lab2

April 15, 2022

1 lab 2

Amit Avigdor 316178144 & Barak Bonker 316177708

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

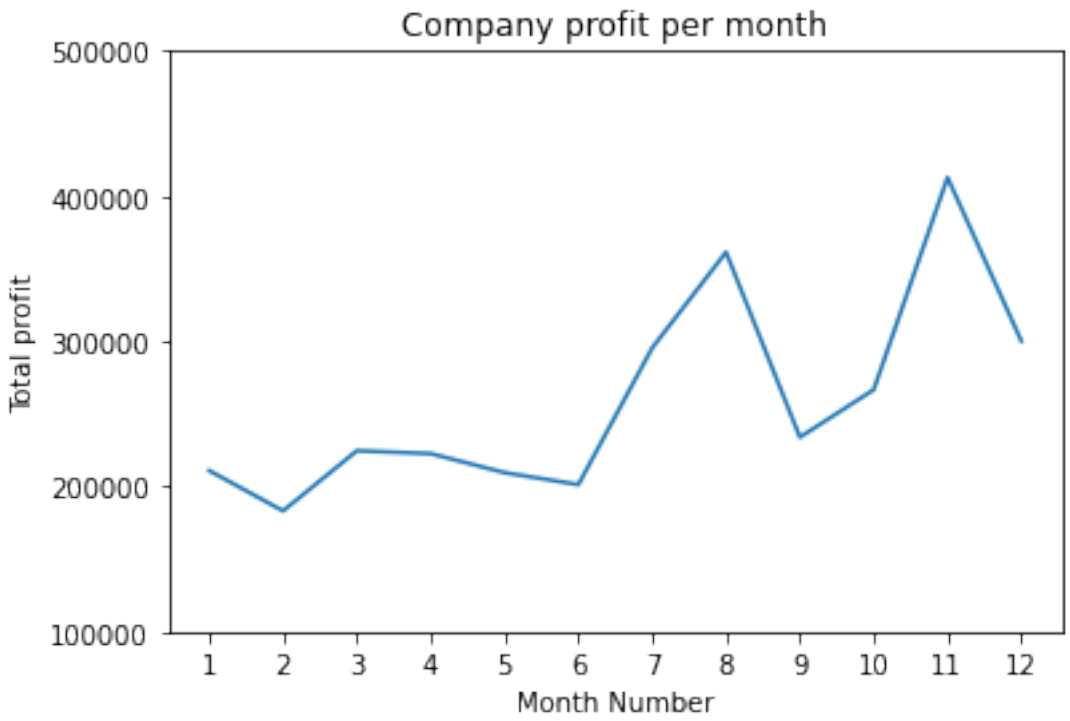
Task 1

[ ]:
```
file = pd.read_csv('company_sales_data.csv')
file.head()
```

[ ]:

	month_number	facecream	facewash	toothpaste	bathingsoap	shampoo	moisturizer	total_units	total_profit
0	1	2500	1500	5200	9200	1200	1500	21100	211000
1	2	2630	1200	5100	6100	2100	1200	18330	183300
2	3	2140	1340	4550	9550	3550	1340	22470	224700
3	4	3400	1130	5870	8870	1870	1130	22270	222700
4	5	3600	1740	4560	7760	1560	1740	20960	209600

[ ]:
```
plt.plot(file["month_number"],file["total_profit"])
plt.title("Company profit per month")
plt.xlabel("Month Number")
plt.ylabel("Total profit")
plt.yticks(range(100000,600000,100000))
plt.xticks(range(1,13))
plt.show()
```



Task 2

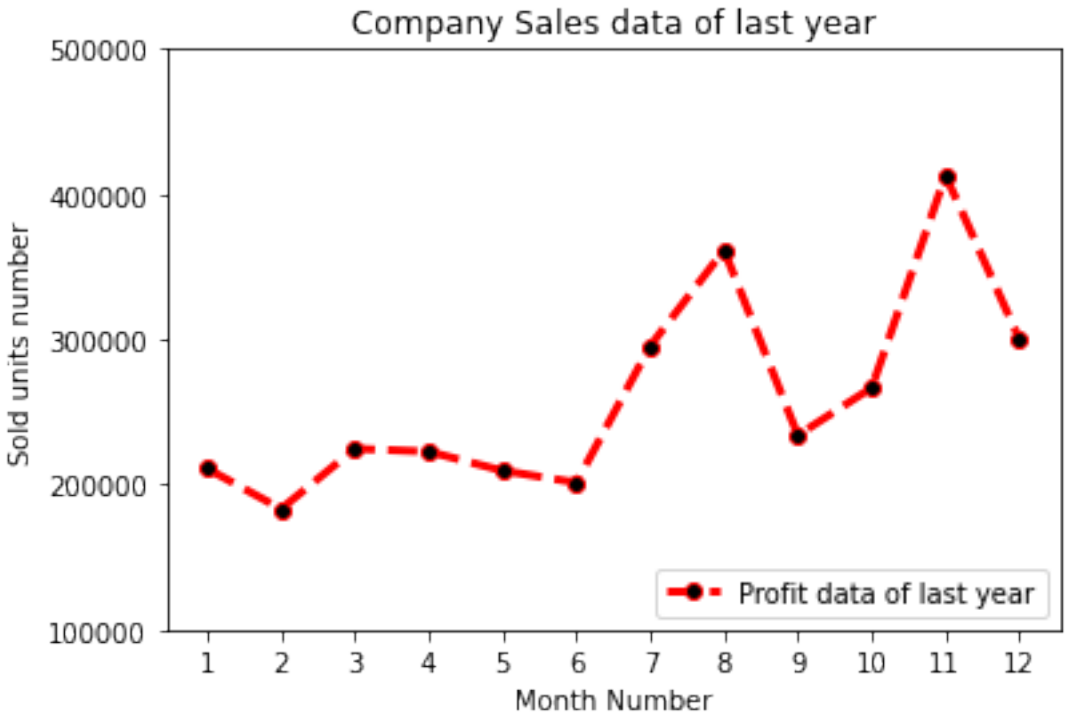
[ ]:
```
file = pd.read_csv('company_sales_data.csv')
file.head()
```

[ ]:

	month_number	facecream	facewash	toothpaste	bathingsoap	shampoo	moisturizer	total_units	total_profit
0	1	2500	1500	5200	9200	1200	1500	21100	211000
1	2	2630	1200	5100	6100	2100	1200	18330	183300
2	3	2140	1340	4550	9550	3550	1340	22470	224700
3	4	3400	1130	5870	8870	1870	1130	22270	222700
4	5	3600	1740	4560	7760	1560	1740	20960	209600

[ ]:
```
plt.plot(file["month_number"],file["total_profit"], linestyle='dashed',␣
↪linewidth=3, marker='o', color="r", markerfacecolor='k', label="Profit data␣
↪of last year")
plt.title("Company Sales data of last year")
plt.xlabel("Month Number")
```

```
plt.ylabel("Sold units number")
plt.yticks(range(100000,600000,100000))
plt.xticks(range(1,13))
plt.legend(loc=4)
plt.show()
```



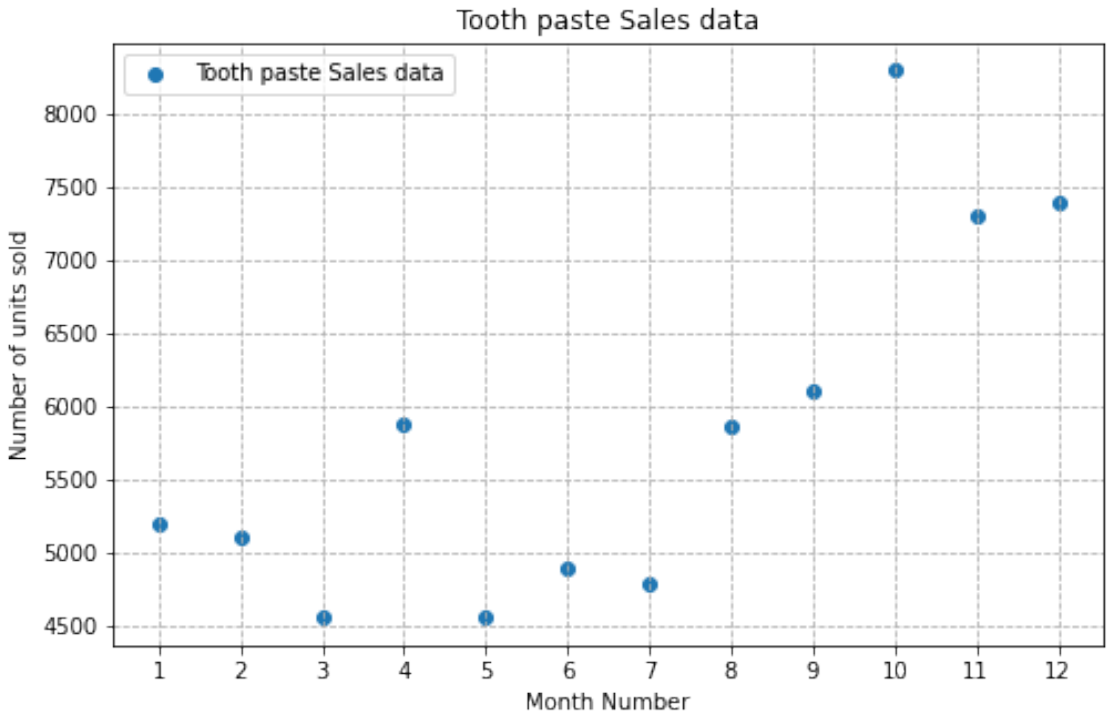
Task 3

[ ]:
```
plt.plot(file["month_number"],file["facecream"],linewidth=3, marker='o',␣
 ↪color="tab:blue", label="Face cream Sales Data")
plt.plot(file["month_number"],file["facewash"],linewidth=3, marker='o',␣
 ↪color="tab:orange", label="Face Wash Sales Data")
plt.plot(file["month_number"],file["toothpaste"],linewidth=3, marker='o',␣
 ↪color="green", label="ToothPaste Sales Data")
plt.plot(file["month_number"],file["bathingsoap"],linewidth=3, marker='o',␣
 ↪color="red", label="ToothPaste Sales Data")
plt.plot(file["month_number"],file["shampoo"],linewidth=3, marker='o',␣
 ↪color="mediumpurple", label="ToothPaste Sales Data")
plt.plot(file["month_number"],file["moisturizer"],linewidth=3, marker='o',␣
 ↪color="tab:brown", label="ToothPaste Sales Data")
plt.title("Sales data")
plt.xlabel("Month Number")
plt.ylabel("Total units in number")
```

```
plt.yticks([1000,2000,4000,6000,8000,10000,12000,15000,18000])
plt.xticks(range(1,13))
plt.legend()
plt.show()
```

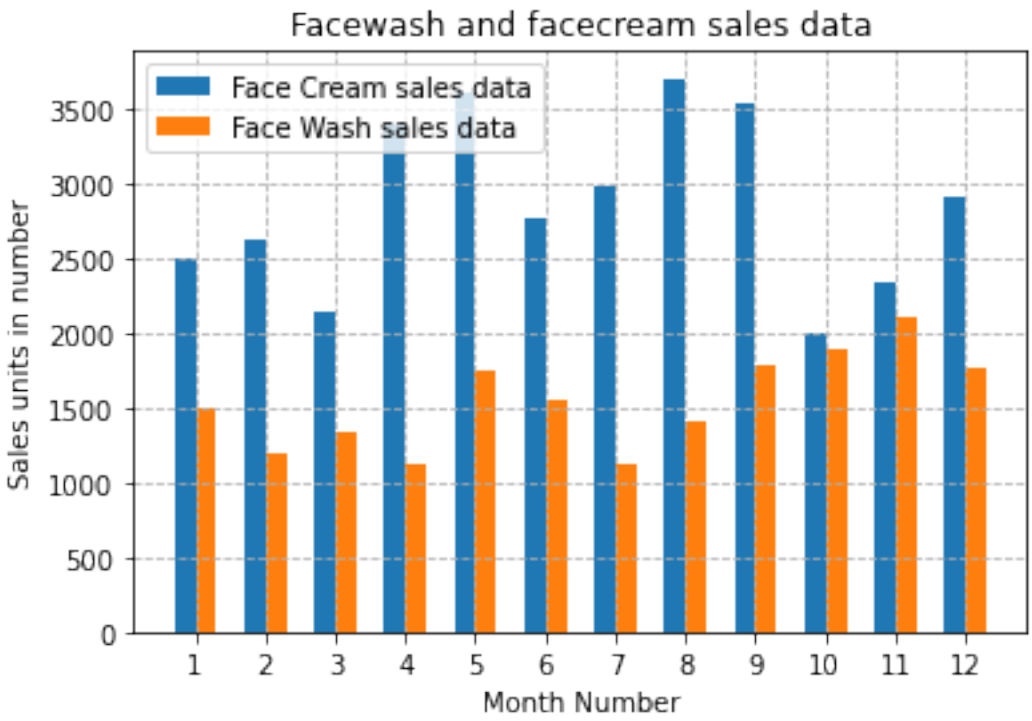
Task 4

[ ]:
```
plt.figure(figsize=(8,5))
plt.scatter(file["month_number"],file["toothpaste"], label="Tooth paste Sales␣
↪data")
plt.title("Tooth paste Sales data")
plt.xlabel("Month Number")
plt.ylabel("Number of units sold")
plt.grid(linestyle='--')
plt.xticks(range(1,13))
plt.legend(loc=2)
plt.show()
```



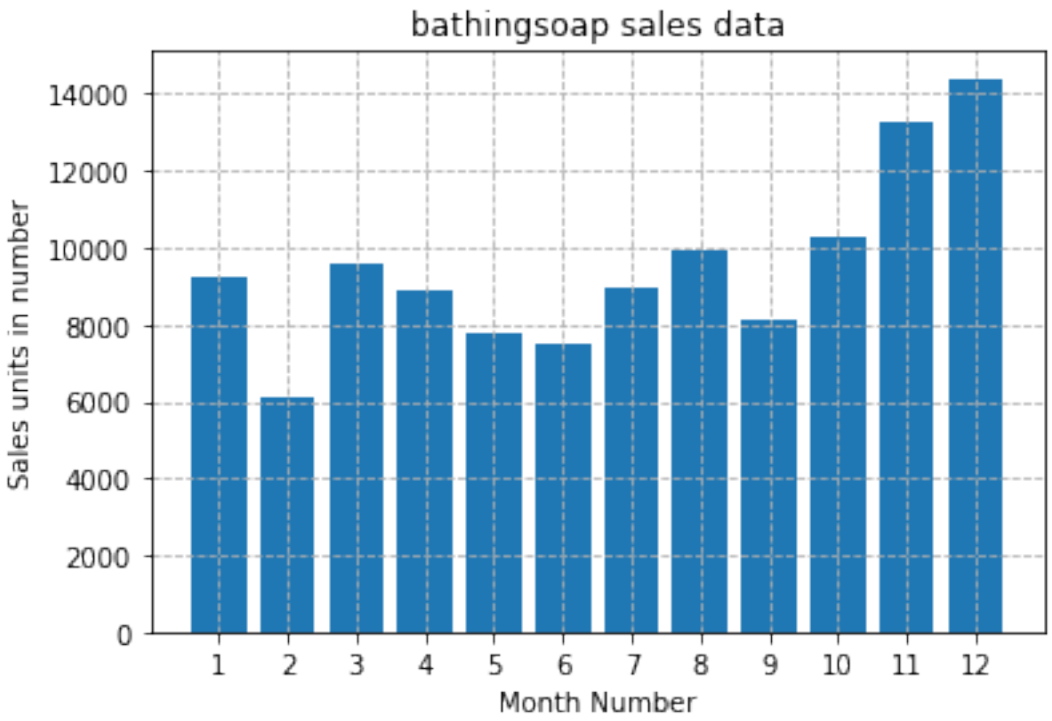
Task 5

```
[ ]: plt.bar(file["month_number"]-0.15,file["facecream"],width = 0.3,label="Face␣
      ↪Cream sales data")
     plt.bar(file["month_number"]+0.15,file["facewash"],width = 0.3,label="Face Wash␣
      ↪sales data")
     plt.title("Facewash and facecream sales data")
     plt.xlabel("Month Number")
     plt.ylabel("Sales units in number")
     plt.grid(linestyle='--')
     plt.xticks(range(1,13))
     plt.yticks(range(0,4000,500))
     plt.legend(loc=2)
     plt.show()
```



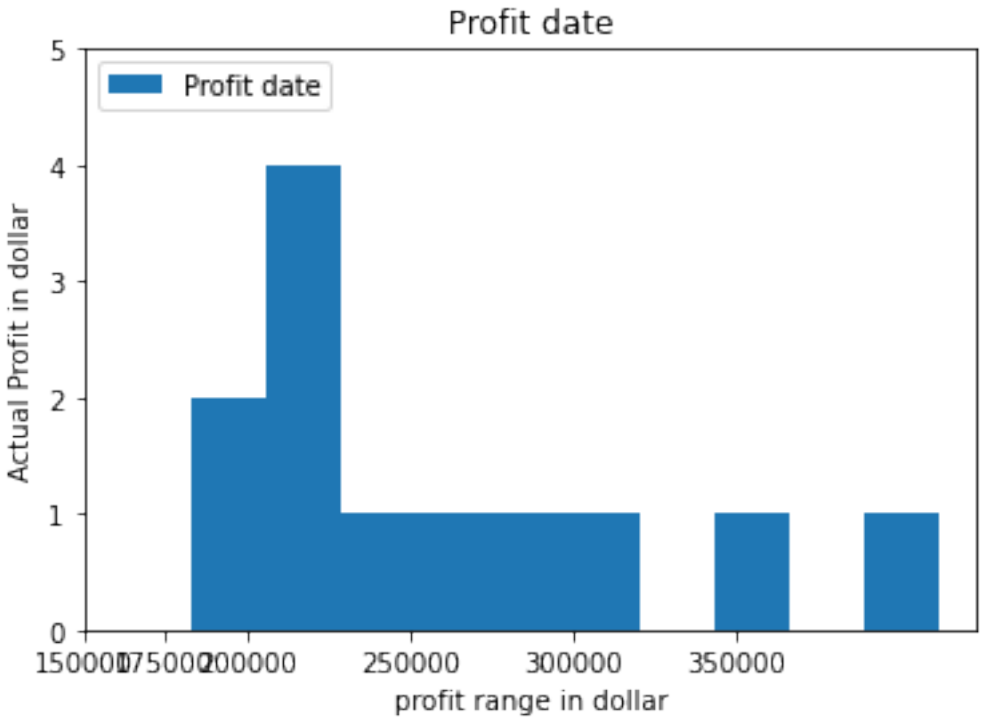
Task 6

[ ]:
```
plt.bar(file["month_number"],file["bathingsoap"])
plt.title("bathingsoap sales data")
plt.xlabel("Month Number")
plt.ylabel("Sales units in number")
plt.grid(linestyle='--')
plt.xticks(range(1,13))
plt.savefig('bathingsoap sales data.png',dpi=400)
plt.show()
```



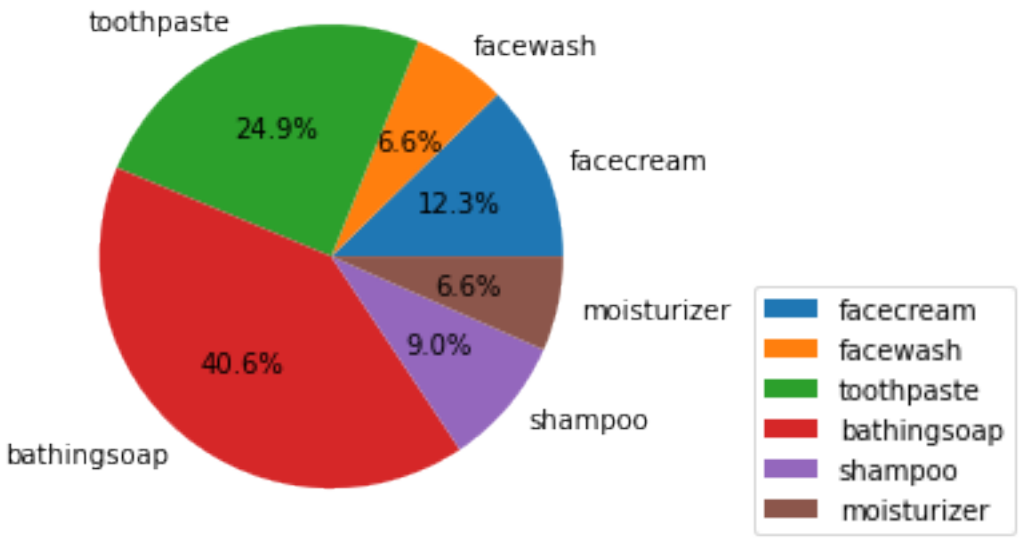
Task 7

[ ]:
```
plt.hist(file["total_profit"],label="Profit date")
plt.title("Profit date")
plt.xlabel("profit range in dollar")
plt.ylabel("Actual Profit in dollar")
plt.xticks([150000,175000,200000,250000,300000,350000])
plt.yticks(range(0,6,1))
plt.legend(loc=2)
plt.show()
```



Task 8

```
[ ]: plt.pie(file.sum()[1:7].tolist(), labels = file.columns[1:7].tolist(),␣
     ↪autopct='%1.1f%%')
     plt.legend(bbox_to_anchor=(1.7,0), loc=4)
     plt.show()
```



Task 9

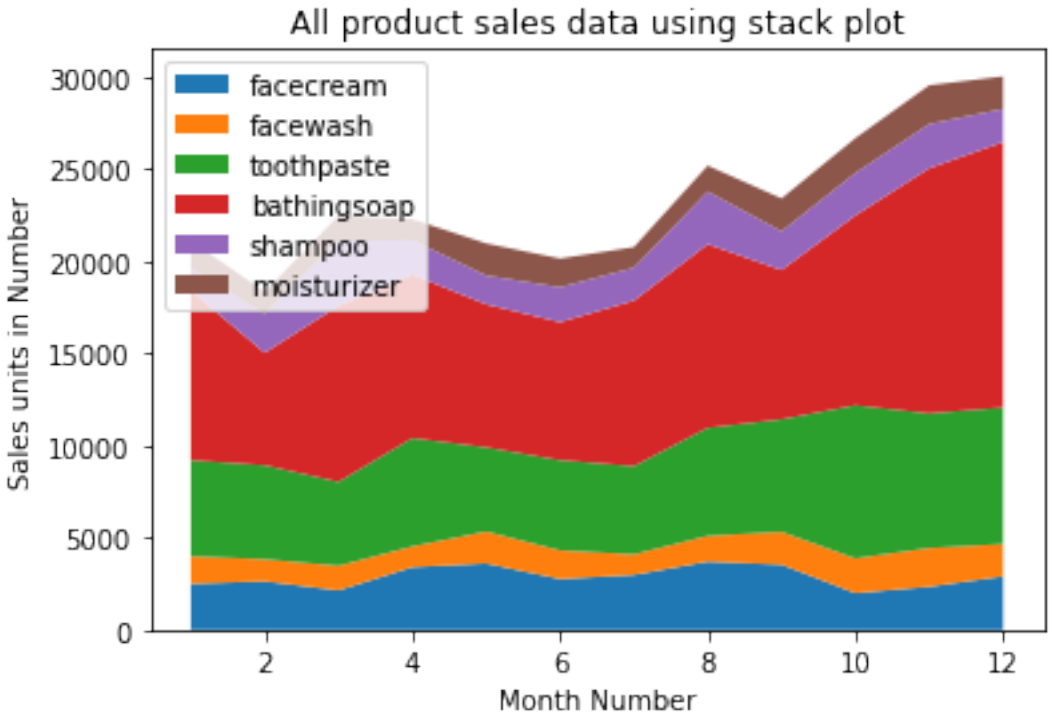
```
[ ]: fig, axs = plt.subplots(2, sharex=True)
plt.xlabel("Month Number")
plt.ylabel("Sales units in number")
axs[0].plot(file["month_number"],file["bathingsoap"],linewidth=3, marker='o',␣
 ↪color="black")
axs[0].set_title("Sales data of a Bathingsoap")
plt.xticks(range(1,13))
axs[1].plot(file["month_number"],file["facewash"],linewidth=3, marker='o',␣
 ↪color="red")
axs[1].set_title("Sales data of a facewash")
```

```
[ ]: Text(0.5, 1.0, 'Sales data of a facewash')
```

Task 10

```
[ ]: data = []
for i in range(1,7):
    data.append(file[file.columns[i]].tolist())
plt.stackplot(file["month_number"], data, labels = file.columns[1:7].tolist())
plt.title("All product sales data using stack plot")
```

```
plt.xlabel("Month Number")
plt.ylabel("Sales units in Number")
plt.yticks(range(0,30001,5000))
plt.legend(loc=2)
plt.show()
```



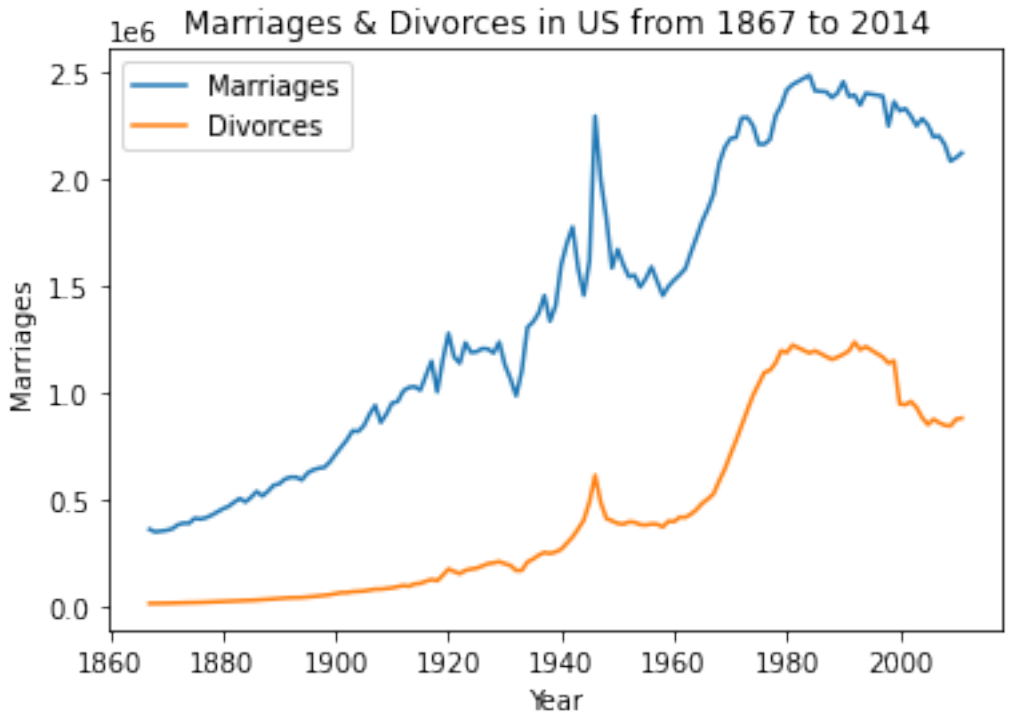
Task 11

[ ]:
```
import seaborn as sns
file = pd.read_csv('us-marriages-divorces-1867-2014.csv')
print(file.head())
```

```
   Year  Marriages  Divorces  Population  Marriages_per_1000  \
0  1867   357000.0   10000.0    36970000                 9.7
1  1868   345000.0   10000.0    37885000                 9.1
2  1869   348000.0   11000.0    38870000                 9.0
3  1870   352000.0   11000.0    39905000                 8.8
4  1871   359000.0   12000.0    41010000                 8.8

   Divorces_per_1000
0                0.3
1                0.3
2                0.3
3                0.3
```

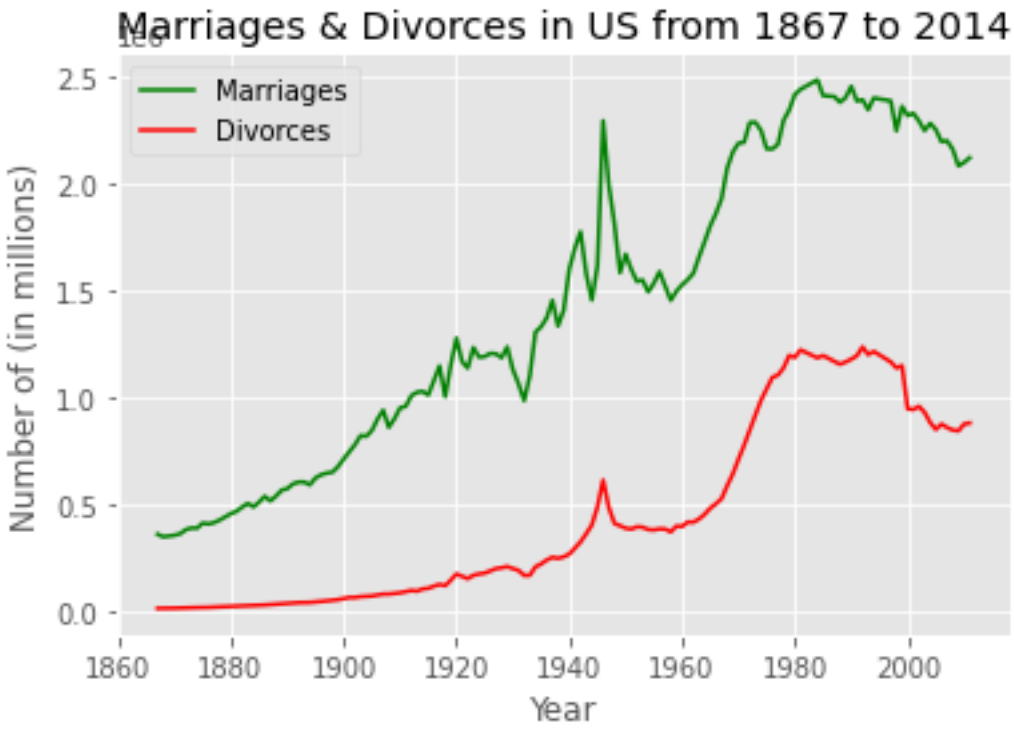
```
4                    0.3
```

[ ]:
```
sns.lineplot(x="Year",y="Marriages",label ="Marriages",data=file).
 ↪set(title="Marriages & Divorces in US from 1867 to 2014")
sns.lineplot(x="Year",y="Divorces",label ="Divorces",data=file)
plt.show()
```



Task 12

[ ]:
```
file = file.dropna()
plt.style.use('ggplot')
plt.plot(file['Year'], file['Marriages'], "-g", label="Marriages")
plt.plot(file['Year'], file['Divorces'], "-r", label="Divorces")
plt.title("Marriages & Divorces in US from 1867 to 2014")
plt.xlabel("Year")
plt.ylabel("Number of (in millions)")
plt.legend()
plt.show()
```



Task 13

[ ]:
```
from bokeh.io import output_file, show
from bokeh.models import ColumnDataSource
from bokeh.plotting import figure
from bokeh.transform import dodge

output_file("bars.html")
Y1900=file.loc[file['Year']== 1900]
Y1950=file.loc[file['Year']== 1950]
Y2000=file.loc[file['Year']== 2000]

Years = ["1900","1950","2000"]
status = ['Marriages', 'Divorces']

data = {'Years' : Years,
        'Marriages'   :␣
↪[Y1900["Marriages"],Y1950["Marriages"],Y2000["Marriages"]],
        'Divorces'    : [Y1900["Divorces"],Y1950["Divorces"],Y2000["Divorces"]]}

source = ColumnDataSource(data=data)
```

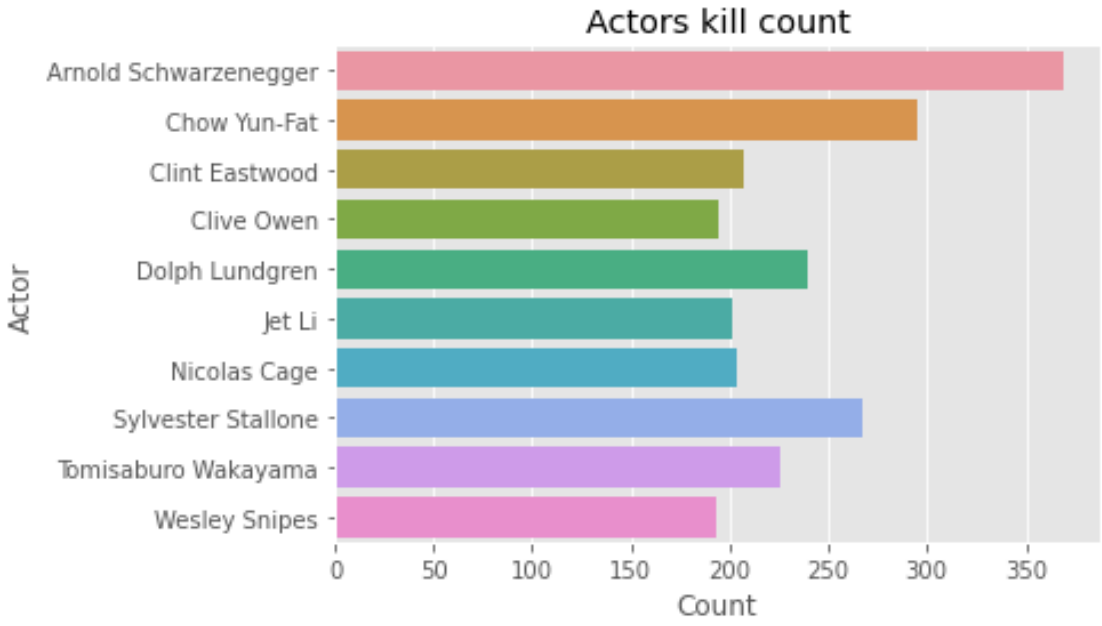
```
p = figure(x_range=Years, height=250, title="comparing the number of marriages␣
 ↪and divorces per capita in the U.S.",
           toolbar_location=None, tools="")

p.vbar(x=dodge('Years', -0.1, range=p.x_range), top='Marriages', width=0.2,␣
 ↪source=source,
       color="#c9d9d3", legend_label="Marriages")

p.vbar(x=dodge('Years',  0.1,  range=p.x_range), top='Divorces', width=0.2,␣
 ↪source=source,
       color="#718dbf", legend_label="Divorces")

p.x_range.range_padding = 0.1
p.xgrid.grid_line_color = None
p.legend.location = "top_left"
p.legend.orientation = "horizontal"

show(p)
```

Task 14

[ ]:
```
import seaborn as sns
file = pd.read_csv('actor_kill_counts.csv')
file
```

[ ]:
```
                  Actor  Count
0  Arnold Schwarzenegger    369
1           Chow Yun-Fat    295
2         Clint Eastwood    207
3             Clive Owen    194
4         Dolph Lundgren    239
5                Jet Li    201
6           Nicolas Cage    204
7     Sylvester Stallone    267
8    Tomisaburo Wakayama    226
9          Wesley Snipes    193
```

[ ]:
```
sns.barplot(x=file['Count'], y=file["Actor"])
plt.title("Actors kill count")
plt.show()
```



Task 15

[ ]:
```
file = pd.read_csv("roman-emperor-reigns.csv")
file
```

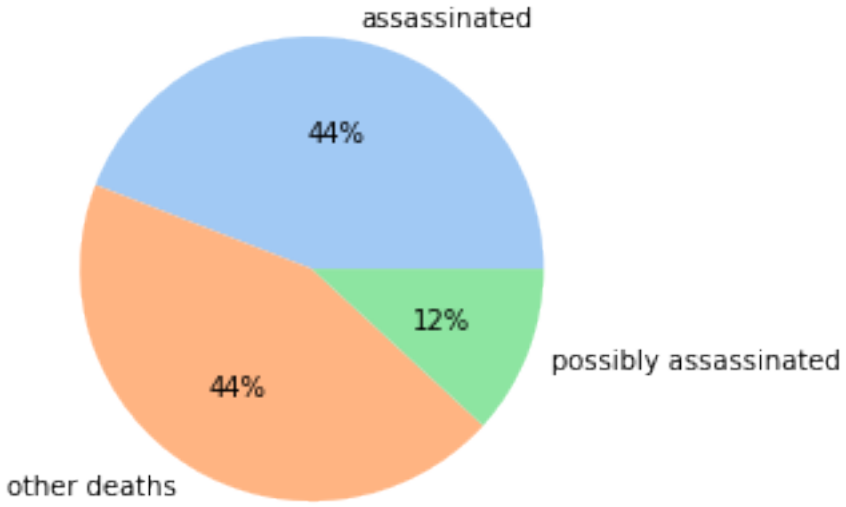
[ ]:

	Emperor	Length_of_Reign	Cause_of_Death
0	Augustus	40.58	Possibly assassinated
1	Tiberius	22.50	Possibly assassinated
2	Caligula	4.83	Assassinated
3	Claudius	13.75	Possibly assassinated
4	Nero	13.67	Suicide
..	…	…	…
63	Valentinian I	11.00	Natural causes
64	Valens	14.00	Killed in battle
65	Gratian	16.00	Assassinated
66	Valentinian II	17.00	Possibly assassinated
67	Theodosius I	16.00	Natural causes

[68 rows x 3 columns]

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


assassinated_emperors = file[file['Cause_of_Death'].apply(lambda x:␣
↪'assassinated' in x.lower())]
```

```
pro_assassinated_emperors = file[file['Cause_of_Death'].apply(lambda x:␣
 ↪'possibly assassinated' in x.lower())]
number_assassinated = len(assassinated_emperors)
pro_assassinated_emperors = len(pro_assassinated_emperors)
other_deaths = len(file) - number_assassinated - pro_assassinated_emperors
data = [number_assassinated, other_deaths, pro_assassinated_emperors]
labels = ['assassinated', 'other deaths', 'possibly assassinated']
colors = sns.color_palette('pastel')[0:5]
plt.pie(data, labels = labels, colors = colors, autopct='%.0f%%')
plt.show()
```



Task 16

[ ]:
```
from bokeh.plotting import figure, show, output_notebook
file = pd.read_csv('arcade-revenue-vs-cs-doctorates.csv')
file
```

[ ]:

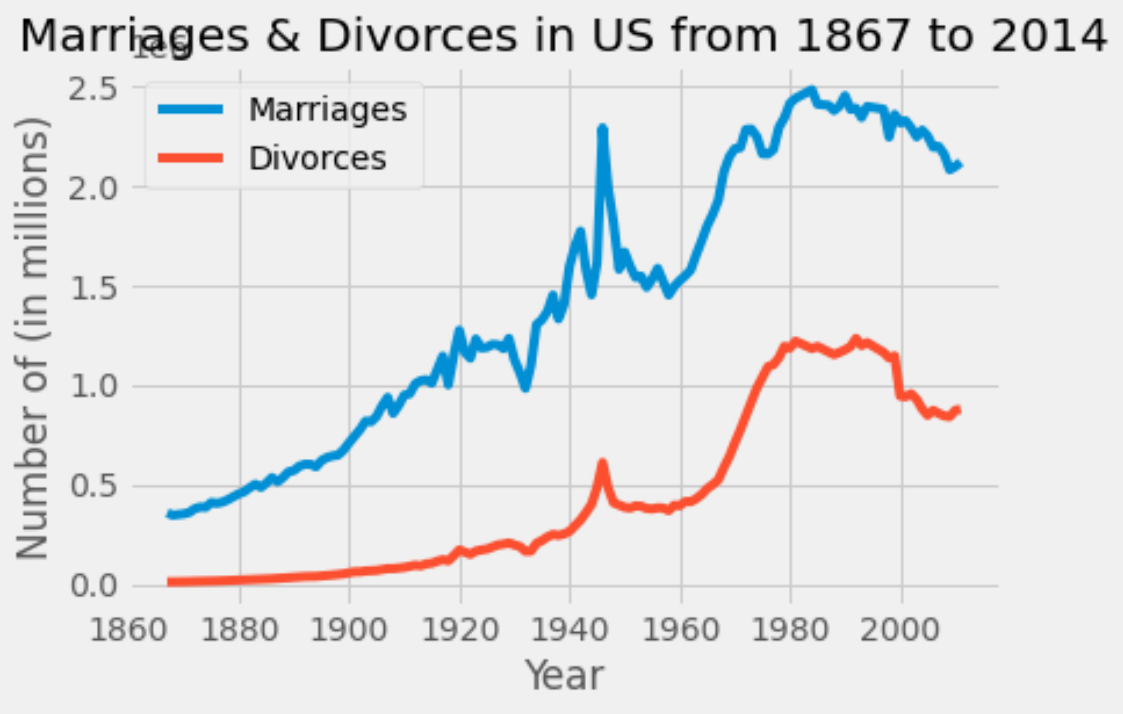
	Year	Total Arcade Revenue (billions)	\
0	2000	1.196	
1	2001	1.176	
2	2002	1.269	
3	2003	1.240	
4	2004	1.307	
5	2005	1.435	
6	2006	1.601	
7	2007	1.654	
8	2008	1.803	

```
9  2009                              1.734

   Computer Science Doctorates Awarded (US)
0                                        861
1                                        830
2                                        809
3                                        867
4                                        948
5                                       1129
6                                       1453
7                                       1656
8                                       1787
9                                       1611
```

[ ]:
```
graph = figure(title = "Arcade renenue to Computer Science Doctorates Awarded")
output_notebook()
graph.xaxis.axis_label = 'Total Arcade Revenue (billions)'
graph.yaxis.axis_label = 'Computer Science Doctorates Awarded (US)'
graph.circle(file["Total Arcade Revenue (billions)"], file["Computer Science␣
 ↪Doctorates Awarded (US)"], size=12)
show(graph)
```

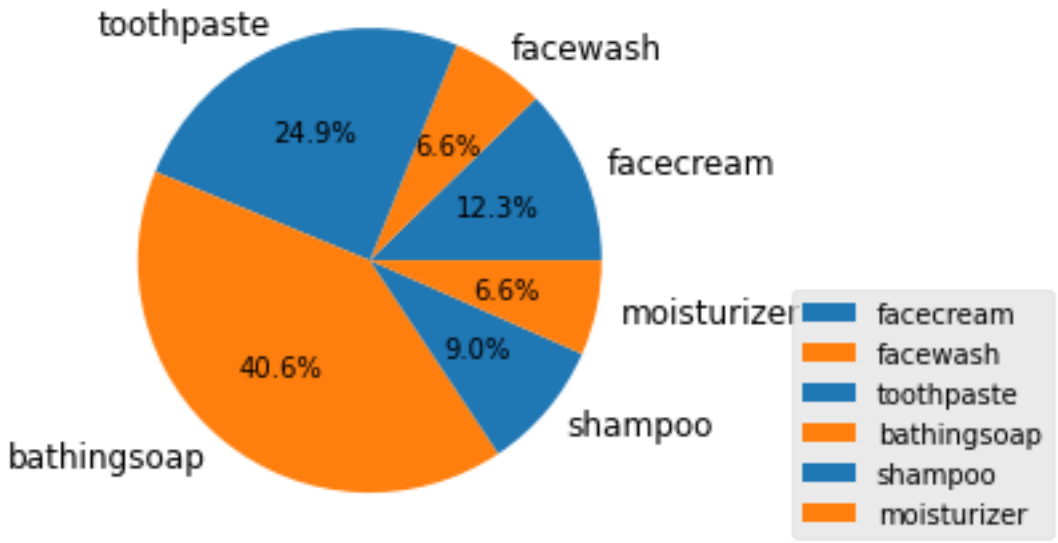
Task 17

[ ]:
```
import matplotlib.pyplot as plt
import numpy as np

file = file.dropna()
plt.style.use('fivethirtyeight')
plt.plot(file['Year'], file['Marriages'], label="Marriages")
plt.plot(file['Year'], file['Divorces'], label="Divorces")
plt.title("Marriages & Divorces in US from 1867 to 2014")
plt.xlabel("Year")
plt.ylabel("Number of (in millions)")
plt.legend()
plt.show()
```



Task 18 & 19

[ ]:
```
import matplotlib as mpl
file = pd.read_csv('company_sales_data.csv')
with mpl.rc_context({"lines.linewidth": 2.5, "axes.prop_cycle": mpl.
 ↪cycler(color=['#1f77b4', '#ff7f0e']), "xtick.labelsize": 12, "ytick.
 ↪labelsize": 12}):
    plt.pie(file.sum()[1:7].tolist(), labels = file.columns[1:7].tolist(),␣
 ↪autopct='%1.1f%%')
    plt.legend(bbox_to_anchor=(1.7,0), loc=4)
    plt.show()
```



Task 20

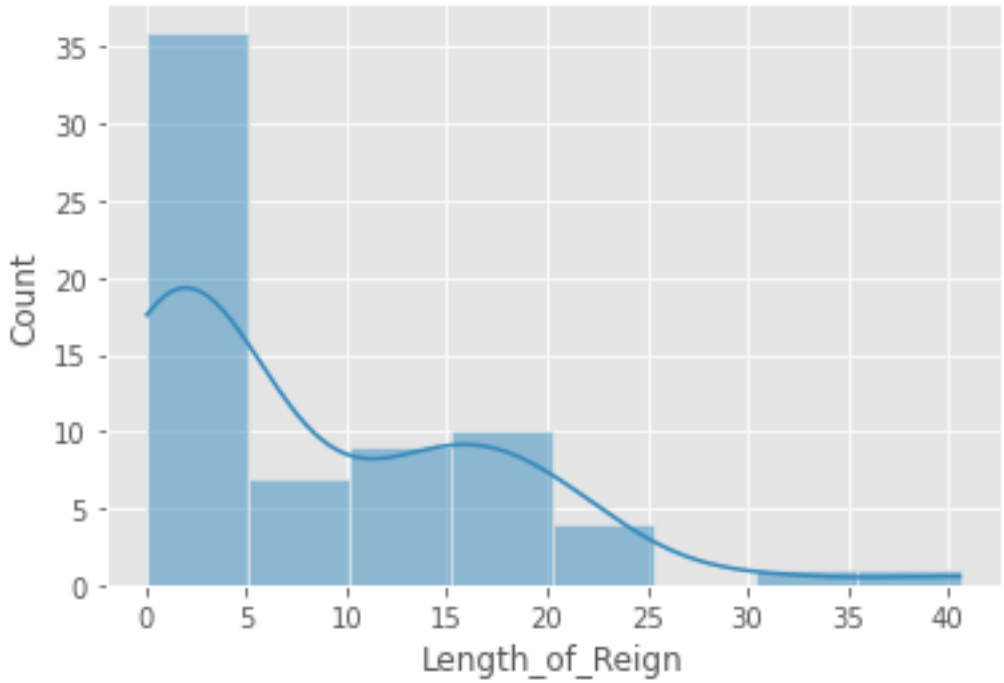
[ ]:
```
file = pd.read_csv('roman-emperor-reigns.csv')
file.head()
```

[ ]:

	Emperor	Length_of_Reign	Cause_of_Death
0	Augustus	40.58	Possibly assassinated
1	Tiberius	22.50	Possibly assassinated
2	Caligula	4.83	Assassinated
3	Claudius	13.75	Possibly assassinated
4	Nero	13.67	Suicide
..	…	…	…
63	Valentinian I	11.00	Natural causes
64	Valens	14.00	Killed in battle
65	Gratian	16.00	Assassinated
66	Valentinian II	17.00	Possibly assassinated
67	Theodosius I	16.00	Natural causes

[68 rows x 3 columns]

[ ]:
```
sns.histplot(data=file, x="Length_of_Reign", kde=True)
plt.show()
```



we can see that most of the emperors didnt reigned much, most of them even reigned less then 5 years and very little more then 25 years

Task 21

[39]:
```
file = pd.read_csv('recent-college-grads-earnings.csv')
file.head()
```

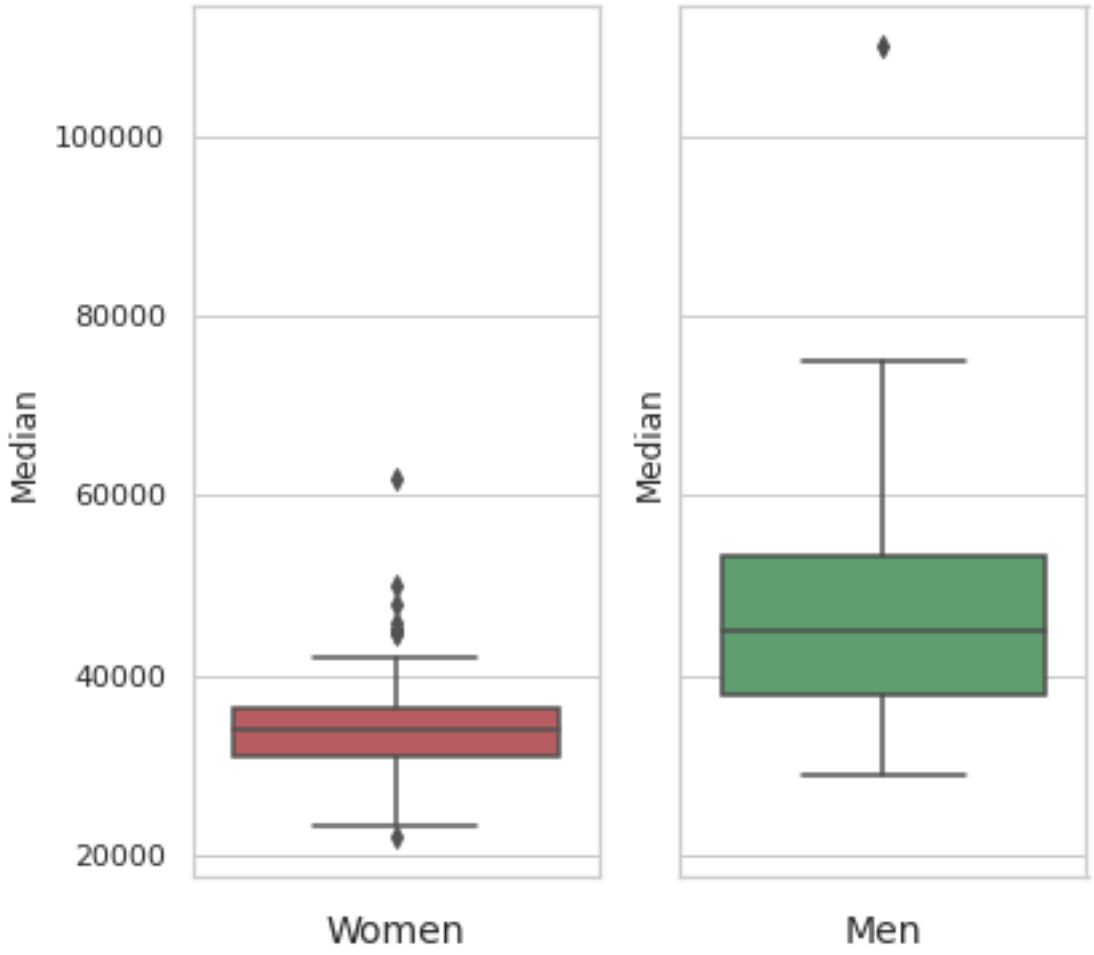
[39]:

	Rank	Major_code	Major	Total	Men	Women	Major_category	ShareWomen	Sample_size	Employed	…	Part_time	Full_time_year_round	Unemployed	Unemployment_rate	Median
0	1	2419	PETROLEUM ENGINEERING	2339.0	2057.0	282.0	Engineering	0.120564	36	1976	…	270	1207	37	0.018381	110000
1	2	2416	MINING AND MINERAL ENGINEERING	756.0	679.0	77.0	Engineering	0.101852	7	640	…	170	388	85	0.117241	75000
2	3	2415	METALLURGICAL ENGINEERING	856.0	725.0	131.0	Engineering	0.153037	3	648	…					
3	4	2417	NAVAL ARCHITECTURE AND MARINE ENGINEERING	1258.0	1123.0	135.0	Engineering	0.107313	16	758	…					
4	5	2405	CHEMICAL ENGINEERING	32260.0	21239.0	11021.0	Engineering	0.341631	289	25694	…					

```
2        133                   340          16           0.024096   73000
3        150                   692          40           0.050125   70000
4       5180                 16697        1672           0.061098   65000

   P25th   P75th  College_jobs  Non_college_jobs  Low_wage_jobs
0  95000  125000          1534               364            193
1  55000   90000           350               257             50
2  50000  105000           456               176              0
3  43000   80000           529               102              0
4  50000   75000         18314              4440            972

[5 rows x 21 columns]
```

[40]:
```
import seaborn as sns
f, axes = plt.subplots(1, 2,figsize=(6,6),sharey=True)
sns.set_theme(style="whitegrid")
womens_majors = file.loc[file["ShareWomen"] >= 0.5]
mans_majors=file.loc[file["ShareWomen"] < 0.5]
sns.boxplot(y=womens_majors["Median"], data=file,color="r",orient='v' ,␣
↪ax=axes[0]).set_xlabel("Women", fontsize=14)
sns.boxplot(y=mans_majors["Median"], data=file,color="g", orient='v' ,␣
↪ax=axes[1]).set_xlabel("Men", fontsize=14)
```

[40]:
```
Text(0.5, 0, 'Men')
```

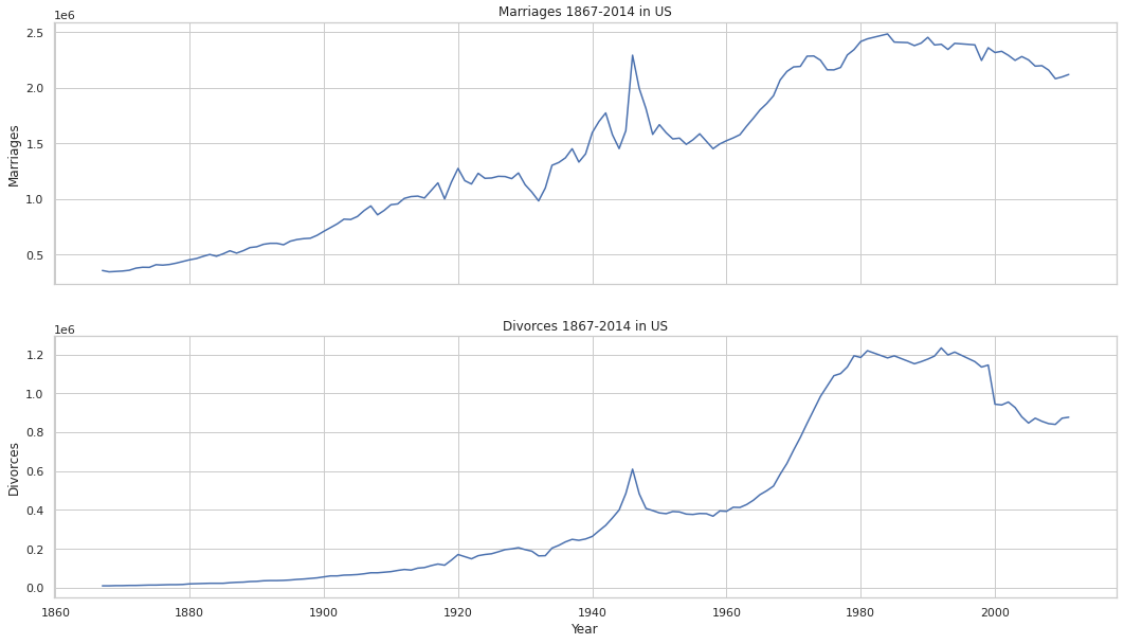


Yes, we can see a significant differences because of the gap between the median of the groups.

Task 22

[58]:
```
file = pd.read_csv('us-marriages-divorces-1867-2014.csv')
file = file.dropna()
fig, axes = plt.subplots(2, 1, figsize=(18, 10), sharex=True)

sns.lineplot(ax=axes[0], data=file, x='Year', y='Marriages')
sns.lineplot(ax=axes[1], data=file, x='Year', y='Divorces')

axes[0].set_title('Marriages 1867-2014 in US')
axes[1].set_title('Divorces 1867-2014 in US')
```

[58]:
```
Text(0.5, 1.0, 'Divorces 1867-2014 in US')
```



Task 23

```
[ ]: file = pd.read_csv('percent-degrees-conferred-women-usa.csv')
     file.head()
```

```
[ ]:    Year  Agriculture  Architecture  Art and Performance    Biology   Business  \
     0  1970     4.229798     11.921005                 59.7  29.088363   9.064439
     1  1971     5.452797     12.003106                 59.9  29.394403   9.503187
     2  1972     7.420710     13.214594                 60.4  29.810221  10.558962
     3  1973     9.653602     14.791613                 60.2  31.147915  12.804602
     4  1974    14.074623     17.444688                 61.9  32.996183  16.204850

        Communications and Journalism  Computer Science  Education  Engineering  \
     0                           35.3               13.6  74.535328          0.8
     1                           35.5               13.6  74.149204          1.0
     2                           36.6               14.9  73.554520          1.2
     3                           38.4               16.4  73.501814          1.6
     4                           40.5               18.9  73.336811          2.2

          English  Foreign Languages  Health Professions  Math and Statistics  \
     0  65.570923               73.8                77.1                 38.0
     1  64.556485               73.9                75.5                 39.0
     2  63.664263               74.6                76.9                 40.2
     3  62.941502               74.9                77.4                 40.9
     4  62.413412               75.3                77.9                 41.8

        Physical Sciences  Psychology  Public Administration  \
```

```
0               13.8        44.4                    68.4
1               14.9        46.2                    65.5
2               14.8        47.6                    62.6
3               16.5        50.4                    64.3
4               18.2        52.6                    66.1

   Social Sciences and History
0                         36.8
1                         36.2
2                         36.1
3                         36.4
4                         37.3
```

[ ]:
```
features = ['Agriculture', 'Architecture', 'Art and Performance', 'Biology',
       'Business', 'Communications and Journalism', 'Computer Science',
       'Education', 'Engineering', 'English', 'Foreign Languages',
       'Health Professions', 'Math and Statistics', 'Physical Sciences',
       'Psychology', 'Public Administration', 'Social Sciences and History']
```

[ ]:
```
plt.figure(figsize = (30, 40))
for i in enumerate(features):
    plt.subplot(3, 6,i[0]+1)
    sns.lineplot(x='Year',y=i[1], data = file)
    plt.xticks(rotation = 45)
    plt.ylim(0, 90)
```


Agriculture
Architecture
Art and Performance
Biology
Business
Communications and Journalism
Computer Science
Education
Engineering
English
Foreign Languages
Health Professions
Math and Statistics
Physical Sciences
Psychology
Public Administration
Social Sciences and History
Year