



# Data Visualization

Dr. Sathien Hunta

School of Information and  
Communication Technology

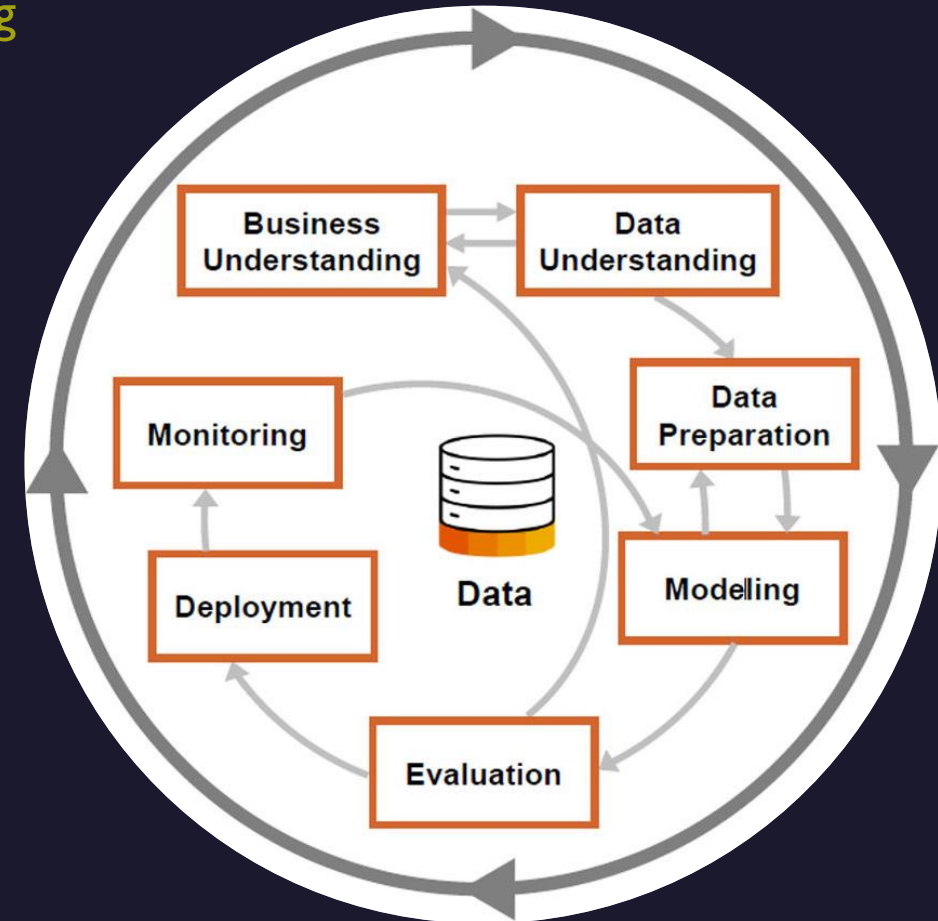
University of Phayao



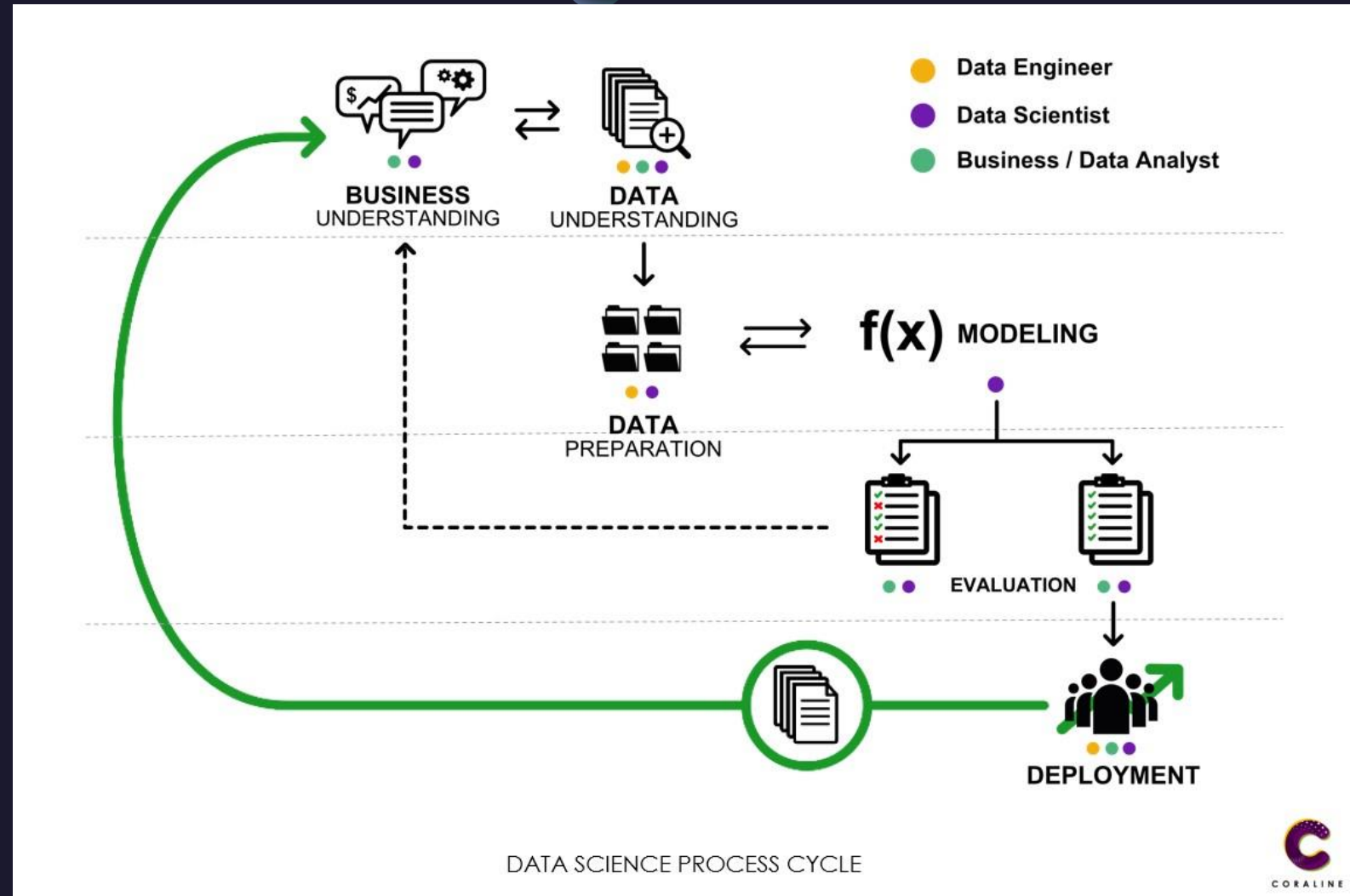
# CRISP-DM

The **C**ross Industry Standard Process for Data Mining (**CRISP-DM**) is a process model that serves as the base for a data science process.

- Business understanding
- Data understanding
- Data preparation
- Modeling
- Evaluation
- Deployment
- Monitoring



# Data Science Process



# Business & Data Understanding



## Business Understanding

ปัญหาคืออะไร

สถานการณ์ปัจจุบันแก้ปัญหายังไร

ข้อกำหนดมีอะไรบ้าง

ผลกระทบทางบวก ทางลบมีอะไรบ้าง

## Data Understanding

1. Gathering Data หรือ การรวบรวมข้อมูล
2. Describing Data หรือ การอธิบายข้อมูล
3. Exploring Data หรือ การวิเคราะห์รายละเอียดของข้อมูล  
(Exploratory Data Analysis (EDA))
4. Verifying Data Quality หรือ การสรุปความพร้อม และคุณภาพของข้อมูล



# ข้อมูล

## ข้อมูลภายนอก

ข้อมูลลูกค้า เจ้าหนี้ อัตราดอกเบี้ยสถาบันการเงิน  
กฎหมายและอัตราภาษีของรัฐบาล ข้อมูลบริษัทคู่แข่ง

## ข้อมูลในองค์กร

ยอดขายประจำปี ข้อมูลผู้ถือหุ้น  
รายงานกำไรขาดทุน ข้อมูล  
พนักงาน

- ข้อมูลส่วนบุคคล
- ข้อมูลพฤติกรรมการทำงาน

Social media



# Data

ID

Attribute, Feature

Label

ID	Outlook	Humidity	Windy	Play
1	Sunny	High	FALSE	No
2	Sunny	High	TRUE	No
3	Overcast	Normal	FALSE	Yes
4	Rainy	High	FALSE	Yes

Value type

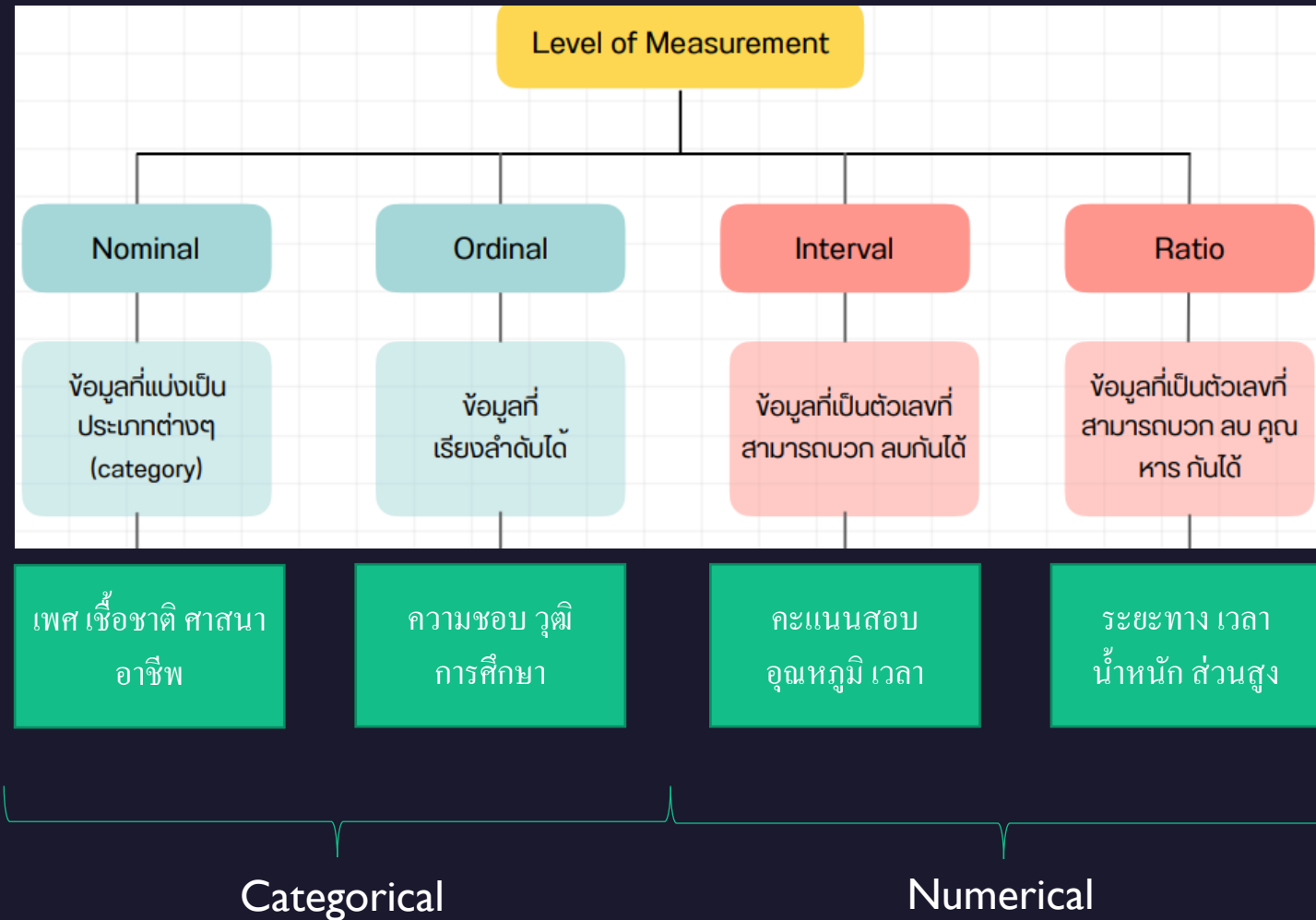
Numeric

Nominal

Binominal




# Levels of measurement in statistics



# Exploratory Data Analysis

- Checking the types of data
- Dropping irrelevant columns
- Renaming the columns
- Dropping the duplicate rows
- Dropping the missing or null values
- Detecting Outliers
- Plotting different features



Please refer to the  
data preparation slide





# Libraries

## NumPy

```
import numpy as np

arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])

print('5th element on 2nd row:', arr[1, 4])
```

## Pandas

```
import pandas

mydataset = {
    'cars': ["BMW", "Volvo", "Ford"],
    'passings': [3, 7, 2]
}

data = pandas.DataFrame(mydataset)

print(data)
```

- **numpy** คือ Library ที่เอาไว้ทำงานกับตัวเลข
- **pandas** คือ Library ที่เอาไว้จัดการกับข้อมูล
- **matplotlib** คือ Library สำหรับการจัดการเรื่องการ **plots**
- **seaborn** คือ Library ที่ช่วยให้การแสดงผลจากสถิติสวยงามและหลากหลายขึ้น

## Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np
```

```
xpoints = np.array([1, 2, 6, 8])
ypoints = np.array([3, 8, 1, 10])
```

```
plt.plot(xpoints, ypoints)
plt.show()
```

# Import libraries and data

```
import numpy as np
```

```
import pandas as pd
```

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

```
import seaborn as sns
```

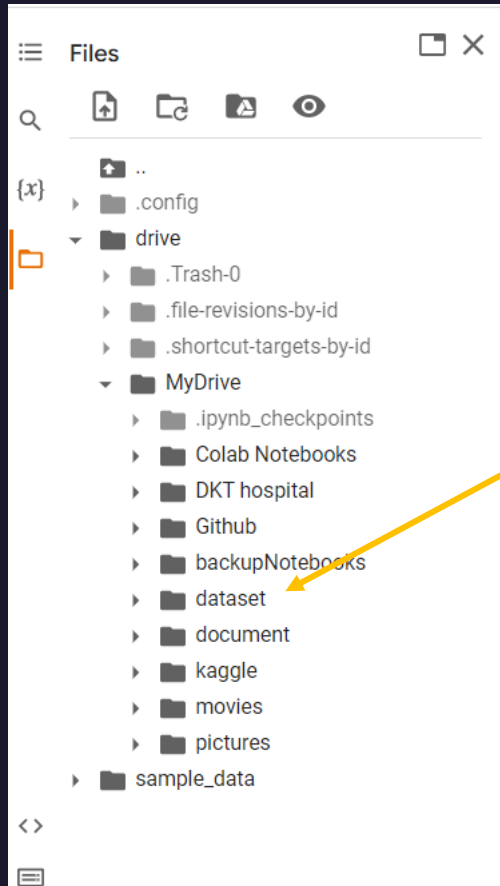
Visualization

Import  
data

```
data = pd.read_csv('PATH') หรือ
```

- `data = pd.read_excel('PATH', encoding = 'utf-8')`
- `data = pd.read_json('data.json')`

# Import data



```
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[6] data_path = 'drive/MyDrive/dataset/'

import pandas as pd
data = pd.read_csv(data_path+'student-utf8.csv')
data.head()
```

	<bound method NDFrame.head of	Student_ID	Study	Rank_study_group	Age_group	GPA_old_group	\
0	6 NaN	มัธยมปลาย	27	2.89			
1	7 ไม่ระบุ	สามัญ	28	3.60			
2	9 ไม่ระบุ	สามัญ	29	3.20			
3	10 ไม่ระบุ	อาชีวะ	20	3.20			
4	11 ไม่ระบุ	อาชีวะ	18	3.10			
...	...	...	...	...			
995	987 ต่างจังหวัด	สามัญ	26	2.70			
996	993 ต่างจังหวัด	อาชีวะ	20	2.50			
997	994 ต่างจังหวัด	สามัญ	29	3.60			
998	996 ต่างจังหวัด	อาชีวะ	18	3.60			
999	1000 ต่างจังหวัด	อาชีวะ	24	3.80			

	Rank_grade_major	Rank_grade_business	Rank_grade_computer	\
0	2.5	3.0	3.5	
1	4.0	1.0	1.0	
2	2.0	2.0	2.0	
3	4.0	1.5	1.5	

# Checking Data Types

Please refer to the  
data preparation slide

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 11 columns):  
#   Column                Non-Null Count  Dtype    
---  ---                  
0   Student_ID            1000 non-null   int64     
1   Study                 988 non-null    object    
2   Rank_study_group      994 non-null    object    
3   Age_group             1000 non-null   int64     
4   GPA_old_group         1000 non-null   float64   
5   Rank_grade_major      1000 non-null   float64   
6   Rank_grade_business    1000 non-null   float64   
7   Rank_grade_computer    1000 non-null   float64   
8   Rank_grade_finance     1000 non-null   object    
9   Rank_grade_total      1000 non-null   object    
10  Major                 1000 non-null   object    
dtypes: float64(4), int64(2), object(5)  
memory usage: 86.1+ KB
```

Info on data types:

```
data.isnull().sum()
```

```
Student_ID      0  
Study           12  
Rank_study_group 6  
Age_group       0  
GPA_old_group   0  
Rank_grade_major 0  
Rank_grade_business 0  
Rank_grade_computer 0  
Rank_grade_finance 0  
Rank_grade_total 0  
Major           0  
dtype: int64
```

Summary of missing values:

```
data.isnull().sum()
```

```
Student_ID      0  
Study           12  
Rank_study_group 6  
Age_group       0  
GPA_old_group   0  
Rank_grade_major 0  
Rank_grade_business 0  
Rank_grade_computer 0  
Rank_grade_finance 0  
Rank_grade_total 0  
Major           0  
dtype: int64
```

```
[15] data.shape  
(1000, 11)
```

```
[16] data.dropna(how='any').shape  
(982, 11)
```

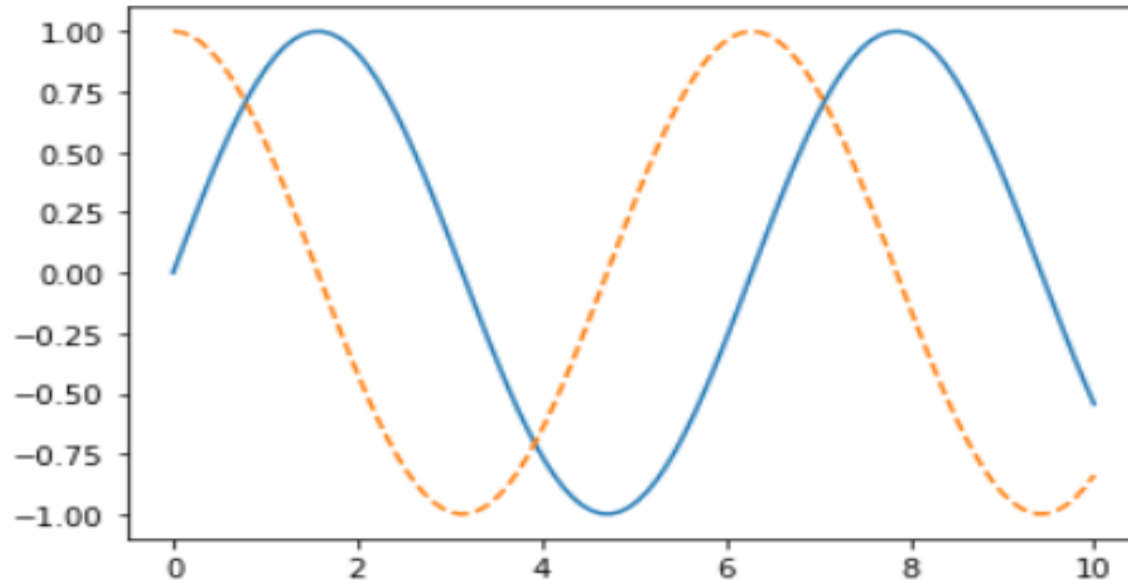
```
data.dropna(how='all').shape  
(1000, 11)
```

Drop missing values:

# Plotting from notebook

```
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
x = np.linspace(0,10,100)
fig = plt.figure()
plt.plot(x,np.sin(x), '-')
plt.plot(x,np.cos(x), '--')
```

[<matplotlib.lines.Line2D at 0x7fb386340b50>]



# Saving Figures to File

✓  
0s



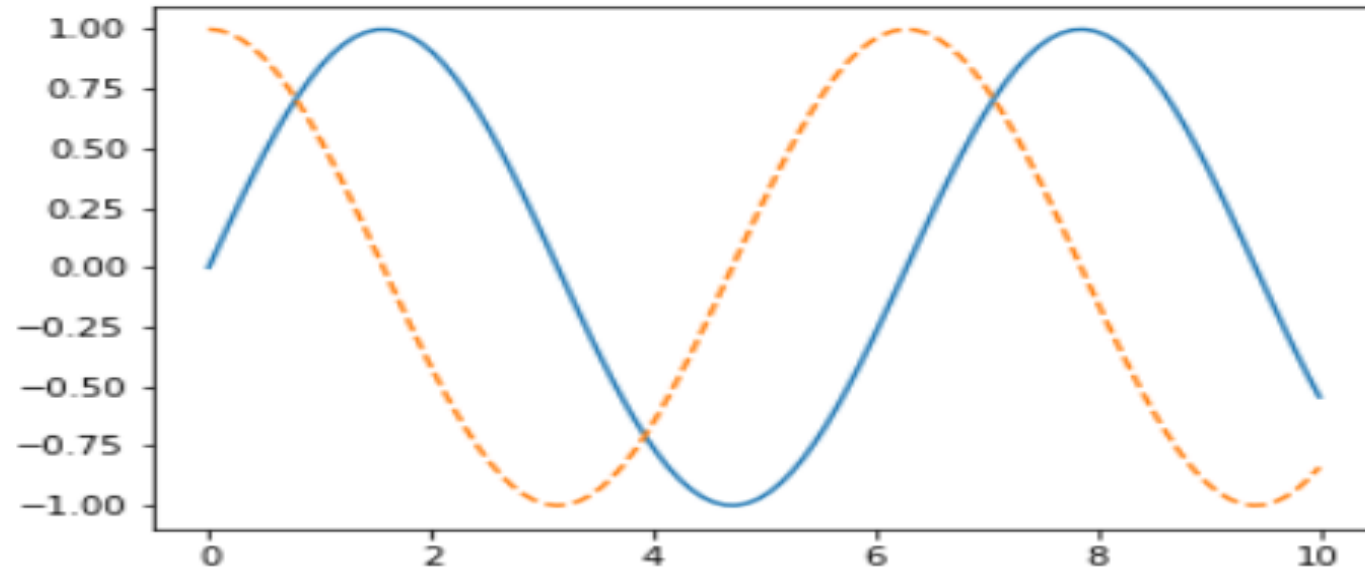
```
fig.savefig('my_figure.png')  
!ls
```



my\_figure.png sample\_data

✓  
0s

```
[13] from IPython.display import Image  
Image('my_figure.png')
```



# MATLAB-style Interface

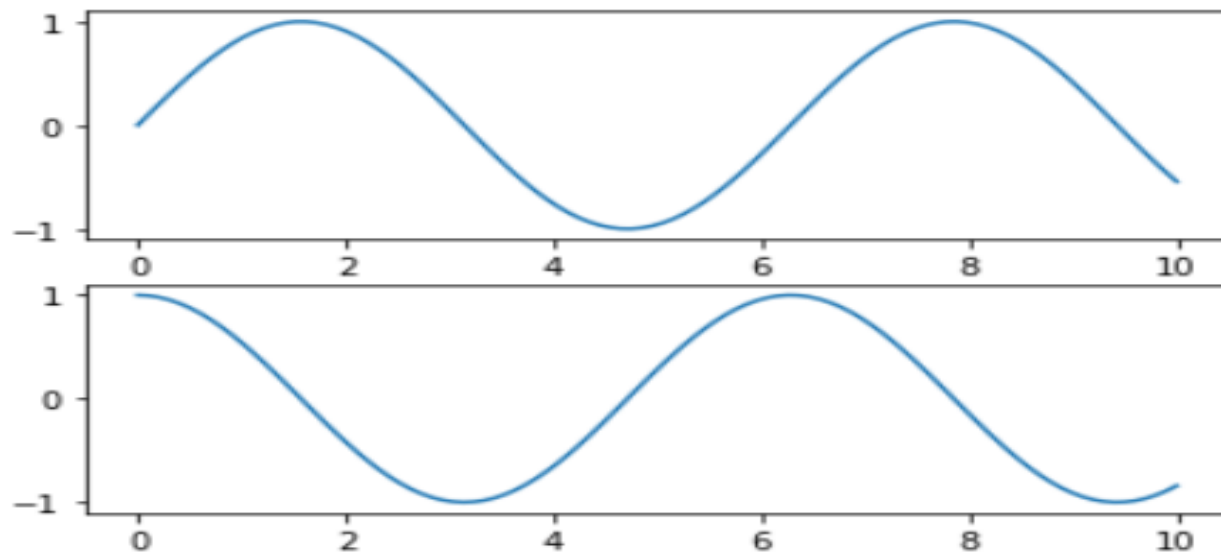
✓  
0s



```
plt.figure() # create a plot figure
```

```
# create the first of two panels and set current axis  
plt.subplot(2, 1, 1) # (rows, columns, panel number)  
plt.plot(x, np.sin(x))
```

```
# create the second panel and set current axis  
plt.subplot(2, 1, 2)  
plt.plot(x, np.cos(x));
```



# Line Plots

✓  
0s



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

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
y1 = [1, 3, 5, 3, 1, 3, 5, 3, 1]
```

```
y2 = [2, 4, 6, 4, 2, 4, 6, 4, 2]
```

```
plt.plot(x, y1, label="line L")
```

```
plt.plot(x, y2, label="line H")
```

```
plt.plot()
```

```
plt.xlabel("x axis")
```

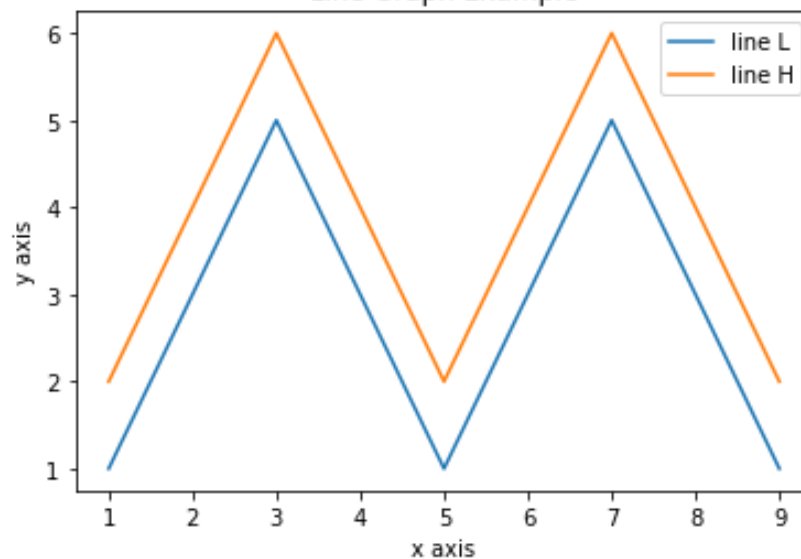
```
plt.ylabel("y axis")
```

```
plt.title("Line Graph Example")
```

```
plt.legend()
```

<matplotlib.legend.Legend at 0x7f056e33aa10>

Line Graph Example





# Scatter Plots

✓  
0s



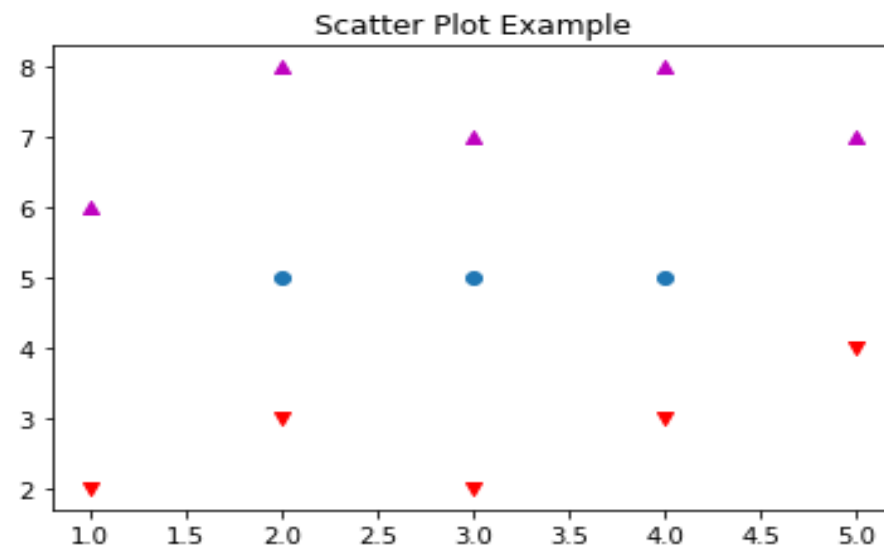
```
import matplotlib.pyplot as plt

x1 = [2, 3, 4]
y1 = [5, 5, 5]

x2 = [1, 2, 3, 4, 5]
y2 = [2, 3, 2, 3, 4]
y3 = [6, 8, 7, 8, 7]

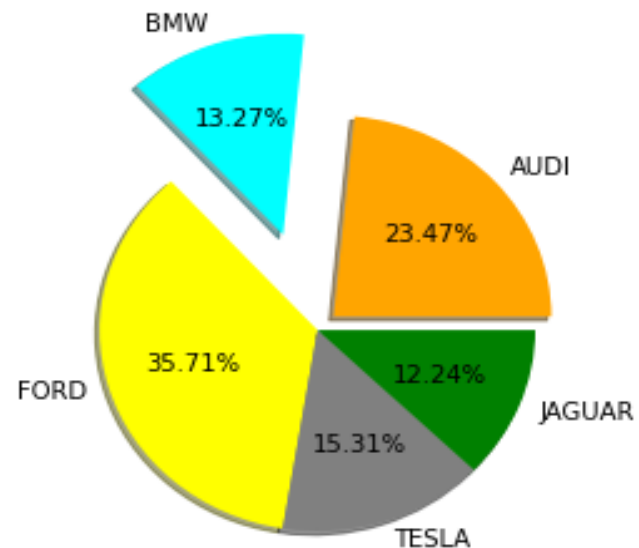
# Markers: https://matplotlib.org/api/markers\_api.html

plt.scatter(x1, y1)
plt.scatter(x2, y2, marker='v', color='r')
plt.scatter(x2, y3, marker='^', color='m')
plt.title('Scatter Plot Example')
plt.show()
```



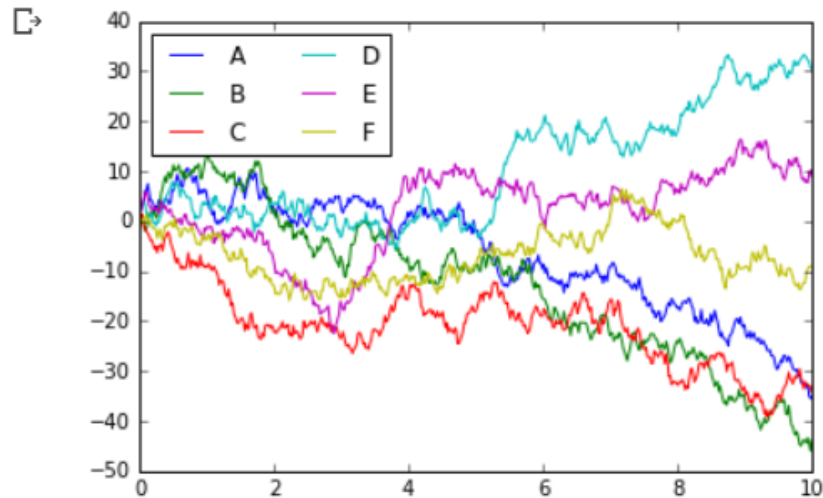
# Pie charts

```
✓ [1] import matplotlib.pyplot as plt  
0s import pandas as pd  
cars = ['AUDI', 'BMW', 'FORD', 'TESLA', 'JAGUAR',]  
data = [23, 13, 35, 15, 12]  
explode = [0.1, 0.5, 0, 0, 0]  
colors = ( "orange", "cyan", "yellow", "grey", "green",)  
# plotting the data  
plt.pie(data, labels=cars, explode=explode, autopct='%1.2f%%', colors=colors, shadow=True)  
plt.show()
```



# Matplotlib vs Seaborn

```
import matplotlib.pyplot as plt
plt.style.use('classic')
%matplotlib inline
import numpy as np
import pandas as pd
rng = np.random.RandomState(0)
x = np.linspace(0, 10, 500)
y = np.cumsum(rng.randn(500, 6), 0)
plt.plot(x, y)
plt.legend('ABCDEF', ncol=2, loc='upper left');
```



```
import seaborn as sns
sns.set()
plt.plot(x,y)
plt.legend('ABCDEF', ncol = 2, loc='upper left')
```

<matplotlib.legend.Legend at 0x7fb38564db50>

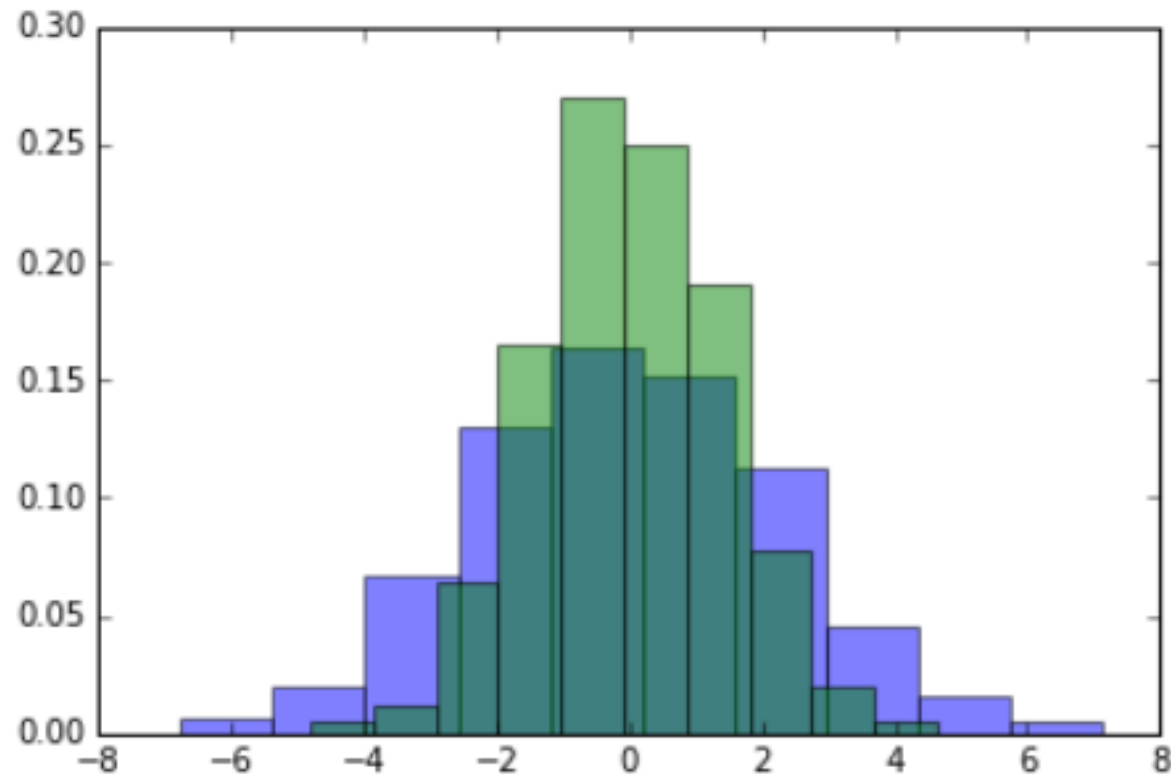


# Histograms

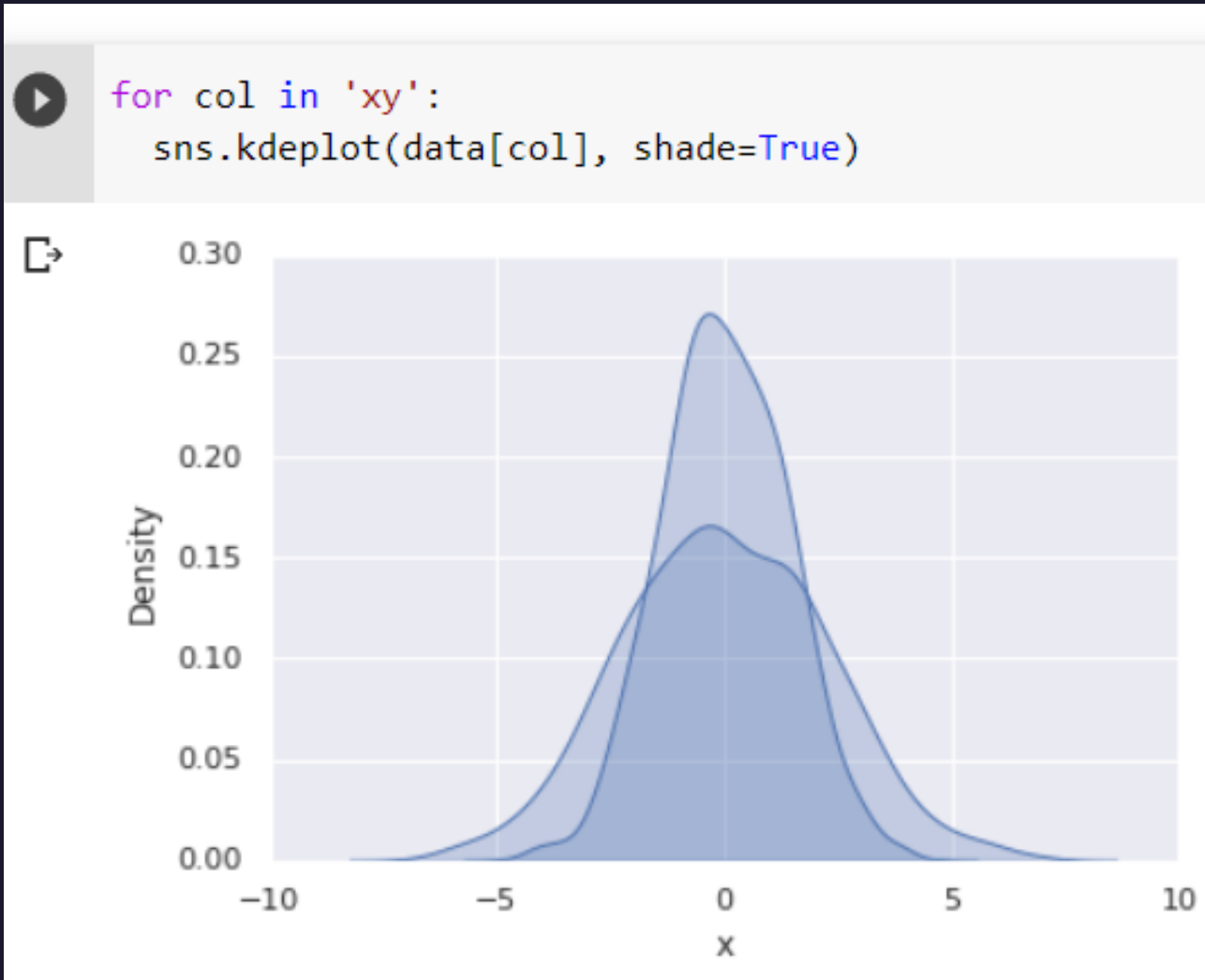


```
data = np.random.multivariate_normal([0, 0], [[5, 2], [2, 2]])  
data = pd.DataFrame(data, columns=['x', 'y'])
```

```
for col in 'xy':  
    plt.hist(data[col], density=True, alpha=0.5)
```



# Kernel Distribution Estimation (KDE)



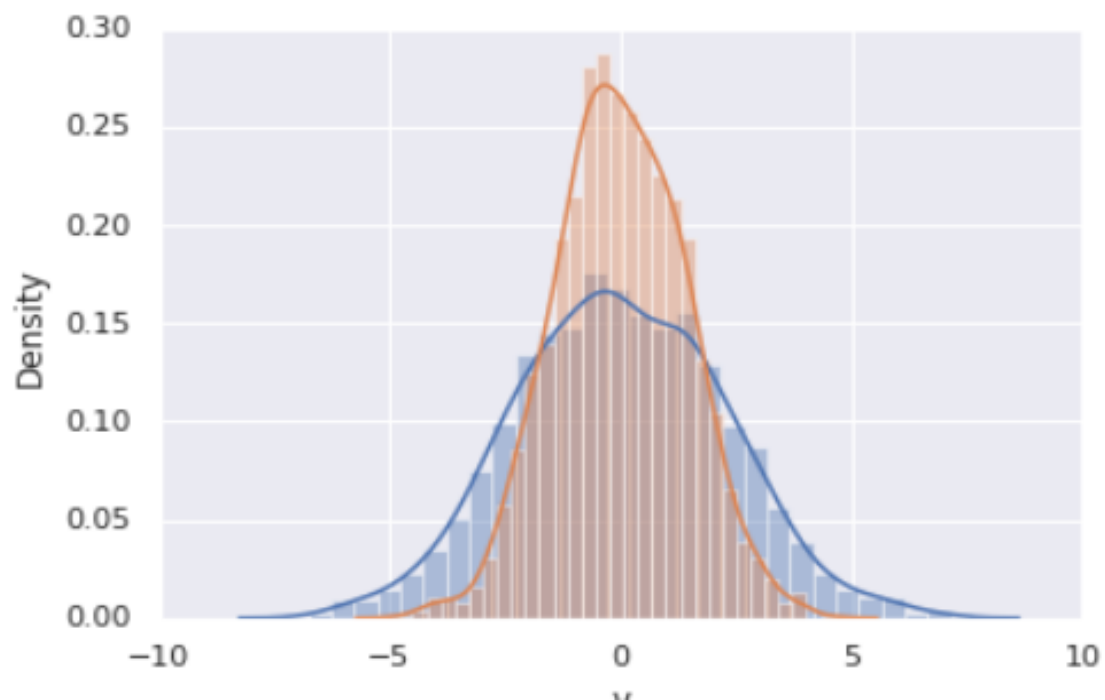
# Densities



```
sns.distplot(data['x'])  
sns.distplot(data['y'])
```



```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py  
warnings.warn(msg, FutureWarning)  
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py  
warnings.warn(msg, FutureWarning)  
<matplotlib.axes._subplots.AxesSubplot at 0x7f10115056d0>
```



# Pair plot

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

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
sns.pairplot(iris, hue='species', height=2.5)
```

<seaborn.axisgrid.PairGrid at 0x7f100dc26b90>



*Iris setosa*



*Iris versicolor*

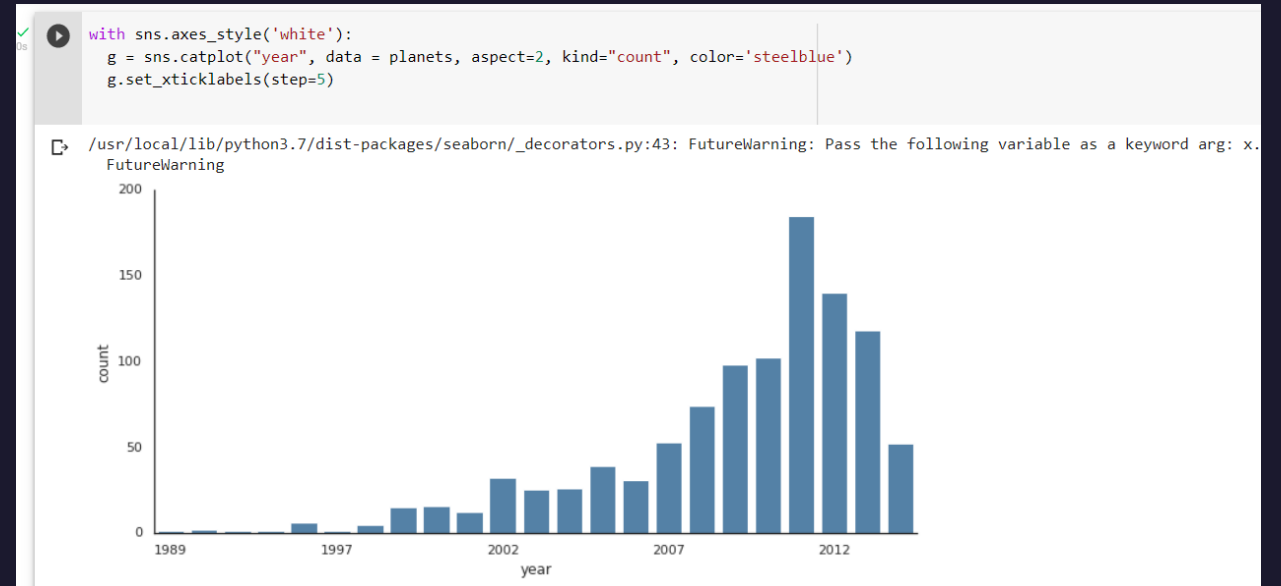
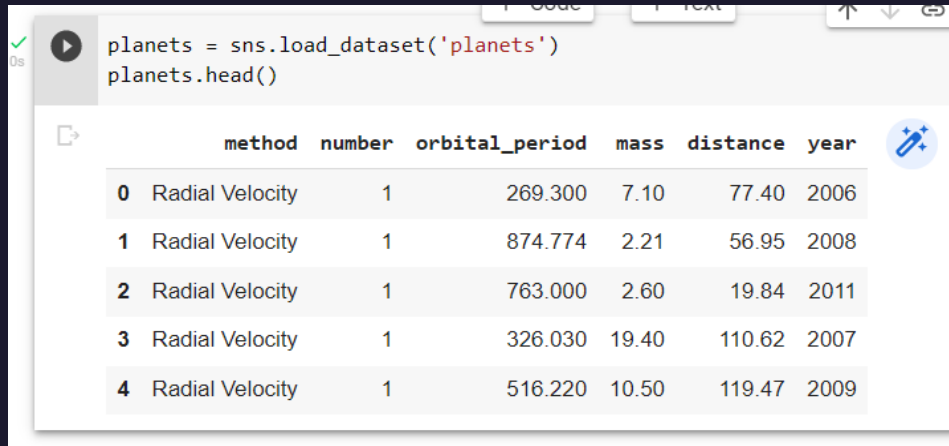


*Iris virginica*

petal

sepal

# Bar plots





# Faceted Histograms

✓  
0s

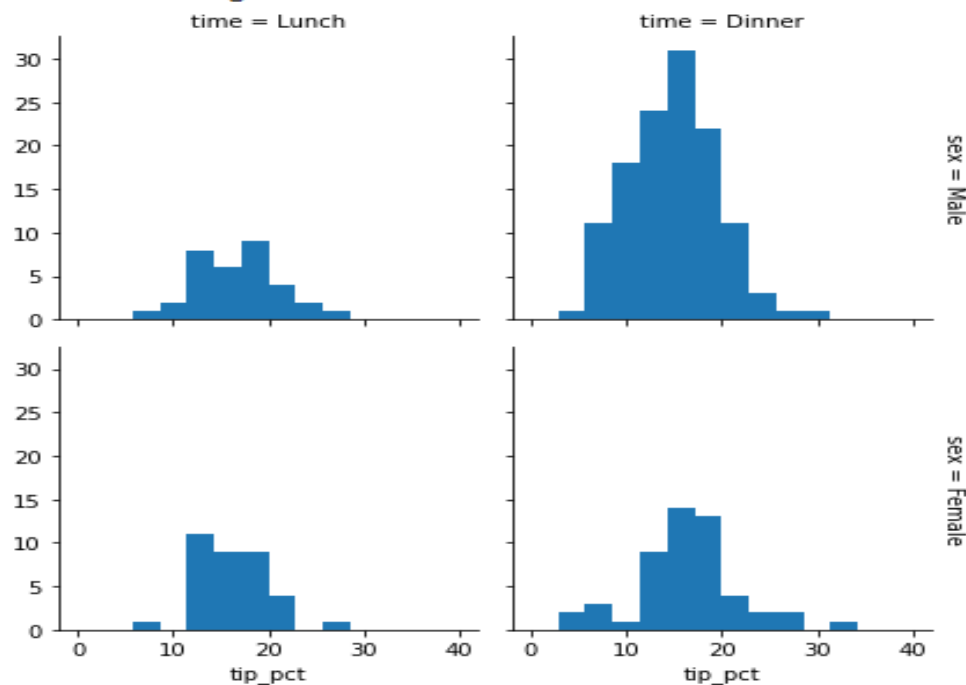
```
[4] import numpy as np  
tips = sns.load_dataset('tips')  
tips.head()
```

	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

✓  
2s

```
tips['tip_pct'] = 100*tips['tip']/tips['total_bill']  
grid = sns.FacetGrid(tips, row="sex", col="time", margin_titles=True)  
grid.map(plt.hist, "tip_pct", bins=np.linspace(0,40,15))
```

<seaborn.axisgrid.FacetGrid at 0x7f8752b5af90>



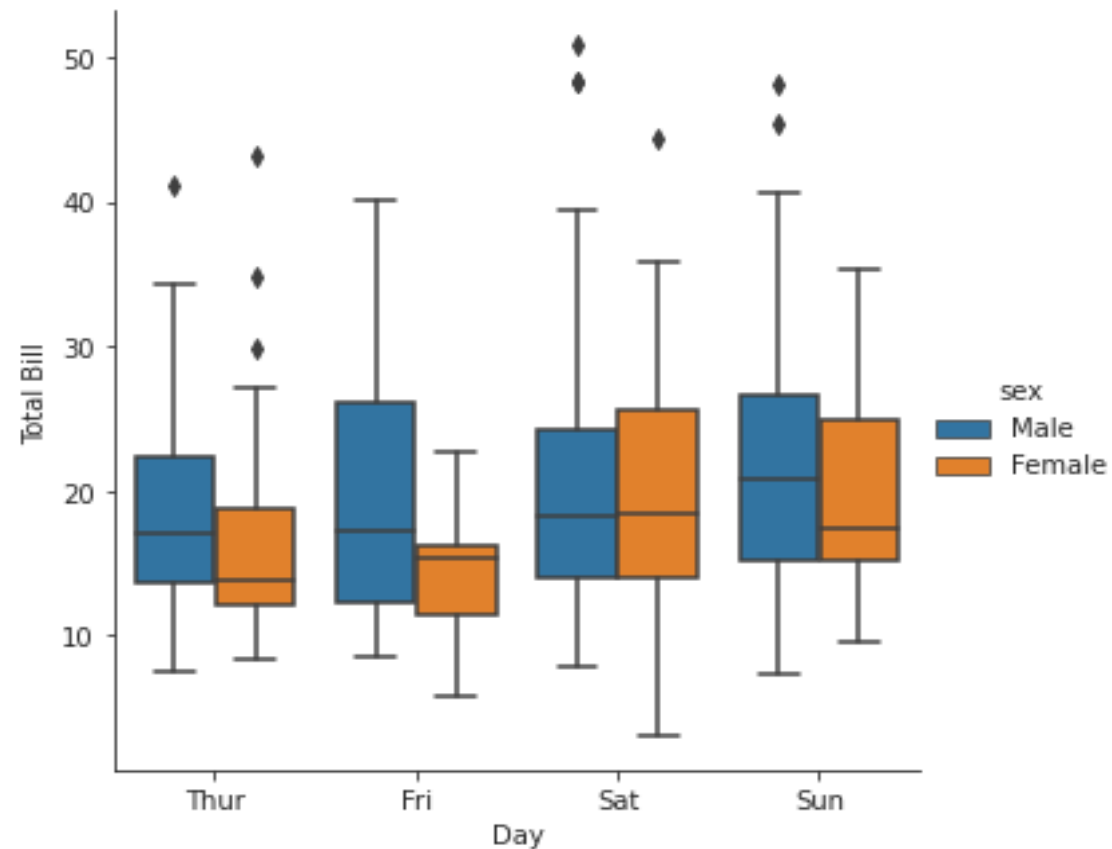
# Factor plots

✓  
0s



```
with sns.axes_style(style='ticks'):  
    g = sns.factorplot("day", "total_bill", "sex", data=tips, kind="box")  
    g.set_axis_labels("Day", "Total Bill")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: U  
warnings.warn(msg)  
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Fut  
FutureWarning
```





# Summary

**Data Visualization** คือ การนำข้อมูลที่ได้มาจากแหล่งข้อมูลต่าง ๆ มาวิเคราะห์ ประมวลผลแล้วนำเสนอออกมาในรูปแบบที่มองเห็นและทำความเข้าใจได้

จุดประสงค์สำคัญ คือ การนำเสนอข้อมูลให้เข้าใจได้ง่าย ซึ่งจุดสำคัญของเนื้อหา และชี้ข้อ เปรียบเทียบให้เห็นอย่างชัดเจน รวมถึงช่วยให้สังเกตเห็นจุดที่น่าสนใจของข้อมูลได้ง่ายขึ้น

# Thank You

Sathien Hunta

[sathien.hu@up.ac.th](mailto:sathien.hu@up.ac.th)

<http://ict.up.ac.th>

