# AI Programming

Lecture 23

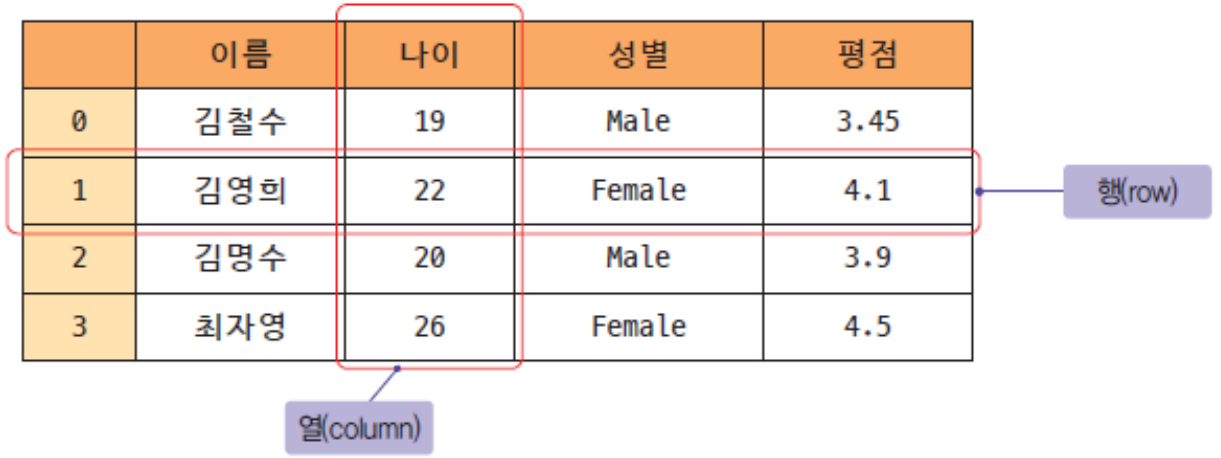
# Preview

- Python Data Analysis Library (Pandas)

# DataFrame Basics

# DataFrame

- **DataFrame 객체**
  - 행과 열로 구성된 2차원 테이블 데이터 구조
  - 열: 특성 (이름, 나이, 성별, 평점)
    - Key로 접근
  - 행: 데이터 (학생0, 학생1,...)
    - Index (숫자)로 접근



# DataFrame

- **DataFrame 생성**
  - pd.DataFrame(dictionary)

```
import pandas as pd

data = {'Name': ['Kim', 'Park', 'Lee', 'Choi'],
        'Age': [20, 23, 21, 26],
        'Grade': [3.5, 4.0, 3.3, 4.2]}
df = pd.DataFrame(data)
df
```

| | Name | Age | Grade |
|---|---|---|---|
| 0 | Kim | 20 | 3.5 |
| 1 | Park | 23 | 4.0 |
| 2 | Lee | 21 | 3.3 |
| 3 | Choi | 26 | 4.2 |

# DataFrame

- **Attributes**

| | Name | Age | Grade |
|---|---|---|---|
| 0 | Kim | 20 | 3.5 |
| 1 | Park | 23 | 4.0 |
| 2 | Lee | 21 | 3.3 |
| 3 | Choi | 26 | 4.2 |

```
df.shape
```

```
(4, 3)
```

```
df.index
```

```
RangeIndex(start=0, stop=4, step=1)
```

```
df.columns
```

```
Index(['Name', 'Age', 'Grade'], dtype='object')
```

```
len(df.index)
```

```
4
```

# DataFrame

- **Row indexing**
  - df.head(num): 처음 num개의 데이터 (행) 출력
  - df.tail(num): 마지막 num개의 데이터 (행) 출력

```
df.head(3)
```

| | Name | Age | Grade |
|---|---|---|---|
| 0 | Kim | 20 | 3.5 |
| 1 | Park | 23 | 4.0 |
| 2 | Lee | 21 | 3.3 |

```
df.tail(2)
```

| | Name | Age | Grade |
|---|---|---|---|
| 2 | Lee | 21 | 3.3 |
| 3 | Choi | 26 | 4.2 |

# DataFrame

- **Column indexing**
  - df[key] (same with the dictionary)

```
df["Age"]
```

```
0    20
1    23
2    21
3    26
Name: Age, dtype: int64
```

```
df["Name"]
```

```
0     Kim
1    Park
2     Lee
3    Choi
Name: Name, dtype: object
```

# DataFrame

- **Column slicing**
  - df[ [key1, key2,...] ]

```
df[["Age", "Name"]]
```

| | Age | Name |
|---|---|---|
| 0 | 20 | Kim |
| 1 | 23 | Park |
| 2 | 21 | Lee |
| 3 | 26 | Choi |

# DataFrame

- **Append column**
  - df[key] = list or ndarray

```
df["Dept"] = ["EE", "ICE", "CS", "ICE"]
df
```

| | Name | Age | Grade | Dept |
|---|---|---|---|---|
| 0 | Kim | 20 | 3.5 | EE |
| 1 | Park | 23 | 4.0 | ICE |
| 2 | Lee | 21 | 3.3 | CS |
| 3 | Choi | 26 | 4.2 | ICE |

```
import numpy as np

df["ID"] = np.array([2018, 2022, 2021, 2019])
df
```

| | Name | Age | Grade | Dept | ID |
|---|---|---|---|---|---|
| 0 | Kim | 20 | 3.5 | EE | 2018 |
| 1 | Park | 23 | 4.0 | ICE | 2022 |
| 2 | Lee | 21 | 3.3 | CS | 2021 |
| 3 | Choi | 26 | 4.2 | ICE | 2019 |

# DataFrame

- **Append rows**
  - df.append(dataframe, ignore_index=True)

```
tmp = pd.DataFrame(
    {"Name": ["Jung", "Oh"],
     "Age": [22, 21],
     "Grade": [4.1, 2.8],
     "Dept": ["ICE", "EE"],
     "ID": [2020, 2019]}
)
tmp
```

| | Name | Age | Grade | Dept | ID |
|---|---|---|---|---|---|
| 0 | Jung | 22 | 4.1 | ICE | 2020 |
| 1 | Oh | 21 | 2.8 | EE | 2019 |

```
df = df.append(tmp, ignore_index=True)
df
```

| | Name | Age | Grade | Dept | ID |
|---|---|---|---|---|---|
| 0 | Kim | 20 | 3.5 | EE | 2018 |
| 1 | Park | 23 | 4.0 | ICE | 2022 |
| 2 | Lee | 21 | 3.3 | CS | 2021 |
| 3 | Choi | 26 | 4.2 | ICE | 2019 |
| 4 | Jung | 22 | 4.1 | ICE | 2020 |
| 5 | Oh | 21 | 2.8 | EE | 2019 |

# DataFrame

- **Delete columns**
  - df.drop([key1, key2], axis=1, inplace=True)

```
df.drop(["Dept", "ID"], axis=1, inplace=True)
df
```

| | Name | Age | Grade |
|---|---|---|---|
| 0 | Kim | 20 | 3.5 |
| 1 | Park | 23 | 4.0 |
| 2 | Lee | 21 | 3.3 |
| 3 | Choi | 26 | 4.2 |
| 4 | Jung | 22 | 4.1 |
| 5 | Oh | 21 | 2.8 |

# DataFrame

- **Delete rows**
  - df.drop(index=num, axis=0, inplace=True)

```
df.drop(index=4, axis=0, inplace=True)
df
```

| | Name | Age | Grade |
|---|---|---|---|
| 0 | Kim | 20 | 3.5 |
| 1 | Park | 23 | 4.0 |
| 2 | Lee | 21 | 3.3 |
| 3 | Choi | 26 | 4.2 |
| 5 | Oh | 21 | 2.8 |

# DataFrame

- **Reset index**
  - df.reset_index(drop=True)

```
df = df.reset_index(drop=True)
df
```

| | Name | Age | Grade |
|---|---|---|---|
| 0 | Kim | 20 | 3.5 |
| 1 | Park | 23 | 4.0 |
| 2 | Lee | 21 | 3.3 |
| 3 | Choi | 26 | 4.2 |
| 4 | Oh | 21 | 2.8 |

# DataFrame

- **Export CSV file**
  - df.to_csv(“directory”, index=False)

```
df.to_csv("students.csv", index=False)
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Name | Age | Grade | |
| 2 | Kim | 20 | 3.5 | |
| 3 | Park | 23 | 4 | |
| 4 | Lee | 21 | 3.3 | |
| 5 | Choi | 26 | 4.2 | |
| 6 | Oh | 21 | 2.8 | |
| 7 | | | | |

# DataFrame

- **Import CSV file**
  - df = pd.read_csv(“directory”)

```
import pandas as pd

df = pd.read_csv("students.csv")
df
```

| | Name | Age | Grade |
|---|---|---|---|
| 0 | Kim | 20 | 3.5 |
| 1 | Park | 23 | 4.0 |
| 2 | Lee | 21 | 3.3 |
| 3 | Choi | 26 | 4.2 |
| 4 | Oh | 21 | 2.8 |

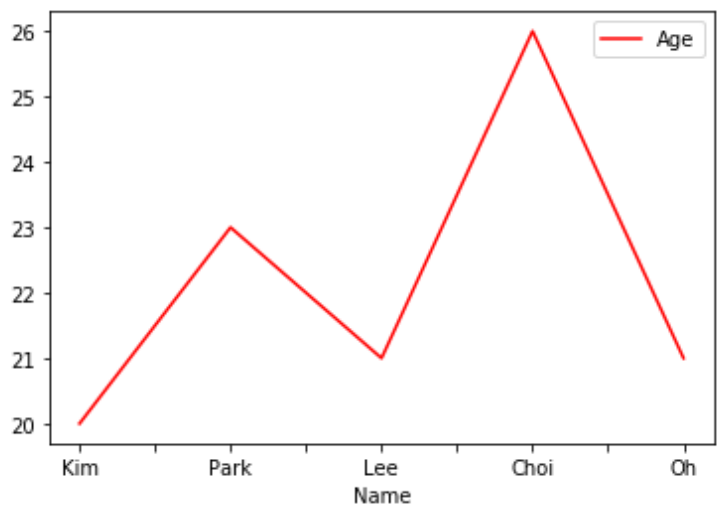
# DataFrame

- **Plots**

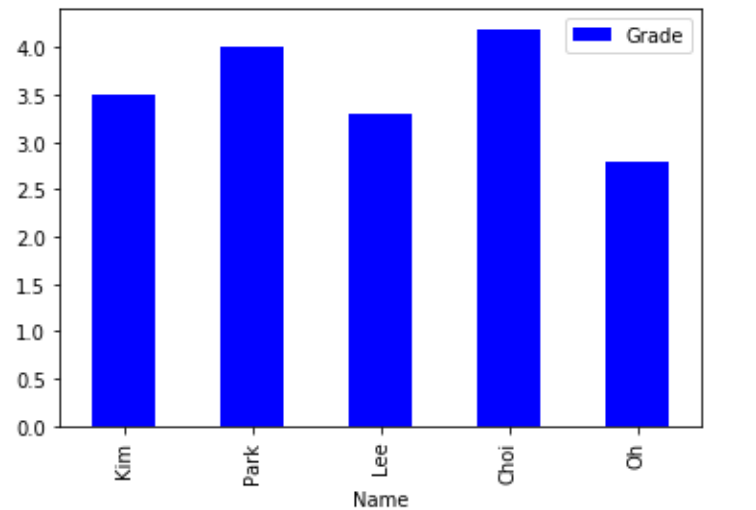
```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv("students.csv")

df.plot(kind="line", x="Name", y="Age", color="red")
plt.show()
```




```
df.plot(kind="bar", x="Name", y="Grade", color="blue")
plt.show()
```



# Titanic Dataset

# Titanic Dataset


| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Passenger | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
| 2 | 1 | 0 | 3 | Braund, M | male | 22 | 1 | 0 | A/5 21171 | 7.25 | | S |
| 3 | 2 | 1 | 1 | Cumings, I | female | 38 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 4 | 3 | 1 | 3 | Heikkinen, | female | 26 | 0 | 0 | STON/O2. | 7.925 | | S |
| 5 | 4 | 1 | 1 | Futrelle, M | female | 35 | 1 | 0 | 113803 | 53.1 | C123 | S |
| 6 | 5 | 0 | 3 | Allen, Mr. | male | 35 | 0 | 0 | 373450 | 8.05 | | S |
| 7 | 6 | 0 | 3 | Moran, Mr | male | | 0 | 0 | 330877 | 8.4583 | | Q |

# Titanic Dataset

- **Titanic dataset**
  - 타이타닉호의 총 탑승객 (2224명)의 40%인 891명에 대한 데이터
  - Survived: 생존 여부
    - 0: 사망, 1: 생존
  - Pclass: 객실 등급
    - 1: Upper, 2: Middle, 3: Lower
  - SibSp: 동반한 형제자매 및 배우자의 수
  - Parch: 동반한 부모 및 자식의 수
  - Fare: 티켓 요금
  - Cabin: 객실 번호
  - Embarked: 승선 위치

# Titanic Dataset

- **Import CSV file**

```
import pandas as pd

titanic = pd.read_csv("train.csv")
titanic
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| 1 | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 2 | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |
| 3 | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1000 | C123 | S |
| 4 | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8.0500 | NaN | S |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 886 | 887 | 0 | 2 | Montvila, Rev. Juozas | male | 27.0 | 0 | 0 | 211536 | 13.0000 | NaN | S |
| 887 | 888 | 1 | 1 | Graham, Miss. Margaret Edith | female | 19.0 | 0 | 0 | 112053 | 30.0000 | B42 | S |
| 888 | 889 | 0 | 3 | Johnston, Miss. Catherine Helen "Carrie" | female | NaN | 1 | 2 | W./C. 6607 | 23.4500 | NaN | S |
| 889 | 890 | 1 | 1 | Behr, Mr. Karl Howell | male | 26.0 | 0 | 0 | 111369 | 30.0000 | C148 | C |
| 890 | 891 | 0 | 3 | Dooley, Mr. Patrick | male | 32.0 | 0 | 0 | 370376 | 7.7500 | NaN | Q |

891 rows × 12 columns

# Titanic Dataset

- **Refinement**
  - titanic.dropna(inplace=True)
    - 결손값 (NaN) 삭제

```
titanic.dropna(inplace=True)
titanic
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1000 | C123 | S |
| 6 | 7 | 0 | 1 | McCarthy, Mr. Timothy J | male | 54.0 | 0 | 0 | 17463 | 51.8625 | E46 | S |
| 10 | 11 | 1 | 3 | Sandstrom, Miss. Marguerite Rut | female | 4.0 | 1 | 1 | PP 9549 | 16.7000 | G6 | S |
| 11 | 12 | 1 | 1 | Bonnell, Miss. Elizabeth | female | 58.0 | 0 | 0 | 113783 | 26.5500 | C103 | S |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 871 | 872 | 1 | 1 | Beckwith, Mrs. Richard Leonard (Sallie Monypeny) | female | 47.0 | 1 | 1 | 11751 | 52.5542 | D35 | S |
| 872 | 873 | 0 | 1 | Carlsson, Mr. Frans Olof | male | 33.0 | 0 | 0 | 695 | 5.0000 | B51 B53 B55 | S |
| 879 | 880 | 1 | 1 | Potter, Mrs. Thomas Jr (Lily Alexenia Wilson) | female | 56.0 | 0 | 1 | 11767 | 83.1583 | C50 | C |
| 887 | 888 | 1 | 1 | Graham, Miss. Margaret Edith | female | 19.0 | 0 | 0 | 112053 | 30.0000 | B42 | S |
| 889 | 890 | 1 | 1 | Behr, Mr. Karl Howell | male | 26.0 | 0 | 0 | 111369 | 30.0000 | C148 | C |

183 rows × 12 columns

# Titanic Dataset

- **Filtering**
  - 20세 미만 승객 접근
    - titanic[titanic[“Age”] < 20]

```
import numpy as np

a = np.array([1, 2, 3, 4, 5])
print(a < 3)
print(a[a < 3])
```

```
[ True  True False False False]
[1 2]
```

```
below20 = titanic[ titanic["Age"] < 20 ]
below20
```

| | PassengerId | Survived | Pclass | Name |
|---|---|---|---|---|
| 7 | 8 | 0 | 3 | Palsson, Master. Gosta Leonard |
| 9 | 10 | 1 | 2 | Nasser, Mrs. Nicholas (Adele Achem) |
| 10 | 11 | 1 | 3 | Sandstrom, Miss. Marguerite Rut |
| 14 | 15 | 0 | 3 | Vestrom, Miss. Hulda Amanda Adolfina |
| 16 | 17 | 0 | 3 | Rice, Master. Eugene |
| ... | ... | ... | ... | ... |
| 855 | 856 | 1 | 3 | Aks, Mrs. Sam (Leah Rosen) |
| 869 | 870 | 1 | 3 | Johnson, Master. Harold Theodor |
| 875 | 876 | 1 | 3 | Najib, Miss. Adele Kiamie "Jane" |
| 877 | 878 | 0 | 3 | Petroff, Mr. Nedelio |
| 887 | 888 | 1 | 1 | Graham, Miss. Margaret Edith |

164 rows × 12 columns

# Titanic Dataset

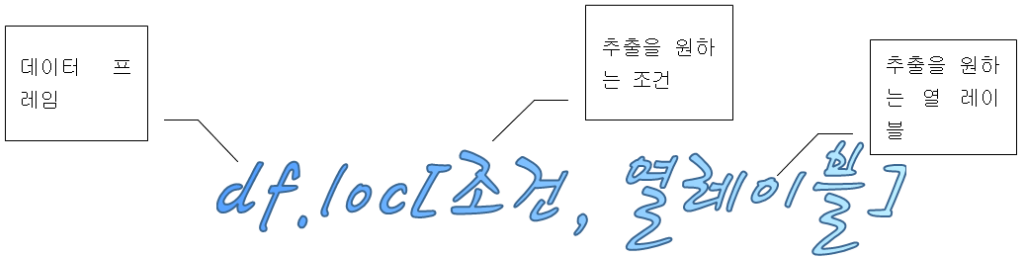
- **Filtering**
  - 1등석 or 2등석에 탑승한 승객 접근
    - titanic[ titanic[“Pclass”].isin([1, 2]) ]

```
titanic[ titanic["Pclass"] == 1 ]
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1000 | C123 | S |
| **6** | 7 | 0 | 1 | McCarthy, Mr. Timothy J | male | 54.0 | 0 | 0 | 17463 | 51.8625 | E46 | S |
| **11** | 12 | 1 | 1 | Bonnell, Miss. Elizabeth | female | 58.0 | 0 | 0 | 113783 | 26.5500 | C103 | S |
| **23** | 24 | 1 | 1 | Sloper, Mr. William Thompson | male | 28.0 | 0 | 0 | 113788 | 35.5000 | A6 | S |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

# Titanic Dataset

- **Filtering**
  - 20세 미만 승객의 이름 특성 (열) 접근
    - titanic.loc[ titanic[“Age”] < 20, “Name” ]




```
titanic.loc[ titanic["Age"]<20, "Name" ]
```

```
7            Palsson, Master. Gosta Leonard
9         Nasser, Mrs. Nicholas (Adele Achem)
10           Sandstrom, Miss. Marguerite Rut
14       Vestrom, Miss. Hulda Amanda Adolfina
16                       Rice, Master. Eugene
                                ...
855                Aks, Mrs. Sam (Leah Rosen)
869          Johnson, Master. Harold Theodor
875        Najib, Miss. Adele Kiamie "Jane"
877                     Petroff, Mr. Nedelio
887              Graham, Miss. Margaret Edith
Name: Name, Length: 164, dtype: object
```

# Titanic Dataset

- **Statistics**

```
titanic.describe()
```

| | PassengerId | Survived | Pclass | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|---|
| count | 891.000000 | 891.000000 | 891.000000 | 714.000000 | 891.000000 | 891.000000 | 891.000000 |
| mean | 446.000000 | 0.383838 | 2.308642 | 29.699118 | 0.523008 | 0.381594 | 32.204208 |
| std | 257.353842 | 0.486592 | 0.836071 | 14.526497 | 1.102743 | 0.806057 | 49.693429 |
| min | 1.000000 | 0.000000 | 1.000000 | 0.420000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 223.500000 | 0.000000 | 2.000000 | 20.125000 | 0.000000 | 0.000000 | 7.910400 |
| 50% | 446.000000 | 0.000000 | 3.000000 | 28.000000 | 0.000000 | 0.000000 | 14.454200 |
| 75% | 668.500000 | 1.000000 | 3.000000 | 38.000000 | 1.000000 | 0.000000 | 31.000000 |
| max | 891.000000 | 1.000000 | 3.000000 | 80.000000 | 8.000000 | 6.000000 | 512.329200 |

# Titanic Dataset

- **Statistics**

```
ages = titanic["Age"]
ages.head(5)
```

```
0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
Name: Age, dtype: float64
```

```
print(ages.shape)
print(ages.ndim)
```

```
(891,)
1
```

```
print(ages.mean())    # 평균
print(ages.std())     # 표준편차
print(ages.median())  # 중앙값
print(ages.max())     # 최대값
print(ages.min())     # 최소값
```

```
29.699117647058822
14.526497332334044
28.0
80.0
0.42
```

# Titanic Dataset

- **Group statistics**

```
titanic[["Age", "Fare"]].median()
```

```
Age      28.0000
Fare     14.4542
dtype: float64
```

```
titanic[["Age", "Fare"]].mean()
```

```
Age      29.699118
Fare     32.204208
dtype: float64
```

# Titanic Dataset

- **Conditional statistics**
  - 생존 여부에 대한 평균값

```
titanic.groupby("Survived").mean()
```

| Survived | PassengerId | Pclass | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|
| 0 | 447.016393 | 2.531876 | 30.626179 | 0.553734 | 0.329690 | 22.117887 |
| 1 | 444.368421 | 1.950292 | 28.343690 | 0.473684 | 0.464912 | 48.395408 |

```
titanic[["Survived", "Age"]].groupby("Survived").mean()
```

| Survived | Age |
|---|---|
| 0 | 30.626179 |
| 1 | 28.343690 |

```
titanic[["Survived", "Fare"]].groupby("Survived").mean()
```

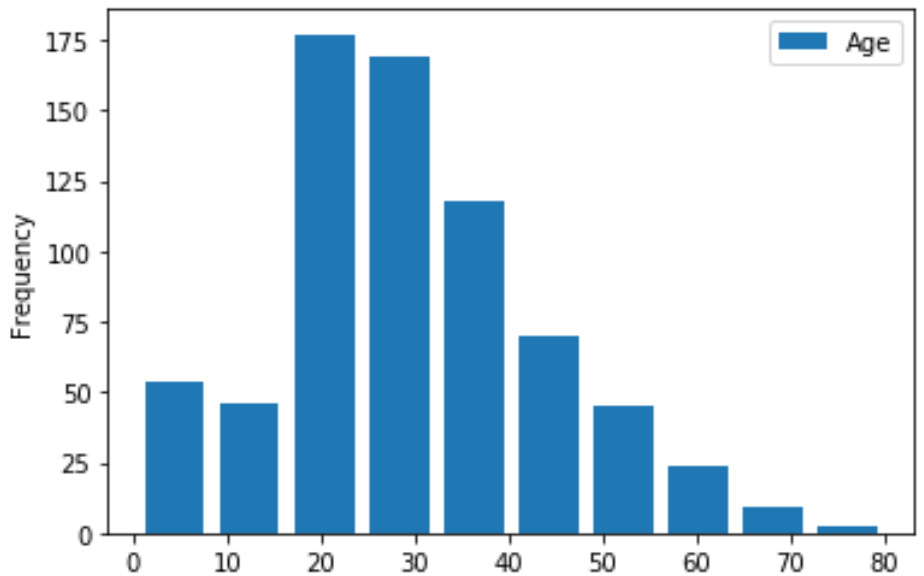
| Survived | Fare |
|---|---|
| 0 | 22.117887 |
| 1 | 48.395408 |

# Titanic Dataset

- **Histogram**

```
import matplotlib.pyplot as plt

titanic[["Age"]].plot(kind="hist", rwidth=0.8)
plt.show()
```

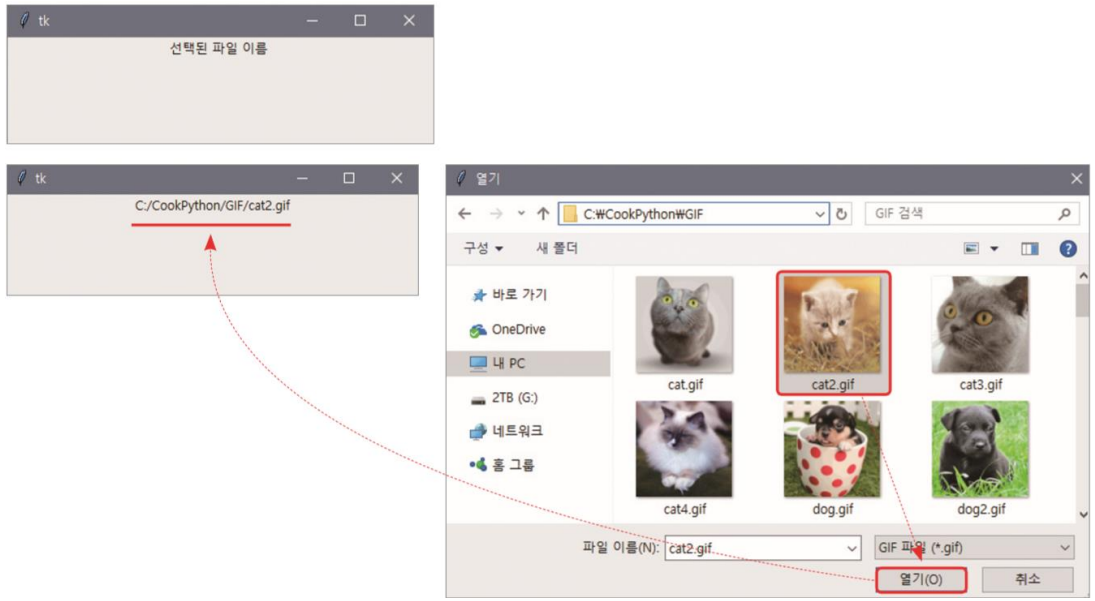


# Summary

- **Pandas**
  - DataFrame
    - Column: features
    - Row: data
  - Dictionary & CSV file

# What's Next?

- **Ch. 10 윈도우 프로그래밍**
  - 주의사항: Jupyter Notebook 사용 금지



# What's Next?

- **AI libraries**
  - Scikit-learn: 머신러닝 라이브러리
    - 머신러닝의이해
    - https://youtube.com/playlist?list=PLZreh0bM4qt4X3dpEgRDv_MK9EdQ2jvjv
  - Pytorch: 딥러닝 라이브러리
    - https://youtube.com/playlist?list=PLQ28Nx3M4JrhkqBVIXg-i5_CVVoS1UzAv
- **Embedded systems**
  - Raspberry Pi, Jetson Nano
    - 임베디드시스템설계

# What's Next?


2학년
3학년
4학년
1학기
2학기
1학기
2학기
1학기
2학기
프로그래밍및실습
AI프로그래밍
AI응용시스템
AI응용설계
머신러닝의 이해
자료구조
전자기학
전자장론
통신이론
디지털무선통신
데이터통신
컴퓨터네트워크
이동통신공학
네트워크설계
전자회로
전자회로실험
통신공학실험
전기회로이론
회로망이론
컴퓨터구조
마이크로프로세서실습
임베디드시스템설계
디지털논리회로설계
기초전기 및 디지털회로실험
제어공학
제어시스템설계
캡스톤디자인 I
캡스톤디자인 II
공학수학
ICT응용수학
신호및시스템
디지털신호처리
멀티미디어시스템설계