

2023년 가을학기

Plotly



Day

05



데이터시각화 기초 - Plotly

CONTENTS

- A. Plotly는 무엇인가?
- B. Plotly Basic Charts
- C. Plotly 데이터 분석을 위한 차트



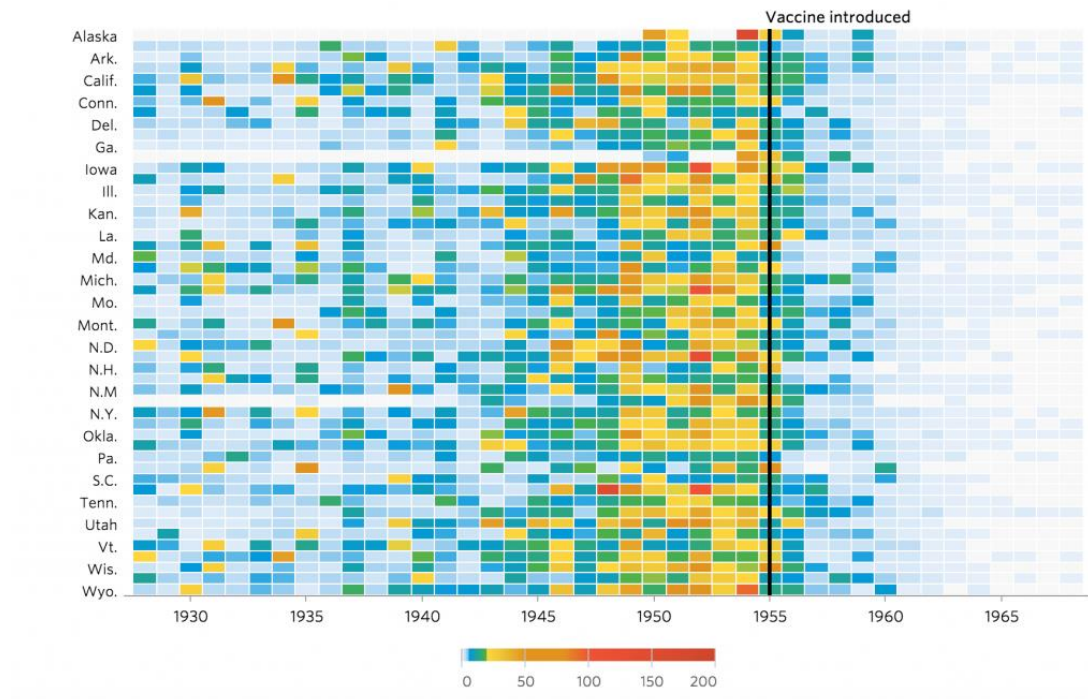
A

Plotly는 무엇인가?

데이터시각화

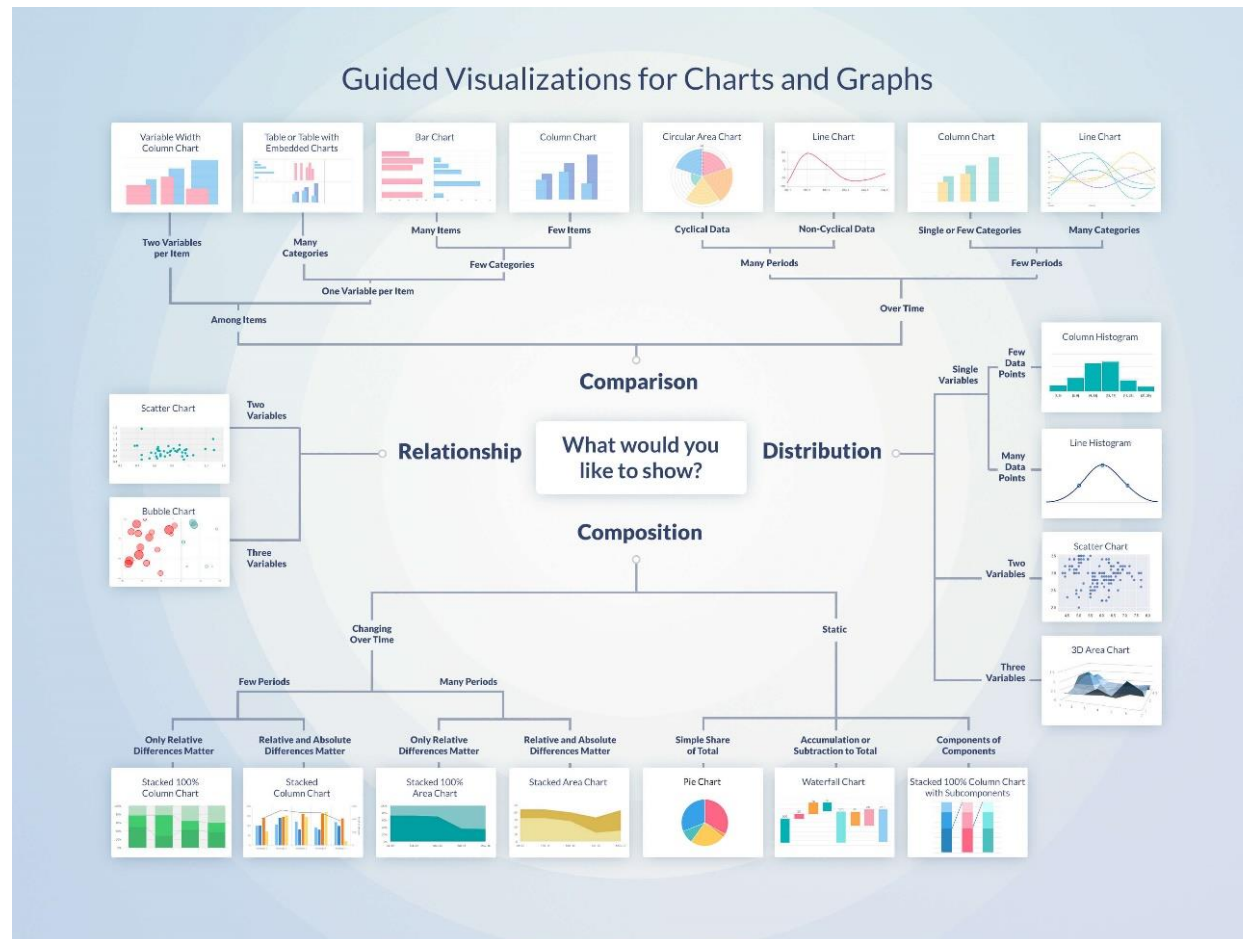
❖ 데이터시각화 무인인가?

- 정보의 시각적 요약을 통해 패턴과 트렌드를 쉽게 식별할 수 있으므로 데이터시각화가 필요합니다.
- Example: Infectious Diseases and The Impact of Vaccines



데이터시각화


❖ Types of data visualization



데이터시각화

❖ 농산물 가격 예측을 위한 데이터시각화 예

농촌진흥청 ABDAS www.BANDICAM.COM

 Agriculture Big Data Analysis System
온라인

MAIN NAVIGATION

- 시설채소 유통 정보 <
- 시설채소 비정형 데이터 정보 <
- 시설채소 빅데이터 정보 분석 <
- 시설채소 빅데이터 예측 시스템 <
- % 생산동향 관리 <
- 데이터수집하기 <

뉴스 수 뉴스 수

기간

결과보기

Copyright © 2014-2016 Almsaeed Studio. All rights reserved. Version 0.1

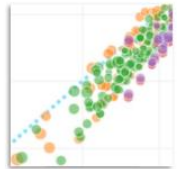
Plotly는 무엇인가?

- ❖ Plotly는 데이터 분석 및 시각화 도구 개발에 참여하는 Montreal 기반 회사입니다.
- ❖ Plotly의 중요 기능 중 일부는 다음과 같습니다.
 - 대화형 그래프를 생성합니다.
 - 그래프는 JSON(JavaScript Object Notation) 데이터 형식으로 저장됩니다.
 - 다른 프로그래밍 언어의 스크립트를 사용하여 읽을 수 있습니다.
 - 그래프는 다양한 래스터 및 벡터 이미지 형식으로 내보낼 수 있습니다.

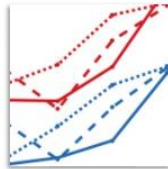
Plotly는 무엇인가?

- ❖ Plotly 파이썬 라이브러리는 40개 이상의 독특한 차트 유형을 지원하는 대화형 오픈 소스 플로팅 라이브러리입니다.

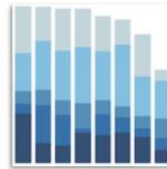
Basic Charts



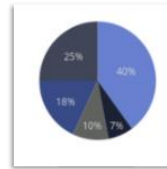
Scatter Plots



Line Charts



Bar Charts



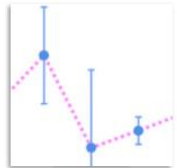
Pie Charts



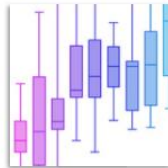
Bubble Charts

[More Basic Charts >](#)

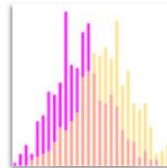
Statistical Charts



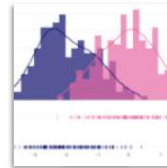
Error Bars



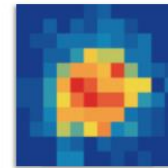
Box Plots



Histograms



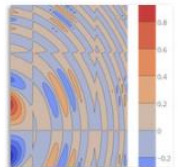
Distplots



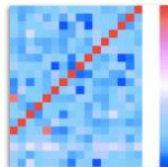
2D Histograms

[More Statistical Charts >](#)

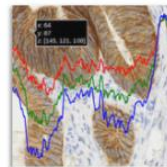
Scientific Charts



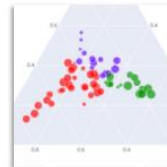
Contour Plots



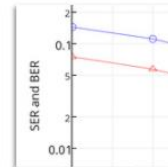
Heatmaps



Imshow



Ternary Plots



Log Plots

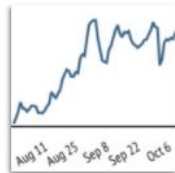
[More Scientific Charts >](#)

Plotly는 무엇인가?

- ❖ Plotly 파이썬 라이브러리는 40개 이상의 독특한 차트 유형을 지원하는 대화형 오픈 소스 플로팅 라이브러리입니다.

Financial Charts

[More Financial Charts >](#)



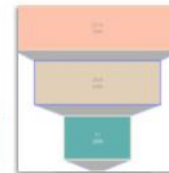
Time Series and Date Axes



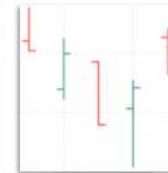
Candlestick Charts



Waterfall Charts



Funnel Chart



OHLC Charts

Maps

[More Maps >](#)



Mapbox Choropleth Maps



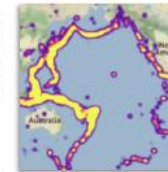
Lines on Mapbox



Filled Area on Maps



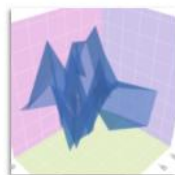
Bubble Maps



Mapbox Density Heatmap

3D Charts

[More 3D Charts >](#)



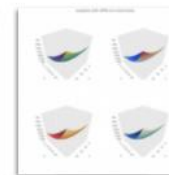
3D Axes



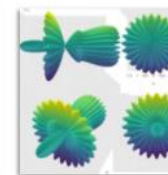
3D Scatter Plots



3D Surface Plots



3D Subplots



3D Camera Controls

Plotly를 배우는 이유는 무엇인가?

❖ 특징



\$30M

Funding toward advancing open-source data visualization and Python & R user interfaces.



33K

Total GitHub Stars for Dash, Plotly.py, & Plotly.js (top 1% of GitHub's most popular software). [Throw us a star!](#)



4M

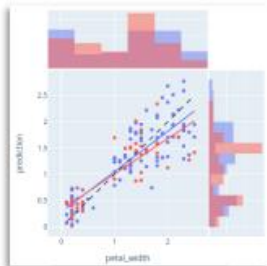
Monthly downloads of Plotly open-source graphing libraries.



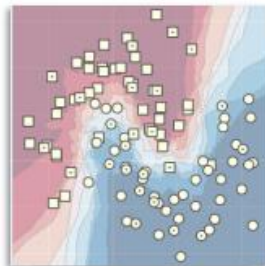
50M+

Plotly users worldwide are making data science and AI accessible to everyone.

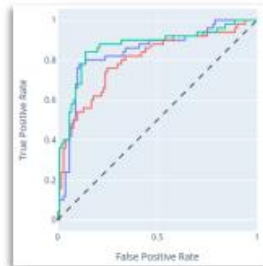
❖ 최근 AI와 기계 학습(Machine Learning)에 활용되고 있습니다.



ML Regression



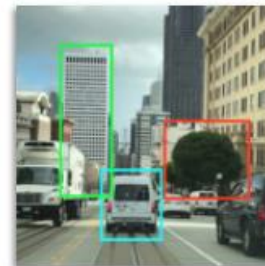
kNN Classification



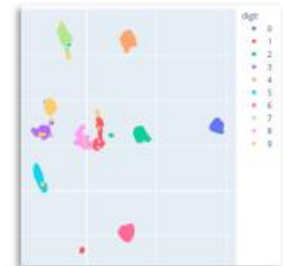
ROC and PR Curves



PCA Visualization



AI/ML Apps with Dash



t-SNE and UMAP projections

Plotly를 배우는 이유는 무엇인가?

❖ Plotly는 여러 대기업에서 활용되고 있습니다.



Plotly 설치

- ❖ Jupyter Notebook을 실행합니다.
- ❖ 명령 프롬프트(CMD)를 실행하고 다음 명령어를 사용합니다.
 - `pip install plotly`
- ❖ 다른 라이브러리 또한 설치해야 합니다.
 - Numpy, pandas
- ❖ plotly 라이브러리 버전을 확인합니다.

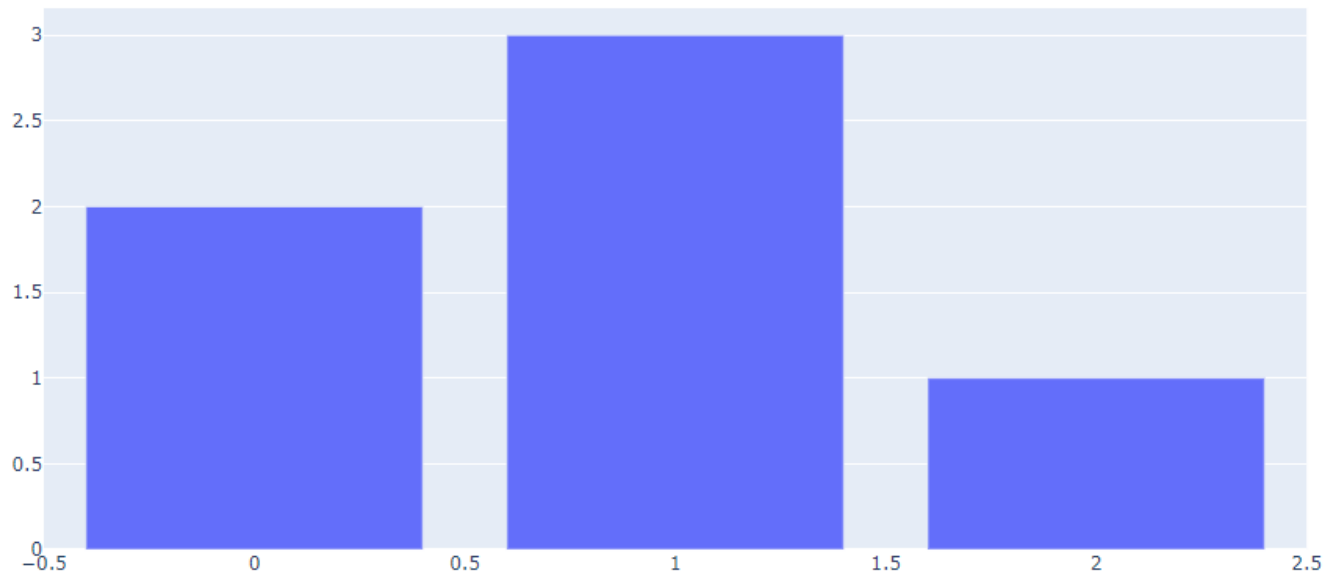
```
import plotly  
print(plotly.__version__)
```

```
//Importing plotly library  
//Printing plotly library version
```

Plotly 설치

❖ 첫 번째 데이터 시각화를 수행하세요.

```
import plotly.graph_objects as go  
  
fig = go.Figure(data=go.Bar(y=[2, 3, 1]))  
fig.show()
```





B

Basic Charts

Bar Chart

❖ go.Bar() function

■ 예제 소스코드

```
import plotly.graph_objs as go

langs = ['C', 'C++', 'Java', 'Python', 'PHP']
students = [23, 17, 35, 29, 12]

data = [go.Bar(
    x = langs,
    y = students
)]

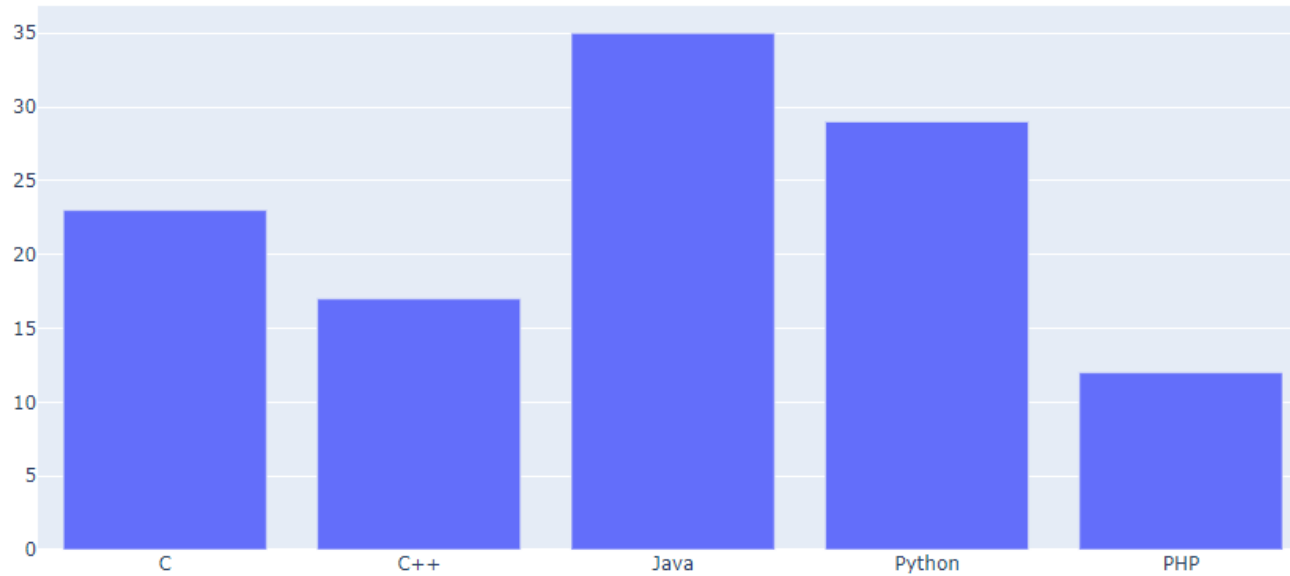
fig = go.Figure(data = data)

fig.show()
```

Bar Chart

❖ go.Bar() function

■ 결과



Bar Chart

❖ Grouped bar chart 만들기

■ 변수 선언

```
import plotly.graph_objs as go  
  
branches = ['CSE', 'Mech', 'Electronics']  
  
fy = [23, 17, 35]  
sy = [20, 23, 30]  
ty = [30, 20, 15]
```

Bar Chart

❖ 그룹화 해야 하는 데이터

- `name` property sets its legend entry

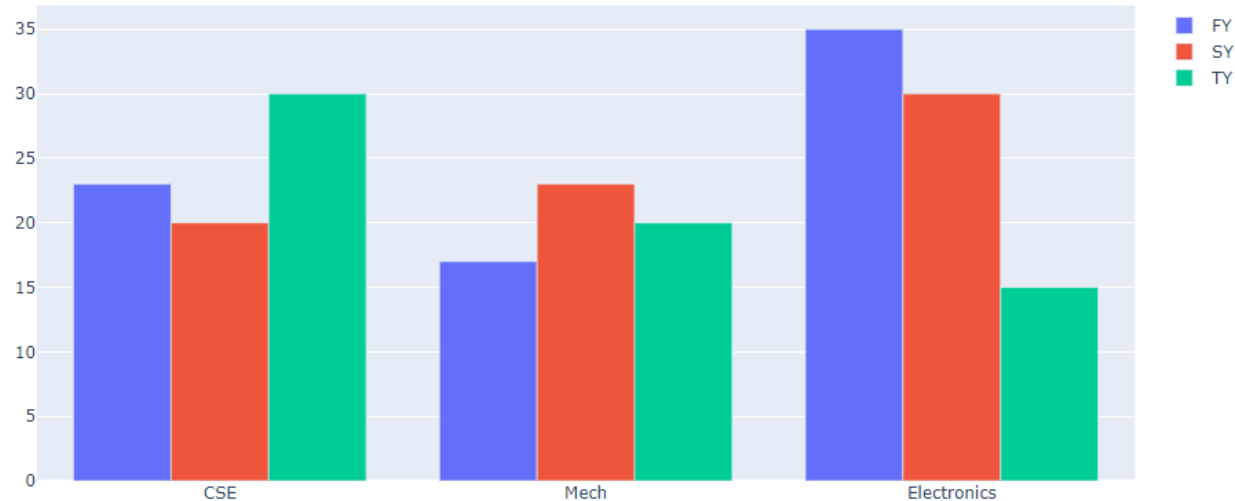
```
trace1 = go.Bar(  
    x = branches,  
    y = fy,  
    name = 'FY'  
)  
trace2 = go.Bar(  
    x = branches,  
    y = sy,  
    name = 'SY'  
)  
trace3 = go.Bar(  
    x = branches,  
    y = ty,  
    name = 'TY'  
)
```

Bar Chart

- ❖ 그룹화 된 막대 차트를 표시하려면 레이아웃 개체의 막대 모드 속성을 그룹으로 설정해야 합니다.
 - 레이아웃 생성 및 결과 출력

```
data = [trace1, trace2, trace3]  
layout = go.Layout(barmode = 'group', title='Departments')  
fig = go.Figure(data = data, layout = layout)  
fig.show()
```

Departments

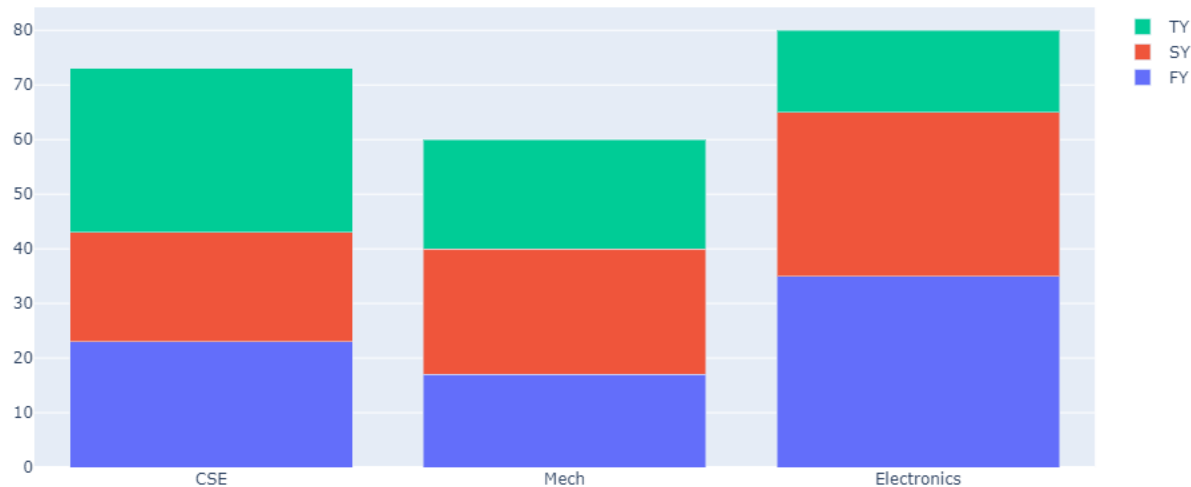


Bar Chart

- ❖ Stack 된 막대 차트를 표시하려면 `stack` 속성을 사용해야 함
 - 레이아웃 생성 및 결과 출력

```
data = [trace1, trace2, trace3]
layout = go.Layout(barmode = 'stack', title='Departments')
fig = go.Figure(data = data, layout = layout)
fig.show()
```

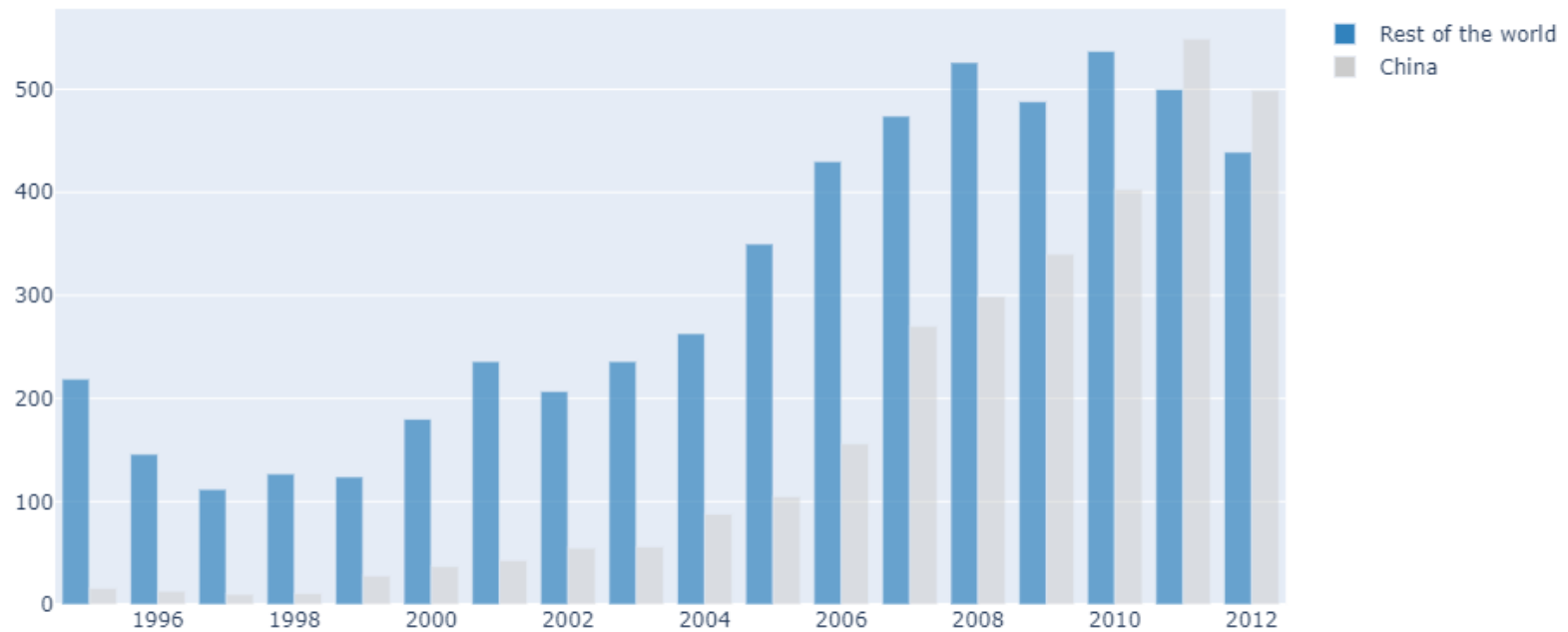
Departments



Bar Chart

❖ Customized colors

Export of Plastic Scrap



Bar Chart

❖ Customized colors

■ 변수 선언

```
import plotly.graph_objs as go
```

```
years = [1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003,  
2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012]
```

```
rest = [219, 146, 112, 127, 124, 180, 236, 207, 236, 263, 350, 430,  
474, 526, 488, 537, 500, 439]
```

```
china = [16, 13, 10, 11, 28, 37, 43, 55, 56, 88, 105, 156, 270, 299,  
340, 403, 549, 499]
```

Bar Chart

❖ Customized colors

- Use marker property for creating customized colors

```
trace1 = go.Bar(  
    x = years,  
    y = rest,  
    name = 'Rest of the World',  
    marker = dict(  
        color = 'rgb(49,130,189)',  
        opacity = 0.7  
    )  
)
```

```
trace2 = go.Bar(  
    x = years,  
    y = china,  
    name = 'China',  
    marker = dict(  
        color = 'rgb(204,204,204)',  
        opacity = 0.5  
    )  
)
```

- Full list of properties
 - https://plotly.com/python-api-reference/generated/plotly.graph_objects.Bar.html

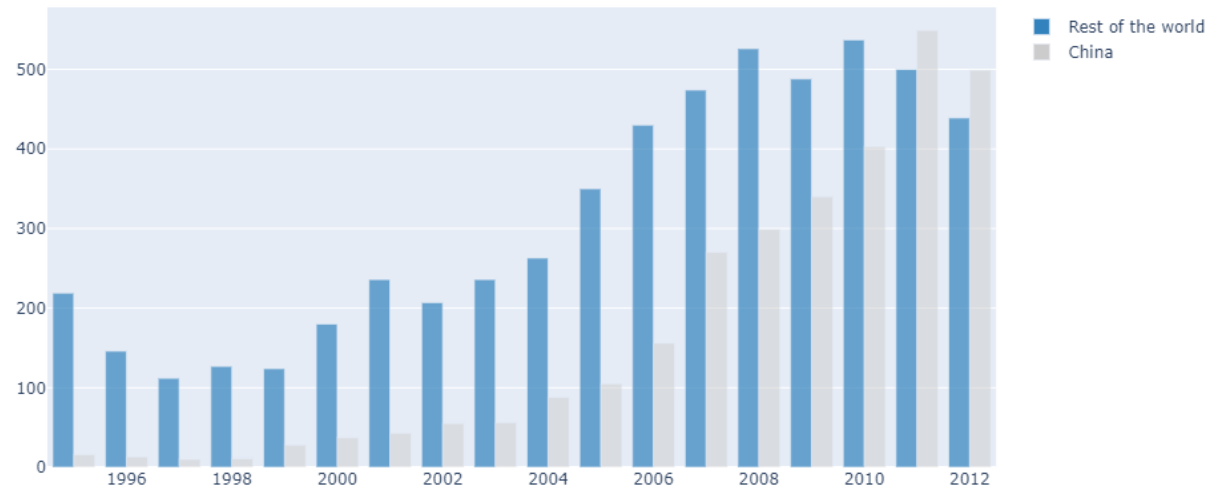
Bar Chart

❖ Customized colors

■ Layout construction

```
data = [trace1, trace2]  
layout = go.Layout(barmode = 'group', title='Export of Plastic Scrap')  
fig = go.Figure(data = data, layout = layout)  
fig.show()
```

Export of Plastic Scrap



Bar Chart

❖ Label annotation

- 변수 선언

```
import plotly.graph_objs as go
```

```
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',  
          'Oct', 'Nov', 'Dec']
```

```
primary_sales = [20, 14, 25, 16, 18, 22, 19, 15, 12, 16, 14, 17]
```

```
secondary_sales = [19, 14, 22, 14, 16, 19, 15, 14, 10, 12, 12, 16]
```

Bar Chart

❖ Label annotation

- Include 'text' and 'textposition' properties

```
trace1 = go.Bar(  
    x = months,  
    y = primary_sales,  
    name = 'Primary Product',  
    marker = dict(  
        color = 'rgb(49,130,189)',  
        opacity = 0.7  
    ),  
    text = primary_sales,  
    textposition = 'auto',  
)
```

```
trace2 = go.Bar(  
    x = months,  
    y = secondary_sales,  
    name = 'Secondary Product',  
    marker = dict(  
        color = 'rgb(204,204,204)',  
        opacity = 0.5  
    ),  
    text = secondary_sales,  
    textposition = 'auto',  
)
```

- Full list of properties
 - https://plotly.com/python-api-reference/generated/plotly.graph_objects.Bar.html

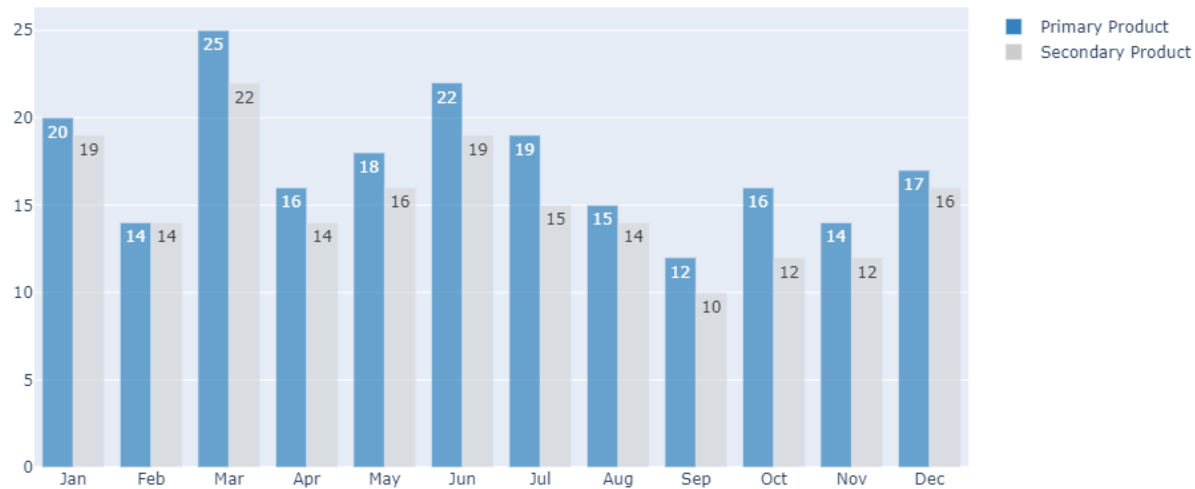
Bar Chart

❖ Label annotation

■ Layout construction

```
data = [trace1, trace2]  
layout = go.Layout(barmode = 'group', title='2020 Sales Report')  
fig = go.Figure(data = data, layout = layout)  
fig.show()
```

2020 Sales Report



Pie Chart

❖ go.Pie() function

- 두개의 필수 arguments는 labels과 values 입니다.

```
import plotly.graph_objs as go

langs = ['C', 'C++', 'Java', 'Python', 'PHP']
students = [23, 17, 35, 29, 12]

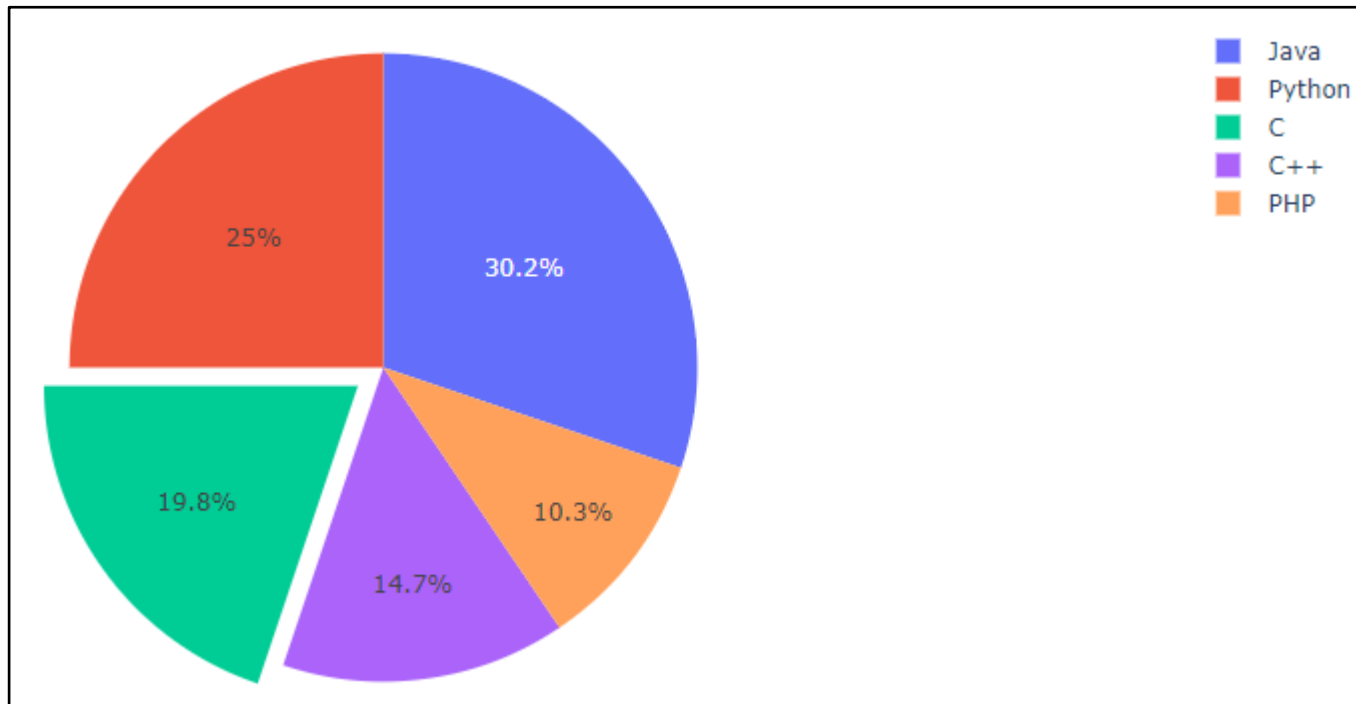
data = [go.Pie(
    labels = langs,
    values = students,
    pull = [0.1, 0, 0, 0, 0]
)]

fig = go.Figure(data=data)
fig.show()
```

Pie Chart

❖ go.Pie() function

■ 결과

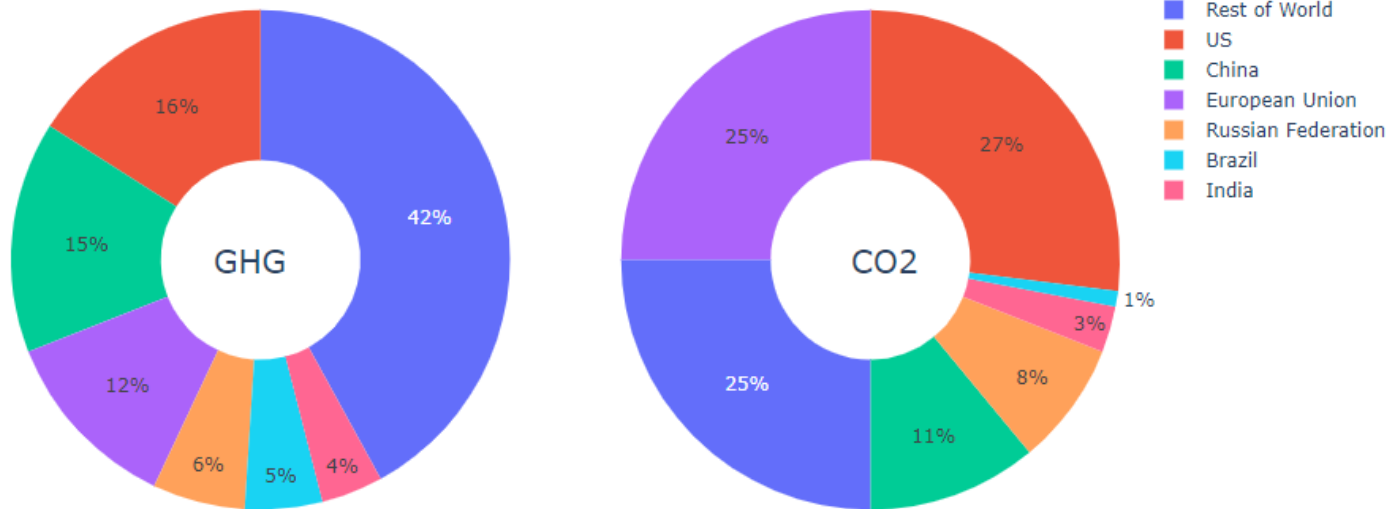


Pie Chart

❖ Subplot

■ Examples

Global Emissions 1990-2011



Pie Chart

❖ subplot() 함수 예

■ 변수 선언

```
import plotly.graph_objects as go
from plotly.subplots import make_subplots

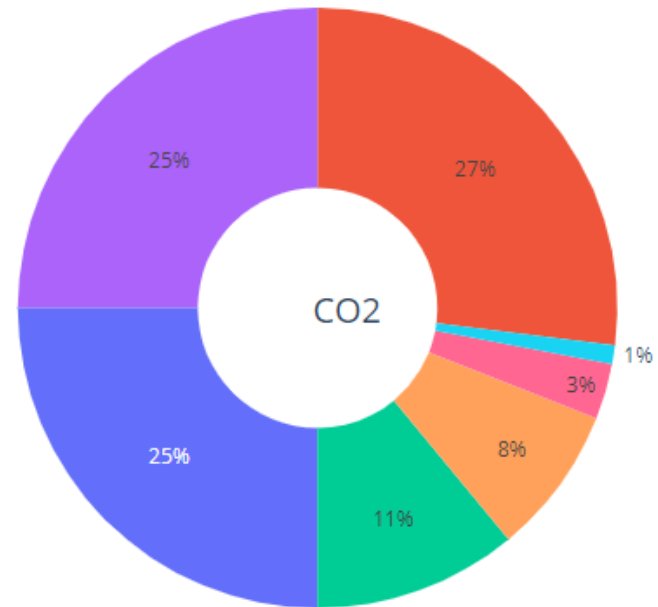
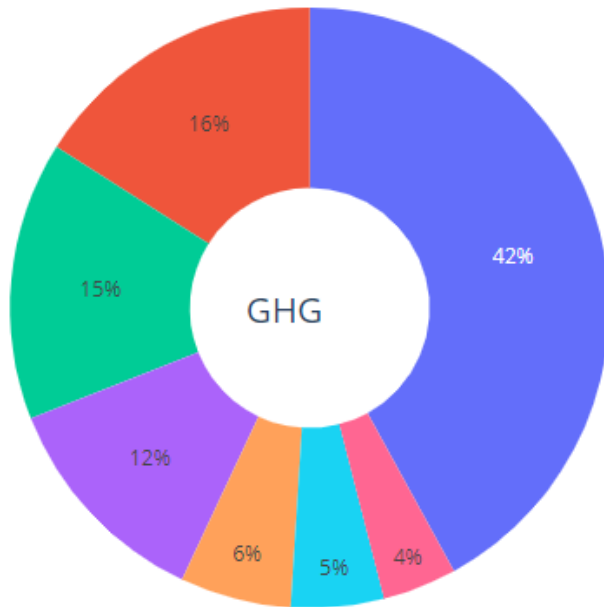
countries = ["US", "China", "European Union", "Russian Federation", "Brazil", "India", "Rest of World"]

ghg = [16, 15, 12, 6, 5, 4, 42]
co2 = [27, 11, 25, 8, 1, 3, 25]
```

Pie Chart

❖ Create subplots: use 'domain' type for Pie subplot

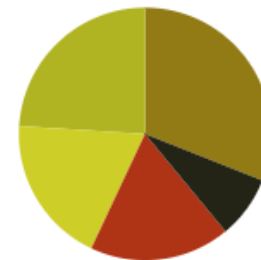
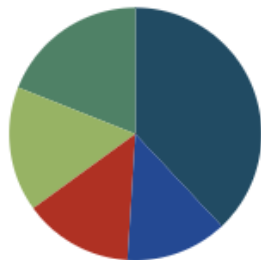
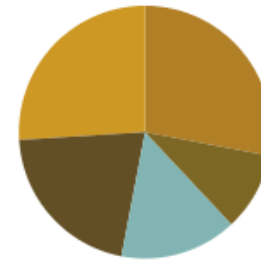
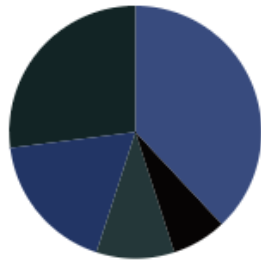
```
fig = make_subplots(rows=1, cols=2, specs=[[{'type':'domain'}, {'type':'domain'}]])
```



Pie Chart

❖ Create subplots: use 'domain' type for Pie subplot

```
specs = [[{'type':'domain'}, {'type':'domain'}], [{"type':'domain'}, {'type':'domain'}]]  
fig = make_subplots(rows=2, cols=2, specs=specs)
```



Pie Chart

❖ Creating traces

- 두개의 필수 arguments는 labels과 values 입니다.

```
fig.add_trace(go.Pie(  
    labels=countries,  
    values=ghg,  
    name="GHG Emissions"),  
    row=1, col=1)
```

```
fig.add_trace(go.Pie(  
    labels=countries,  
    values=co2,  
    name="CO2 Emissions"),  
    row = 1, col = 2)
```

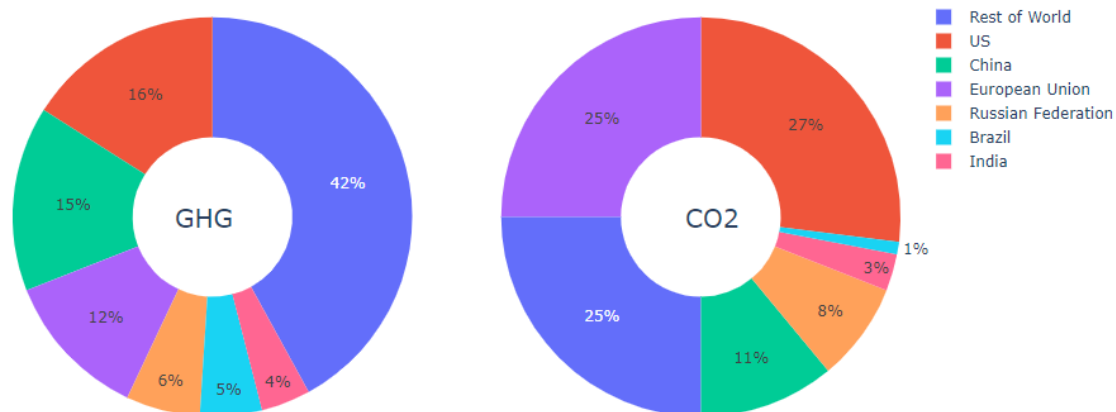
Pie Chart

❖ Creating layout

```
fig.update_traces(hole=.4, hoverinfo="label+percent+name")

fig.update_layout(
    title_text="Global Emissions 1990-2011",
    annotations=[dict(text='GHG', x=0.18, y=0.5, font_size=20, showarrow=False),
                  dict(text='CO2', x=0.82, y=0.5, font_size=20, showarrow=False)]
fig.show()
```

Global Emissions 1990-2011



Scatter Plot

❖ go.Scatter() function

- 난수 생성

```
import plotly.graph_objs as go  
import numpy as np
```

```
N = 100
```

```
x_vals = np.linspace(0, 1, N)
```

```
y1 = np.random.randn(N) + 5
```

```
y2 = np.random.randn(N)
```

```
y3 = np.random.randn(N) - 5
```

Scatter Plot

❖ `go.Scatter()` function

- 모드 속성은 데이터 포인트의 모양을 결정합니다.

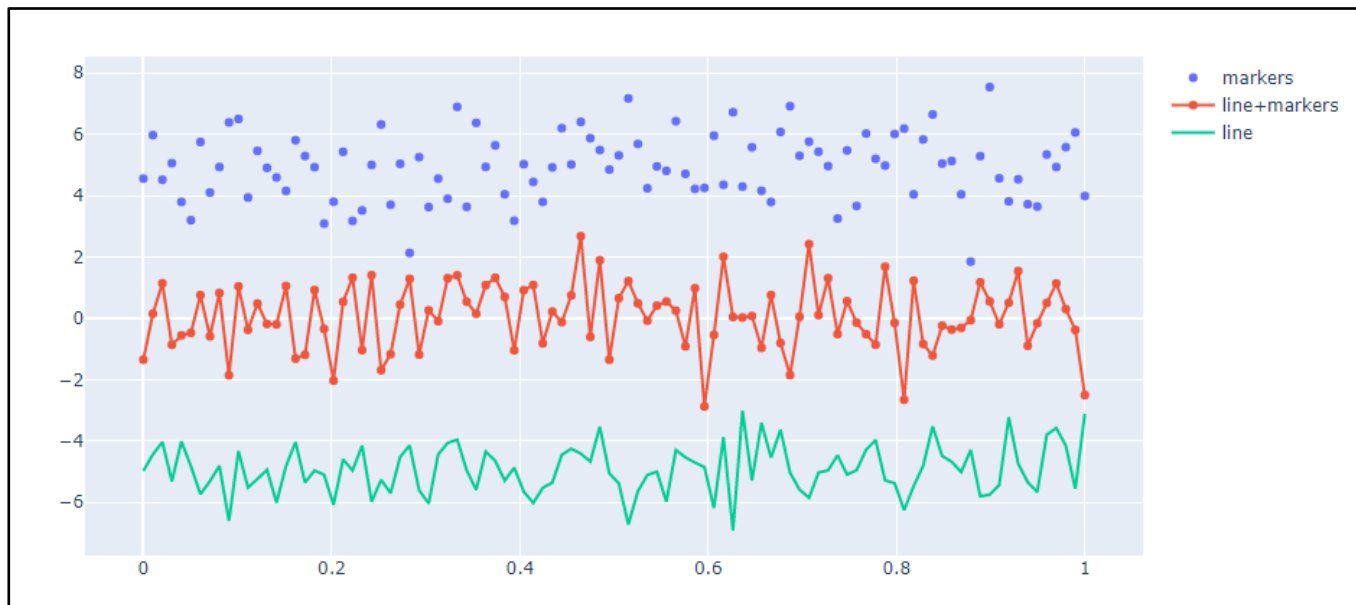
```
trace0 = go.Scatter(  
    x = x_vals,  
    y = y1,  
    mode = 'markers',  
    name = 'markers'  
)  
trace1 = go.Scatter(  
    x = x_vals,  
    y = y2,  
    mode = 'lines+markers',  
    name = 'line+markers'  
)  
trace2 = go.Scatter(  
    x = x_vals,  
    y = y3,  
    mode = 'lines',  
    name = 'line'  
)
```

Scatter Plot

❖ go.Scatter() function

- 데이터 결합 및 결과 출력

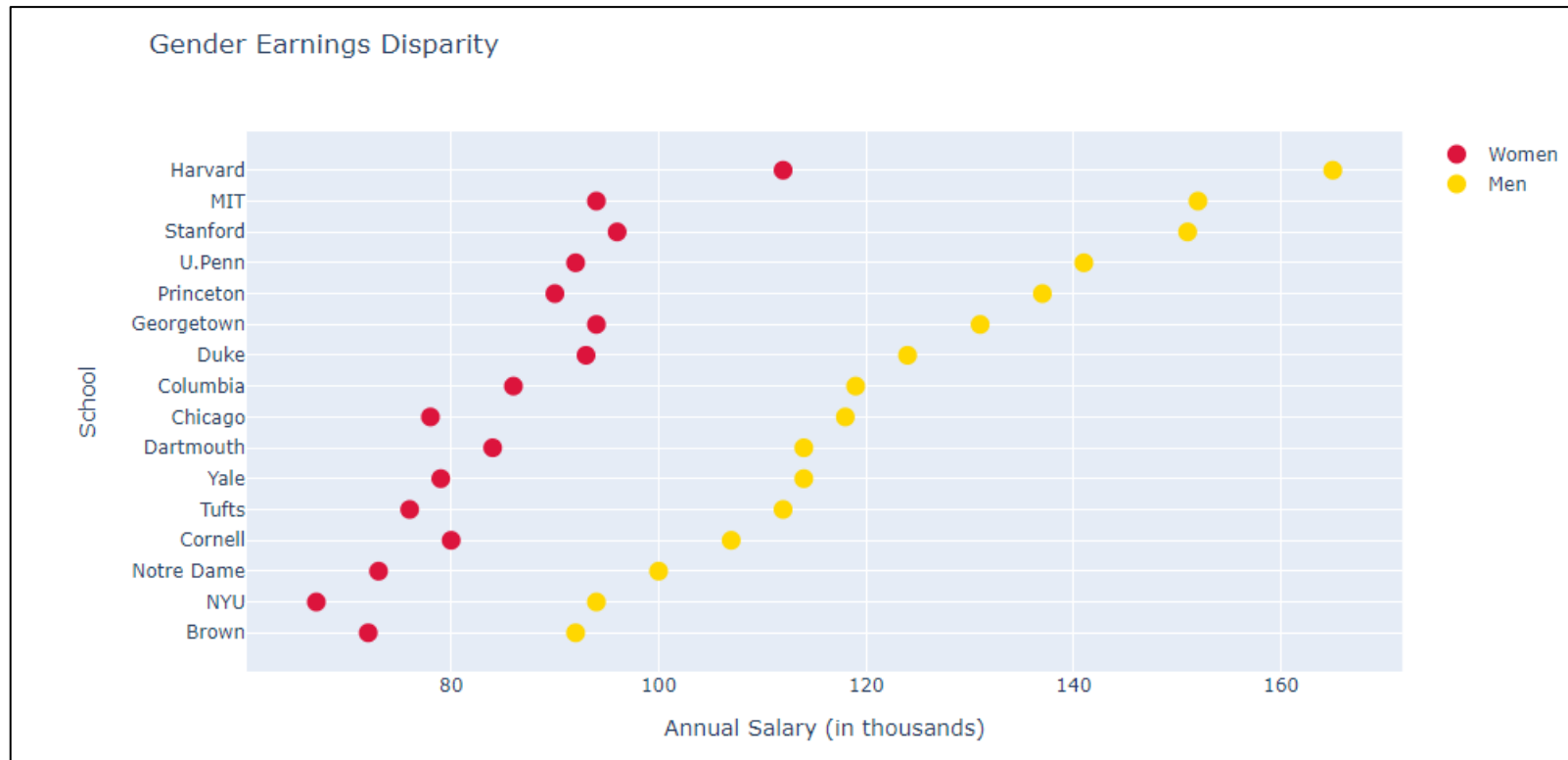
```
data = [trace0, trace1, trace2]  
fig = go.Figure(data = data)  
fig.show()
```



Scatter Plot

❖ `go.Scatter()` function

■ Example



Dot Plots

❖ `go.Scatter`를 사용하여 dot plots를 만들 수도 있습니다.

- 변수 선언

```
import plotly.graph_objects as go
```

```
schools = ["Brown", "NYU", "Notre Dame", "Cornell", "Tufts", "Yale",  
           "Dartmouth", "Chicago", "Columbia", "Duke", "Georgetown",  
           "Princeton", "U.Penn", "Stanford", "MIT", "Harvard"]
```


Dot Plots

❖ `go.Scatter`를 사용하여 dot plots를 만들 수도 있습니다.

■ 데이터 준비

```
trace1 = go.Scatter(  
    x=[72, 67, 73, 80, 76, 79, 84, 78, 86, 93, 94, 90, 92, 96, 94, 112],  
    y=schools,  
    marker=dict(color="crimson", size=12),  
    mode="markers",  
    name="Women",  
)  
  
trace2 = go.Scatter(  
    x=[92, 94, 100, 107, 112, 114, 114, 118, 119, 124, 131, 137, 141, 151, 152, 165],  
    y=schools,  
    marker=dict(color="gold", size=12),  
    mode="markers",  
    name="Men",  
)
```

Dot Plots

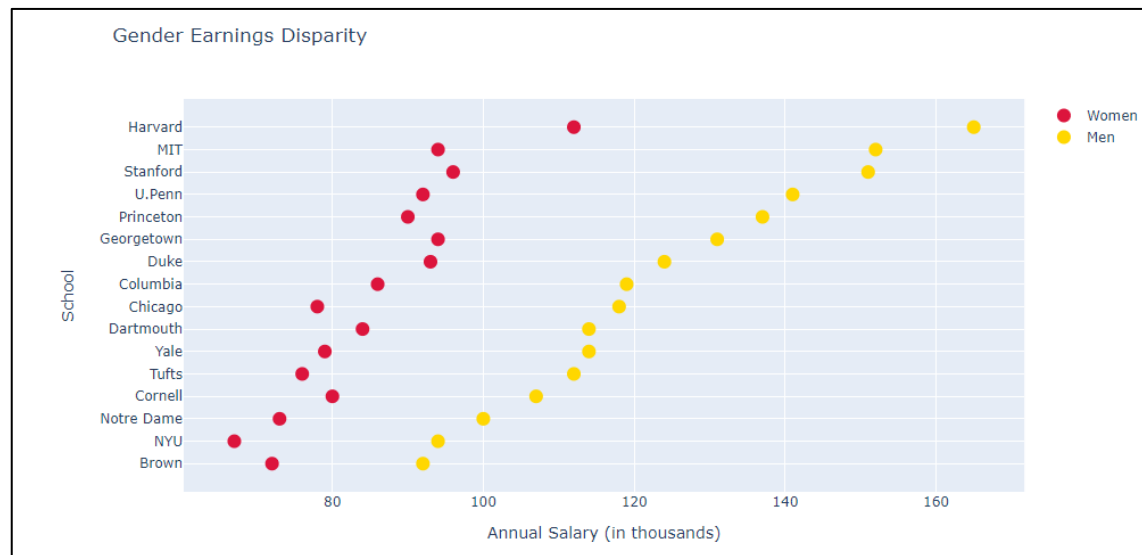
❖ `go.Scatter`를 사용하여 dot plots를 만들 수도 있습니다.

- 레이아웃 설정 및 결과 출력

```
data = [trace1, trace2]

layout = go.Layout(title="Gender Earnings Disparity",
                   xaxis_title="Annual Salary (in thousands)",
                   yaxis_title="School")
fig = go.Figure(data=data, layout = layout)

fig.show()
```



C

데이터 분석을 위한 차트



Histogram

- ❖ Histogram은 수치 데이터의 분포를 표시하는 데 사용됩니다.
- ❖ `go.Histogram`은 histogram을 만드는 데 사용됩니다.

```
import plotly.graph_objects as go
import numpy as np

np.random.seed(1)

x = np.random.randn(500)

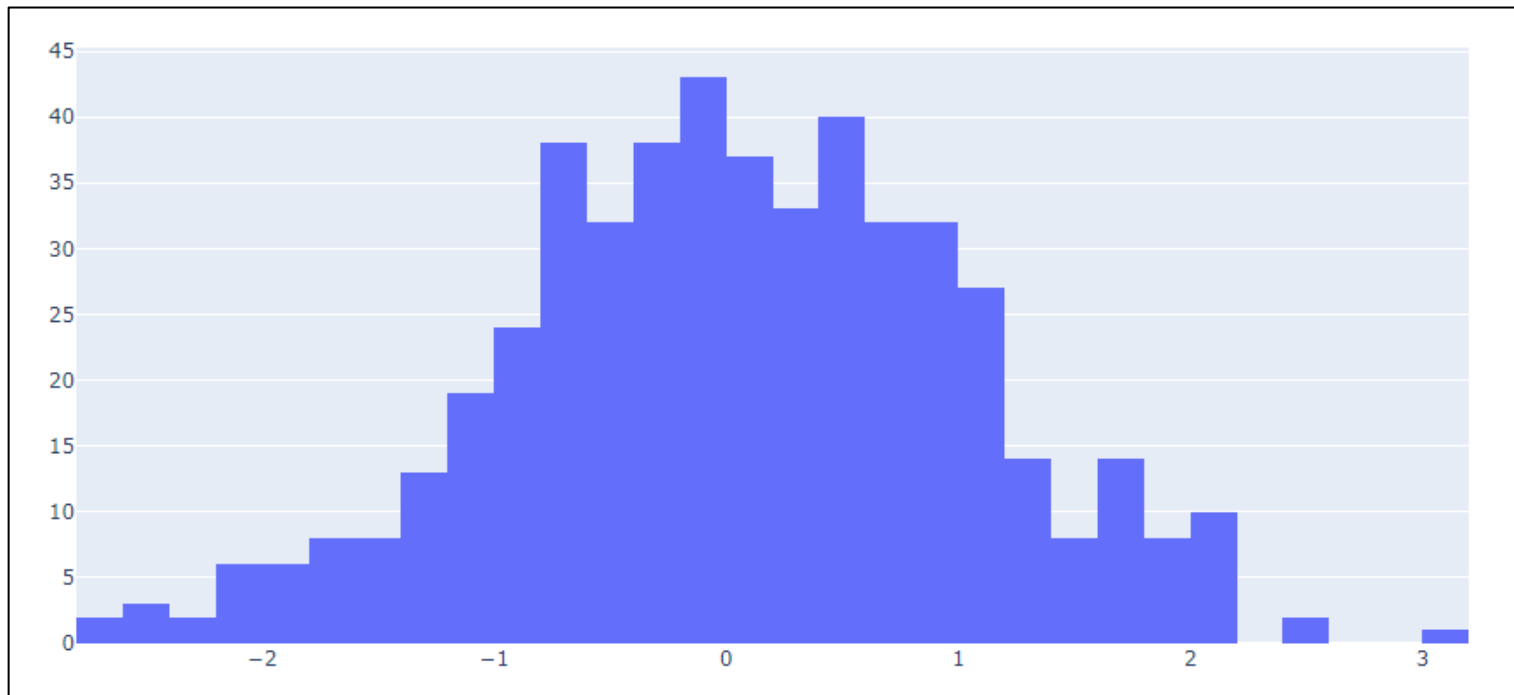
data = [go.Histogram(
    x = x
)]

fig = go.Figure(data)
fig.show()
```

Histogram

❖ `go.Histogram`은 histogram을 만드는 데 사용됩니다.

- 결과 출력



Histogram

❖ Overlaid histogram

```
import plotly.graph_objects as go

import numpy as np

x0 = np.random.randn(500)
x1 = np.random.randn(500) + 1

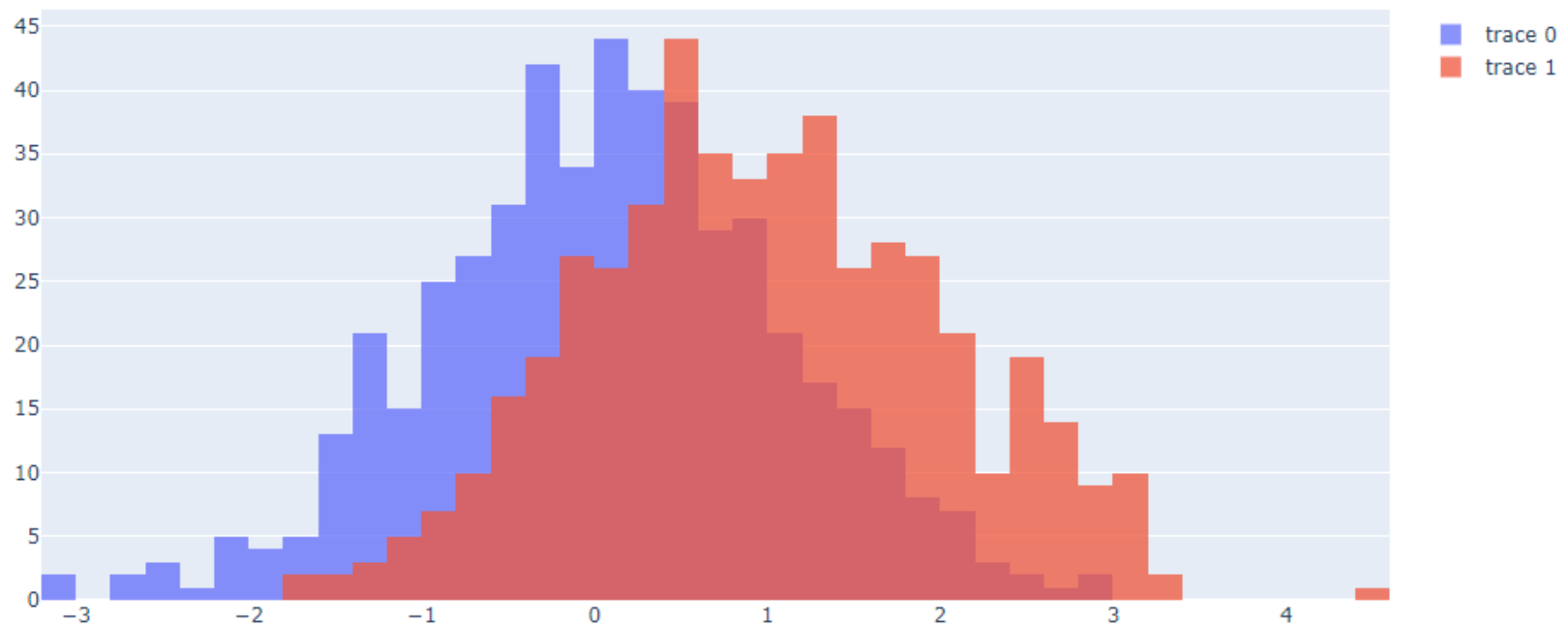
fig = go.Figure()
fig.add_trace(go.Histogram(x=x0))
fig.add_trace(go.Histogram(x=x1))

fig.update_layout(barmode='overlay')
fig.update_traces(opacity=0.75)
fig.show()
```

Histogram

❖ Overlaid histogram

■ Result



Pandas를 이용한 데이터 시각화

❖ Pandas를 이용한 Plotly histogram 제작

```
import plotly.graph_objects as go
import pandas as pd

movies_df = pd.read_csv("D:/IMDB-Movie-Data.csv", index_col="Title")

x_values = movies_df['Rating']

mydata = go.Histogram(x=x_values)

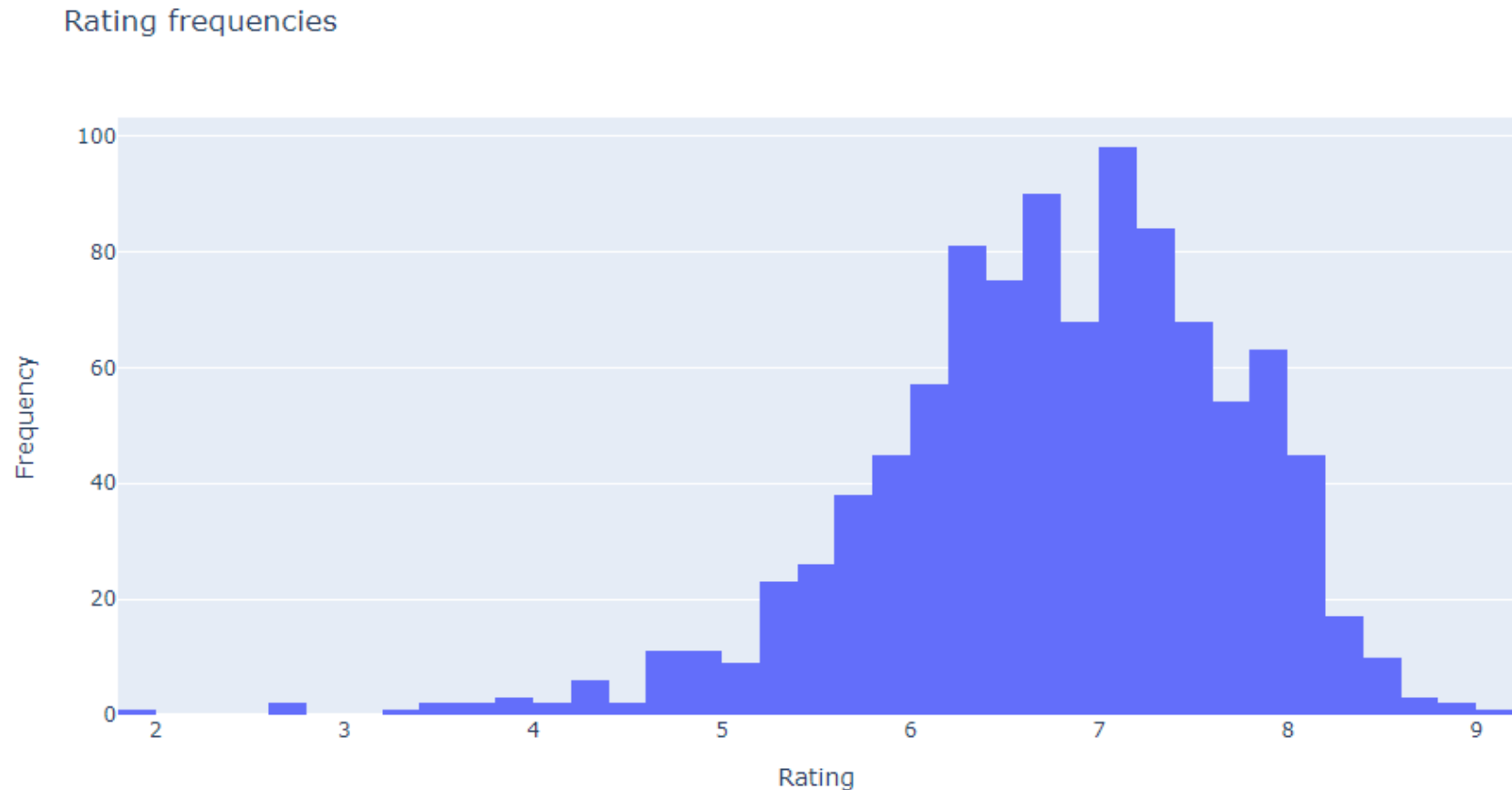
mylayout = go.Layout(title='Rating frequencies',
                     xaxis_title = 'Rating',
                     yaxis_title="Frequency")

fig = go.Figure(data = mydata, layout = mylayout)

fig.show()
```


Pandas를 이용한 데이터 시각화

❖ Pandas를 이용한 Plotly histogram 제작



2D Histogram

- ❖ 2D Histogram은 데이터 분포를 확인하기 위해 사용되는 히트맵이라고 부릅니다.
- ❖ `go.Histogram2d()`는 2D Histogram을 만드는 데 사용됩니다.

```
import plotly.graph_objects as go

import numpy as np
np.random.seed(1)

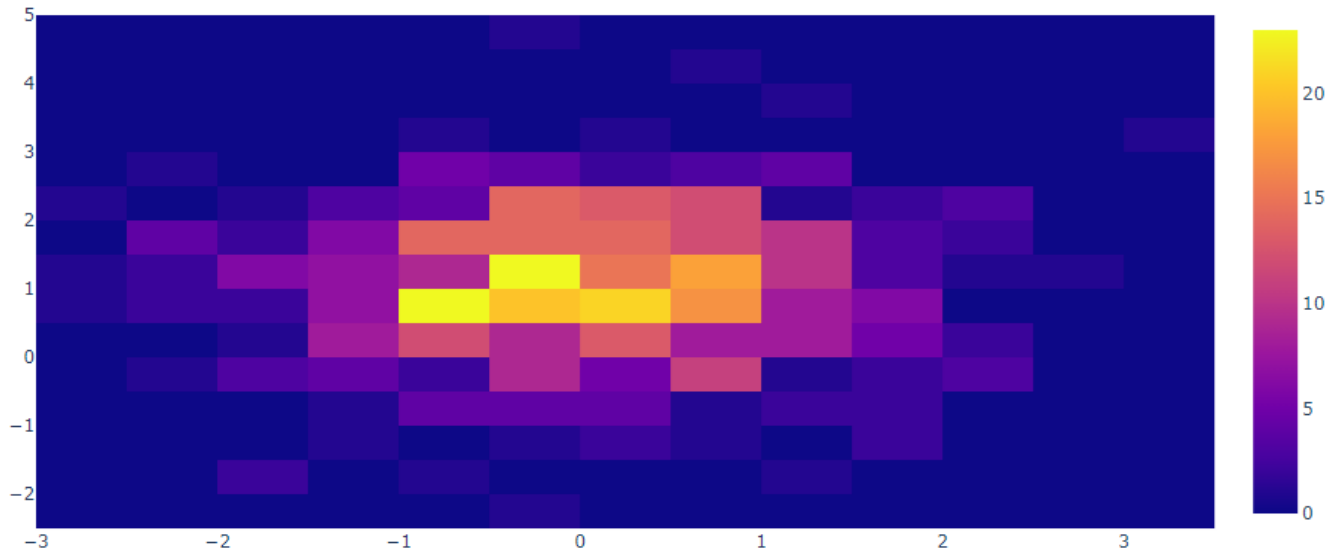
x = np.random.randn(500)
y = np.random.randn(500)+1

fig = go.Figure(go.Histogram2d(
    x=x,
    y=y
))
fig.show()
```

2D Histogram

❖ `go.Histogram2d()`는 2D Histogram을 만드는 데 사용됩니다.

■ 결과 출력



2D Histogram

❖ Example

```
import plotly.graph_objects as go

import numpy as np

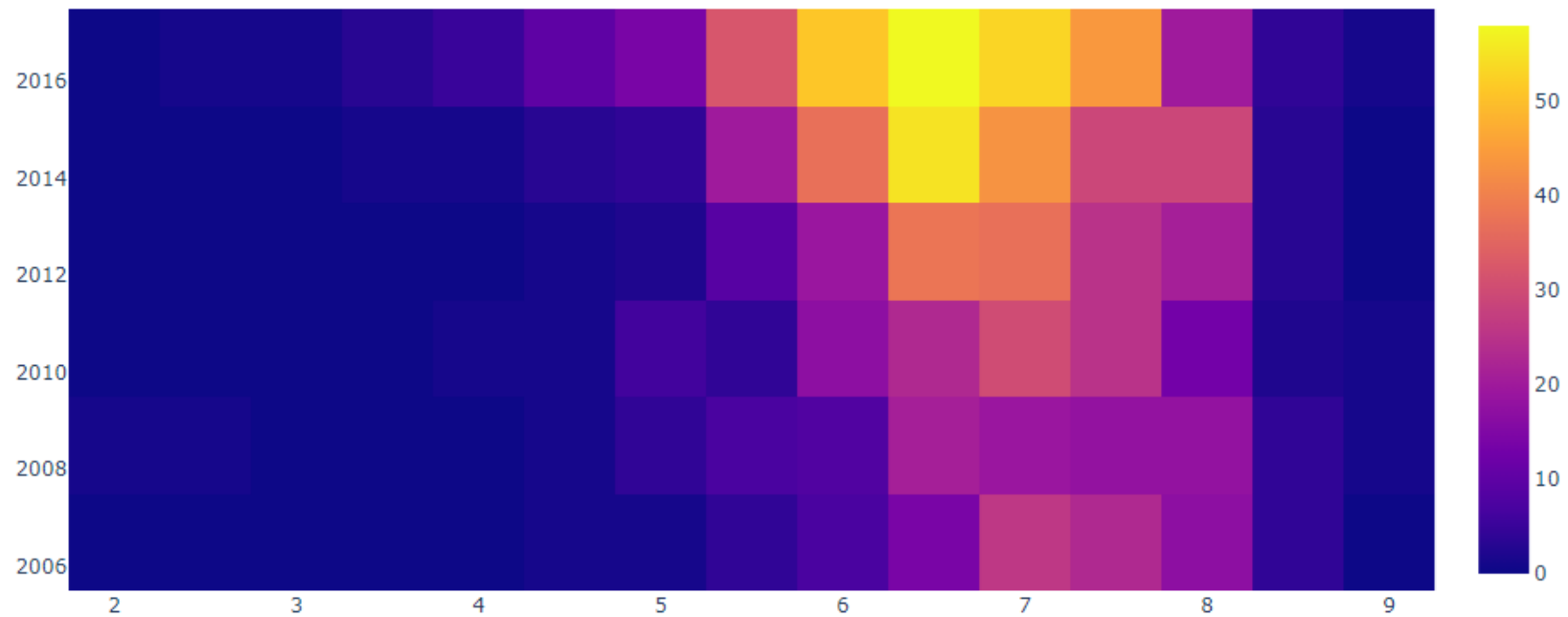
movies_df = pd.read_csv("D:/IMDB-Movie-Data.csv", index_col="Title")

x = movies_df['Rating']
y = movies_df['Year']

fig = go.Figure(go.Histogram2d(
    x=x,
    y=y
))
fig.show()
```

2D Histogram

❖ 결과 출력



Boxplot

- ❖ Boxplot은 주로 데이터에서 이상치를 탐지하는데 사용됩니다.
- ❖ `go.Box()`는 `boxplots`을 생성하는 데 사용됩니다.

```
import plotly.graph_objects as go

yaxis = [1140, 1460, 489, 594, 502, 508, 370, 200]

data = go.Box(y = yaxis)

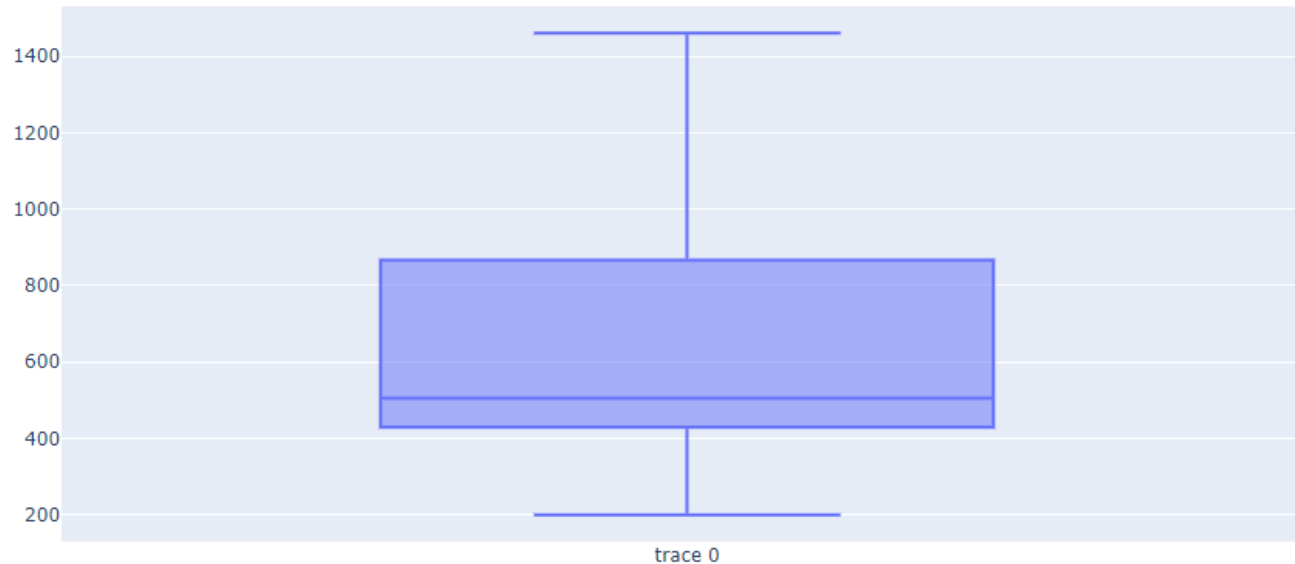
fig = go.Figure(data)

fig.show()
```

Boxplot

❖ `go.Box()`는 boxplots을 생성하는 데 사용됩니다.

- 결과 출력



Pandas를 이용한 데이터 시각화

❖ Pandas를 이용한 Plotly boxplot 제작

```
import plotly.graph_objects as go
import pandas as pd

movies_df = pd.read_csv("D:/IMDB-Movie-Data.csv", index_col="Title")

y_values = movies_df['Rating']

mydata = go.Box(y = y_values)

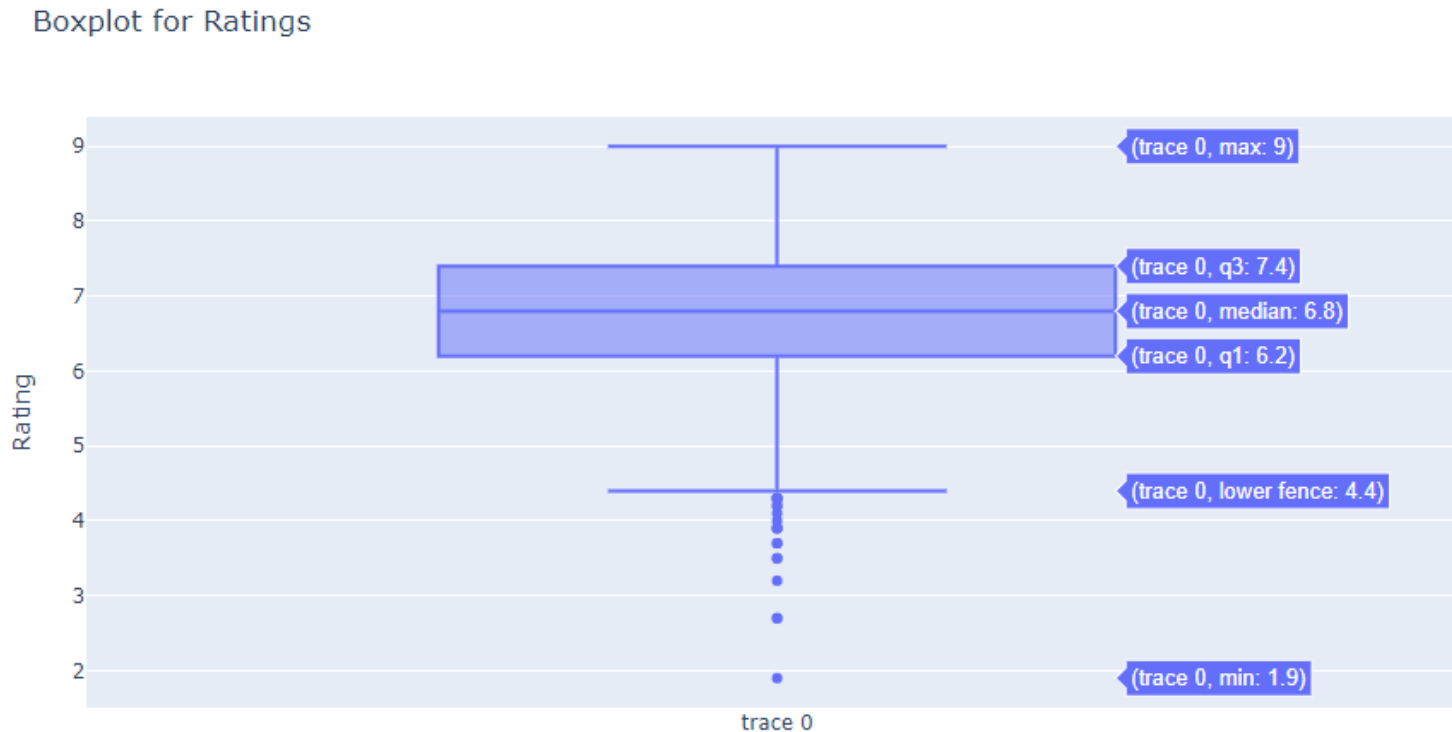
mylayout = go.Layout(title='Boxplot for Ratings', yaxis_title="Rating")

fig = go.Figure(mydata, mylayout)

fig.show()
```


Pandas를 이용한 데이터 시각화

❖ Pandas를 이용한 Plotly boxplot 제작



Violin plot

- ❖ Violin plots은 다른 값에서 데이터의 밀도를 표시한다는 점을 제외하면 box plots과 유사합니다.
- ❖ `go.Violin()` 사용
 - box plot을 표시하려면, `boxplot_visible` 속성을 `True`로 설정합니다.

```
import plotly.graph_objects as go
import numpy as np

np.random.seed(10)
c1 = np.random.normal(100, 10, 200)
c2 = np.random.normal(80, 30, 200)

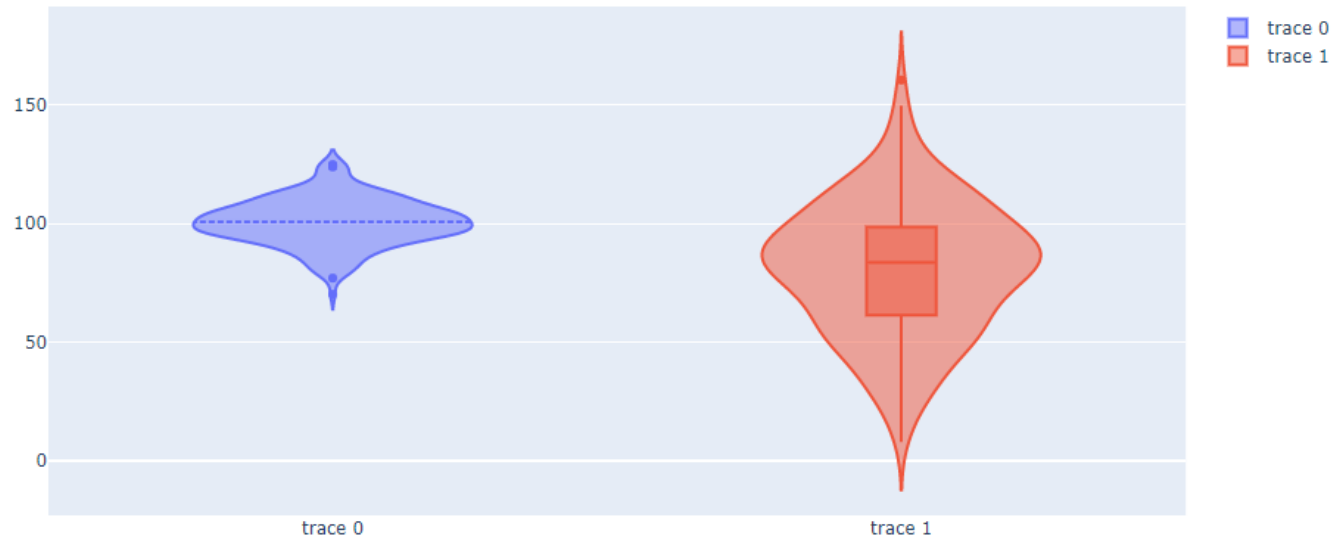
trace1 = go.Violin(y = c1, meanline_visible = True)
trace2 = go.Violin(y = c2, box_visible = True)

data = [trace1, trace2]
fig = go.Figure(data = data)
fig.show()
```

Violin plot

❖ go.Violin() 사용

■ 결과 출력



Homework for Lecture 5

❖ Given the following task:

1. Read `usedcars.csv`
2. Draw bar graph OR pie chart
3. Draw scatter plot OR line plot
4. Draw histogram OR 2D histogram
5. Draw boxplot OR violin plot

❖ Submission: source code, result screenshots and result explanation



감사합니다!