

최종작성: 2021.12.05(일)

# 사례연구 6

DATA VISUALIZE

2021.12.05(일) 제출

**C3 조**

조원: 오준서 천성한 한호종 황윤수

## 내용

서론 .....	6
본론 .....	7
1. 막대차트 .....	7
1) 공통 데이터 .....	7
2) graphics(R) 패키지를 이용한 시각화 .....	7
3) ggplot2(R) 패키지를 이용한 시각화 .....	7
4) plotly(R) 패키지를 이용한 시각화 .....	8
5) matplotlib(py) 패키지를 이용한 시각화 .....	8
6) 각 패키지별 비교결과 .....	8
2. 누적 막대차트 .....	9
1) 공통 데이터 .....	9
2) graphics(R) 패키지를 이용한 시각화 .....	9
3) ggplot2(R) 패키지를 이용한 시각화 .....	10
4) plotly(R) 패키지를 이용한 시각화 .....	10
5) matplotlib(py) 패키지를 이용한 시각화 .....	11
6) 각 패키지별 비교결과 .....	11
3. 점 차트 .....	12
1) 공통 데이터 .....	12
2) graphics(R) 패키지를 이용한 시각화 .....	12
3) ggplot2(R) 패키지를 이용한 시각화 .....	13
4) plotly(R) 패키지를 이용한 시각화 .....	13
5) matplotlib(py) 패키지를 이용한 시각화 .....	14
6) 각 패키지별 비교결과 .....	14
4. 원형 차트 .....	15
1) 공통 데이터 .....	15

2)	graphics(R) 패키지를 이용한 시각화 .....	15
3)	ggplot2(R) 패키지를 이용한 시각화 .....	16
4)	plotly(R) 패키지를 이용한 시각화 .....	16
5)	matplotlib(py) 패키지를 이용한 시각화 .....	17
6)	각 패키지별 비교결과 .....	17
5.	상자 그래프 .....	18
1)	공통 데이터 .....	18
2)	graphics(R) 패키지를 이용한 시각화 .....	18
3)	ggplot2(R) 패키지를 이용한 시각화 .....	19
4)	plotly(R) 패키지를 이용한 시각화 .....	19
5)	matplotlib(py) 패키지를 이용한 시각화 .....	20
6)	각 패키지별 비교결과 .....	20
6.	히스토그램 .....	21
1)	공통 데이터 .....	21
2)	graphics(R) 패키지를 이용한 시각화 .....	21
3)	ggplot2(R) 패키지를 이용한 시각화 .....	22
4)	plotly(R) 패키지를 이용한 시각화 .....	22
5)	matplotlib(py) 패키지를 이용한 시각화 .....	23
6)	각 패키지별 비교결과 .....	23
7.	산점도 .....	24
1)	공통 데이터 .....	24
2)	graphics(R) 패키지를 이용한 시각화 .....	24
3)	ggplot2(R) 패키지를 이용한 시각화 .....	25
4)	plotly(R) 패키지를 이용한 시각화 .....	25
5)	matplotlib(py) 패키지를 이용한 시각화 .....	26
6)	각 패키지별 비교결과 .....	26

8.	중첩자료 시각화.....	27
1)	공통 데이터.....	27
2)	graphics(R) 패키지를 이용한 시각화 .....	27
3)	ggplot2(R) 패키지를 이용한 시각화 .....	28
4)	plotly(R) 패키지를 이용한 시각화 .....	28
5)	matplotlib(py) 패키지를 이용한 시각화 .....	29
6)	각 패키지별 비교결과.....	29
9.	변수간의 비교 시각화.....	30
1)	공통 데이터.....	30
2)	graphics(R) 패키지를 이용한 시각화 .....	30
3)	ggally(R) 패키지를 이용한 시각화 .....	31
4)	matplotlib(py) 패키지를 이용한 시각화 .....	31
5)	각 패키지별 비교결과.....	31
10.	밀도 그래프.....	32
1)	공통 데이터.....	32
2)	graphics(R) 패키지를 이용한 시각화 .....	32
3)	ggplot2(R) 패키지를 이용한 시각화 .....	33
4)	plotly(R) 패키지를 이용한 시각화 .....	33
5)	matplotlib(py) 패키지를 이용한 시각화 .....	34
11.	3 차원 산점도 그래프 .....	35
1)	공통 데이터.....	35
2)	scatterplot3d(R) 패키지를 이용한 시각화 .....	35
3)	plotly(R) 패키지를 이용한 시각화 .....	36
4)	matplotlib(py) 패키지를 이용한 시각화 .....	36
5)	각 패키지별 비교결과.....	37
12.	지도 시각화.....	38

1)	공통 데이터.....	38
2)	ggmap(R), plotly(R) 패키지를 이용한 시각화.....	38
3)	시각화 결과.....	39
결론 .....		40
부록 .....		41

## 서론

Python & R 을 활용하여 각종 그래프를 시각화하고 각 패키지별 비교결과를 분석한 보고서이다.

시각화에 사용된 데이터 셋은 다음과 같다.

**막대차트:** 분기별 매출액 데이터

**누적 막대차트:** VADeaths (버지니아주 하위계층 사망비율) 데이터

**점 차트:** 분기별 매출액 데이터

**원형 차트:** 분기별 매출액 데이터

**상자 그래프:** VADeaths (버지니아주 하위계층 사망비율) 데이터

**히스토그램:** iris 데이터

**산점도:** 1 ~ 100 사이의 난수 10 개

**중첩자료 시각화:** galton 데이터

**변수간의 비교 시각화:** iris 데이터

**밀도 그래프:** iris 데이터

**3 차원 산점도:** iris 데이터

**지도 시각화:** population201901.csv

# 본론

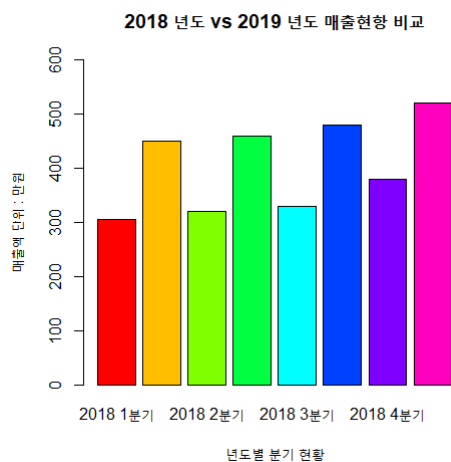
## 1. 막대차트

### 1) 공통 데이터

	2018 1 분기	2019 1 분기	2018 2 분기	2019 2 분기	2018 3 분기	2019 3 분기	2018 4 분기	2019 4 분기
매출액 (단위: 만원)	305	450	320	460	330	480	380	520

(표 1-1) 막대 차트 생성에 사용한 데이터

### 2) graphics(R) 패키지를 이용한 시각화



그림

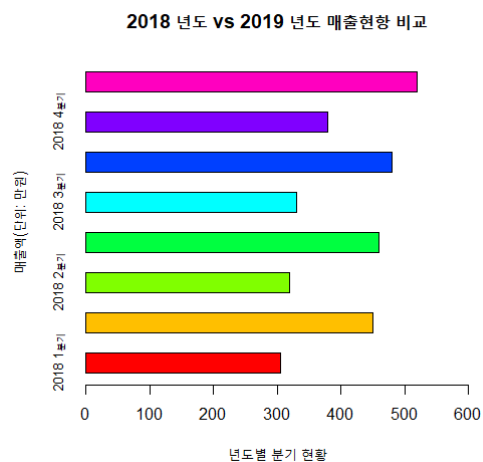
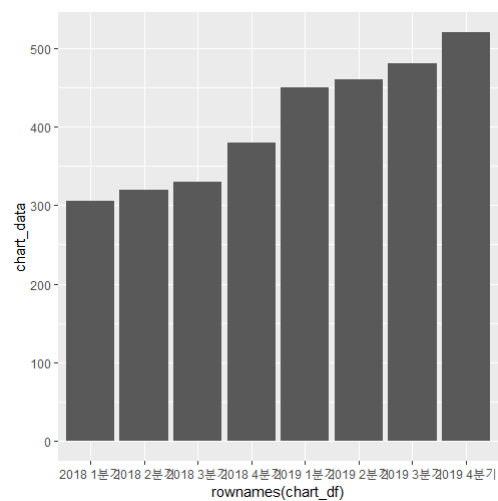


그림 1-2)

### 3) ggplot2(R) 패키지를 이용한 시각화



그림

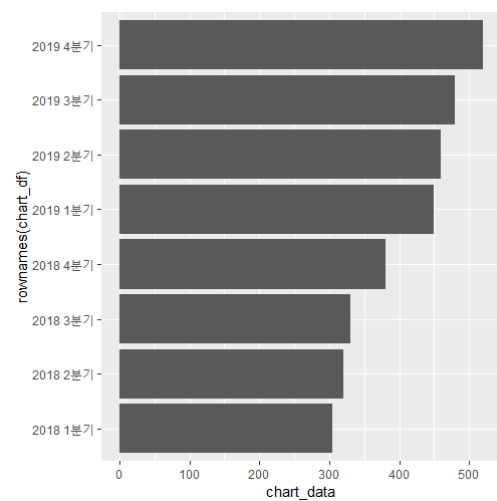
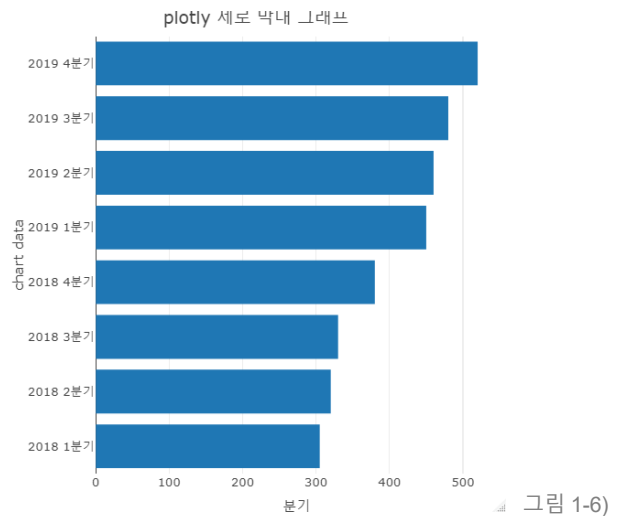
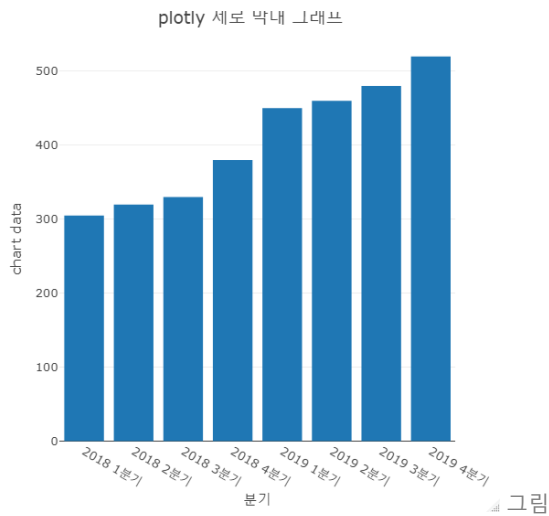


그림 1-4)

#### 4) plotly(R) 패키지를 이용한 시각화



#### 5) matplotlib(py) 패키지를 이용한 시각화

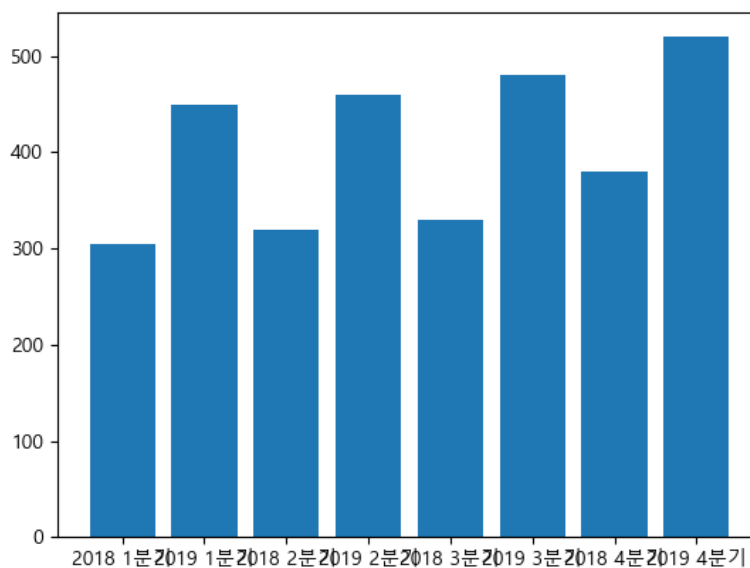


그림 1-7)

#### 6) 각 패키지별 비교결과

막대차트(가로,세로)에서는 특수한 옵션없이 x,y 축의 이름만 설정하여 그래프를 그렸다.

그 결과 plotly(R) 패키지를 활용한 그래프가 가장 시인성이 좋았다. y 축의 값에 따라 오름차순으로 자동정렬 되었으며, x 축이름이 길어져도 대각선으로 이름들을 출력해서 x 축이름들이 서로 겹치지 않은 부분이 편리했다.



## 2. 누적 막대차트

### 1) 공통 데이터

VADeaths	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37	19.3
65-69	41	30.9	54.6	35.1
70-74	66	54.3	71.1	50

(표 2-1) 누적 막대 차트 생성에 사용한 VADeaths 데이터

### 2) graphics(R) 패키지를 이용한 시각화

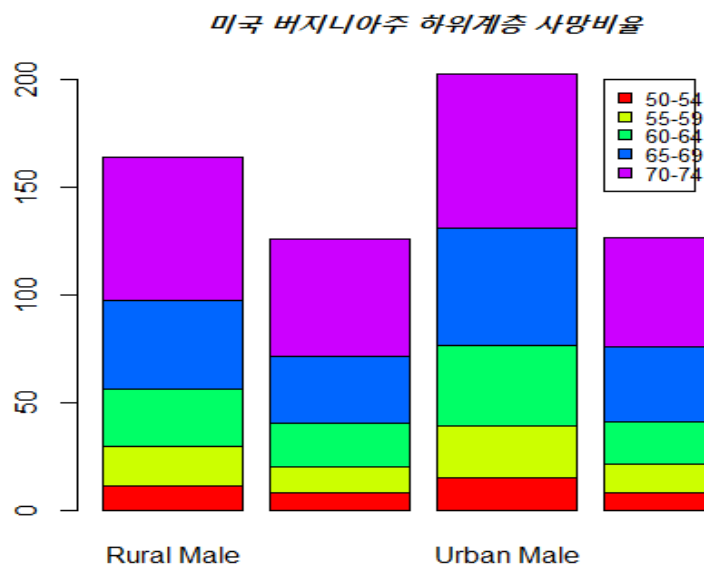


그림 2-1)

범례의 위치와 색상을 별도로 지정하여 R 기본 내장 패키지로 시각화한 그림이다.

### 3) ggplot2(R) 패키지를 이용한 시각화

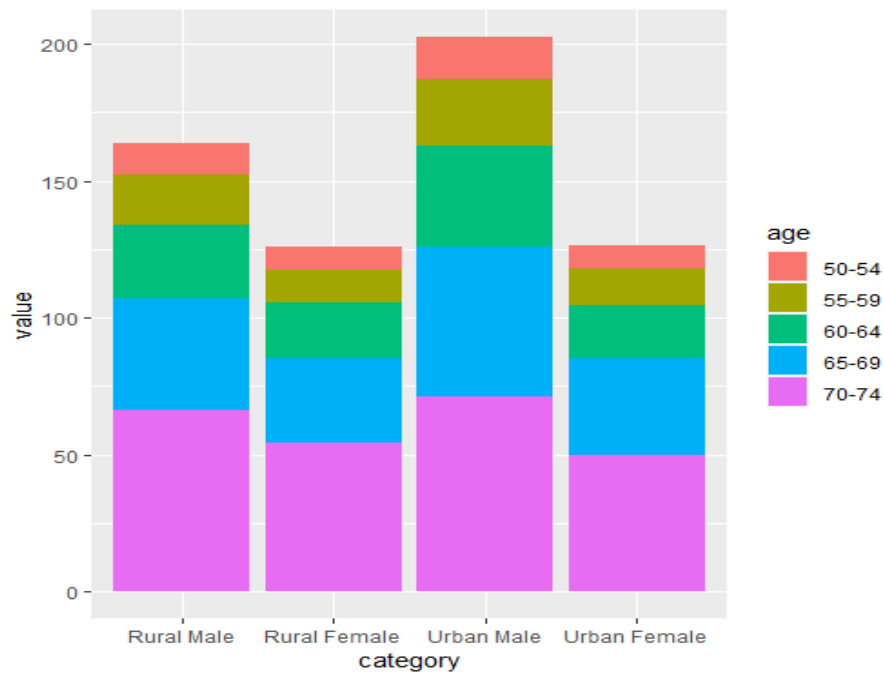


그림 2-2)

이 패키지에서는 범례의 기준과 그래프 타입만을 지정하여 시각화하였다.

### 4) plotly(R) 패키지를 이용한 시각화

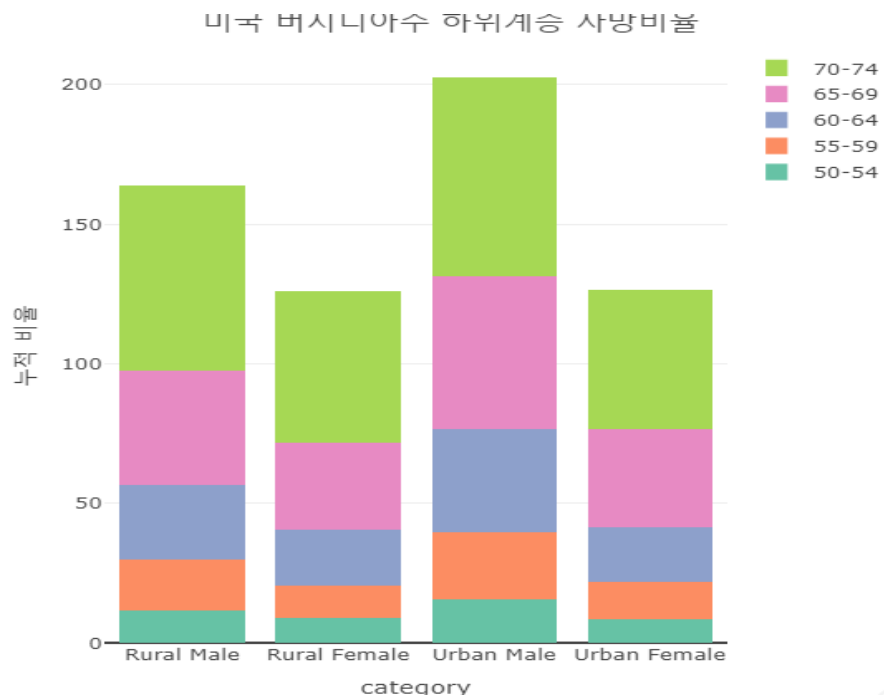


그림 2-3)

위와 동일하나 각 축의 이름과 제목을 추가하였다.

## 5) matplotlib(py) 패키지를 이용한 시각화

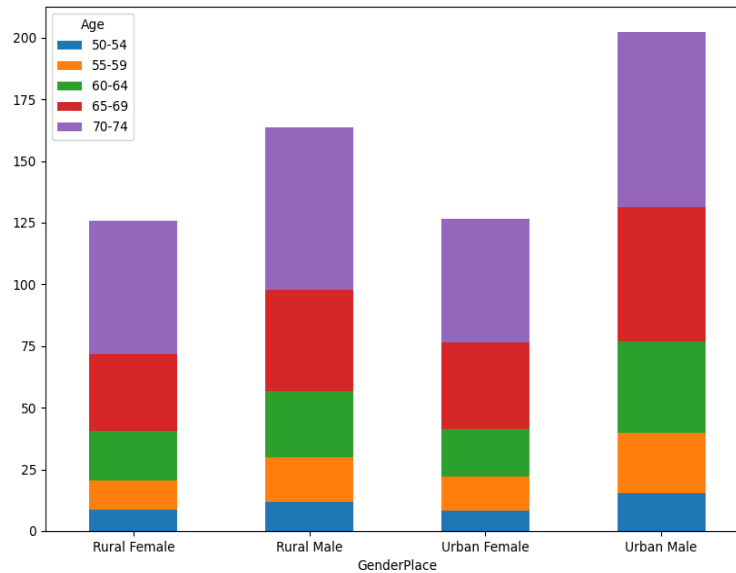


그림 2-4)

matplotlib(Py) 패키지를 활용한 시각화 그림이다.

## 6) 각 패키지별 비교결과

R 내장 그래픽 패키지를 통한 시각화에서는 x 축 값이 생략되는 문제가 발생하였으나 나머지 패키지들에서는 정상적으로 그래프가 출력되었고, 누적 막대 그래프에서는 눈금값이 촘촘한 그래프가 좀더 비교하기에 수월했다.

### 3. 점 차트

#### 1) 공통 데이터

VADeaths	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37	19.3
65-69	41	30.9	54.6	35.1
70-74	66	54.3	71.1	50

(표 3-1) 점 차트 생성에 사용한 VADeaths 데이터

#### 2) graphics(R) 패키지를 이용한 시각화

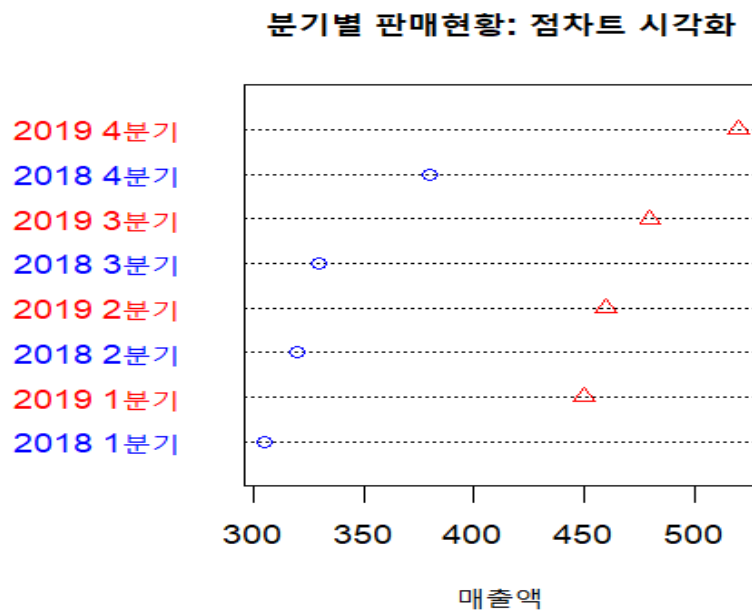


그림 3-1)

R 기본 내장 그래픽 패키지를 통해 점 크기, 색상, x 축 이름, y 축 눈금 색상, 그래프 제목을 별도로 지정하여 그린 그래프이다.

### 3) ggplot2(R) 패키지를 이용한 시각화

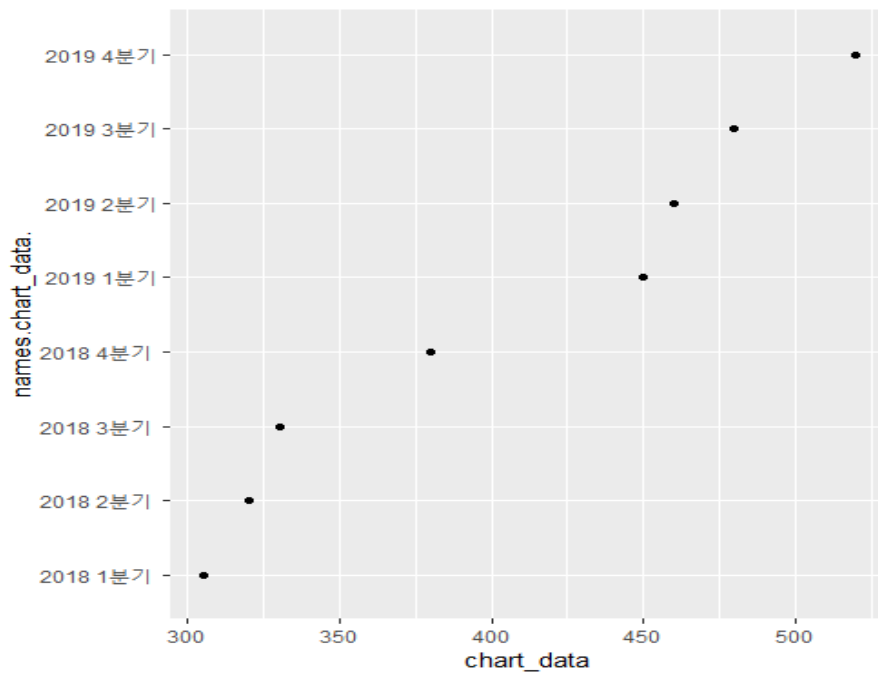


그림 3-2)

별도의 옵션 지정 없이 ggplot2 를 이용하여 시각화한 그래프이다.

### 4) plotly(R) 패키지를 이용한 시각화

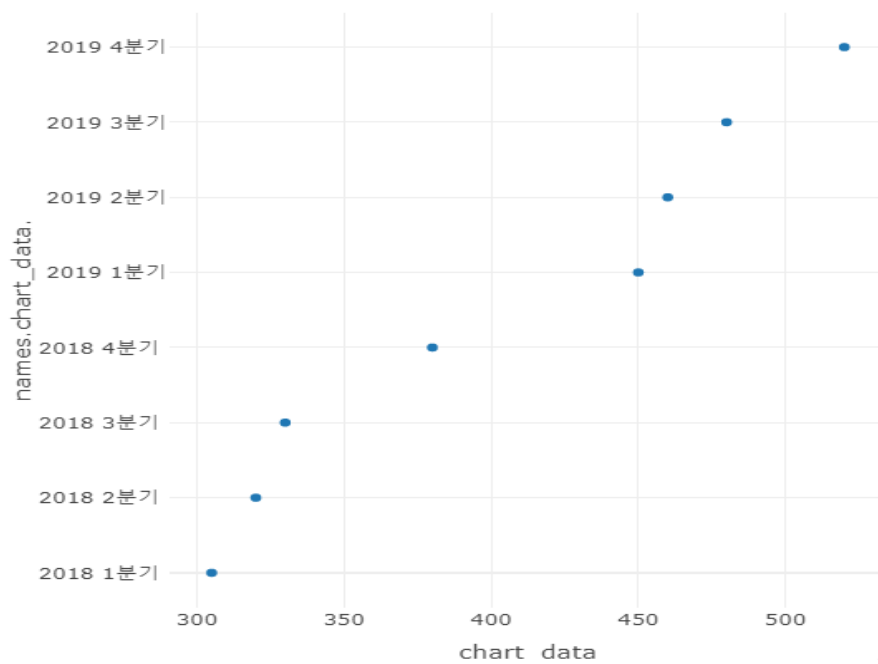


그림 3-3)

별도의 옵션 지정 없이 plotly(R) 함수를 이용하여 시각화한 그래프이다.

## 5) matplotlib(py) 패키지를 이용한 시각화

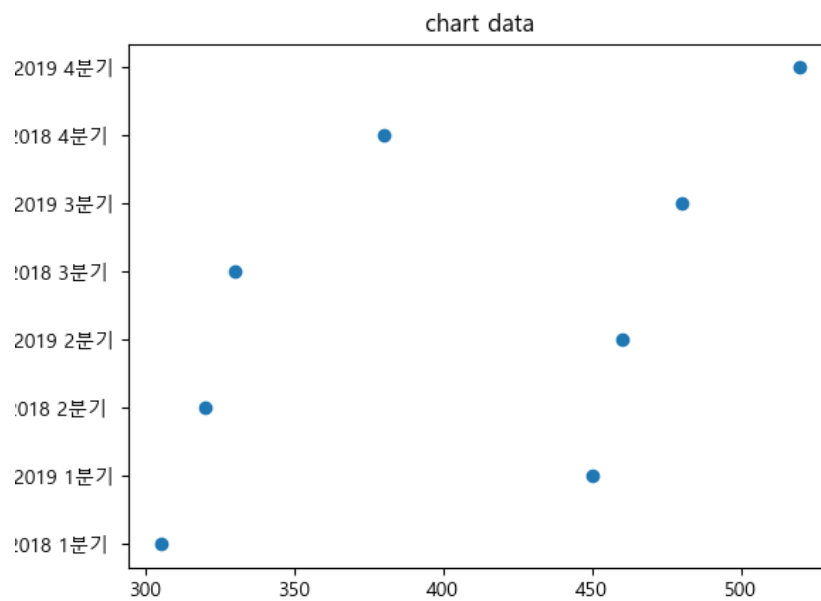


그림 3-4)

별도의 옵션 지정 없이 matplotlib(Py) 함수를 이용하여 시각화한 그래프이다.

## 6) 각 패키지별 비교결과

점 차트 그래프에서는 격자가 표시되어 있는 그래프가 수치들을 파악하기 쉬웠다. 각 패키지 중에서는 연도별로 색상을 지정한 R 내장 그래픽 함수가 시인성이 가장 좋았다.

#### 4. 원형 차트

##### 1) 공통 데이터

VADeaths	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37	19.3
65-69	41	30.9	54.6	35.1
70-74	66	54.3	71.1	50

(표 4-1) 원형 차트 생성에 사용한 VADeaths 데이터

##### 2) graphics(R) 패키지를 이용한 시각화

#### 2018~2019년도 분기별 매출현황

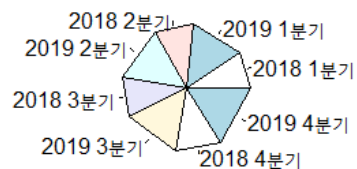


그림 4-1)

그래픽 패키지에서 `pie()` 함수를 이용하여 시각화한 그래프이다.

설정된 옵션은 아래와 같다.

**edges = 10** : 각 지점의 연결 선의 호의 각도를 평평하게 조정

**radius = 1** : 그래프의 크기 조정

### 3) ggplot2(R) 패키지를 이용한 시각화

2018~2019년도 분기별 매출현황

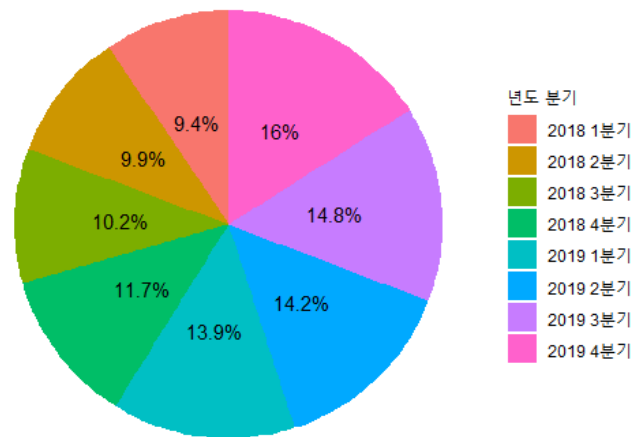


그림 4-2)

ggplot2(R) 패키지에서는 차트 안에 %를 나타내는 옵션과 차트 제목, 범주 제목을 설정하였다.

### 4) plotly(R) 패키지를 이용한 시각화

2018~2019년도 분기별 매출현황

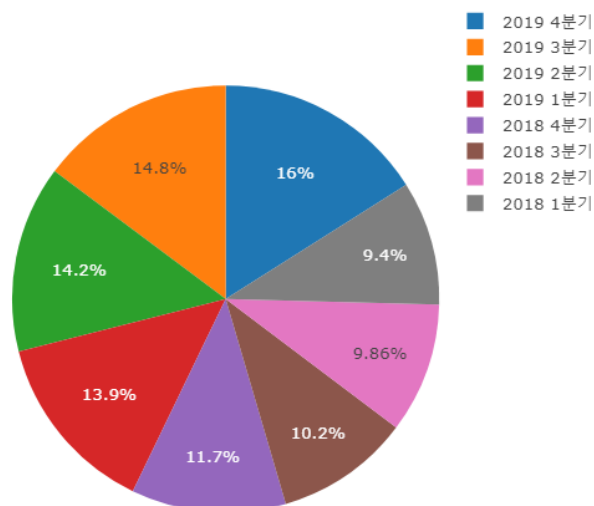


그림 4-3)

plotly(R) 패키지에서는 차트 제목만 설정하여 시각화한 결과이다.



## 5) matplotlib(py) 패키지를 이용한 시각화

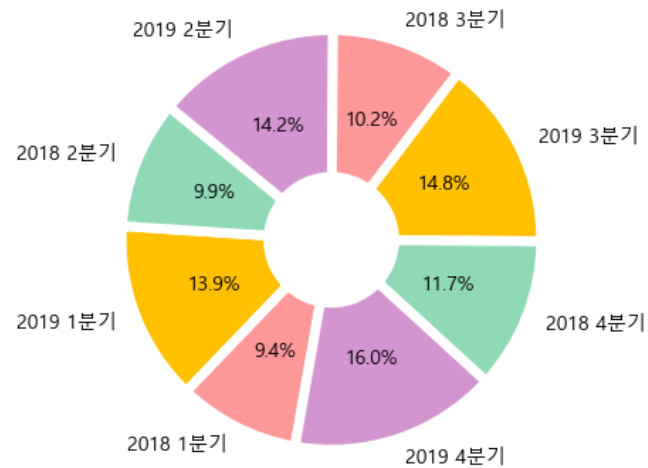


그림 4-4)

## 6) 각 패키지별 비교결과

원형 차트에서는 범례가 많아질수록 색상을 통한 구분보다 그래프 주변에 직접 값을 표기하는 것이 시인성이 좋았다. 그런 측면에서 matplotlib(Py) 패키지를 통한 시각화가 다수의 범례를 표기하는 매출액 그래프 중 가장 효과적이었음을 알 수 있다.

## 5. 상자 그래프

### 1) 공통 데이터

VADeaths	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37	19.3
65-69	41	30.9	54.6	35.1
70-74	66	54.3	71.1	50

(표 5-1) 원형 차트 생성에 사용한 VADeaths 데이터

### 2) graphics(R) 패키지를 이용한 시각화

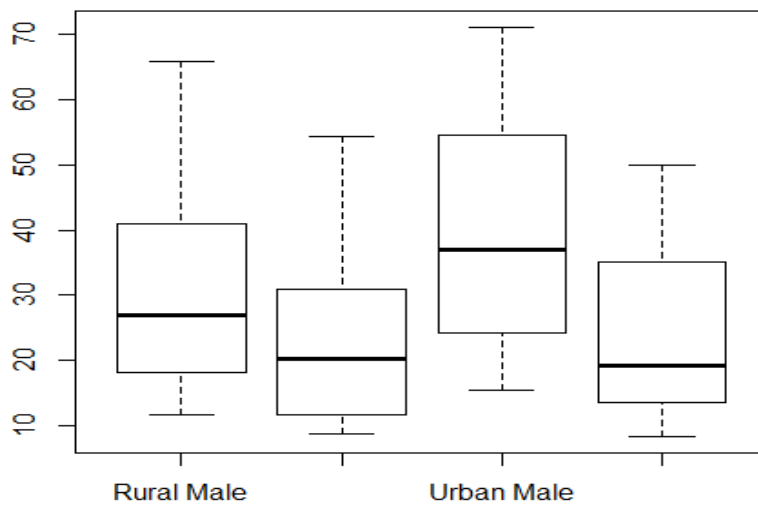


그림 5-1)

R 내장 그래픽 패키지를 통해 별도의 옵션 지정없이 시각화한 그래프이다.

### 3) ggplot2(R) 패키지를 이용한 시각화

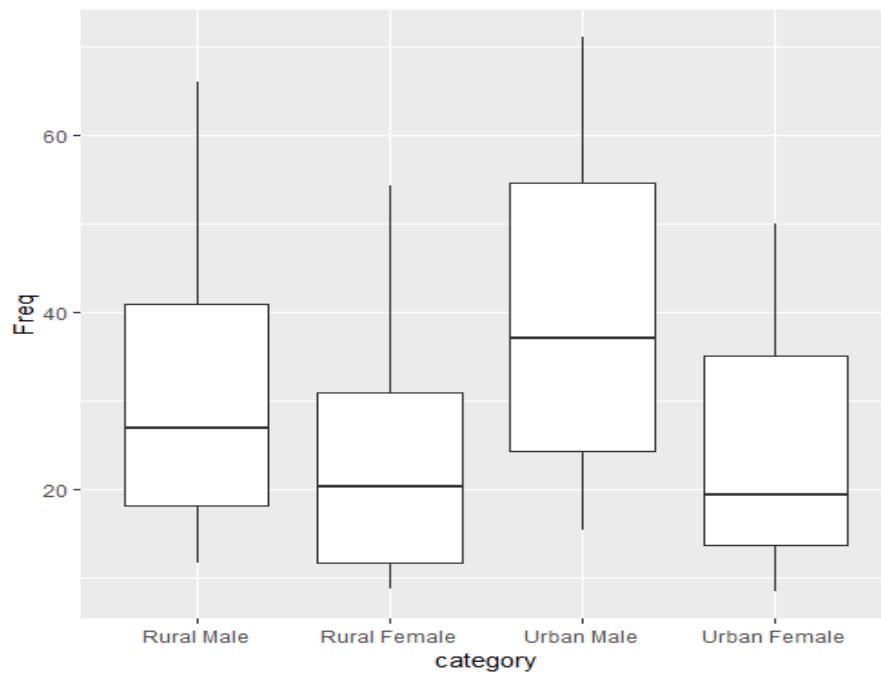


그림 5-2)

ggplot2(R) 패키지를 통해 별도의 옵션 지정없이 시각화한 그래프이다.

### 4) plotly(R) 패키지를 이용한 시각화

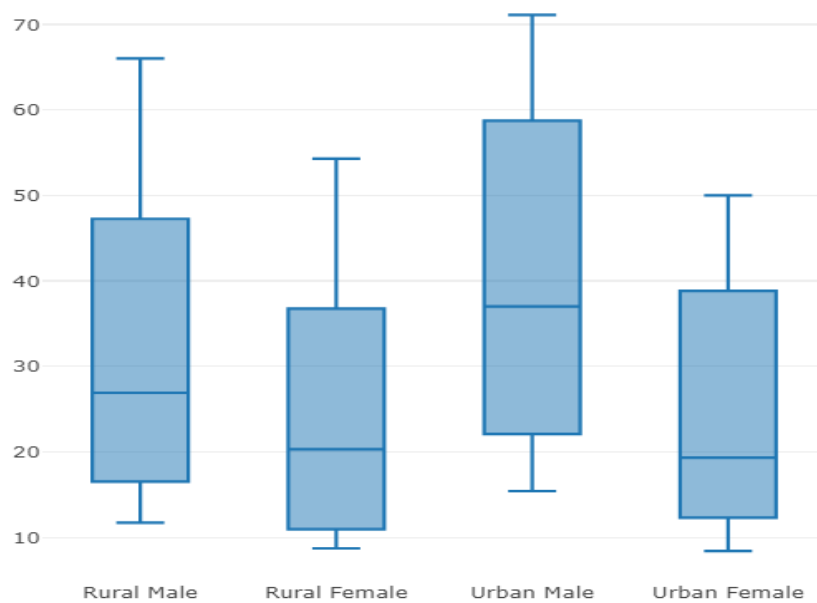


그림 5-3)

plotly(R) 패키지를 통해 별도의 옵션 지정없이 시각화한 그래프이다.

## 5) matplotlib(py) 패키지를 이용한 시각화

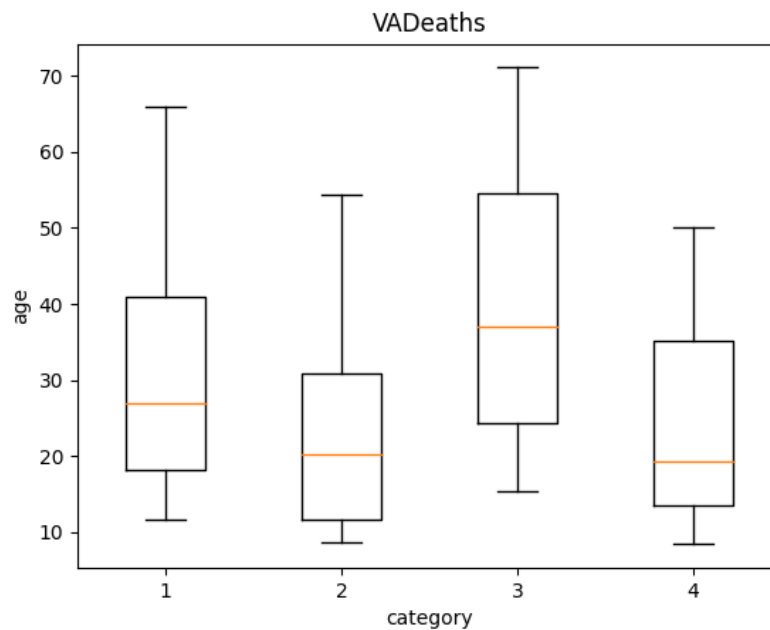


그림 5-4)

주어진 데이터를 matplotlib(Py) 패키지를 활용하여 그래프 제목과 각 축의 이름을 별도로 지정하고 시각화한 그래프이다.

## 6) 각 패키지별 비교결과

별도로 옵션을 지정하지 않았을 때는 plotly(R) 패키지에서 시각화한 그래프가 가장 알아보기 쉬운 눈금과 축이름을 가졌음을 알 수 있다.

## 6. 히스토그램

### 1) 공통 데이터

Index No.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa

...(생략) (표 6-1) 히스토그램 생성에 사용한 iris 데이터

### 2) graphics(R) 패키지를 이용한 시각화

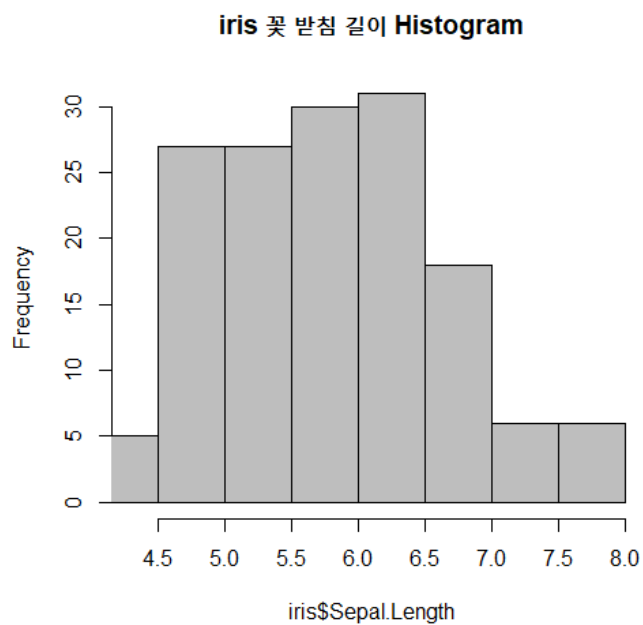


그림 6-1)

R 내장 그래픽 패키지를 통해 iris 데이터의 꽃 받침 길이를 기준으로 지정하여 히스토그램으로 시각화한 그래프이다.

### 3) ggplot2(R) 패키지를 이용한 시각화

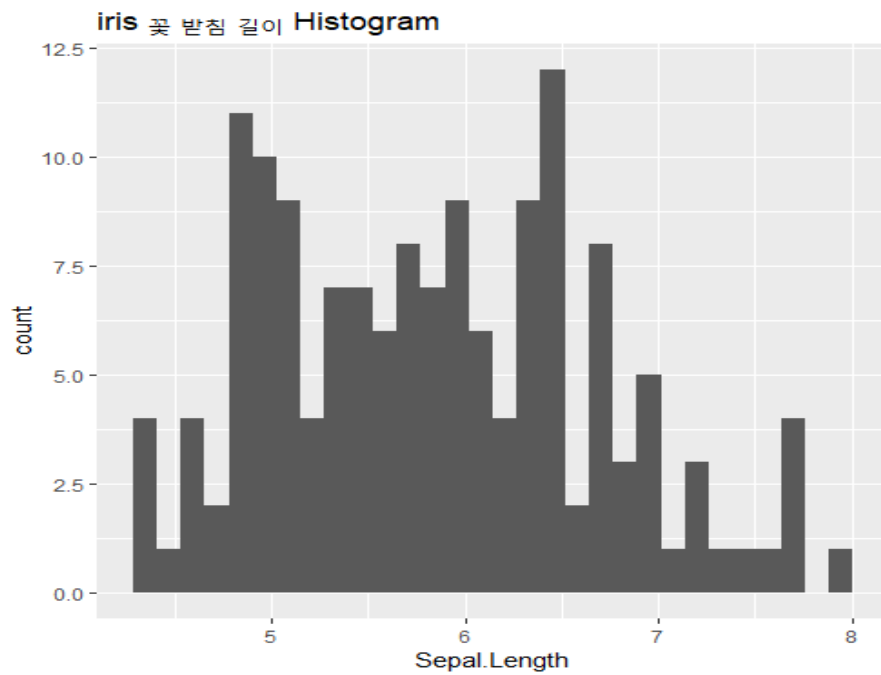


그림 6-2)

iris 데이터를 ggplot2(R) 패키지를 통해 sepal.length 를 기준으로 히스토그램을 나타낸 그래프이다.

### 4) plotly(R) 패키지를 이용한 시각화

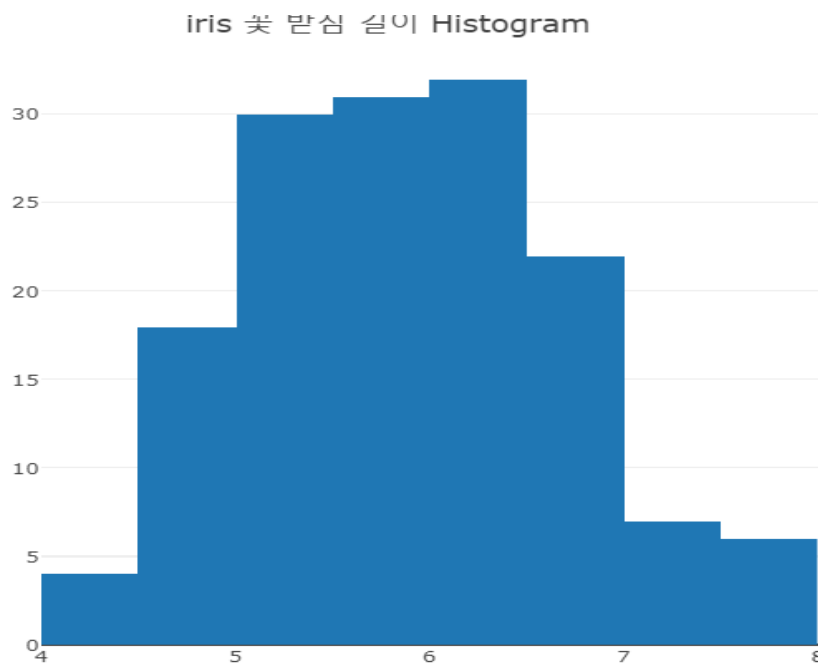


그림 6-3)

## 5) matplotlib(py) 패키지를 이용한 시각화

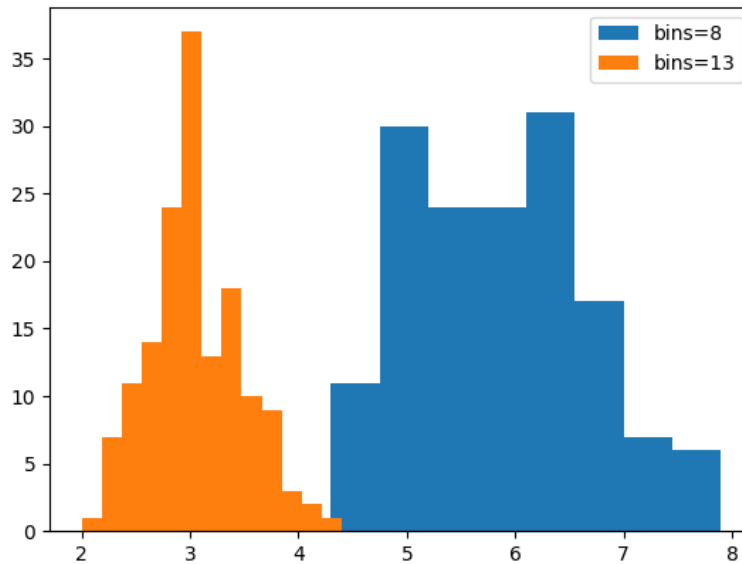


그림 6-4)

## 6) 각 패키지별 비교결과

히스토그램에서는 패키지별로 시각화 결과가 많은 차이를 보였다. 또한, 5.5~6.5 사이의 데이터가 가장 많이 나왔음을 알 수 있다..

## 7. 산점도

### 1) 공통 데이터

산점도는 다음 데이터를 기준으로 그렸다.

	price	no
1	96.241006	1
2	50.597622	2
3	34.894016	3
4	40.117647	4
5	76.598465	5
6	92.601992	6
7	25.984746	7
8	13.800023	8
9	97.143501	9
10	8.141265	10

...(생략) (표7-1) 산점도 생성에 사용한 난수 데이터

### 2) graphics(R) 패키지를 이용한 시각화

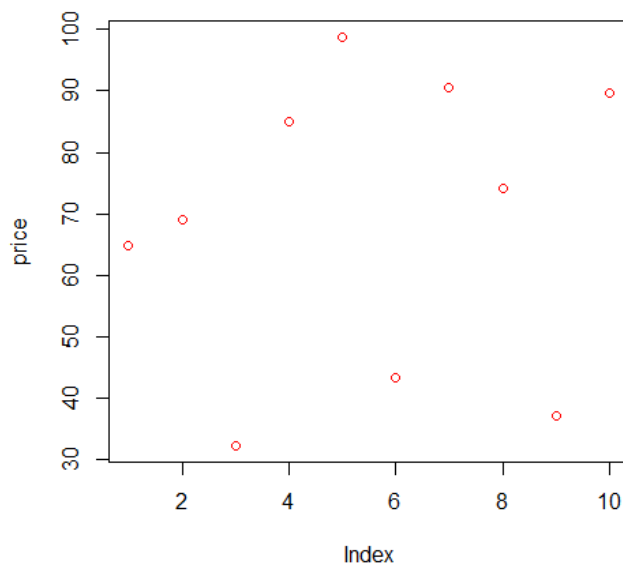


그림7-1)



### 3) ggplot2(R) 패키지를 이용한 시각화

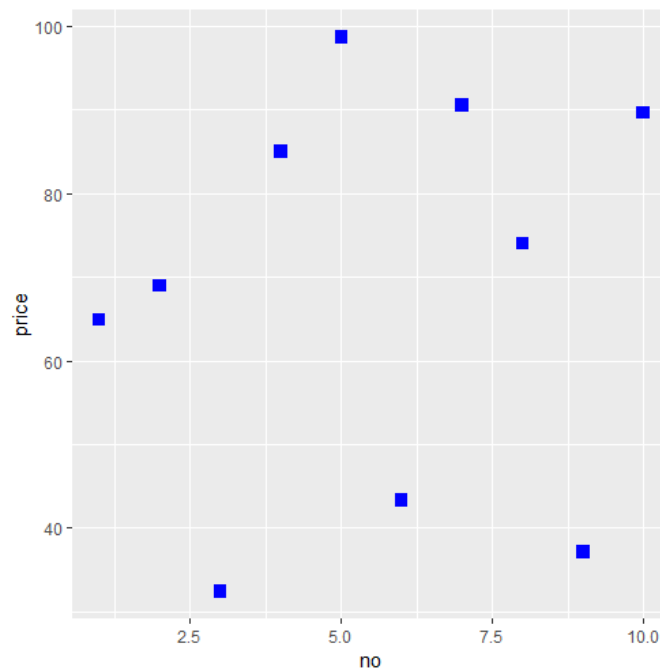


그림 1-1)

위 그래프는 ggplot2 패키지의 `geom_point()` 함수를 사용했다.

**shape = 15** : 15 번 점 모양으로 설정

**size = 3** : 3 의 크기

**colour = 'blue'** : 점 색깔을 파란색으로 변경.

### 4) plotly(R) 패키지를 이용한 시각화

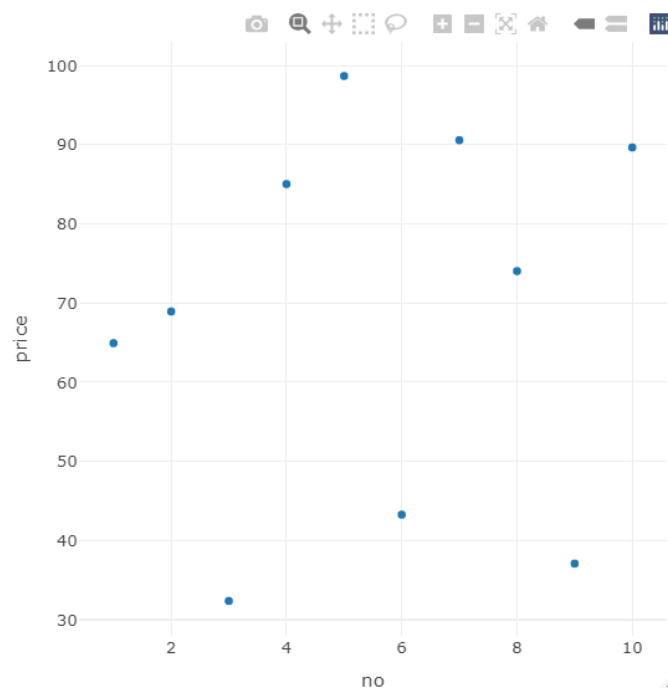


그림 1-1)

plotly 패키지의 `plot_ly()` 함수를 사용한 그래프는 `type='scatter'` 옵션을 사용하여 그렸다.

## 5) matplotlib(py) 패키지를 이용한 시각화

python 의 matplotlib 패키지에서는 `.scatter` 매소드를 사용하여 산점도 그래프를 그렸다.

ggplot2 의 layer 개념과 plotly 의 반응형을 제외하면 산점도에서는 각각의 큰 차이가 없다.

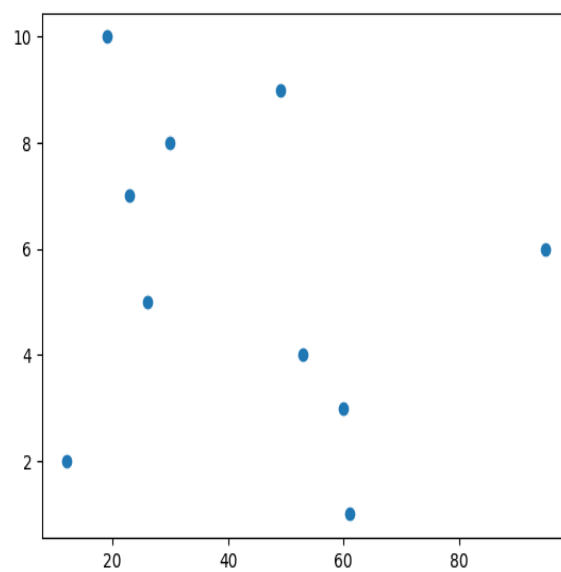


그림 1-1)

## 6) 각 패키지별 비교결과

산점도 시각화에서는 패키지별로 큰 차이가 없었다.

## 8. 중첩자료 시각화

### 1) 공통 데이터

galton	child	parent
1	61.7	70.5
2	61.7	68.5
3	61.7	65.5
4	61.7	64.5
5	61.7	64

(표8-1) 중첩자료 시각화에 사용한 galton 데이터

### 2) graphics(R) 패키지를 이용한 시각화

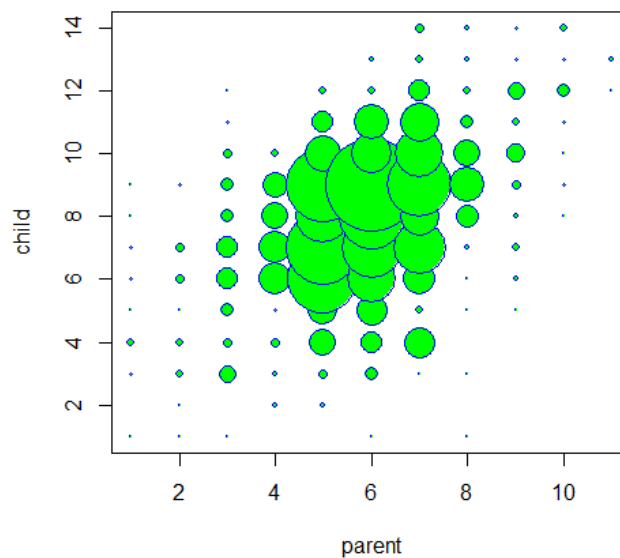


그림 8-1)

설정된 옵션은 아래와 같다.

**pch = 21** : 점의 모양을 21 번으로 설정

**col = blue** : 테두리 색상 설정

**gb = green** : 채우기 색상 설정

**cex = 0.2 \* galtondata\$freq** : 점의 크기 설정

**xlab = parent** : x 축 이름 설정

**ylab = child** : y 축 이름 설정

### 3) ggplot2(R) 패키지를 이용한 시각화

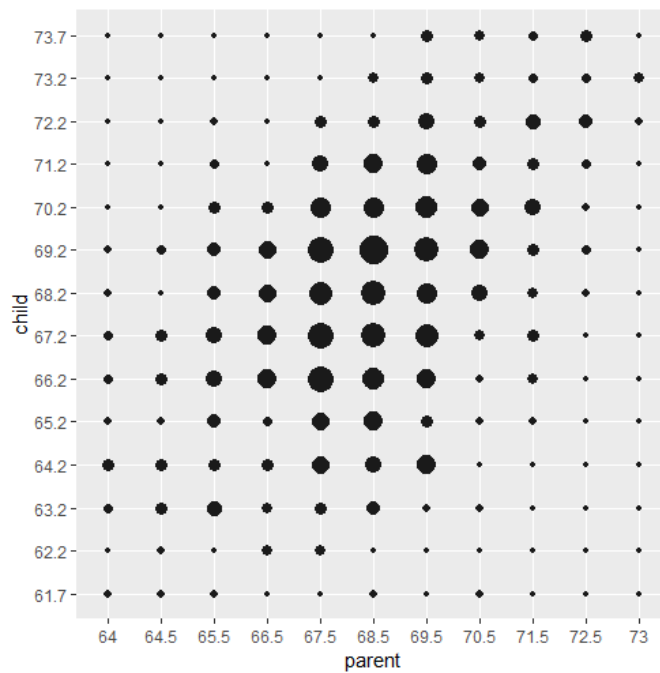


그림 8-2)

아래는 각 설정들에 대한 설명이다.

**range = c(1,7)** 점 크기의 최소&최대를 설정

**colour="grey10"** : 점의 색을 설정

**aes(size=freq)** : 빈도수에 따른 점 크기를 분배

### 4) plotly(R) 패키지를 이용한 시각화

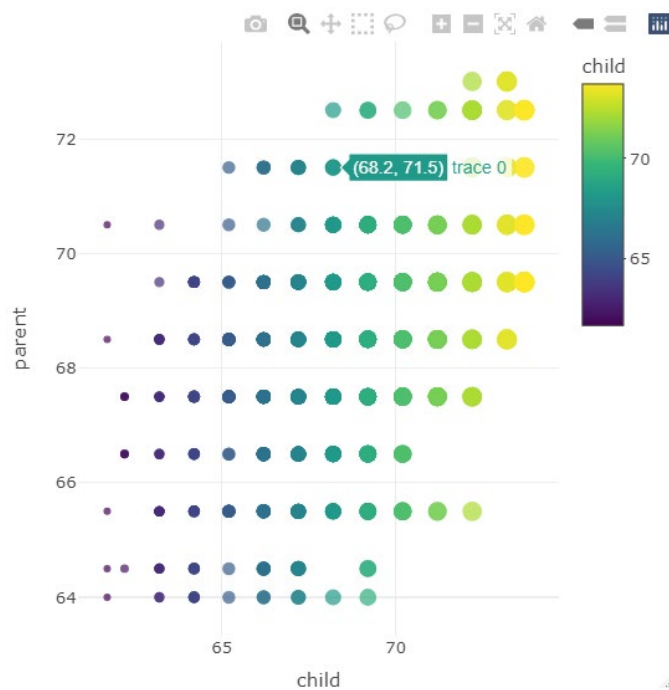


그림 8-3)

## 5) matplotlib(py) 패키지를 이용한 시각화

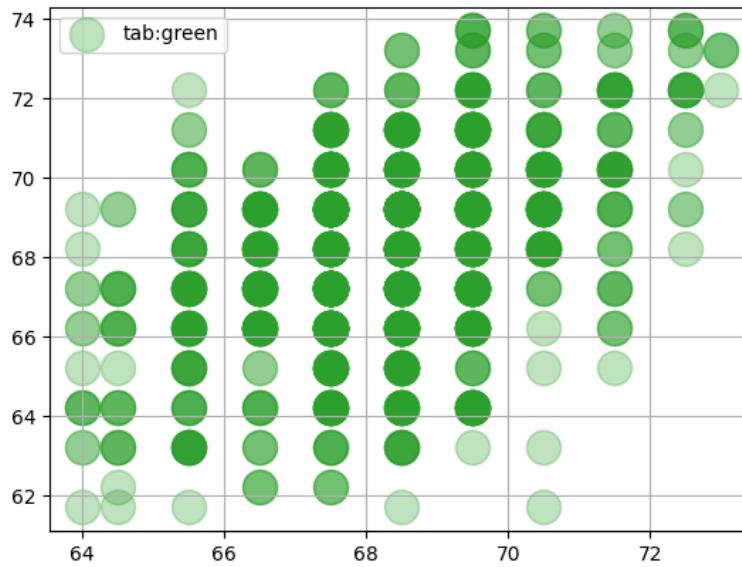


그림 8-4)

## 6) 각 패키지별 비교결과

plotly 패키지와 matplotlib 패키지는 결국 외부 패키지를 사용해야 하는 번거로움이 있지만, ggplot2는 자체적으로 밀도를 표현할 수 있다.

## 9. 변수간의 비교 시각화

### 1) 공통 데이터

Index No.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa

...(생략) (표 9-1) 변수간 비교 시각화에 사용한 iris 데이터

### 2) graphics(R) 패키지를 이용한 시각화

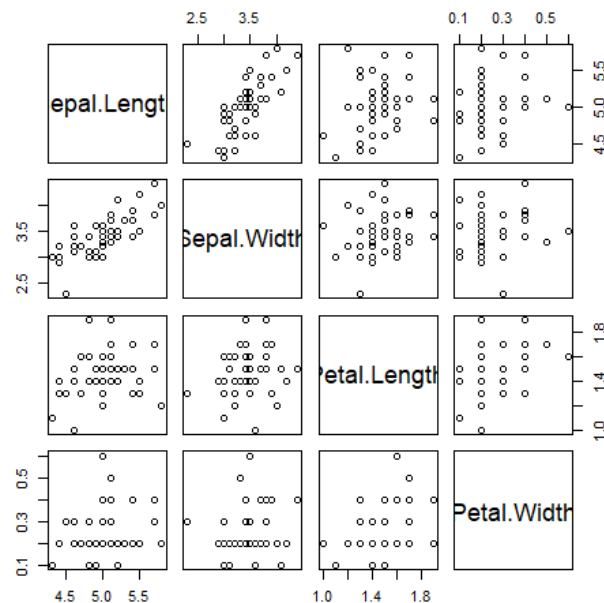


그림 9-1)

각 변수간의 산점도를 R 기본 내장 그래픽 패키지로 시각화한 그래프이다.

### 3) ggally(R) 패키지를 이용한 시각화

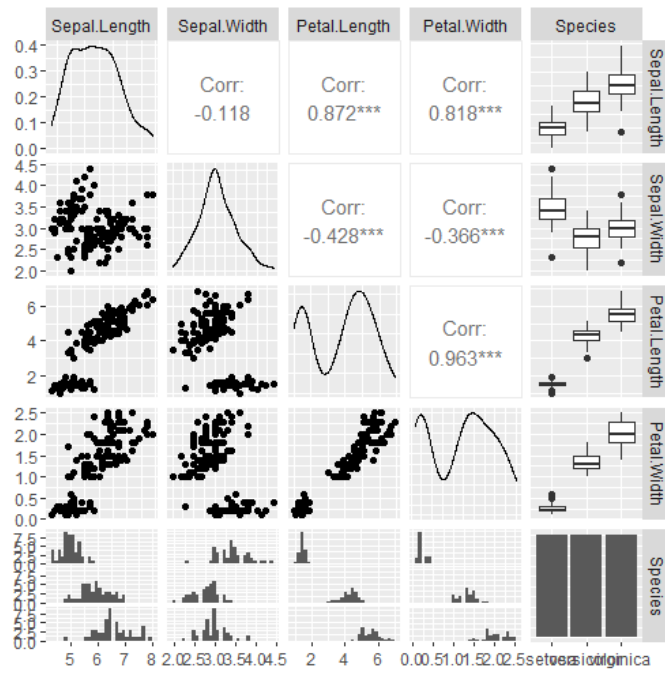


그림 9-2)

### 4) matplotlib(py) 패키지를 이용한 시각화

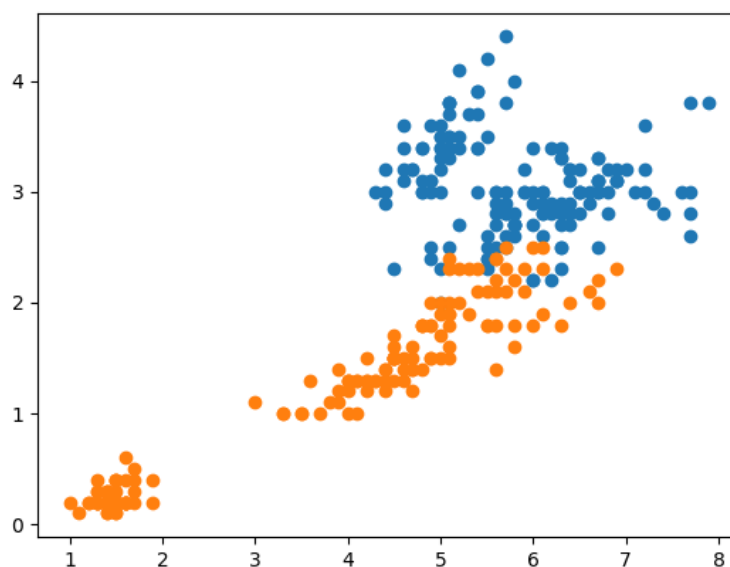


그림 9-3)

### 5) 각 패키지별 비교결과

plotly(R) 패키지에는 변수간 비교를 위한 행렬을 구해주는 함수는 없는 것 같다.

대신 ggally(R) 패키지에 있는 ggpairs() 함수를 사용하면 매우 쉽게 각 변수들을 비교할 수 있다.

## 10. 밀도 그래프

### 1) 공통 데이터

Index No.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa

...(생략) (표 10-1) 밀도 그래프 시각화에 사용한 iris 데이터

### 2) graphics(R) 패키지를 이용한 시각화

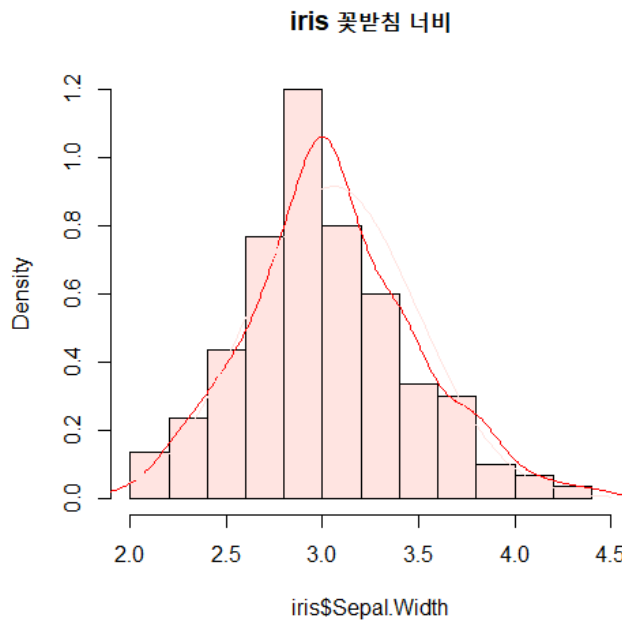


그림 10-1)

이 그래프는 R 기본 내장 그래픽 패키지에서 아래와 같은 옵션들을 적용하여 시각화 하였다.

**xlim = 2.0~4.5** : x 축 범위지정

**lines = red** : 선 색상 지정

**curve.col = mistyrose** : 그래프 채우기 색상 지정



### 3) ggplot2(R) 패키지를 이용한 시각화

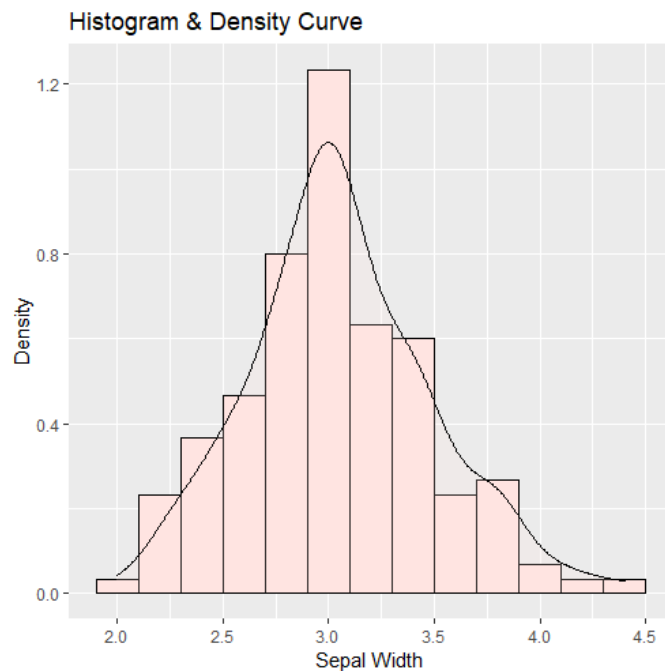


그림 10-2)

**geom\_histogram()** 함수 적용 옵션은 아래와 같다

**aes(y=..density..)** : 밀도값으로 히스토그램 생성

**geom\_density()** : 밀도선 추가

### 4) plotly(R) 패키지를 이용한 시각화

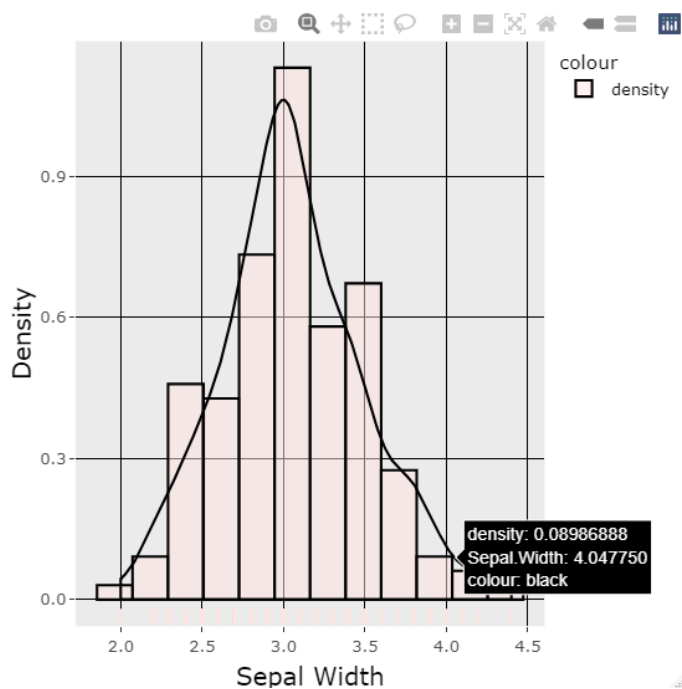


그림 10-3)

plotly 패키지에서는 ggplotly 를 사용하였는데 ggplot2 패키지와 유사하다.

## 5) matplotlib(py) 패키지를 이용한 시각화

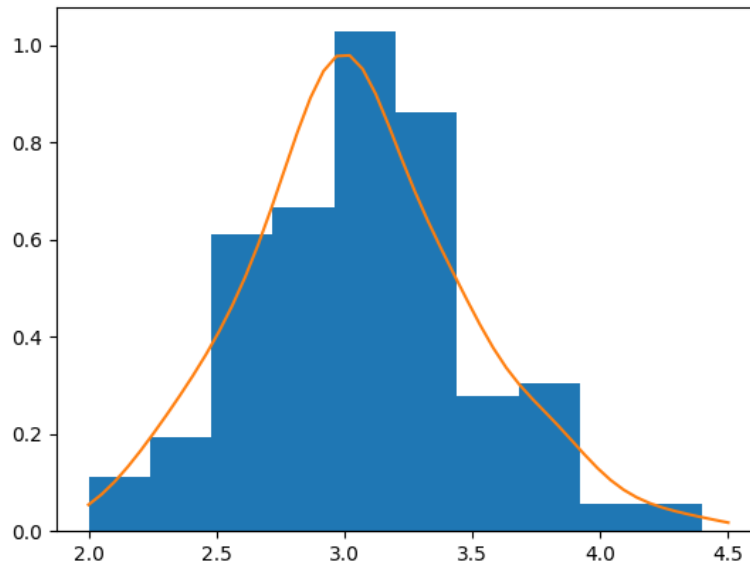


그림 10-4)

python 에서 matplotlib 패키지와 scipy 패키지를 사용했다.

scipy.stats 에서 gaussian\_kde 를 통해 밀도를 계산하여 .hist 매소드와 .plot 매소드를 사용하여 밀도 그래프를 그렸다.

## 11. 3 차원 산점도 그래프

### 1) 공통 데이터

Index No.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa

...(생략) (표 11-1) 3 차원 산점도 시각화에 사용한 iris 데이터

### 2) scatterplot3d(R) 패키지를 이용한 시각화

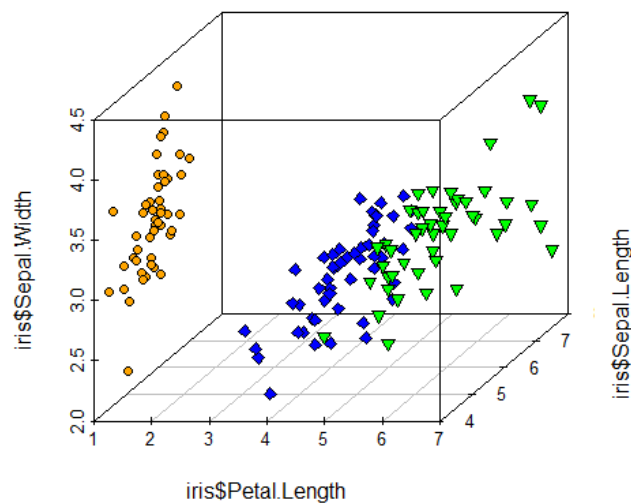


그림 11-1)

R 기본 내장 그래픽 패키지 대신 scatterplot3d 패키지를 통해 시각화한 그래프이다.

꽃의 종별로 색상을 구분하였으며 점 모양 설정은 아래와 같다

setosa : pch = 21 / versicolor : pch = 23 / virginica : pch = 25

### 3) plotly(R) 패키지를 이용한 시각화

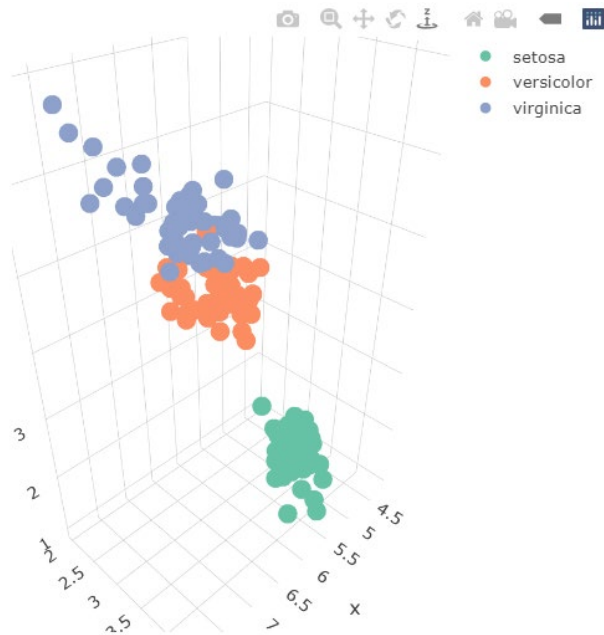


그림 11-2)

**x = Sepal Length**

**y = Sepal width**

**z = Petal Length**

**color = Species** 로 plotly 패키지를 적용하여 시각화한 그래프이다

### 4) matplotlib(py) 패키지를 이용한 시각화

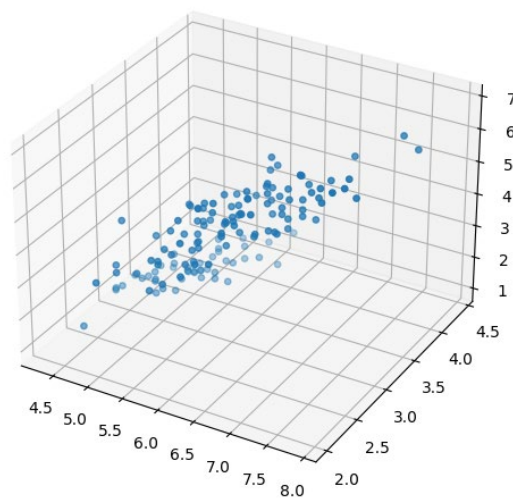


그림 11-3)

## 5) 각 패키지별 비교결과

plotly(R) 패키지의 그래프는 다소 알아보기 어려운 점의 크기와 모양을 갖고 있어서 별도의 조정이 필요해 보인다.

이 중에서도 matplotlib(Py) 패키지를 통해 시각화 한 그래프가 투명도 조정 및 격자 생성으로 제일 알아보기 쉬운 그래프로 보인다.

## 12. 지도 시각화

### 1) 공통 데이터

지역명	LAT	LON	총인구수	남자인구수	여자인구수	세대수
서울특별시	37.5651	126.9896	9,766,288	4,772,822	4,993,466	4,271,551
부산광역시	35.16277	129.0447	3,438,259	1,690,527	1,747,732	1,481,315
대구광역시	35.87975	128.5667	2,460,382	1,218,478	1,241,904	1,022,727
인천광역시	37.45622	126.7052	2,955,916	1,482,441	1,473,475	1,216,126
광주광역시	35.16008	126.8515	1,459,832	722,902	736,930	604,691
대전광역시	36.3731	127.3888	1,488,725	743,851	744,874	626,103

...(생략) (표 12-1) 지도 시각화에 사용한 인구수 데이터

### 2) ggmap(R), plotly(R) 패키지를 이용한 시각화

2019년도 1월 내안민국 인구수

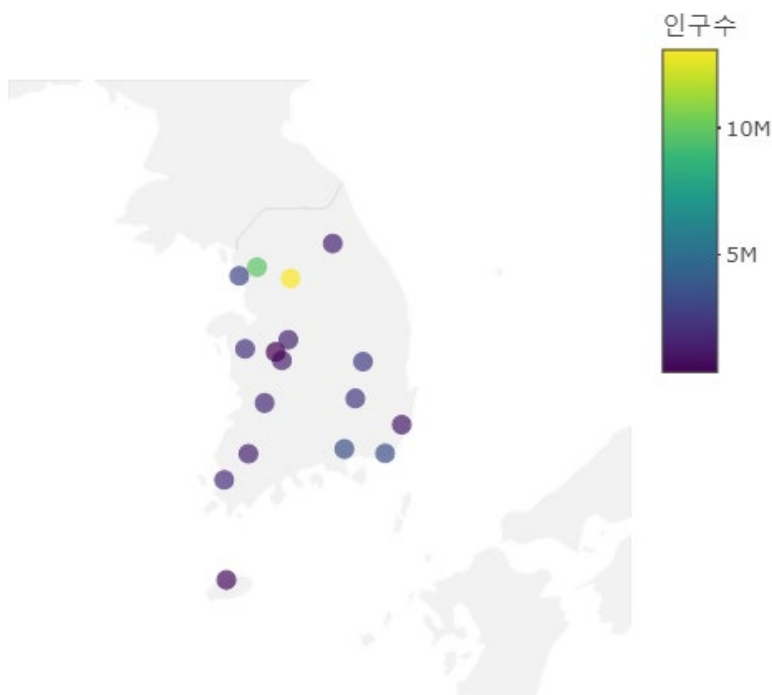


그림 12-1)

제목, 범주, 포인트의 색상 등을 별도로 지정하여 시각화한 지도이다.

### 3) 시각화 결과

matplotlib(Py) 패키지를 통한 지도시각화는 찾지 못했다.

먼저 plotly 패키지의 layout.geo 에 들어갈 옵션을 list 형식으로 나열한 gmap 객체와 gmap2 에 대한 설명이다.

#### **gmap**

**scope = 'asia'** : 아시아지역의 지도를 지정

**showland = TRUE** : 지도에서 땅의 색상을 채우기 여부 설정

**landcolor = toRGB("gray95")** : 땅의 색상을 지정

**countrycolor = toRGB("gray85"), countrywidth = 0.5** : 국경선의 색상과 두께를 지정

#### **gmap2** (gmap 을 포함)

**resolution = 50** : 지도의 해상도를 설정(값의 단위는 km/mm)

*ex) 50 은 1:50,000,000 의 축척 비율에 해당. 기본값은 110*

**list(lonaxis = list(range = c(123, 133))), list(lataxis = list(range = c(32, 40)))** :

우리나라의 경도와 위도 범위를 지정

**plot\_geo()**: 데이터값의 경도, 위도를 지정

**add\_markers()** : 지정된 위치에 symbol 의 모양, 크기, 설명 text 추가

## 결론

R 기본 내장 패키지인 `graphics` 패키지는 매우 쉬운 방법으로 그래프를 시각화 할 수 있었다.

`ggplot` 패키지는 미적요소를 추가/변경 하는 것이 간편했다.

`plotly` 패키지는 마우스를 갖다대면 관련 정보가 나오는 옵션이 있어서 매우 유용했다.

`matplotlib` 패키지 또한 준수한 성능을 보였다.

결론적으로, 기본적인 성능은 크게 다르지 않았기 때문에 본인이 잘 다룰 수 있고 원하는 옵션과 기능이 있는 패키지를 선택하여 시각화하는 능력이 중요했다.



## 부록

전체 코드

파이썬: Project6\_Visualize(py)

R: Project6\_Visualize

데이터 셋

galton.csv

iris.csv

VADeaths.csv

population201901.csv

사진

output 폴더 참조