EDA

March 15, 2021

1 Gender Voice Recognition

Dane: https://www.apispreadsheets.com/datasets/119

1.1 Importy

[61]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[62]:
```
voice_df = pd.read_csv("gender_voice_dataset.csv")
```

1.2 Analiza wstępna

[63]:
```
voice_df.columns
```

[63]:
```
Index(['meanfreq', 'sd', 'median', 'Q25', 'Q75', 'IQR', 'skew', 'kurt',
       'sp.ent', 'sfm', 'mode', 'centroid', 'meanfun', 'minfun', 'maxfun',
       'meandom', 'mindom', 'maxdom', 'dfrange', 'modindx', 'label'],
      dtype='object')
```

[64]:
```
voice_df.head()
```

[64]:

	meanfreq	sd	median	Q25	…	maxdom	dfrange	modindx	label
0	0.059781	0.064241	0.032027	0.015071	…	0.007812	0.000000	0.000000	male
1	0.066009	0.067310	0.040229	0.019414	…	0.054688	0.046875	0.052632	male
2	0.077316	0.083829	0.036718	0.008701	…	0.015625	0.007812	0.046512	male
3	0.151228	0.072111	0.158011	0.096582	…	0.562500	0.554688	0.247119	male
4	0.135120	0.079146	0.124656	0.078720	…	5.484375	5.476562	0.208274	male

```
[5 rows x 21 columns]
```

[65]:
```
voice_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3168 entries, 0 to 3167
Data columns (total 21 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   meanfreq  3168 non-null   float64
 1   sd        3168 non-null   float64
 2   median    3168 non-null   float64
 3   Q25       3168 non-null   float64
 4   Q75       3168 non-null   float64
 5   IQR       3168 non-null   float64
 6   skew      3168 non-null   float64
 7   kurt      3168 non-null   float64
 8   sp.ent    3168 non-null   float64
 9   sfm       3168 non-null   float64
 10  mode      3168 non-null   float64
 11  centroid  3168 non-null   float64
 12  meanfun   3168 non-null   float64
 13  minfun    3168 non-null   float64
 14  maxfun    3168 non-null   float64
 15  meandom   3168 non-null   float64
 16  mindom    3168 non-null   float64
 17  maxdom    3168 non-null   float64
 18  dfrange   3168 non-null   float64
 19  modindx   3168 non-null   float64
 20  label     3168 non-null   object
dtypes: float64(20), object(1)
memory usage: 519.9+ KB
```

[66]:
```
voice_df.describe()
```

[66]:

	meanfreq	sd	…	dfrange	modindx
count	3168.000000	3168.000000	…	3168.000000	3168.000000
mean	0.180907	0.057126	…	4.994630	0.173752
std	0.029918	0.016652	…	3.520039	0.119454
min	0.039363	0.018363	…	0.000000	0.000000
25%	0.163662	0.041954	…	2.044922	0.099766
50%	0.184838	0.059155	…	4.945312	0.139357
75%	0.199146	0.067020	…	6.992188	0.209183
max	0.251124	0.115273	…	21.843750	0.932374

[8 rows x 20 columns]

[67]:
```
count_man=voice_df[voice_df['label']=='male'].shape[0]
print('Liczba labeli male:',count_man)
print('Liczba labeli female:',voice_df.shape[0]-count_man)
```

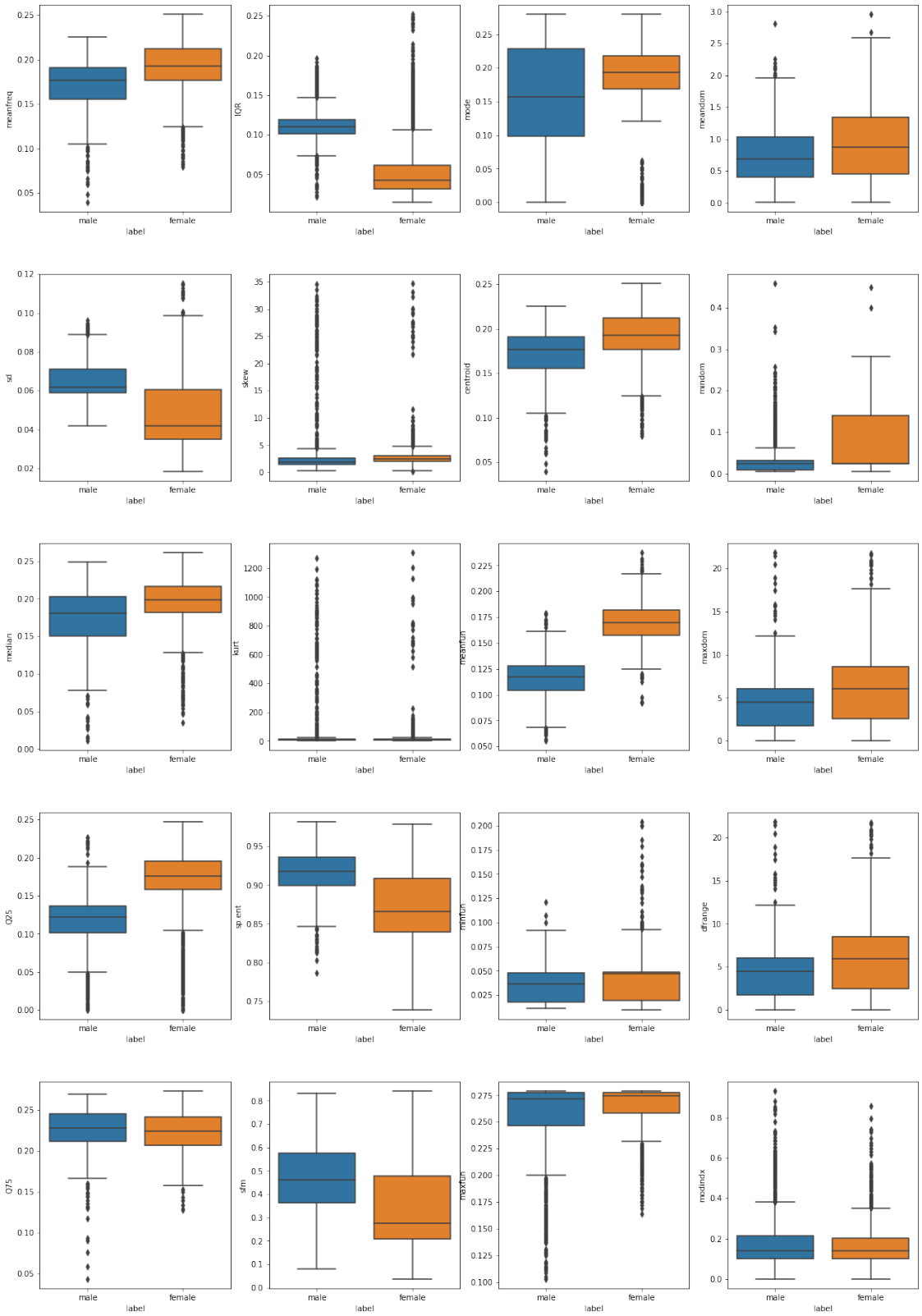
```
Liczba labeli male: 1584
Liczba labeli female: 1584
```

[68]:
```
voice_df.isnull().sum()
```

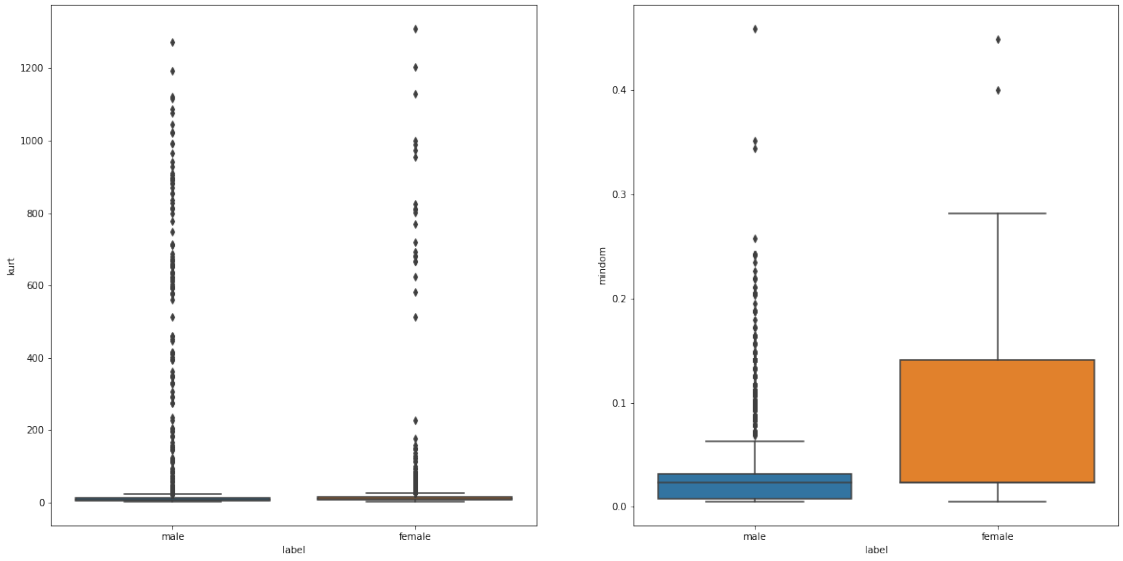
[68]:
```
meanfreq    0
sd          0
median      0
Q25         0
Q75         0
IQR         0
skew        0
kurt        0
sp.ent      0
sfm         0
mode        0
centroid    0
meanfun     0
minfun      0
maxfun      0
meandom     0
mindom      0
maxdom      0
dfrange     0
modindx     0
label       0
dtype: int64
```

1.3 Eksploracyjna Analiza Danych

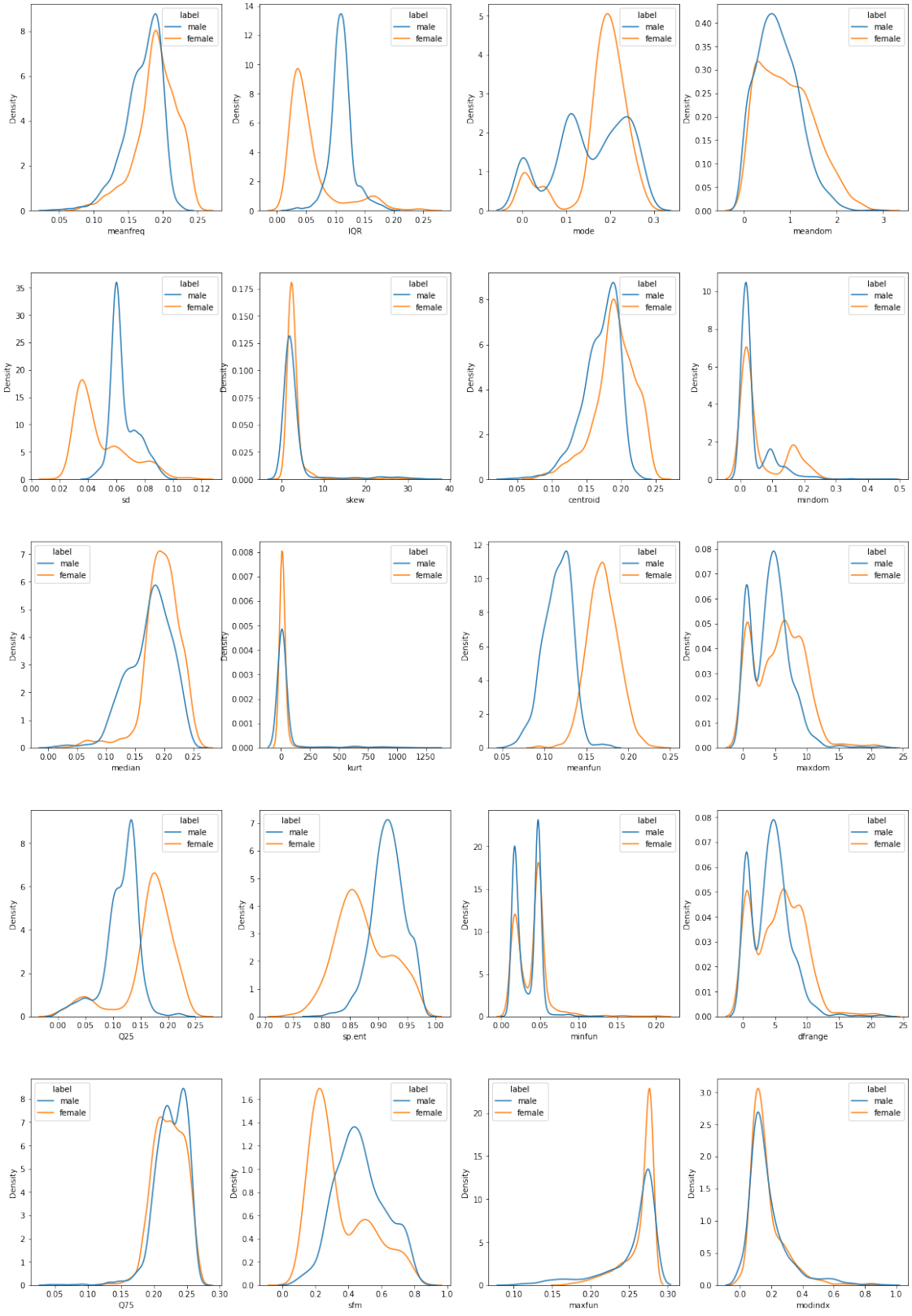
[69]:
```
fig, axs = plt.subplots(nrows = 5, ncols=4, figsize=(20,30))
plt.subplots_adjust(hspace=0.3)
i = 0
j = 0
for col in voice_df.columns:
  if col == "label":
    continue
  sns.boxplot(data=voice_df, y=col, x = "label", ax = axs[i][j])
  i+=1
  if i == 5:
    i = 0
    j += 1
```


meanfreq
IQR
mode
meandom
sd
skew
centroid
mindom
median
kurt
meanfun
maxdom
Q25
sp.ent
minfun
dfrange
Q75
sfm
maxfun
modindx
male
female
label

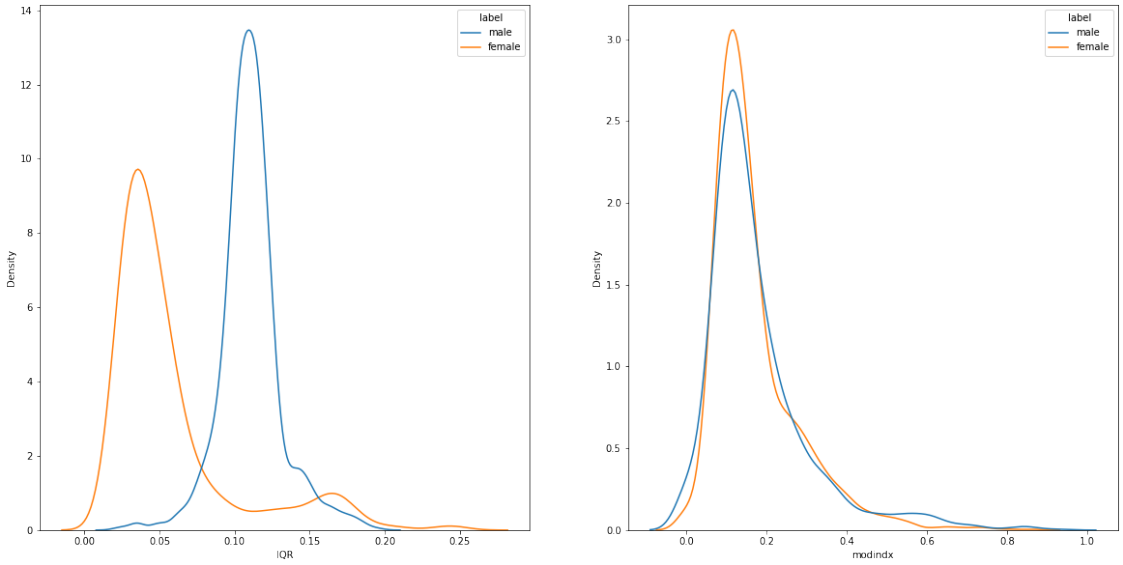
[70]:
```
fig, (ax1, ax2) = plt.subplots(nrows = 1, ncols=2, figsize=(20,10))
plt.subplots_adjust(hspace=0.3)
sns.boxplot(data=voice_df, y="kurt", x = "label", ax = ax1)
sns.boxplot(data=voice_df, y="mindom", x = "label", ax = ax2)
plt.show()
```



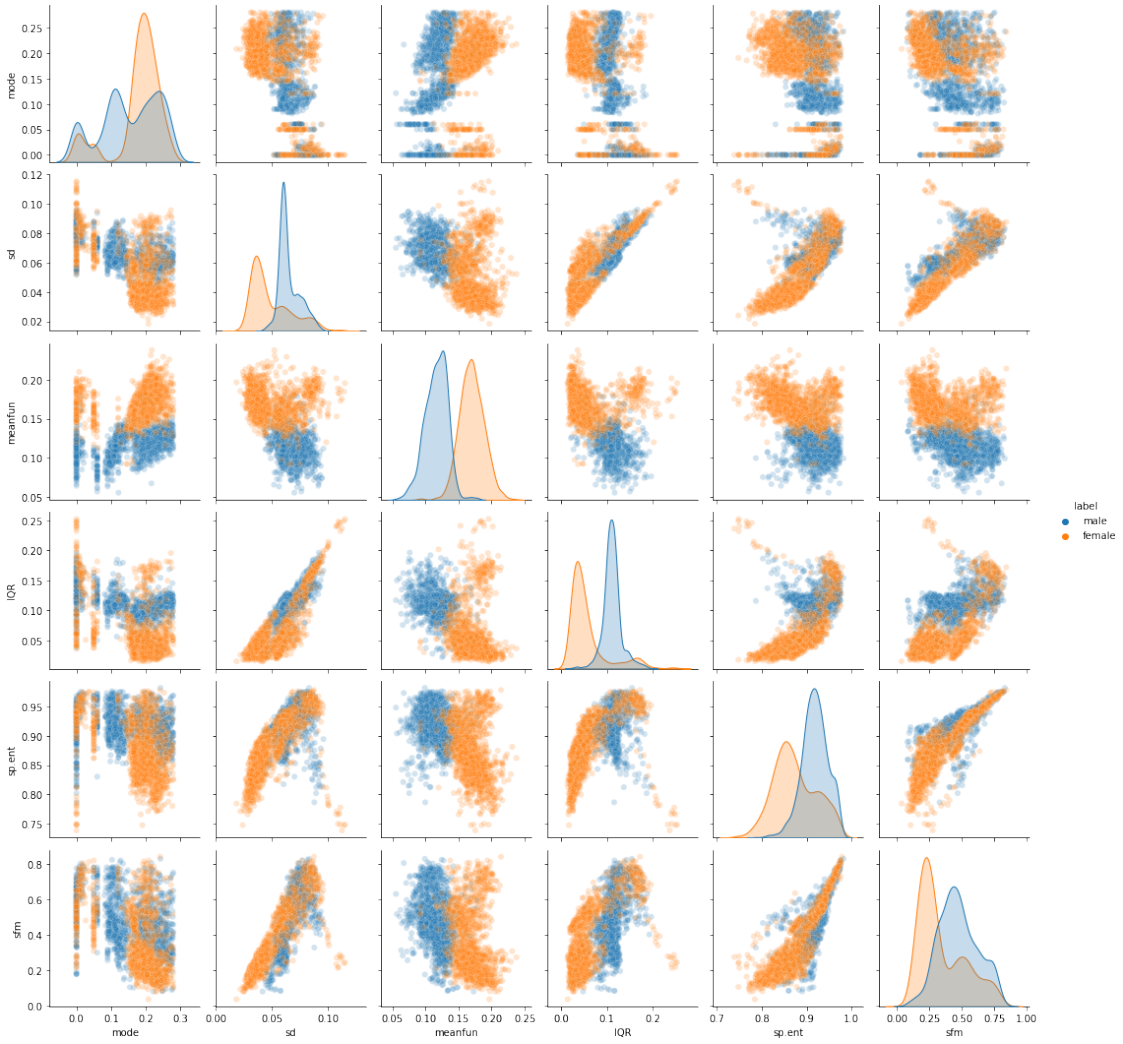
[71]:
```
fig, axs = plt.subplots(nrows = 5, ncols=4, figsize=(20,30))
plt.subplots_adjust(hspace=0.3)
i = 0
j = 0
for col in voice_df.columns:
  if col == "label":
    continue
  sns.kdeplot(data=voice_df, x=col, hue = "label", ax = axs[i][j])
  i+=1
  if i == 5:
    i = 0
    j += 1
```


label
male
female
Density
meanfreq
IQR
mode
meandom
sd
skew
centroid
mindom
median
kurt
meanfun
maxdom
Q25
sp.ent
minfun
dfrange
Q75
sfm
maxfun
modindx

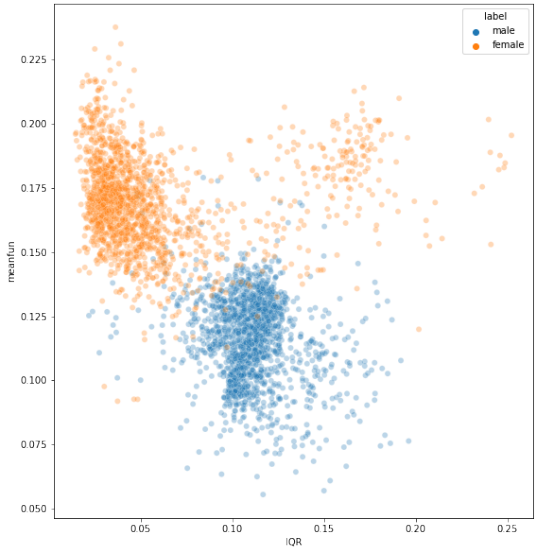
[72]:
```
fig, (ax1, ax2) = plt.subplots(nrows = 1, ncols=2, figsize=(20,10))
sns.kdeplot(data=voice_df, x="modindx", hue = "label", ax = ax2)
sns.kdeplot(data=voice_df, x="IQR", hue = "label", ax = ax1)
plt.show()
```

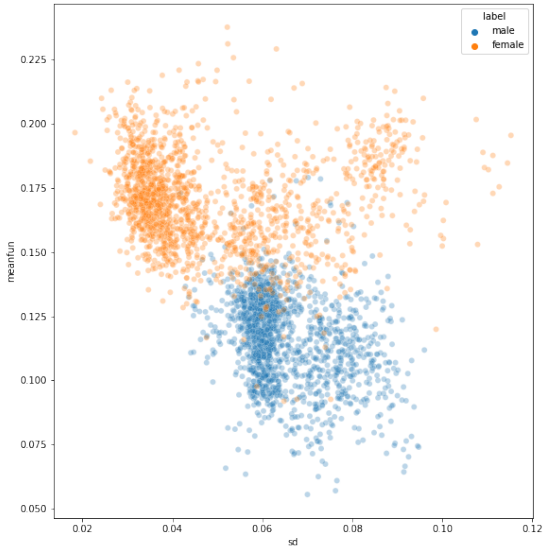


[55]:
```
sns.pairplot(data=voice_df, vars=["mode", "sd", "meanfun", "IQR", "sp.ent",␣
↪"sfm"], hue="label", plot_kws={"alpha":0.2})
plt.show()
```



[56]:
```
fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(20,10))
sns.scatterplot(data=voice_df, x = "IQR", y = "meanfun", hue = "label", alpha=0.
 ↪3, ax=ax1)
sns.scatterplot(data=voice_df, x = "sd", y = "meanfun", hue = "label", alpha=0.
 ↪3, ax=ax2)
plt.show()
```





1.4 Znaleziona prosta klasyfikacja

[57]:
```
def gender(row):
  if row["meanfun"] >= 0.14:
    return "female"
  else:
    if row["IQR"] >= 0.07:
      return "male"
    else:
      return "female"
```

[58]:
```
dt = voice_df[["meanfun", "IQR"]]
result = voice_df["label"]
```

[59]:
```
result_function = dt.apply((lambda row : gender(row)), axis = 1)
```

[60]:
```
c = pd.concat([result_function, result], axis = 1)
acc = 1-len(c.loc[c[0]!=c["label"]])/len(voice_df)
print(f"Accuracy is equal to {acc}")
```

```
Accuracy is equal to 0.9564393939393939
```

[60]:
```

```