Exploratory Data Analysis

Hongchang Gao

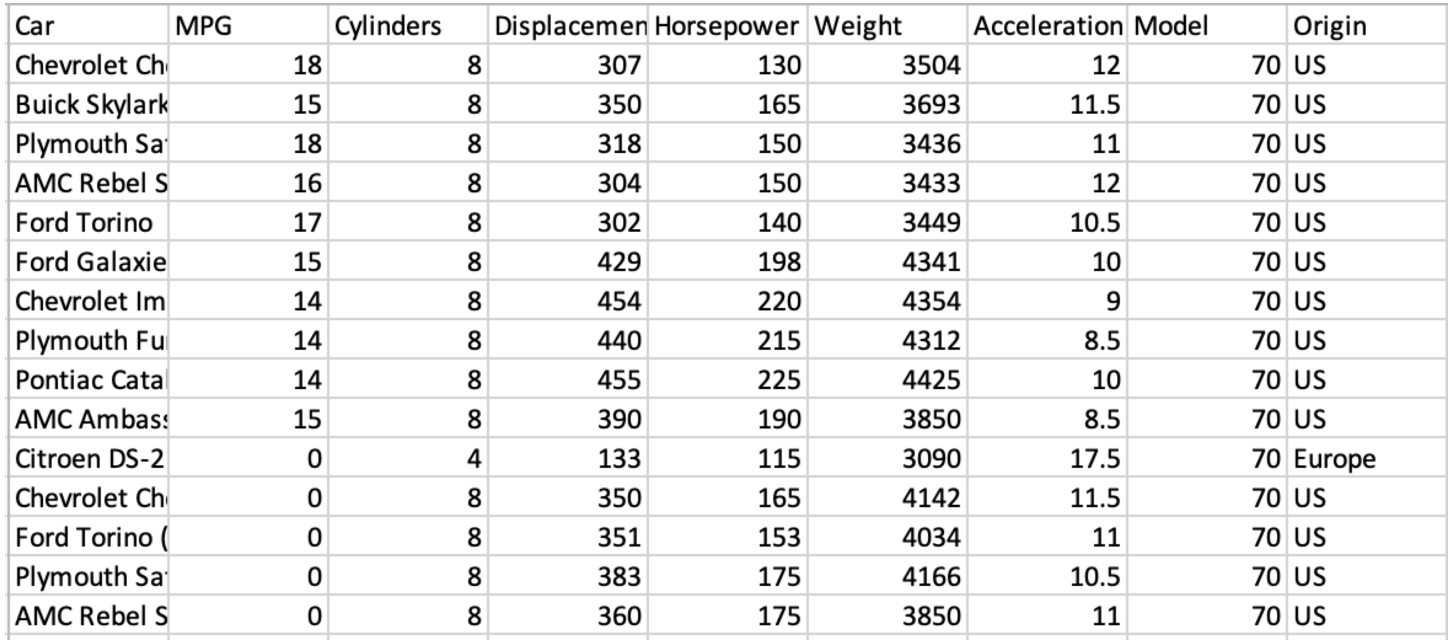
Spring 2024

Exploratory Data Analysis

- Exploratory Data Analysis (EDA): different data have different properties
  - Once a dataset is loaded, the first step is to learn something about it.
  - The dataset is small, directly inspect it
  - The dataset is large, visualize it or explore its statistics

Tabular Data

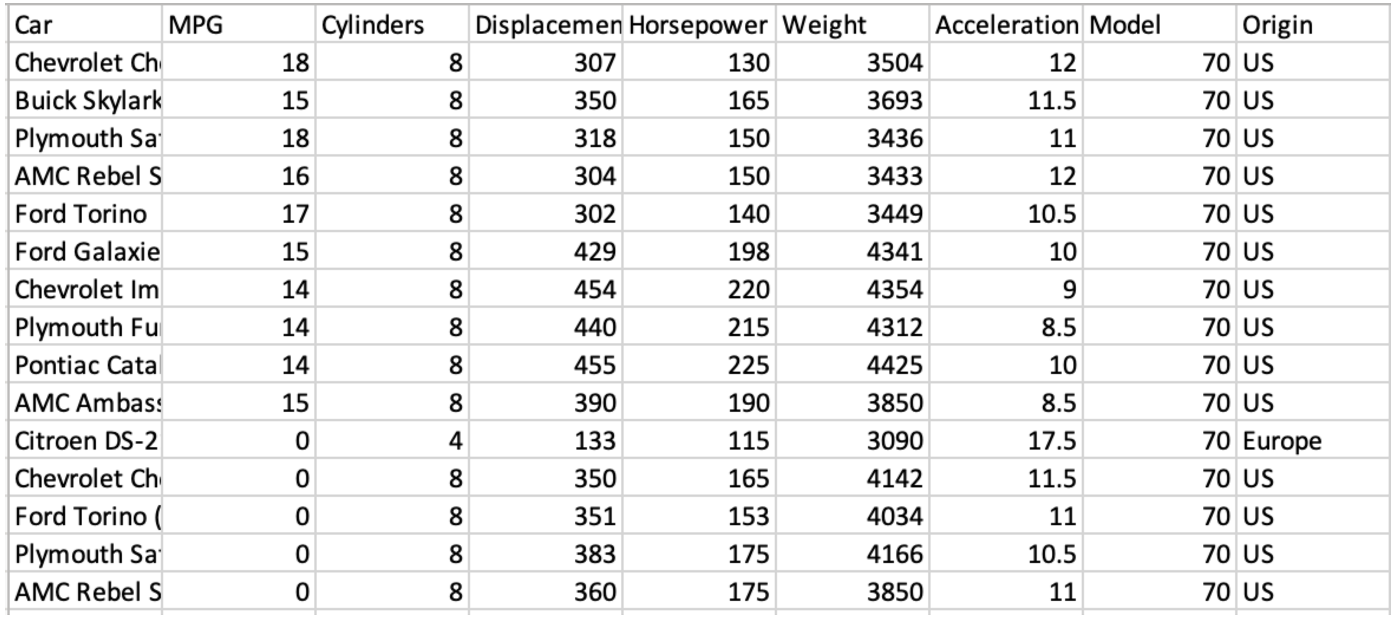
- Tabular Data
  - dataset is represented as a 2d array
  - rows, also referred to as examples, data points
  - columns, also referred to as attributes, features, variables



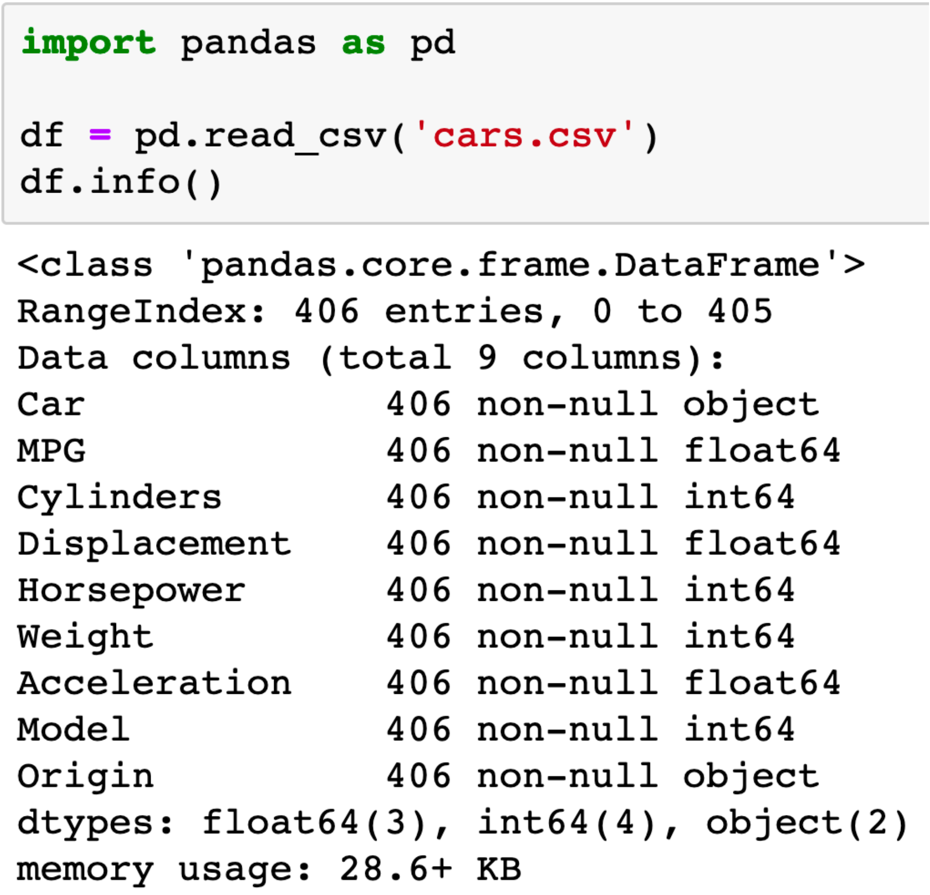
Car	MPG	Cylinders	Displacemen	Horsepower	Weight	Acceleration	Model	Origin
Chevrolet Ch	18	8	307	130	3504	12	70	US
Buick Skylark	15	8	350	165	3693	11.5	70	US
Plymouth Sa	18	8	318	150	3436	11	70	US
AMC Rebel S	16	8	304	150	3433	12	70	US
Ford Torino	17	8	302	140	3449	10.5	70	US
Ford Galaxie	15	8	429	198	4341	10	70	US
Chevrolet Im	14	8	454	220	4354	9	70	US
Plymouth Fu	14	8	440	215	4312	8.5	70	US
Pontiac Catal	14	8	455	225	4425	10	70	US
AMC Ambass	15	8	390	190	3850	8.5	70	US
Citroen DS-2	0	4	133	115	3090	17.5	70	Europe
Chevrolet Ch	0	8	350	165	4142	11.5	70	US
Ford Torino (	0	8	351	153	4034	11	70	US
Plymouth Sa	0	8	383	175	4166	10.5	70	US
AMC Rebel S	0	8	360	175	3850	11	70	US

Tabular Data

- Numerical features
- Categorical features



Car	MPG	Cylinders	Displacemen	Horsepower	Weight	Acceleration	Model	Origin
Chevrolet Ch	18	8	307	130	3504	12	70	US
Buick Skylark	15	8	350	165	3693	11.5	70	US
Plymouth Sa	18	8	318	150	3436	11	70	US
AMC Rebel S	16	8	304	150	3433	12	70	US
Ford Torino	17	8	302	140	3449	10.5	70	US
Ford Galaxie	15	8	429	198	4341	10	70	US
Chevrolet Im	14	8	454	220	4354	9	70	US
Plymouth Fu	14	8	440	215	4312	8.5	70	US
Pontiac Catal	14	8	455	225	4425	10	70	US
AMC Ambas	15	8	390	190	3850	8.5	70	US
Citroen DS-2	0	4	133	115	3090	17.5	70	Europe
Chevrolet Ch	0	8	350	165	4142	11.5	70	US
Ford Torino (	0	8	351	153	4034	11	70	US
Plymouth Sa	0	8	383	175	4166	10.5	70	US
AMC Rebel S	0	8	360	175	3850	11	70	US

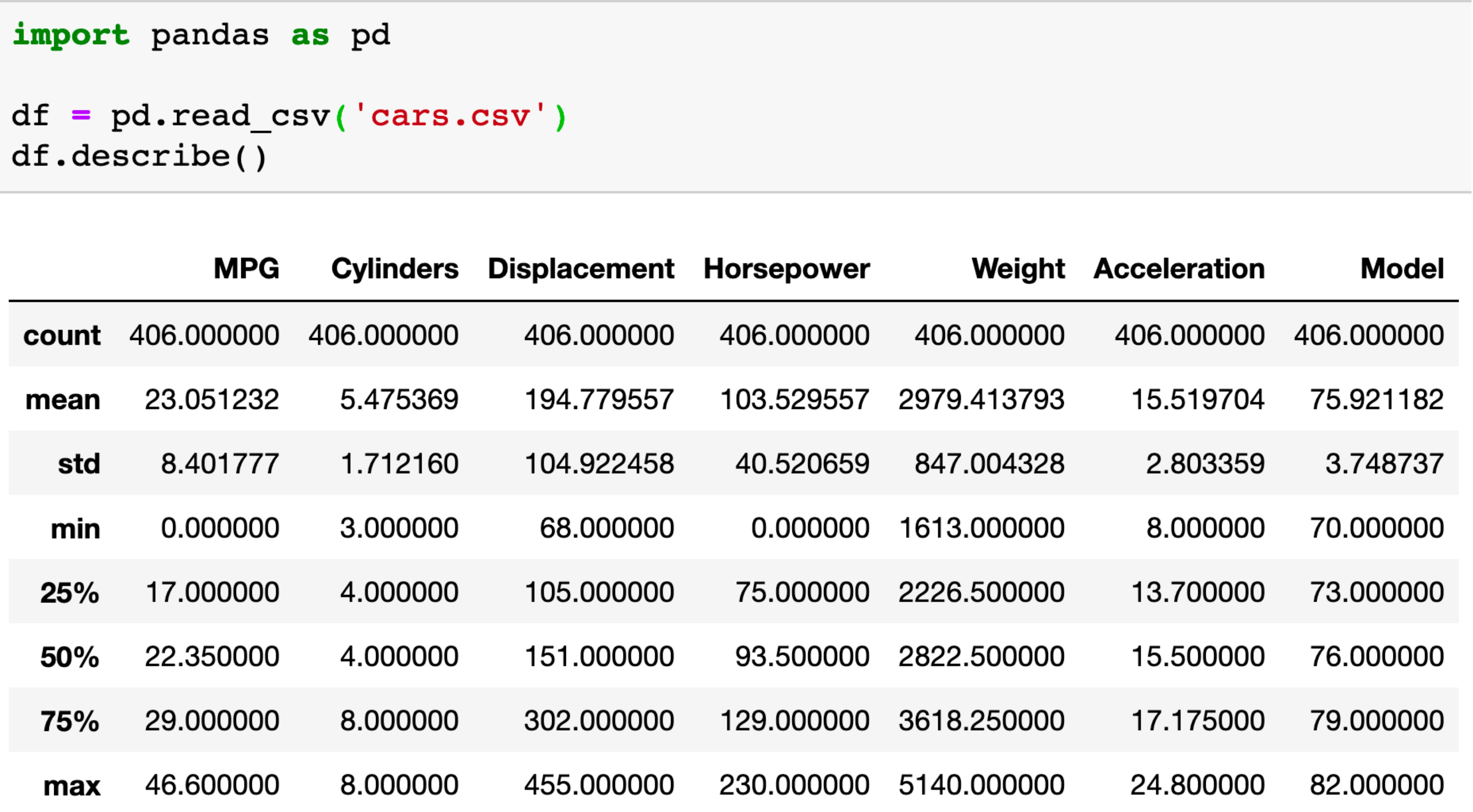


```python
import pandas as pd

df = pd.read_csv('cars.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 406 entries, 0 to 405
Data columns (total 9 columns):
Car             406 non-null object
MPG             406 non-null float64
Cylinders       406 non-null int64
Displacement    406 non-null float64
Horsepower      406 non-null int64
Weight          406 non-null int64
Acceleration    406 non-null float64
Model           406 non-null int64
Origin          406 non-null object
dtypes: float64(3), int64(4), object(2)
memory usage: 28.6+ KB
```

Tabular Data

- Statistics of numerical features

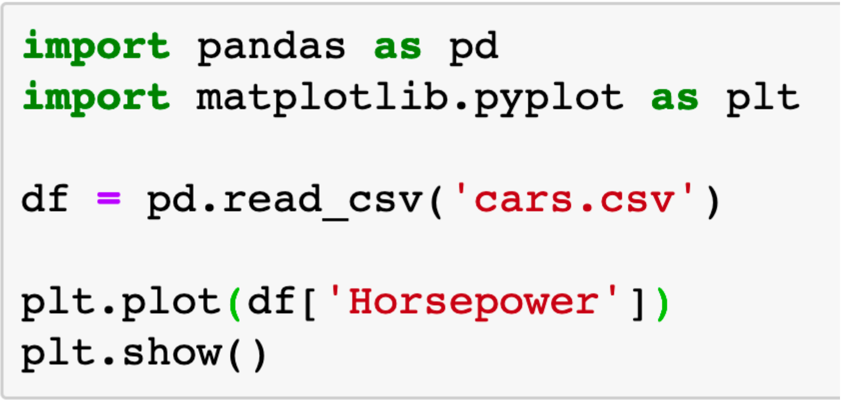


```python
import pandas as pd

df = pd.read_csv('cars.csv')
df.describe()
```

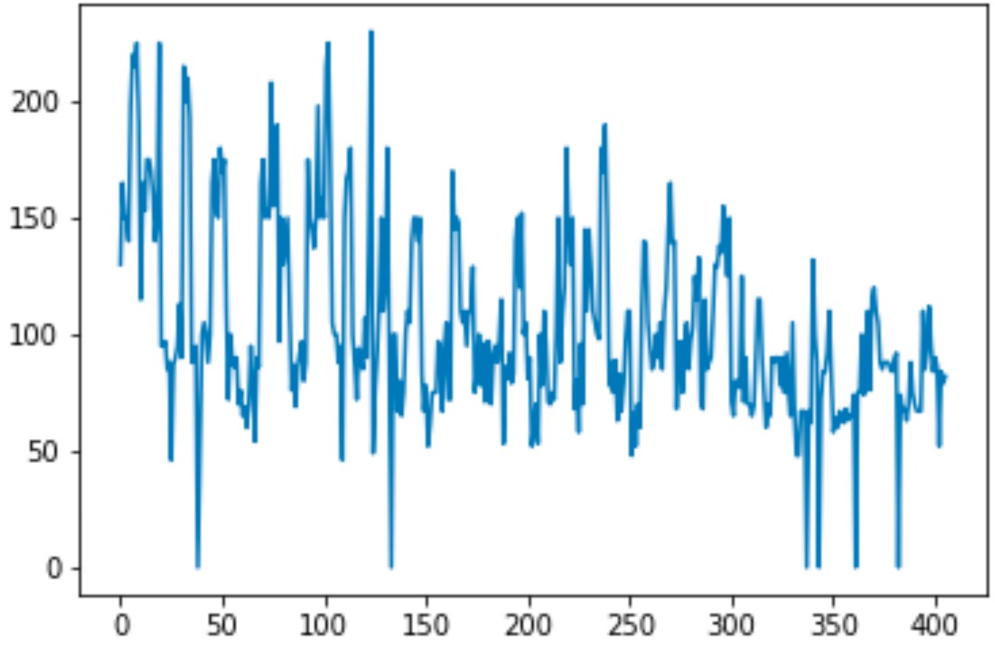
	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model
count	406.000000	406.000000	406.000000	406.000000	406.000000	406.000000	406.000000
mean	23.051232	5.475369	194.779557	103.529557	2979.413793	15.519704	75.921182
std	8.401777	1.712160	104.922458	40.520659	847.004328	2.803359	3.748737
min	0.000000	3.000000	68.000000	0.000000	1613.000000	8.000000	70.000000
25%	17.000000	4.000000	105.000000	75.000000	2226.500000	13.700000	73.000000
50%	22.350000	4.000000	151.000000	93.500000	2822.500000	15.500000	76.000000
75%	29.000000	8.000000	302.000000	129.000000	3618.250000	17.175000	79.000000
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000	82.000000

Tabular Data

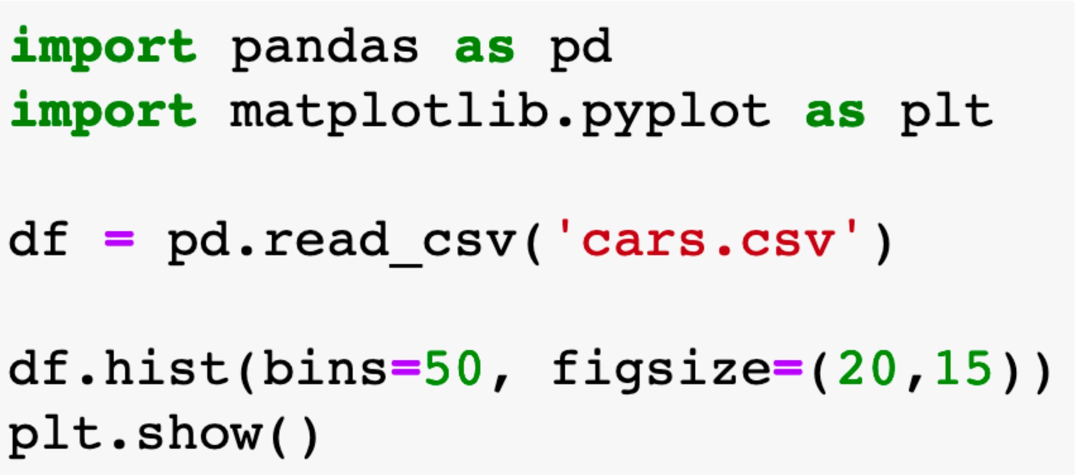
- Visualize numerical features
  - Line plot: extreme values?



```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('cars.csv')

plt.plot(df['Horsepower'])
plt.show()
```

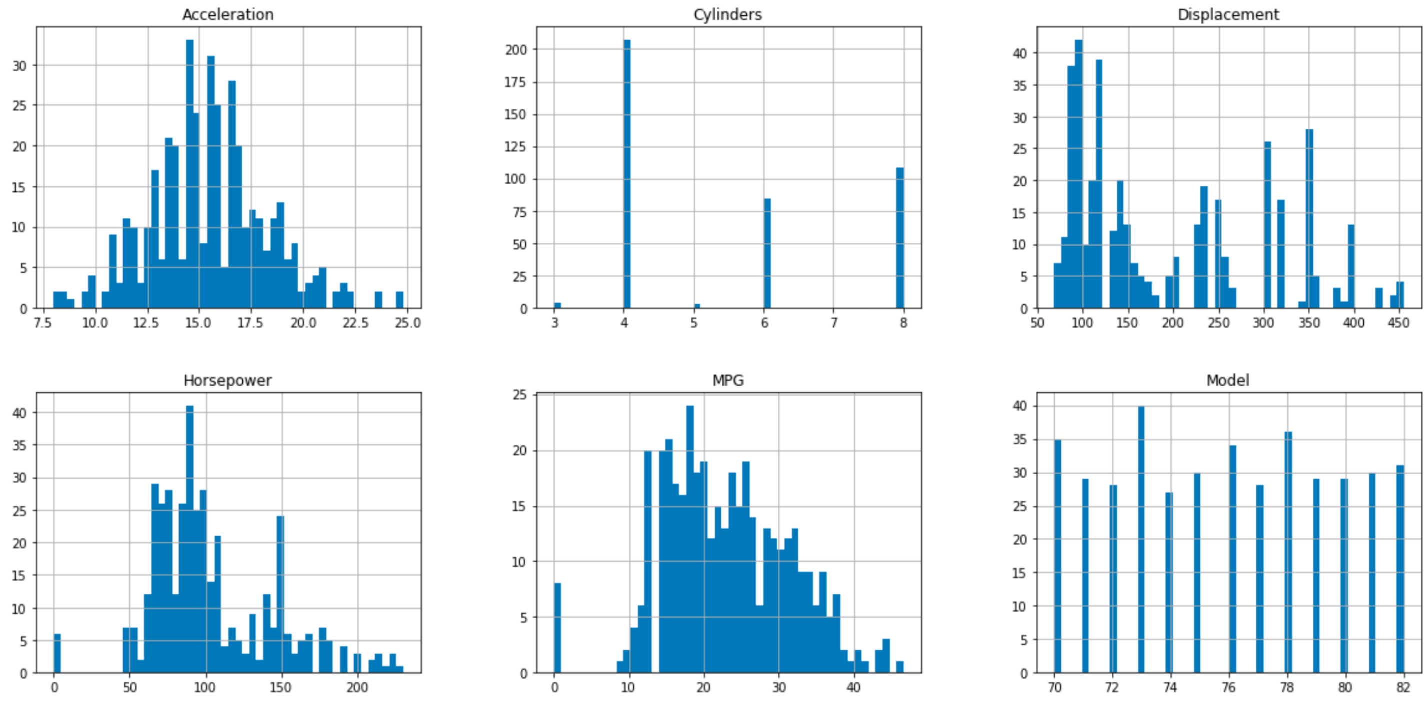


Tabular Data



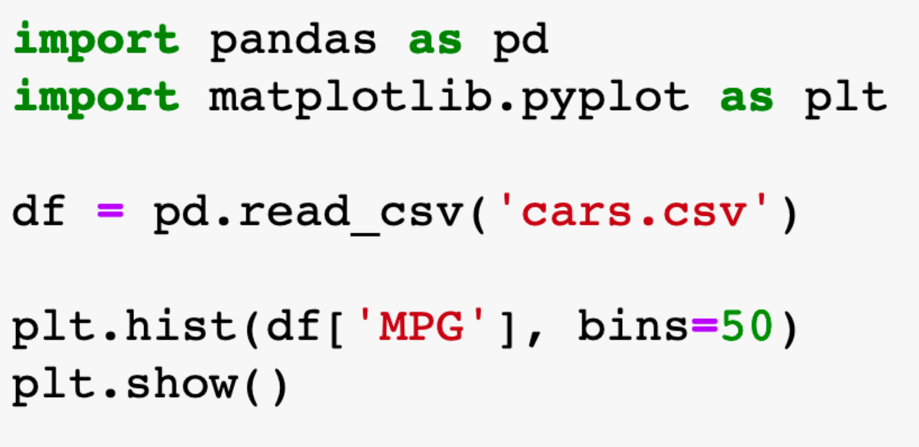
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('cars.csv')

df.hist(bins=50, figsize=(20,15))
plt.show()
```

- Visualize numerical features
  - Hist plot: distribution?
  - It displays the frequency of attribute values within small intervals ranging from the minimum to the maximum value of the attribute.

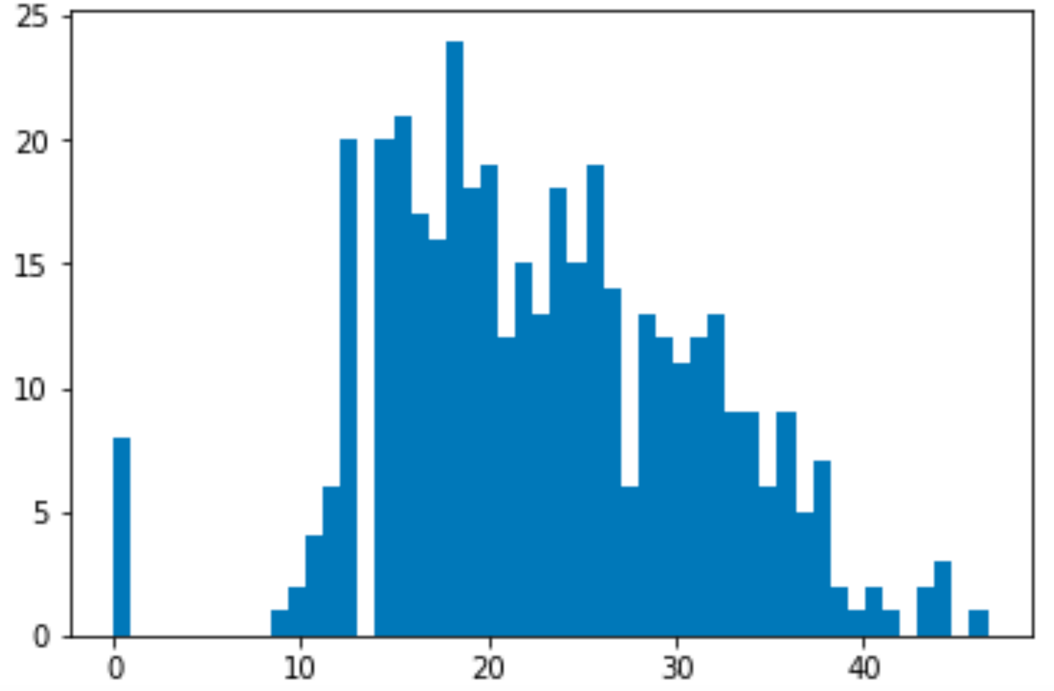


Tabular Data

- Visualize numerical features
  - Hist plot: distribution?

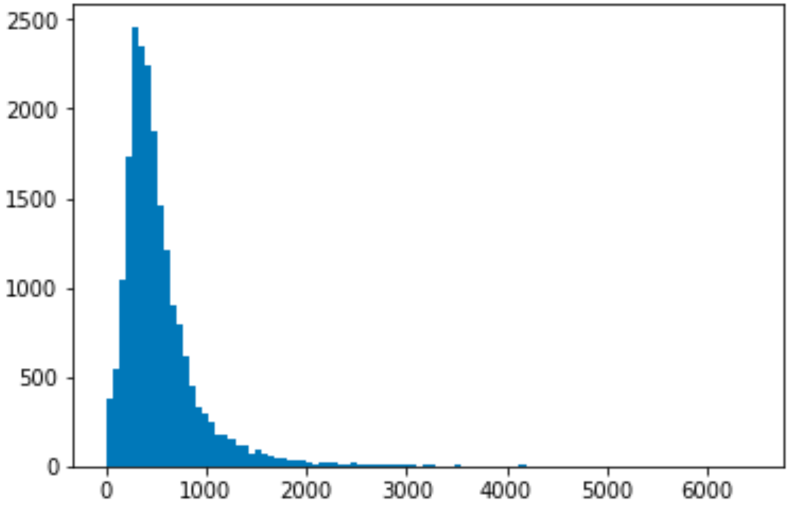


```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('cars.csv')

plt.hist(df['MPG'], bins=50)
plt.show()
```

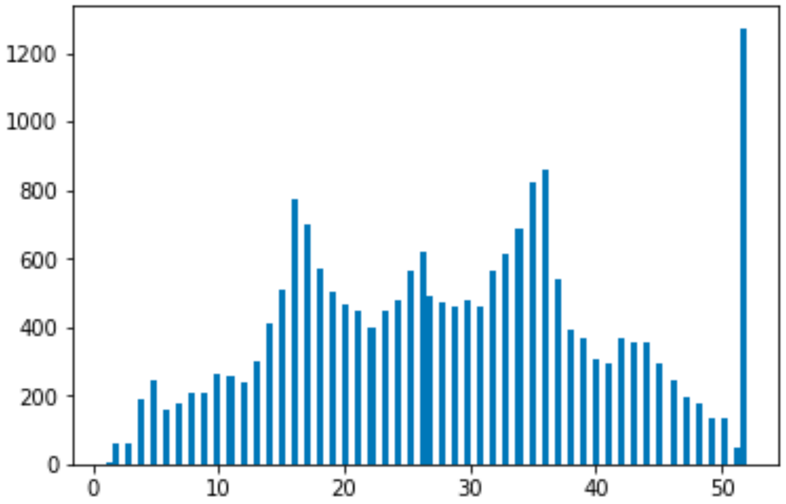


Tabular Data

- Visualize numerical features
  - Long tail distribution VS non-long tail distribution



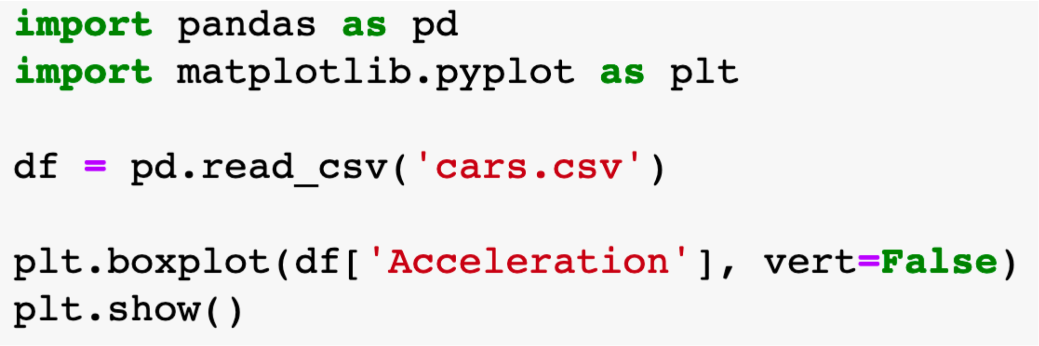
Long tail distribution
many values far from the "head" or central part of the distribution



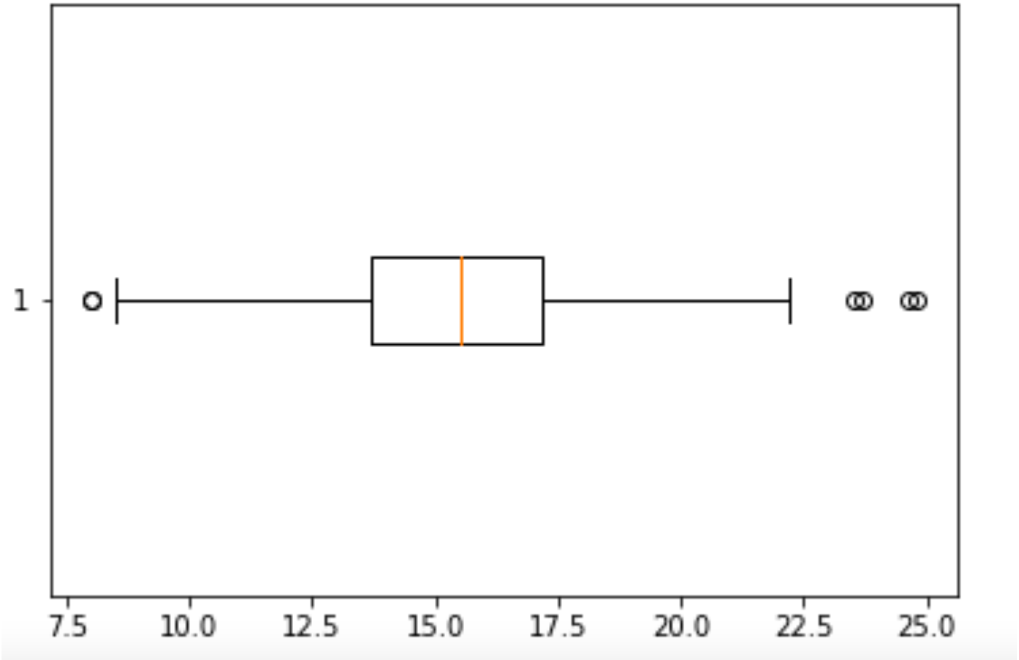
Non-long tail distribution

Tabular Data

- Visualize numerical features
  - Boxplot: show statistics of features
    - Minimum
    - Maximum
    - Media
    - First quartile
    - Third quartile

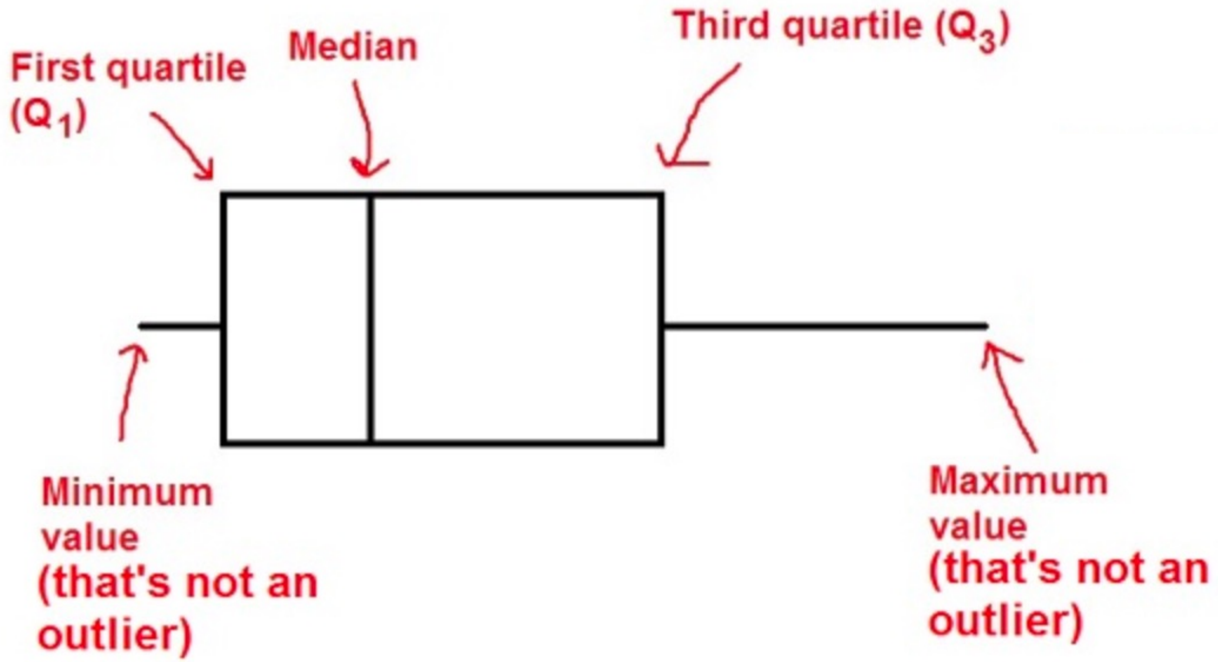


```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('cars.csv')

plt.boxplot(df['Acceleration'], vert=False)
plt.show()
```



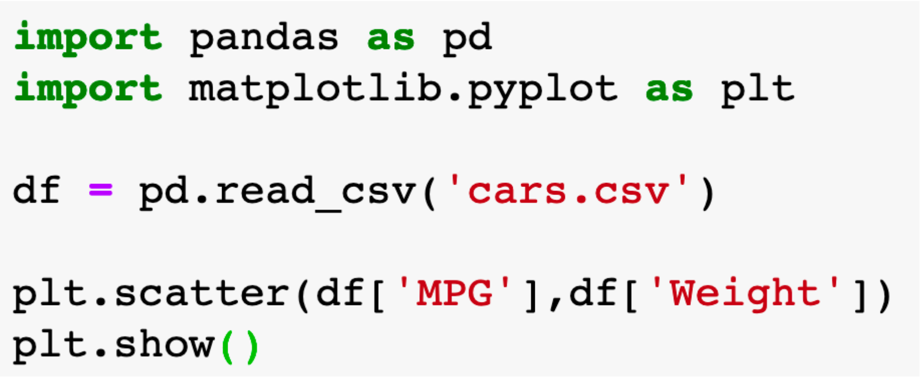
Tabular Data

- Visualize numerical features
  - Boxplot: show statistics of features

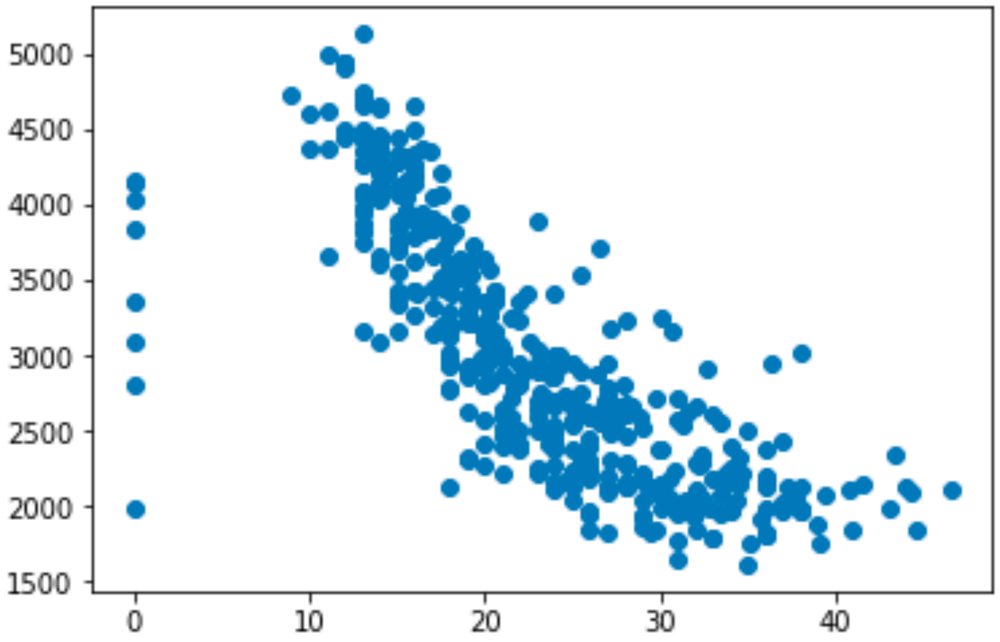

First quartile (Q_1)
Median
Third quartile (Q_3)
Minimum value (that's not an outlier)
Maximum value (that's not an outlier)

Tabular Data

- Exploring pairs of features:
  - Disclose the *dependencies* or *correlations* between the attributes.

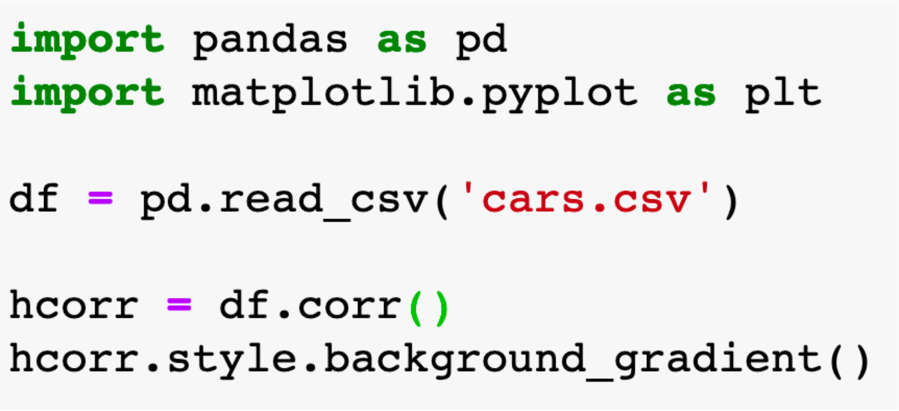


```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('cars.csv')

plt.scatter(df['MPG'],df['Weight'])
plt.show()
```

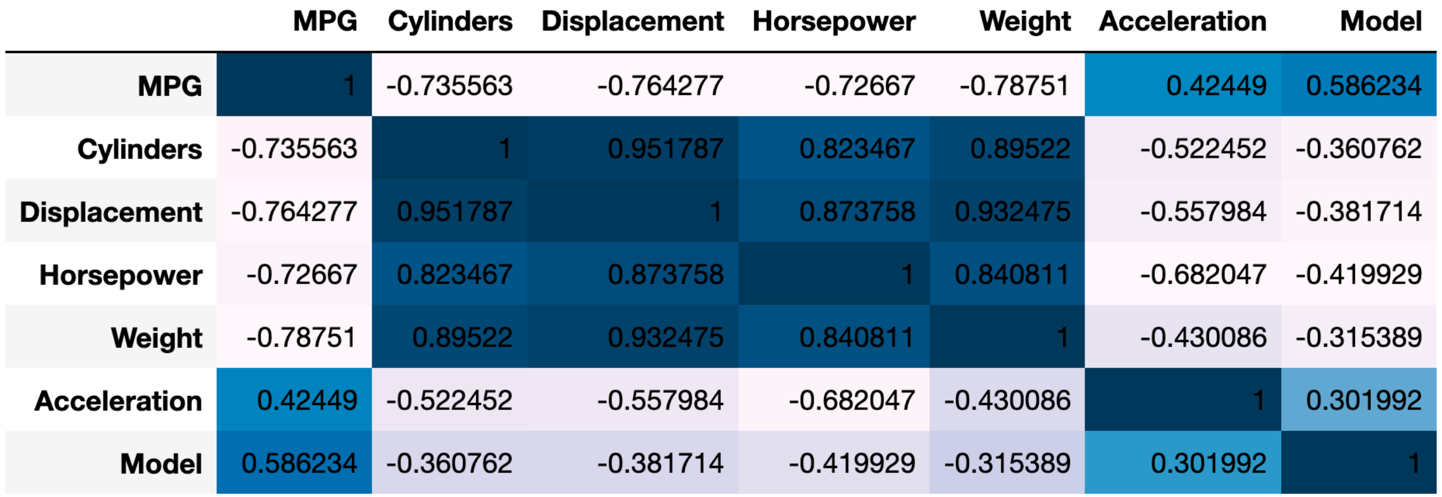


Tabular Data

- Exploring pairs of features:
  - Disclose the *dependencies* or *correlations* between the attributes.



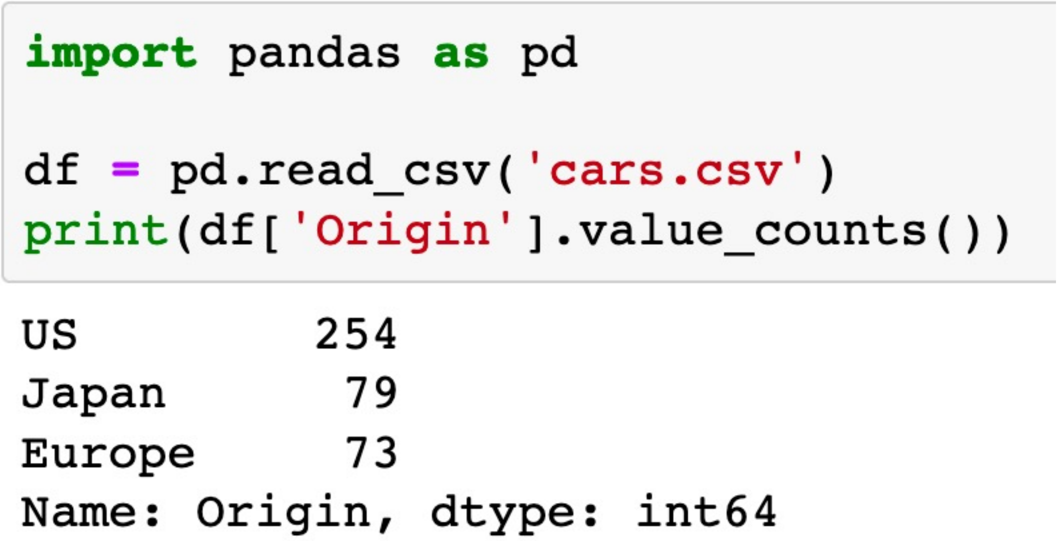
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('cars.csv')

hcorr = df.corr()
hcorr.style.background_gradient()
```



	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model
MPG	1	-0.735563	-0.764277	-0.72667	-0.78751	0.42449	0.586234
Cylinders	-0.735563	1	0.951787	0.823467	0.89522	-0.522452	-0.360762
Displacement	-0.764277	0.951787	1	0.873758	0.932475	-0.557984	-0.381714
Horsepower	-0.72667	0.823467	0.873758	1	0.840811	-0.682047	-0.419929
Weight	-0.78751	0.89522	0.932475	0.840811	1	-0.430086	-0.315389
Acceleration	0.42449	-0.522452	-0.557984	-0.682047	-0.430086	1	0.301992
Model	0.586234	-0.360762	-0.381714	-0.419929	-0.315389	0.301992	1

Tabular Data

- Categorical features

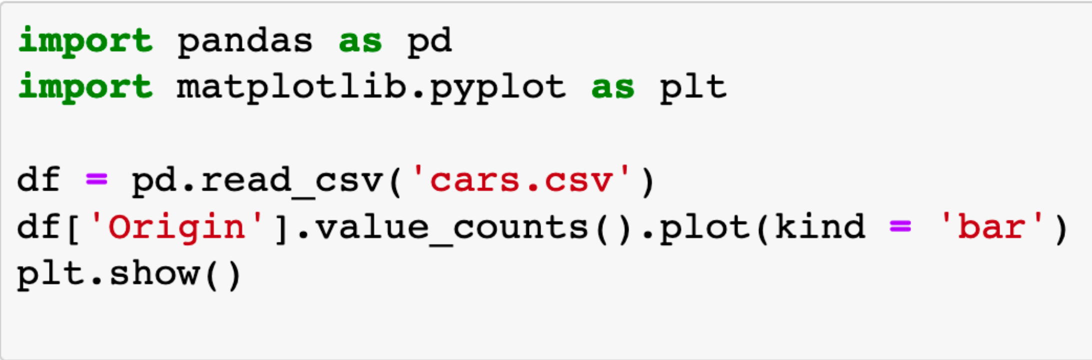


```python
import pandas as pd

df = pd.read_csv('cars.csv')
print(df['Origin'].value_counts())
```

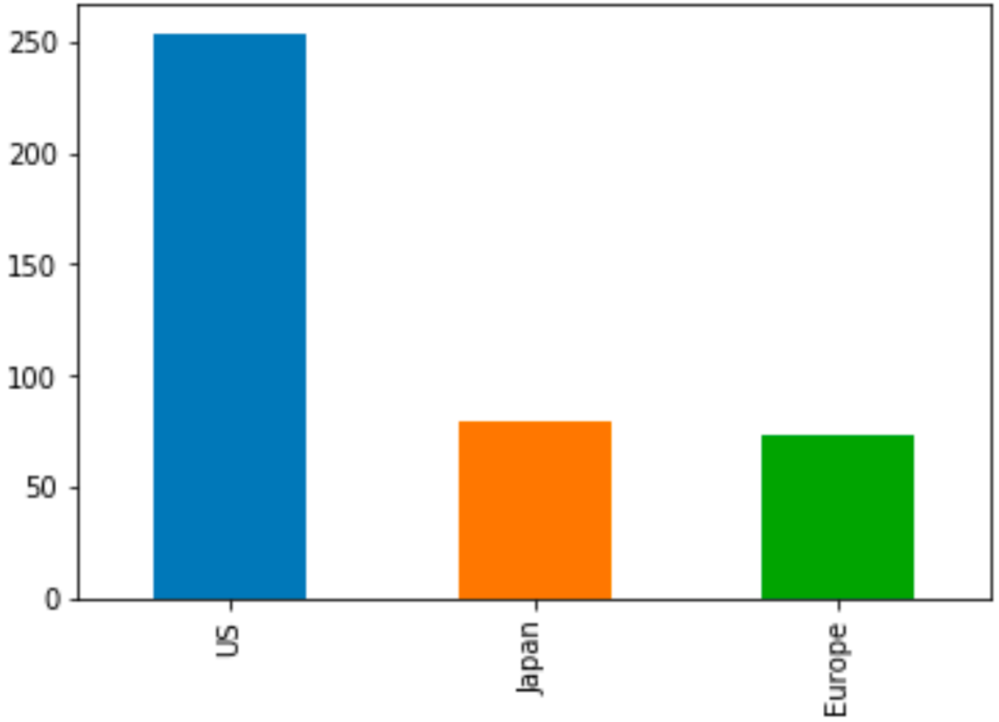
```
US          254
Japan        79
Europe       73
Name: Origin, dtype: int64
```

Tabular Data

- Visualization of categorical features

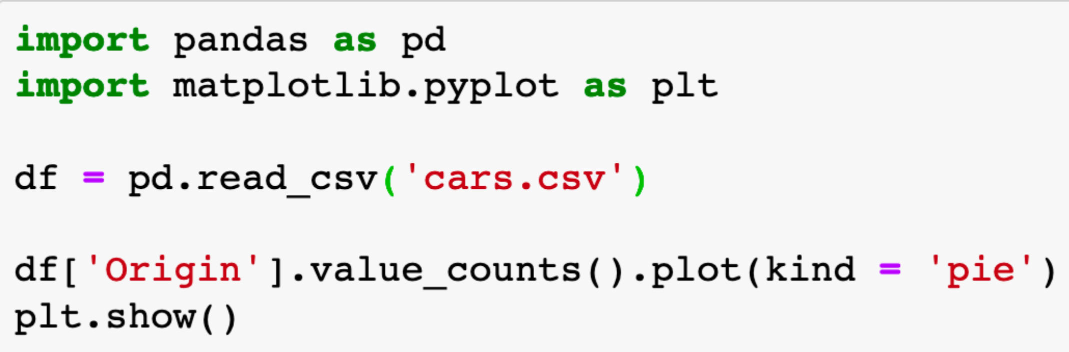


```python
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('cars.csv')
df['Origin'].value_counts().plot(kind = 'bar')
plt.show()
```

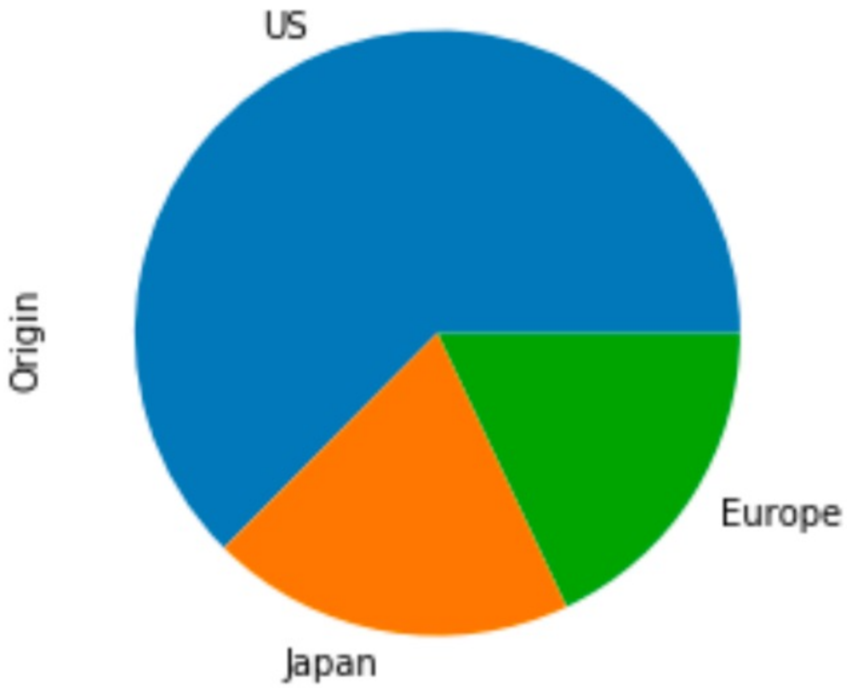


Tabular Data

- Visualization of categorical features

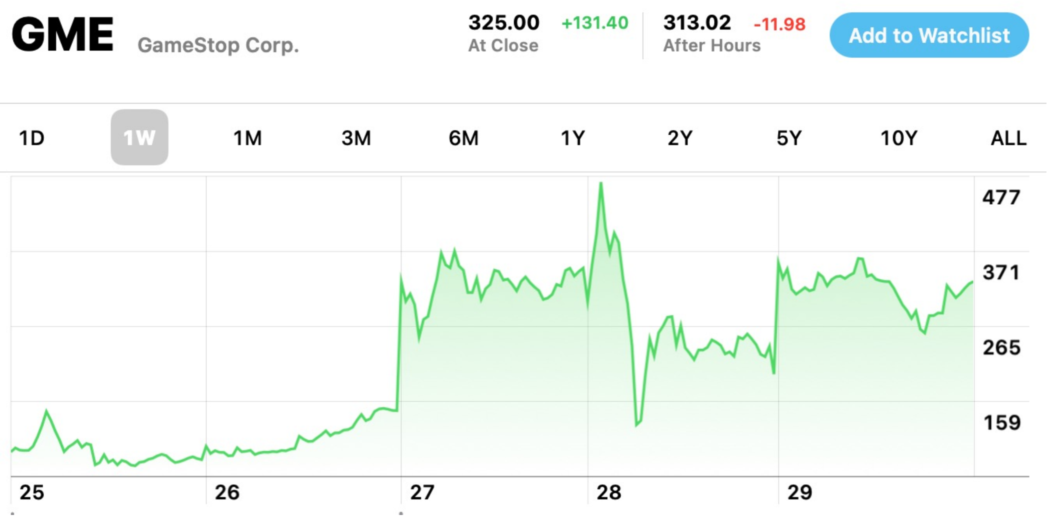


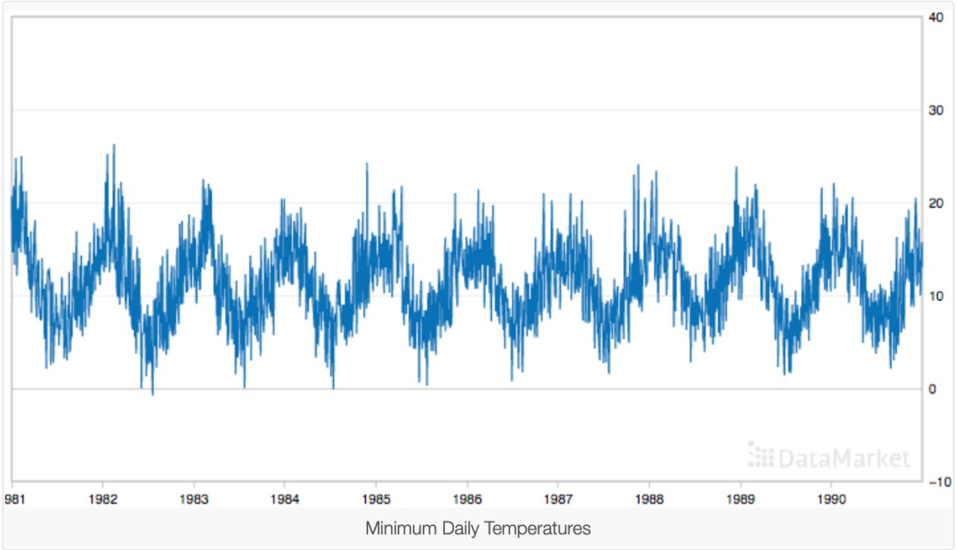
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('cars.csv')

df['Origin'].value_counts().plot(kind = 'pie')
plt.show()
```



Temporal Data

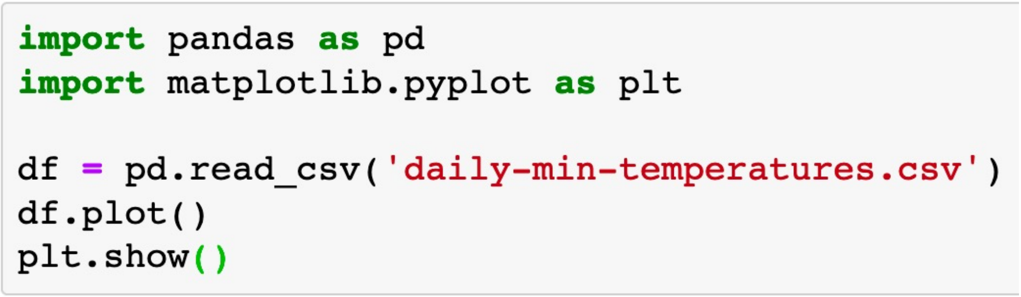
- A value or values that change in time
    - Trend, seasonality



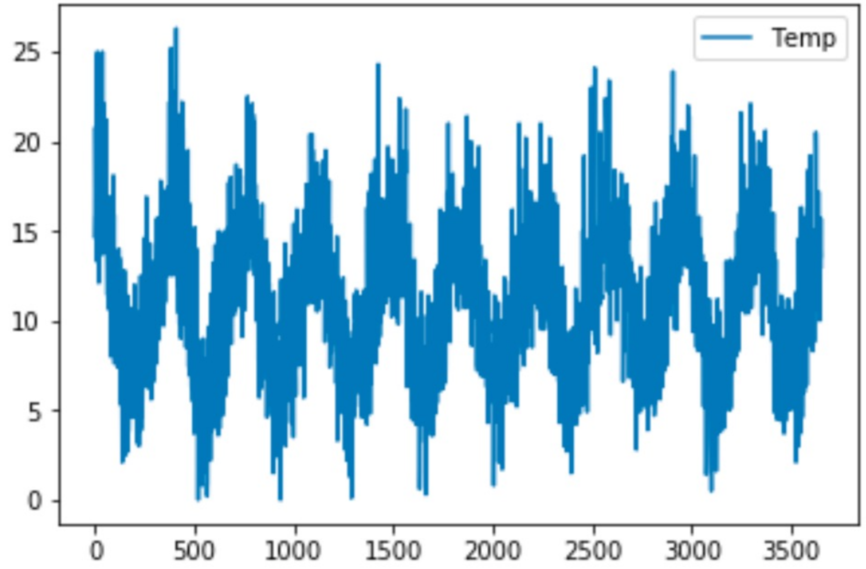


Temporal Data

- Visualization of temporal data

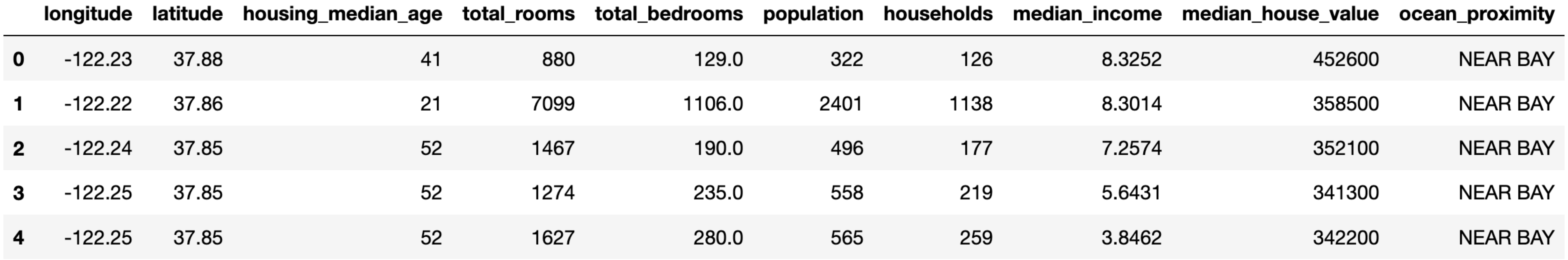


```python
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('daily-min-temperatures.csv')
df.plot()
plt.show()
```



Spatial Data

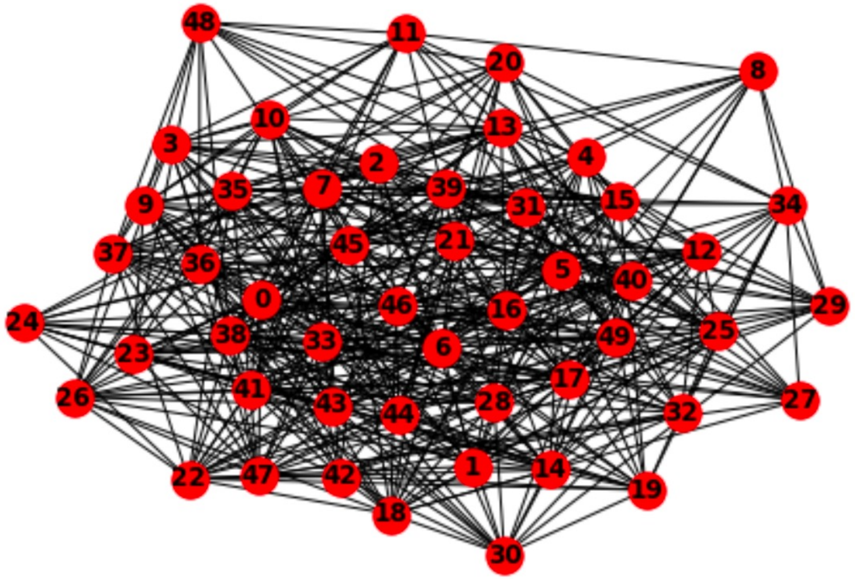
- A value or values that change over space



	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	NEAR BAY
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	NEAR BAY
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	NEAR BAY
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	NEAR BAY
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	NEAR BAY

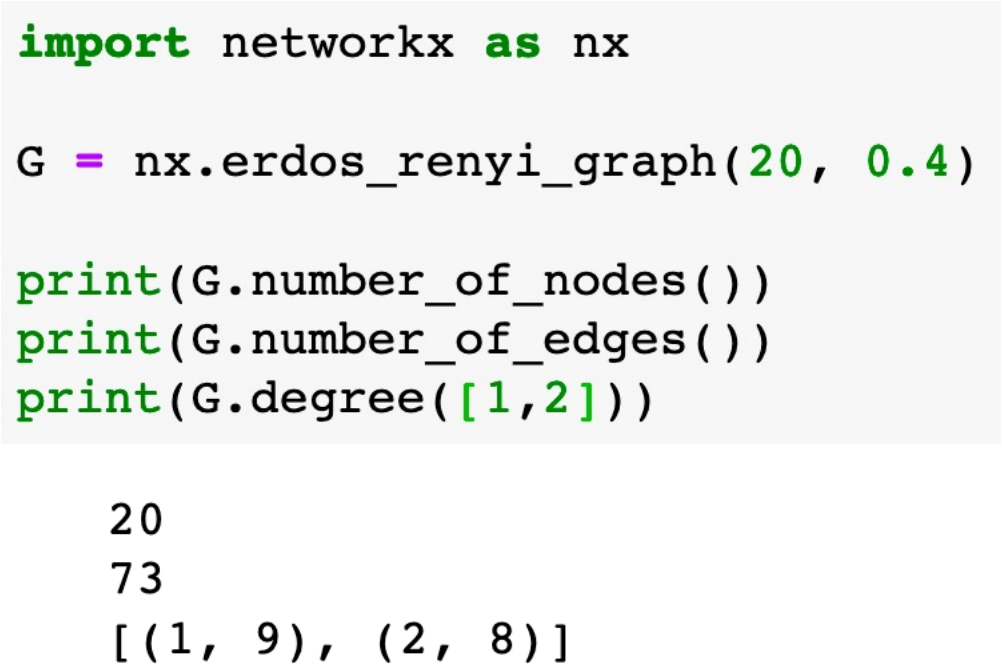
Graph Data

- Dataset is represented as a graph, where nodes and edges can have their own properties and values



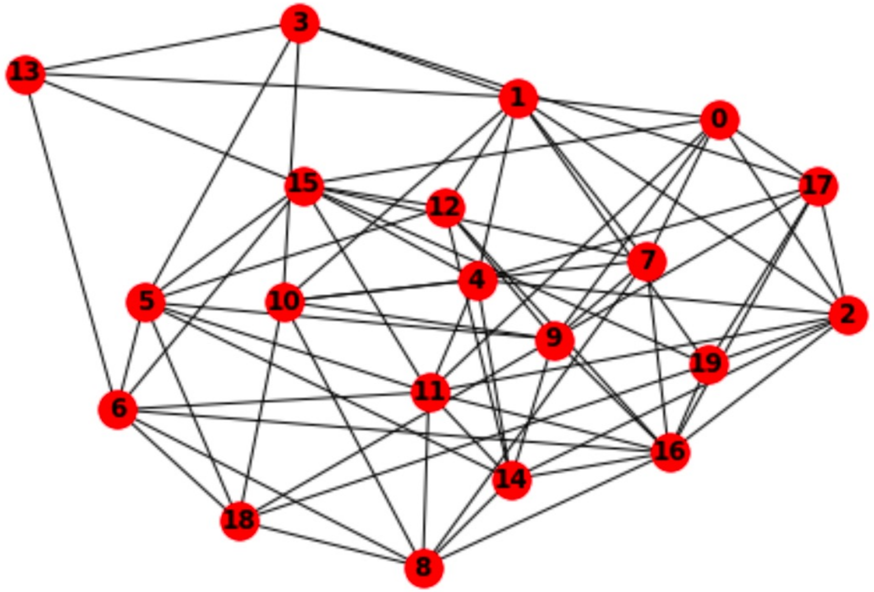
Graph Data

- Statistics of Graph
  - Nodes, edges, degrees



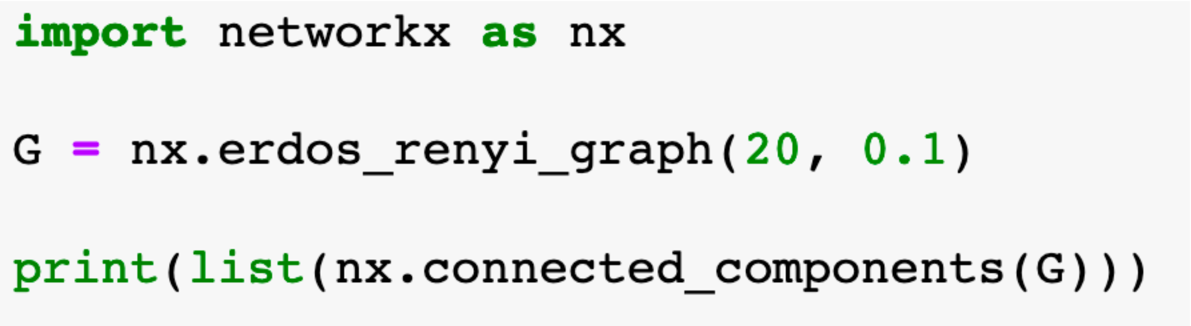
```
import networkx as nx

G = nx.erdos_renyi_graph(20, 0.4)

print(G.number_of_nodes())
print(G.number_of_edges())
print(G.degree([1,2]))
```

```
20
73
[(1, 9), (2, 8)]
```

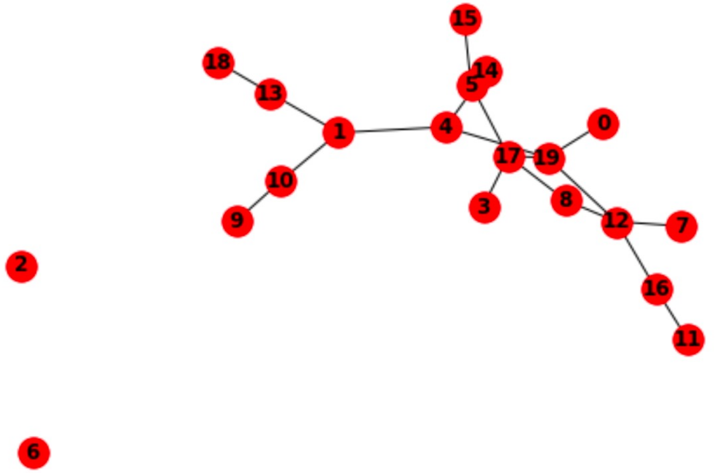


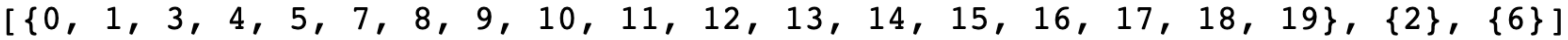
Graph Data

- Number of connected components



```
import networkx as nx

G = nx.erdos_renyi_graph(20, 0.1)

print(list(nx.connected_components(G)))
```

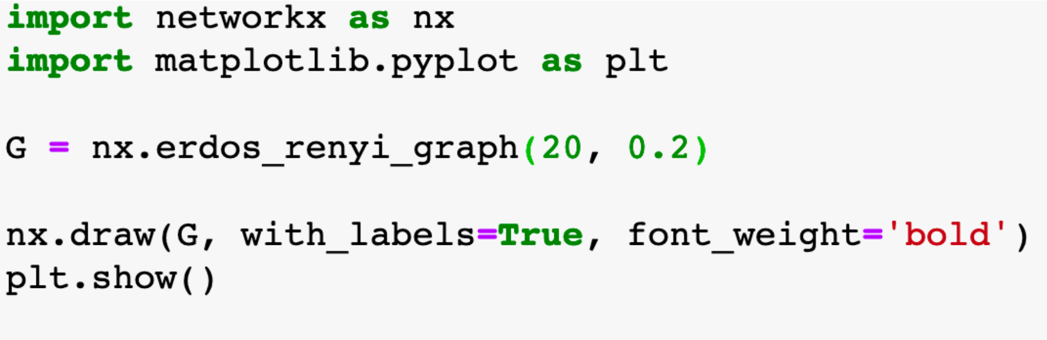




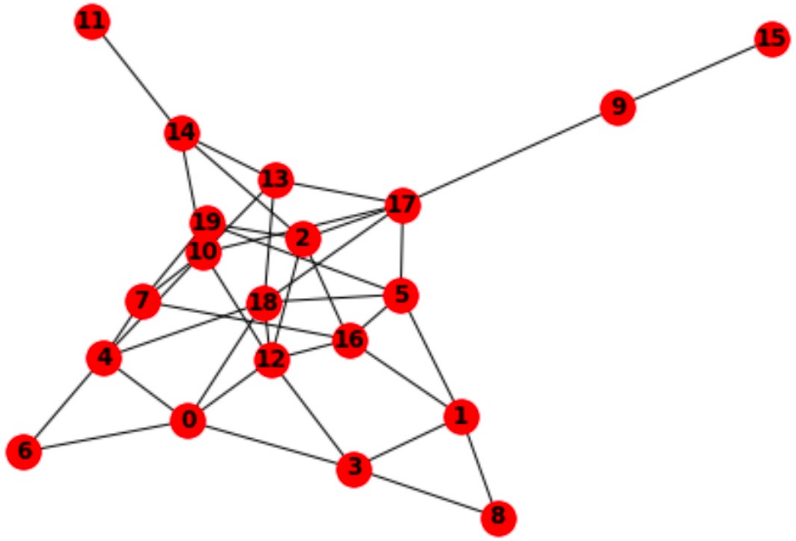
```
[{0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19}, {2}, {6}]
```

Graph Data

- Visualization of graph data



```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.erdos_renyi_graph(20, 0.2)

nx.draw(G, with_labels=True, font_weight='bold')
plt.show()
```


11
15
9
14
13
17
19
2
10
7
18
5
16
4
12
1
0
6
3
8