

```
In [5]: #importing libraries
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
%matplotlib inline
#import plotly.offline as py
from matplotlib import pyplot as pp
#py.init_notebook_mode(connected=True)
#import plotly.graph_objs as go

#import plotly.tools as tls
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

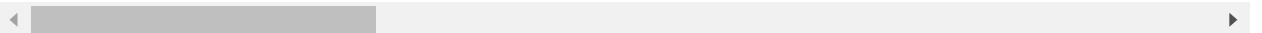
```
In [6]: df=pd.read_csv('Wellbeing_and_lifestyle_data.csv')
```

```
In [7]: df.head()
```

Out[7]:

	Timestamp	FRUITS_VEGGIES	DAILY_STRESS	PLACES_VISITED	CORE_CIRCLE	SUPPORTING_
0	7/7/15	3	2	2	5	
1	7/7/15	2	3	4	3	
2	7/7/15	2	3	3	4	
3	7/7/15	3	3	10	3	
4	7/7/15	5	1	3	3	

5 rows × 23 columns



Data Cleaning

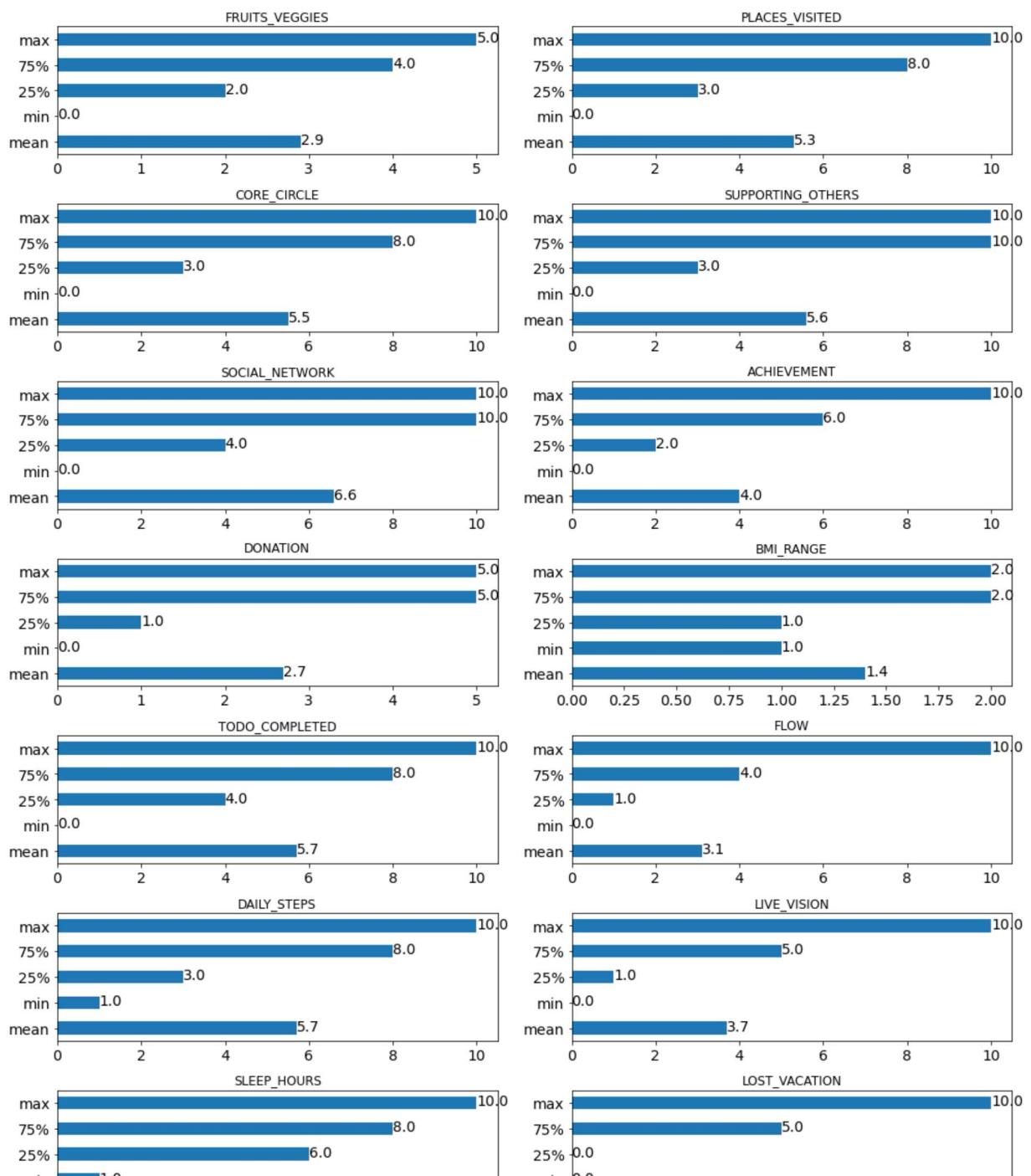
```
In [8]: df['AGE']=df['AGE'].replace('Less than 20', '20 or less')
```

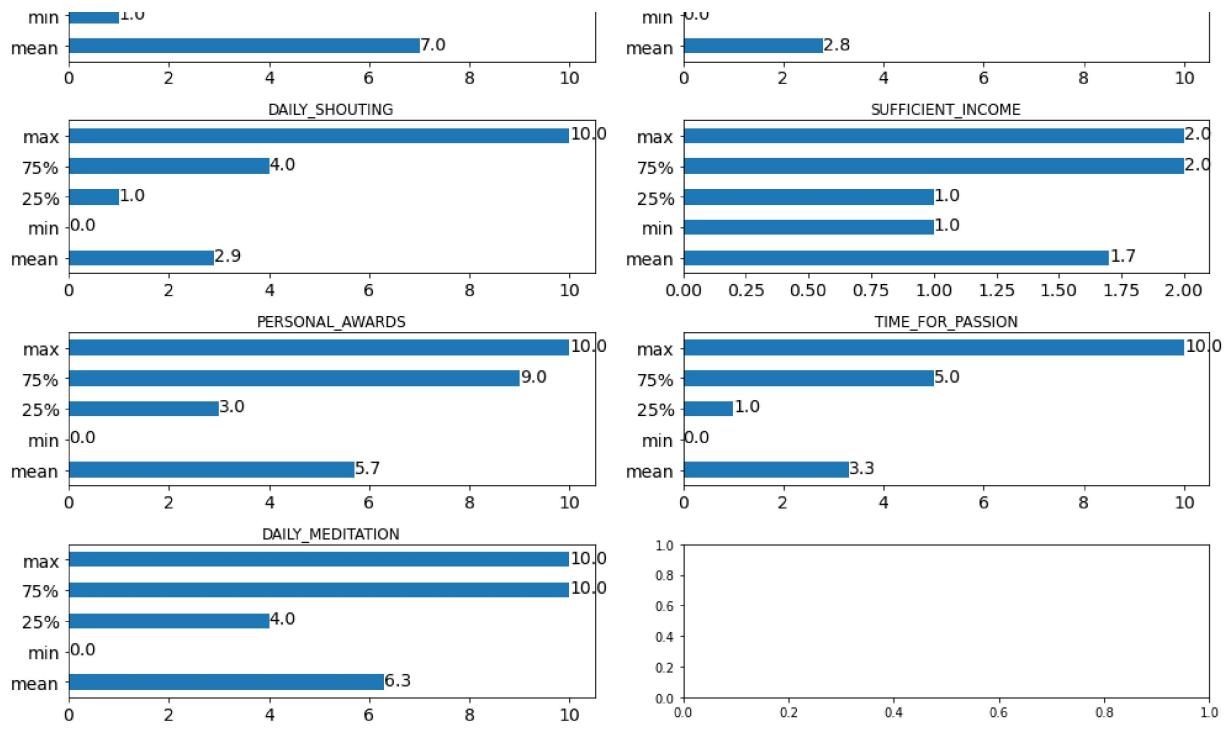
```
In [9]: column = df.columns  
column
```

```
Out[9]: Index(['Timestamp', 'FRUITS_VEGGIES', 'DAILY_STRESS', 'PLACES_VISITED',  
               'CORE_CIRCLE', 'SUPPORTING_OTHERS', 'SOCIAL_NETWORK', 'ACHIEVEMENT',  
               'DONATION', 'BMI_RANGE', 'TODO_COMPLETED', 'FLOW', 'DAILY_STEPS',  
               'LIVE_VISION', 'SLEEP_HOURS', 'LOST_VACATION', 'DAILY_SHOUTING',  
               'SUFFICIENT_INCOME', 'PERSONAL_AWARDS', 'TIME_FOR_PASSION',  
               'DAILY_MEDITATION', 'AGE', 'GENDER'],  
              dtype='object')
```

```
In [12]: df=df.dropna(axis=1)
```

```
In [14]: def descriptive(df):
    desc=df.describe().round(1).drop({'count', 'std', '50%'}, axis=0)
    i=-0.1
    j=0
    Row = int(round(len(desc.columns.tolist())/2+0.1))
    f,ax = plt.subplots(Row,2, figsize=(28,18))
    for name in desc.columns.tolist():
        desc[name].plot(kind='barh', figsize=(14,24), title=name, ax=ax[round(i),j])
        for k, v in enumerate(desc[name].tolist()):
            ax[round(i), j].text(v, k-0.1, str(v), color='black', size = 14)
        i +=0.5
        if j==0: j=1
        else: j=0
    f.tight_layout()
descriptive(df)
```



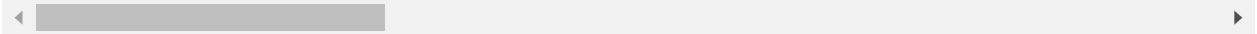


```
In [15]: df['MONTH'] = pd.DatetimeIndex(df['Timestamp']).month
df.head(3)
```

Out[15]:

	Timestamp	FRUITS_VEGGIES	DAILY_STRESS	PLACES_VISITED	CORE_CIRCLE	SUPPORTING_
0	7/7/15	3	2	2	2	5
1	7/7/15	2	3	4	3	
2	7/7/15	2	3	3	4	

3 rows × 24 columns



Exploratory Data Analysis (EDA)

Healthy body

```
In [16]: df2 = df.pivot_table(values='BMI_RANGE', index=['AGE'], columns=['GENDER'], )
df2.head()
```

Out[16]:

GENDER	Female	Male
AGE		
20 or less	1.213714	1.224490
21 to 35	1.354757	1.323077
36 to 50	1.442637	1.513878
51 or more	1.524862	1.522779

In [17]: # *HEALTHY BODY*

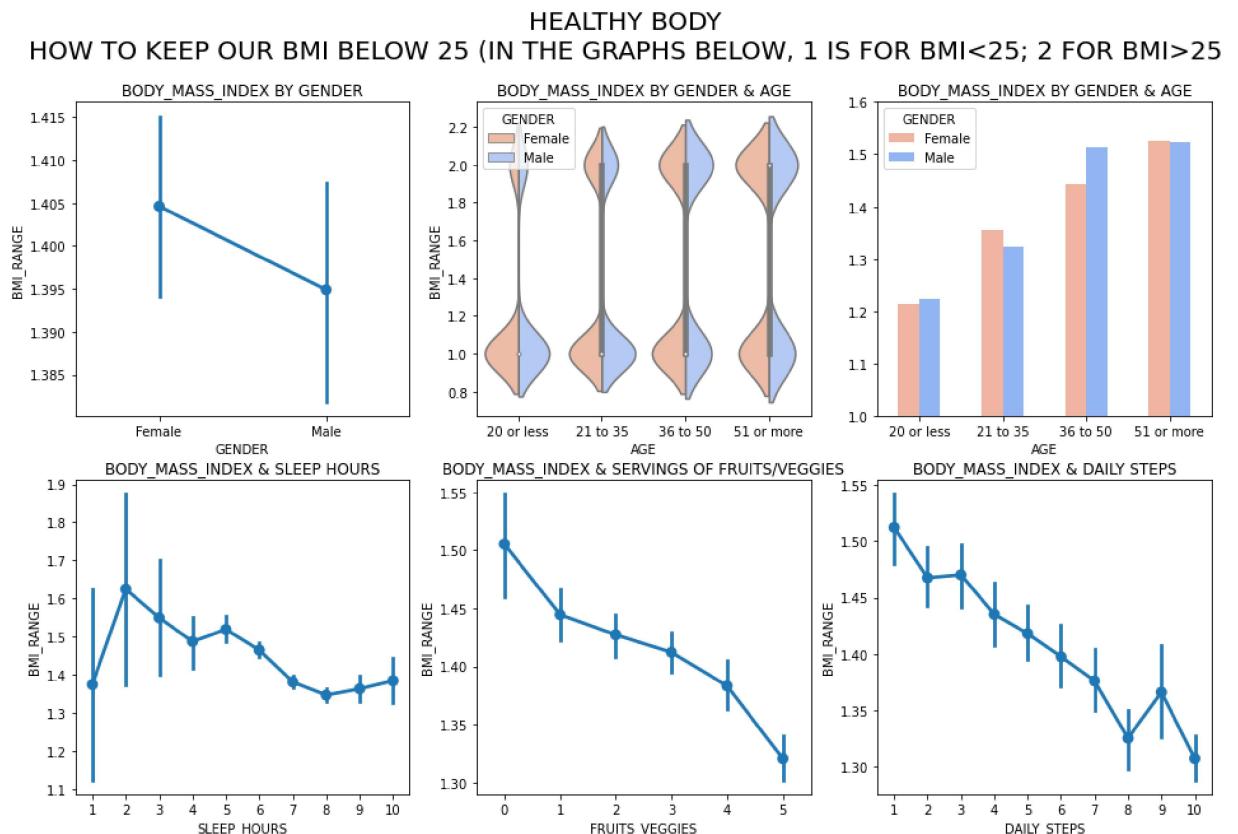
```
f,ax = plt.subplots(2,3,figsize=(16,10))
ax[0,0].set_title('BODY_MASS_INDEX BY GENDER')
ax[0,1].set_title('BODY_MASS_INDEX BY GENDER & AGE')
ax[0,2].set_title('BODY_MASS_INDEX BY GENDER & AGE')
ax[1,0].set_title('BODY_MASS_INDEX & SLEEP HOURS')
ax[1,1].set_title('BODY_MASS_INDEX & SERVINGS OF FRUITS/VEGGIES')
ax[1,2].set_title('BODY_MASS_INDEX & DAILY STEPS')

sns.pointplot(x = 'GENDER', y = 'BMI_RANGE', data=df, ax = ax[0,0])
sns.violinplot(x = 'AGE', y = 'BMI_RANGE', hue = 'GENDER', data = df, palette='colorblind',
                order=['20 or less', '21 to 35', '36 to 50', '51 or more'], split=True)

ax[0,2].set_ylim([1, 1.6])
df2.plot(kind='bar', color=('darksalmon', 'cornflowerblue'), alpha=0.7, ax = ax[0,2])
ax[0,2].tick_params(axis='x', rotation=0)

sns.pointplot(x = 'SLEEP_HOURS', y = 'BMI_RANGE', data=df, ax = ax[1,0])
sns.pointplot(x = 'FRUITS_VEGGIES', y = 'BMI_RANGE', data=df, ax = ax[1,1])
sns.pointplot(x = 'DAILY_STEPS', y = 'BMI_RANGE', data=df, ax = ax[1,2])

f.suptitle('HEALTHY BODY\nHOW TO KEEP OUR BMI BELOW 25 (IN THE GRAPHS BELOW, 1 IS FOR BMI<25; 2 FOR BMI>25')
plt.show()
```



Healthy mind

In [18]: df['DAILY_STRESS']=pd.to_numeric(df['DAILY_STRESS'], errors = 'coerce')

```
In [19]: df3 = df.pivot_table(values='DAILY_STRESS', index=['AGE'], columns=['GENDER'], )
df3.head()
```

Out[19]:

GENDER	Female	Male
AGE		
20 or less	3.032000	2.317784
21 to 35	3.004357	2.561567
36 to 50	2.943297	2.789197
51 or more	2.731492	2.421412

```
In [20]: df3 = df.pivot_table(values='DAILY_STRESS', index=['AGE'], columns=['GENDER'], )
df3.head()
```

Out[20]:

GENDER	Female	Male
AGE		
20 or less	3.032000	2.317784
21 to 35	3.004357	2.561567
36 to 50	2.943297	2.789197
51 or more	2.731492	2.421412

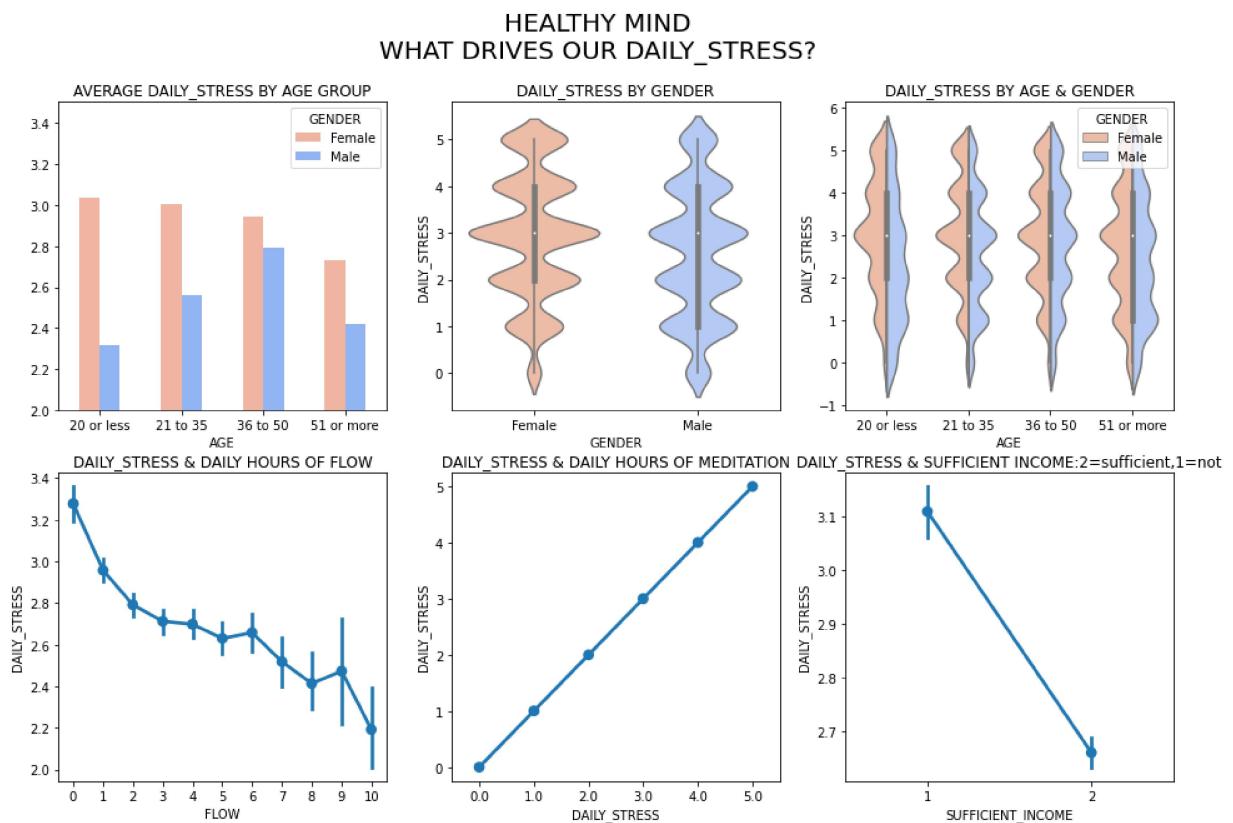
In [21]: # *HEALTHY MIND*

```
f,ax = plt.subplots(2,3,figsize=(16,10))
ax[0,0].set_title('AVERAGE DAILY_STRESS BY AGE GROUP')
ax[0,1].set_title('DAILY_STRESS BY GENDER')
ax[0,2].set_title('DAILY_STRESS BY AGE & GENDER')
ax[1,0].set_title('DAILY_STRESS & DAILY HOURS OF FLOW')
ax[1,1].set_title('DAILY_STRESS & DAILY HOURS OF MEDITATION')
ax[1,2].set_title('DAILY_STRESS & SUFFICIENT INCOME:2=sufficient,1=not')

ax[0,0].set_ylim([2, 3.5])
df3.plot(kind='bar', color=('darksalmon', 'cornflowerblue'), alpha=0.7, ax = ax[0,0])
ax[0,0].tick_params(axis='x', rotation=0)

sns.violinplot(x= 'GENDER',y='DAILY_STRESS', palette='coolwarm_r', data=df, ax = sns.violinplot(x = 'AGE', y = 'DAILY_STRESS', hue = 'GENDER', palette='coolwarm_r', order=['20 or less', '21 to 35', '36 to 50', '51 or more'], split=True)
sns.pointplot(x = 'FLOW', y = 'DAILY_STRESS', data=df, ax = ax[1,0])
sns.pointplot(x = 'DAILY_STRESS', y = 'DAILY_STRESS', data=df, ax = ax[1,1])
sns.pointplot(x = 'SUFFICIENT_INCOME', y = 'DAILY_STRESS', data=df, ax = ax[1,2])

f.suptitle('HEALTHY MIND\nWHAT DRIVES OUR DAILY_STRESS?', fontsize=20)
plt.show()
```



Experise

```
In [22]: df4 = df.pivot_table(values='ACHIEVEMENT', index=['AGE'], columns=['GENDER'], )
df4.head()
```

Out[22]:

GENDER	Female	Male
AGE		
20 or less	4.364571	4.163265
21 to 35	3.889252	3.677389
36 to 50	3.943297	4.069017
51 or more	3.964088	4.227790

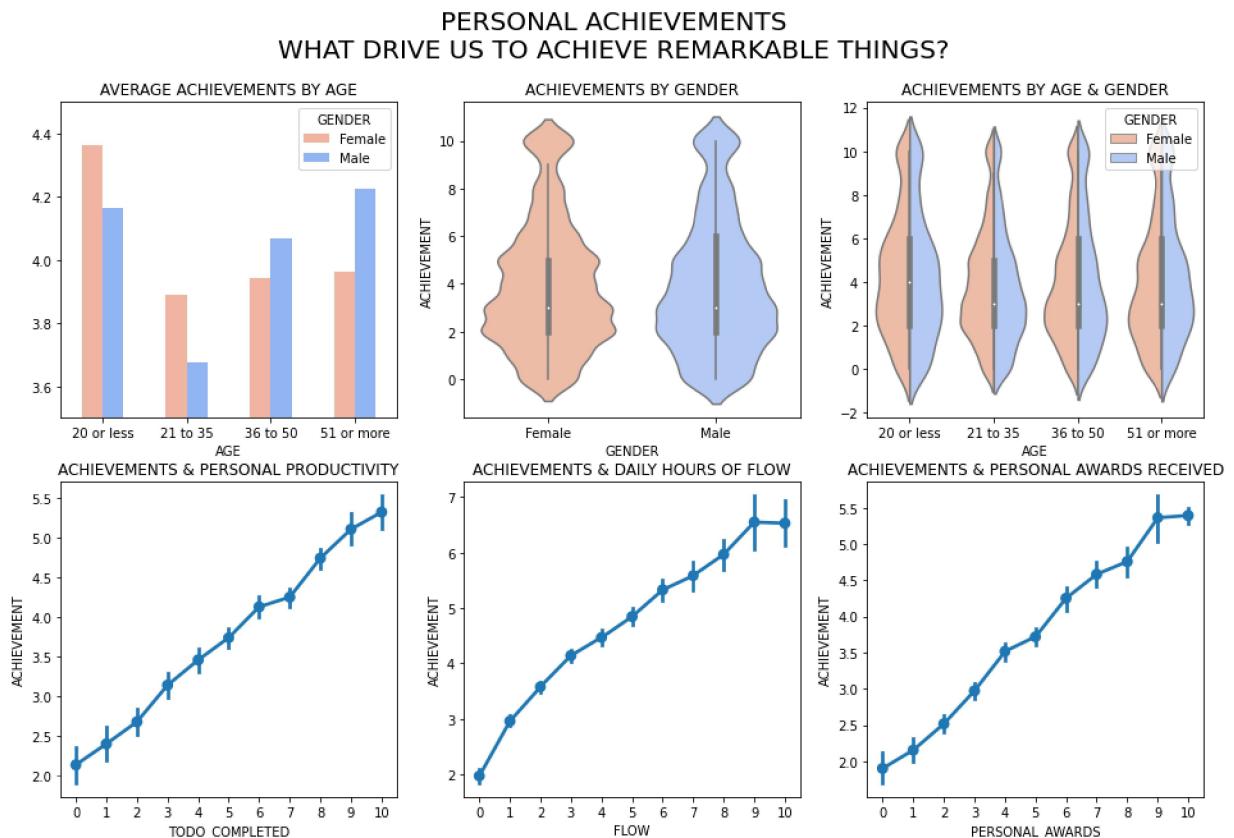
In [23]: # EXPERTISE

```
f,ax = plt.subplots(2,3,figsize=(16,10))
ax[0,0].set_title('AVERAGE ACHIEVEMENTS BY AGE')
ax[0,1].set_title('ACHIEVEMENTS BY GENDER')
ax[0,2].set_title('ACHIEVEMENTS BY AGE & GENDER')
ax[1,0].set_title('ACHIEVEMENTS & PERSONAL PRODUCTIVITY')
ax[1,1].set_title('ACHIEVEMENTS & DAILY HOURS OF FLOW')
ax[1,2].set_title('ACHIEVEMENTS & PERSONAL AWARDS RECEIVED')

ax[0,0].set_ylim([3.5, 4.5])
df4.plot(kind='bar', color=('darksalmon', 'cornflowerblue'), alpha=0.7, ax = ax[0,0])
ax[0,0].tick_params(axis='x', rotation=0)

sns.violinplot(x= 'GENDER',y='ACHIEVEMENT', palette='coolwarm_r', data=df, ax = ax[0,1])
sns.violinplot(x = 'AGE', y = 'ACHIEVEMENT', palette='coolwarm_r', hue = 'GENDER',
               order=['20 or less', '21 to 35', '36 to 50', '51 or more'], split=True)
sns.pointplot(x = 'TODO_COMPLETED', y = 'ACHIEVEMENT', data=df, ax = ax[1,0])
sns.pointplot(x = 'FLOW', y = 'ACHIEVEMENT', data=df, ax = ax[1,1])
sns.pointplot(x = 'PERSONAL_AWARDS', y = 'ACHIEVEMENT', data=df, ax = ax[1,2])

f.suptitle('PERSONAL ACHIEVEMENTS\nWHAT DRIVE US TO ACHIEVE REMARKABLE THINGS?', 
plt.show()
```



Connection

```
In [24]: df5 = df.pivot_table(values='CORE_CIRCLE', index=['AGE'], columns=['GENDER'], )
df5.head()
```

Out[24]:

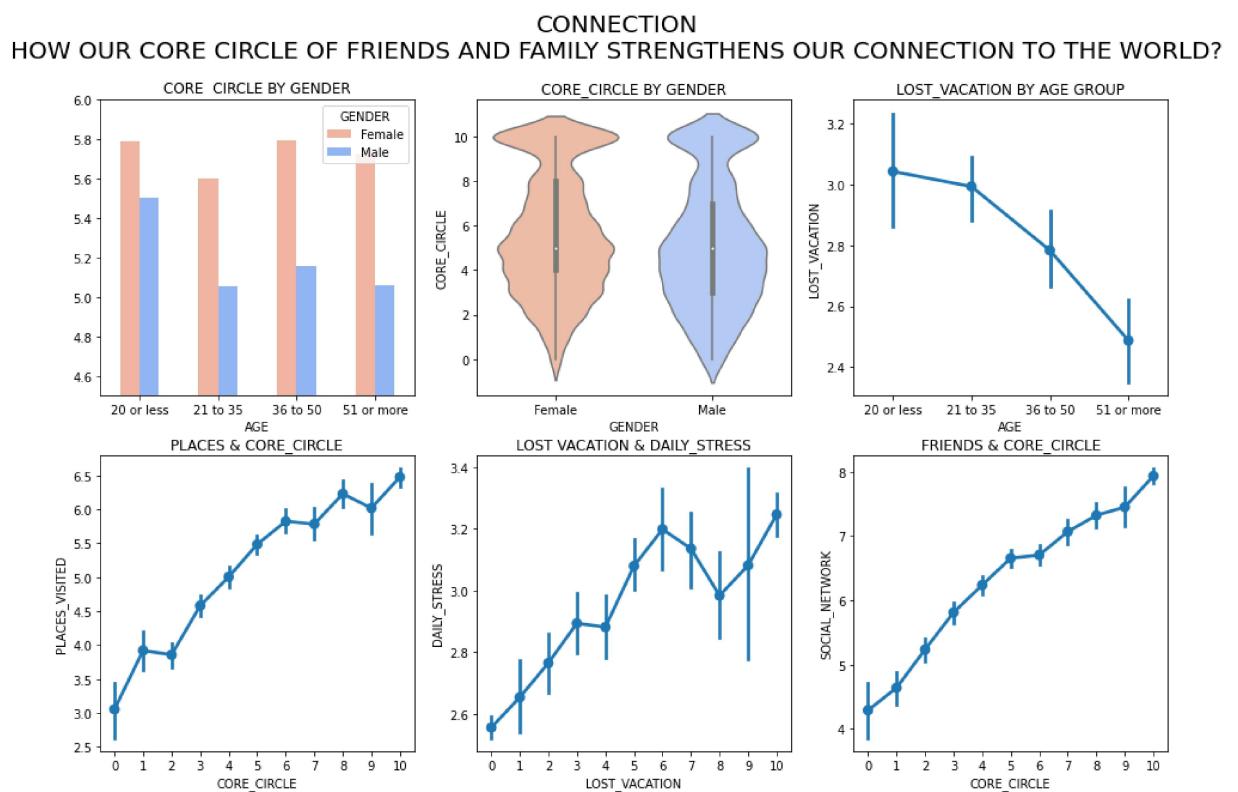
GENDER	Female	Male
AGE		
20 or less	5.787429	5.501458
21 to 35	5.603123	5.052681
36 to 50	5.796044	5.155289
51 or more	5.720442	5.060364

```
In [25]: # CONNECTION
f,ax = plt.subplots(2,3,figsize=(16,10))
ax[0,0].set_title('CORE CIRCLE BY GENDER')
ax[0,1].set_title('CORE_CIRCLE BY GENDER')
ax[0,2].set_title('LOST_VACATION BY AGE GROUP')
ax[1,0].set_title('PLACES & CORE_CIRCLE')
ax[1,1].set_title('LOST VACATION & DAILY_STRESS')
ax[1,2].set_title('FRIENDS & CORE_CIRCLE')

ax[0,0].set_ylim([4.5, 6])
df5.plot(kind='bar', color=('darksalmon', 'cornflowerblue'), alpha=0.7, ax = ax[0,0])
ax[0,0].tick_params(axis='x', rotation=0)

sns.violinplot(x= 'GENDER',y='CORE_CIRCLE', palette='coolwarm_r', data=df, ax = ax[0,1])
sns.pointplot(x = 'AGE', y = 'LOST_VACATION',order=['20 or less', '21 to 35', '36 to 50', '51 or more'], data=df, ax = ax[0,2])
sns.pointplot(x = 'CORE_CIRCLE', y = 'PLACES_VISITED', data=df, ax = ax[1,0])
sns.pointplot(x = 'LOST_VACATION', y = 'DAILY_STRESS', data=df, ax = ax[1,1])
sns.pointplot(x = 'CORE_CIRCLE', y = 'SOCIAL_NETWORK', data=df, ax = ax[1,2])

f.suptitle('CONNECTION\nHOW OUR CORE CIRCLE OF FRIENDS AND FAMILY STRENGTHENS OUR CONNECTION TO THE WORLD')
plt.show()
```



Passion

```
In [26]: df6 = df.pivot_table(values='TIME_FOR_PASSION', index=['AGE'], columns=['GENDER'])
df6.head()
```

Out[26]:

GENDER	Female	Male
AGE		
20 or less	3.172571	3.699708
21 to 35	3.275236	3.234033
36 to 50	3.105055	3.130533
51 or more	3.353039	3.525057

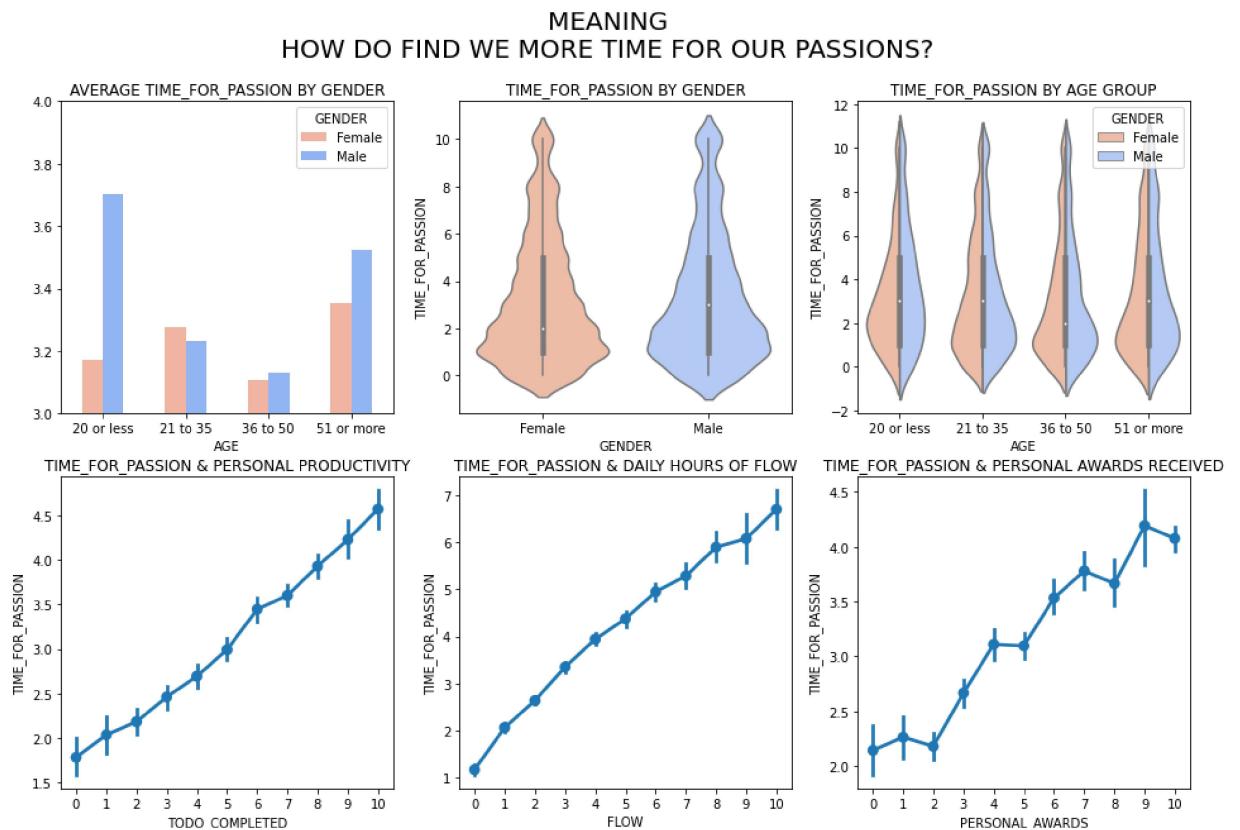
In [27]: # PASSION

```
f,ax = plt.subplots(2,3,figsize=(16,10))
ax[0,0].set_title('AVERAGE TIME_FOR_PASSION BY GENDER')
ax[0,1].set_title('TIME_FOR_PASSION BY GENDER')
ax[0,2].set_title('TIME_FOR_PASSION BY AGE GROUP')
ax[1,0].set_title('TIME_FOR_PASSION & PERSONAL PRODUCTIVITY')
ax[1,1].set_title('TIME_FOR_PASSION & DAILY HOURS OF FLOW')
ax[1,2].set_title('TIME_FOR_PASSION & PERSONAL AWARDS RECEIVED')

ax[0,0].set_ylim([3, 4])
df6.plot(kind='bar', color=('darksalmon', 'cornflowerblue'), alpha=0.7, ax = ax[0,0])
ax[0,0].tick_params(axis='x', rotation=0)

sns.violinplot(x= 'GENDER',y='TIME_FOR_PASSION', palette='coolwarm_r', data=df, ax=ax[0,1])
sns.violinplot(x = 'AGE', y = 'TIME_FOR_PASSION', palette='coolwarm_r', hue = 'GENDER', order=['20 or less', '21 to 35', '36 to 50', '51 or more'], split=True)
sns.pointplot(x = 'TODO_COMPLETED', y = 'TIME_FOR_PASSION', data=df, ax = ax[1,0])
sns.pointplot(x = 'FLOW', y = 'TIME_FOR_PASSION', data=df, ax = ax[1,1])
sns.pointplot(x = 'PERSONAL_AWARDS', y = 'TIME_FOR_PASSION', data=df, ax = ax[1,2])

f.suptitle('MEANING\nHOW DO FIND WE MORE TIME FOR OUR PASSIONS?', fontsize=20)
plt.show()
```



In []:

