EDA 및 전처리

1. 데이터 시각화

SK 네트웍스

Data Insight

개요

matplotlib 기본 차트

matplotlib customization

matplotlib with pandas

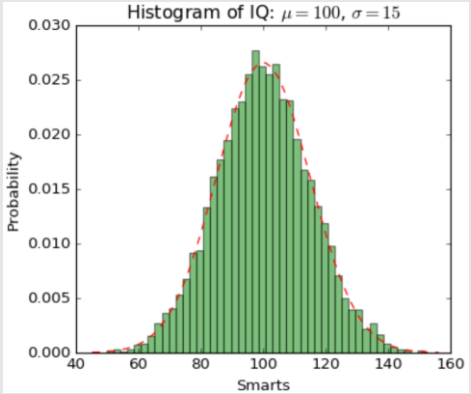
seaborn package

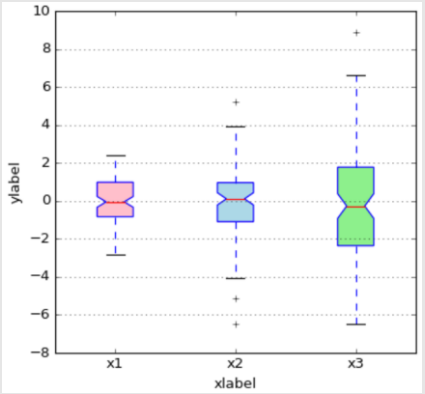
개요

- ✓ 목표
- ✓ 데이터 시각화란?

Data Visualization

- ✓ 수많은 데이터를 한눈에 파악하는 두 가지 방법
  - 기초 통계량
  - 데이터 시각화
- ✓ Very important in Data Analysis
  - 데이터 탐색.(EDA)
  - 데이터로부터 얻은 인사이트 보고.





목표

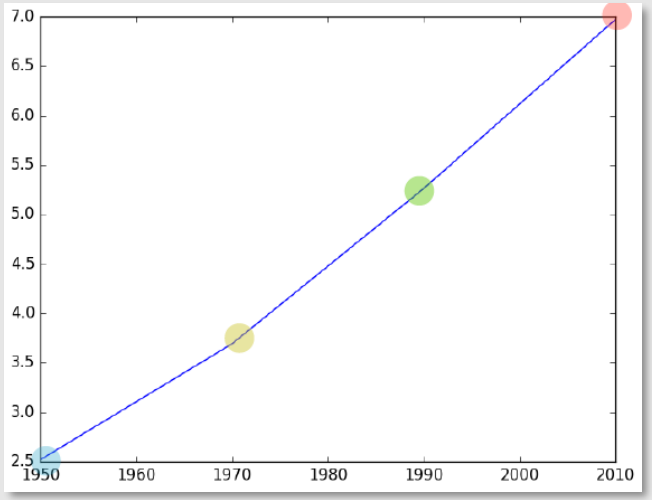
1. matplotlib 패키지의 기본 함수들을 사용할 수 있습니다.
2. 각 차트별로 원래 목적에 맞게 사용할 수 있습니다.
3. 기본 차트에 축 레이블, 타이틀 등을 추가할 수 있습니다.
4. seaborn 패키지를 이용하여 분석용 차트를 구성할 수 있습니다.

matplotlib 기본 차트

- ✓ Basic plot
- ✓ Scatter
- ✓ Histogram
- ✓ Boxplot
- ✓ Bar chart
- ✓ Pie chart

Basic Plot

- ✓ matplotlib의 subpackage인 pyplot
- ✓ plt.plot : Line Chart
  - 보통 날짜(시간) 축(X축)의 변화에 따른 연속형 변수 값의 추세를 살펴볼 때 사용
  - plt.plot(x축, y축)



Python

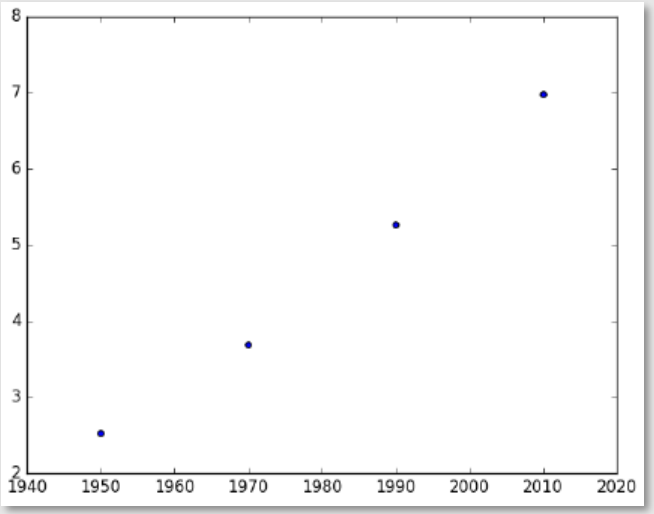
```
import matplotlib.pyplot as plt

year = [1950, 1970, 1990, 2010]

pop = [2.519, 3.692, 5.263, 6.972]

plt.plot(year, pop)
plt.show()
```

```
year = [1950, 1970, 1990, 2010]
pop = [2.519, 3.692, 5.263, 6.972]
```

Scatter Plot

✓plt.scatter() :

- 두 연속형 변수의 값의 분포(상관관계)를 살펴볼 때 사용
- 두 연속형 변수의 비 : 비율 KPI(생산성, 영업이익율 등)
- plt.scatter(x축, y축)

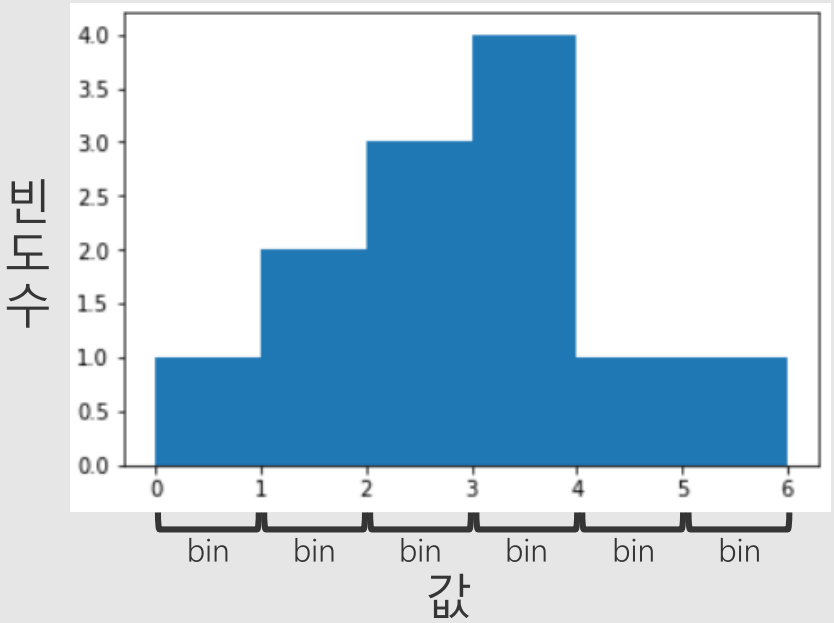


Python

```
import matplotlib.pyplot as plt

year = [1950, 1970, 1990, 2010]

pop = [2.519, 3.692, 5.263, 6.972]

plt.scatter(year, pop)
plt.show()
```

Histogram

✓연속형 변수의 분포를 살펴볼 때 사용

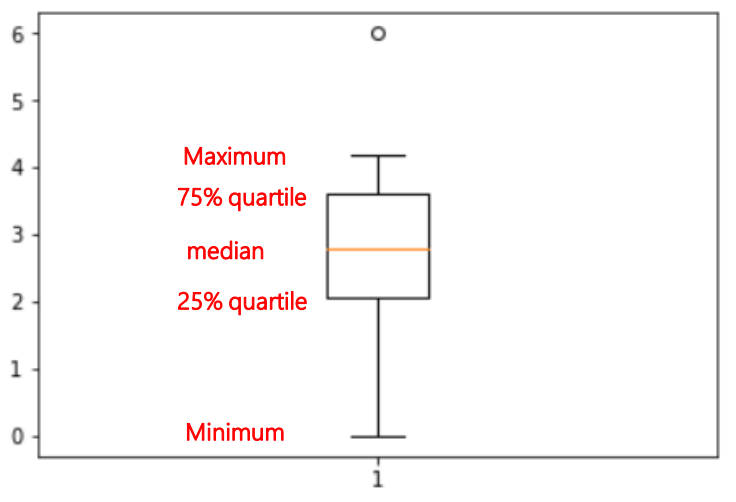


Python

```
values =
[0,1.6,1.4,2.2,2.5,2.6,3.2,3.5,3
            ,3.9,4.2,6]
plt.hist(values, bins = 6)
plt.show()
```

Box Plot

✓연속형 변수의 분포를 살펴볼 때 사용

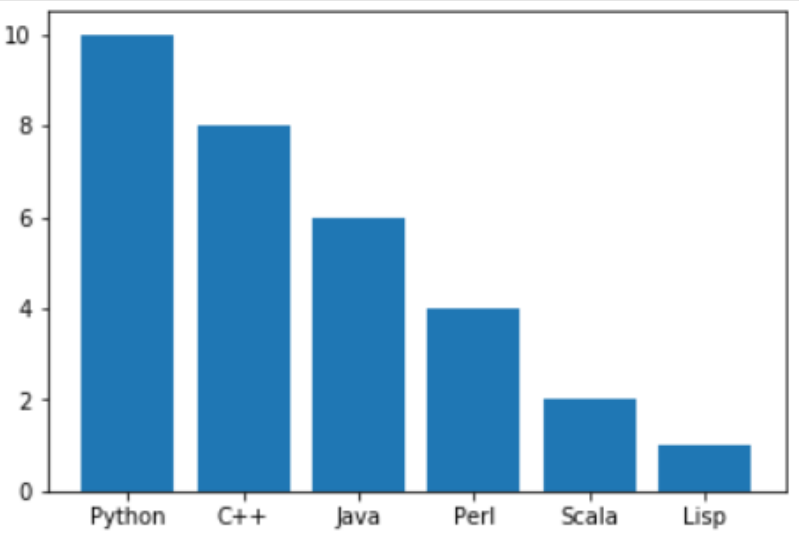


Python

```
values =
[0,1.6,1.4,2.2,2.5,2.6,3.2,3.5,3
            ,3.9,4.2,6]
plt.boxplot(values)
plt.show()
```

Bar plot

✓범주형 변수끼리 연속형 값을 비교하기 위해 사용

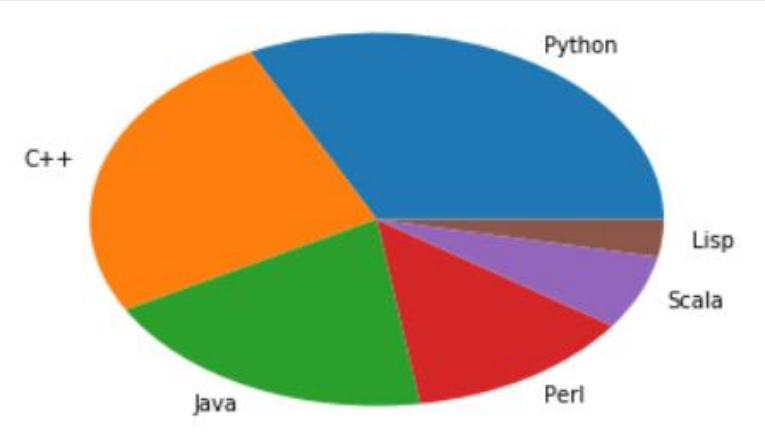


Python

```
import numpy as np
objects = ('Python', 'C++', 'Java', 'Perl',
'Scala', 'Lisp')
performance = [10,8,6,4,2,1]

plt.bar(objects, performance)
plt.show()
```

Pie Chart

✓범주형 변수끼리 연속형 값의 비율을 비교하기 위해 사용



Python

```
import numpy as np
objects = ('Python', 'C++', 'Java', 'Perl',
'Scala', 'Lisp')
performance = [10,8,6,4,2,1]

plt.pie(performance, labels =objects)
plt.show()
```

Customization

- ✓ Axis Labels & Title
- ✓ Ticks
- ✓ Common Axis
- ✓ Subplot

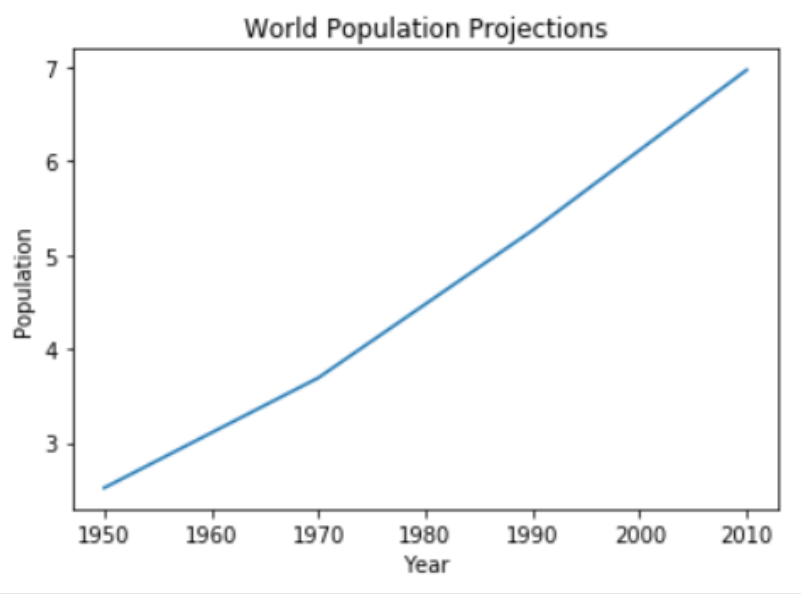
Axis Label & Title

✓Axis Label

- X축 : plt.xlabel('레이블 이름')
- Y축 : plt.ylabel('레이블 이름')

✓Title

- plt.title('타이틀 이름')



Python

```
import matplotlib.pyplot as plt

year = [1950, 1970, 1990, 2010]
pop = [2.519, 3.692, 5.263, 6.972]

plt.plot(year, pop)

# label
plt.xlabel('Year')
plt.ylabel('Population')

# Title
plt.title('World Population Projections')

plt.show()
```

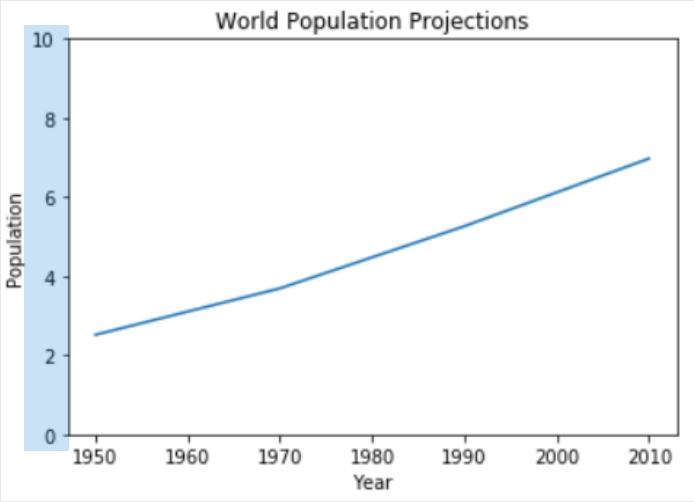
Ticks

✓Ticks : 축의 값 범위를 지정/변환

- X축 : plt.xticks([값 범위])
- Y축 : plt.yticks([값 범위])

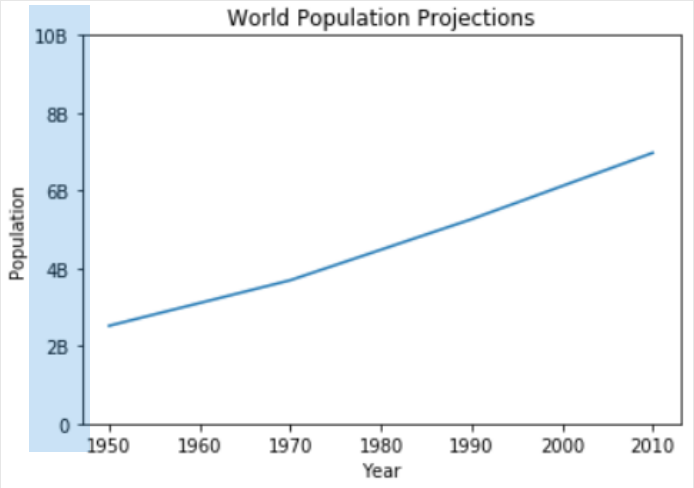
Python

```
# Ticks
plt.yticks([0, 2, 4, 6, 8, 10])
```



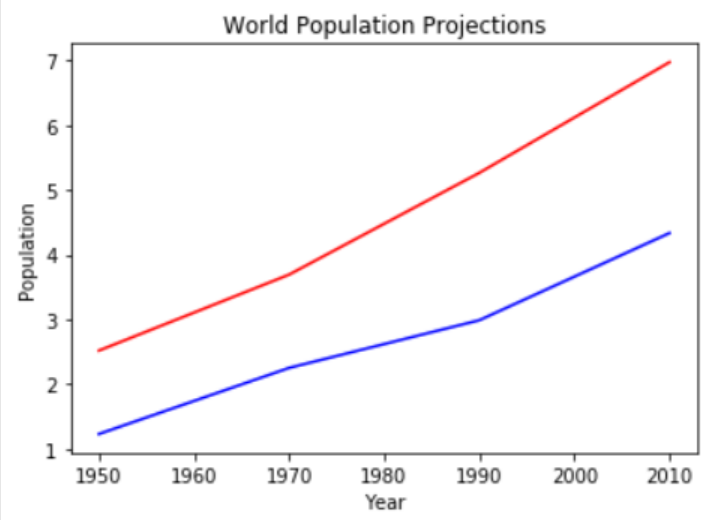
Python

```
# Ticks
plt.yticks([0, 2, 4, 6, 8, 10]
        , ['0','2B','4B','6B','8B','10B'])
```



Common Axis

✓ x축의 값을 맞춰주고
✓ 두 개의 플롯을 그리면 축을 공유



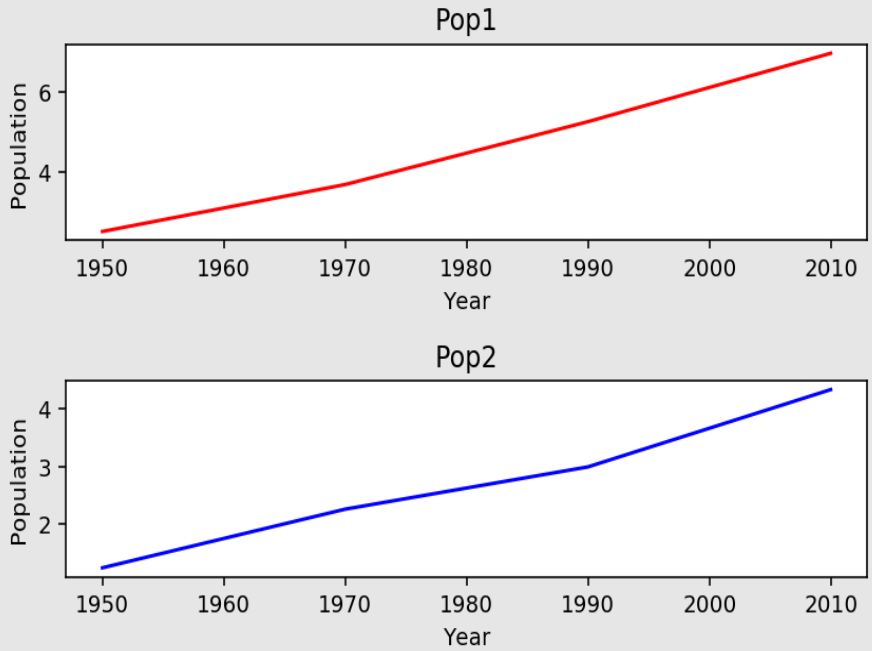
Python

```
year = [1950, 1970, 1990, 2010]
pop1 = [2.519, 3.692, 5.263, 6.972]
pop2 = [1.231, 2.252, 2.988, 4.334]

plt.plot(year, pop1, 'red')
plt.plot(year, pop2, 'blue')

# label
plt.xlabel('Year')
plt.ylabel('Population')

# Title
plt.title('World Population Projections')

plt.show()
```

Subplot

✓ 두개 이상의 차트를 한 화면에 표현

✓ subplot(nrows, ncolumns, index)



Python

```
# plot1
plt.subplot(2, 1, 1)
plt.plot(year, pop1, 'red')
plt.xlabel('Year')
plt.ylabel('Population')
plt.title('Pop1')

# plot2
plt.subplot(2, 1, 2)
plt.plot(year, pop2, 'blue')
plt.xlabel('Year')
plt.ylabel('Population')
plt.title('Pop2’)

plt.tight_layout()
plt.show()
```

matplotlib with pandas

dataframe with plot

- ✓ Pandas의 Series나 Dataframe은 plot이라는 method를 내장
- ✓ plot은 matplotlib를 내부적으로 Import하여 사용
- ✓ 종류
  - .plot()
  - .plot.bar()
  - .pie()
  - .hist()
  - .plot.kde()
  - .boxplot()
  - .scatter()
  - .area()

Scatter

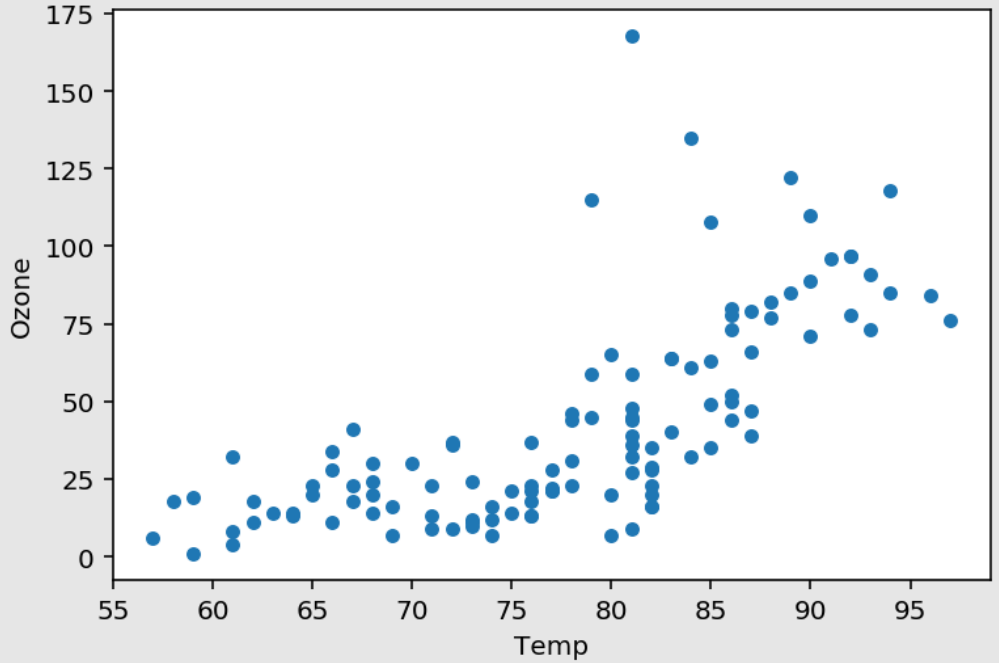
- ✓ Dataframe에 메소드로 plot을 붙여서 바로 차트를 그려볼 수 있다.

Python

```
aq = pd.read_csv("airquality.csv")
aq.plot(kind = "scatter",  x= "Temp", y = "Ozone")
```

aq.head()

	Ozone	Solar.R	Wind	Temp	Month	Day
0	41.0	190.0	7.4	67	5	1
1	36.0	118.0	8.0	72	5	2
2	12.0	149.0	12.6	74	5	3
3	18.0	313.0	11.5	62	5	4
4	NaN	NaN	14.3	56	5	5

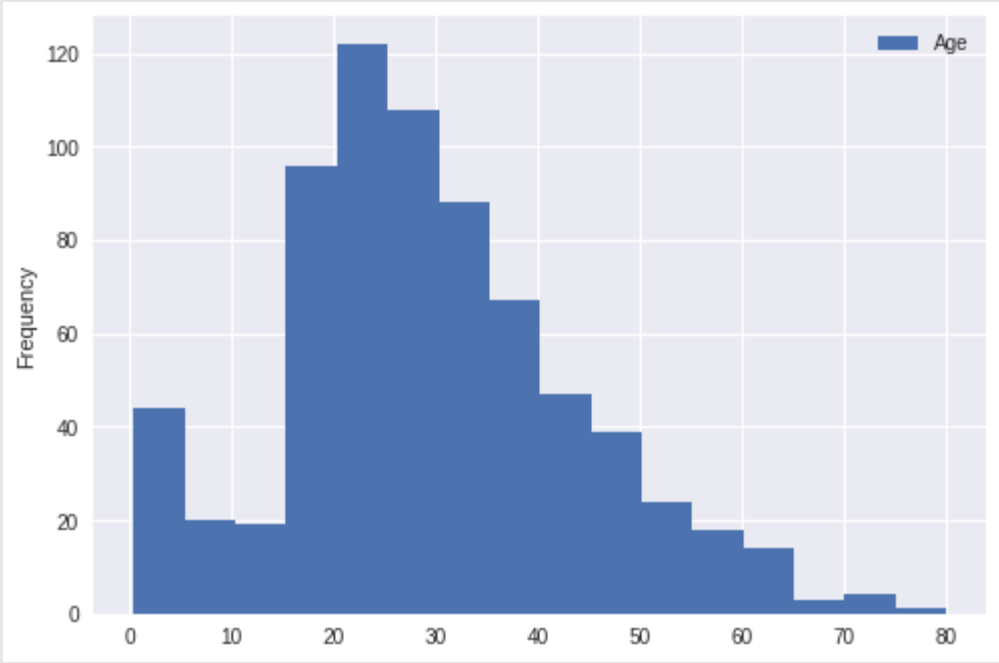


histogram

- ✓ Dataframe에 메소드로 plot을 붙여서 바로 차트를 그려볼 수 있다.

Python

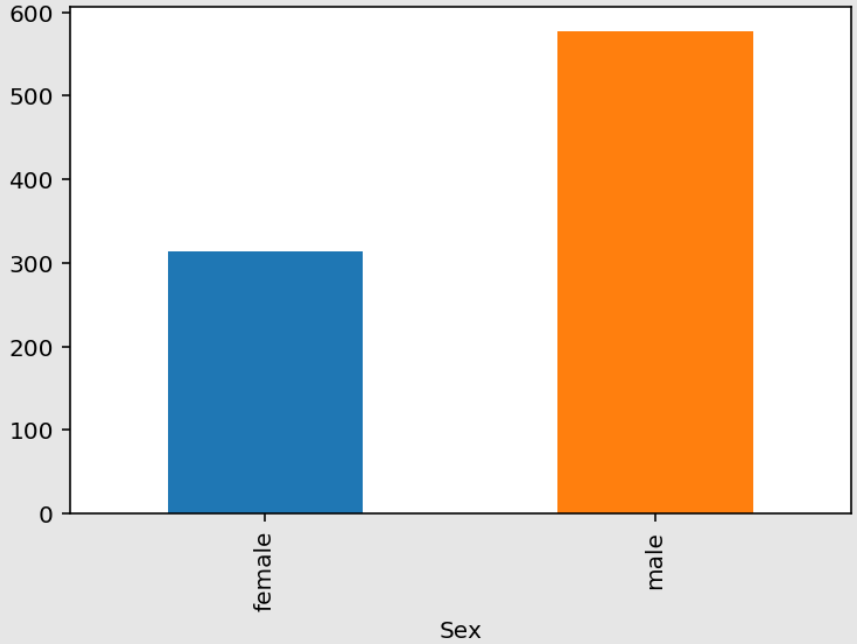
```
ti = pd.read_csv("titanic_sample.csv")
ti.plot(kind = "hist", y = "Age", bins = 16)
```

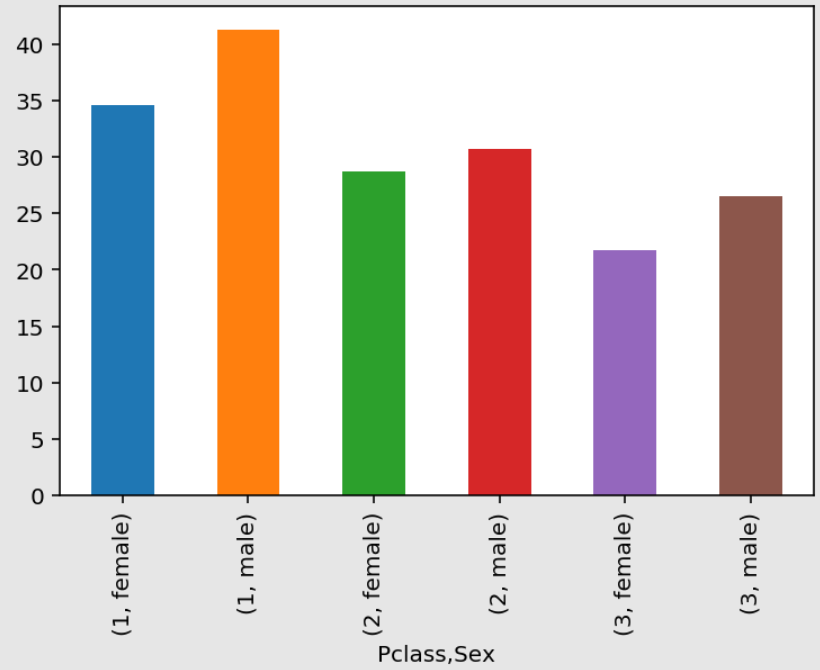


Barplot with group by

Python

```
ti = pd.read_csv("sample_data/titanic_simple.csv")
ti.groupby("Sex")['PassengerId'].count().plot(kind = 'bar’)
ti.groupby(["Pclass","Sex"])["Age"].mean().plot(kind = "bar")
```

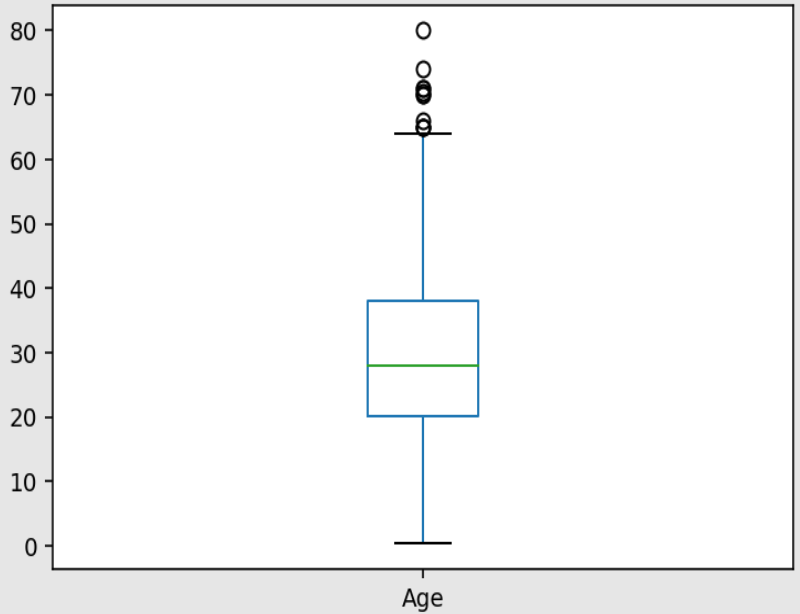




boxplot

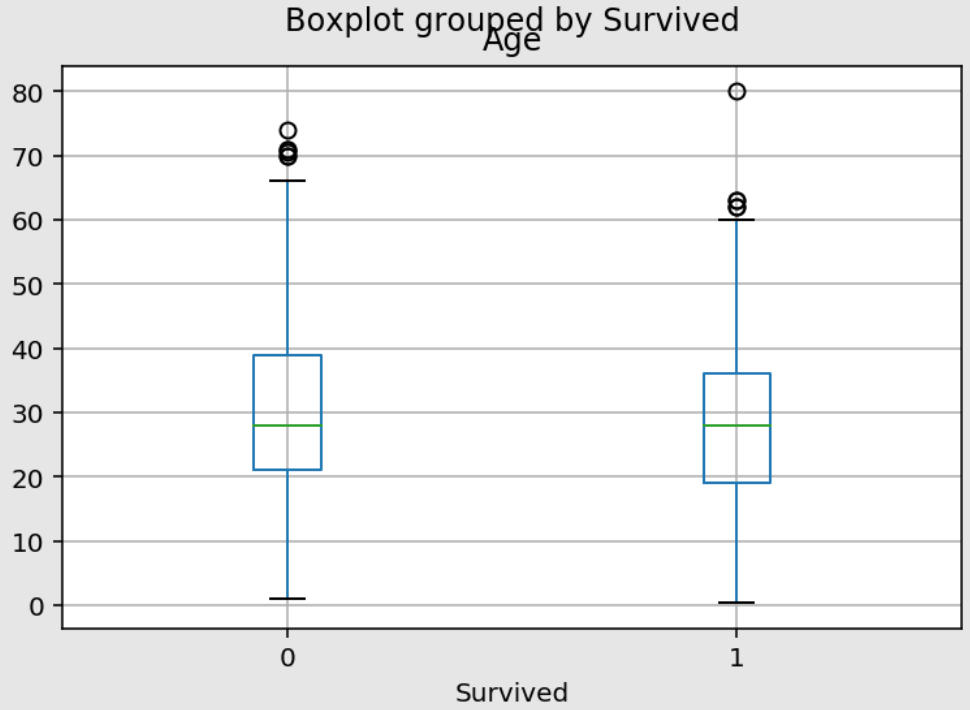
Python

```
ti.plot(kind = "box", y ="Age")
```



Python

```
ti.boxplot("Age", by ="Survived")
```



seaborn package

seaborn 패키지 개요

- ✓ seaborn은 matplotlib 기반으로 다양한 색상과 차트 기능을 추가한 시각화 패키지
- ✓ matplotlib에서 제공하지 않는 통계 차트들 제공

Python

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

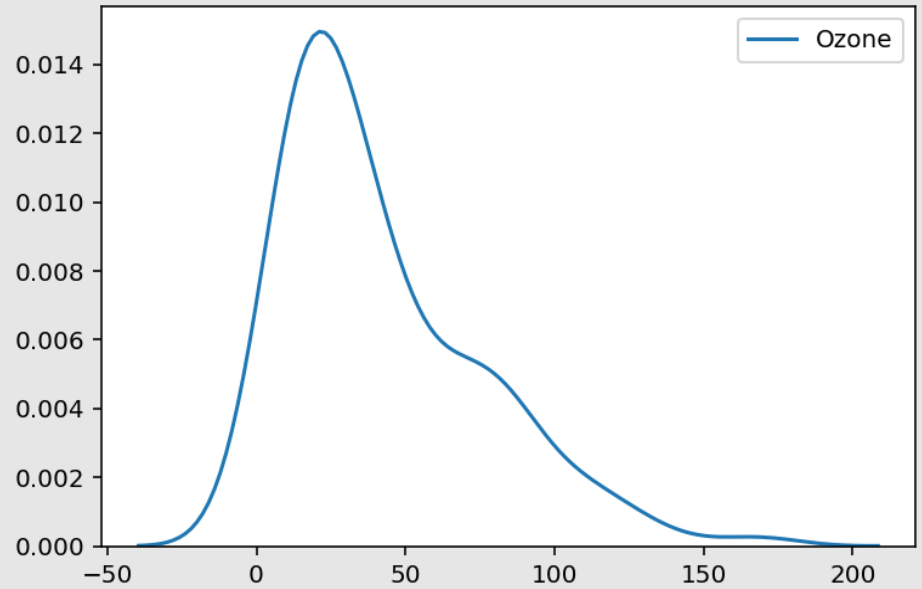
distplot(Density plot & histogram)

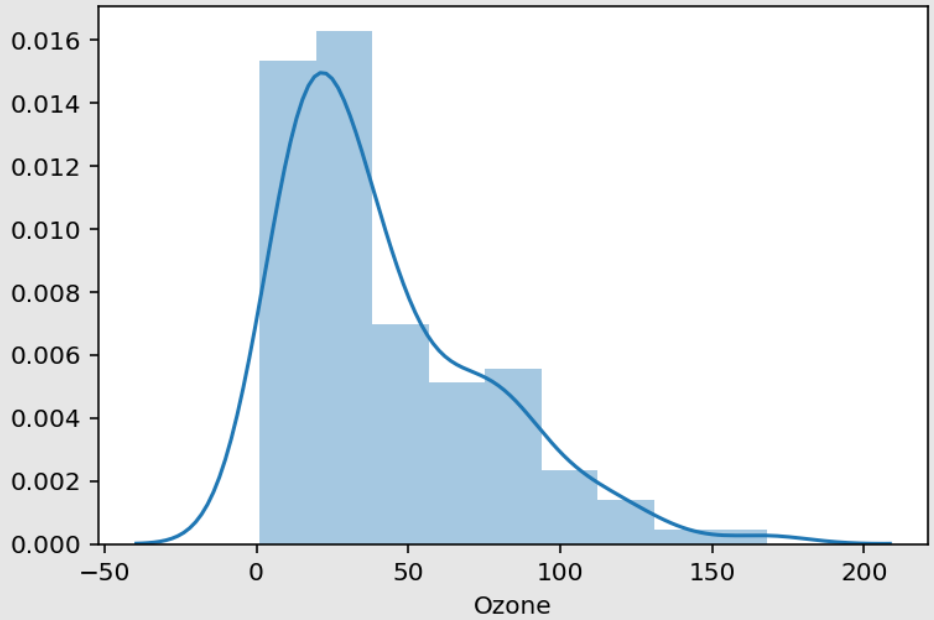
Python

```
# Density plot만 그리기
sns.kdeplot(aq["Ozone"])
plt.show()
```

Python

```
# distplot으로 densityplot과 histogram 함께그리기
sns.distplot(aq[aq["Ozone"].notnull()]["Ozone"])
plt.show()
```

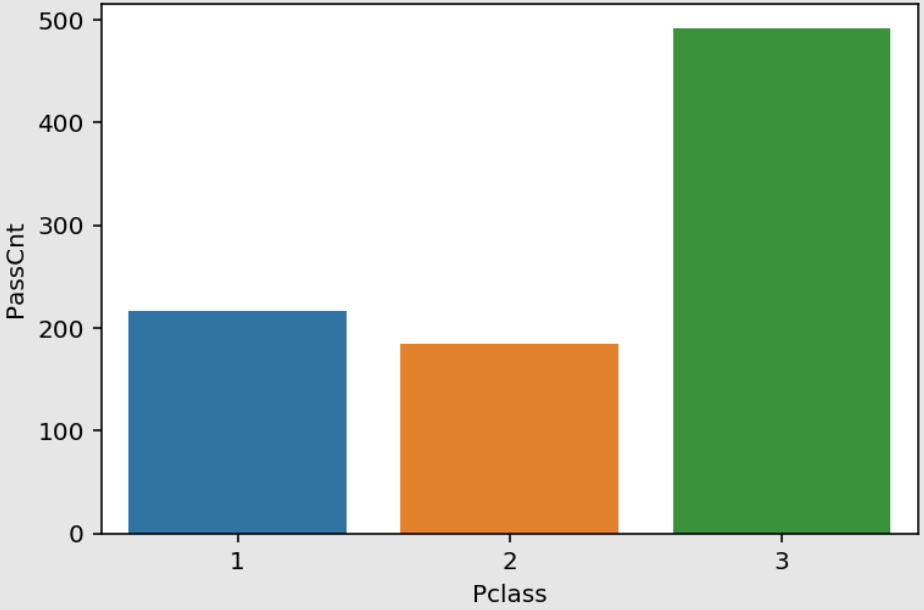




countplot vs. barplot

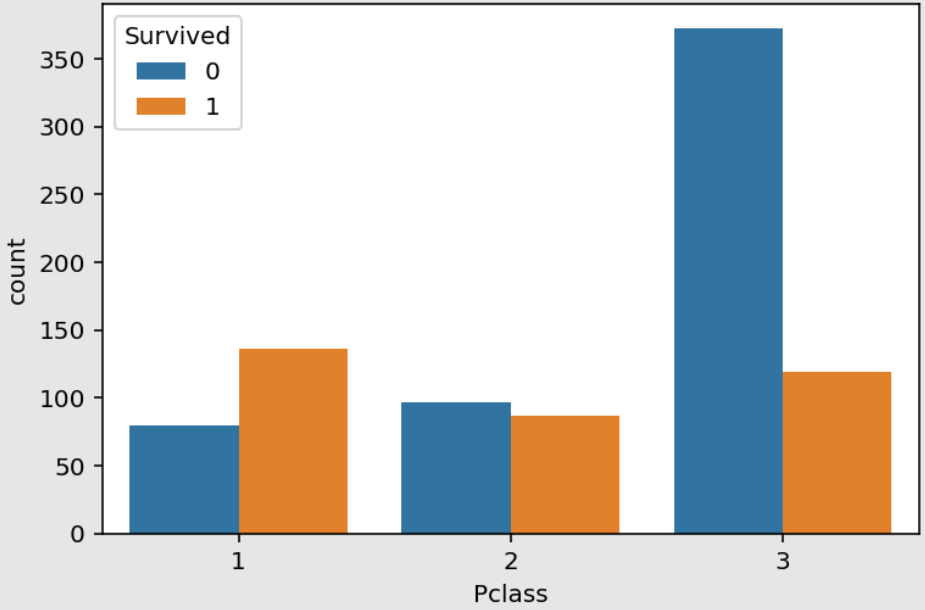
Python

```
ti_2 = ti.groupby("Pclass", as_index =
False)[["PassengerId"]].count()
ti_2.columns.values[1] = "PassCnt"
sns.barplot(x="Pclass", y = "PassCnt",
data=ti_2)
```



Python

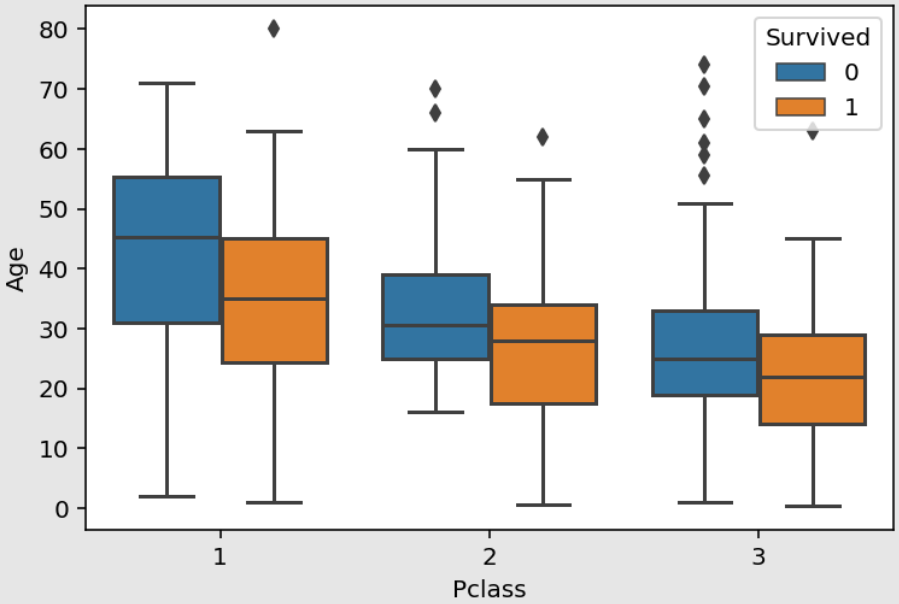
```
sns.countplot(x="Pclass", data=ti
                    , hue = "Survived")
```

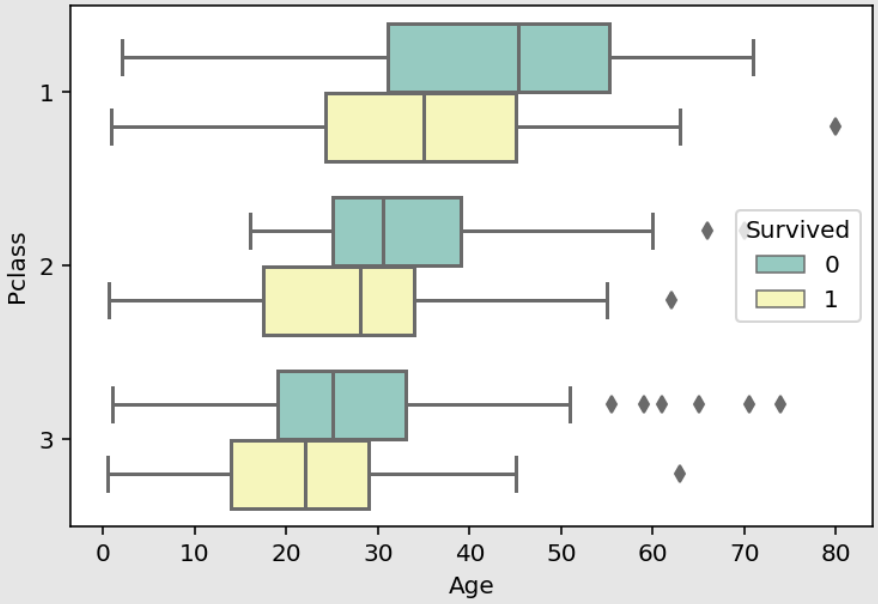


Box plot

Python

```
sns.boxplot(x="Pclass", y="Age", hue="Survived", data=ti)
sns.boxplot(y="Pclass", x="Age", hue="Survived", data=ti, orient = 'h'
                  , palette = "Set3")
```

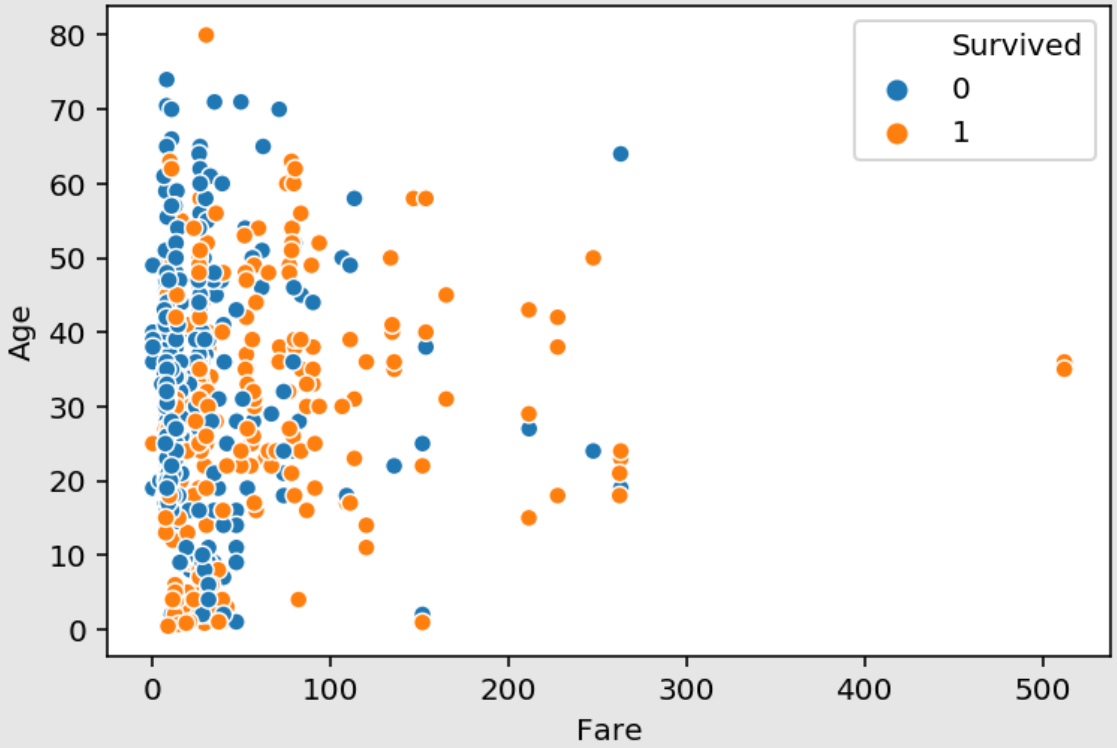




Scatterplot

Python

```
sns.scatterplot(x="Fare", y="Age", hue="Survived", data=ti)
```

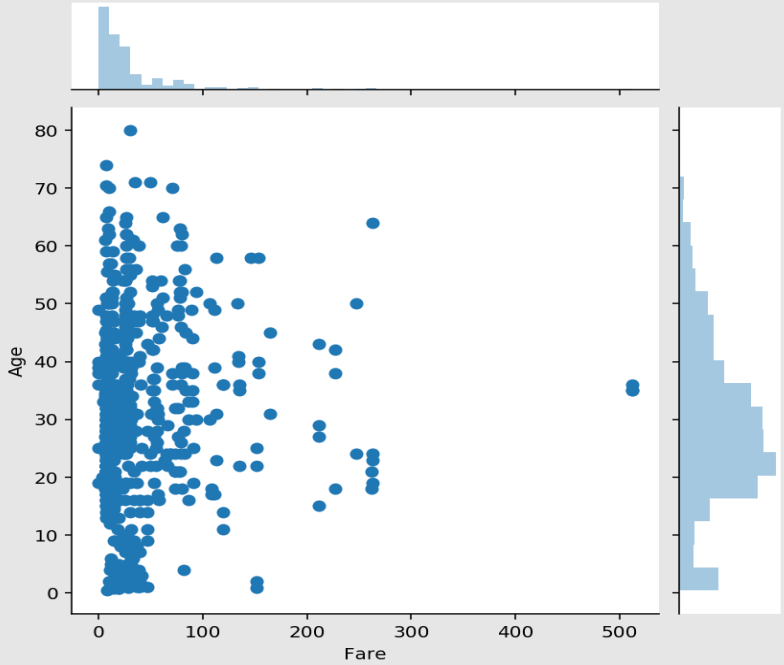


특별한 Chart : jointplot

✓ Histogram과 Scatter가 같이 표현

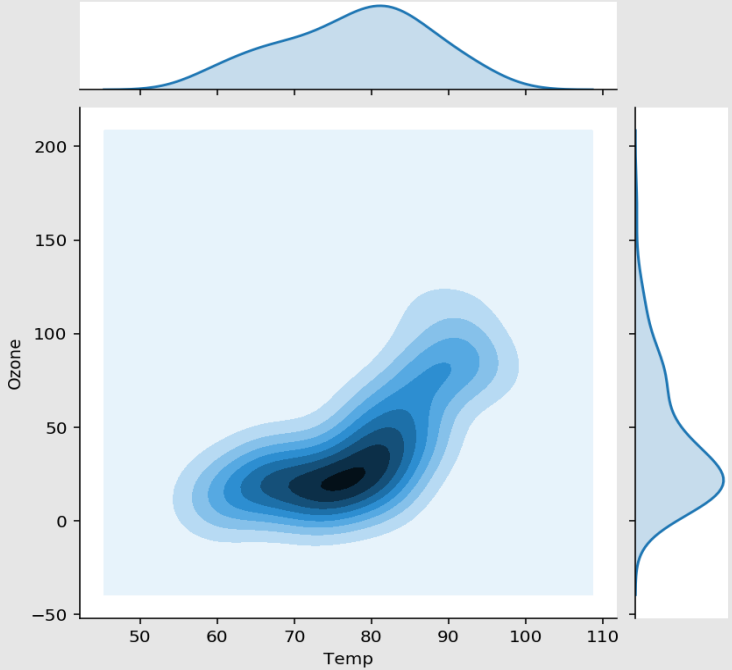
Python

```
sns.jointplot(x="Fare", y="Age", data=ti)
```



Python

```
sns.jointplot(x="Temp", y="Ozone“
        , data=aq, kind = 'kde')
```

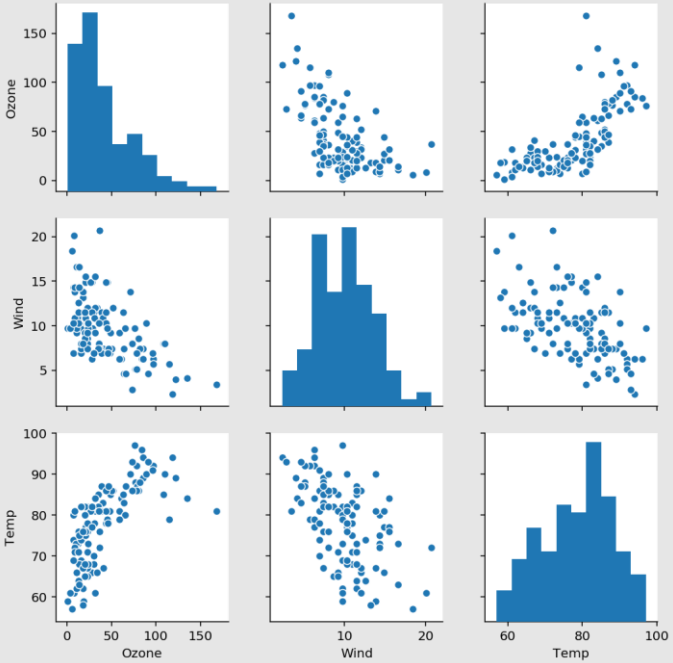


특별한 Chart : pairplot

✓ 상관관계를 살펴볼때.

Python

```
sns.pairplot(aq2
        , vars = ['Ozone','Wind','Temp'])
```



Python

```
sns.pairplot(ti2, vars = ['Age','Fare’]
                        , hue = "Survived")
```