

EDA

March 17, 2022

```
[1]: import sys
# !{sys.executable} -m pip install -U pandas-profiling[notebook]
# !jupyter nbextension enable --py widgetsnbextension
# !pip install matplotlib
# !pip install graphviz
```

```
[2]: import os
import numpy as np
import pandas as pd
```

```
[5]: df= pd.read_csv(r'C:\Users\zdehg\Downloads\archive\DASS_data_21.02.19\data.
→csv', error_bad_lines=False, warn_bad_lines=False, sep=r'\t' )
```

C:\ProgramData\Anaconda3\lib\site-packages\pandas\util_decorators.py:311:
ParserWarning: Falling back to the 'python' engine because the 'c' engine does
not support regex separators (separators > 1 char and different from '\s+' are
interpreted as regex); you can avoid this warning by specifying engine='python'.

return func(*args, **kwargs)

C:\ProgramData\Anaconda3\lib\site-
packages\IPython\core\interactiveshell.py:3444: FutureWarning: The
warn_bad_lines argument has been deprecated and will be removed in a future
version.

exec(code_obj, self.user_global_ns, self.user_ns)

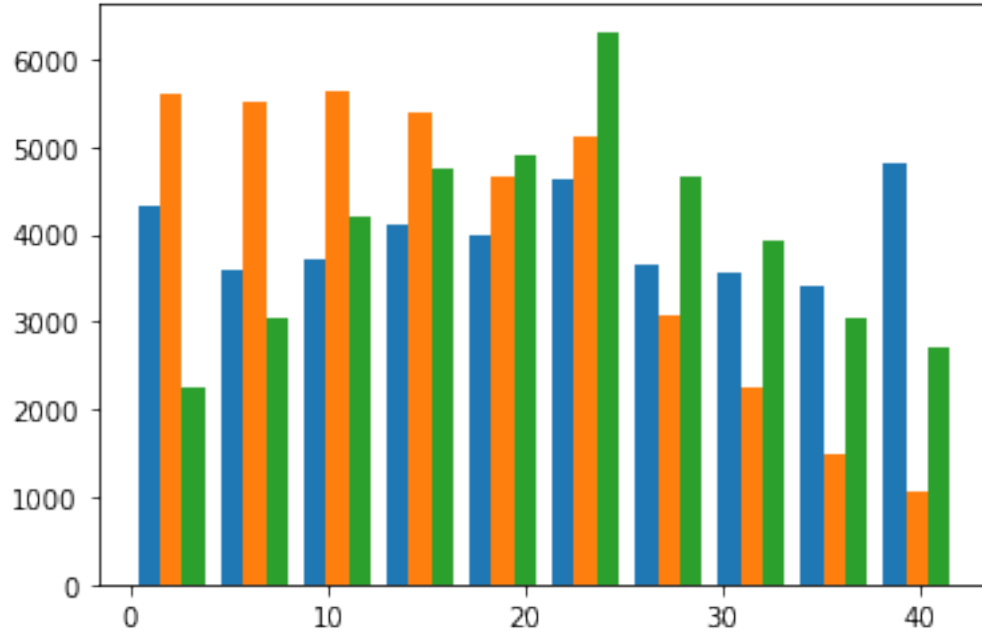
C:\ProgramData\Anaconda3\lib\site-
packages\IPython\core\interactiveshell.py:3444: FutureWarning: The
error_bad_lines argument has been deprecated and will be removed in a future
version.

exec(code_obj, self.user_global_ns, self.user_ns)

```
[19]: import matplotlib.pyplot as plt
```

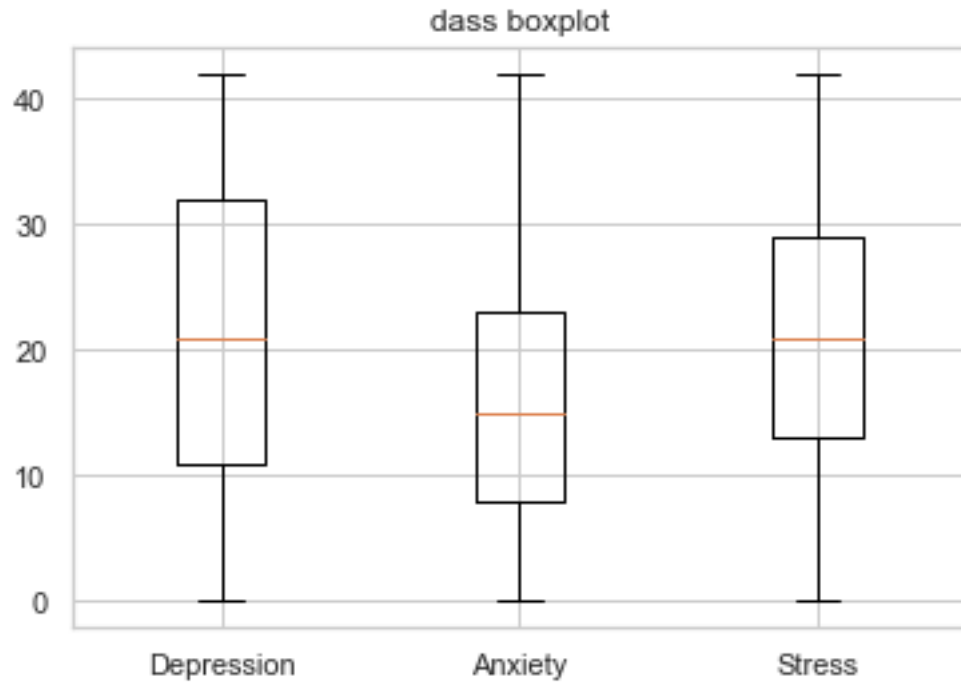
```
[20]: plt.hist(dass)
#It can be sen that this class variable is not imbalanced toward a certain
→result
```

```
[20]: (array([[4318., 3594., 3728., 4101., 3980., 4622., 3657., 3555., 3419.,
              4801.],
              [5602., 5523., 5625., 5378., 4652., 5111., 3070., 2259., 1480.,
              1075.],
              [2238., 3037., 4208., 4756., 4897., 6315., 4655., 3920., 3041.,
              2708.]]),
       array([ 0. ,  4.2,  8.4, 12.6, 16.8, 21. , 25.2, 29.4, 33.6, 37.8, 42. ]),
       <a list of 3 BarContainer objects>)
```



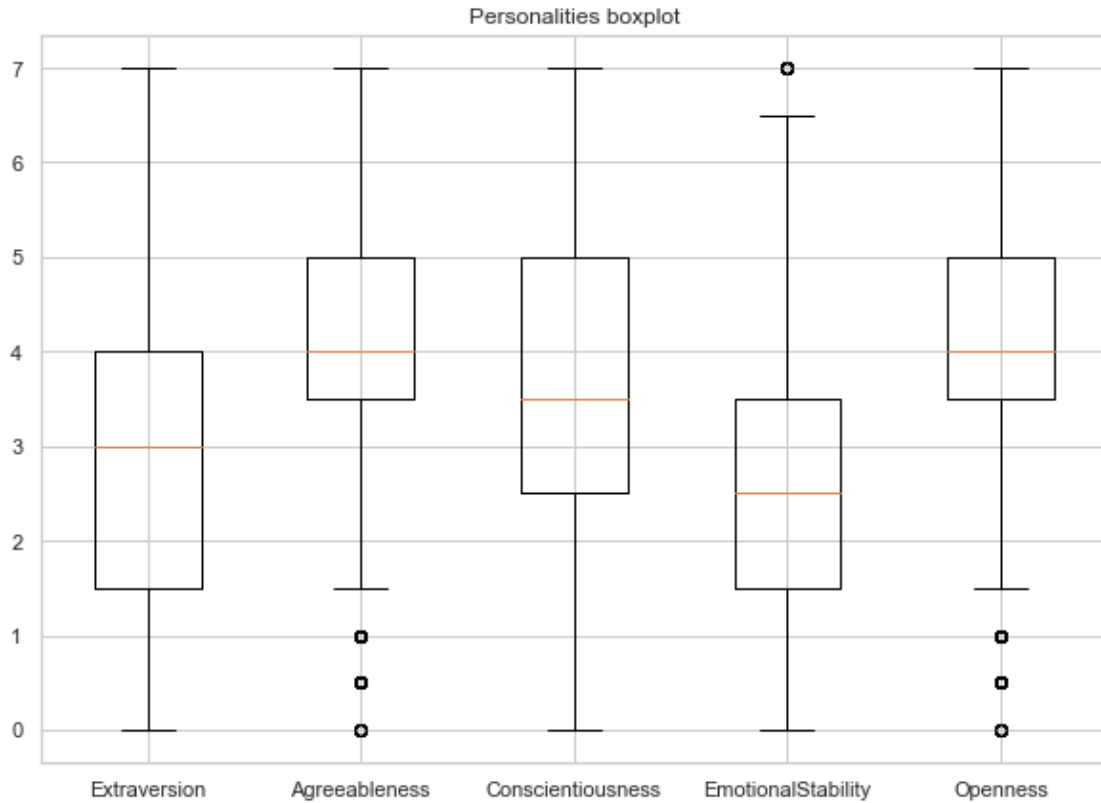
```
[256]: fig, axs = plt.subplots()
        axs.boxplot(dass, labels = ['Depression', 'Anxiety', 'Stress'], )
        axs.set_title('dass boxplot')
```

```
[256]: Text(0.5, 1.0, 'dass boxplot')
```



```
[257]: fig = plt.figure(figsize =(10, 7))
      axs = fig.add_subplot(111)
      axs.boxplot(personalities, labels= ['Extraversion', 'Agreeableness', 'Conscientiousness', 'EmotionalStability', 'Openness'])
      axs.set_title('Personalities boxplot')
```

```
[257]: Text(0.5, 1.0, 'Personalities boxplot')
```



```
[21]: def plot_correlation(df, cmap='RdBu_r'):
    size = len(df.columns)
    fig, ax = plt.subplots(figsize=(2 * size, 2 * size))
    corr = df.corr()

    im = ax.matshow(corr, cmap=cmap
                    )
    for (i, j), z in np.ndenumerate(corr):
        ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center',
                bbox=dict(boxstyle='round', facecolor='white', edgecolor='0.5'))

    plt.xticks(range(df.select_dtypes(['number']).shape[1]), df.
    ↪select_dtypes(['number']).columns, fontsize=14, rotation=90)
    plt.yticks(range(df.select_dtypes(['number']).shape[1]), df.
    ↪select_dtypes(['number']).columns, fontsize=14)
    cb = plt.colorbar(im)
    ax.tick_params(labelsize=14)
    plt.show()
```

```
[22]: character = pd.concat([dass, personalities], axis=1)
plot_correlation(character, cmap='viridis')
```

