EDA and
Data Visualization

1. EDA

2. R의 시각화 패키지-ggIpot2

3. Python의 시각화 패키지-Matpoltlib

4.그 외 시각화 패키지

EDA

데이터를 수집했을때 이를 다양한 각도에서 관찰하고 이해하는 과정

- **Comment**

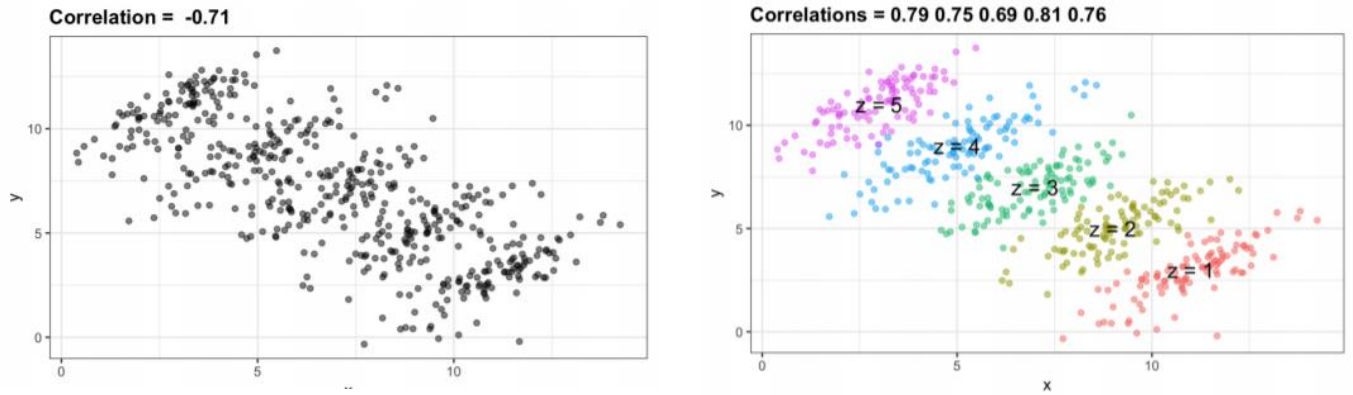
“Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise.” — John Tukey

“탐색적 데이터 분석 (EDA)은 우리가 존재한다고 믿는 것들은 물론이고 존재하지 않는다고 믿는 것들을 발견하려는 태도, 유연성, 그리고 자발성이다.” — Schut Rachel

EDA의 필요성

- 데이터의 분포 및 값을 검토함으로써 데이터가 표현하는 현상을 더 잘 이해하고, 데이터에 대한 잠재적인 문제를 발견할 수 있다.

- 다양한 각도에서 살펴보는 과정을 통해 문제 정의 단계에서 미처 발생하지 못한 다양한 패턴을 발견하고, 이를 바탕으로 기존의 가설을 수정하거나 새로운 가설을 세울 수 있다.



(simpson's paradox)

EDA의 과정

①분석의 목적과 변수가 무엇이 있는지 확인. 개별 변수의 이름이나 설명을 가지는지 확인.

②데이터를 전체적으로 살펴보기 : 데이터에 문제가 없는지 확인. head나 tail부분을 확인, 추가적으로 다양한 탐색. (이상치, 결측치 등을 확인하는 과정)

③데이터의 개별 속성값을 관찰 : 각 속성값이 예측한 범위와 분포를 갖는지 확인. 만약 그렇지 않다면, 이유가 무엇인지를 확인해 본다.

④속성 간의 관계에 초점을 맞추어, 개별 속성 관찰에서 찾아내지 못했던 패턴을 발견한다. (상관관계, **시각화** 등)

R - ggplot2

ggplot2

A Layered Grammar of Graphics (Hadley Wickham)

All Grammatical Elements

Element	Description
Data	The dataset being plotted.
Aesthetics	The scales onto which we *map* our data.
Geometries	The visual elements used for our data.
Facets	Plotting small multiples.
Statistics	Representations of our data to aid understanding.
Coordinates	The space on which the data will be plotted.
Themes	All non-data ink.

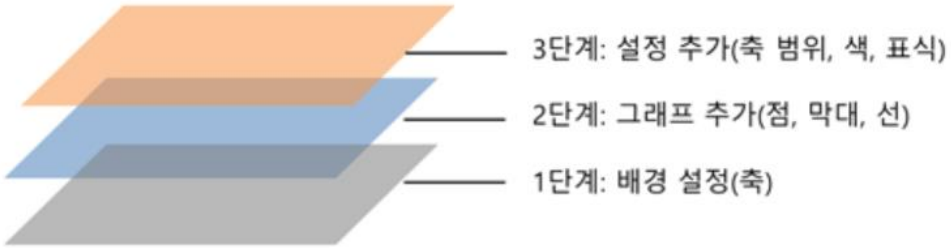
3개의 필수 Layer

4개의 부가 Layer

```
ggplot(data, aes()) +
 geom_***()
```

```
ggplot(data, aes()) +
 geom_***() +
 facet_***() +
 stat_***() +
 coord_***() +
 theme_***()
```

ggplot2



ggplot2 레이어 구조

1단계 – Data, Aesthetics

2단계 – Geometrics

3단계 – 4개의 부가 layer

ggplot2 활용 - 0단계

성별과 연령대에 따른 코로나 확진자 분포

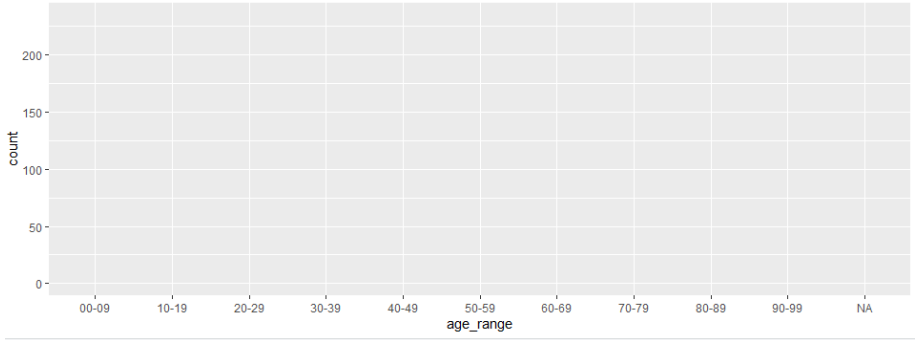
```
library(ggplot2)
library(tidyverse)
head(patient)
```

```
# A tibble: 6 x 3
# Groups:   sex [1]
  sex    age_range count
  <chr>  <chr>     <int>
1 female 00-09        11
2 female 10-19        26
3 female 20-29       208
4 female 30-39       117
5 female 40-49       186
6 female 50-59       234
```

ggplot2 활용 - 1단계

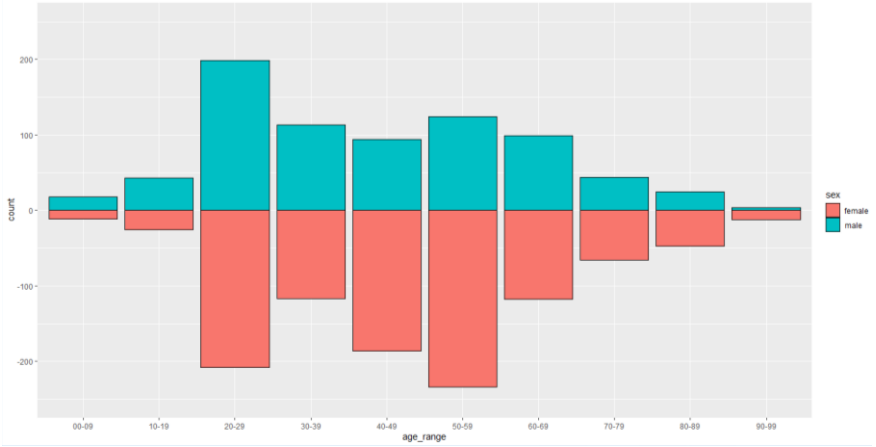
1단계. Data, Aesthetics

```
patient %>%
  ggplot(aes(x = age_range, y = count, fill = sex))
```



ggplot2 활용 - 2단계

```
patient %>%
  ggplot(aes(x = age_range, y = count, fill = sex)) +
  geom_blank(
    # dummy data
    data = data.frame(
      age_range = c('00-09', '00-09'),
      sex = c('male', 'female'),
      count = c(250, -250)
    )
  ) +
  geom_col(data = . %>% subset(sex == 'male'),
           aes(y = count), color = 'black') +
  geom_col(data = . %>% subset(sex == 'female'),
           aes(y = -1 * count),
           position = 'identity', color = 'black')
```

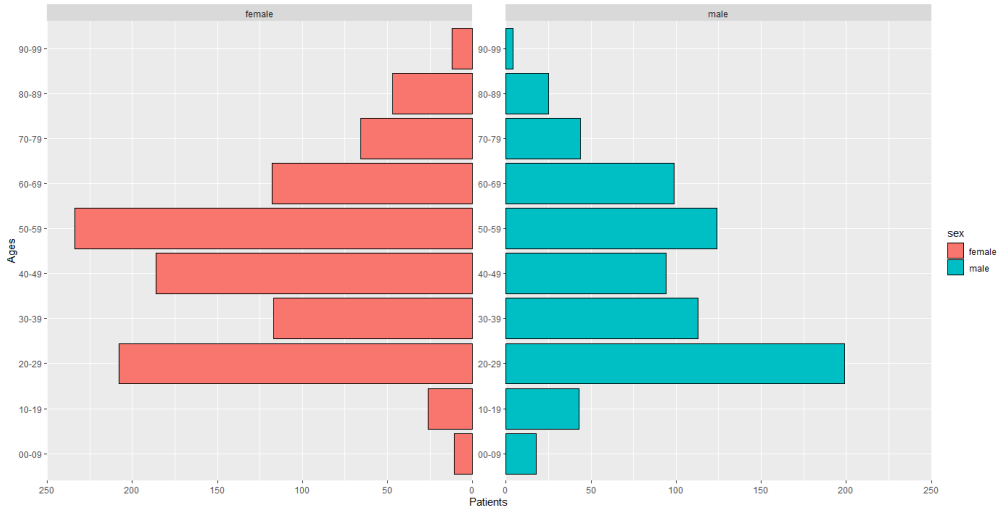


ggplot2 활용 - 3단계

```
patient %>%
  ggplot(aes(x = age_range, y = count, fill = sex)) +
```

(2단계)

```
  coord_flip() +
  facet_wrap(
    ~sex,
    scales = 'free'
  ) +
  scale_y_continuous(
    expand = c(0, 0),
    labels = abs
  ) +
  labs(x = 'Ages', y = 'Patients')
```



Python - matplotlib

matplotlib

Workflow

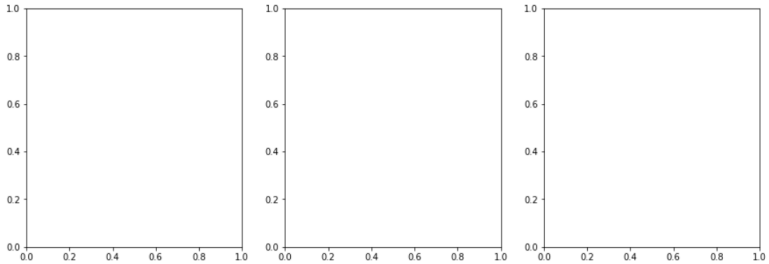
0. Load package and prepare data
1. Create plot
2. Plot
3. Customize plot
4. Save plot
5. Show plot

Matplotlib 활용 - 0단계

```
import matplotlib
import matplotlib.pyplot as plt
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
fig, ax = plt.subplots(1,3, figsize=(15,5))
ax[0].plot(x,y, marker='o')
ax[1].scatter(x,y, c = 'red')
ax[2].bar(x,y)
ax[0].set_title('line chart')
ax[1].set_title('scatter plot')
ax[2].set_title('bar chart')
plt.savefig('plot.png')
plt.show()
```

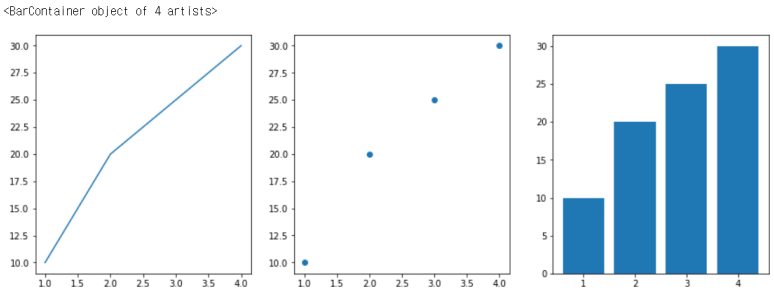
matplotlib 활용 - 1단계

```
import matplotlib
import matplotlib.pyplot as plt
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
fig, ax = plt.subplots(1,3, figsize=(15,5))
ax[0].plot(x,y, marker='o')
ax[1].scatter(x,y, c = 'red')
ax[2].bar(x,y)
ax[0].set_title('line chart')
ax[1].set_title('scatter plot')
ax[2].set_title('bar chart')
plt.savefig('plot.png')
plt.show()
```



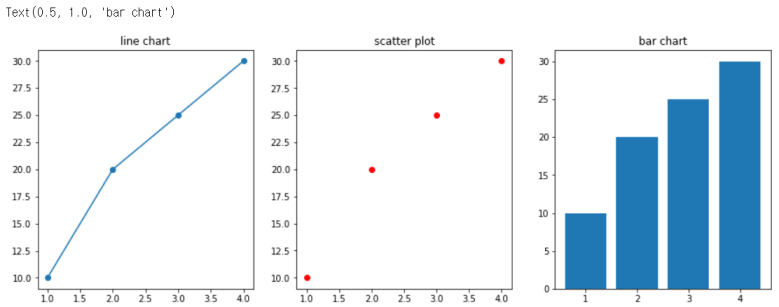
Matplotlib 활용 - 2단계

```
import matplotlib
import matplotlib.pyplot as plt
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
fig, ax = plt.subplots(1,3, figsize=(15,5))
ax[0].plot(x,y, marker='o')
ax[1].scatter(x,y, c = 'red')
ax[2].bar(x,y)
ax[0].set_title('line chart')
ax[1].set_title('scatter plot')
ax[2].set_title('bar chart')
plt.savefig('plot.png')
plt.show()
```



Matplotlib 활용 - 3단계

```
import matplotlib
import matplotlib.pyplot as plt
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
fig, ax = plt.subplots(1,3, figsize=(15,5))
ax[0].plot(x,y, marker='o')
ax[1].scatter(x,y, c = 'red')
ax[2].bar(x,y)
ax[0].set_title('line chart')
ax[1].set_title('scatter plot')
ax[2].set_title('bar chart')
plt.savefig('plot.png')
plt.show()
```

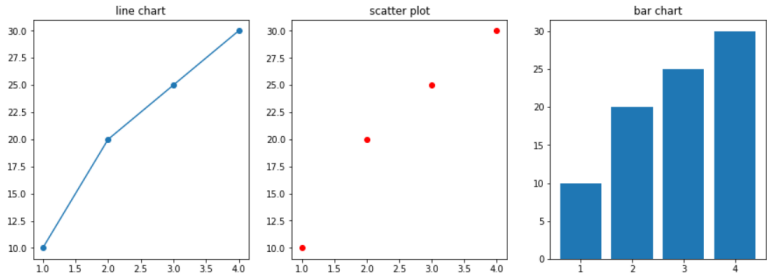


Matplotlib 활용 - 4단계

```
import matplotlib
import matplotlib.pyplot as plt
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
fig, ax = plt.subplots(1,3, figsize=(15,5))
ax[0].plot(x,y, marker='o')
ax[1].scatter(x,y, c = 'red')
ax[2].bar(x,y)
ax[0].set_title('line chart')
ax[1].set_title('scatter plot')
ax[2].set_title('bar chart')
plt.savefig('plot.png')
plt.show()
```

Matplotlib 활용 - 5단계

```
import matplotlib
import matplotlib.pyplot as plt
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
fig, ax = plt.subplots(1,3, figsize=(15,5))
ax[0].plot(x,y, marker='o')
ax[1].scatter(x,y, c = 'red')
ax[2].bar(x,y)
ax[0].set_title('line chart')
ax[1].set_title('scatter plot')
ax[2].set_title('bar chart')
plt.savefig('plot.png')
plt.show()
```



그 외

*그 외 유용한 시각화 패키지
R – ‘vcd’ 패키지에 있는 mosaicplot : 다차원 범주형 데이터 시각화에 적절
Python – ‘seaborn’ 패키지: 통계적 분포를 살펴보는 데에 적절
– ‘plotly’ 패키지 : 인터렉티브한 시각화 가능

reference

- A Layered Grammar of Graphics(Hadley WICKHAM)
- Python_Matplotlib_Cheat_Sheet

Q & A