data_analysis

November 6, 2017

```
In [30]: import os
         import glob
         import sox
         import tqdm
         import matplotlib
         import numpy as np
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         from datetime import timedelta
         from multiprocessing import Pool
         from sklearn.preprocessing import StandardScaler
         from sklearn.manifold import TSNE

         %matplotlib inline
```

datapath_root contains directories with separated male and female voice recordings. The audio files were preprocessed to eliminate silent regions. Each directory contains a number of recordings of an individual and a README file with some metadata. The metadata is not uniform in structure. It might miss information or outright lie.

```
In [2]: num_parallel = 16
        datapath_root = '/home/tracek/Data/gender/raw/'
        datapath_male = os.path.join(datapath_root, 'male/')
        datapath_female = os.path.join(datapath_root, 'female/')
```

Let's find all the recordings

```
In [3]: waves_male_paths = glob.glob(datapath_male + '/**/*.wav', recursive=True)
        waves_female_paths = glob.glob(datapath_female + '/**/*.wav', recursive=True)
        readme_paths = glob.glob(datapath_root + '/**/README', recursive=True)
```

Sanity check - do we always get a readme?

```
In [40]: assert len(os.listdir(datapath_male)) + len(os.listdir(datapath_female)) == len(readme_
```

I am curious how long are all recordings and whether there are no empty recordings, where the algorithm for trimming silence got overly enthusiastic

```
In [41]: def get_info(path):
             info = sox.file_info.info(path)
             info['path'] = path
             if 'num_samples' not in info:
                 print('No samples in ', path)
             return info
```

Let's do this in parallel. On my computer I am rarely IO bound - the beauty of ultra-fast NVMe drice (~3200 MB/s read)

```
In [6]: pool = Pool(processes=num_parallel)
        male_info = pool.map(get_info, waves_male_paths)
        female_info = pool.map(get_info, waves_female_paths)
```

For the future - check e.g. age range. Youth will have different vocal range

```
In [7]: def get_readme_info(path):
            d = {}
            with open(path, 'r') as readme:
                for line in readme:
                    gender_match = re.search("Gender: (\W*\w+\W*)", line, re.IGNORECASE)
                    age_match = re.search("Age Range: (\W*\w+\W*)", line, re.IGNORECASE)
                    lang_match = re.search("Language: (\W*\w+\W*)", line, re.IGNORECASE)
```

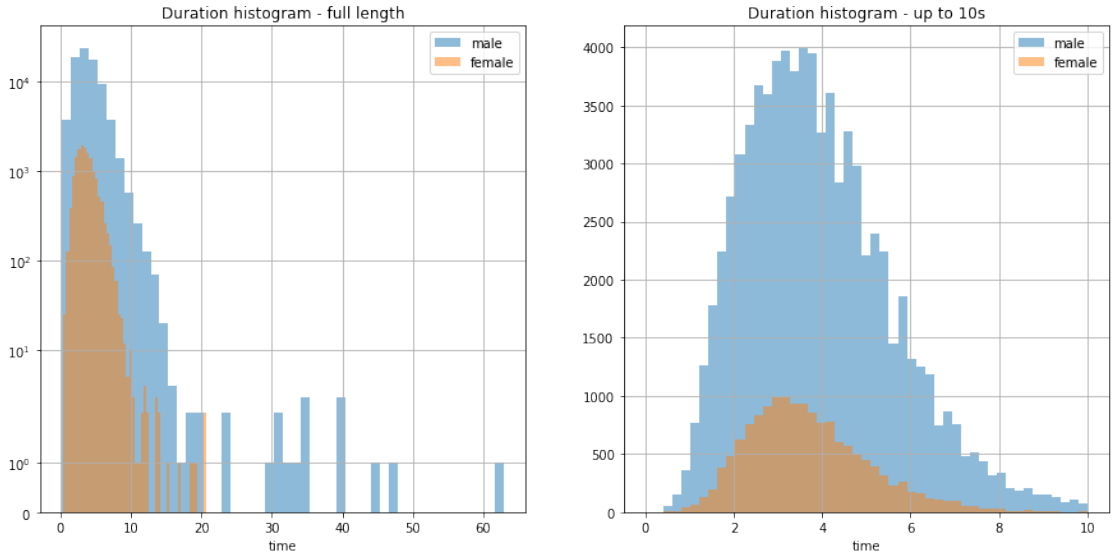
Clearly there more guys talking

```
In [8]: duration_male = np.array([info['duration'] for info in male_info])
        duration_female = np.array([info['duration'] for info in female_info])
        total_male = int(duration_male.sum())
        total_female = int(duration_female.sum())
        print('Total duration of male recordings: {} '.format(str(timedelta(seconds=total_male))
        print('Total duration of female recordings: {} '.format(str(timedelta(seconds=total_fema
```

```
Total duration of male recordings: 3 days, 15:03:21
Total duration of female recordings: 15:32:04
```

```
In [9]: no_bins = 50
        fig, ax = plt.subplots(1,2, figsize=(15,7))
        ax[0].set_yscale('symlog')
        _ = ax[0].hist(duration_male, bins=no_bins, alpha=0.5, label='male')
        _ = ax[0].hist(duration_female, bins=no_bins, alpha=0.5, label='female')
        bins = np.linspace(0, 10, no_bins)
        _ = ax[1].hist(duration_male, bins=bins, alpha=0.5, label='male')
        _ = ax[1].hist(duration_female, bins=bins, alpha=0.5, label='female')
        ax[0].legend(loc='upper right')
        ax[0].set_title('Duration histogram - full length')
        ax[0].set_xlabel('time')
        ax[0].grid(True)
```

```
ax[1].legend(loc='upper right')
ax[1].set_title('Duration histogram - up to 10s')
ax[1].set_xlabel('time')
ax[1].grid(True)
```




```
In [10]: name_duration_tuples_m = [(info['path'], info['duration']) for info in male_info]
         name_duration_tuples_m_short = [(info['path'], info['duration']) for info in male_info
```

Very few short recordings - good

```
In [11]: len(name_duration_tuples_m_short)
```

```
Out[11]: 12
```

Let's get some statistics

```
In [42]: datapath = '/home/tracek/Data/gender/gender_warbler.csv'
         data = pd.read_csv(datapath)
         male_df = data[data['label'] == 0]
         female_df = data[data['label'] == 1]
         print('Male recordings: ', len(male_df))
         print('Female recordings: ', len(female_df))
         pd.set_option('display.max_columns', len(male_df.columns.values))
         pd.set_option('display.max_rows', len(male_df))

         male_stats = male_df.describe()
         female_stats = female_df.describe()

         male_corr = male_df.corr()
         female_corr = female_df.corr()
```

```
Male recordings:  78820
Female recordings:  15066
```

```
In [13]: male_stats
```

Out[13]:

	meanfreq	sd	median	Q25	Q75
count	78820.000000	78820.000000	78820.000000	78820.000000	78820.000000
mean	0.158190	0.069968	0.155800	0.106229	0.217577
std	0.029525	0.011912	0.040531	0.040964	0.030279
min	0.000048	0.000924	0.000000	0.000000	0.000114
25%	0.142832	0.062430	0.129322	0.092997	0.205044
50%	0.159605	0.070046	0.152600	0.111946	0.222386
75%	0.177101	0.077571	0.183902	0.128866	0.237317
max	0.260308	0.113707	0.270406	0.259467	0.276782

	IQR	skew	kurt	sp.ent	sfm
count	78820.000000	78820.000000	78820.000000	78820.000000	78820.000000
mean	0.111348	5.315956	105.732651	0.913650	0.522978
std	0.031203	7.640572	265.642245	0.056788	0.150907
min	0.000114	0.096627	1.362474	0.081732	0.000029
25%	0.099816	1.817707	6.767020	0.899896	0.442717
50%	0.112615	2.460985	10.636244	0.922687	0.542471
75%	0.123661	3.532095	20.589989	0.940810	0.629068
max	0.258316	51.242799	2774.534469	0.988251	0.907603

	mode	centroid	meanfun	minfun	maxfun
count	78820.000000	78820.000000	78820.000000	78820.000000	78820.000000
mean	0.113875	0.158190	0.111827	0.024538	0.238085
std	0.082187	0.029525	0.021039	0.016960	0.040491
min	0.000000	0.000048	0.016000	0.015640	0.016000
25%	0.049931	0.142832	0.098321	0.016194	0.219178
50%	0.116667	0.159605	0.111734	0.017817	0.253968
75%	0.166339	0.177101	0.123941	0.024206	0.271186
max	0.280000	0.260308	0.249851	0.197531	0.275862

	meandom	mindom	maxdom	dfrange	modindx
count	78820.000000	78820.000000	78820.000000	78820.000000	78820.000000
mean	0.490626	0.046838	2.842126	2.795288	0.193379
std	0.423132	0.080493	2.165304	2.149576	0.108625
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.180060	0.000000	0.617188	0.585938	0.122613
50%	0.397569	0.000000	3.070312	3.015625	0.183710
75%	0.700120	0.085938	4.726562	4.664062	0.258026
max	4.855699	2.609375	6.992188	6.992188	1.000000

	label
count	78820.0

```
mean       0.0
std        0.0
min        0.0
25%        0.0
50%        0.0
75%        0.0
max        0.0
```

In [14]:
```
female_stats
```

Out[14]:

	meanfreq	sd	median	Q25	Q75	\
count	15066.000000	15066.000000	15066.000000	15066.000000	15066.000000	
mean	0.185398	0.051220	0.190744	0.158069	0.219492	
std	0.027135	0.016517	0.031762	0.044361	0.024374	
min	0.000050	0.001123	0.000000	0.000000	0.000121	
25%	0.177202	0.038919	0.181398	0.157535	0.204035	
50%	0.188914	0.047223	0.194186	0.170499	0.220898	
75%	0.202092	0.059854	0.208257	0.181184	0.236503	
max	0.252869	0.123865	0.270012	0.253907	0.276212	

	IQR	skew	kurt	sp.ent	sfm	\
count	15066.000000	15066.000000	15066.000000	15066.000000	15066.000000	
mean	0.061423	3.109131	34.942888	0.883406	0.360126	
std	0.039292	4.204126	138.466819	0.049327	0.144613	
min	0.000121	0.058957	1.706946	0.082314	0.000043	
25%	0.036971	1.711029	5.797064	0.861708	0.257301	
50%	0.050785	2.219656	8.370690	0.888449	0.320151	
75%	0.068883	2.912152	12.897301	0.911048	0.447784	
max	0.258962	46.825122	2308.549017	0.979994	0.855575	

	mode	centroid	meanfun	minfun	maxfun	\
count	15066.000000	15066.000000	15066.000000	15066.000000	15066.000000	
mean	0.176823	0.185398	0.166105	0.030376	0.256262	
std	0.064844	0.027135	0.022139	0.025143	0.021417	
min	0.000000	0.000050	0.041972	0.015640	0.086957	
25%	0.170521	0.177202	0.156878	0.016789	0.246154	
50%	0.188432	0.188914	0.169095	0.020000	0.262295	
75%	0.211336	0.202092	0.179428	0.031311	0.271186	
max	0.280000	0.252869	0.257717	0.225352	0.275862	

	meandom	mindom	maxdom	dfrange	modindx	\
count	15066.000000	15066.000000	15066.000000	15066.000000	15066.000000	
mean	0.667128	0.100262	3.885722	3.785460	0.200783	
std	0.410212	0.080838	2.275746	2.250809	0.091063	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.338033	0.000000	1.048828	1.015625	0.135274	
50%	0.618518	0.148438	4.835938	4.695312	0.191833	
75%	0.924594	0.164062	5.703125	5.585938	0.256827	

```
max        2.927557      1.210938      6.992188      6.992188      1.000000

              label
count  15066.0
mean       1.0
std        0.0
min        1.0
25%        1.0
50%        1.0
75%        1.0
max        1.0
```

We can already notice that, as suspected, acoustic parameters differ for males and females. Are these parameters correlated?

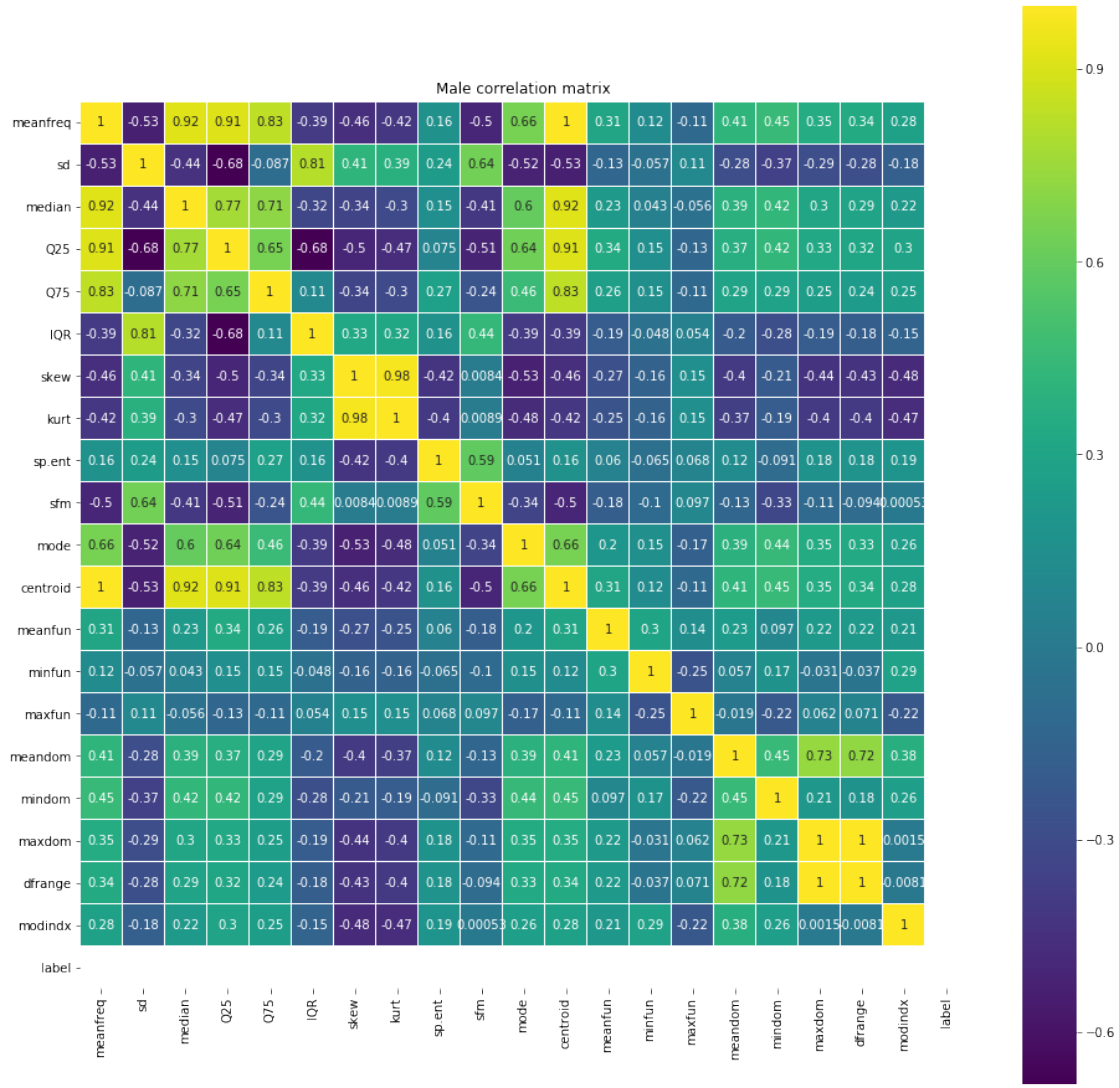
```
In [22]: plt.figure(figsize=(16, 16))
         plt.title('Male correlation matrix')

         sns.heatmap(male_corr,
                     xticklabels=male_corr.columns.values,
                     yticklabels=male_corr.columns.values,
                     linewidths=0.2,
                     vmax=1.0,
                     square=True,
                     cmap=plt.cm.viridis,
                     linecolor='white',
                     annot=True)

         male_corr
```

Out[22]:

	meanfreq	sd	median	Q25	Q75	IQR	\
meanfreq	1.000000	-0.531370	0.918285	0.908345	0.830313	-0.386770	
sd	-0.531370	1.000000	-0.441099	-0.682203	-0.086575	0.811592	
median	0.918285	-0.441099	1.000000	0.772453	0.711091	-0.324061	
Q25	0.908345	-0.682203	0.772453	1.000000	0.653534	-0.678638	
Q75	0.830313	-0.086575	0.711091	0.653534	1.000000	0.112408	
IQR	-0.386770	0.811592	-0.324061	-0.678638	0.112408	1.000000	
skew	-0.457958	0.410973	-0.342813	-0.499486	-0.337403	0.328323	
kurt	-0.417237	0.388134	-0.301344	-0.469299	-0.303856	0.321247	
sp.ent	0.159927	0.243810	0.147395	0.075059	0.265974	0.159556	
sfm	-0.499641	0.642498	-0.407684	-0.512691	-0.244582	0.435730	
mode	0.655460	-0.522495	0.601589	0.635642	0.456325	-0.391672	
centroid	1.000000	-0.531370	0.918285	0.908345	0.830313	-0.386770	
meanfun	0.309700	-0.127773	0.225871	0.337124	0.260794	-0.189511	
minfun	0.120628	-0.057107	0.042936	0.150121	0.154114	-0.047532	
maxfun	-0.113266	0.111296	-0.055700	-0.125235	-0.114132	0.053659	
meandom	0.414225	-0.284795	0.394752	0.371944	0.292881	-0.204088	
mindom	0.449952	-0.365445	0.423723	0.423727	0.287644	-0.277151	

```
maxdom    0.350724 -0.286867  0.304268  0.328561  0.246390 -0.192247
dfrange   0.336441 -0.275282  0.290627  0.315098  0.237422 -0.183275
modindx   0.281725 -0.177614  0.219111  0.296324  0.248244 -0.148128
label          NaN       NaN       NaN       NaN       NaN       NaN

              skew      kurt    sp.ent       sfm      mode  centroid  \
meanfreq -0.457958 -0.417237  0.159927 -0.499641  0.655460  1.000000
sd        0.410973  0.388134  0.243810  0.642498 -0.522495 -0.531370
median   -0.342813 -0.301344  0.147395 -0.407684  0.601589  0.918285
Q25      -0.499486 -0.469299  0.075059 -0.512691  0.635642  0.908345
Q75      -0.337403 -0.303856  0.265974 -0.244582  0.456325  0.830313
IQR       0.328323  0.321247  0.159556  0.435730 -0.391672 -0.386770
skew      1.000000  0.978248 -0.423898  0.008372 -0.526812 -0.457958
kurt      0.978248  1.000000 -0.397813  0.008904 -0.476433 -0.417237
sp.ent   -0.423898 -0.397813  1.000000  0.590999  0.050762  0.159927
sfm       0.008372  0.008904  0.590999  1.000000 -0.341765 -0.499641
mode     -0.526812 -0.476433  0.050762 -0.341765  1.000000  0.655460
centroid -0.457958 -0.417237  0.159927 -0.499641  0.655460  1.000000
meanfun  -0.272721 -0.251485  0.060196 -0.183386  0.201729  0.309700
minfun   -0.158513 -0.157671 -0.065375 -0.100184  0.153088  0.120628
maxfun    0.146785  0.150416  0.068242  0.096793 -0.170067 -0.113266
meandom  -0.397926 -0.367938  0.121628 -0.132090  0.389690  0.414225
mindom   -0.208972 -0.194626 -0.091384 -0.327100  0.443671  0.449952
maxdom   -0.438794 -0.403165  0.177869 -0.105780  0.347388  0.350724
dfrange  -0.434180 -0.398827  0.182592 -0.094306  0.333316  0.336441
modindx  -0.484166 -0.471864  0.186912  0.000530  0.256782  0.281725
label          NaN       NaN       NaN       NaN       NaN       NaN

             meanfun    minfun    maxfun   meandom    mindom    maxdom  \
meanfreq    0.309700  0.120628 -0.113266  0.414225  0.449952  0.350724
sd         -0.127773 -0.057107  0.111296 -0.284795 -0.365445 -0.286867
median      0.225871  0.042936 -0.055700  0.394752  0.423723  0.304268
Q25         0.337124  0.150121 -0.125235  0.371944  0.423727  0.328561
Q75         0.260794  0.154114 -0.114132  0.292881  0.287644  0.246390
IQR        -0.189511 -0.047532  0.053659 -0.204088 -0.277151 -0.192247
skew       -0.272721 -0.158513  0.146785 -0.397926 -0.208972 -0.438794
kurt       -0.251485 -0.157671  0.150416 -0.367938 -0.194626 -0.403165
sp.ent      0.060196 -0.065375  0.068242  0.121628 -0.091384  0.177869
sfm        -0.183386 -0.100184  0.096793 -0.132090 -0.327100 -0.105780
mode        0.201729  0.153088 -0.170067  0.389690  0.443671  0.347388
centroid    0.309700  0.120628 -0.113266  0.414225  0.449952  0.350724
meanfun     1.000000  0.302264  0.143757  0.233844  0.096639  0.224012
minfun      0.302264  1.000000 -0.254629  0.057173  0.174438 -0.030728
maxfun      0.143757 -0.254629  1.000000 -0.019064 -0.218438  0.062438
meandom     0.233844  0.057173 -0.019064  1.000000  0.450313  0.731991
mindom      0.096639  0.174438 -0.218438  0.450313  1.000000  0.213274
maxdom      0.224012 -0.030728  0.062438  0.731991  0.213274  1.000000
dfrange     0.222032 -0.037485  0.071074  0.720485  0.177389  0.999331
```

```
modindx   0.212940  0.287109 -0.224046  0.381278  0.257251  0.001493
label          NaN       NaN       NaN       NaN       NaN       NaN

           dfrange   modindx  label
meanfreq  0.336441  0.281725    NaN
sd       -0.275282 -0.177614    NaN
median    0.290627  0.219111    NaN
Q25       0.315098  0.296324    NaN
Q75       0.237422  0.248244    NaN
IQR      -0.183275 -0.148128    NaN
skew     -0.434180 -0.484166    NaN
kurt     -0.398827 -0.471864    NaN
sp.ent    0.182592  0.186912    NaN
sfm      -0.094306  0.000530    NaN
mode      0.333316  0.256782    NaN
centroid  0.336441  0.281725    NaN
meanfun   0.222032  0.212940    NaN
minfun   -0.037485  0.287109    NaN
maxfun    0.071074 -0.224046    NaN
meandom   0.720485  0.381278    NaN
mindom    0.177389  0.257251    NaN
maxdom    0.999331  0.001493    NaN
dfrange   1.000000 -0.008129    NaN
modindx  -0.008129  1.000000    NaN
label          NaN       NaN    NaN
```



In [23]:
```
plt.figure(figsize=(16, 16))
plt.title('Female correlation matrix')

sns.heatmap(female_corr,
            xticklabels=female_corr.columns.values,
            yticklabels=female_corr.columns.values,
            linewidths=0.2,
            vmax=1.0,
            square=True,
            cmap=plt.cm.viridis,
            linecolor='white',
            annot=True)

female_corr
```

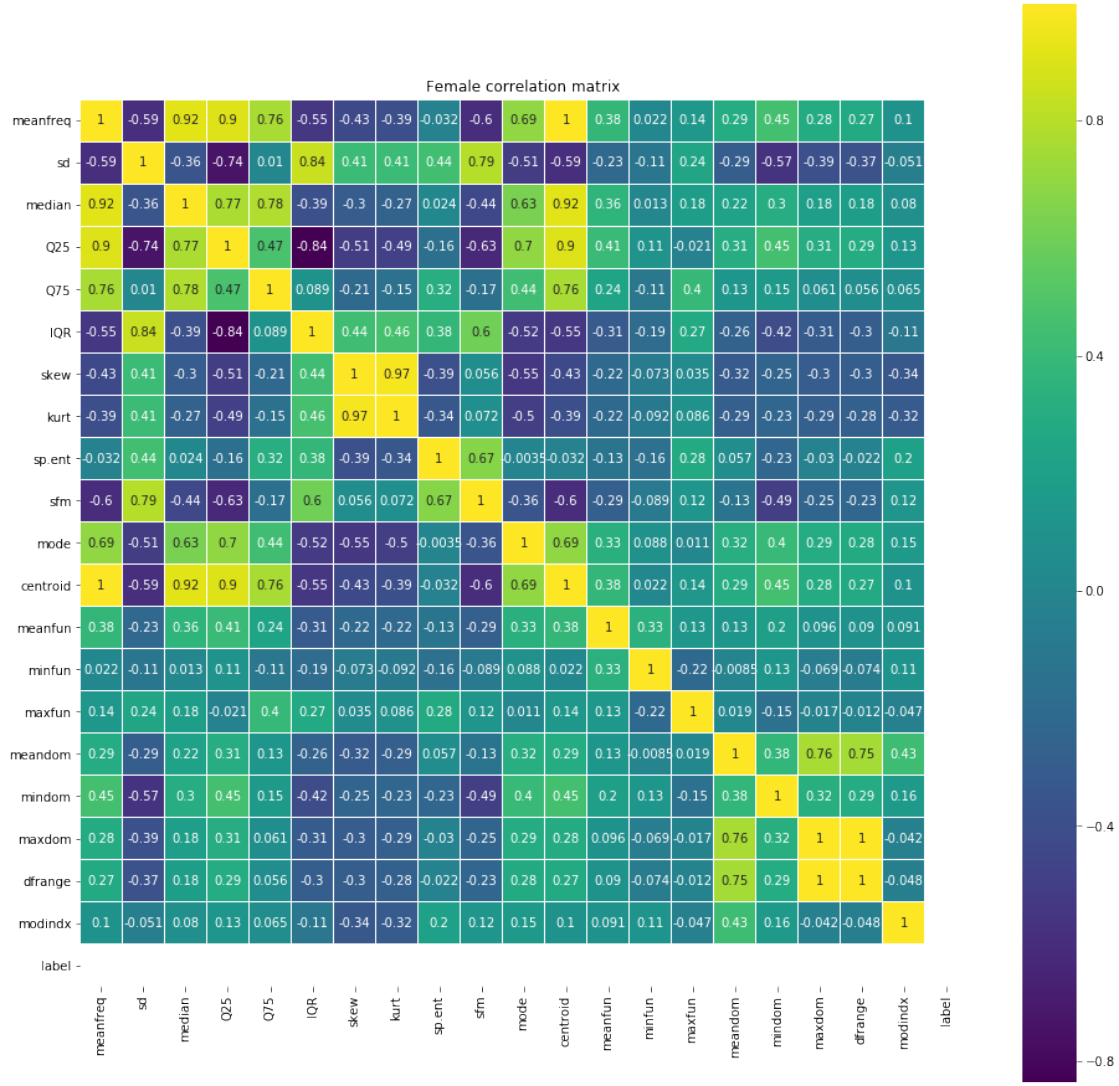
Out[23]:

	meanfreq	sd	median	Q25	Q75	IQR
meanfreq	1.000000	-0.591699	0.918497	0.902134	0.756762	-0.549060
sd	-0.591699	1.000000	-0.360005	-0.741397	0.010176	0.843342
median	0.918497	-0.360005	1.000000	0.769501	0.779386	-0.385284
Q25	0.902134	-0.741397	0.769501	1.000000	0.470786	-0.836948
Q75	0.756762	0.010176	0.779386	0.470786	1.000000	0.088815
IQR	-0.549060	0.843342	-0.385284	-0.836948	0.088815	1.000000
skew	-0.432546	0.413481	-0.302692	-0.507232	-0.213375	0.440297
kurt	-0.393026	0.414254	-0.265863	-0.490065	-0.150340	0.460018
sp.ent	-0.031598	0.444014	0.023799	-0.163741	0.316771	0.381363
sfm	-0.603696	0.791996	-0.438997	-0.626415	-0.169233	0.602237
mode	0.692733	-0.508388	0.628579	0.696789	0.437554	-0.515241
centroid	1.000000	-0.591699	0.918497	0.902134	0.756762	-0.549060
meanfun	0.377440	-0.226911	0.357200	0.407962	0.237149	-0.313474
minfun	0.022280	-0.114754	0.012623	0.109540	-0.109853	-0.191814
maxfun	0.139019	0.241278	0.176223	-0.021341	0.396178	0.269854
meandom	0.292911	-0.292050	0.220147	0.305391	0.129861	-0.264227
mindom	0.449077	-0.569494	0.304623	0.453366	0.154398	-0.416068
maxdom	0.282852	-0.390213	0.184843	0.307263	0.061192	-0.308938
dfrange	0.269858	-0.374083	0.175950	0.294385	0.056324	-0.297418
modindx	0.102016	-0.050739	0.080306	0.131133	0.064652	-0.107942
label	NaN	NaN	NaN	NaN	NaN	NaN

	skew	kurt	sp.ent	sfm	mode	centroid
meanfreq	-0.432546	-0.393026	-0.031598	-0.603696	0.692733	1.000000
sd	0.413481	0.414254	0.444014	0.791996	-0.508388	-0.591699
median	-0.302692	-0.265863	0.023799	-0.438997	0.628579	0.918497
Q25	-0.507232	-0.490065	-0.163741	-0.626415	0.696789	0.902134
Q75	-0.213375	-0.150340	0.316771	-0.169233	0.437554	0.756762
IQR	0.440297	0.460018	0.381363	0.602237	-0.515241	-0.549060
skew	1.000000	0.973781	-0.389195	0.055770	-0.546284	-0.432546
kurt	0.973781	1.000000	-0.337551	0.071647	-0.499239	-0.393026
sp.ent	-0.389195	-0.337551	1.000000	0.669291	-0.003498	-0.031598
sfm	0.055770	0.071647	0.669291	1.000000	-0.357639	-0.603696
mode	-0.546284	-0.499239	-0.003498	-0.357639	1.000000	0.692733
centroid	-0.432546	-0.393026	-0.031598	-0.603696	0.692733	1.000000
meanfun	-0.217733	-0.218243	-0.127057	-0.288747	0.327645	0.377440
minfun	-0.072983	-0.091520	-0.161566	-0.088524	0.087710	0.022280
maxfun	0.035086	0.085665	0.279201	0.121086	0.011486	0.139019
meandom	-0.316585	-0.285158	0.056906	-0.129590	0.316084	0.292911
mindom	-0.246191	-0.228461	-0.229515	-0.493119	0.400851	0.449077
maxdom	-0.300883	-0.286103	-0.029763	-0.245365	0.290727	0.282852
dfrange	-0.295374	-0.281068	-0.021849	-0.230373	0.279551	0.269858
modindx	-0.341491	-0.319344	0.201129	0.121188	0.153887	0.102016
label	NaN	NaN	NaN	NaN	NaN	NaN

	meanfun	minfun	maxfun	meandom	mindom	maxdom
meanfreq	0.377440	0.022280	0.139019	0.292911	0.449077	0.282852

```
sd       -0.226911 -0.114754  0.241278 -0.292050 -0.569494 -0.390213
median    0.357200  0.012623  0.176223  0.220147  0.304623  0.184843
Q25       0.407962  0.109540 -0.021341  0.305391  0.453366  0.307263
Q75       0.237149 -0.109853  0.396178  0.129861  0.154398  0.061192
IQR      -0.313474 -0.191814  0.269854 -0.264227 -0.416068 -0.308938
skew     -0.217733 -0.072983  0.035086 -0.316585 -0.246191 -0.300883
kurt     -0.218243 -0.091520  0.085665 -0.285158 -0.228461 -0.286103
sp.ent   -0.127057 -0.161566  0.279201  0.056906 -0.229515 -0.029763
sfm      -0.288747 -0.088524  0.121086 -0.129590 -0.493119 -0.245365
mode      0.327645  0.087710  0.011486  0.316084  0.400851  0.290727
centroid  0.377440  0.022280  0.139019  0.292911  0.449077  0.282852
meanfun   1.000000  0.332927  0.134741  0.127876  0.196230  0.096437
minfun    0.332927  1.000000 -0.216163 -0.008518  0.125697 -0.068934
maxfun    0.134741 -0.216163  1.000000  0.019060 -0.145003 -0.017305
meandom   0.127876 -0.008518  0.019060  1.000000  0.375460  0.759007
mindom    0.196230  0.125697 -0.145003  0.375460  1.000000  0.324557
maxdom    0.096437 -0.068934 -0.017305  0.759007  0.324557  1.000000
dfrange   0.090457 -0.074212 -0.012289  0.753931  0.292238  0.999423
modindx   0.091181  0.112750 -0.046971  0.427506  0.158740 -0.042154
label          NaN       NaN       NaN       NaN       NaN       NaN

           dfrange   modindx  label
meanfreq  0.269858  0.102016    NaN
sd       -0.374083 -0.050739    NaN
median    0.175950  0.080306    NaN
Q25       0.294385  0.131133    NaN
Q75       0.056324  0.064652    NaN
IQR      -0.297418 -0.107942    NaN
skew     -0.295374 -0.341491    NaN
kurt     -0.281068 -0.319344    NaN
sp.ent   -0.021849  0.201129    NaN
sfm      -0.230373  0.121188    NaN
mode      0.279551  0.153887    NaN
centroid  0.269858  0.102016    NaN
meanfun   0.090457  0.091181    NaN
minfun   -0.074212  0.112750    NaN
maxfun   -0.012289 -0.046971    NaN
meandom   0.753931  0.427506    NaN
mindom    0.292238  0.158740    NaN
maxdom    0.999423 -0.042154    NaN
dfrange   1.000000 -0.048323    NaN
modindx  -0.048323  1.000000    NaN
label          NaN       NaN    NaN
```



Indeed they are! In fact we can already drop some of them as they are identical. We can safely remove dfrange, *range of dominant frequency measured across the acoustic signal*, as it corresponds to maxdom. Why? *dfrange* is simply difference between *maxdom* and *mindom* - and *mindom* happens always to be zero. Mean frequency (*meanfreq*) seems to be an idiom to *centriod*.

Before we move further, let's make sure data types are OK

```
In [24]: data.dtypes

Out[24]: filename     object
         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
meandom     float64
mindom      float64
maxdom      float64
dfrange     float64
modindx     float64
label         int64
dtype: object
```

We already know we can drop some features. *Filename* is also of little use. Let's rehearse what features are at our disposal:

- meanfreq: mean frequency (in kHz)
- sd: standard deviation of frequency
- median: median frequency (in kHz)
- Q25: first quantile (in kHz)
- Q75: third quantile (in kHz)
- IQR: interquantile range (in kHz)
- skew: skewness
- kurt: kurtosis
- sp.ent: spectral entropy
- sfm: spectral flatness
- mode: mode frequency
- meanfun: average of fundamental frequency measured across acoustic signal
- minfun: minimum fundamental frequency measured across acoustic signal
- maxfun: maximum fundamental frequency measured across acoustic signal
- meandom: average of dominant frequency measured across acoustic signal
- mindom: minimum of dominant frequency measured across acoustic signal
- maxdom: maximum of dominant frequency measured across acoustic signal

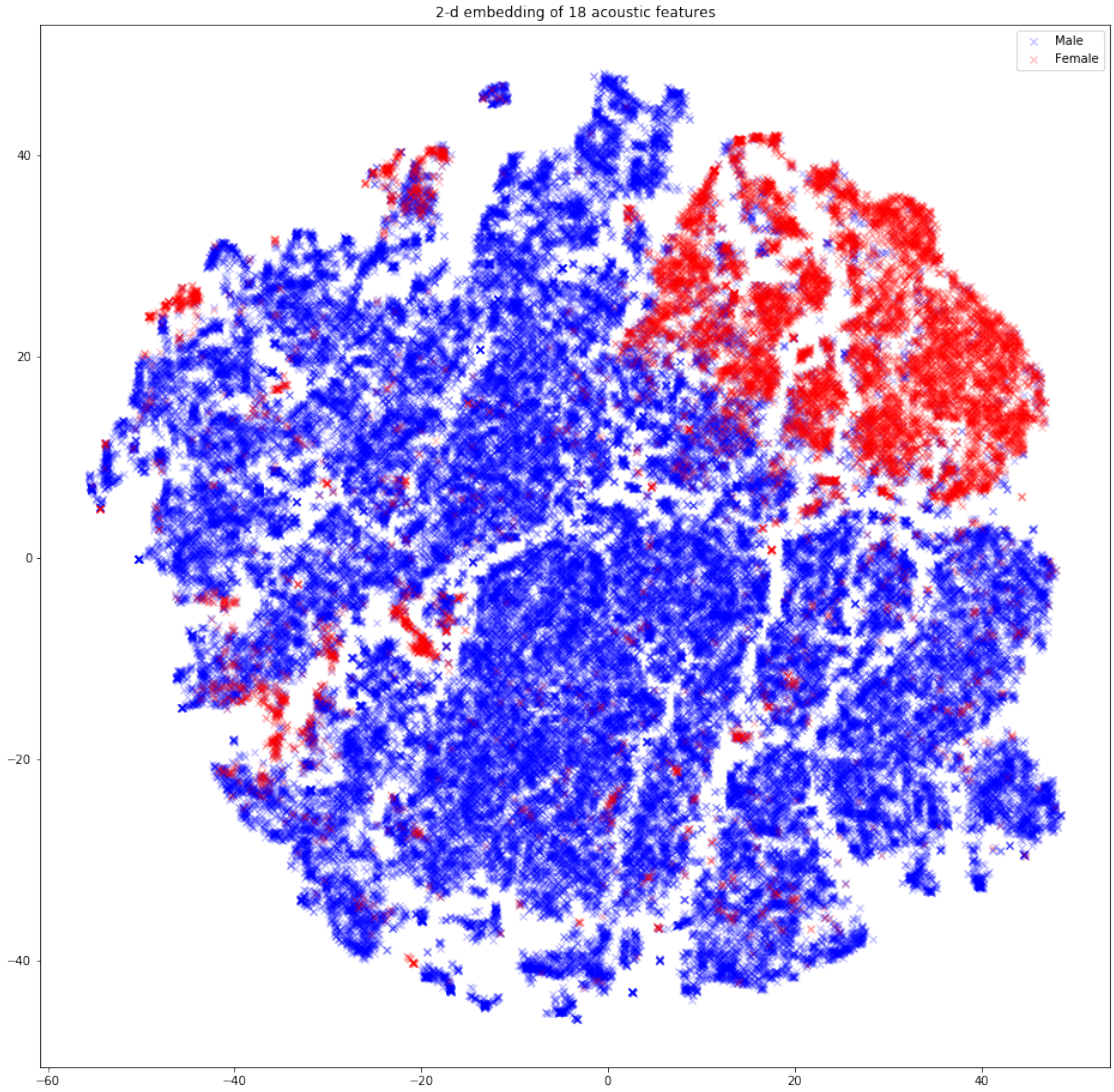
```
In [28]: y = data.pop('label')
         data = data.drop(['centroid', 'dfrange', 'filename'], axis=1)
         X = StandardScaler().fit_transform(data)
```

With the collected features, is it possible at tell the gender apart? Based on common experience - yes. After all, we can usually distinguish gender by voice, and we do this by interpreting acoustic properties of the signal. Female voice sounds higher. Let's see if we can somehow check it.

We have 18 features, meaning 18-dimensional space. As 3-d creatures, we barely manage with our imagination in 3-d and 2-d is by far preferred. One of the most accomplished methods for 2-d embedding of high-dimensional spaces is t-SNE: t-distributed stochastic neighbor embedding.

Wiki: https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding

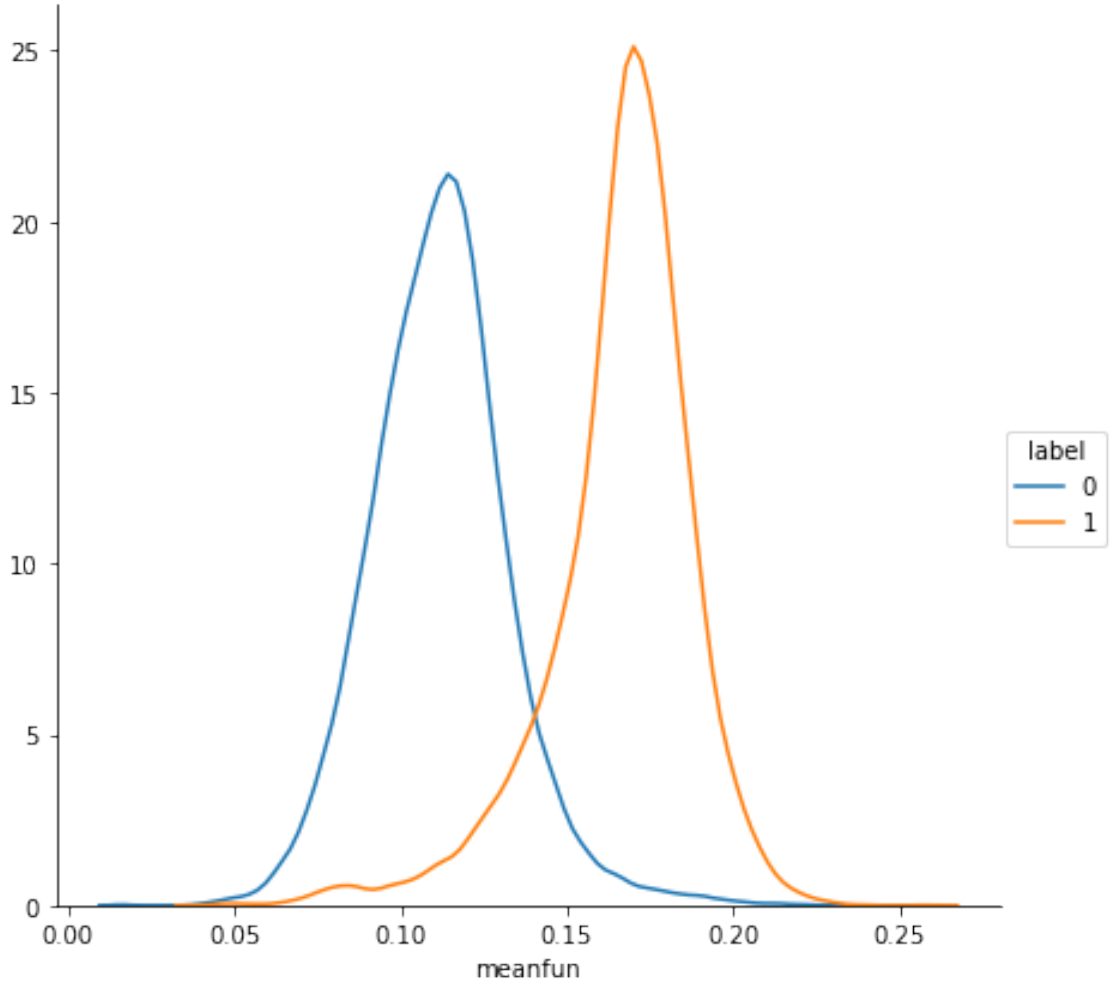
```
In [31]: tsne = TSNE(n_components=2, init='pca', random_state=0)
         X = tsne.fit_transform(X)

In [47]: plt.figure(figsize=(16, 16))
         plt.title('2-d embedding of 18 acoustic features')
         plt.scatter(X[np.where(y == 0), 0],
                     X[np.where(y == 0), 1],
                     marker='x', color='b',
                     linewidth='1', alpha=0.3, label='Male')
         plt.scatter(X[np.where(y == 1), 0],
                     X[np.where(y == 1), 1],
                     marker='x', color='r',
                     linewidth='1', alpha=0.3, label='Female')
         plt.legend()

Out[47]: <matplotlib.legend.Legend at 0x7f9931c97fd0>
```



Looks great! Clearly male and female can be **somehow** separated. Why females seem to be grouped in a corner why guys are dominating the plot? Remember that there are 5 times more males! Not only that, they also seem to represent much wider acoustic spectrum through e.g. various accents.

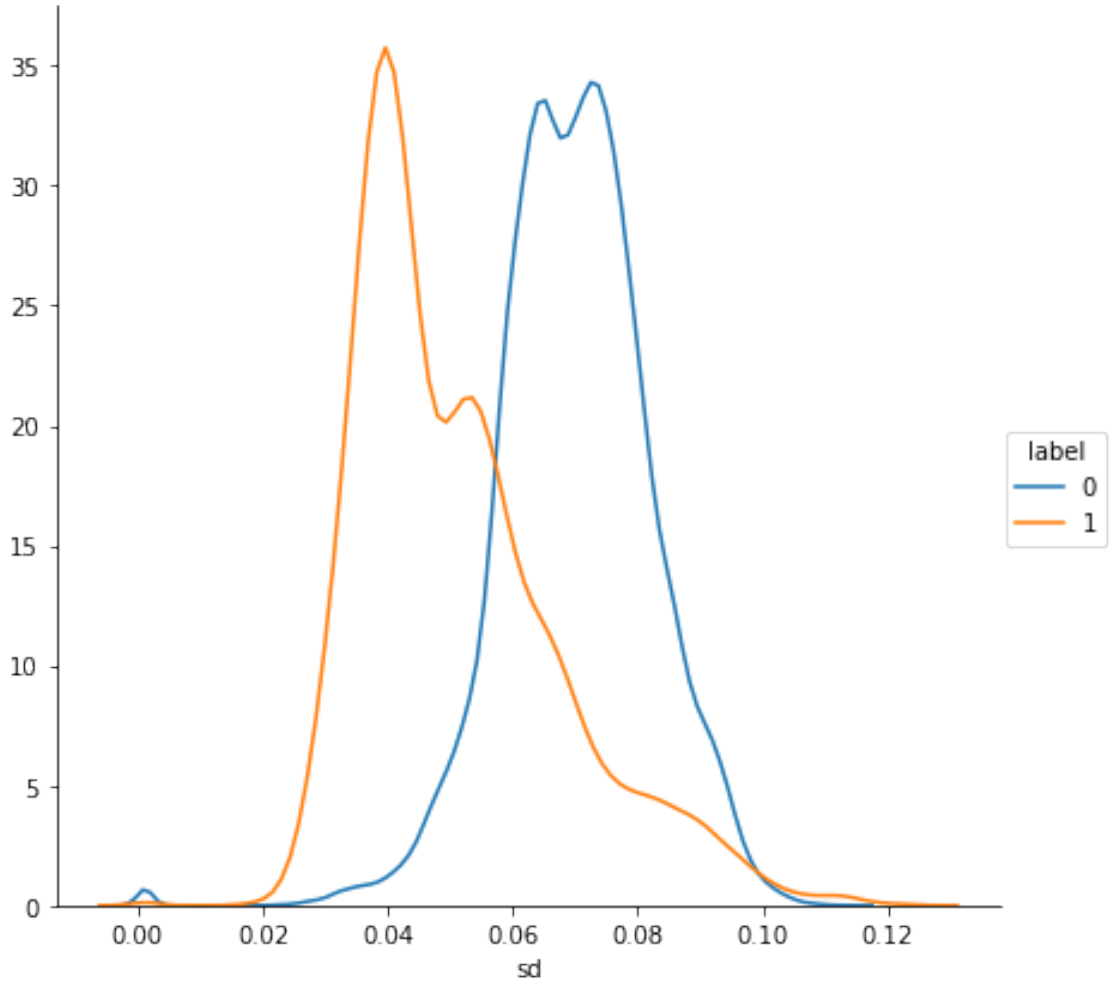
We can read on Wiki (https://en.wikipedia.org/wiki/Voice_frequency) that typically fundamental frequency is different for males and females. Let's see if it holds with our data.

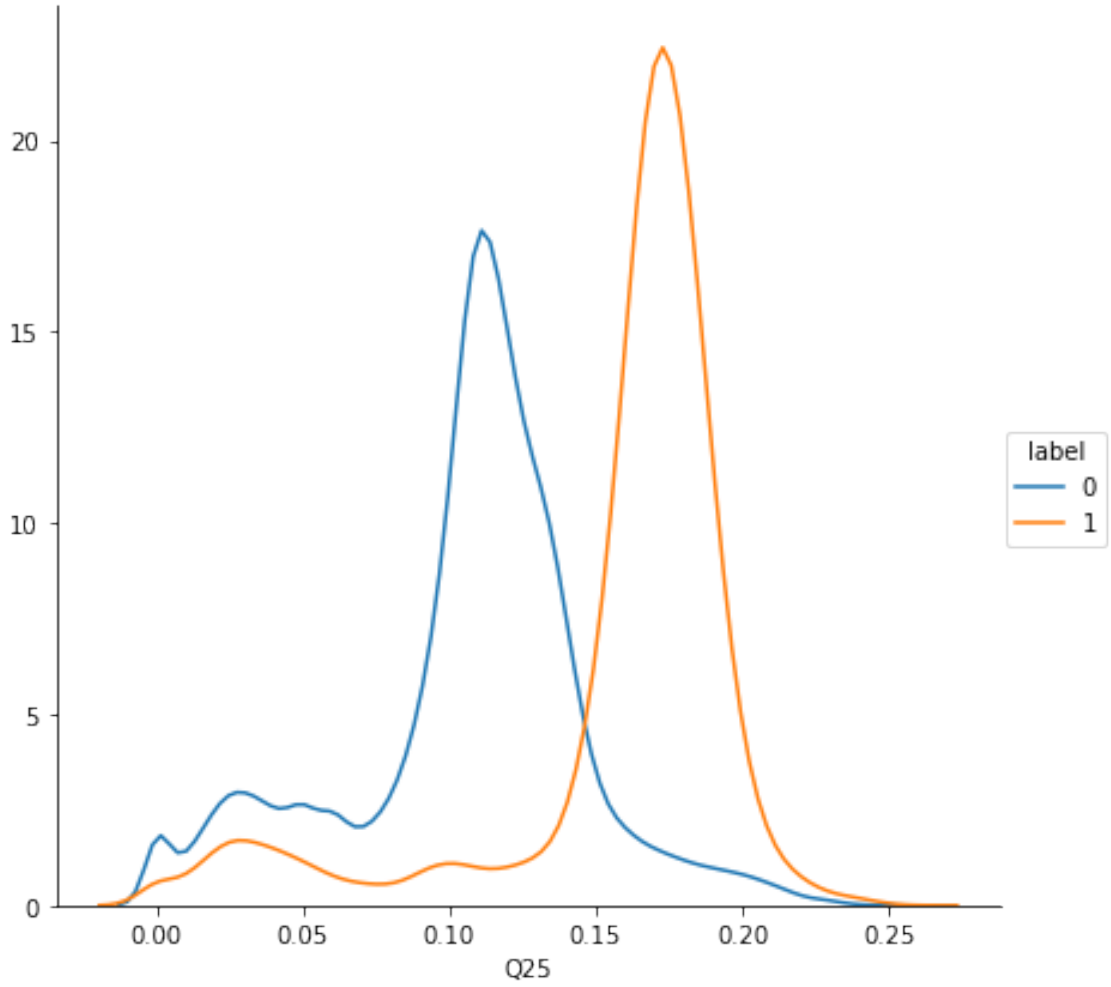
```
In [53]: data = pd.read_csv(datapath).drop(['centroid', 'dfrange', 'filename'], axis=1)

In [45]: sns.FacetGrid(data, hue="label", size=6).map(sns.kdeplot, "meanfun").add_legend()

Out[45]: <seaborn.axisgrid.FacetGrid at 0x7f9930f614a8>
```

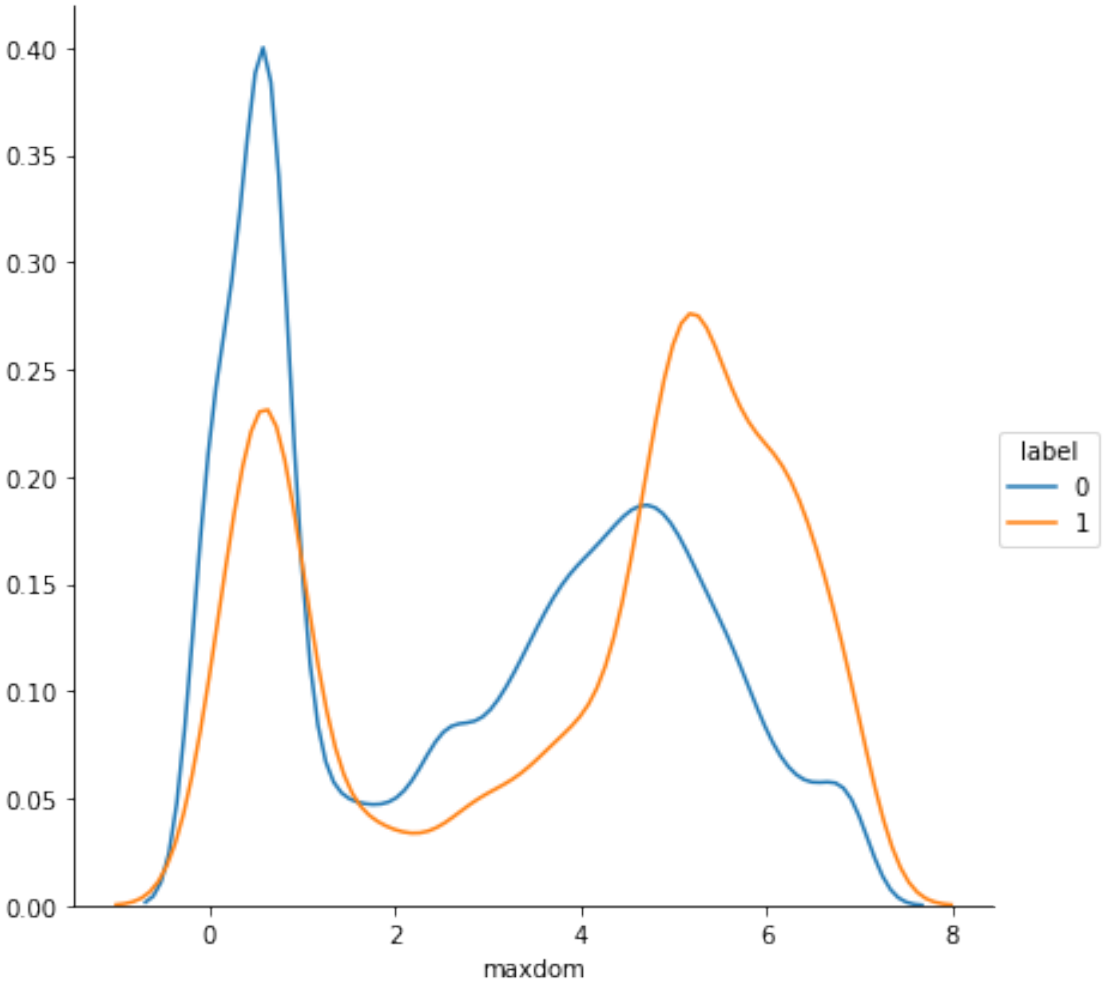


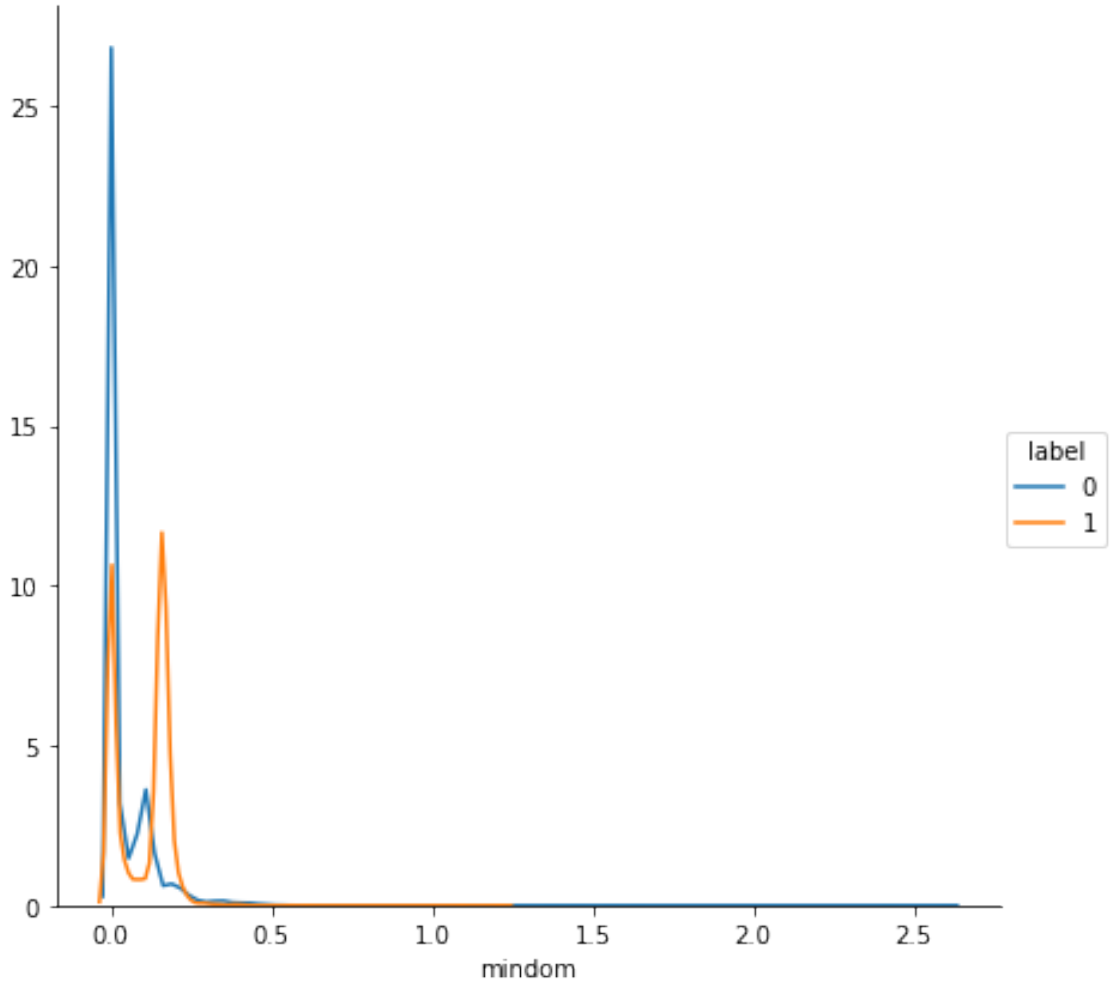
Very promising! Let's see how it looks like for all features.

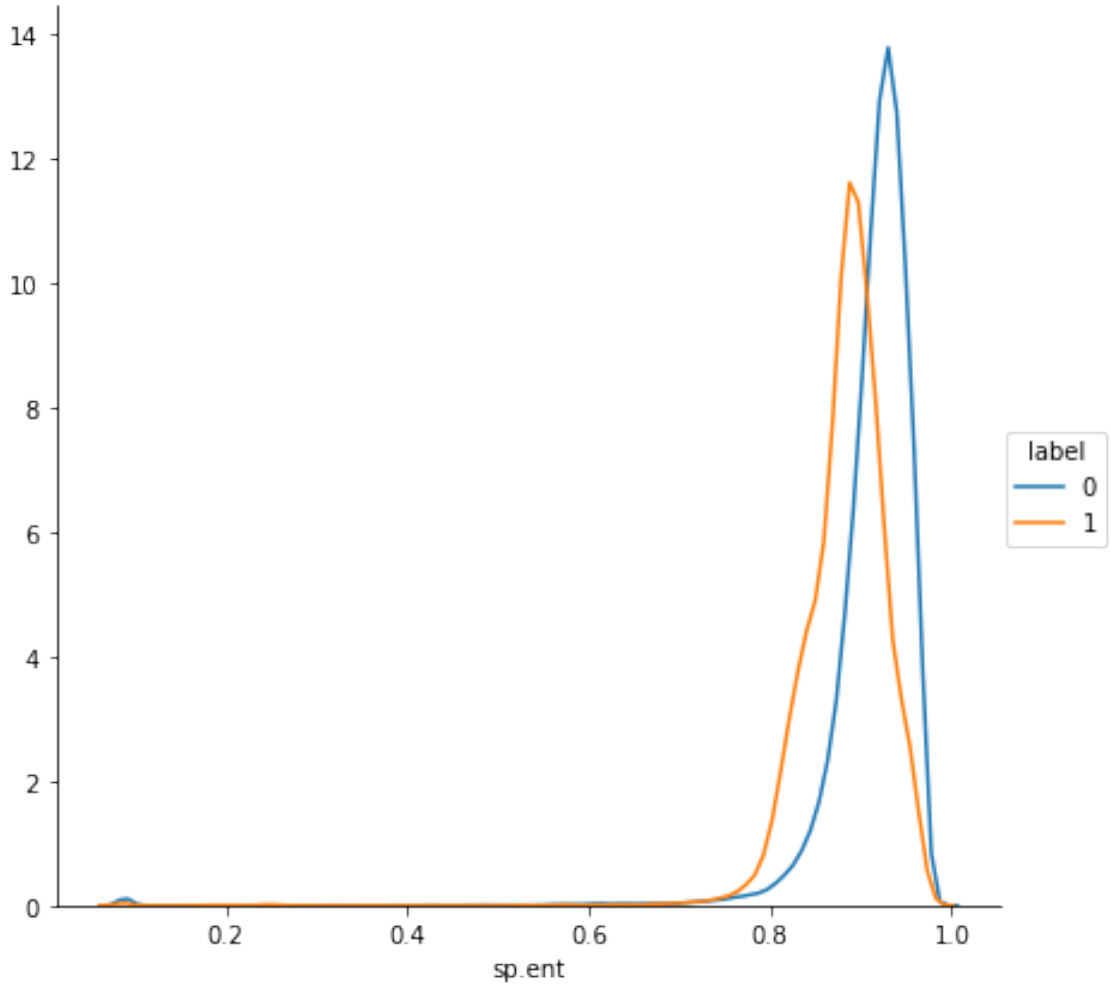
```
In [65]: for name in (set(data.columns.values) - {'label'}):
             sns.FacetGrid(data, hue="label", size=6).map(sns.kdeplot, name).add_legend()
```

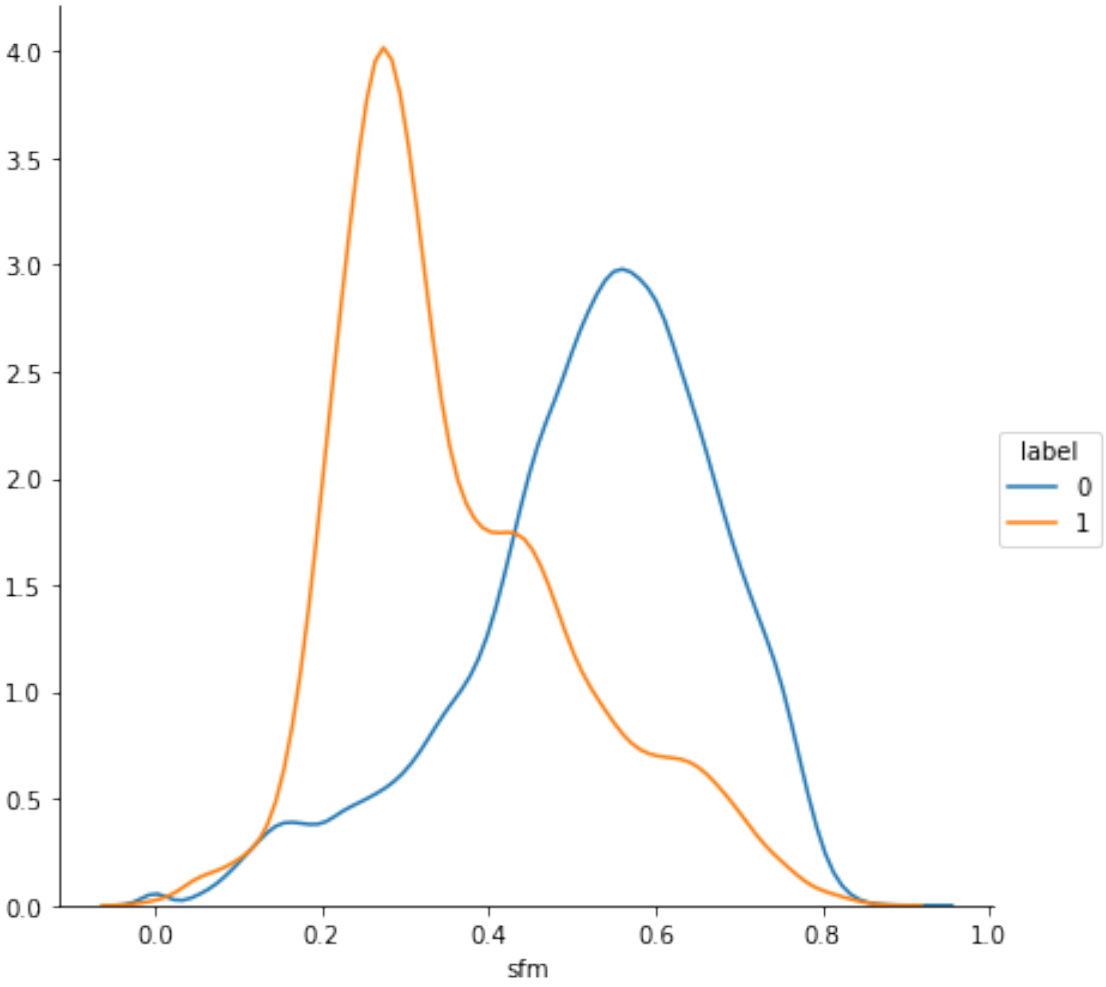

35
30
25
20
15
10
5
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
sd
label
0
1

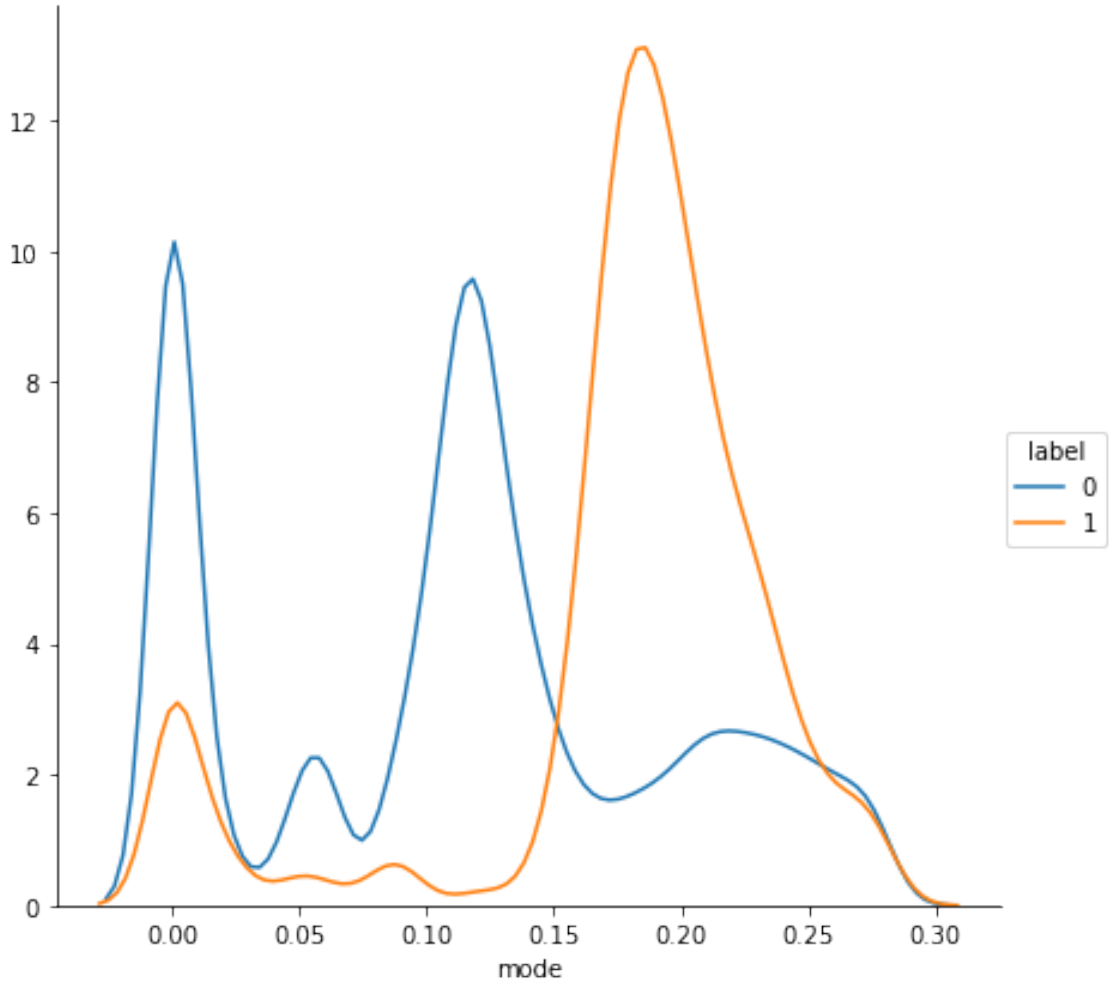

20
15
10
5
0
0.00
0.05
0.10
0.15
0.20
0.25
Q25
label
0
1

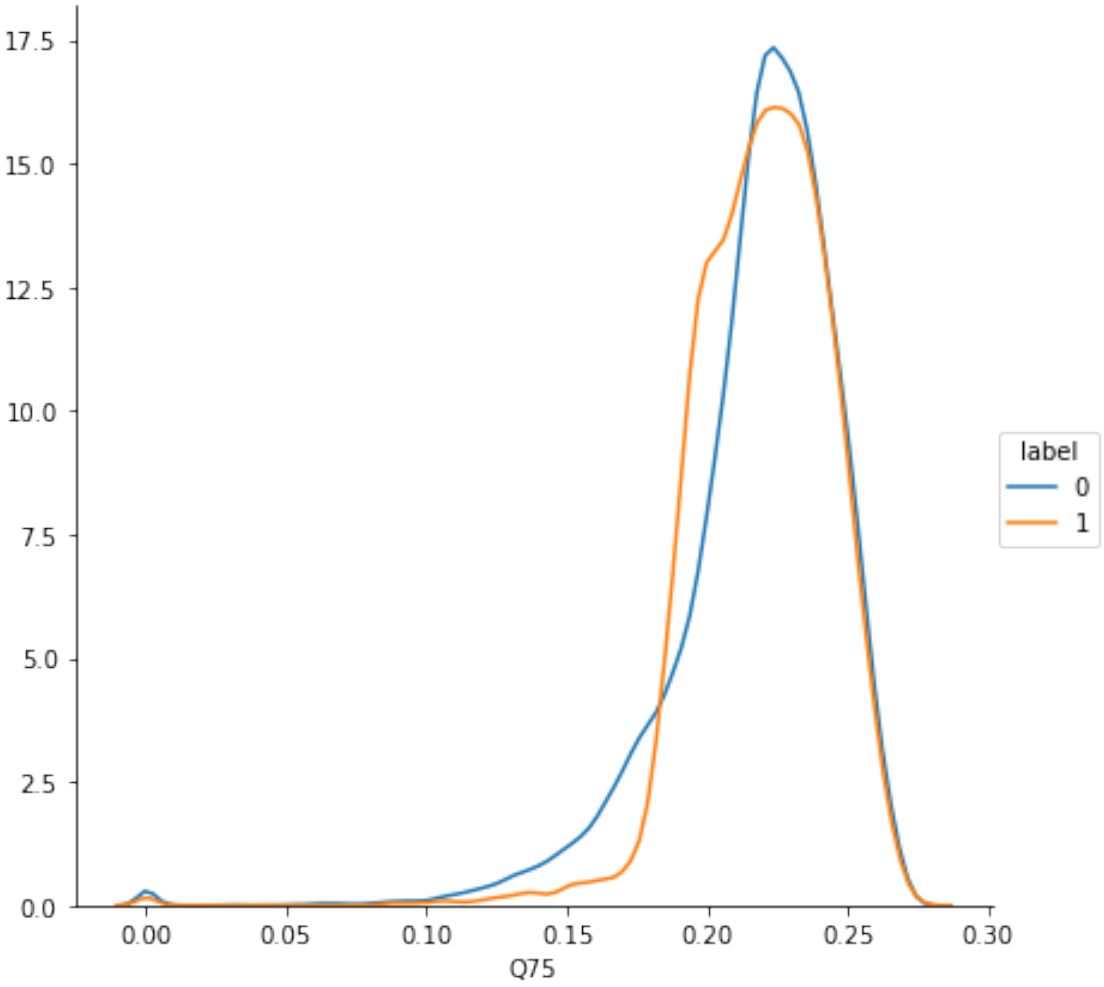

0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
maxdom
label
0
1

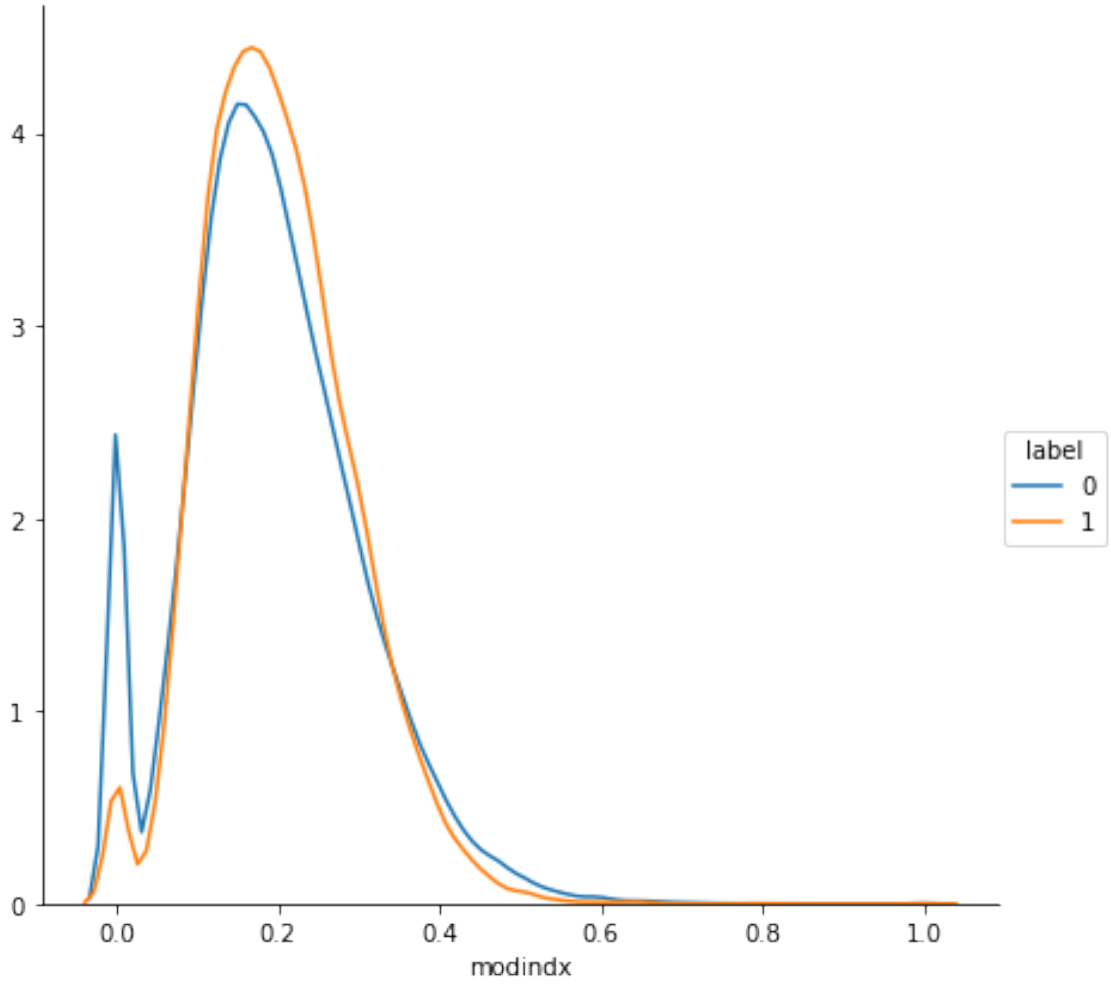

25
20
15
10
5
0
0.0
0.5
1.0
1.5
2.0
2.5
mindom
label
0
1

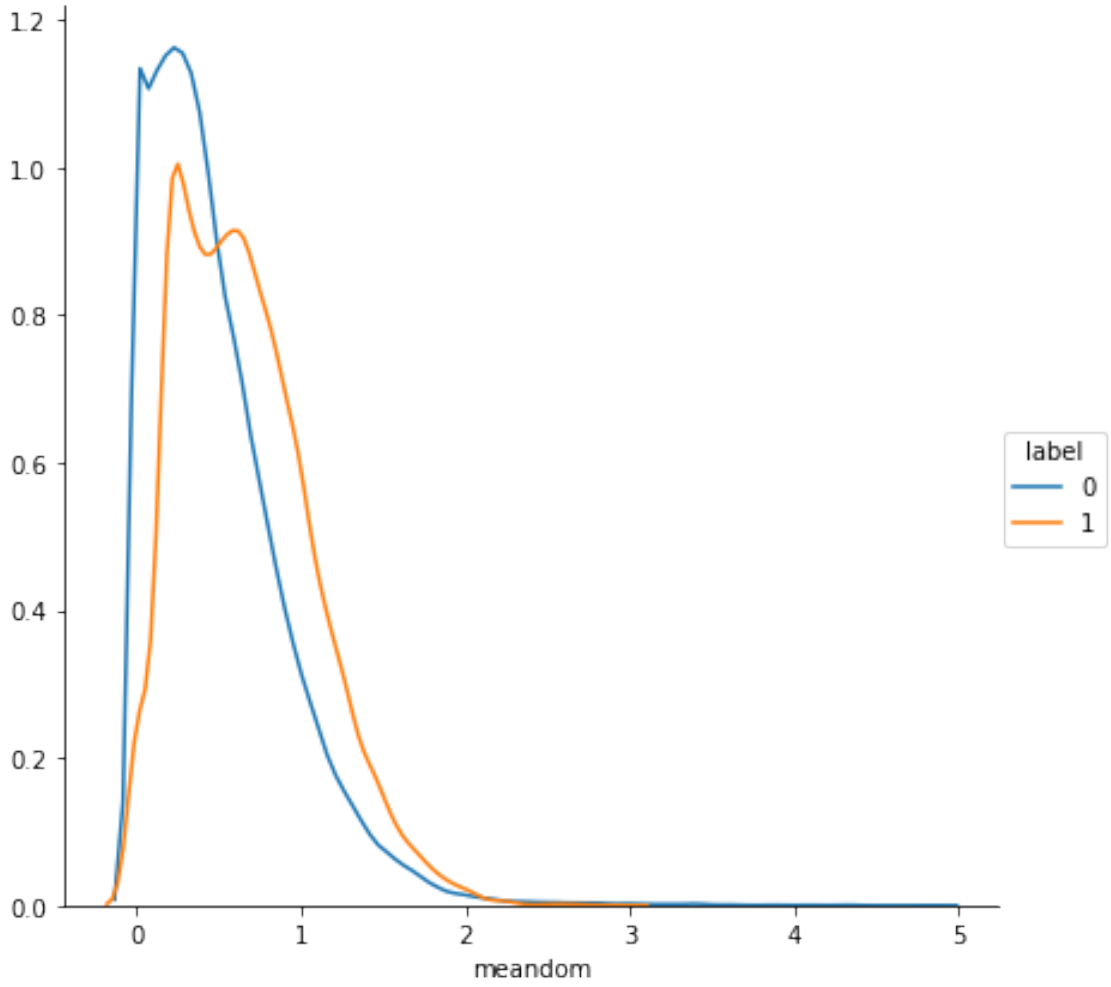

14
12
10
8
6
4
2
0
0.2
0.4
0.6
0.8
1.0
sp.ent
label
0
1

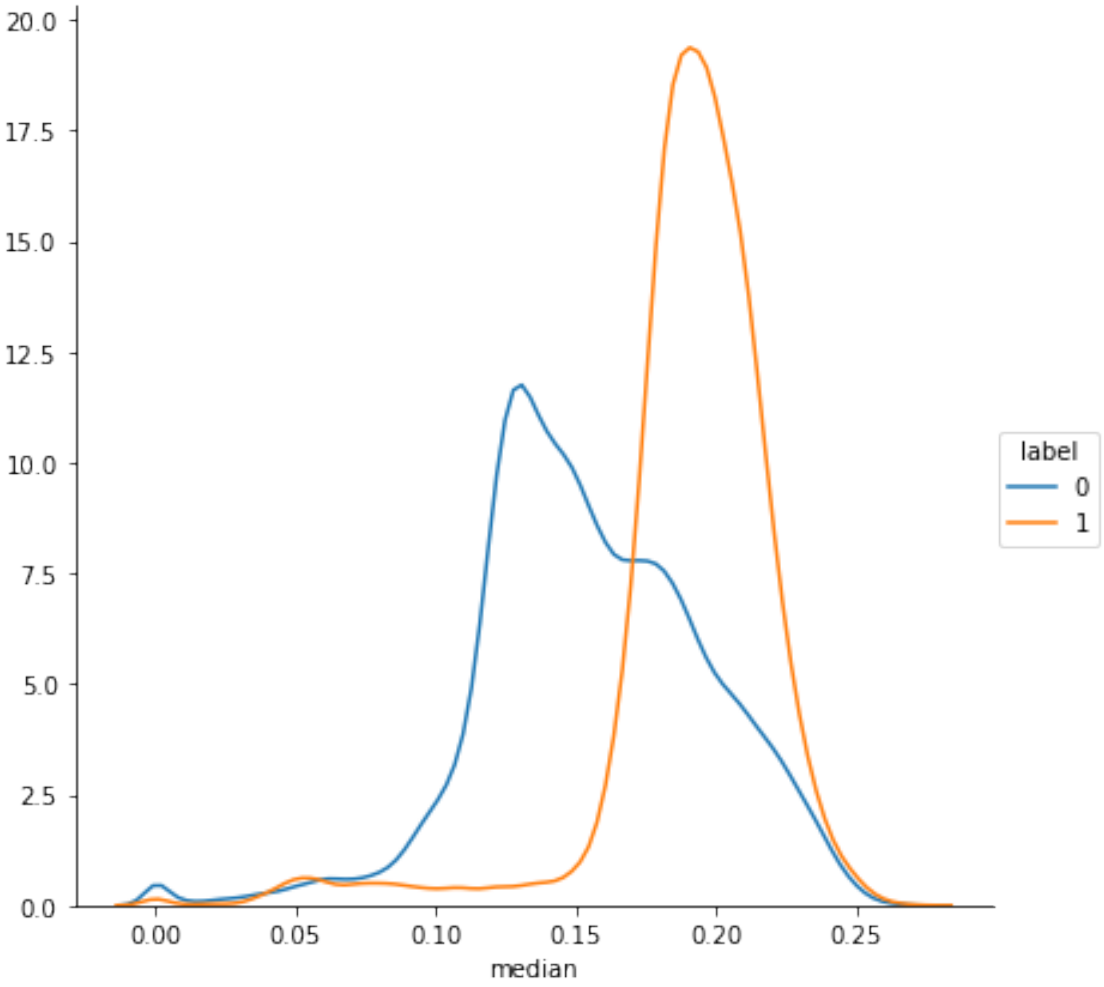

4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0
sfm
label
0
1

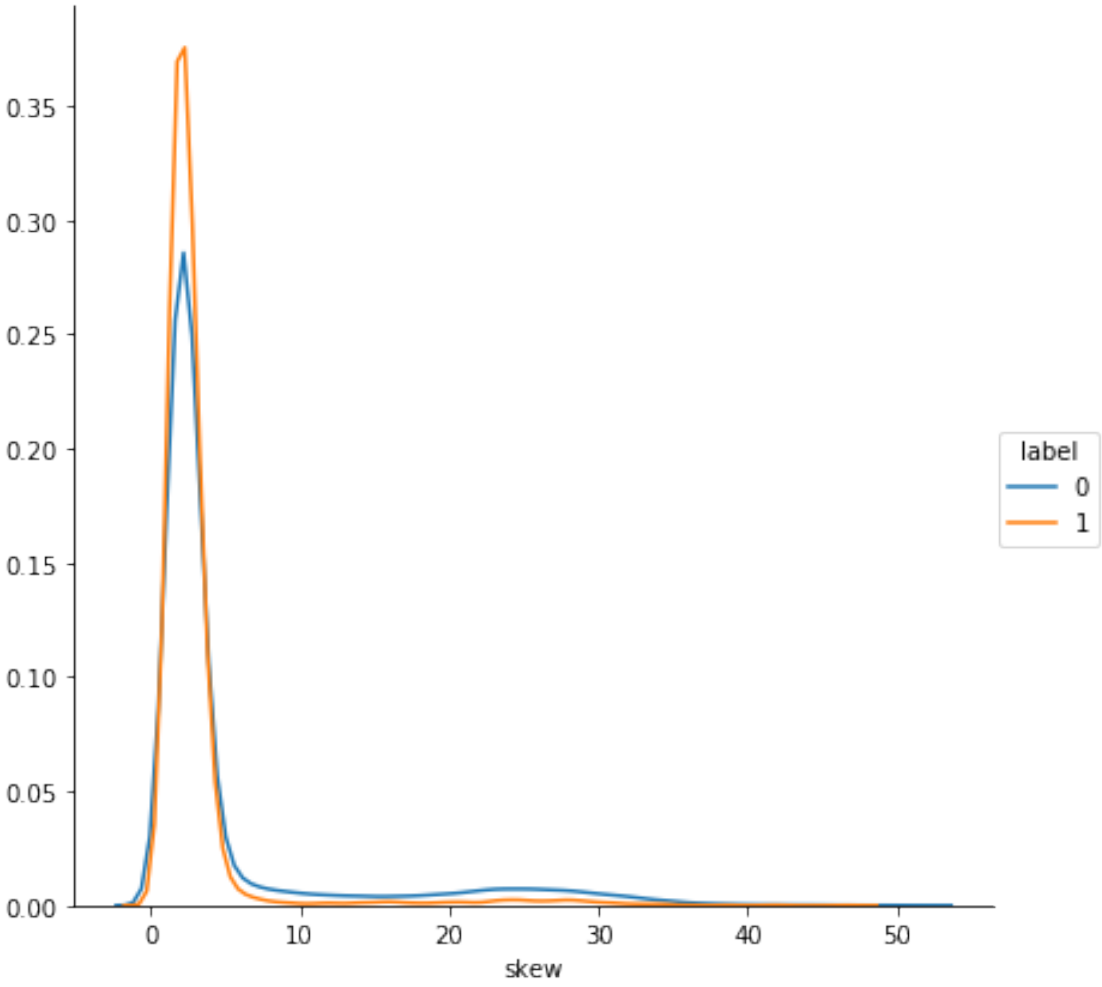

12
10
8
6
4
2
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
mode
label
0
1

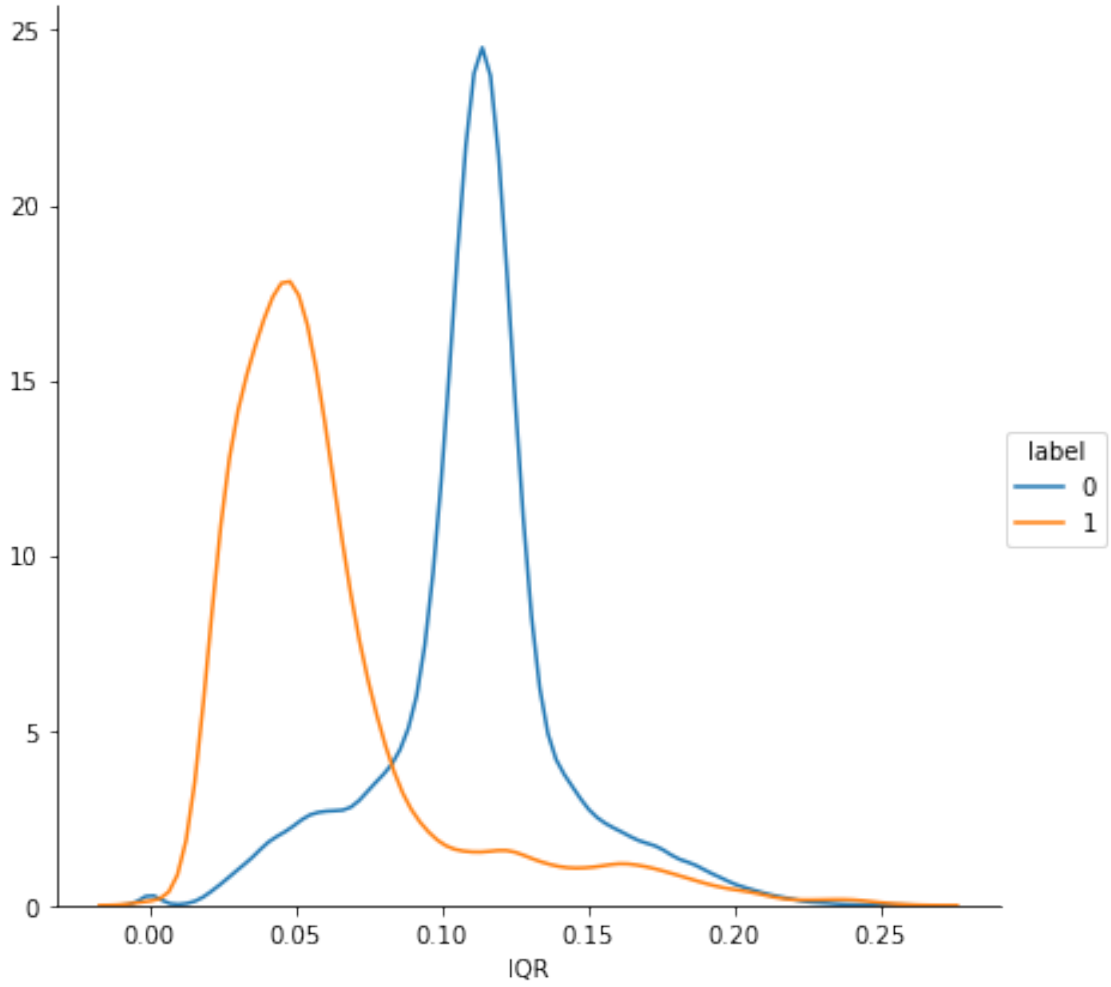

17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Q75
label
0
1

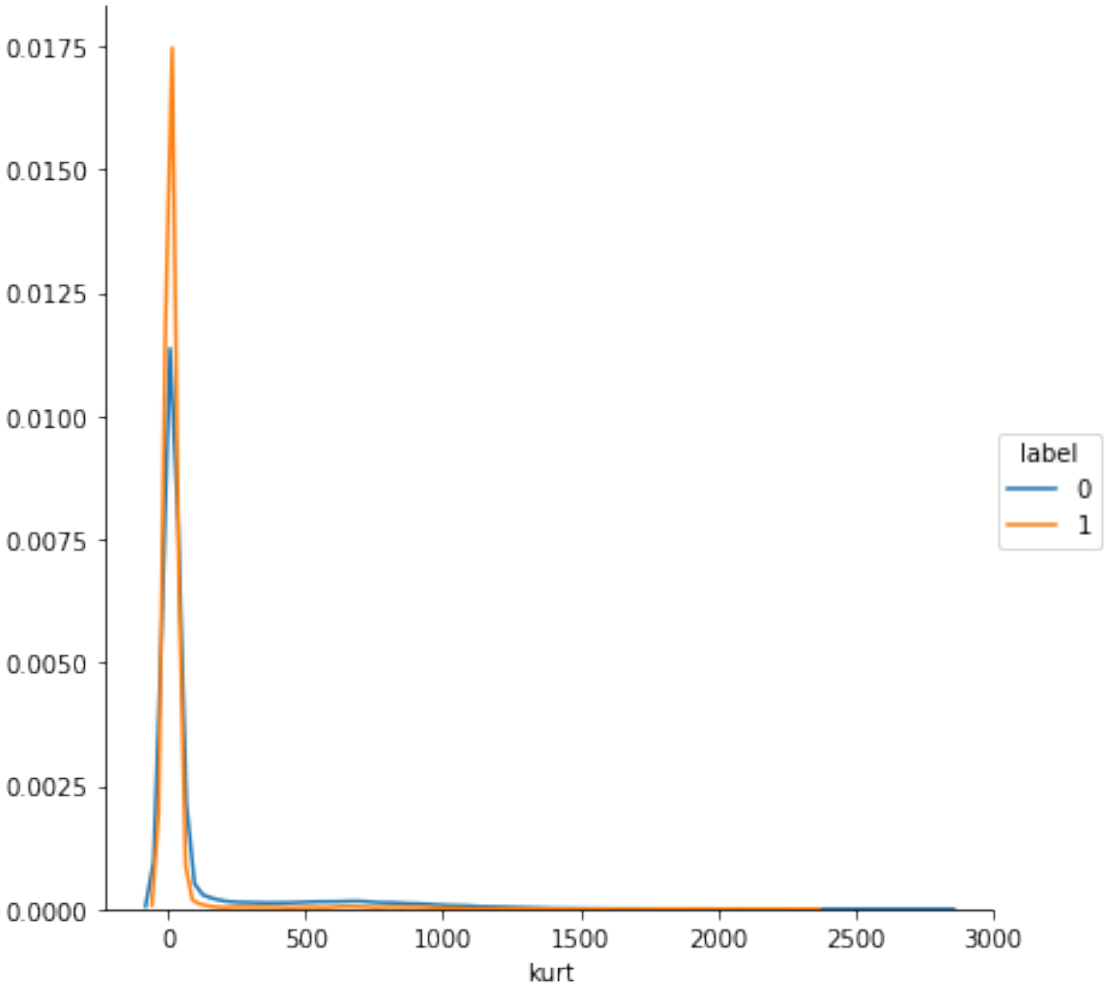

4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0
modindx
label
0
1

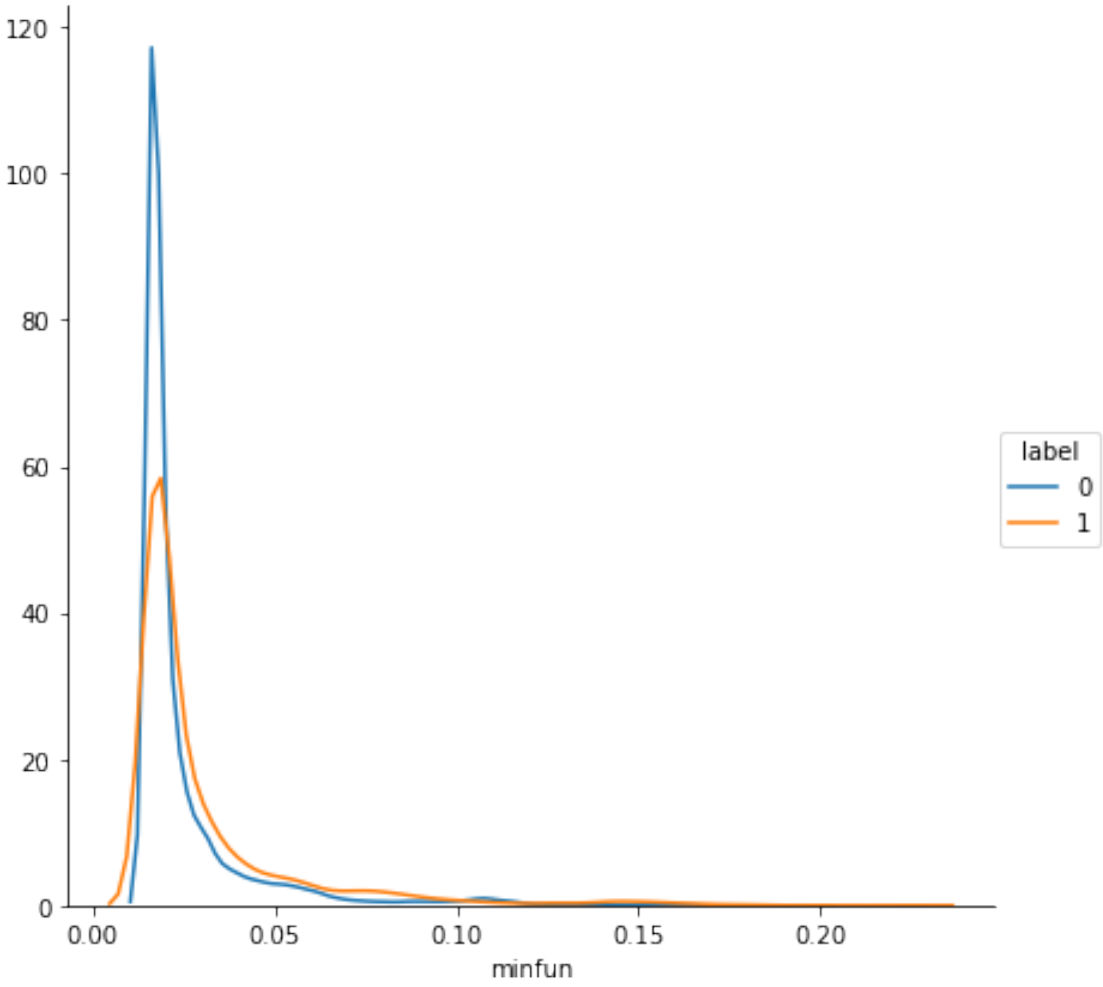

1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
meandom
label
0
1

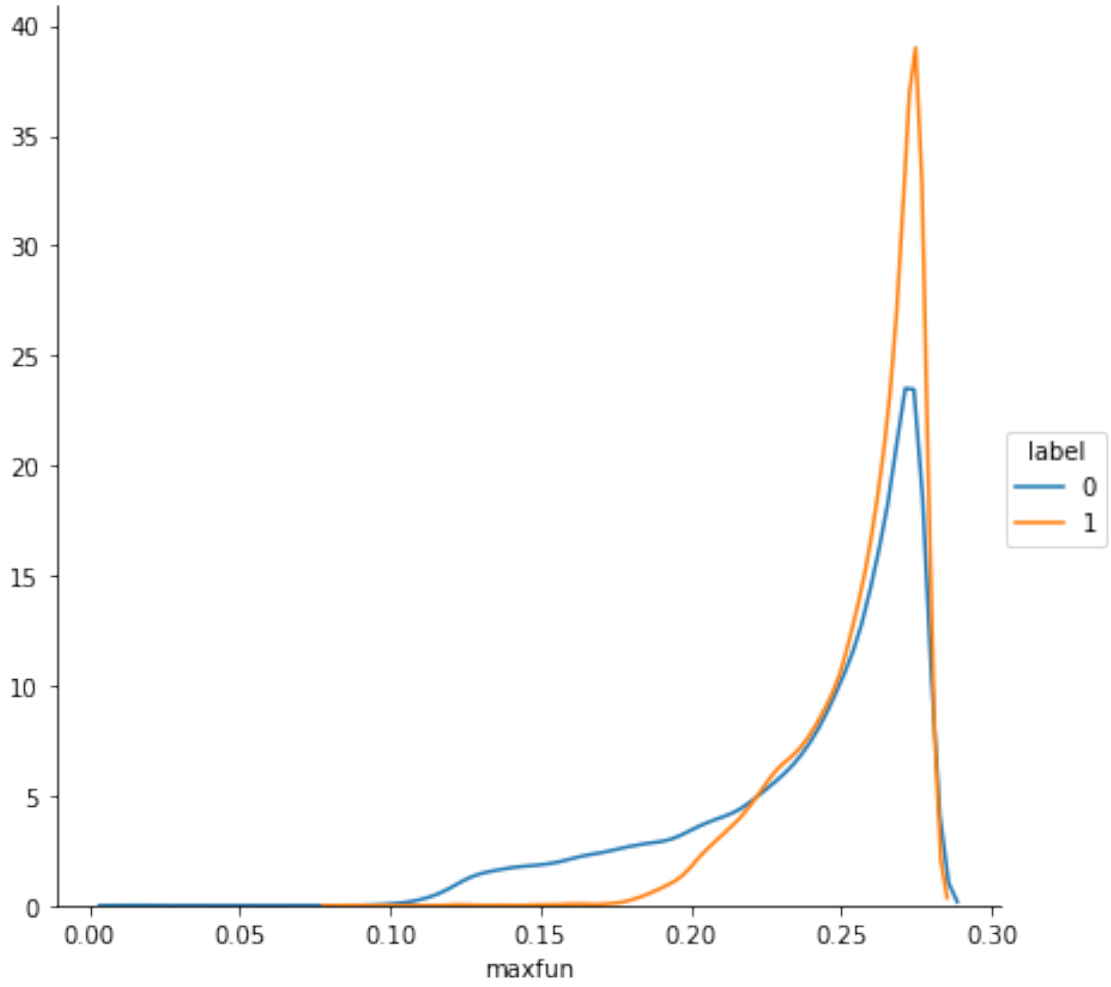

20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.00
0.05
0.10
0.15
0.20
0.25
median
label
0
1

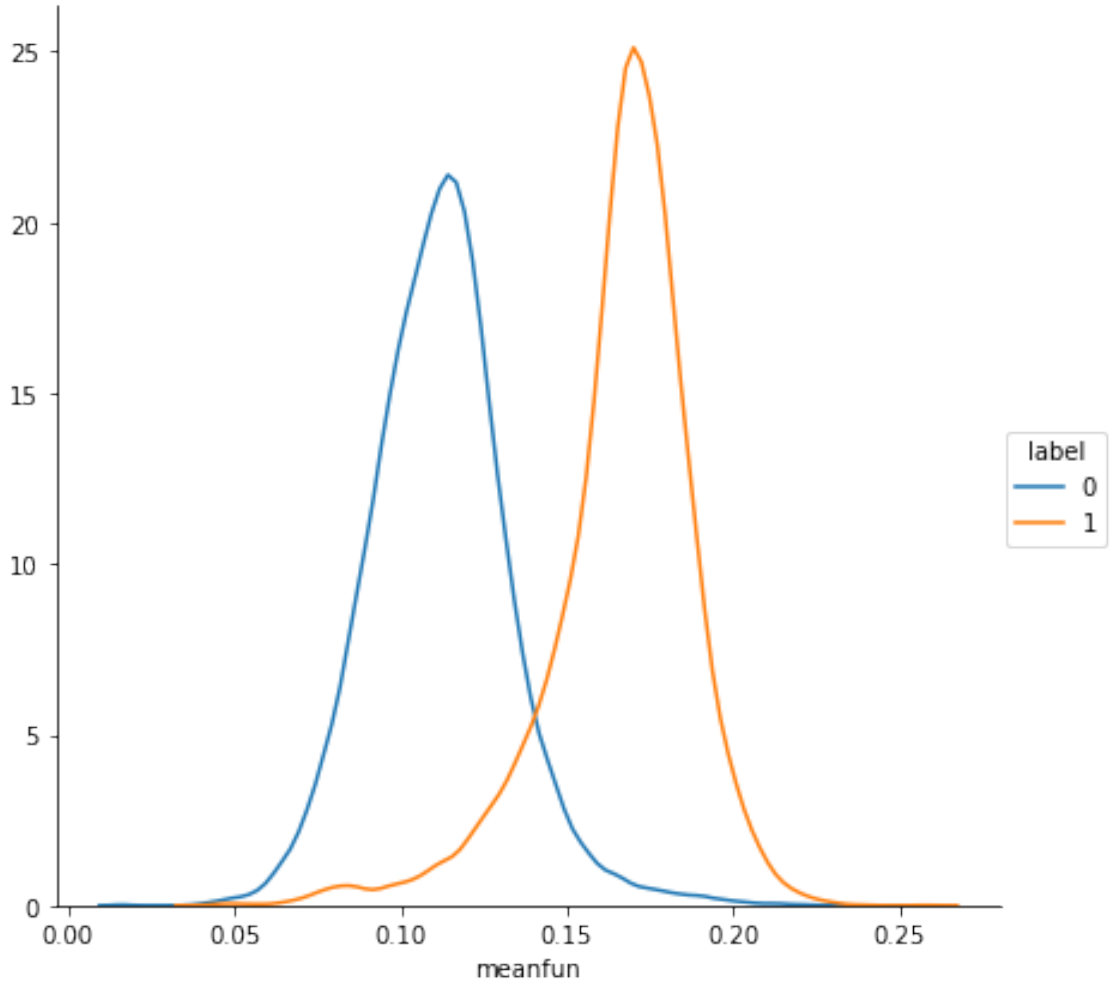

0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
skew
label
0
1

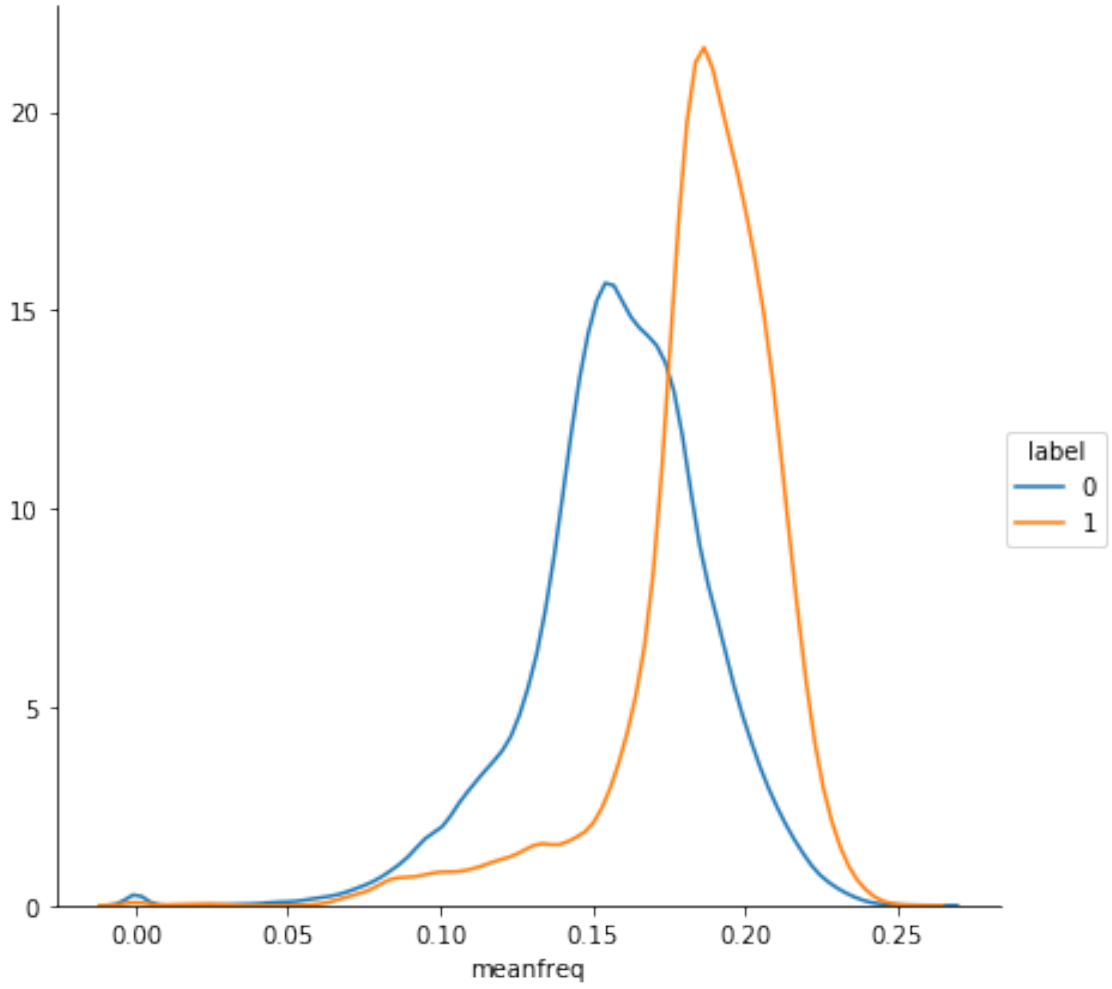

25
20
15
10
5
0
0.00
0.05
0.10
0.15
0.20
0.25
IQR
label
0
1


0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
500
1000
1500
2000
2500
3000
kurt
label
0
1


120
100
80
60
40
20
0
0.00
0.05
0.10
0.15
0.20
minfun
label
0
1

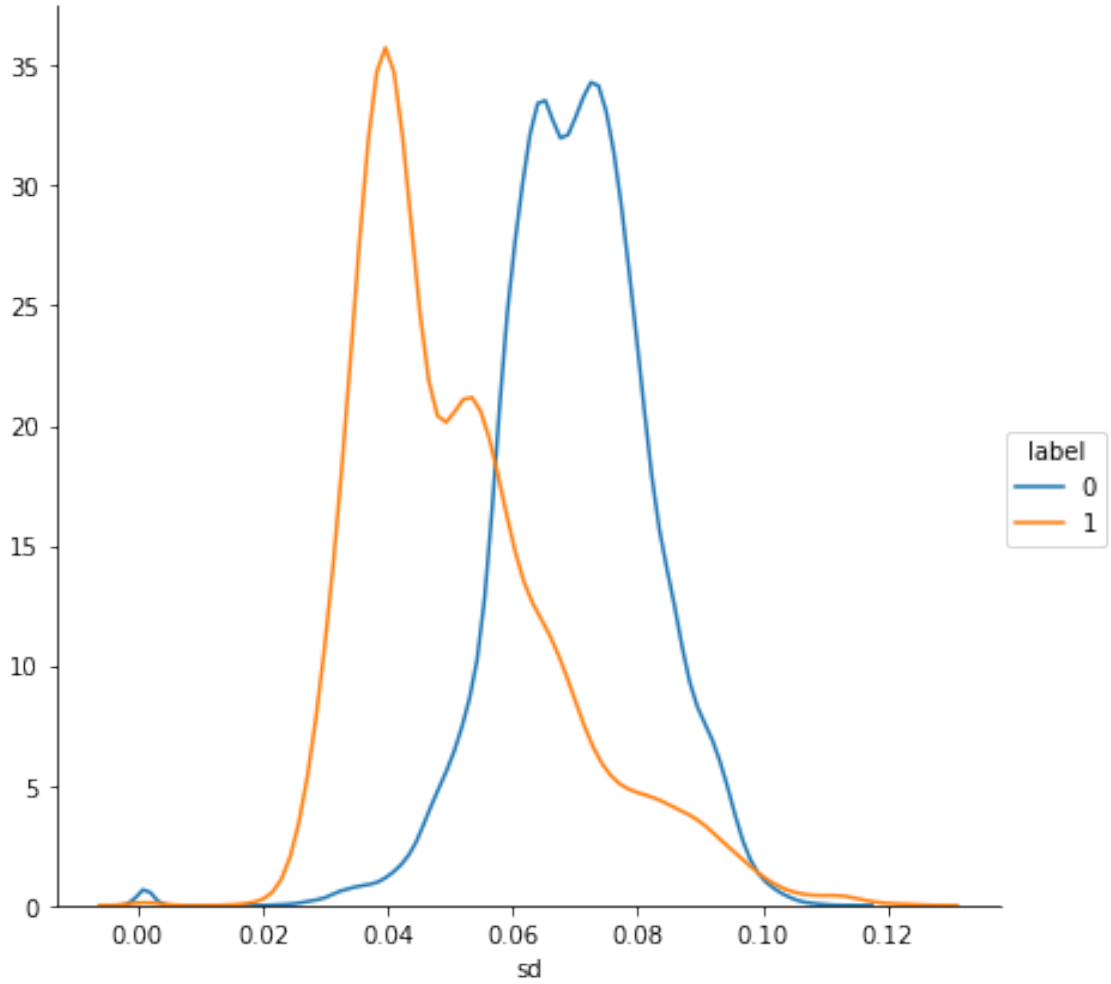

40
35
30
25
20
15
10
5
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
maxfun
label
0
1


25
20
15
10
5
0
0.00
0.05
0.10
0.15
0.20
0.25
meanfun
label
0
1




```
In [64]: sns.FacetGrid(data, hue="label", size=6).map(sns.kdeplot, "sd").add_legend()
```

```
Out[64]: <seaborn.axisgrid.FacetGrid at 0x7f992cee9780>
```


35
30
25
20
15
10
5
0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
sd
label
0
1