#·Lab03·-·Exploratory·Data·Analysis
##·Nguyen·Quoc·Tuan·-·19522476
##·Link·github:·https://github.com/tuNQws/data_mining.git

I. Matplotlib

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

```
import numpy as np
```

```
x = np.linspace(0,5,11)
y = x ** 2
```

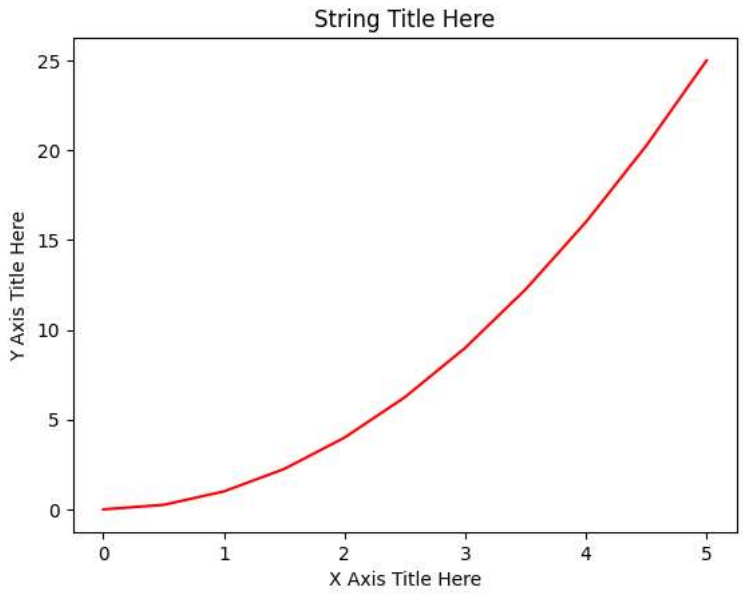
```
x
```

```
array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

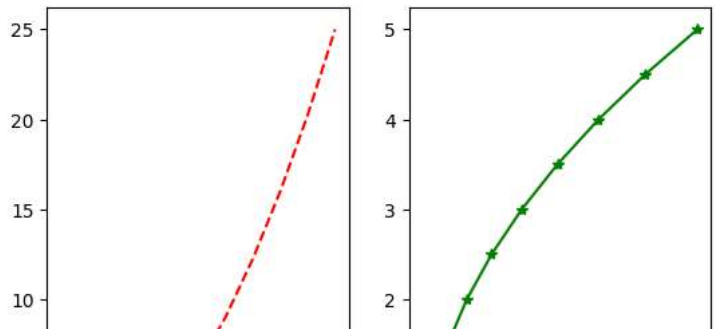
```
y
```

```
array([ 0.  ,  0.25,  1.  ,  2.25,  4.  ,  6.25,  9.  , 12.25, 16.  ,
       20.25, 25.  ])
```

```
plt.plot(x, y, 'r') # 'r' is the color red
plt.xlabel('X Axis Title Here')
plt.ylabel('Y Axis Title Here')
plt.title('String Title Here')
plt.show()
```

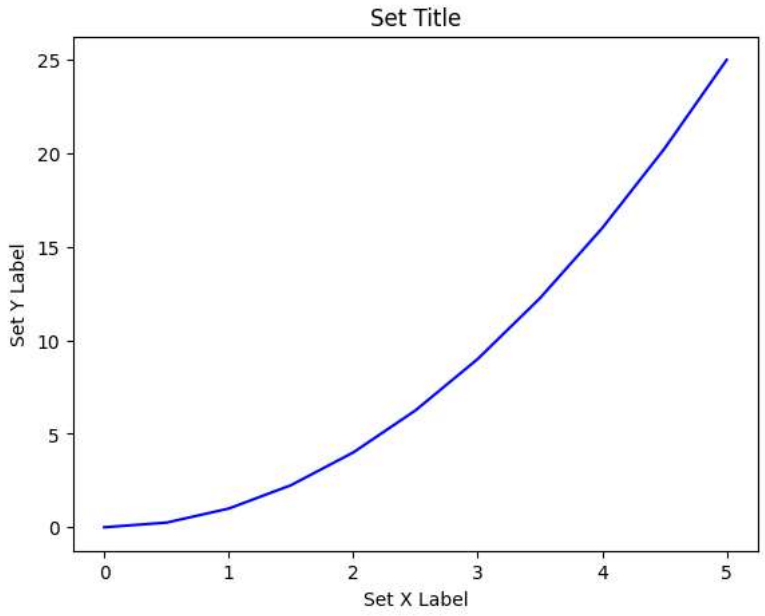



```
# plt.subplot(nrows, ncols, plot_number)
plt.subplot(1,2,1)
plt.plot(x,y,'r--')
plt.subplot(1,2,2)
plt.plot(y,x,'g*-');
```




```
fig = plt.figure()
axes = fig.add_axes([0.1, 0.1, 0.8, 0.8])
axes.plot(x,y,'b')
axes.set_xlabel('Set X Label')
axes.set_ylabel('Set Y Label')
axes.set_title('Set Title')
```

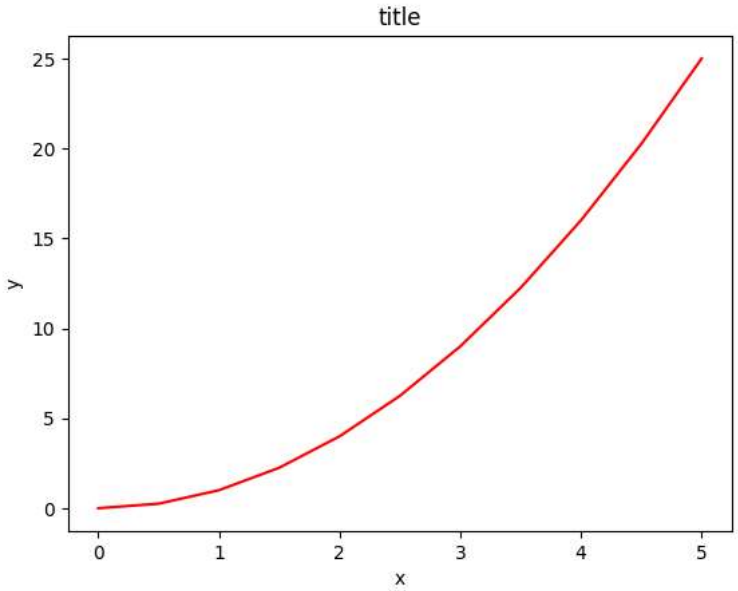
```
Text(0.5, 1.0, 'Set Title')
```



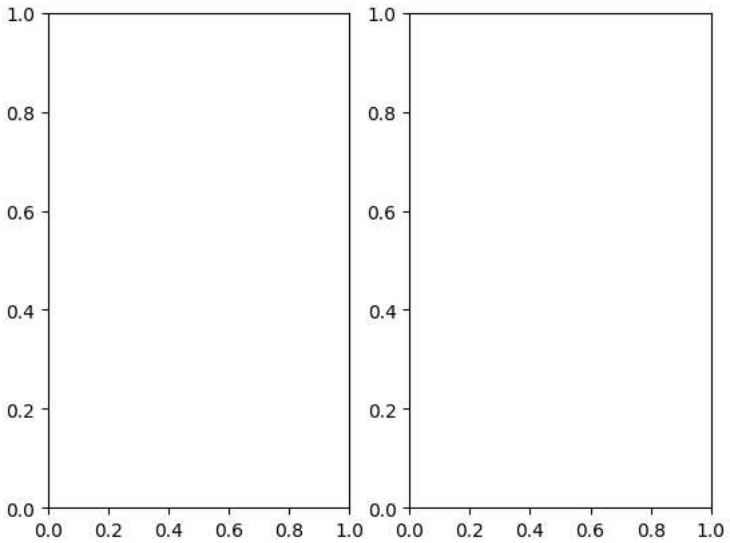

```
fig = plt.figure()
axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8])
axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3])
axes1.plot(x,y,'b')
axes1.set_xlabel('X_Label_axes2')
axes2.set_ylabel('Y_Label_axes2')
axes1.set_title('Axes 2 Title')
axes2.plot(y,x,'r')
axes2.set_xlabel('X_Label_axes2')
axes2.set_ylabel('Y_Label_axes2')
axes2.set_title('Axes 2 Title')
```

```
Text(0.5, 1.0, 'Axes 2 Title')
```


```
fig,axes = plt.subplots()
axes.plot(x,y,'r')
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.set_title('title');
```




```
fig, axes = plt.subplots(nrows = 1, ncols = 2)
```

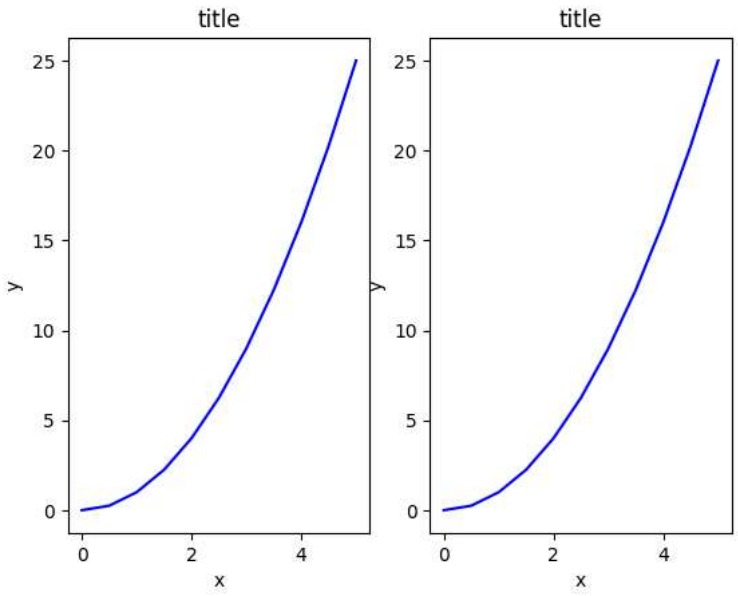



```
axes
```

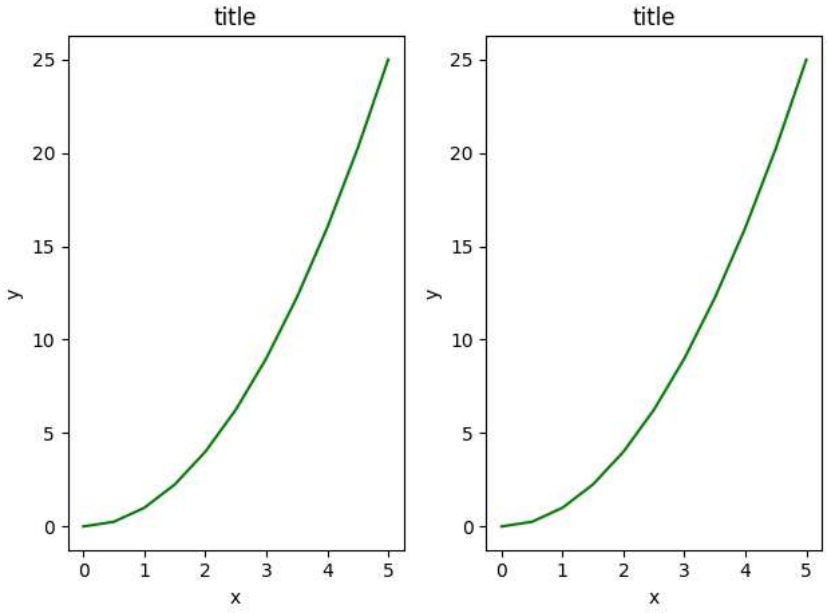
```
array([<Axes: >, <Axes: >], dtype=object)
```

```
for ax in axes:
  ax.plot(x,y,'b')
  ax.set_xlabel('x')
  ax.set_ylabel('y')
```

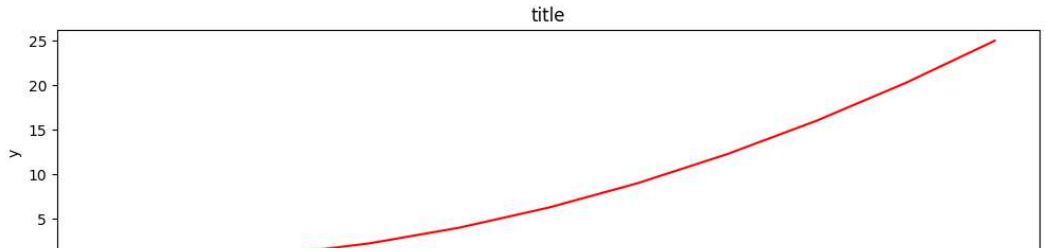
```
  ax.set_title('title')
fig
```




```
fig,axes = plt.subplots(nrows = 1, ncols = 2)
for ax in axes:
  ax.plot(x,y,'g')
  ax.set_xlabel('x')
  ax.set_ylabel('y')
  ax.set_title('title')
fig
plt.tight_layout()
```




```
fig,axes = plt.subplots(figsize = (12,3))
axes.plot(x,y,'r')
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.set_title('title');
```




```
fig.savefig("filename.png")
```

```
fig.savefig("filename.png", dpi = 200)
```

```
ax.set_title("title")
```

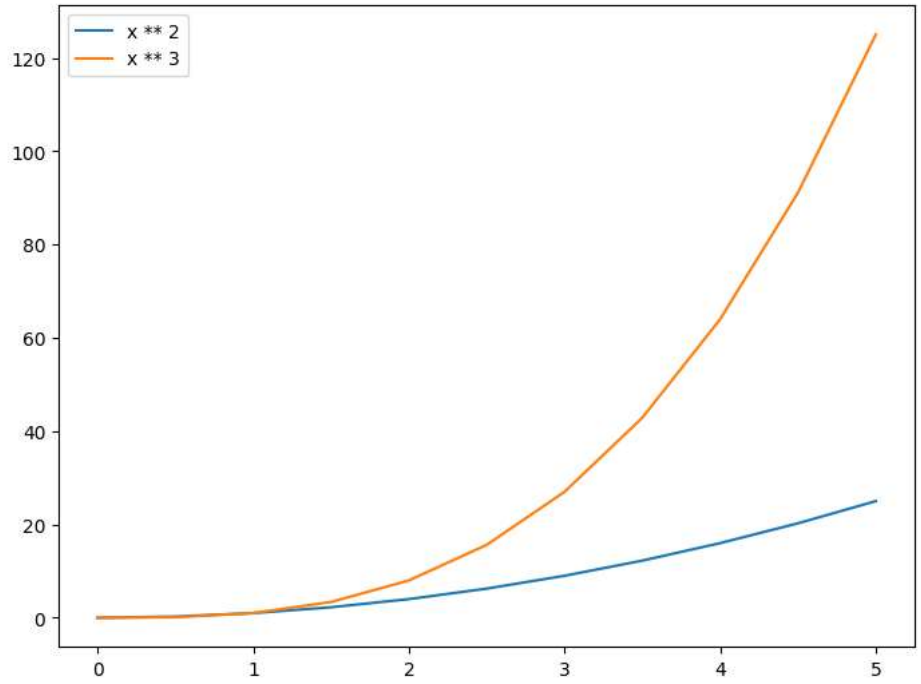
```
     Text(0.5, 1.0, 'title')
```

```
ax.set_xlabel("x")
ax.set_ylabel("y")
```

```
     Text(24.000000000000007, 0.5, 'y')
```

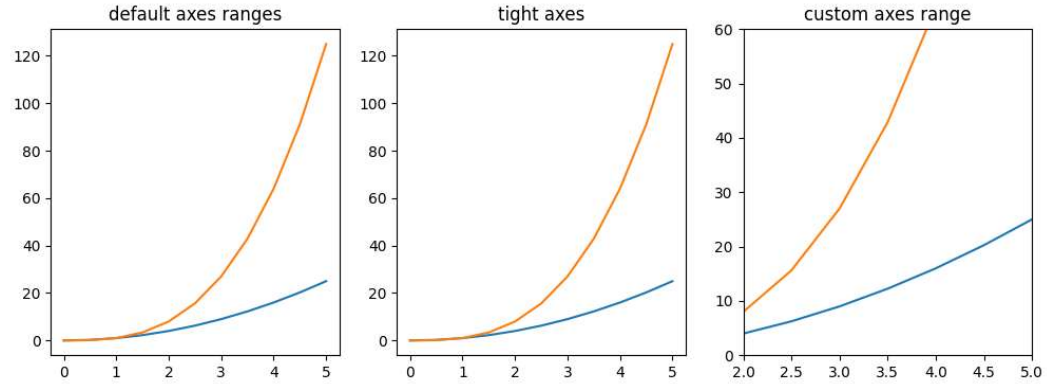
```
fig = plt.figure()
ax = fig.add_axes([0, 0, 1, 1])
ax.plot(x, x ** 2,label = "x ** 2")
ax.plot(x, x ** 3,label = "x ** 3")
ax.legend()
```

```
     <matplotlib.legend.Legend at 0x7f765eb9f400>
```




```
fig,axes = plt.subplots(1, 3, figsize = (12, 4))
axes[0].plot(x, x ** 2, x, x ** 3)
axes[0].set_title("default axes ranges")

axes[1].plot(x, x ** 2,x , x ** 3)
axes[1].axis('tight')
axes[1].set_title("tight axes")

axes[2].plot(x, x ** 2,x , x ** 3)
axes[2].set_ylim([0, 60])
axes[2].set_xlim([2, 5])
axes[2].set_title("custom axes range")
```

```
Text(0.5, 1.0, 'custom axes range')
```



II. Seaborn

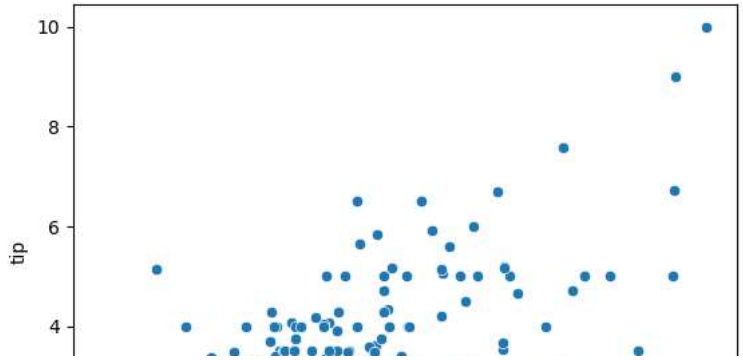
```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sns
%matplotlib inline
sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

```
tips = sns.load_dataset("tips")
tips.head()
```

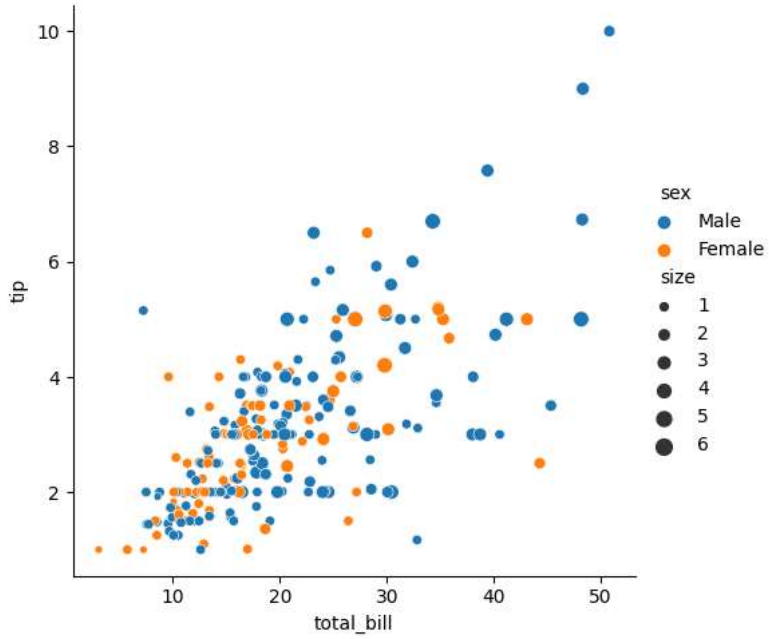
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
ax = sns.scatterplot(x = "total_bill", y = "tip", data = tips)
```



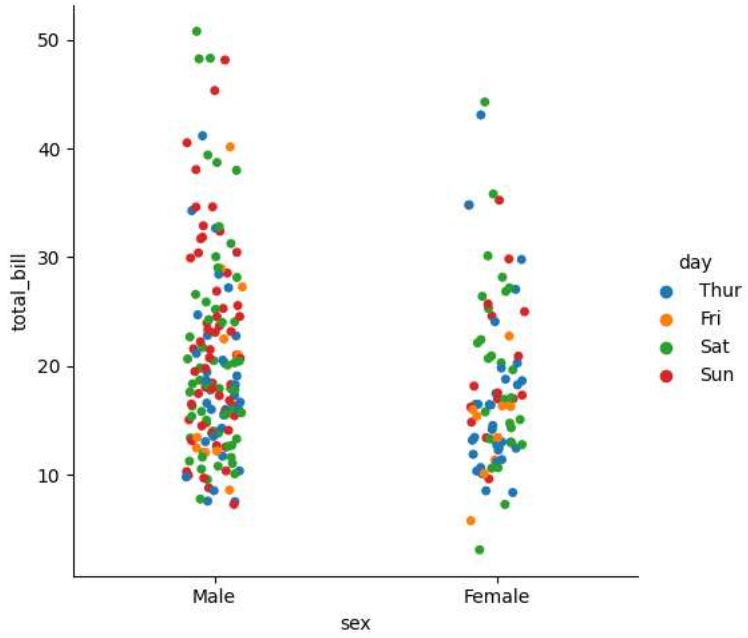

```
sns.relplot(x= "total_bill", y = "tip", data = tips, kind = "scatter",  hue = "sex", size = "size",)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f7657742ee0>
```



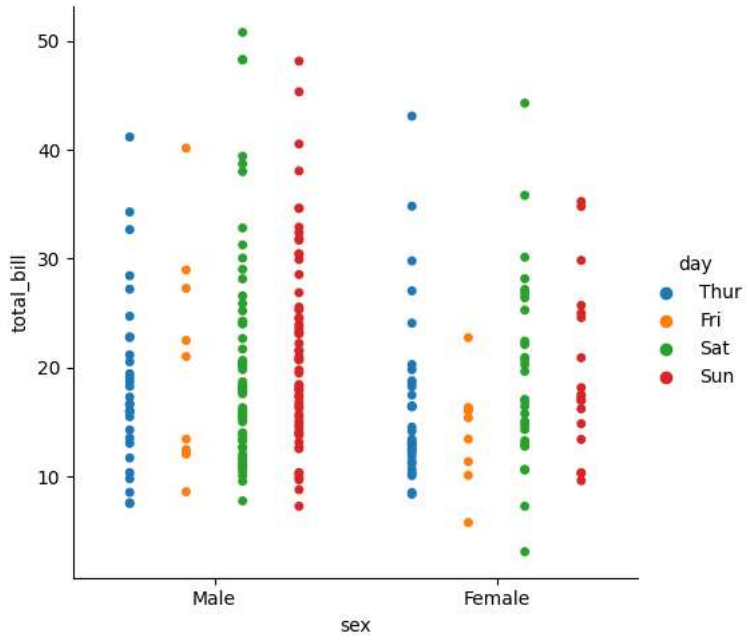

```
sns.catplot(x ="sex", y = "total_bill", hue = "day", data = tips, kind = "strip")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f765755d2e0>
```



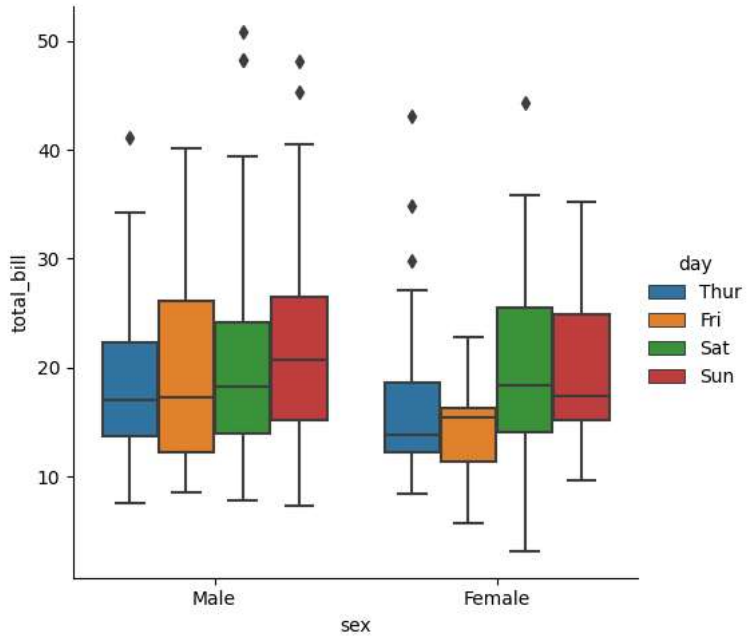
```
sns.catplot(x = "sex", y="total_bill", hue = "day", data = tips, kind = "strip"
            ,jitter = False, dodge = True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f76574e54c0>
```



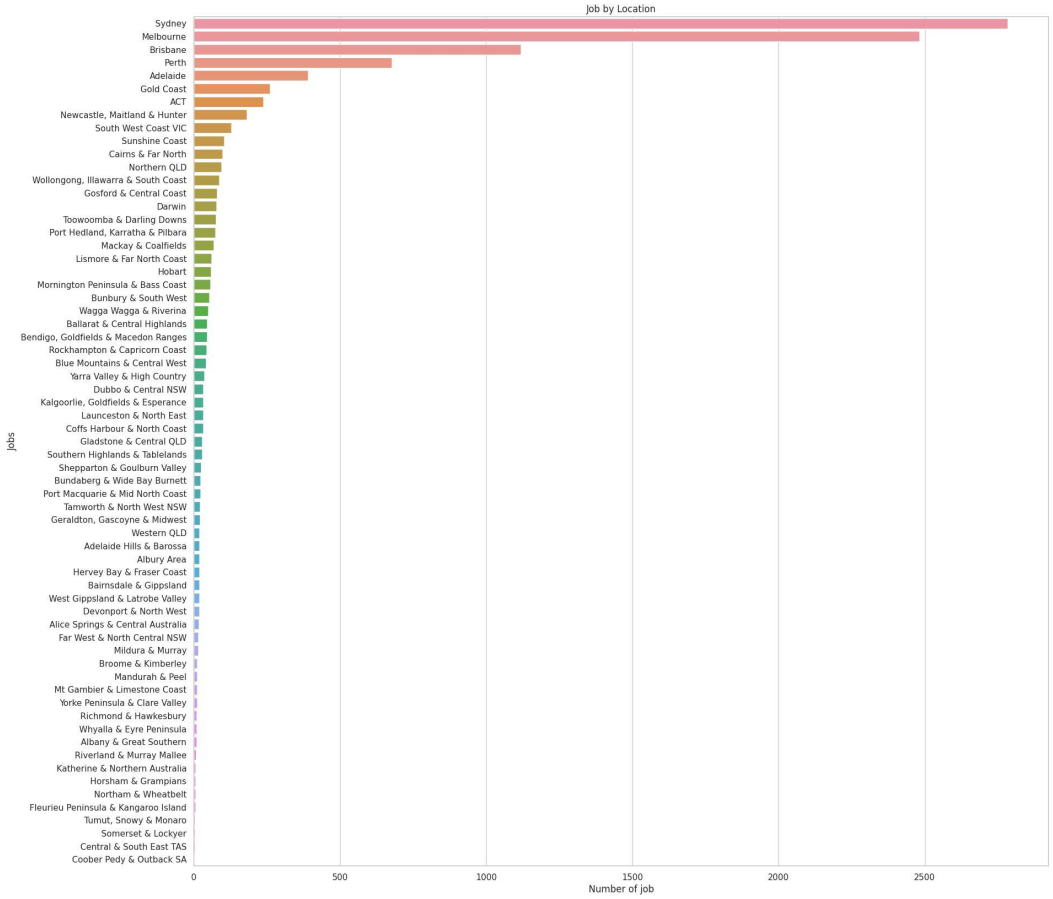

```
sns.catplot(x = "sex", y = "total_bill", hue = "day", data = tips , kind = "box")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f7657398e20>
```




```
# III. Exercises
## 1. Job market


# Load the dataset
df = pd.read_csv('/content/job-market.csv')

# Group the data by location and count the jobs, then sort by the number of jobs
jobs_by_location = df.groupby('Location')['Title'].count().sort_values(ascending = False)


sns.set(style="whitegrid")
plt.figure(figsize=(20, 20))
sns.barplot(y=jobs_by_location.index, x=jobs_by_location.values)
plt.title('Job by Location')
plt.xlabel('Number of job')
```

```
plt.ylabel('Jobs')
plt.show()
```

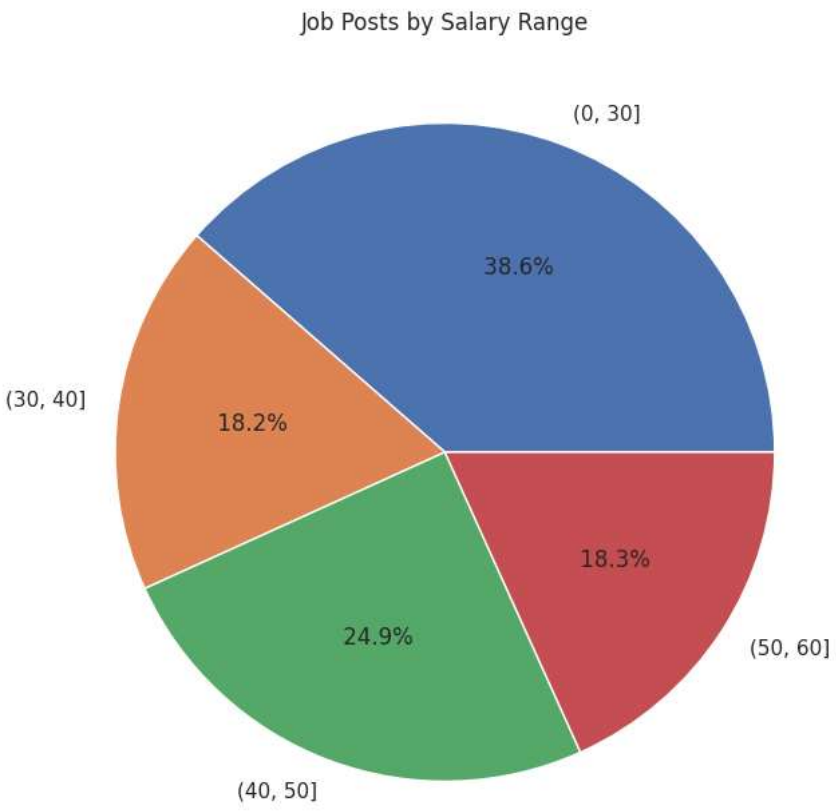



```
df['salary_range'] = pd.cut(df['HighestSalary'], bins=[0,30,40,50,60])

df['mean_salary'] = (df['LowestSalary'] + df['HighestSalary']) / 2

counts = df.groupby('salary_range').size().reset_index(name='count')

counts
```

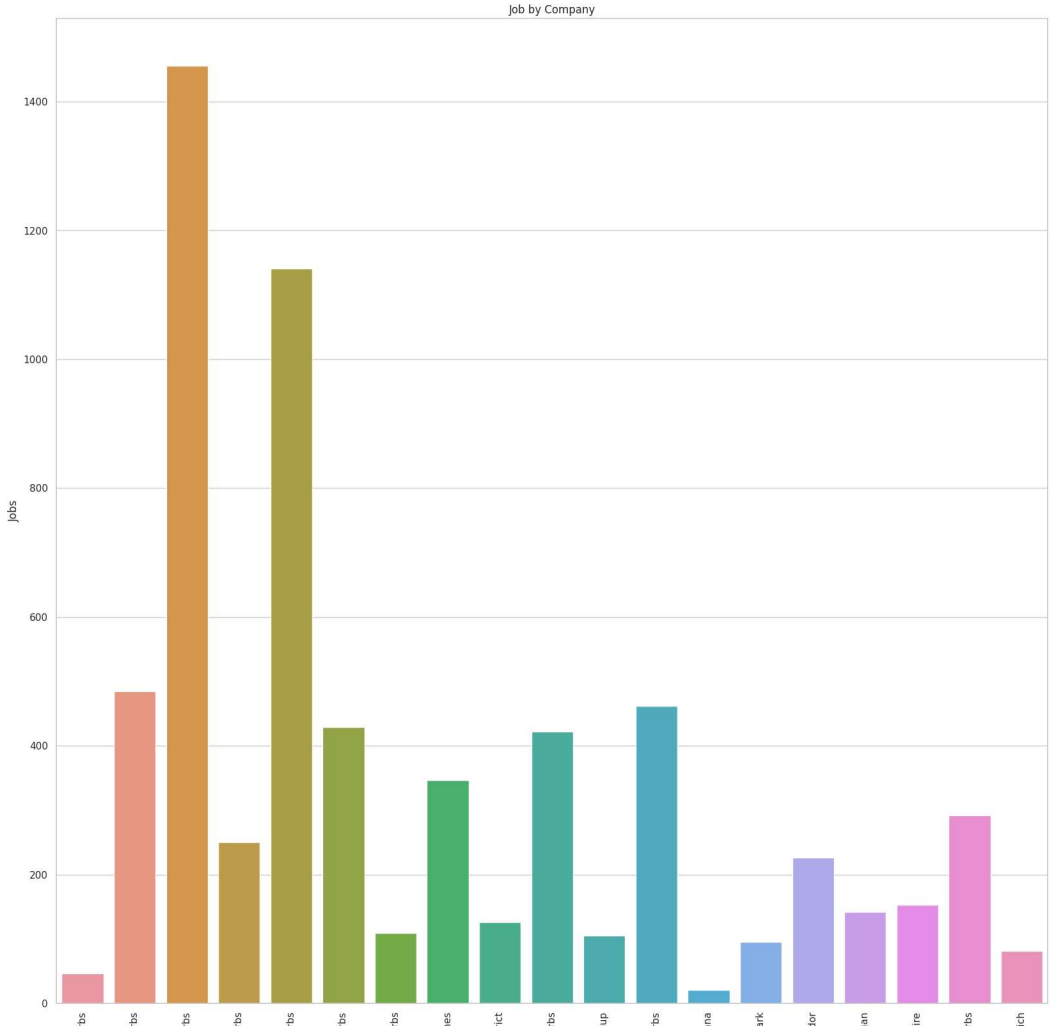
	salary_range	count
0	(0, 30]	3894
1	(30, 40]	1842
2	(40, 50]	2519
3	(50, 60]	1844

```
plt.figure(figsize=(8, 8))
plt.title('Job Posts by Salary Range')
sns.set_palette('pastel')
```

```
plt.pie(counts['count'], labels=counts['salary_range'], autopct='%5.1f%%')
plt.show()
```



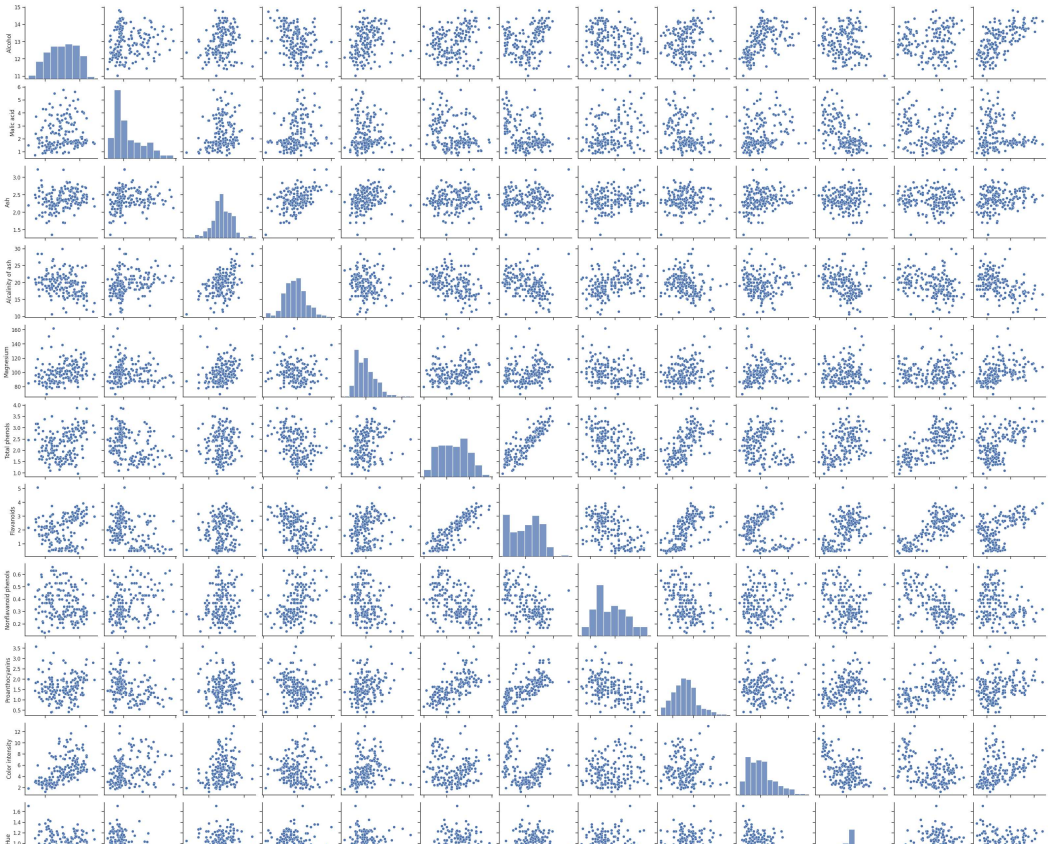

```
jobs_by_company = df.groupby('Area')['Title'].count()

sns.set(style="whitegrid")
plt.figure(figsize=(20, 20))
sns.barplot(y=jobs_by_company.values, x=jobs_by_company.index)
plt.title('Job by Company')
plt.xlabel('Number of job')
plt.xticks(rotation=90)
plt.ylabel('Jobs')
plt.show()
```



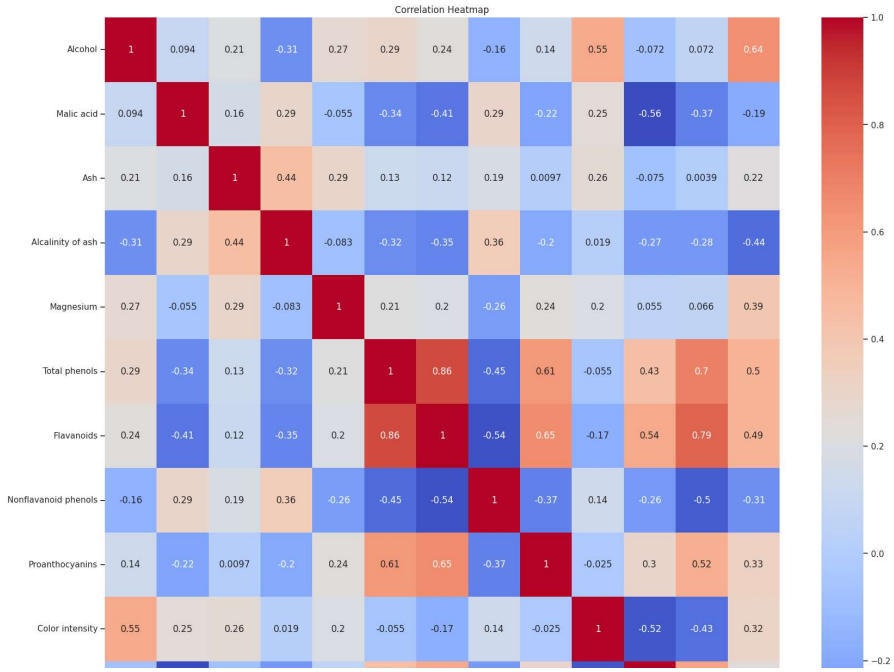
2. Data correlation

```
df = pd.read_csv("/content/wine.data.csv")
```

```
label = df.iloc[:, 0]
data = df.iloc[:, 1:]
sns.set(style='ticks')
sns.pairplot(data)
plt.show()
```




```
corr_matrix = data.corr()

plt.figure(figsize=(20, 20))
sns.heatmap(corr_matrix, cmap='coolwarm', annot=True)
plt.title('Correlation Heatmap')
plt.show()
```




```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
wine_scaled = scaler.fit_transform(df)

kmeans = KMeans(n_clusters=3, random_state=0)
wine_clusters = kmeans.fit_predict(wine_scaled)

df['cluster'] = wine_clusters
sns.pairplot(df, hue='cluster',diag_kind="hist")
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 1
  warnings.warn(
<seaborn.axisgrid.PairGrid at 0x7f40ebd3bf70>
```

✓ 2m 6s completed at 10:17 PM