
Name: PPE, dtype: float64
```

```
data.corr()['DFA'].plot(kind='bar')
```

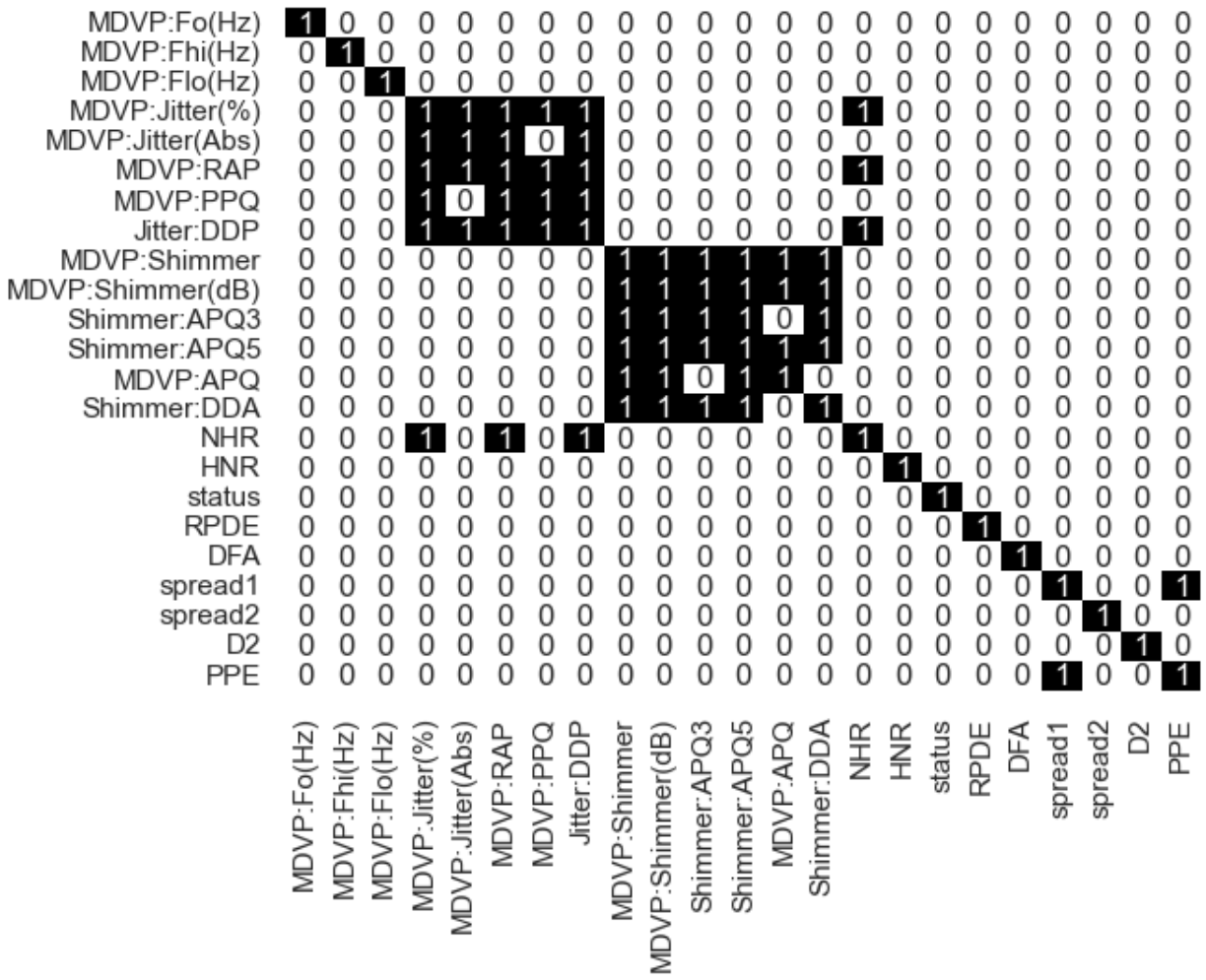
```
<AxesSubplot:>
```



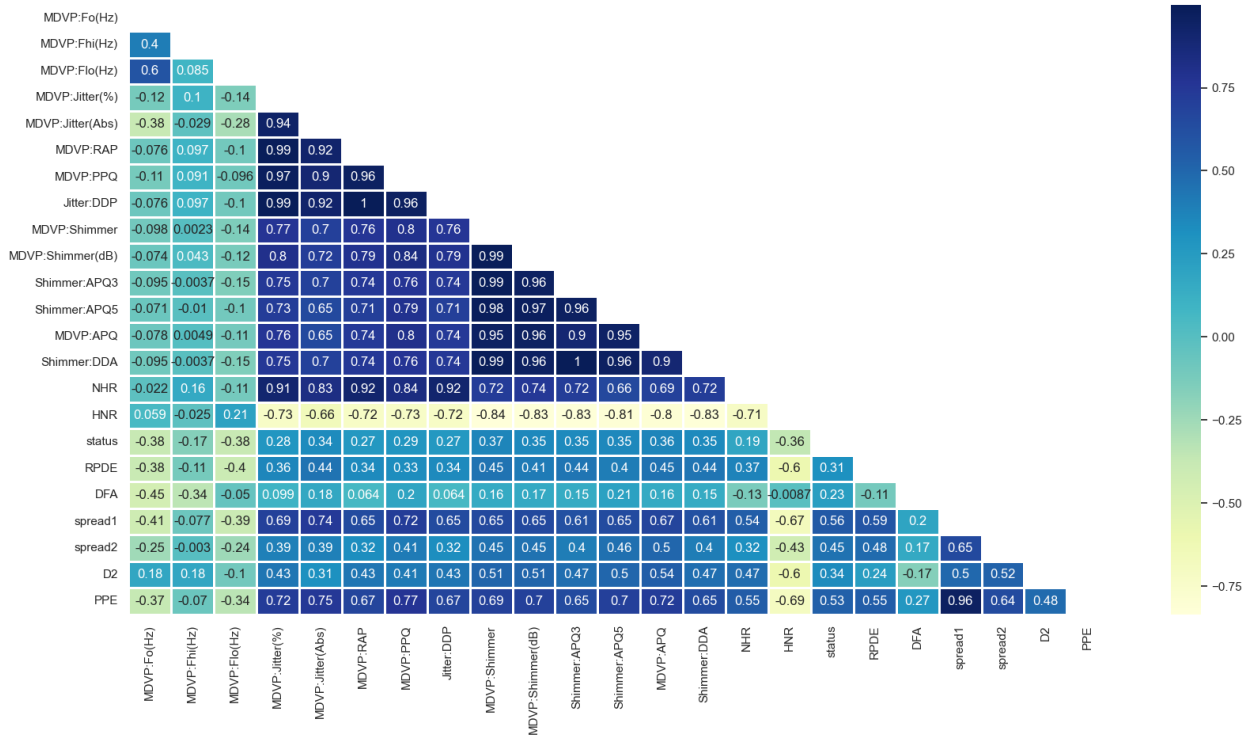

```
plt.figure(figsize=(20,15))
corr = data.corr()
sns.heatmap(data.corr(), cmap="hsv_r", annot=True)
plt.show()
```



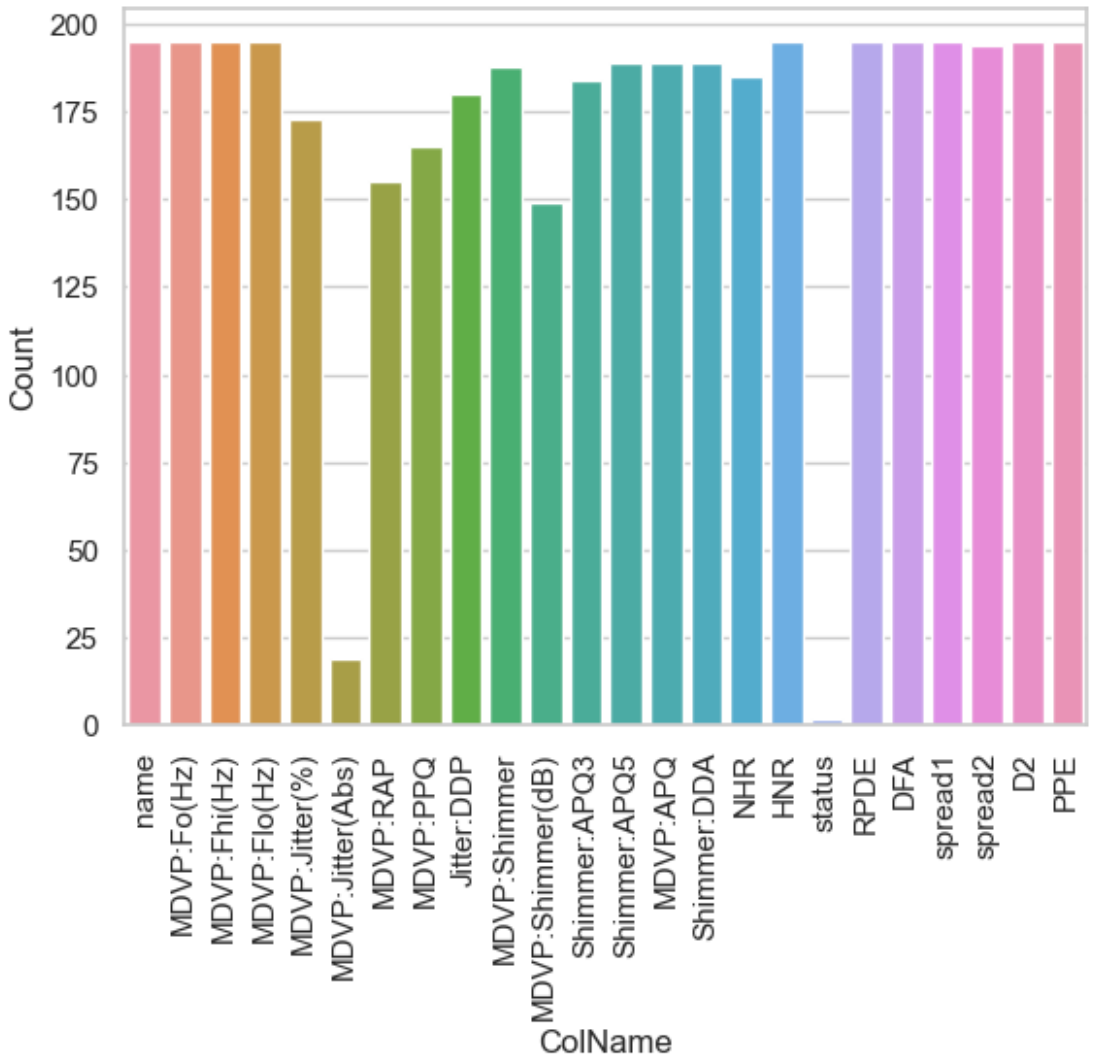

```
sns.heatmap(data.corr() > 0.9, annot=True,
cbar=False,cmap="gnuplot2_r")
plt.show()
```



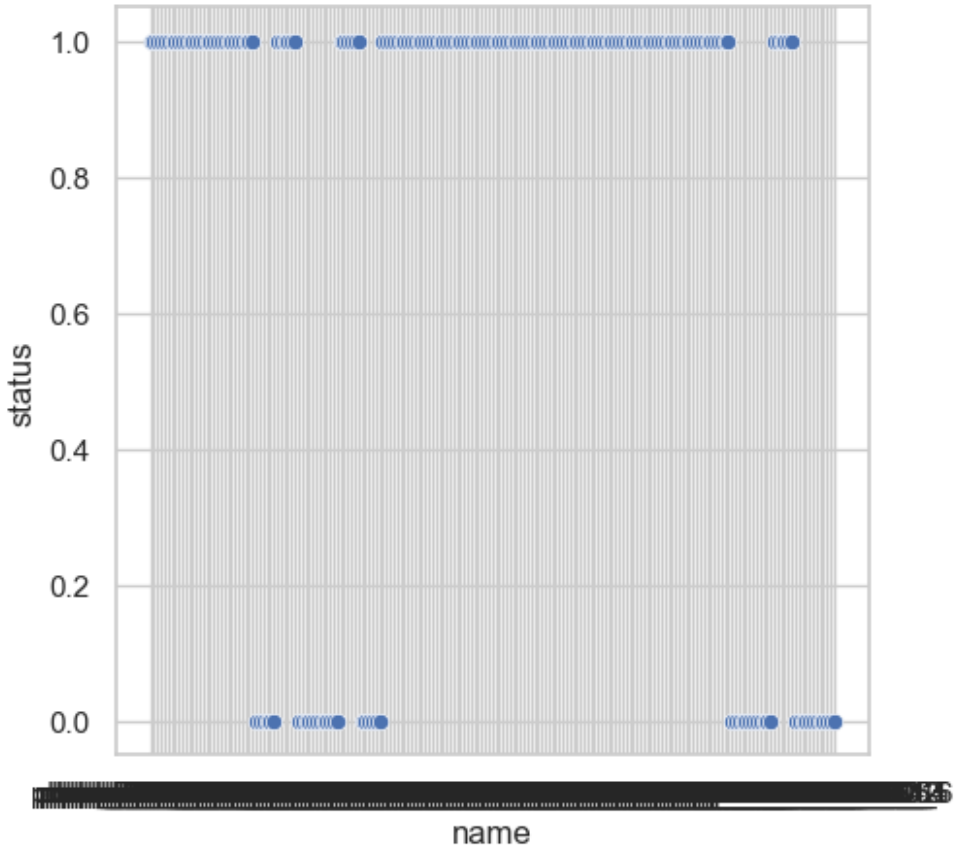

```
plt.figure(figsize=(20,10))
corr = data.corr()
mask=np.triu(np.ones_like(corr,dtype=bool))

sns.heatmap(data=corr, mask=mask,
cmap="YlGnBu",annot=True,linewidth=2)
plt.show()
```



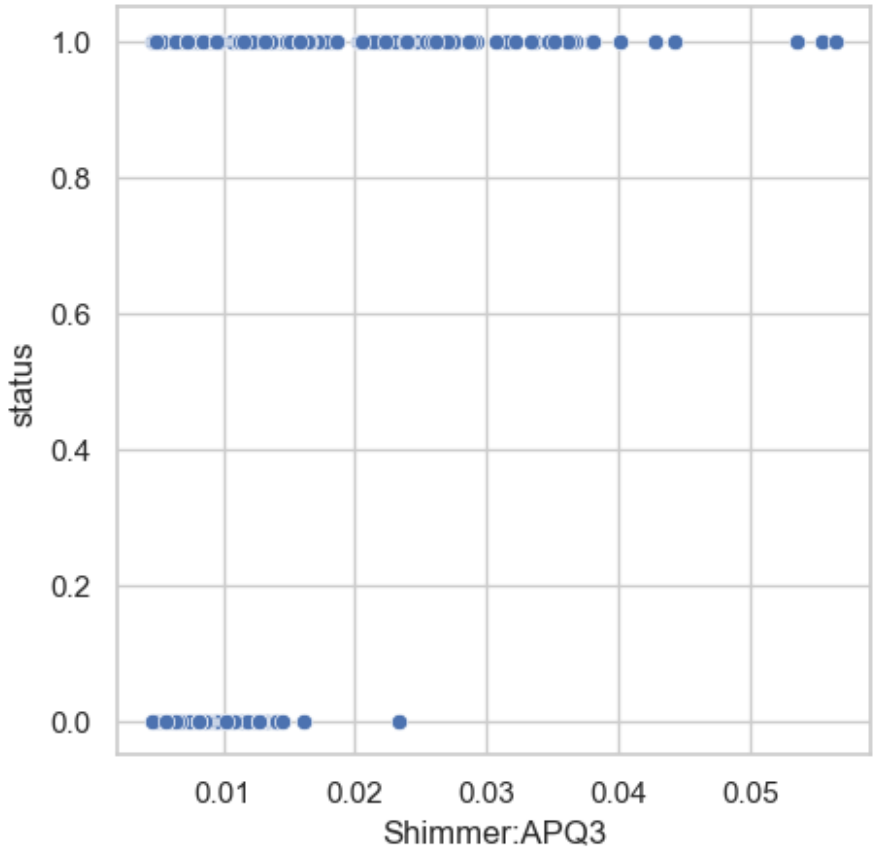

```
unique=data.nunique().to_frame()
unique.columns=['Count']
unique.index.names=['ColName']
unique=unique.reset_index()
sns.set(style='whitegrid',color_codes='True')
sns.barplot(x='ColName', y = 'Count', data = unique)
plt.xticks(rotation=90)
plt.show()
```




```
plt.figure(figsize=(5,5))
sns.scatterplot(x=data['name'],y=data['status'])
plt.show()
```

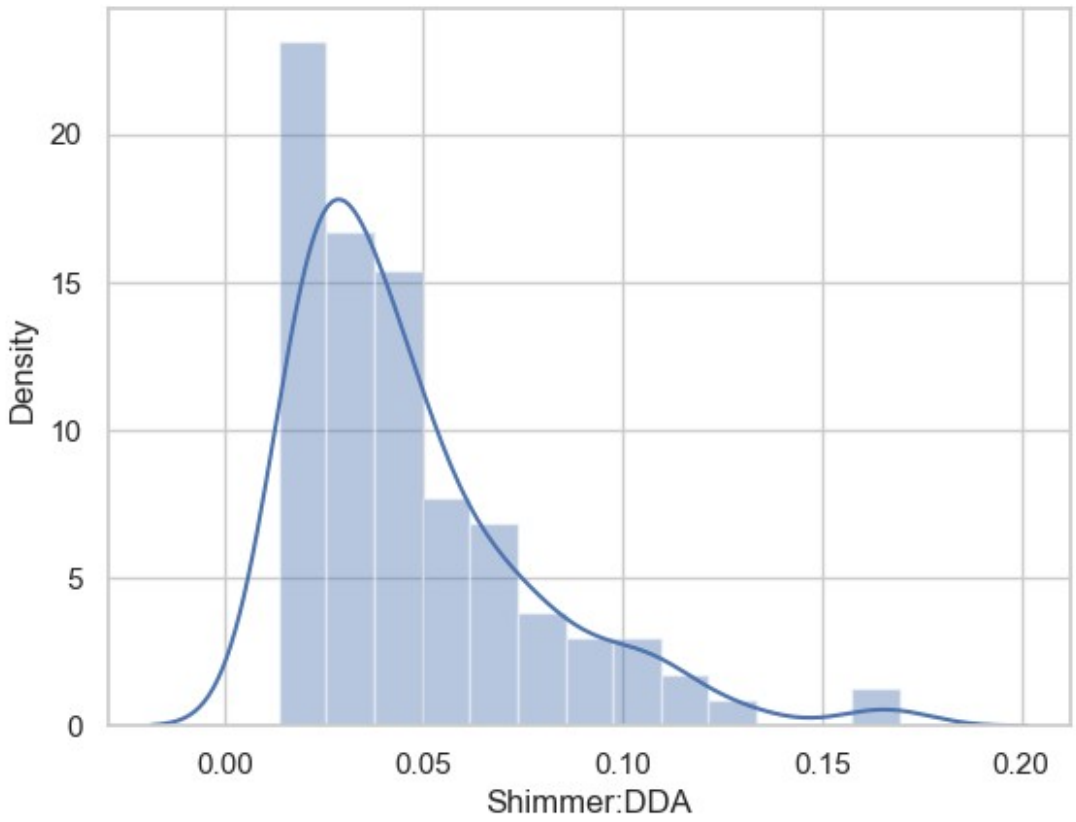



```
plt.figure(figsize=(5,5))
sns.scatterplot(x=data['Shimmer:APQ3'],y=data['status'])
plt.show()
```

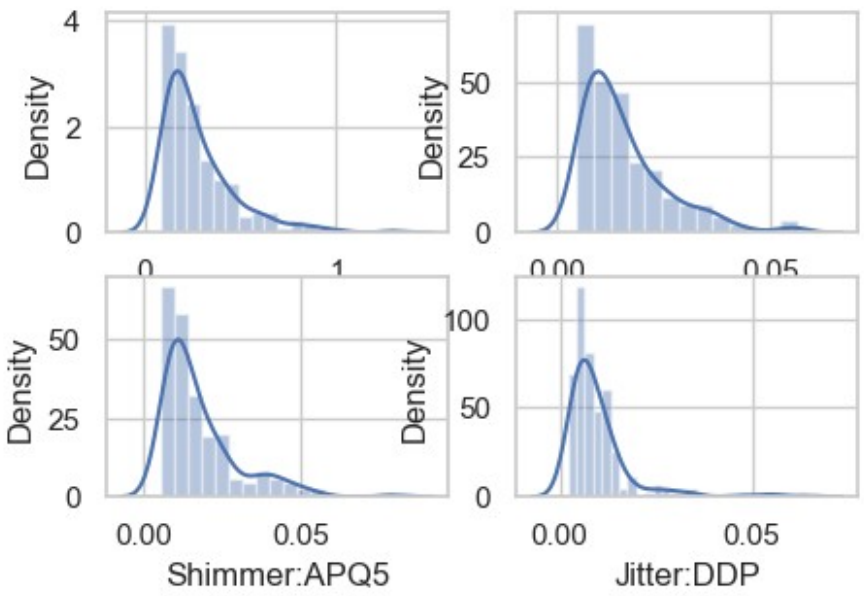



```
sns.distplot(data['Shimmer:DDA'])
```

```
<AxesSubplot:xlabel='Shimmer:DDA', ylabel='Density'>
```

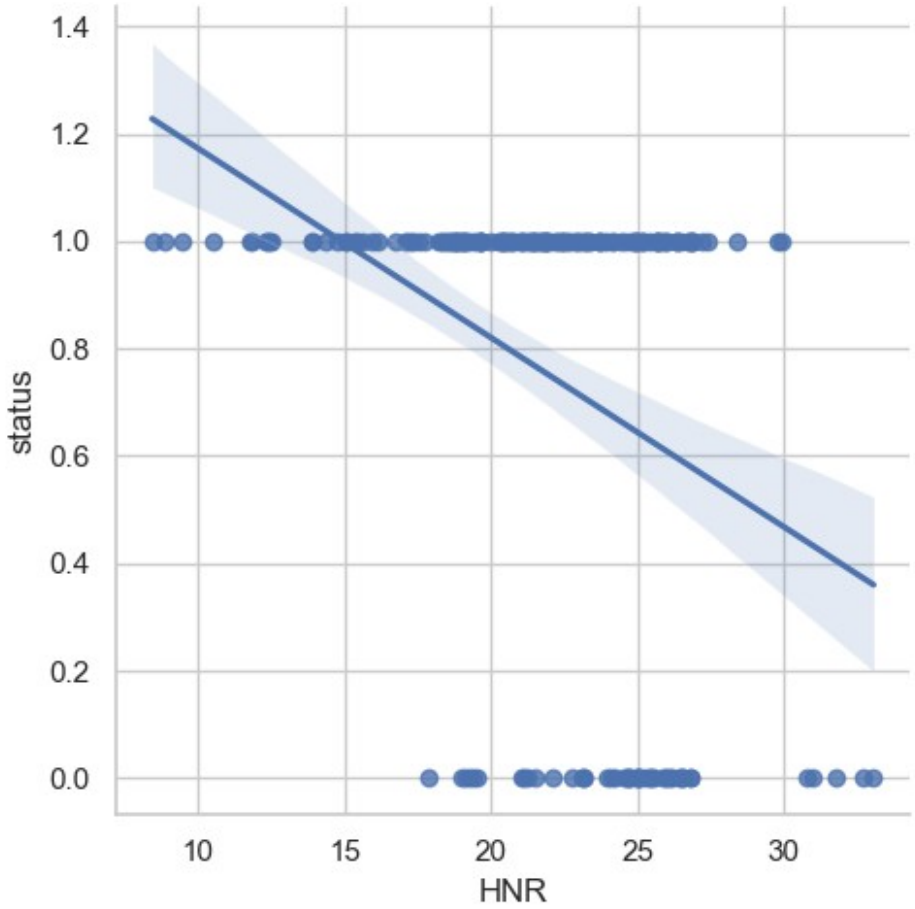



```
features =
['MDVP:Shimmer(dB)','Shimmer:APQ3','Shimmer:APQ5','Jitter:DDP']
plt.subplots(figsize=(5,5))

for i, col in enumerate(features):
  plt.subplot(3,2,i+1)
  sns.distplot(data[col])
plt.show()
```



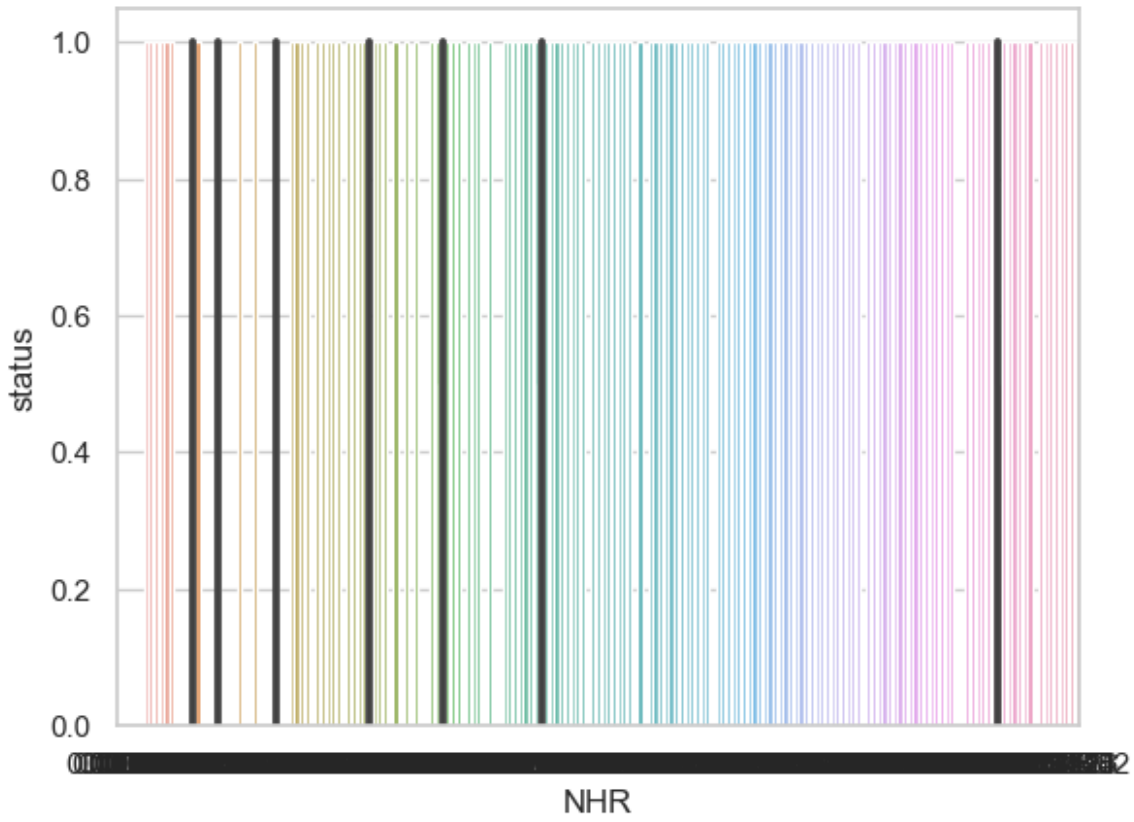

```
sns.lmplot(x='HNR', y='status', data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x2a179cd3dc0>
```




```
sns.barplot(x='NHR', y='status', data=data)
```

```
<AxesSubplot:xlabel='NHR', ylabel='status'>
```




```
data.value_counts()

name            MDVP:Fo(Hz)  MDVP:Fhi(Hz)  MDVP:Flo(Hz)
MDVP:Jitter(%)  MDVP:Jitter(Abs)  MDVP:RAP  MDVP:PPQ  Jitter:DDP
MDVP:Shimmer  MDVP:Shimmer(dB)  Shimmer:APQ3  Shimmer:APQ5  MDVP:APQ
Shimmer:DDA  NHR    HNR     status  RPDE   DFA    spread1  spread2  D2
PPE
phon_R01_S01_1  119.992      157.302       74.997        0.008
0.000             0.004     0.006     0.011       0.044         0.426
0.022         0.031         0.030     0.065        0.022  21.033  1
0.415  0.815  -4.813   0.266    2.301  0.285    1
phon_R01_S35_1  169.774      191.759       151.451       0.016
0.000             0.009     0.009     0.026       0.081         0.821
0.038         0.054         0.088     0.114        0.075  12.359  1
0.562  0.794  -3.298   0.415    3.414  0.458    1
phon_R01_S31_3  156.239      195.107       79.820        0.007
0.000             0.004     0.004     0.012       0.023         0.224
0.013         0.014         0.017     0.039        0.020  19.196  1
0.619  0.728  -5.944   0.143    1.930  0.182    1
phon_R01_S31_4  145.174      198.109       80.637        0.007
0.000             0.004     0.004     0.012       0.024         0.233
0.012         0.015         0.019     0.037        0.019  18.857  1
0.638  0.736  -5.594   0.128    1.766  0.223    1
phon_R01_S31_5  138.145      197.238       81.114        0.005
```

```
0.000             0.003     0.003     0.009       0.028         0.246
0.015         0.018         0.023     0.045        0.018  18.178  1
0.623  0.738  -5.540   0.087    1.821  0.214    1

..
phon_R01_S18_2  136.969      166.607       66.004        0.009
0.000             0.005     0.005     0.015       0.031         0.308
0.016         0.018         0.026     0.049        0.027  19.979  1
0.498  0.729  -5.325   0.206    2.292  0.226    1
phon_R01_S18_3  143.533      162.215       65.809        0.011
0.000             0.006     0.005     0.019       0.054         0.478
0.032         0.024         0.034     0.095        0.049  20.338  1
0.513  0.731  -5.870   0.152    2.118  0.186    1
phon_R01_S18_4  148.090      162.824       67.343        0.008
0.000             0.005     0.004     0.014       0.054         0.497
0.034         0.025         0.036     0.101        0.024  21.718  1
0.487  0.727  -6.261   0.121    2.137  0.142    1
phon_R01_S18_5  142.729      162.408       65.476        0.008
0.000             0.005     0.004     0.014       0.035         0.365
0.019         0.019         0.029     0.056        0.026  20.264  1
0.489  0.730  -5.721   0.159    2.278  0.181    1
phon_R01_S50_6  214.289      260.277       77.973        0.006
0.000             0.003     0.003     0.009       0.019         0.190
0.010         0.012         0.014     0.031        0.044  21.209  0
0.463  0.664  -5.724   0.191    2.555  0.149    1
Length: 195, dtype: int64
```

```
data['status'].value_counts()
```

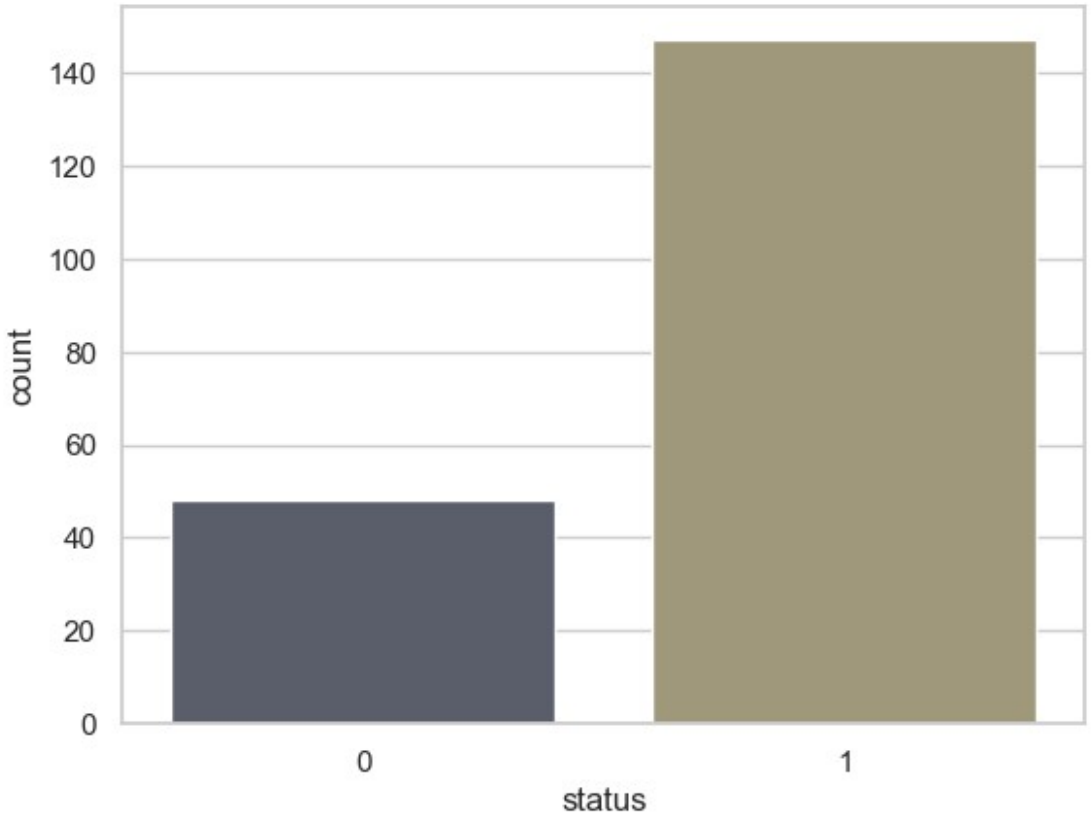
```
1    147
0     48
Name: status, dtype: int64
```

```
data['MDVP:RAP'].value_counts()
```

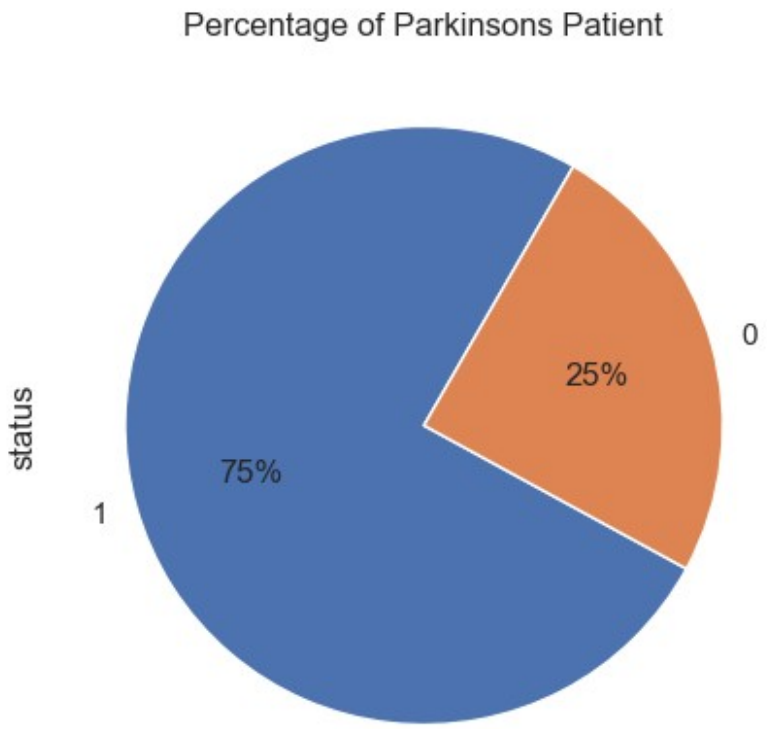
```
0.002    5
0.001    3
0.004    3
0.003    3
0.002    3
        ..
0.001    1
0.001    1
0.002    1
0.001    1
0.003    1
Name: MDVP:RAP, Length: 155, dtype: int64
```

```
sns.set_style('whitegrid')
sns.countplot(x='status',data=data,palette='cividis')
```

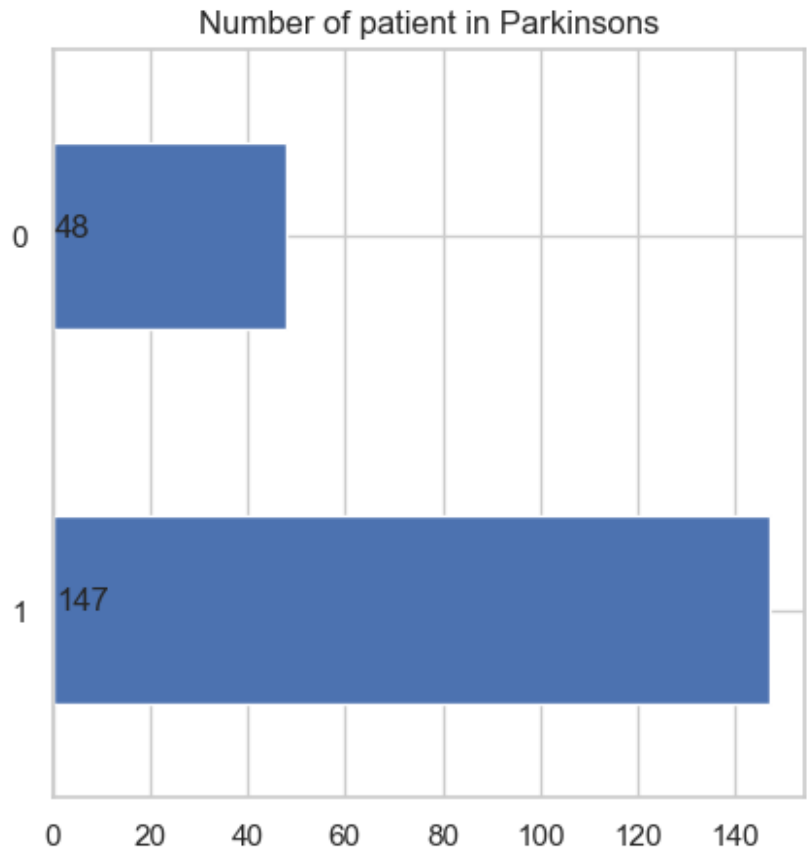
```
<AxesSubplot:xlabel='status', ylabel='count'>
```



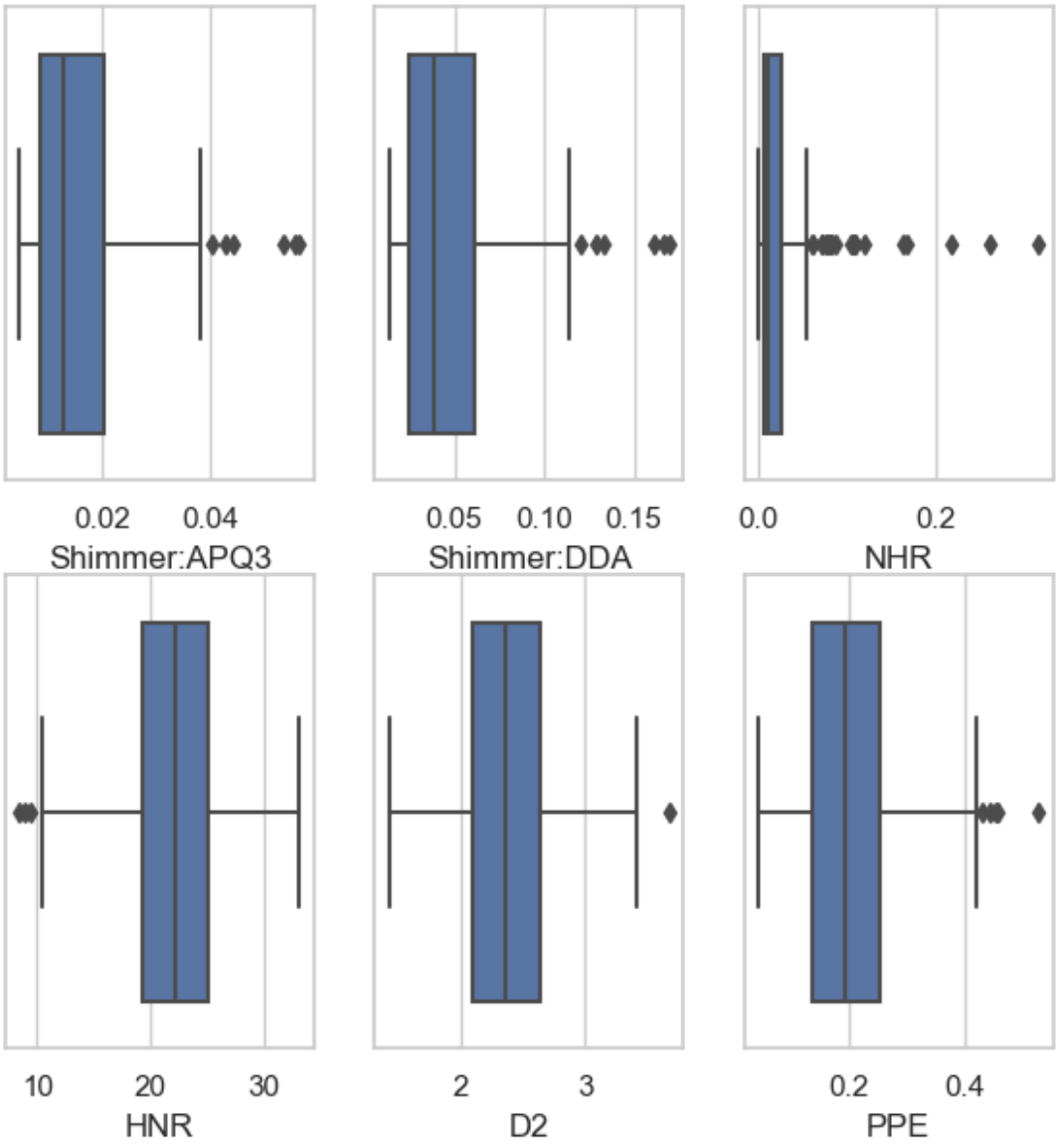

```
fig,ax=plt.subplots(nrows=1,ncols=1,sharey=False,figsize=(5,5))
ax=data['status'].value_counts().plot.pie(x='name',y='NHR',autopct='%1
.0f%%',startangle=60,ax=ax);
ax.set(title='Percentage of Parkinsons Patient')
plt.show()
```




```
fig,ax=plt.subplots(nrows=1,ncols=1,sharey=False,figsize=(5,5))
data['status'].value_counts().plot(kind='barh',ax=ax)
for i, j in enumerate(data['status'].value_counts().values):
    ax.text(.5,i,j,fontsize=12)
ax.set(title='Number of patient in Parkinsons')
plt.show()
```




```
features = ['Shimmer:APQ3','Shimmer:DDA','NHR','HNR','D2','PPE']

plt.subplots(figsize=(7,7))

for i, col in enumerate(features):
  plt.subplot(2,3,i+1)
  sns.boxplot(data[col])
plt.show()
```




```
import plotly.express as px
fig=px.scatter_3d(data,x='name',y='status',z='PPE',
                  size='HNR',
                  color='D2')
fig.show()
```


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"hovertemplate":"name=%{x}<br>status=%{y}<br>PPE=%{z}<br>HNR=%
{marker.size}<br>D2=%{marker.color}<extra></
extra>","legendgroup":"","marker":{"color":
[2.301442,2.486855,2.342259,2.405554,2.33218,2.18756,1.854785,2.064693
,2.322511,2.432792,2.407313,2.642476,2.041277,2.519422,2.125618,2.2055
46,2.264501,3.007463,3.10901,2.856676,2.73971,2.557536,2.916777,2.5475
08,2.692176,2.846369,2.589702,2.314209,2.241742,1.957961,1.743867,2.10
```

```
3106,1.512275,1.544609,1.423287,2.447064,2.477082,2.536527,2.269398,2.
382544,2.374073,2.361532,2.416838,2.256699,2.330716,2.3658,2.392122,2.
028612,2.079922,2.054419,1.840198,2.431854,1.972297,2.223719,1.986899,
2.014606,1.92294,2.021591,1.827012,1.831691,2.460791,2.32156,2.278687,
2.498224,2.003032,2.118596,2.359973,2.291558,2.118496,2.137075,2.27792
7,2.642276,2.205024,1.928708,2.225815,1.862092,2.007923,1.777901,2.017
753,2.398422,2.645959,2.232576,2.428306,2.053601,3.099301,3.098256,2.6
54271,3.13655,3.007096,3.671155,3.317586,2.344876,2.344336,2.080121,2.
143851,2.344348,2.473239,2.671825,2.441612,2.634633,2.991063,2.638279,
2.690917,2.004055,2.065477,1.994387,2.129924,2.499148,2.296873,2.60874
9,2.550961,2.502336,2.376749,2.489191,2.938114,2.702355,2.640798,2.975
889,2.816781,2.925862,2.68624,2.655744,2.090438,2.174306,1.929715,1.76
5957,1.821297,1.996146,2.328513,2.108873,2.539724,2.527742,2.51632,2.0
34827,2.375138,2.631793,2.445502,2.672362,2.419253,2.445646,2.963799,2
.665133,2.465528,2.470746,2.576563,2.840556,3.413649,3.142364,3.274865
,2.910213,2.958815,3.079221,3.184027,2.01353,2.45113,2.439597,2.699645
,2.964568,2.8923,2.103014,2.151121,2.442906,2.408689,1.871871,2.560422
,2.235197,1.852402,1.881767,2.88245,2.266432,2.095237,2.193412,1.88900
2,1.852542,1.872946,1.974857,2.004719,2.449763,2.251553,2.845109,2.264
226,2.679185,2.209021,2.027228,2.120412,2.058658,2.161936,2.152083,1.9
1399,2.316346,2.657476,2.784312,2.679772,2.138608,2.555477],"coloraxis
":"coloraxis","size":
[21.033,19.085,20.651,20.644,19.649,21.378,24.886,26.892,21.812,21.862
,21.118,21.414,25.703,24.889,24.922,25.175,22.333,20.376,17.28,17.153,
17.536,19.493,22.468,20.422,23.831,22.066,25.908,25.119,25.97,25.678,2
6.775,30.94,30.775,32.684,33.047,31.732,23.216,24.951,26.738,26.31,26.
822,26.453,22.736,23.145,25.368,25.032,24.602,26.805,23.162,24.971,25.
135,25.03,24.692,25.429,21.028,20.767,21.422,22.817,22.603,21.66,25.55
4,26.138,25.856,25.964,26.415,24.547,19.56,19.979,20.338,21.718,20.264
,18.57,25.742,24.178,25.438,25.197,23.37,25.82,21.875,19.2,19.055,19.6
59,20.536,22.244,13.893,16.176,15.924,13.922,14.739,11.866,11.744,19.6
64,18.78,20.969,22.219,21.693,22.663,15.338,15.433,12.435,8.867,15.06,
10.489,26.759,28.409,27.421,29.746,26.833,29.928,21.934,23.239,22.407,
21.305,23.671,21.864,23.693,26.356,25.69,25.02,24.581,24.743,27.166,18
.305,18.784,19.196,18.857,18.178,18.33,26.842,26.369,23.949,26.017,23.
389,25.619,17.06,17.707,19.013,16.747,17.366,18.801,18.54,15.648,18.70
2,18.687,20.68,20.366,12.359,14.367,12.298,14.989,12.529,8.441,9.449,2
1.52,21.824,22.431,22.953,19.075,21.534,19.651,20.437,19.388,18.954,21
.219,18.447,24.078,24.679,21.083,19.269,21.02,21.528,26.436,26.55,26.5
47,25.445,26.005,26.143,24.151,24.412,23.683,23.133,22.866,23.008,23.0
79,22.085,24.199,23.958,25.023,24.775,19.368,19.517,19.147,17.883,19.0
2,21.209],"sizemode":"area","sizeref":8.26175e-
2,"symbol":"circle"},"mode":"markers","name":"","scene":"scene","showl
egend":false,"type":"scatter3d","x":
["phon_R01_S01_1","phon_R01_S01_2","phon_R01_S01_3","phon_R01_S01_4","
phon_R01_S01_5","phon_R01_S01_6","phon_R01_S02_1","phon_R01_S02_2","ph
on_R01_S02_3","phon_R01_S02_4","phon_R01_S02_5","phon_R01_S02_6","phon
_R01_S04_1","phon_R01_S04_2","phon_R01_S04_3","phon_R01_S04_4","phon_R
01_S04_5","phon_R01_S04_6","phon_R01_S05_1","phon_R01_S05_2","phon_R01
```

_S05_3","phon_R01_S05_4","phon_R01_S05_5","phon_R01_S05_6","phon_R01_S
06_1","phon_R01_S06_2","phon_R01_S06_3","phon_R01_S06_4","phon_R01_S06
_5","phon_R01_S06_6","phon_R01_S07_1","phon_R01_S07_2","phon_R01_S07_3
","phon_R01_S07_4","phon_R01_S07_5","phon_R01_S07_6","phon_R01_S08_1",
"phon_R01_S08_2","phon_R01_S08_3","phon_R01_S08_4","phon_R01_S08_5","p
hon_R01_S08_6","phon_R01_S10_1","phon_R01_S10_2","phon_R01_S10_3","pho
n_R01_S10_4","phon_R01_S10_5","phon_R01_S10_6","phon_R01_S13_1","phon_
R01_S13_2","phon_R01_S13_3","phon_R01_S13_4","phon_R01_S13_5","phon_R0
1_S13_6","phon_R01_S16_1","phon_R01_S16_2","phon_R01_S16_3","phon_R01_
S16_4","phon_R01_S16_5","phon_R01_S16_6","phon_R01_S17_1","phon_R01_S1
7_2","phon_R01_S17_3","phon_R01_S17_4","phon_R01_S17_5","phon_R01_S17_
6","phon_R01_S18_1","phon_R01_S18_2","phon_R01_S18_3","phon_R01_S18_4"
,"phon_R01_S18_5","phon_R01_S18_6","phon_R01_S19_1","phon_R01_S19_2","
phon_R01_S19_3","phon_R01_S19_4","phon_R01_S19_5","phon_R01_S19_6","ph
on_R01_S20_1","phon_R01_S20_2","phon_R01_S20_3","phon_R01_S20_4","phon
_R01_S20_5","phon_R01_S20_6","phon_R01_S21_1","phon_R01_S21_2","phon_R
01_S21_3","phon_R01_S21_4","phon_R01_S21_5","phon_R01_S21_6","phon_R01
_S21_7","phon_R01_S22_1","phon_R01_S22_2","phon_R01_S22_3","phon_R01_S
22_4","phon_R01_S22_5","phon_R01_S22_6","phon_R01_S24_1","phon_R01_S24
_2","phon_R01_S24_3","phon_R01_S24_4","phon_R01_S24_5","phon_R01_S24_6
","phon_R01_S25_1","phon_R01_S25_2","phon_R01_S25_3","phon_R01_S25_4",
"phon_R01_S25_5","phon_R01_S25_6","phon_R01_S26_1","phon_R01_S26_2","p
hon_R01_S26_3","phon_R01_S26_4","phon_R01_S26_5","phon_R01_S26_6","pho
n_R01_S27_1","phon_R01_S27_2","phon_R01_S27_3","phon_R01_S27_4","phon_
R01_S27_5","phon_R01_S27_6","phon_R01_S27_7","phon_R01_S31_1","phon_R0
1_S31_2","phon_R01_S31_3","phon_R01_S31_4","phon_R01_S31_5","phon_R01_
S31_6","phon_R01_S32_1","phon_R01_S32_2","phon_R01_S32_3","phon_R01_S3
2_4","phon_R01_S32_5","phon_R01_S32_6","phon_R01_S33_1","phon_R01_S33_
2","phon_R01_S33_3","phon_R01_S33_4","phon_R01_S33_5","phon_R01_S33_6"
,"phon_R01_S34_1","phon_R01_S34_2","phon_R01_S34_3","phon_R01_S34_4","
phon_R01_S34_5","phon_R01_S34_6","phon_R01_S35_1","phon_R01_S35_2","ph
on_R01_S35_3","phon_R01_S35_4","phon_R01_S35_5","phon_R01_S35_6","phon
_R01_S35_7","phon_R01_S37_1","phon_R01_S37_2","phon_R01_S37_3","phon_R
01_S37_4","phon_R01_S37_5","phon_R01_S37_6","phon_R01_S39_1","phon_R01
_S39_2","phon_R01_S39_3","phon_R01_S39_4","phon_R01_S39_5","phon_R01_S
39_6","phon_R01_S42_1","phon_R01_S42_2","phon_R01_S42_3","phon_R01_S42
_4","phon_R01_S42_5","phon_R01_S42_6","phon_R01_S43_1","phon_R01_S43_2
","phon_R01_S43_3","phon_R01_S43_4","phon_R01_S43_5","phon_R01_S43_6",
"phon_R01_S44_1","phon_R01_S44_2","phon_R01_S44_3","phon_R01_S44_4","p
hon_R01_S44_5","phon_R01_S44_6","phon_R01_S49_1","phon_R01_S49_2","pho
n_R01_S49_3","phon_R01_S49_4","phon_R01_S49_5","phon_R01_S49_6","phon_
R01_S50_1","phon_R01_S50_2","phon_R01_S50_3","phon_R01_S50_4","phon_R0
1_S50_5","phon_R01_S50_6"],"y":
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0
,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0],"z":

```
[0.284654,0.368674,0.332634,0.368975,0.410335,0.357775,0.211756,0.163755,0.231571,0.271362,0.24974,0.275931,0.138512,0.199889,0.1701,0.234589,0.218164,0.430788,0.377429,0.322111,0.365391,0.259765,0.285695,0.253556,0.215961,0.219514,0.147403,0.162999,0.108514,0.135242,8.5569e-2,6.8501e-2,9.632e-2,5.6141e-2,4.4539e-2,5.761e-2,0.165827,0.173218,0.141929,0.160691,0.130554,0.11573,9.5032e-2,0.117399,9.147e-2,0.102706,9.7336e-2,8.6398e-2,0.133867,0.128872,0.103561,0.105993,0.119308,0.147491,0.3167,0.344834,0.335041,0.314464,0.326197,0.316395,0.101516,9.8555e-2,0.103224,9.3534e-2,7.3581e-2,9.1546e-2,0.226156,0.226247,0.18558,0.141958,0.180828,0.242981,0.18818,0.225461,0.244512,0.228624,0.193918,0.232744,0.260015,0.277948,0.327978,0.260633,0.264666,0.177275,0.242119,0.200423,0.144614,0.220968,0.194052,0.332086,0.301952,0.13412,0.186489,0.160809,0.160812,0.164916,0.151709,0.340623,0.260375,0.378483,0.370961,0.356881,0.444774,0.113942,9.3193e-2,0.112878,0.106802,0.105306,0.11513,0.185668,0.23252,0.13639,0.268144,0.177807,0.115515,0.274407,0.170106,0.28278,0.251972,0.220657,0.152428,0.234809,0.229892,0.215558,0.181988,0.222716,0.214075,0.196535,0.112856,0.183572,0.169923,0.170633,0.232209,0.141422,0.24308,0.228319,0.259451,0.274387,0.209191,0.184985,0.277227,0.231723,0.209863,0.189032,0.159777,0.232861,0.457533,0.336085,0.418646,0.270173,0.301487,0.527367,0.454721,0.168581,0.247455,0.206256,0.220546,0.261305,0.249703,0.216638,0.244948,0.238281,0.22052,0.212386,0.367233,0.119652,9.1604e-2,7.5587e-2,0.202879,0.100881,9.622e-2,0.160376,0.174152,0.179677,0.163118,0.184067,0.174429,0.132703,0.160306,0.19273,0.144105,0.19771,0.156368,0.215724,0.252404,0.214346,0.120605,0.138868,0.121777,0.112838,0.13305,0.168895,0.131728,0.123306,0.148569]}],"layout":{"coloraxis":{"colorbar":{"title":
{"text":"D2"}},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"legend":
{"itemsizing":"constant","tracegroupgap":0},"margin":{"t":60},"scene":
{"domain":{"x":[0,1],"y":[0,1]},"xaxis":{"title":
{"text":"name"}},"yaxis":{"title":{"text":"status"}},"zaxis":{"title":
{"text":"PPE"}}},"template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpolar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"choropleth":[{"colorbar":
```

```
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
```

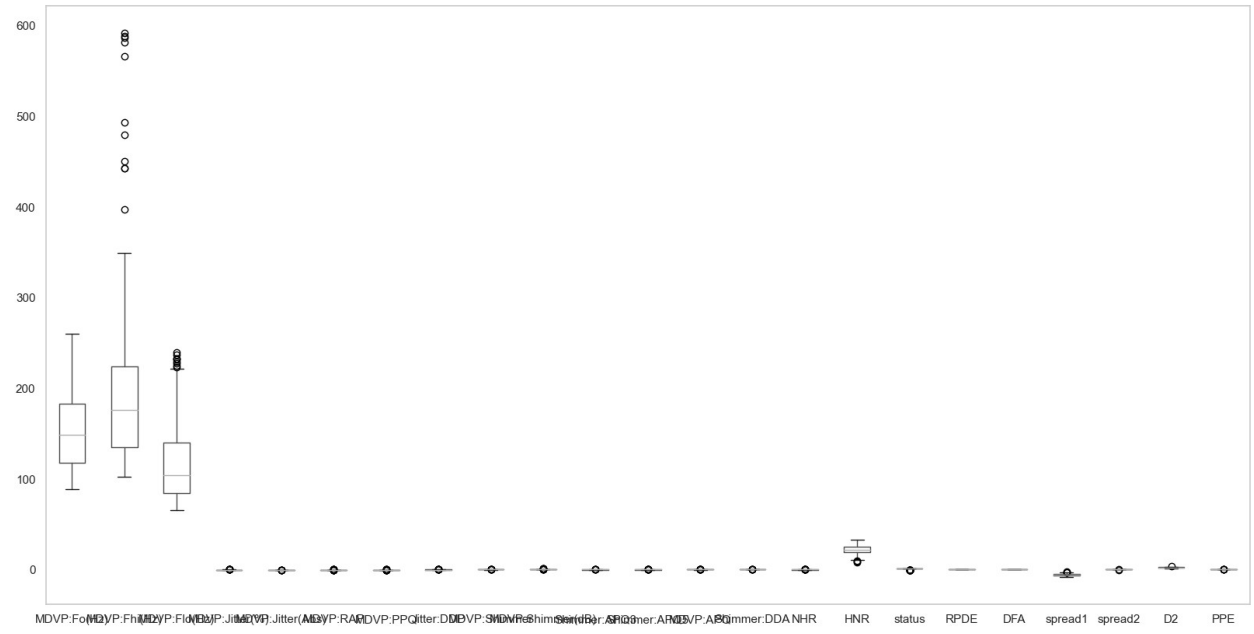
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":

{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"},"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"","title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"","title":{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}}}}

```
data.nunique().sort_values()
```

```
ColName
status                 2
MDVP:Jitter(Abs)      19
MDVP:Shimmer(dB)     149
MDVP:RAP             155
MDVP:PPQ             165
MDVP:Jitter(%)       173
Jitter:DDP           180
Shimmer:APQ3         184
NHR                  185
MDVP:Shimmer         188
Shimmer:APQ5         189
MDVP:APQ             189
Shimmer:DDA          189
spread2              194
spread1              195
RPDE                 195
DFA                  195
name                 195
D2                   195
MDVP:Flo(Hz)         195
MDVP:Fhi(Hz)         195
MDVP:Fo(Hz)          195
HNR                  195
PPE                  195
dtype: int64
```

```
plt.figure(figsize=(20,10))
data.boxplot(grid=False)
plt.show()
```




```
data.drop(columns=['name'],inplace=True,axis=1)

x = data.drop(columns = 'status')

# Getting Predicting Value
y = data['status']

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=0)

from sklearn.linear_model import LogisticRegression
reg = LogisticRegression()
reg.fit(x_train,y_train)

LogisticRegression()

y_pred=reg.predict(x_test)
from sklearn.metrics import
accuracy_score,classification_report,confusion_matrix,r2_score
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print("Training Score: ",reg.score(x_train,y_train)*100)
```

```
              precision    recall  f1-score   support

           0       0.88      0.70      0.78        10
           1       0.90      0.97      0.93        29

    accuracy                           0.90        39
   macro avg       0.89      0.83      0.86        39
weighted avg       0.90      0.90      0.89        39

[[ 7  3]
 [ 1 28]]
Training Score:  86.53846153846155
```

```
data = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
data.head()
```

```
     Actual  Predicted
83        1          1
12        1          1
33        0          0
113       1          1
171       0          1
```

```
dtree = DecisionTreeClassifier(max_depth=6, random_state=1)

dtree.fit(x_train,y_train)
```

```
DecisionTreeClassifier(max_depth=6, random_state=1)
```

```
y_pre=dtree.predict(x_test)
from sklearn.metrics import
classification_report,confusion_matrix,accuracy_score,mean_squared_err
or
print(classification_report(y_test,y_pre))
```

```
              precision    recall  f1-score   support

           0       0.67      0.80      0.73        10
           1       0.93      0.86      0.89        29

    accuracy                           0.85        39
   macro avg       0.80      0.83      0.81        39
weighted avg       0.86      0.85      0.85        39
```

```
print(confusion_matrix(y_test,y_pre))
```

```
[[ 8  2]
 [ 4 25]]
```

```
print("Training Score: ",dtree.score(x_train,y_train)*100)
```

```
Training Score:  100.0
```

```
print(accuracy_score(y_test,y_pre)*100)
```

```
84.61538461538461
```

```
data = pd.DataFrame({'Actual': y_test, 'Predicted': y_pre})
data.head()
```

```
     Actual  Predicted
83        1          1
12        1          1
33        0          0
113       1          1
171       0          1
```

```
data=pd.DataFrame({'Models':['LOGREG','DT'],
                   'Accuracy':[accuracy_score(y_test,y_pred)*100,
                               accuracy_score(y_test,y_pre)*100]})
data
```

```
   Models  Accuracy
0  LOGREG    89.744
1      DT    84.615
```

```
plt.figure(figsize=(3,3))
sns.barplot(data['Models'],data['Accuracy'])
plt.xticks(rotation=90)
plt.show()
```