

```
[5]: import os
import pandas as pd
os.makedirs(r"C:\Users\HP\Downloads\projects\voice",exist_ok=True)
df.get('id')
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

In [13]: os.listdir(r"C:\Users\HP\Downloads\voice.csv")
Out[13]: ['voice.csv']

In [14]: df=pd.read_csv('Downloads\voice.csv')

In [15]: df.shape
Out[15]: (3168, 21)

In [16]: df.columns
Out[16]: ['meanfreq', 'sd', 'median', 'Q25', 'Q75', 'IQR', 'skew', 'kurt', 'sp.ent', 'sfm', ..., 'centroid', 'meantun', 'minfun', 'maxfun', 'meandow', 'mindow', 'maxdow', 'dfrange', 'l', 'y', 'label']

In [17]: Index(['meanfreq', 'sd', 'median', 'Q25', 'Q75', 'IQR', 'skew', 'kurt', 'sp.ent', 'sfm', 'mode', 'centroid', 'meanfun', 'minfun', 'maxfun', 'meandow', 'mindow', 'maxdow', 'dfrange', 'l', 'y', 'label'],
dtype='object')

In [18]: df.dtypes
Out[18]: meanfreq    float64
sd            float64
median        float64
Q25           float64
Q75           float64
IQR           float64
skew          float64
kurt          float64
sp.ent        float64
sfm           float64
mode          float64
centroid      float64
meanfun       float64
minfun        float64
maxfun        float64
meandow       float64
mindow        float64
maxdow        float64
dfrange       float64
l             float64
y             object
label         object
dtype: object

In [19]: df.isnull().sum()
Out[19]: meanfreq    0
sd            0
median       0
Q25          0
Q75          0
IQR          0
skew         0
kurt         0
sp.ent       0
sfm          0
mode         0
centroid     0
meanfun      0
minfun       0
maxfun       0
meandow      0
mindow       0
maxdow       0
dfrange      0
l            0
y            0
label        0
dtype: object

In [20]: df.duplicated().sum()
Out[20]: 2

In [21]: df.drop_duplicates(inplace=True)
In [22]: df.duplicated().sum()
Out[22]: 0

In [23]: from sklearn.preprocessing import LabelEncoder

In [24]: le=LabelEncoder()
df['label']=le.fit_transform(df['label'])

In [25]: df.dtypes
Out[25]: meanfreq    float64
sd            float64
median        float64
Q25           float64
Q75           float64
IQR           float64
skew          float64
kurt          float64
sp.ent        float64
sfm           float64
mode          float64
centroid      float64
meanfun       float64
minfun        float64
maxfun        float64
meandow       float64
mindow        float64
maxdow        float64
dfrange       float64
l             float64
y             object
label         object
dtype: object

In [26]: for i in df.columns:
sns.boxplot(x=df[i])
plt.show()

In [27]: plt.figure(figsize=(10,10))
plt.show()

In [28]: df.describe(percentiles=[0.05,0.3,0.6,0.97])
Out[28]: count    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000
mean    0.029886    0.057130    0.185993    0.140440    0.224748    0.084308    3.140955    36.587139    0.895144    0.408348    ...    0.180888    0.142811    0.036801    0.258837
std     0.029898    0.010566    0.032054    0.040887    0.023636    0.042790    4.241768    134.969236    0.044889    0.171740    ...    0.029916    0.032314    0.019275    0.030085
min     0.009263    0.002850    0.000775    0.000000    0.000000    0.000000    0.014688    0.147755    2.068465    0.788561    ...    0.009693    0.005895    0.000775    0.000000
99%     0.235933    0.033166    0.116739    0.040871    0.087405    0.205486    1.123890    376.4775    0.816748    0.186478    ...    0.129323    0.099323    0.031796    0.102404
30%     0.156657    0.040504    0.170516    0.119812    0.212390    0.048423    1.745807    6.111152    0.871282    0.289529    ...    0.166467    0.121406    0.019356    0.258065
50%     0.184814    0.059162    0.180016    0.140255    0.225668    0.094209    2.197923    8.327893    0.901823    0.396472    ...    0.184814    0.140088    0.046110    0.271196
60%     0.190215    0.061142    0.197949    0.156401    0.232926    0.103688    2.441471    9.966827    0.912889    0.450361    ...    0.190215    0.155254    0.047198    0.275862
97%     0.233175    0.088103    0.197949    0.222027    0.259724    0.167038    15.921249    383.055102    0.966112    0.754317    ...    0.233175    0.198011    0.069264    0.279070
max     0.251231    0.115273    0.261224    0.247347    0.273469    0.252225    34.755463    1309.612887    0.981997    0.842936    ...    0.251231    0.237636    0.204882    0.279114

10 rows x 21 columns

In [29]: le=df['label'].quantile(0.25)
y=df['label'].quantile(0.75)
e=y-l
u=u+1.5*e

Out[30]: 25.6242825818888

In [31]: df['label']=np.where(df['kurt']>27.27,df['kurt'])

In [32]: sns.boxplot(x=df['kurt'])

In [33]: <AxesSubplot: xlabel='kurt'>

In [34]: df.describe()
Out[34]: count    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000
mean    0.029886    0.057130    0.185993    0.140440    0.224748    0.084308    3.140955    36.587139    0.895144    0.408348    ...    0.180888    0.142811    0.036801    0.258837
std     0.029898    0.010566    0.032054    0.040887    0.023636    0.042790    4.241768    134.969236    0.044889    0.171740    ...    0.029916    0.032314    0.019275    0.030085
min     0.009263    0.002850    0.000775    0.000000    0.000000    0.000000    0.014688    0.147755    2.068465    0.788561    ...    0.009693    0.005895    0.000775    0.000000
99%     0.235933    0.033166    0.116739    0.040871    0.087405    0.205486    1.123890    376.4775    0.816748    0.186478    ...    0.129323    0.099323    0.031796    0.102404
30%     0.156657    0.040504    0.170516    0.119812    0.212390    0.048423    1.745807    6.111152    0.871282    0.289529    ...    0.166467    0.121406    0.019356    0.258065
50%     0.184814    0.059162    0.180016    0.140255    0.225668    0.094209    2.197923    8.327893    0.901823    0.396472    ...    0.184814    0.140088    0.046110    0.271196
60%     0.190215    0.061142    0.197949    0.156401    0.232926    0.103688    2.441471    9.966827    0.912889    0.450361    ...    0.190215    0.155254    0.047198    0.275862
97%     0.233175    0.088103    0.197949    0.222027    0.259724    0.167038    15.921249    383.055102    0.966112    0.754317    ...    0.233175    0.198011    0.069264    0.279070
max     0.251231    0.115273    0.261224    0.247347    0.273469    0.252225    34.755463    1309.612887    0.981997    0.842936    ...    0.251231    0.237636    0.204882    0.279114

10 rows x 21 columns

In [35]: l=df['kurt'].quantile(0.25)
y=df['kurt'].quantile(0.75)
e=y-l
u=u+1.5*e

Out[36]: 25.6242825818888

In [37]: df['label']=np.where(df['kurt']>27.27,df['kurt'])

In [38]: sns.boxplot(x=df['kurt'])

In [39]: <AxesSubplot: xlabel='kurt'>

In [40]: df.describe()
Out[40]: count    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000
mean    0.029886    0.057130    0.185993    0.140440    0.224748    0.084308    3.140955    36.587139    0.895144    0.408348    ...    0.180888    0.142811    0.036801    0.258837
std     0.029898    0.010566    0.032054    0.040887    0.023636    0.042790    4.241768    134.969236    0.044889    0.171740    ...    0.029916    0.032314    0.019275    0.030085
min     0.009263    0.002850    0.000775    0.000000    0.000000    0.000000    0.014688    0.147755    2.068465    0.788561    ...    0.009693    0.005895    0.000775    0.000000
99%     0.235933    0.033166    0.116739    0.040871    0.087405    0.205486    1.123890    376.4775    0.816748    0.186478    ...    0.129323    0.099323    0.031796    0.102404
30%     0.156657    0.040504    0.170516    0.119812    0.212390    0.048423    1.745807    6.111152    0.871282    0.289529    ...    0.166467    0.121406    0.019356    0.258065
50%     0.184814    0.059162    0.180016    0.140255    0.225668    0.094209    2.197923    8.327893    0.901823    0.396472    ...    0.184814    0.140088    0.046110    0.271196
60%     0.190215    0.061142    0.197949    0.156401    0.232926    0.103688    2.441471    9.966827    0.912889    0.450361    ...    0.190215    0.155254    0.047198    0.275862
97%     0.233175    0.088103    0.197949    0.222027    0.259724    0.167038    15.921249    383.055102    0.966112    0.754317    ...    0.233175    0.198011    0.069264    0.279070
max     0.251231    0.115273    0.261224    0.247347    0.273469    0.252225    34.755463    1309.612887    0.981997    0.842936    ...    0.251231    0.237636    0.204882    0.279114

10 rows x 21 columns

In [41]: plt.figure(figsize=(10,8))
sns.heatmap(a,annot=True,cmap='RdBu')
plt.show()

In [42]: df.describe()
Out[42]: count    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000
mean    0.029886    0.057130    0.185993    0.140440    0.224748    0.084308    3.140955    36.587139    0.895144    0.408348    ...    0.180888    0.142811    0.036801    0.258837
std     0.029898    0.010566    0.032054    0.040887    0.023636    0.042790    4.241768    134.969236    0.044889    0.171740    ...    0.029916    0.032314    0.019275    0.030085
min     0.009263    0.002850    0.000775    0.000000    0.000000    0.000000    0.014688    0.147755    2.068465    0.788561    ...    0.009693    0.005895    0.000775    0.000000
99%     0.235933    0.033166    0.116739    0.040871    0.087405    0.205486    1.123890    376.4775    0.816748    0.186478    ...    0.129323    0.099323    0.031796    0.102404
30%     0.156657    0.040504    0.170516    0.119812    0.212390    0.048423    1.745807    6.111152    0.871282    0.289529    ...    0.166467    0.121406    0.019356    0.258065
50%     0.184814    0.059162    0.180016    0.140255    0.225668    0.094209    2.197923    8.327893    0.901823    0.396472    ...    0.184814    0.140088    0.046110    0.271196
60%     0.190215    0.061142    0.197949    0.156401    0.232926    0.103688    2.441471    9.966827    0.912889    0.450361    ...    0.190215    0.155254    0.047198    0.275862
97%     0.233175    0.088103    0.197949    0.222027    0.259724    0.167038    15.921249    383.055102    0.966112    0.754317    ...    0.233175    0.198011    0.069264    0.279070
max     0.251231    0.115273    0.261224    0.247347    0.273469    0.252225    34.755463    1309.612887    0.981997    0.842936    ...    0.251231    0.237636    0.204882    0.279114

10 rows x 21 columns

In [43]: l=df['kurt'].quantile(0.25)
y=df['kurt'].quantile(0.75)
e=y-l
u=u+1.5*e

Out[44]: 25.6242825818888

In [45]: df['label']=np.where(df['kurt']>27.27,df['kurt'])

In [46]: sns.boxplot(x=df['kurt'])

In [47]: <AxesSubplot: xlabel='kurt'>

In [48]: df.describe()
Out[48]: count    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000
mean    0.029886    0.057130    0.185993    0.140440    0.224748    0.084308    3.140955    36.587139    0.895144    0.408348    ...    0.180888    0.142811    0.036801    0.258837
std     0.029898    0.010566    0.032054    0.040887    0.023636    0.042790    4.241768    134.969236    0.044889    0.171740    ...    0.029916    0.032314    0.019275    0.030085
min     0.009263    0.002850    0.000775    0.000000    0.000000    0.000000    0.014688    0.147755    2.068465    0.788561    ...    0.009693    0.005895    0.000775    0.000000
99%     0.235933    0.033166    0.116739    0.040871    0.087405    0.205486    1.123890    376.4775    0.816748    0.186478    ...    0.129323    0.099323    0.031796    0.102404
30%     0.156657    0.040504    0.170516    0.119812    0.212390    0.048423    1.745807    6.111152    0.871282    0.289529    ...    0.166467    0.121406    0.019356    0.258065
50%     0.184814    0.059162    0.180016    0.140255    0.225668    0.094209    2.197923    8.327893    0.901823    0.396472    ...    0.184814    0.140088    0.046110    0.271196
60%     0.190215    0.061142    0.197949    0.156401    0.232926    0.103688    2.441471    9.966827    0.912889    0.450361    ...    0.190215    0.155254    0.047198    0.275862
97%     0.233175    0.088103    0.197949    0.222027    0.259724    0.167038    15.921249    383.055102    0.966112    0.754317    ...    0.233175    0.198011    0.069264    0.279070
max     0.251231    0.115273    0.261224    0.247347    0.273469    0.252225    34.755463    1309.612887    0.981997    0.842936    ...    0.251231    0.237636    0.204882    0.279114

10 rows x 21 columns

In [49]: plt.figure(figsize=(10,8))
sns.heatmap(a,annot=True,cmap='RdBu')
plt.show()

In [50]: df.describe()
Out[50]: count    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000
mean    0.029886    0.057130    0.185993    0.140440    0.224748    0.084308    3.140955    36.587139    0.895144    0.408348    ...    0.180888    0.142811    0.036801    0.258837
std     0.029898    0.010566    0.032054    0.040887    0.023636    0.042790    4.241768    134.969236    0.044889    0.171740    ...    0.029916    0.032314    0.019275    0.030085
min     0.009263    0.002850    0.000775    0.000000    0.000000    0.000000    0.014688    0.147755    2.068465    0.788561    ...    0.009693    0.005895    0.000775    0.000000
99%     0.235933    0.033166    0.116739    0.040871    0.087405    0.205486    1.123890    376.4775    0.816748    0.186478    ...    0.129323    0.099323    0.031796    0.102404
30%     0.156657    0.040504    0.170516    0.119812    0.212390    0.048423    1.745807    6.111152    0.871282    0.289529    ...    0.166467    0.121406    0.019356    0.258065
50%     0.184814    0.059162    0.180016    0.140255    0.225668    0.094209    2.197923    8.327893    0.901823    0.396472    ...    0.184814    0.140088    0.046110    0.271196
60%     0.190215    0.061142    0.197949    0.156401    0.232926    0.103688    2.441471    9.966827    0.912889    0.450361    ...    0.190215    0.155254    0.047198    0.275862
97%     0.233175    0.088103    0.197949    0.222027    0.259724    0.167038    15.921249    383.055102    0.966112    0.754317    ...    0.233175    0.198011    0.069264    0.279070
max     0.251231    0.115273    0.261224    0.247347    0.273469    0.252225    34.755463    1309.612887    0.981997    0.842936    ...    0.251231    0.237636    0.204882    0.279114

10 rows x 21 columns

In [51]: plt.figure(figsize=(10,8))
sns.heatmap(a,annot=True,cmap='RdBu')
plt.show()

In [52]: df.describe()
Out[52]: count    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000    3168.000000
mean    0.029886    0.057130    0.185993    0.140440    0.224748    0.084308    3.140955    36.587139    0.895144    0.408348    ...    0.180888    0.142811    0.036801    0.258837
std     0.029898    0.010566    0.032054    0.040887    0.023636    0.042790    4.241768    134.969236    0.044889    0.171740    ...    0.029916    0.032314    0.019275    0.030085
min     0.009263    0.002850    0.000775    0.000000    0.000000    0.000000    0.014688    0.147755    2.068465    0.788561    ...    0.009693    0.005895    0.000775    0.000000
99%     0.235933    0.033166    0.116739    0.040871    0.087405    0.205486    1.123890    376.4775    0.816748    0.186478    ...    0.129323    0.099323    0.031796    0.102404
30%     0.156657    0.040504    0.170516    0.119812    0.212390    0.048423    1.745807    6.111152    0.871282    0.289529    ...    0.166467    0.121406    0.019356    0.258065
50%     0.184814    0.059162    0.180016    0.140255    0.225668    0.094209    2.197923    8.327893    0.901823    0.396472    ...    0.184814    0.140088    0.046110    0.271196
60%     0.190215    0.061142    0.197949    0.156401    0.23
```