
빅데이터 분석 결과 시각화

— 이상언 —

1. 패키지 특징과 주요 함수

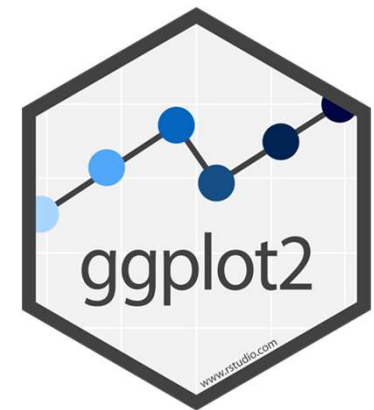
2

1. ggplot2 패키지

데이터를 미적으로 시각화 할때 사용하는 대중적인 시각화 패키지 이다.

ggplot2는 The Grammar of Graphics를 기반으로 그래픽을 선언적으로 생성하는 시스템입니다 . 데이터를 제공하고 ggplot2에 변수를 미학에 매핑하는 방법, 사용할 그래픽 기본 요소를 알려주고 세부 사항을 처리합니다.

주요함수	설명
<code>aes_colour_fill_alpha</code>	색상 관련 미학 : 색상, 채우기 및 알파
<code>aes</code>	미적 매핑 구성
<code>aes_all</code>	문자형 벡터가 주어지면 ID 매핑 집합을 만듦
<code>aes_group_order</code>	미학 : 그룹화
<code>aes_linetype_size_shape</code>	미학 관련 미학 : 선 종류, 크기, 모양
<code>annotation_map</code>	주석 :지도
<code>discrete_scale</code>	이산 스케일 생성자
<code>guides</code>	스케일 별 가이드 설정



[가이드 링크](#) 

1. 패키지 특징과 주요 함수

2. plotly 패키지

반응형 시각화로, 기존의 시각화보다 더 효율적으로 데이터를 파악할 수 있다.

오픈 소스 JavaScript 그래프 라이브러리 [plotly.js](#) 를 통해 대화 형 웹 그래픽을 생성하기위한 R 패키지입니다 .

주요함수	설명
add_trace	플롯 시각화에 트레이스 추가
hide_colorbar	색상 막대 숨기기
add_data	플롯 시각화에 데이터 추가
bbox	회전 된 문자열의 경계 상자 추정
plotly_build	플롯 객체 '빌드'
EVENT_DATA	반쪽에 액세스 사용자 입력 이벤트 데이터
plot_mapbox	플롯 맵 박스 객체를 시작
to_basic	도형을 "기본" 도형으로 변환



[가이드 링크](#) 

1. 패키지 특징과 주요 함수

4

3. esquisse 패키지

builder 기능으로 손쉽고 효과적으로 정보를 빠르게 시각화 할 수 있다.

데이터를 빠르게 탐색하여 보유한 정보를 추출할 수 있도록 하는 패키지 이다. [ggplot2]로 시각화 하고 [dplyr] 필터링 하고 생성된 코드를 검색할 수 있다.

주요함수	설명
ggplot_to_ppt	ggplot 개체를 PowerPoint로 내보내는 유틸리티
ggplotly	ggplot2 구문을 사용하여 플롯 그래프 생성
build_aes	플롯에서 사용할 미학 빌드
potential_geoms	데이터에 따른 잠재적 기하학
esquisser	ggplot2 로 쉽게 플롯을 생성 할 수있는 추가 기능
which_pal_scale	적절한 컬러 스케일 자동 선택
match_geom_args	인수 목록을 기하학의 인수와 일치
dragulaInput	드래그 앤 드롭 입력 위젯



[가이드 링크](#) 

1. 패키지 특징과 주요 함수

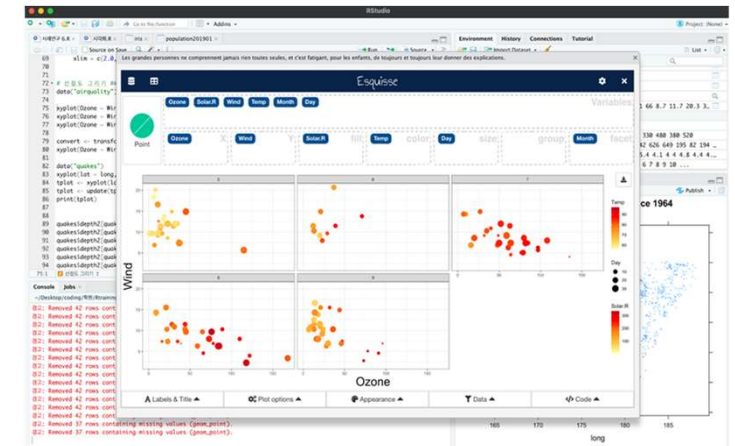
5

4. lattice 패키지

직교형태의 그래픽 툴 (막대그래프, 산점도, 선그래프, 상자그림 등..) 시각화를 돕는다.

Lattice는 Trellis 그래픽에서 영감을받은 강력하고 우아한 R 용 고급 데이터 시각화 시스템이다. 이는 다변량 데이터에 중점을두고 설계되었으며 특히 "작은 다중"플롯을 생성한다. 격자는 일반적인 그래픽 요구 사항에 충분하며 대부분의 비표준 요구 사항을 처리 할 수있을만큼 유연하다.

주요함수	설명
histogram	히스토그램 및 커널 밀도 플롯
cloud	3d 산점도 및 와이어 프레임 표면도
qqmath	이론적 분포가있는 QQ 플롯
xyplot.ts	시계열 플로팅 방법
update.trellis	격자 개체 검색 및 업데이트
make.groups	여러 벡터의 그룹화 된 데이터
panel.functions	유용한 패널 기능 구성 요소



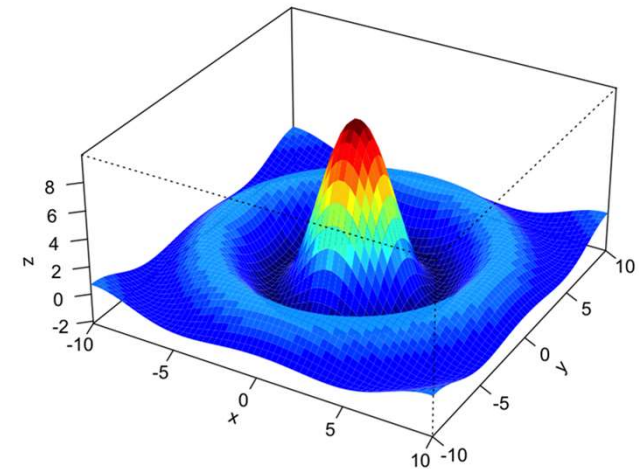
가이드 링크 

5. RGL 패키지

R에서 OPEN GL을 사용할 수 있는 패키지가 있습니다. OPEN GL은 open graphics library 의 줄임말이다. 2D, 3D 렌더링을 해 주는 라이브러리이다. 즉, 3D 시각화 장치 시스템

대화 형 관점 탐색 기능 (마우스 + 휠 지원) 및 R 프로그래밍 인터페이스를 제공합니다. 반면에 WebGL은 웹 브라우저에서 렌더링됩니다. rgl은 입력 파일을 생성하고 브라우저는 이미지를 표시합니다.

주요함수	설명
plot3d	3D 산점도
arrow3d	화살표 그리기
grid3d	3D 플롯에 그리드 추가
arc3d	호 그리기
ellipse3d	타원체 만들기
points3d	기본 모양 추가
light	광원 추가



[가이드 링크](#) 

1. 패키지 특징과 주요 함수

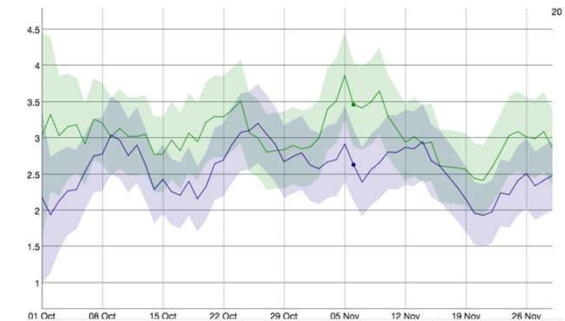
7

6. Dygraphs 패키지

빠르고 유연한 오픈 소스 JavaScript 차트 라이브러리이다.

R에서 시계열 데이터를 차트로 작성하는 데 다음과 같은 풍부한 기능을 제공합니다.

주요함수	설명
ygraph	시계열 데이터에 대한 dygraph 대화 형 플롯
dyDataHandler	dygraph 데이터 핸들러 포함
dygraph	시계열 데이터에 대한 dygraph 대화 형 플롯
dyLimit	다이 그래프 한계선
dyPlugin	dygraph 플러그인 포함
dyRebase	Dygraph를 사용 하여 밀집 빗자루 차트 용 데이터 핸들러 리베이스
dyCSS	dygraph 레이블 및 범례 용 CSS



[가이드 링크](#) 

1. 패키지 특징과 주요 함수

8

7. googleVis 패키지

구글이 만든 무빙 그래프(설정에 의해서 움직이는 그래프), 인터랙티브 그래프(마우스 움직임에 반응해 실시간으로 형태가 변하는 그래프) R에서 시계열 데이터를 차트로 작성하는 데 다음과 같은 풍부한 기능을 제공한다.

주요함수	설명
<code>gvisGauge</code>	계기판
<code>gvisPieChart</code>	원차트
<code>gvisMap</code>	구글 지도



출처 링크

1. 패키지 특징과 주요 함수

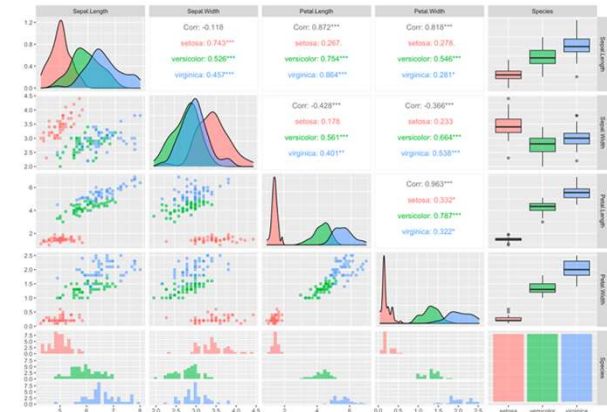
9

8. GGally 패키지

그래픽 문법의 R 플로팅 시스템이다. ggplot2의 복잡함을 줄인 확장형이다.

쌍대 그림 행렬, 산점도 그림행렬, 평행 좌표 그림, 생존 그림 및 네트워크 그리는 여러 함수 가 포함 되어 있다..

주요함수	설명
ggcorr	상관관계 시각화
ggally_points	산점도 플로팅
skewness:	왜도 플로팅
ggts	다중 시계열
ggally_cor	산점도의 상관 관계
ggally_nostic_se_fit	ggnostic-적합치 표준 오차



[가이드 링크](#) 

2. 교재 데이터 13 가지 시각화

10

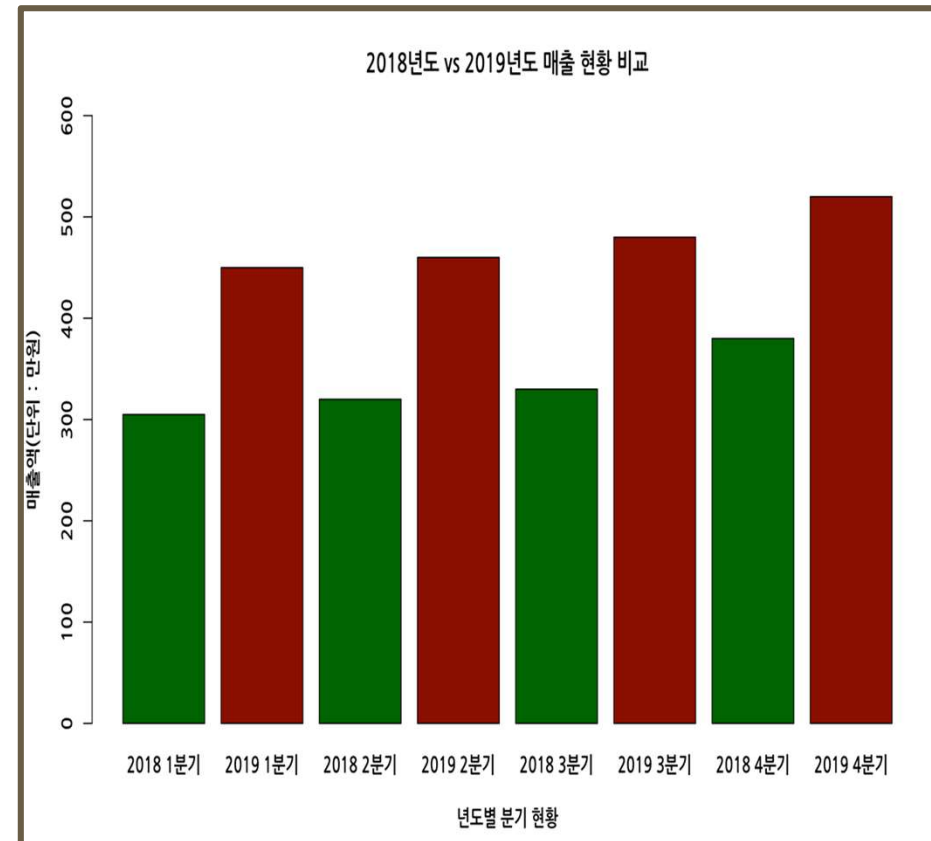
1. 막대 차트 - 세로

패키지 : graphics (기본 내장 함수)

R 코드 : `barplot(chart_data, xlim = c(0, 600),
col = rep(c('darkgreen','darkred'),4),
main = "2018 vs 2019 매출 현황 비교",
family="NanumGothicBold",
ylab = "매출액", xlab = "년도별 분기 현황")`

데이터 :

	chart.data
2018년 1분기	305
2019년 1분기	450
2018년 2분기	320
2019년 2분기	460
2018년 3분기	330
2019년 3분기	480
2018년 4분기	380
2019년 4분기	520



2. 교재 데이터 13 가지 시각화

11

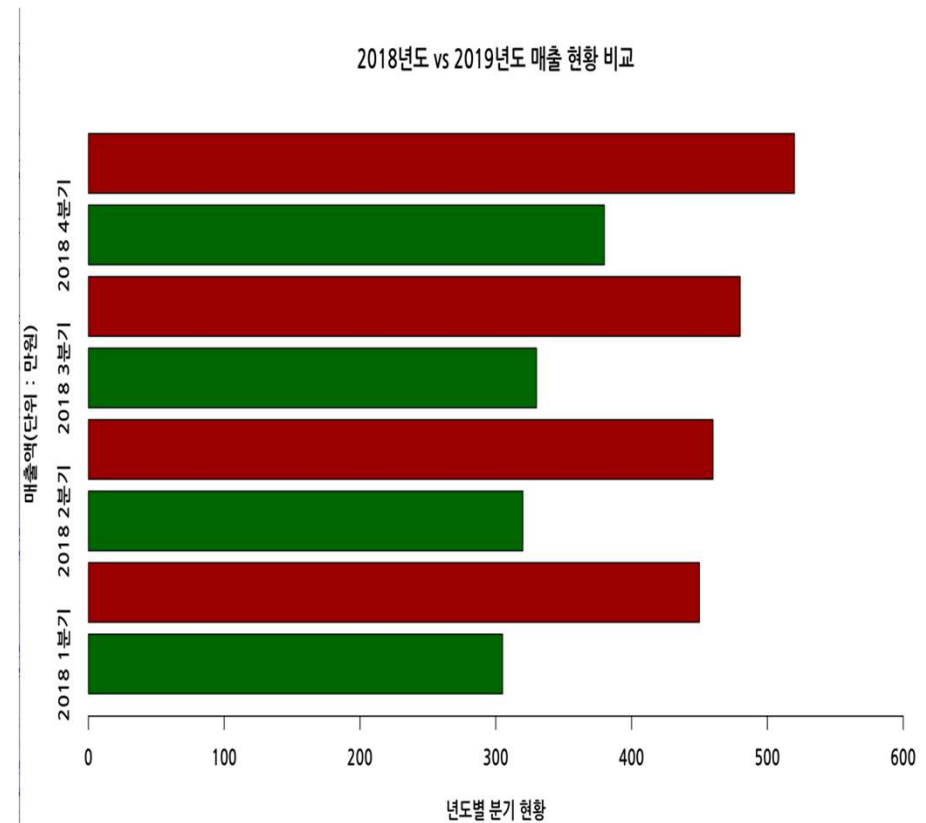
2. 막대차트 - 가로

패키지 : graphics (기본 내장 함수)

R 코드 : `barplot(chart_data, xlim = c(0, 600),
col = rep(c('darkgreen','darkred'),4),
main = "2018 vs 2019 매출 현황 비교",
family="NanumGothicBold", horiz = T,
ylab = "매출액", xlab = "년도별 분기 현황")`

데이터 :

	chart.data
2018년 1분기	305
2019년 1분기	450
2018년 2분기	320
2019년 2분기	460
2018년 3분기	330
2019년 3분기	480
2018년 4분기	380
2019년 4분기	520



2. 교재 데이터 13 가지 시각화

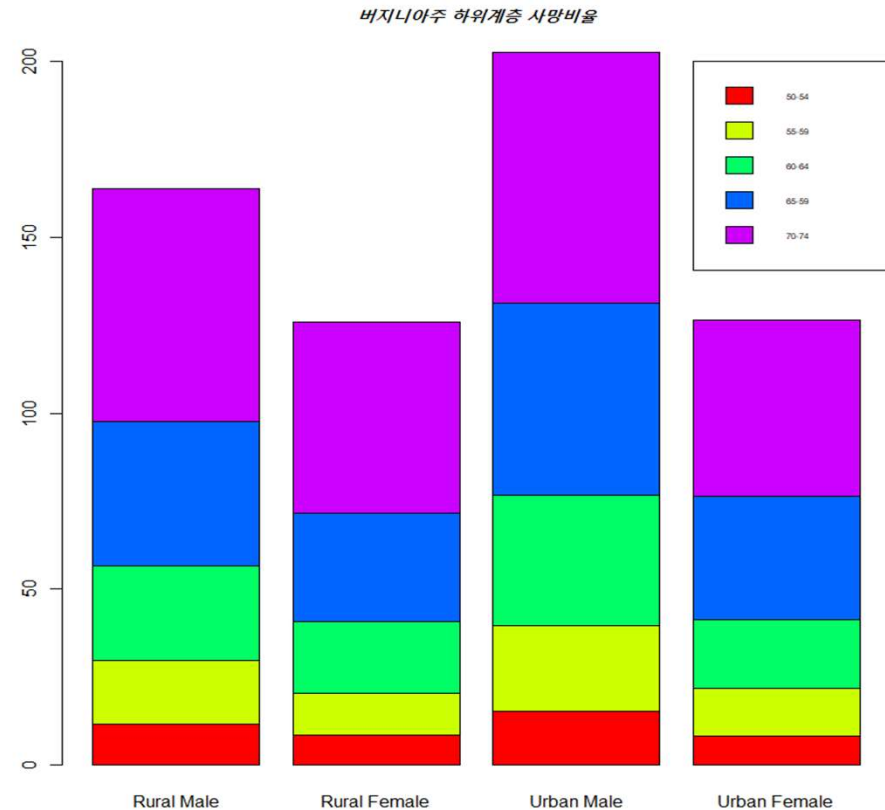
12

3. 누적막대 차트 (barplot)

패키지 : graphics (기본 내장 함수)

R 코드 : `barplot(VADeaths, beside = F,`
`col = rainbow(5),`
`main = "버지니아주 하위계층 사망비율",`
`font.main = 4)`

데이터 : VADeaths (R 기본 데이터 셋)



2. 교재 데이터 13 가지 시각화

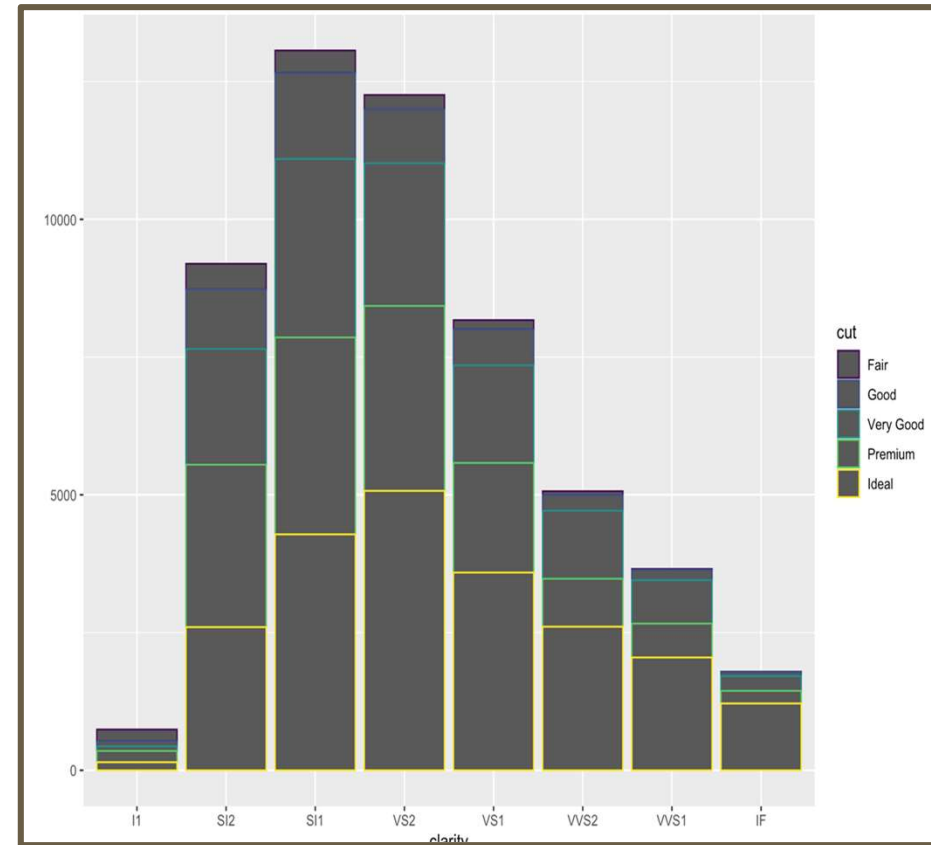
13

3. 누적막대 차트 (qplot)

패키지 : ggplot2

R 코드 : `qplot(clarity, data = diamonds,
 colour = cut, geom = 'bar')`

데이터 : diamonds (ggplot2 내장 데이터)



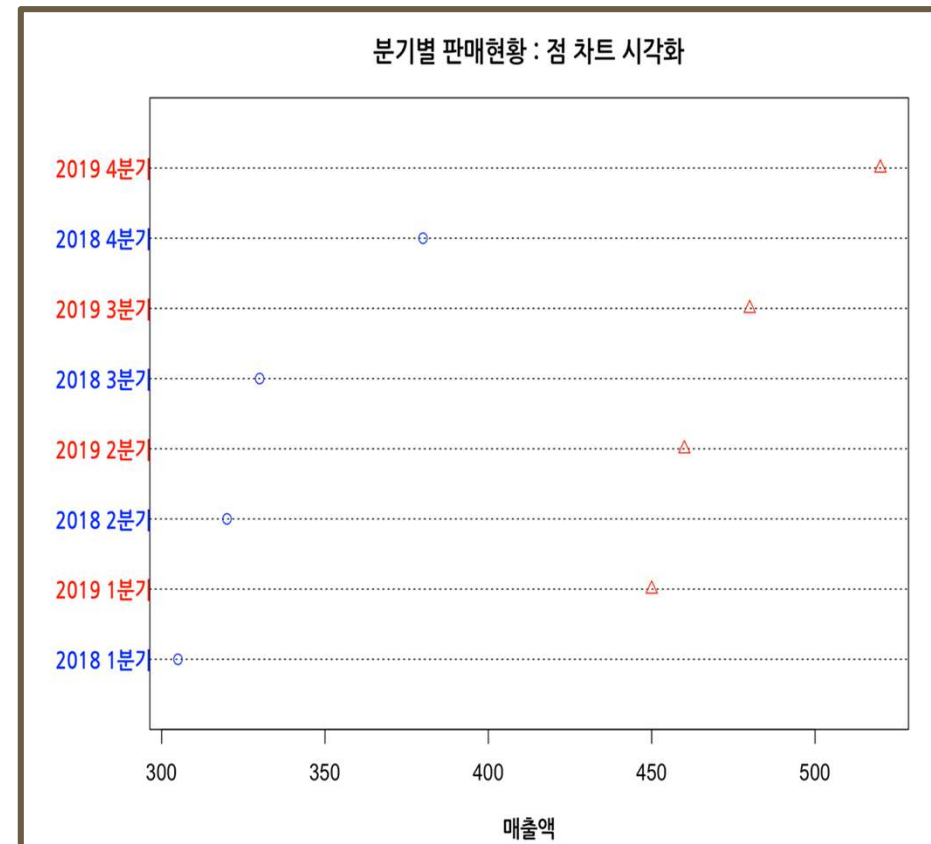
4. 점 차트

패키지 : graphics (기본 내장 함수)

R 코드 : `dotchart(chart.data, color = c("blue", "red"),
lcolor = "black", pch = 1:2,
labels = names(chart.data), xlab = "매출액",
main = "분기별 판매현황: 점차트 시각화",
cex = 1.2)`

데이터 :

	chart.data
2018년 1분기	305
2019년 1분기	450
2018년 2분기	320
2019년 2분기	460
2018년 3분기	330
2019년 3분기	480
2018년 4분기	380
2019년 4분기	520



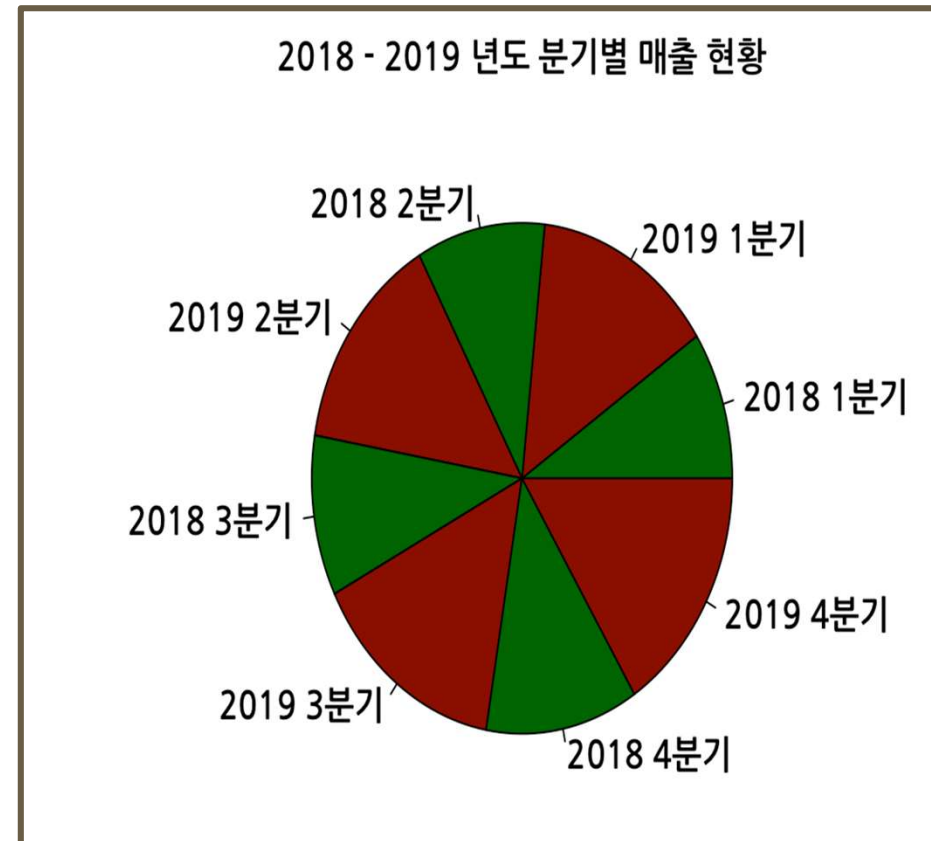
5. 원형 차트

패키지 : graphics (기본 내장 함수)

R 코드 : `pie(chart.data,
labels = names(chart.data),
col = rep(c('darkgreen','darkred'),4)
title("2018-2019 분기별 매출현황")`

데이터 :

	chart.data
2018년 1분기	305
2019년 1분기	450
2018년 2분기	320
2019년 2분기	460
2018년 3분기	330
2019년 3분기	480
2018년 4분기	380
2019년 4분기	520

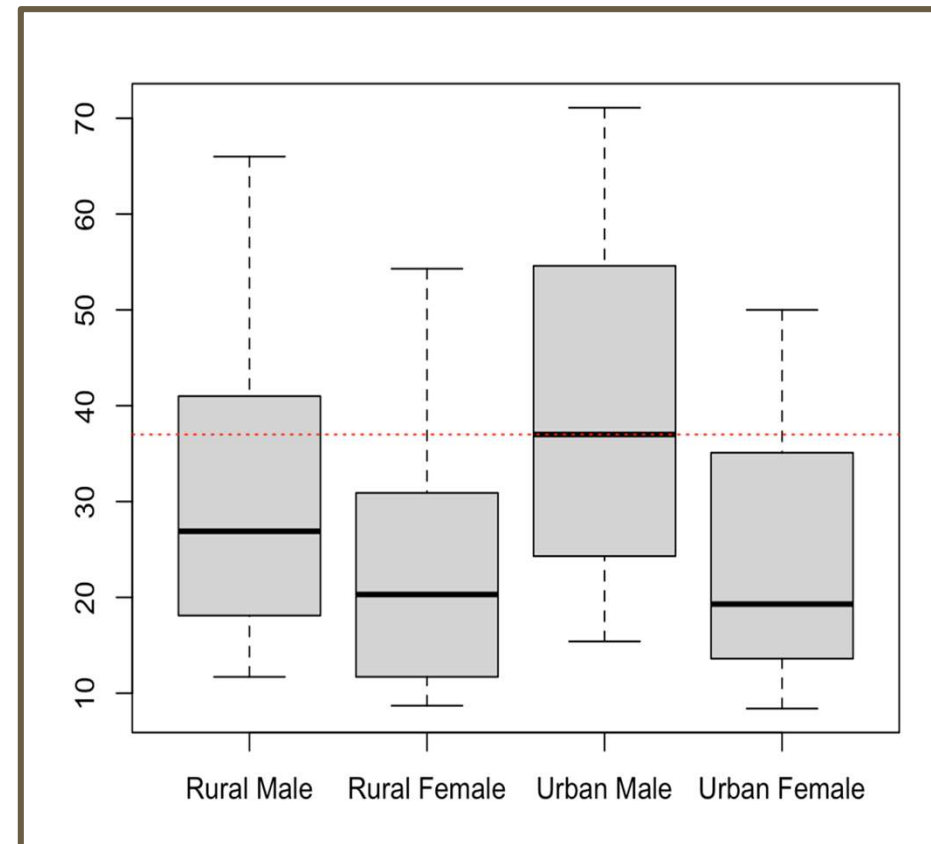


6. 상자 그래프

패키지 : graphics (기본 내장 함수)

R 코드 : `boxplot(VADeaths, range = 0)`
`abline(h = 37, lty = 1, col = "red")`

데이터 : VADeaths (R 기본 데이터 셋)



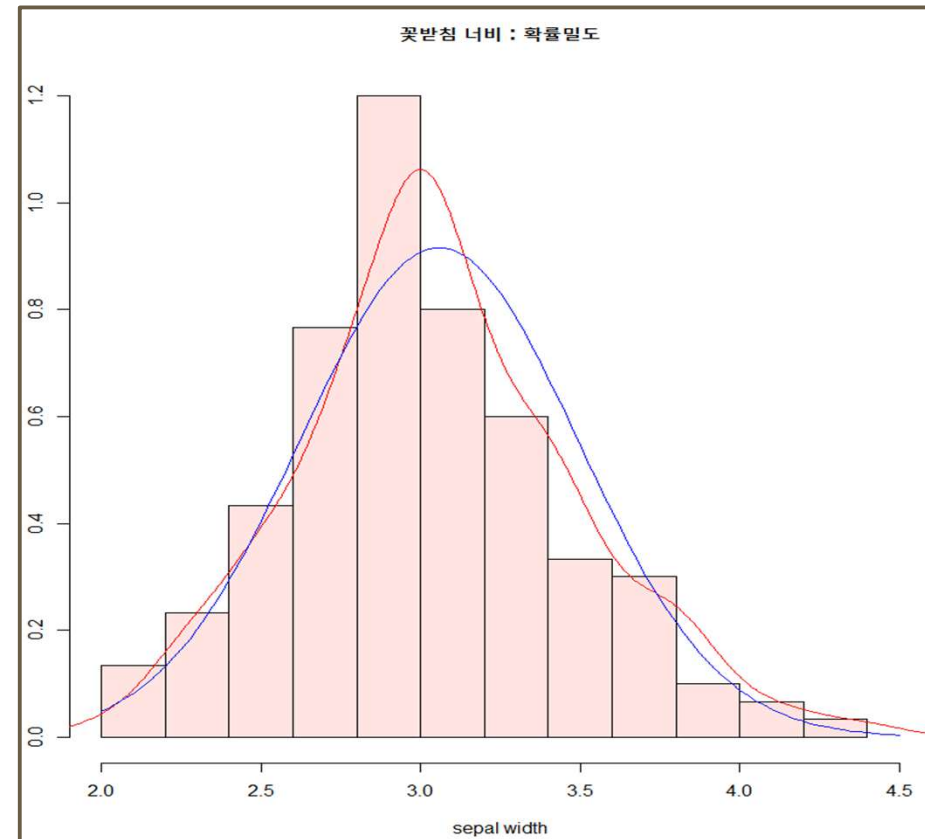
7. 히스토그램

패키지 : graphics (기본 내장 함수)

R 코드 : `hist(iris$Sepal.Width, xlab = "sepal width",
col = "mistyrose", freq = F,
main = "꽃받침 너비: 확률밀도",
xlim = c(2, 4.5)) # 확률밀도로 그리기`

`lines(density(iris$Sepal.Width), col = "red")`
밀도 선추가

데이터 : iris (R 기본 데이터 셋)

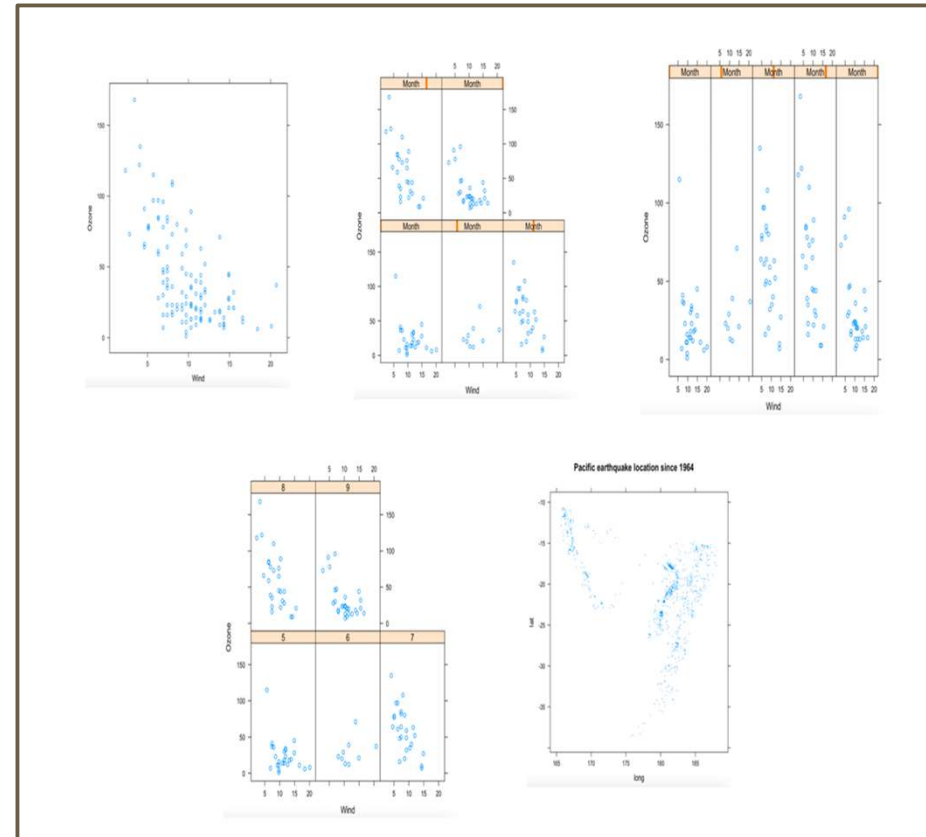


8. 산점도

패키지 : lattice

R 코드 : `xyplot(Ozone ~ Wind, data = airquality)`
`xyplot(Ozone ~ Wind | Month, data = airquality)`
`xyplot(Ozone ~ Wind | Month, data = airquality,`
`layout = c(5, 1))`
`xyplot(Ozone ~ Wind | Month, data = convert)`

데이터 : airquality (R 기본 데이터 셋)



2. 교재 데이터 13 가지 시각화

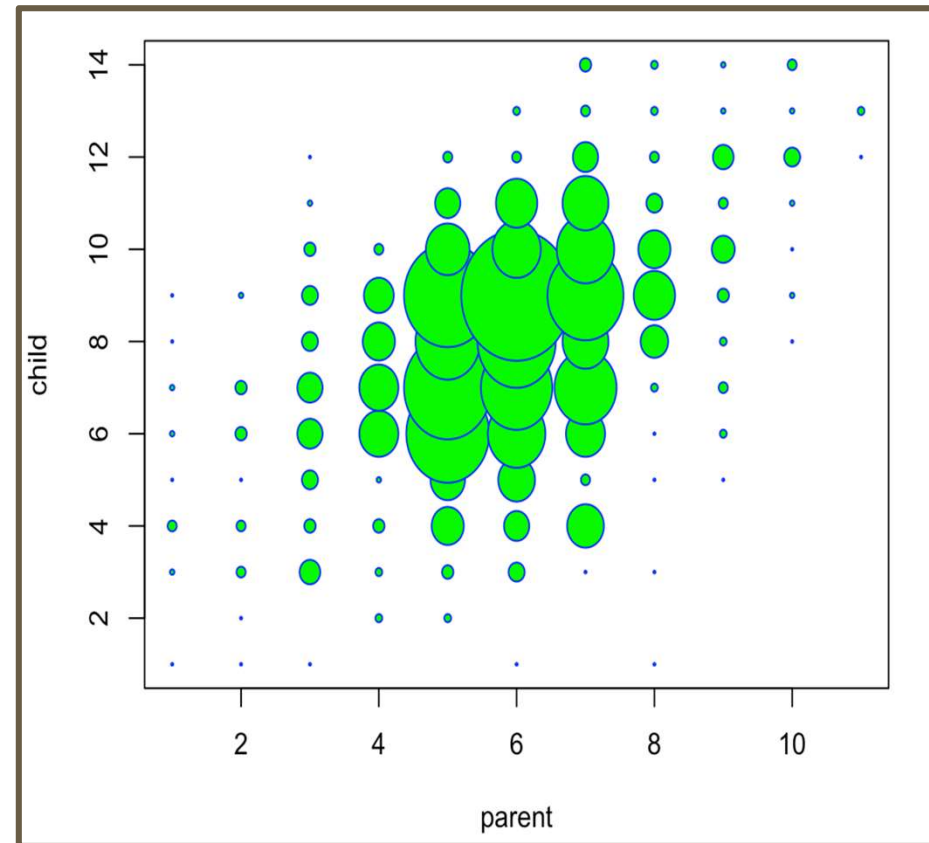
19

9. 중첩자료

패키지 : graphics (기본 내장 함수)

R 코드 : `plot(parent, child, pch = 21, col = "blue",
bg = "green", cex = 0.2 * galton.data$freq)`

데이터 : galton (UsingR)

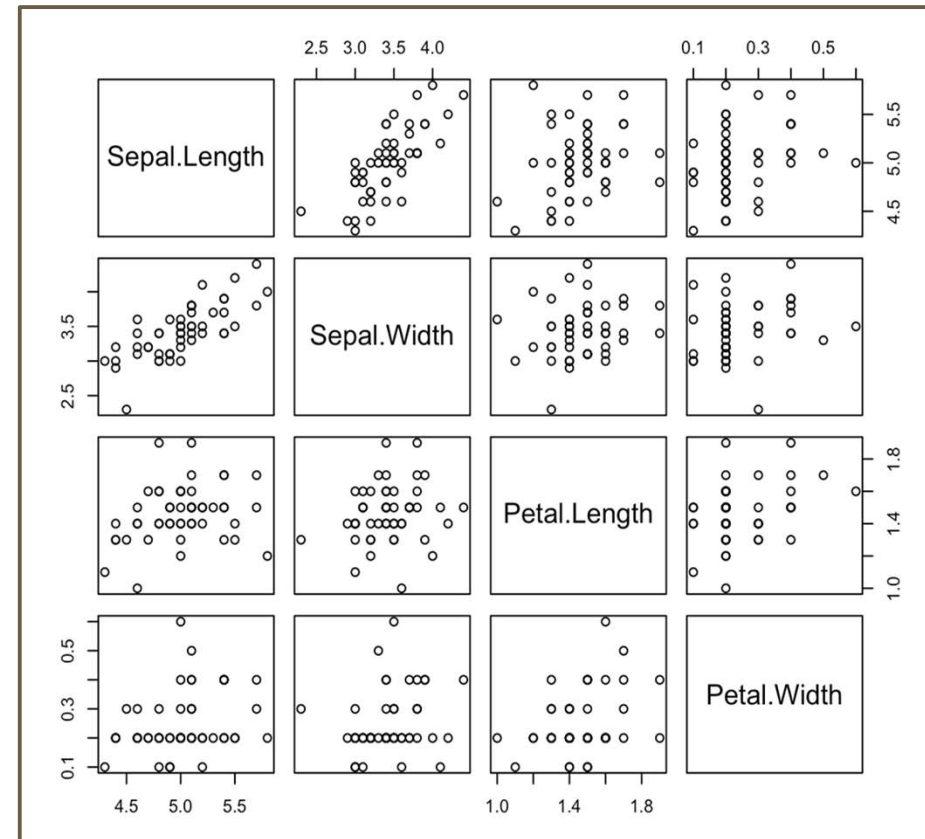


10. 변수간의 비교

패키지 : graphics (기본 내장 함수)

R 코드 : `pairs(iris[iris$Species == "setosa", 1:4])`

데이터 : iris (R 기본 데이터 셋)



2. 교재 데이터 13 가지 시각화

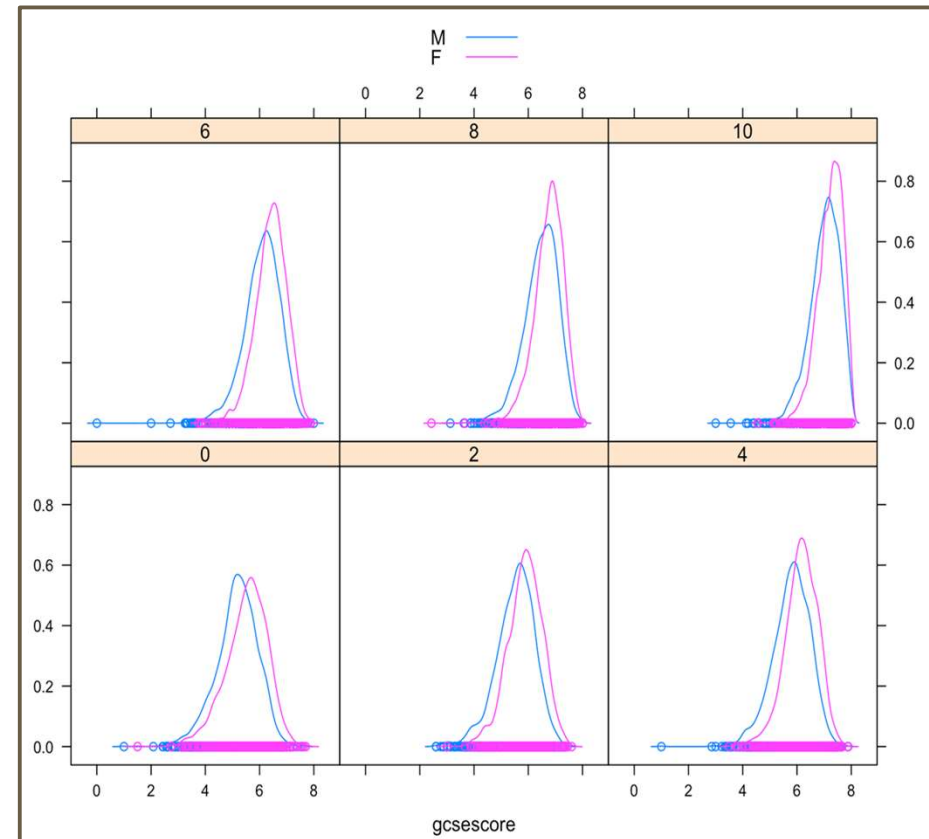
21

11. 밀도그래프

패키지 : lattice

R 코드 : `densityplot(~ gcsescore | factor(score),
data = Chem97,
groups = gender, plot.points = T,
auto.key = T)`

데이터 : Chem97 (mlmRev)



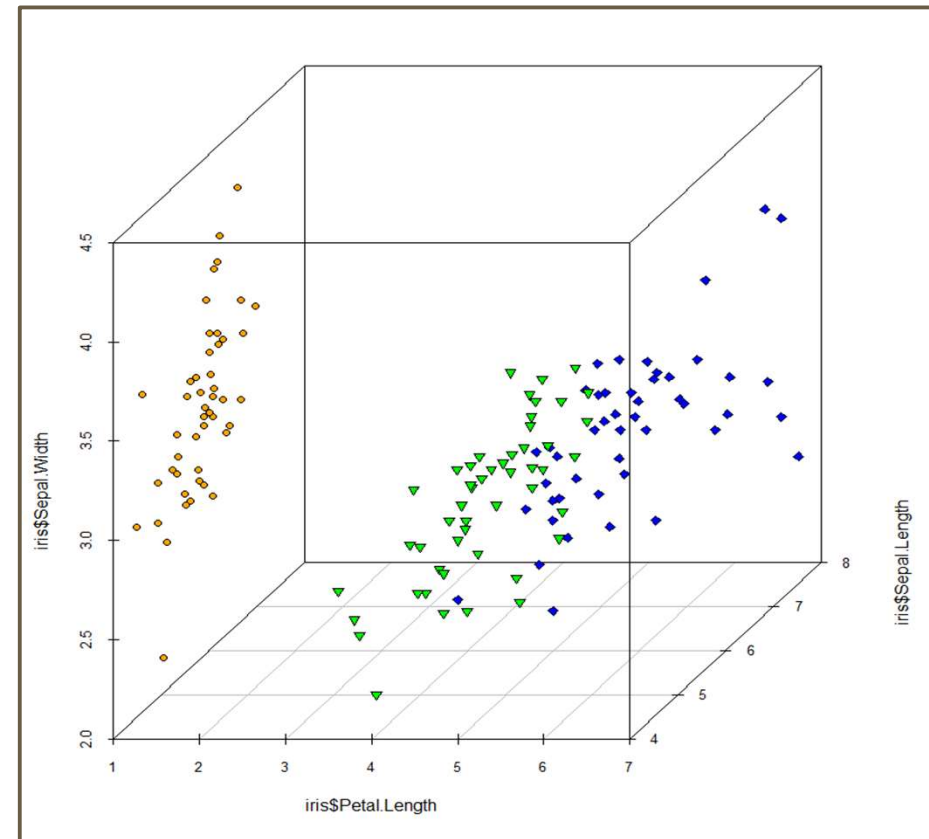
12. 3차원 산점도

패키지 : iris (R 기본 데이터 셋)

R 코드 : `d3 <- scatterplot3d(iris$Petal.Length,
iris$Sepal.Length,
iris$Sepal.Width,
type = "n")
d3$points3d(iris.setosa$Petal.Length,
iris.setosa$Sepal.Length,
iris.setosa$Sepal.Width,
bg = "orange", pch = 21)`

.... 중간 생략

데이터 : iris (R 기본 데이터 셋)



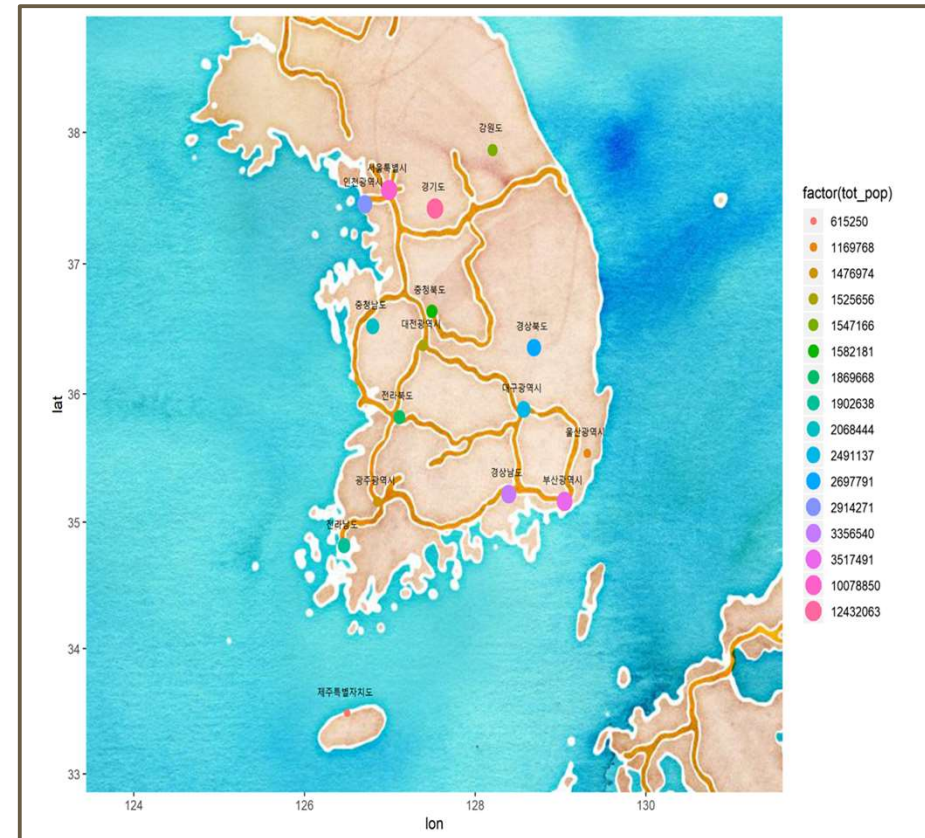
13. 지도

패키지 : ggmap

R 코드 :

```
layer1 <- ggmap(daegu.map)
+ geom_point(data = tot.pop.df,
  aes(x = lon, y = lat,
    color = factor(tot.pop),
    size = factor(tot.pop)))
+ geom_text(data = tot.pop.df,
  aes(x = lon + 0.01, y = lat + 0.05,
    label = region), size = 3)
```

데이터 : [population201901](#) (링크 참조)



3. 교재 데이터 시각화 비교

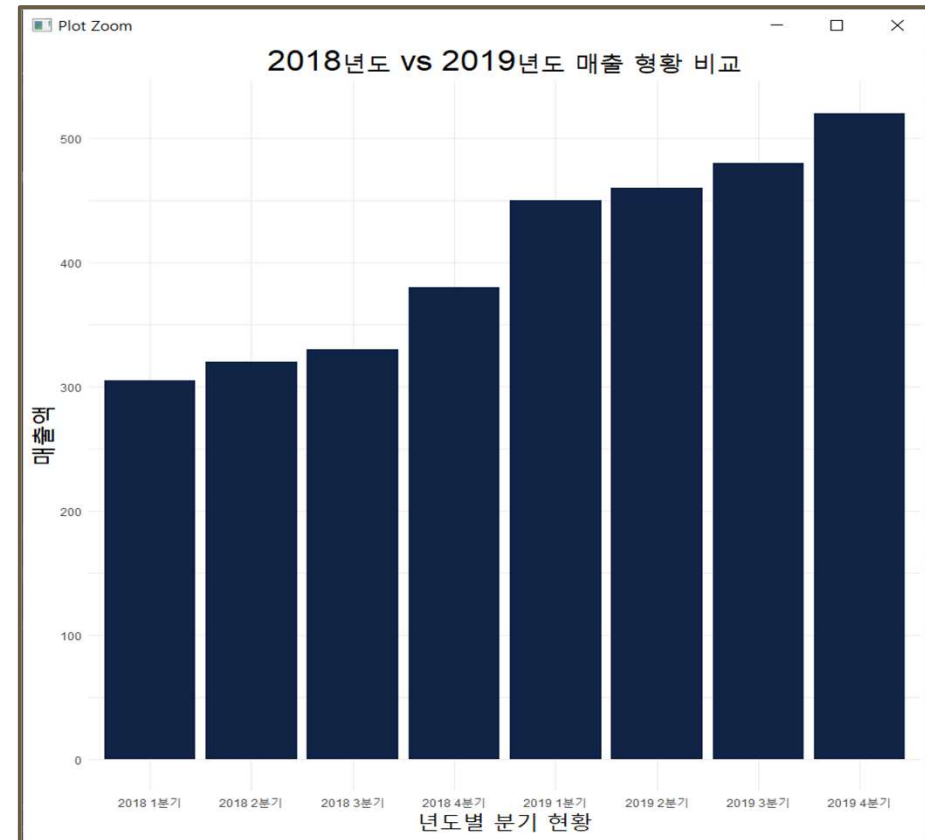
1. 막대 그래프 - 세로

패키지 : `esquisse + ggplot2`

함수 : `ggplot()`

데이터 : 사용자 정의 데이터

추가 : `Addins > builder` 활용



3. 교재 데이터 시각화 비교

2. 막대 그래프 - 가로

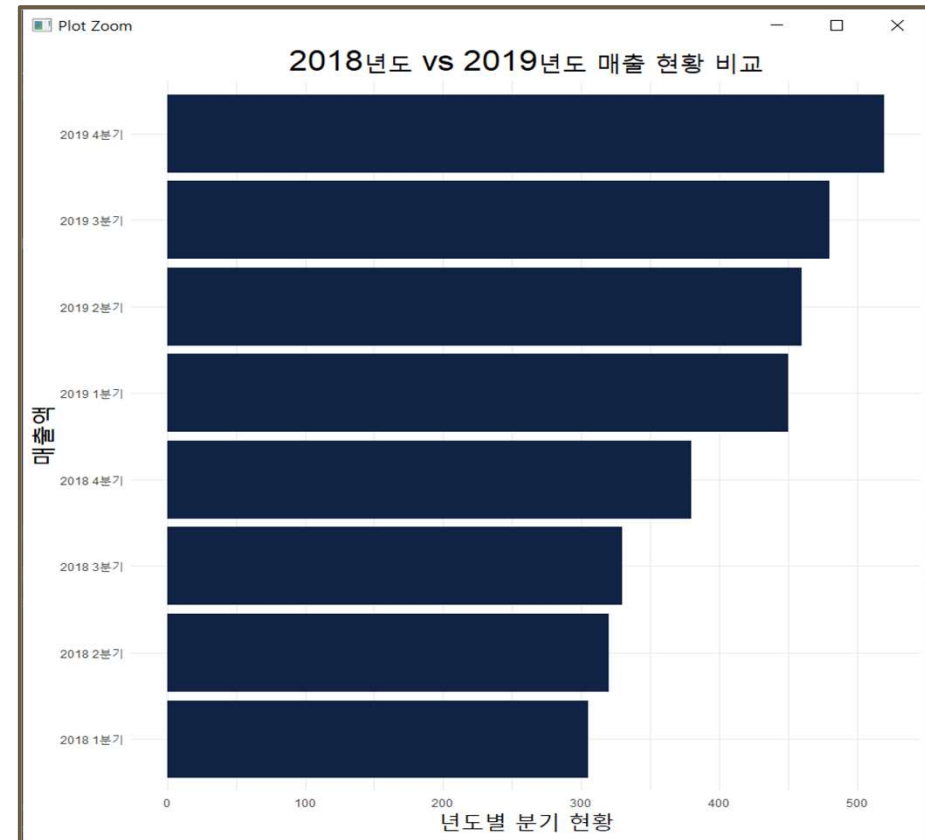
패키지 : `esquisse + ggplot2`

함수 : `ggplot() + geom_bar()`

데이터 : 사용자 정의 데이터

추가 : `Addins > builder` 활용

25



3. 교재 데이터 시각화 비교

26

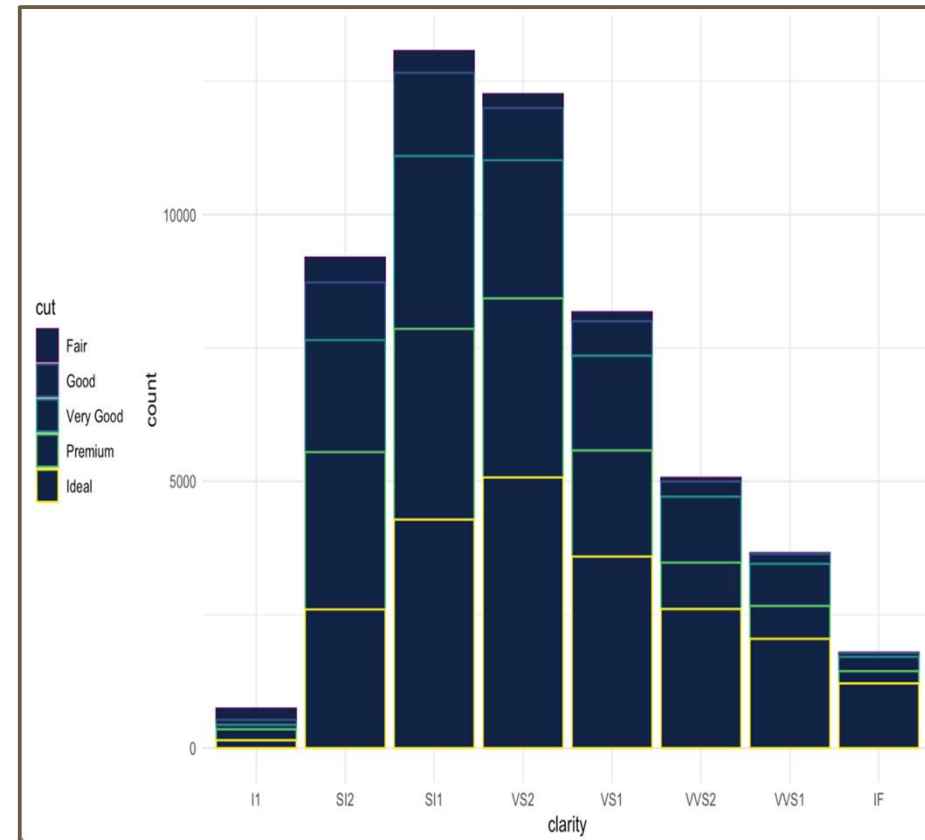
3. 누적막대 그래프

패키지 : `esquisse` + `ggplot2`

함수 : `ggplot()` + `geom_bar()`

데이터 : 사용자 정의 데이터

추가 : `Addins > builder` 활용



3. 교재 데이터 시각화 비교

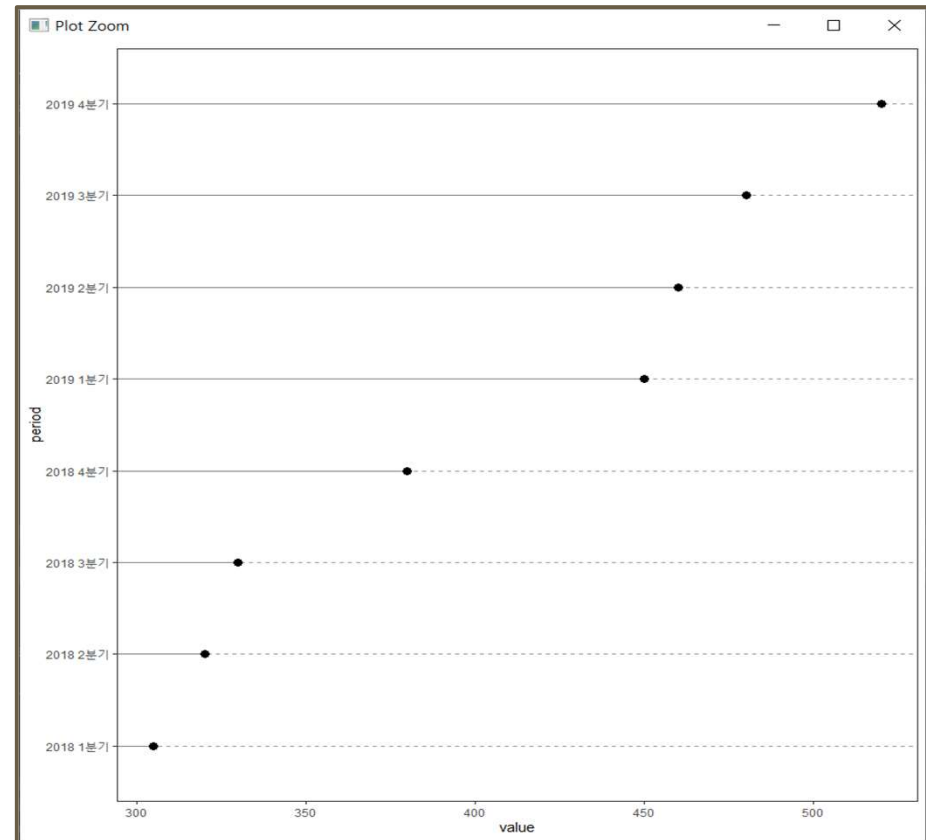
4. 점차트

패키지 : *esquisse* + *ggplot2*

함수 : *ggplot()* + *dotchart()*

데이터 : 사용자 정의 데이터

27



3. 교재 데이터 시각화 비교

5. 원형 차트

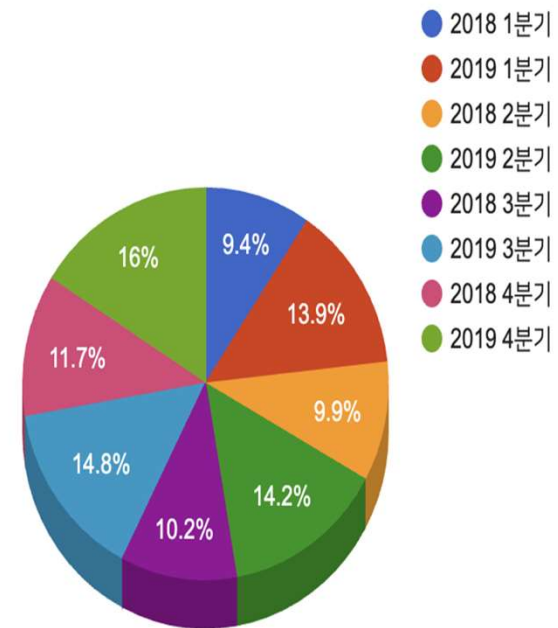
패키지 : *googleVis*

함수 : *gvisPieChart()*

데이터 : 사용자 정의 데이터

28

2018년도 vs 2019년도 매출 현황



3. 교재 데이터 시각화 비교

6. 상자 그래프

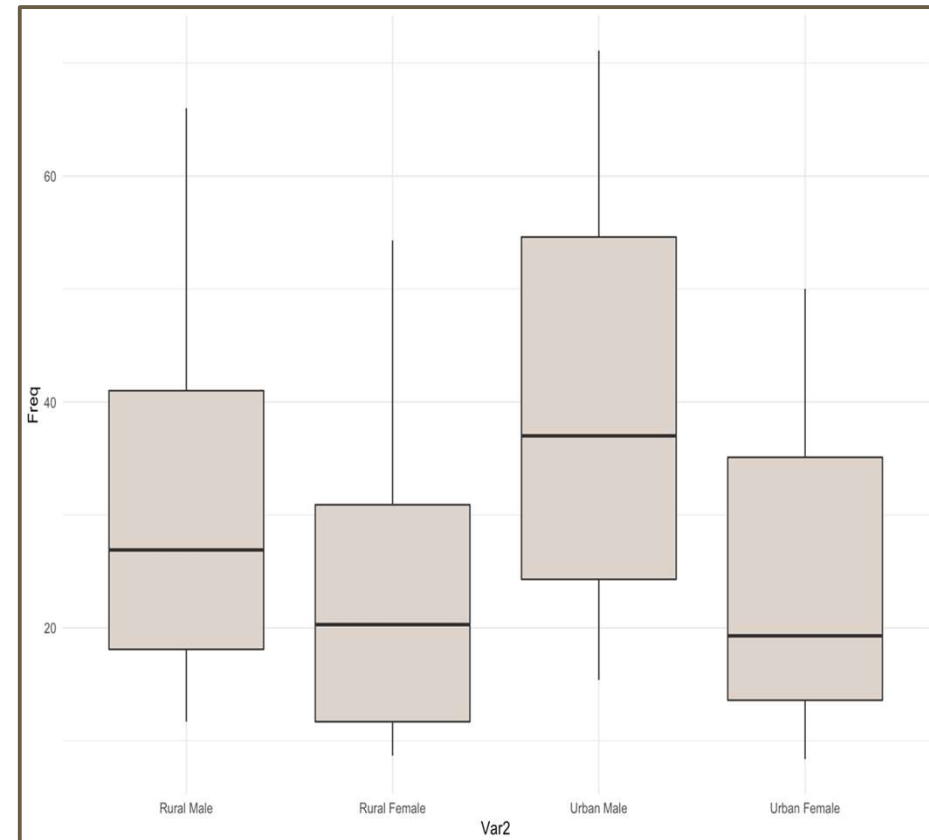
패키지 : *esquisse* + *ggplot2*

함수 : *ggplot()* + *geom_boxplot()*

데이터 : *VADeaths*

Addins > builder 활용

29



3. 교재 데이터 시각화 비교

30

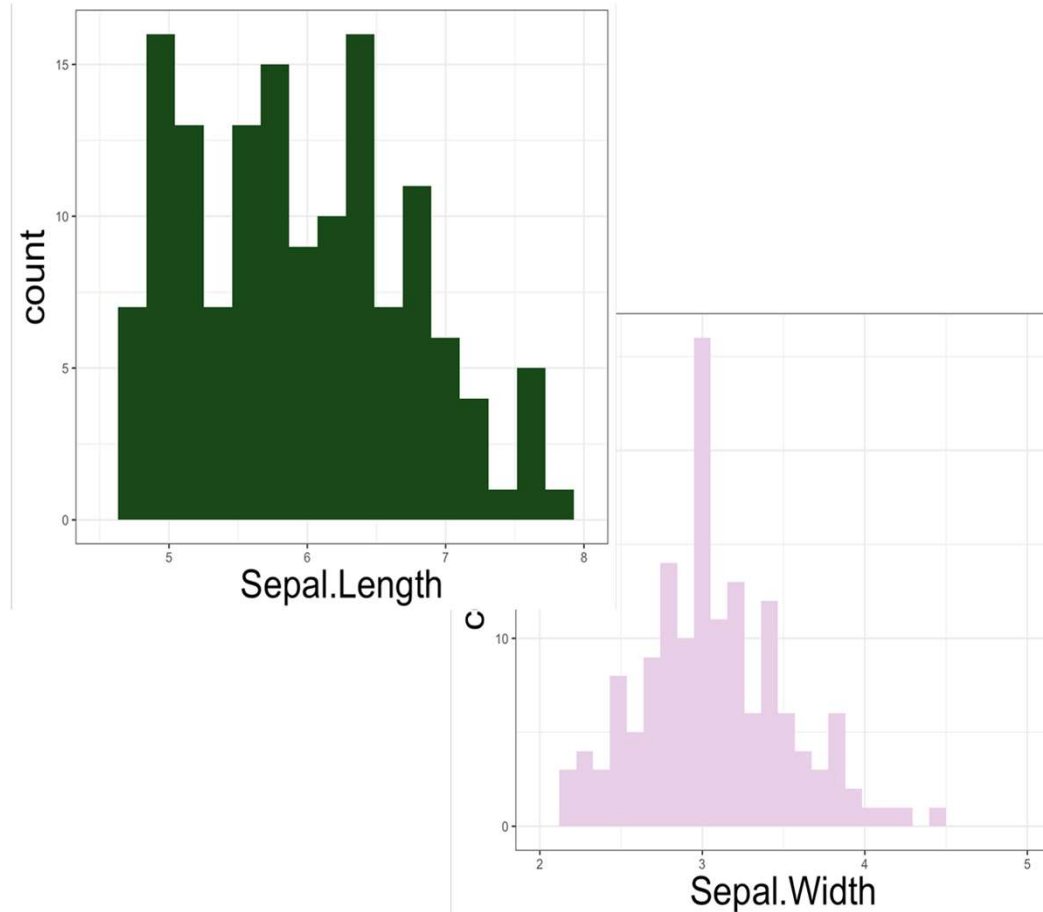
7. 히스토그램

패키지 : *esquisse* + *ggplot2*

함수 : *ggplot()* + *geom_histogram()*

데이터 : *iris*

Addins > builder 활용



3. 교재 데이터 시각화 비교

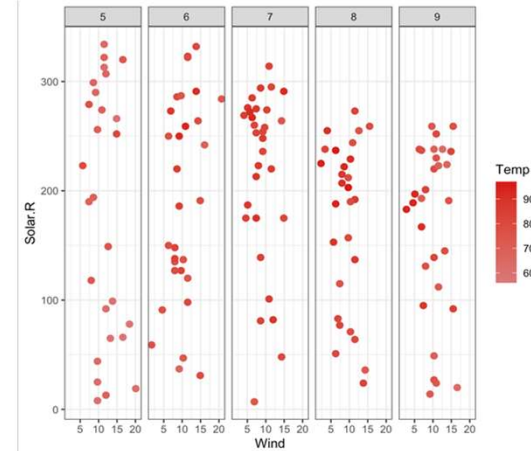
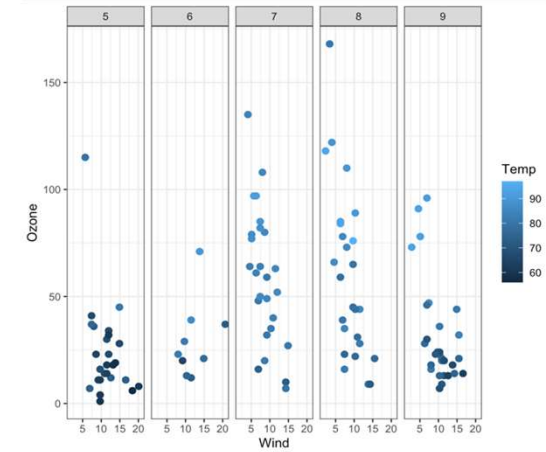
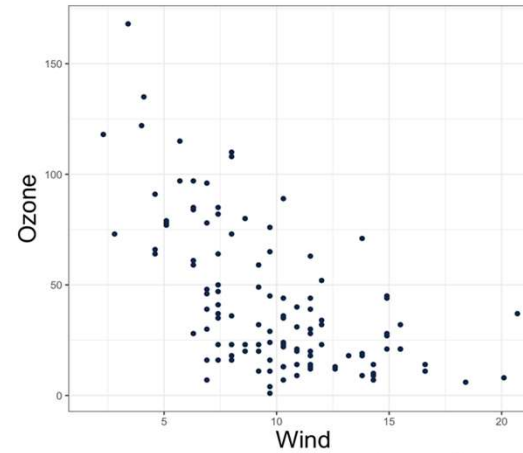
8. 산점도

패키지 : *esquisse* + *ggplot2*

함수 : *ggplot()* + *geom_point()*

데이터 : *airquality*

Addins > builder 활용



3. 교재 데이터 시각화 비교

32

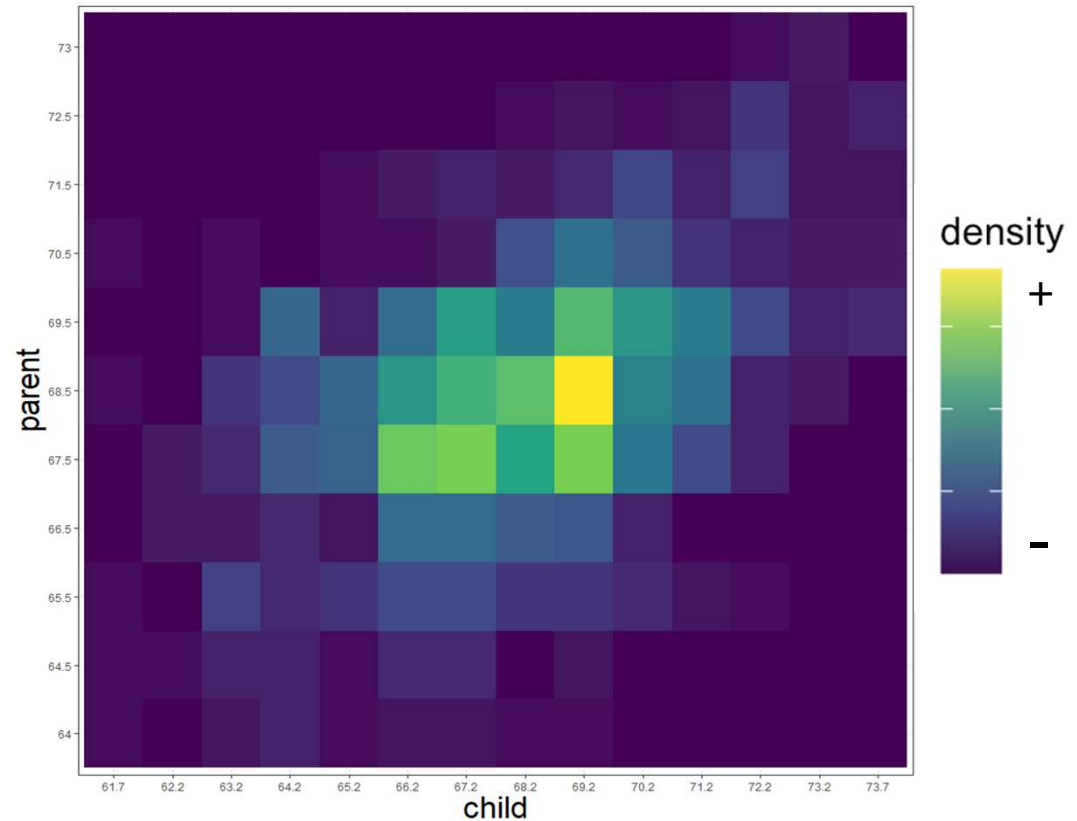
9. 중첩 자료

패키지 : `esquisse` + `ggplot2`

함수 : `ggplot()` + `scale_fill_viridis_c()`

데이터 : `galton`

`Addins > builder` 활용



3. 교재 데이터 시각화 비교

33

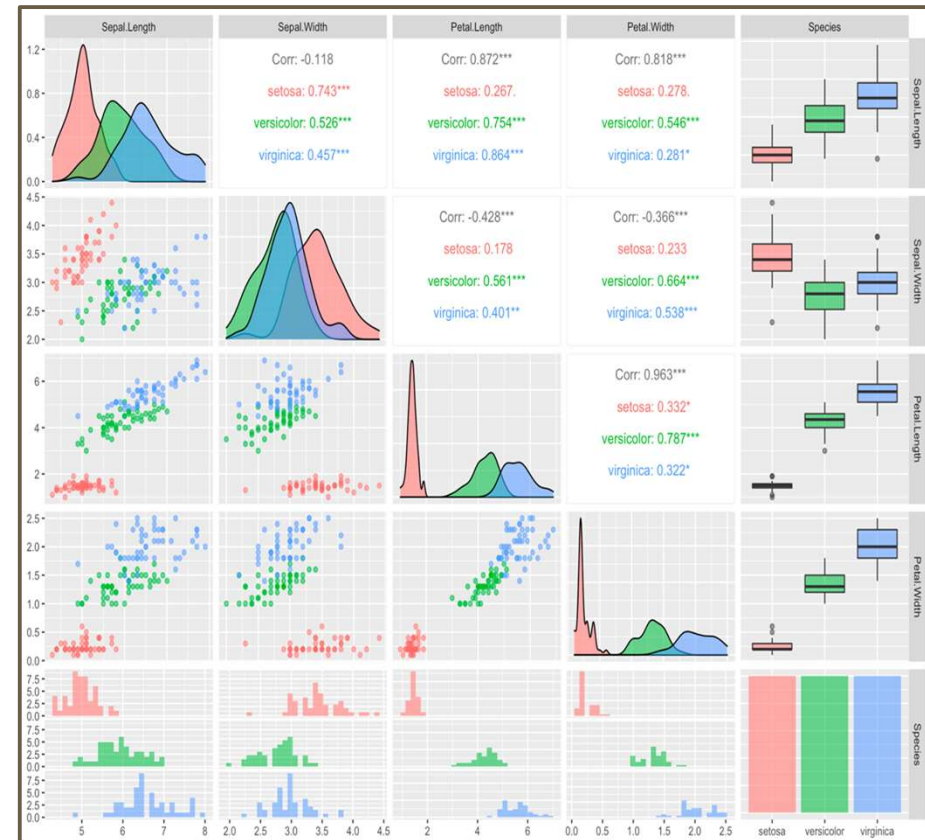
10. 변수간의 비교

패키지 : *GGally*

함수 : *ggpairs()*

ggpairs(iris,aes(colour=Species,alpha=0.4))

데이터 : *iris*



3. 교재 데이터 시각화 비교

11. 밀도그래프

패키지 : `ggplot2`

함수 : `ggplot()` +

+ `aes(x = gcse_score, colour = gender)`

+

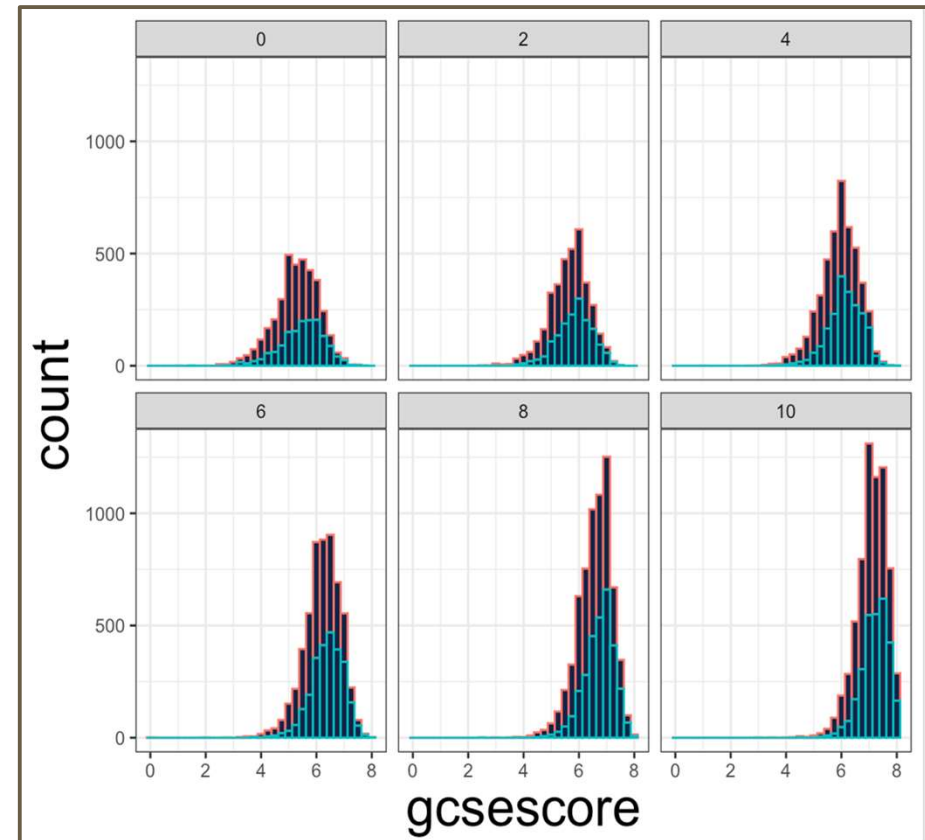
`geom_histogram()` +

`scale_color_hue(direction = 1)` +

`geom_density()` +

`facet_wrap(vars(score))`

데이터 : `Chem97(mlmRev)`



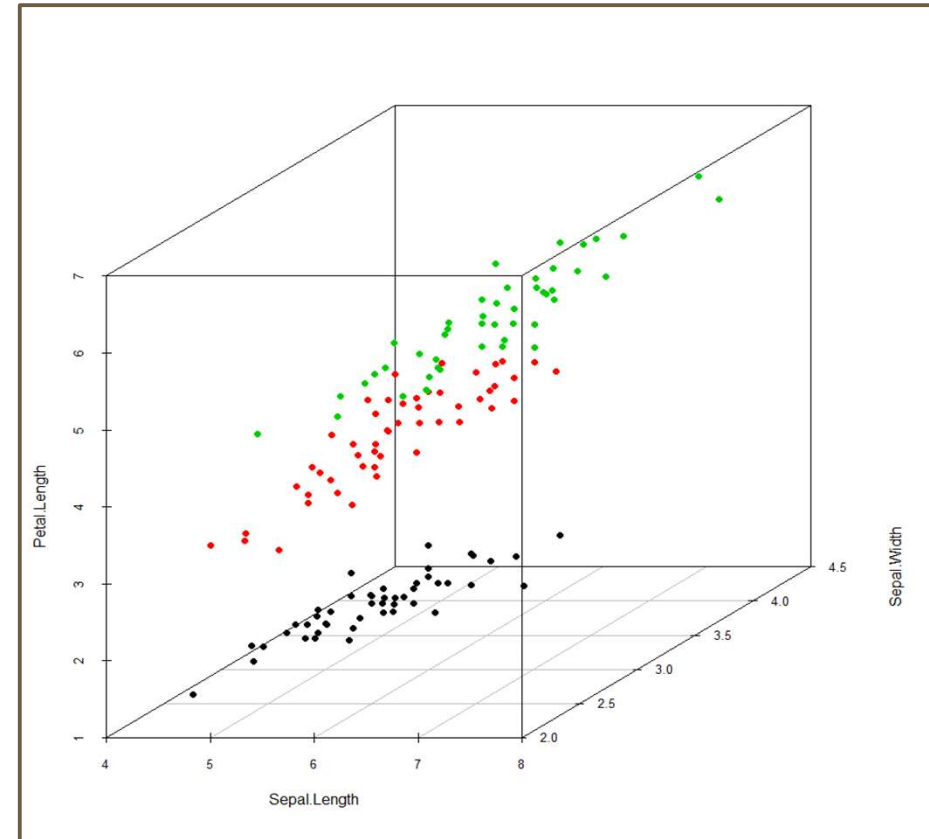
3. 교재 데이터 시각화 비교

12. 3차원 산점도

패키지 : `scatterplot3d`

함수 : `scatterplot3d()`

데이터 : `iris(R 기본 데이터 셋)`

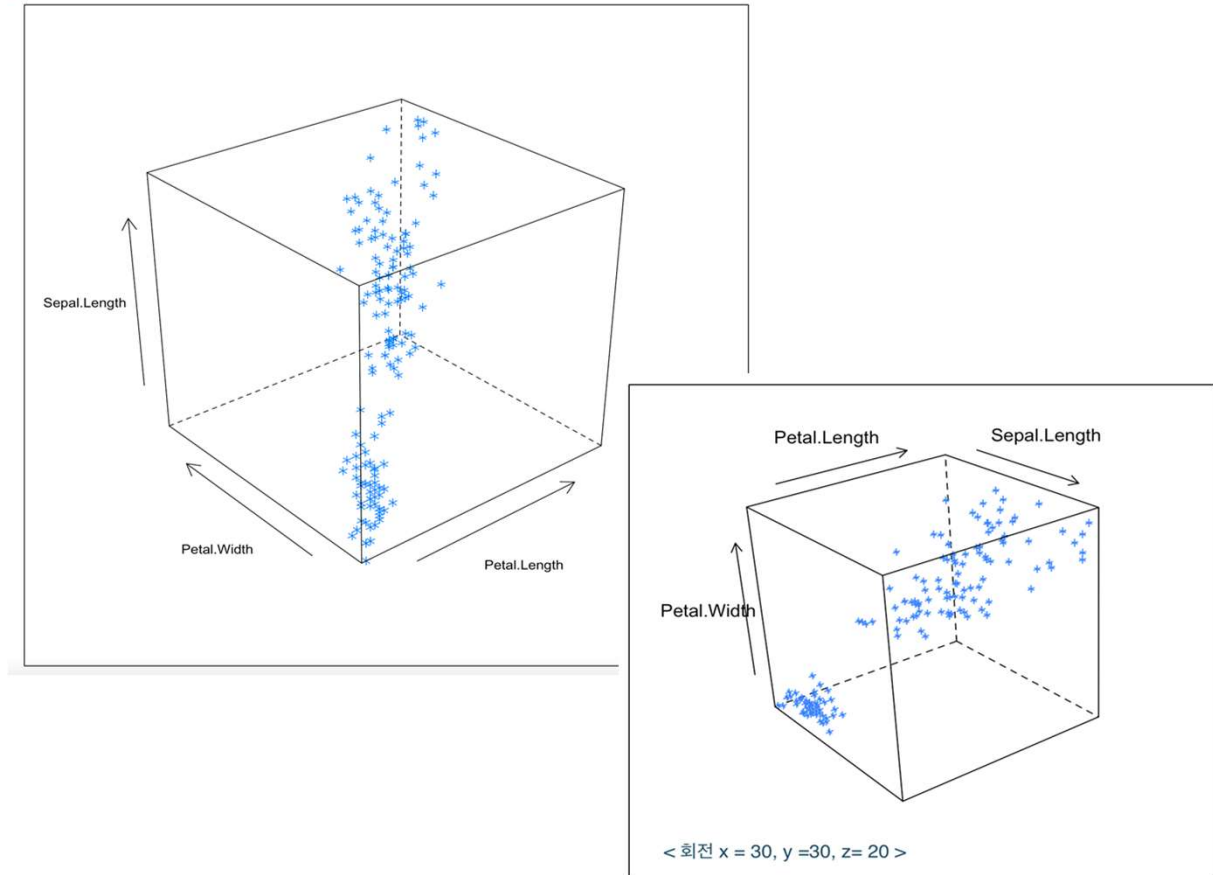


12. 3차원 산점도

패키지 : `ggplot2`

함수 : `cloud()`

데이터 : *iris*(R 기본 데이터 셋)

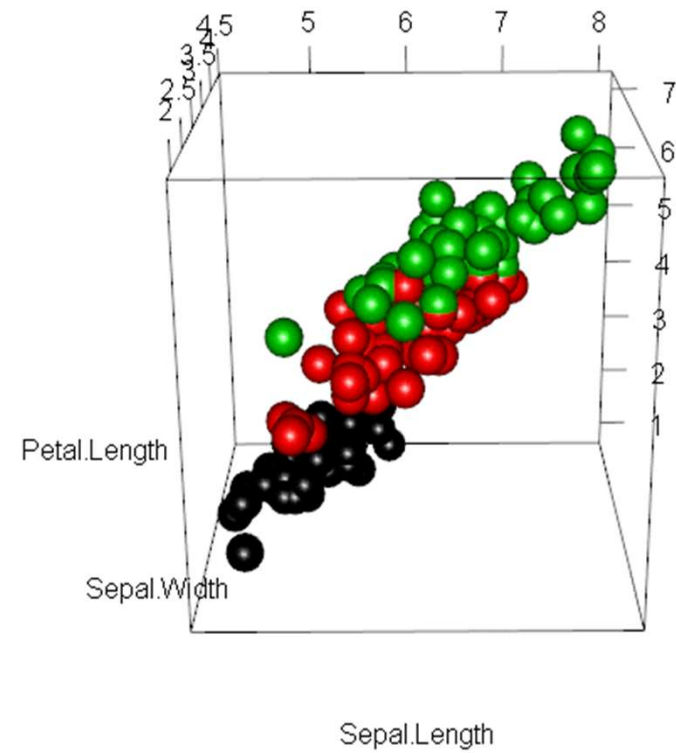


12. 3차원 산점도

패키지 : *rgl*

함수 : *plot3d*

데이터 : *iris*(R 기본 데이터 셋)



3. 교재 데이터 시각화 비교

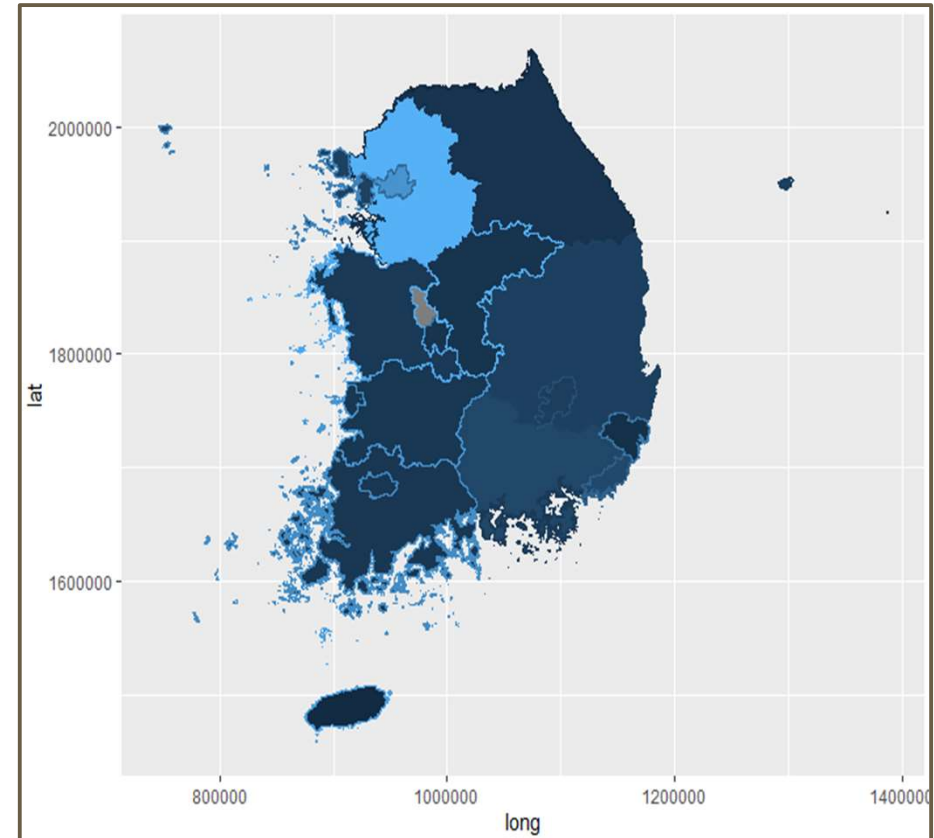
38

13. 지도

패키지 : `ggplot2`

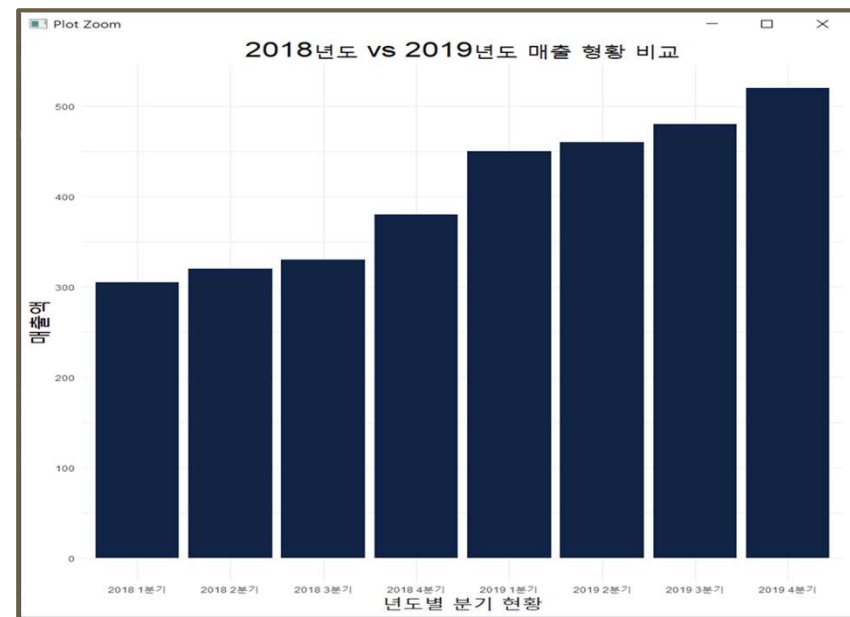
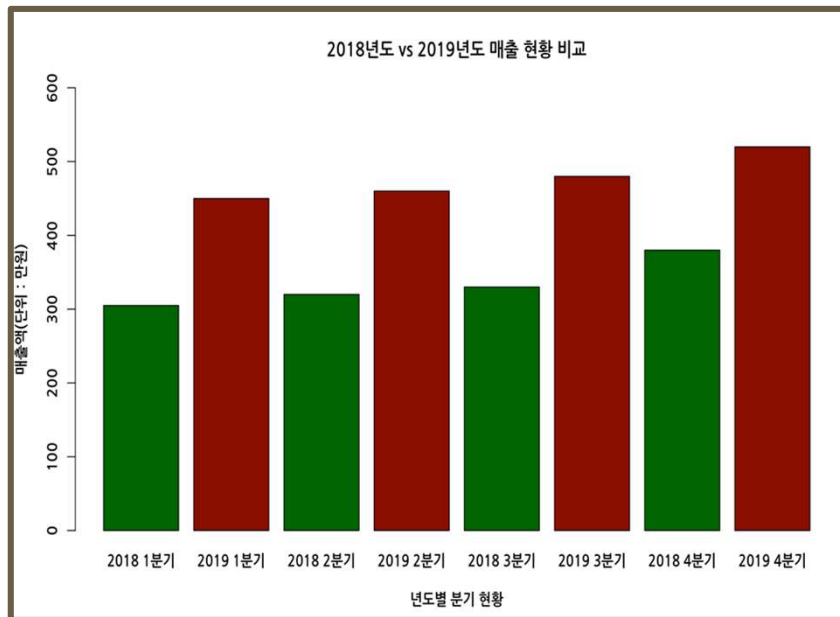
함수 : `ggplot(data = df_map_All_join,`
 `aes(x = long, y = lat,`
 `group = group,`
 `color = id,`
 `fill = 총인구수)) +`
 `geom_polygon() +`
 `theme(legend.position = "none")`

데이터 : [population201901](#) (링크 참조)



4. 시각화 비교 분석 - 막대그래프 *(교재 | 추가)

39



1. 막대 그래프

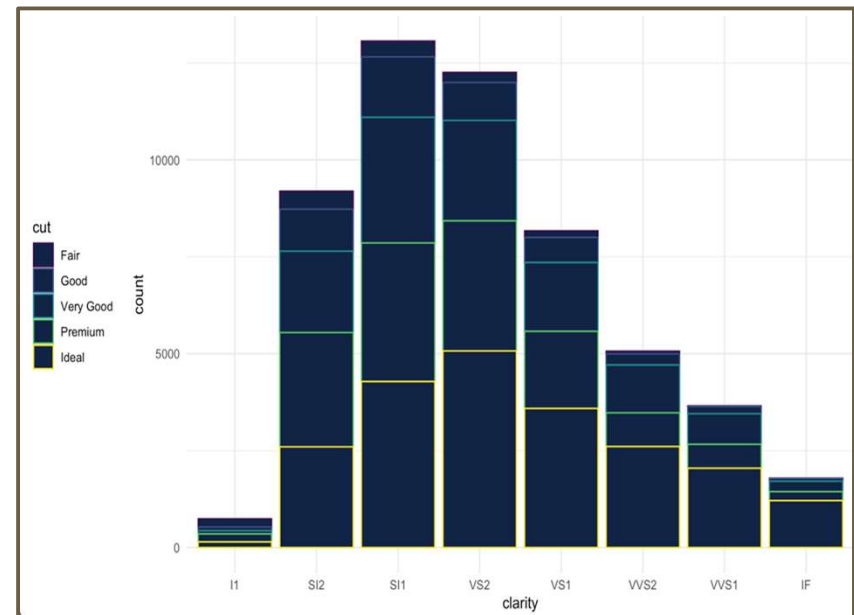
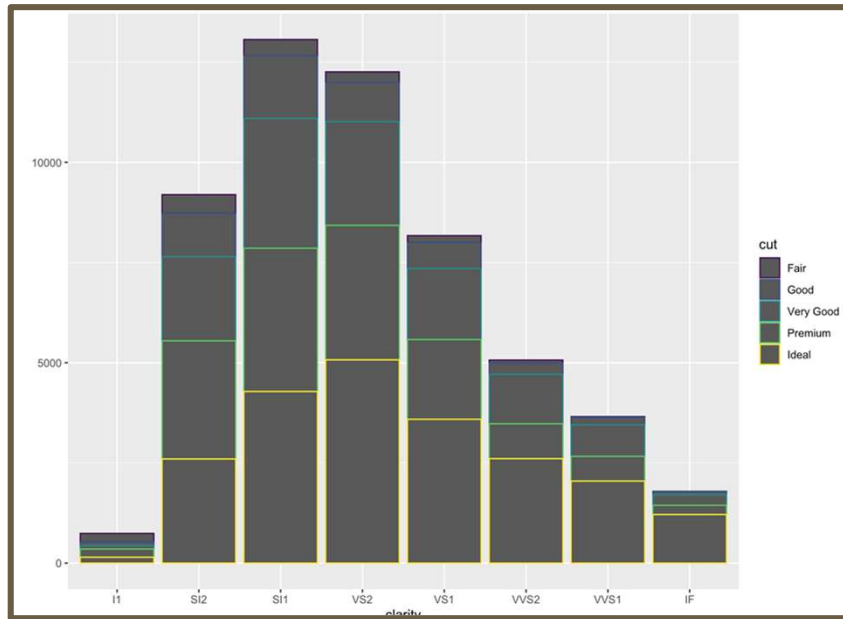
가로, 세로 막대 그래프는 패키지 종류에 관계 없이 직관적으로 코드가 적용되며 시각화에서도 편차 없이 안정적인 성능을 보인다.

R 내부 패키지인 **graphics**에서 제공되는 **plot** 함수에 경우 색상을 다양하게 구상할 수 있도록 하는 등 미적 요소 맵핑이 간단하고 반면 **esquisse** 패키지는 시각화에 있어 손쉽고 빠르지만 미적 요소 맵핑에 복잡한 구성을 갖고 있다.

이는 **esquisse** 경우 통계적 성능이 우선시 되기 때문으로 보인다.

4. 시각화 비교 분석 - 누적 막대 차트

40



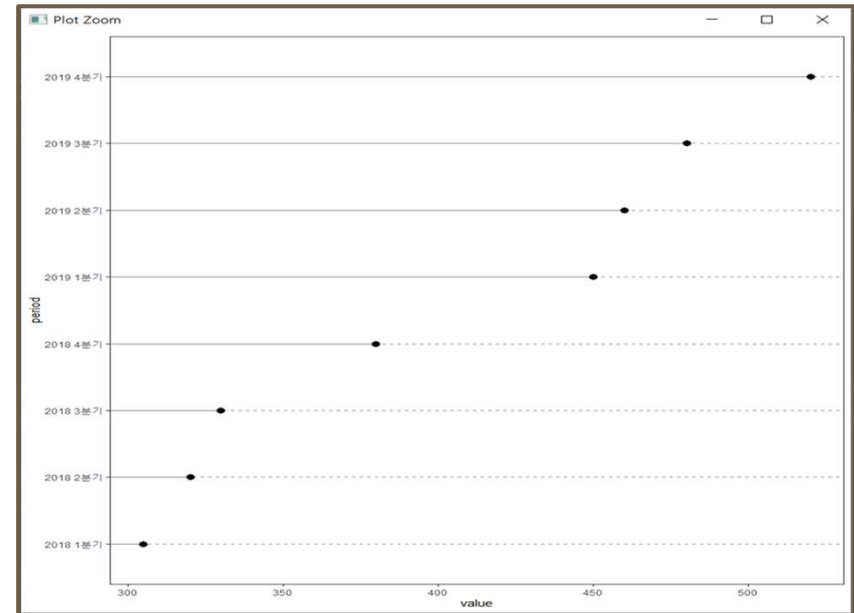
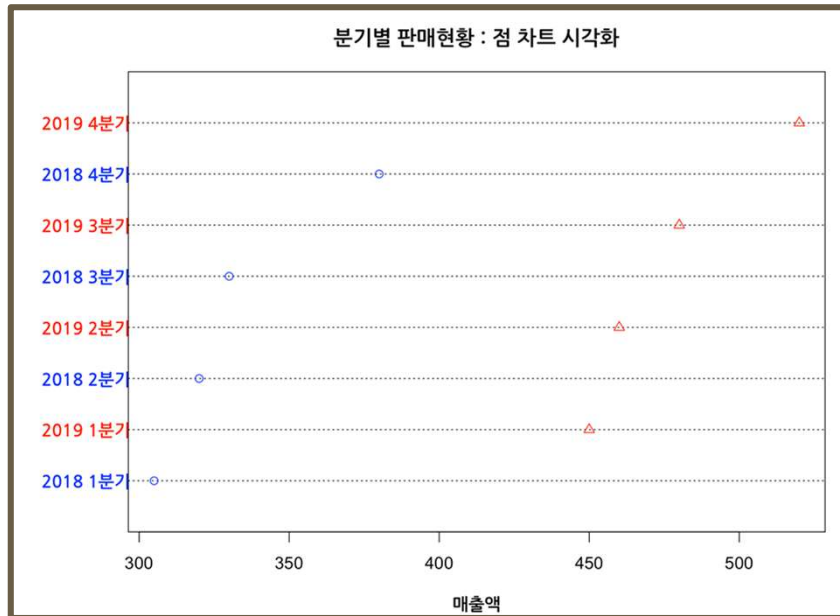
2. 누적 막대 차트

가장 많이 사용 되는 **plot** 패키지는 **ggplot2** 이며, 이와 비교한 패키지는 **esquisse** 이다. **esquisse**는 **ggplot2**를 활용하기 때문에 시각화에서는 큰 차이가 없지만 데이터 분석과 **dplyr**의 코드가 혼재 되어 있어 입력해야할 코드는 차이가 있다.

ggplot2의 경우 미적 요소 맵핑에 특화되어 코드가 간결하고 직관적인 장점이 있다. **builder** 활용으로 인해 모든 시각화 환경이 코드화 되어 길게 표현됨으로 같은 이미지 임에도 불구하고 간결하지 않다.

4. 시각화 비교 분석 - 점 차트

41



3. 점 차트

점 차트에서 기본 내장 함수와 비교 함수는 **esquisse**의 **ggplot2**을 활용했다.

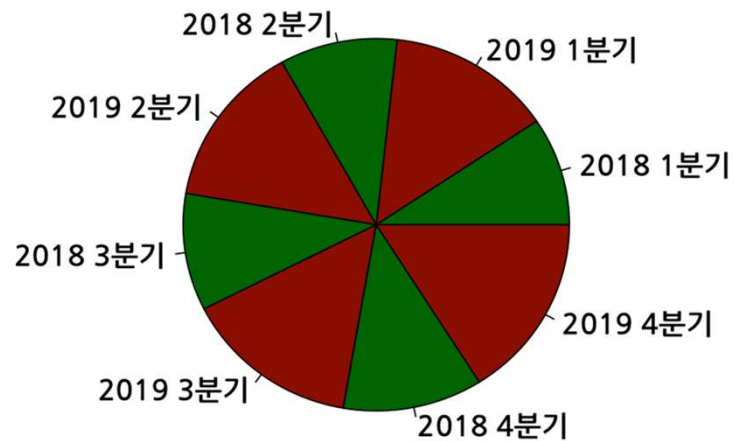
코드는 동일하게 **dotchart**를 사용하지만 추가 미적 요소 맵핑에선 차이가 있다.

esquisse 함수의 경우 2018년도와 2019년도가 매출액에 자동 정렬 기능을 갖고 있어 데이터에 더 최적화 된 것을 알 수 있다.

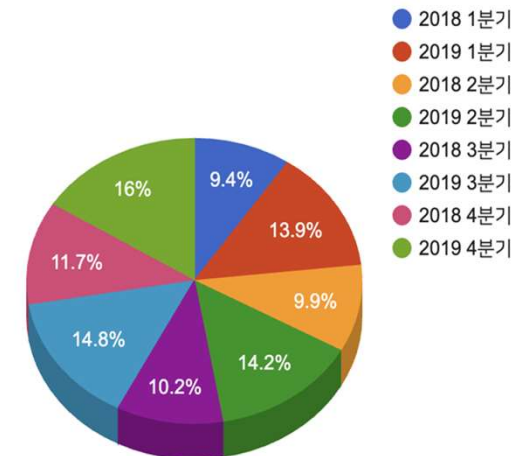
4. 시각화 비교 분석 - 원 차트

41

2018 - 2019 년도 분기별 매출 현황



2018년도 vs 2019년도 매출 현황



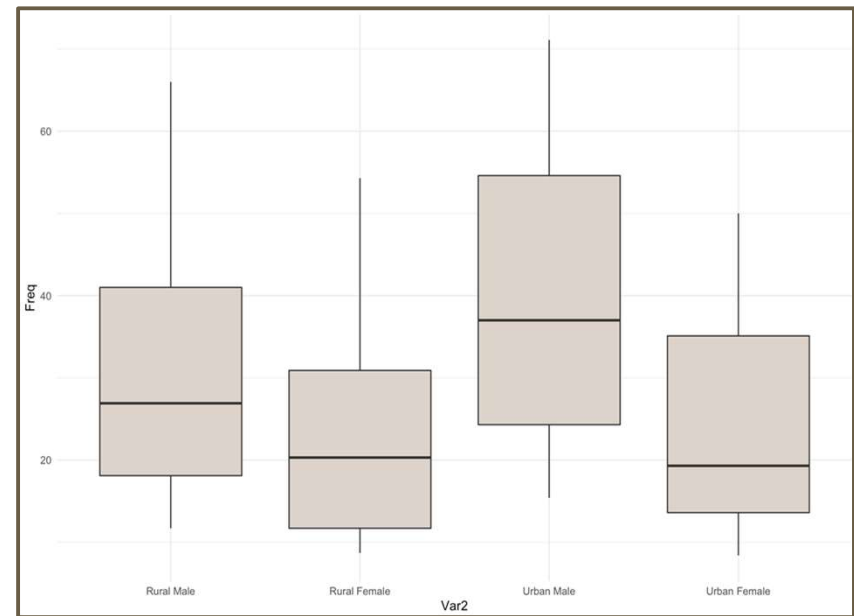
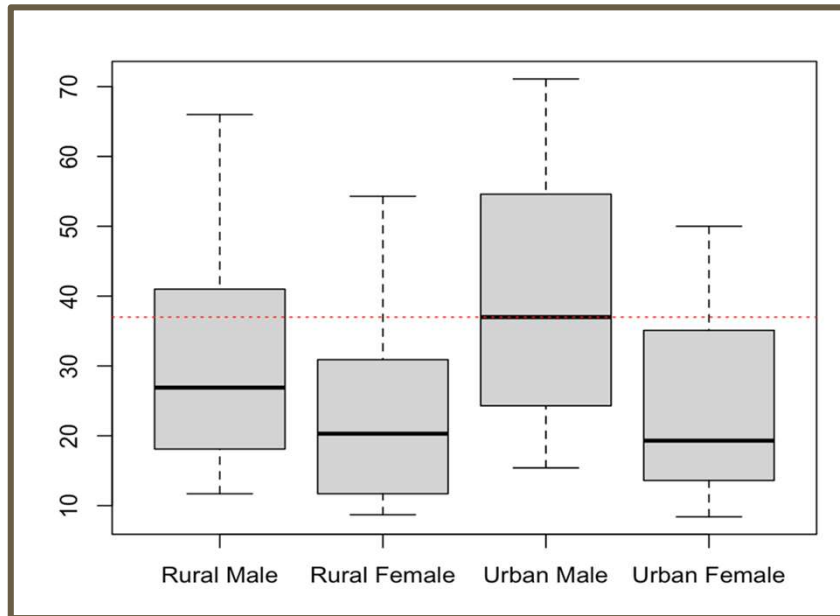
4. 원 차트

기본 내장 함수와 비교 시각화는 **googleVis** 패키지를 활용하였다.

내장 함수보다 **googleVis** 는 입체적 이며(음영 등 의 추가 기능), 비율로 표현되어 있어 직관적인 데이터 파악에 유리하다.

4. 시각화 비교 분석 - 상자 그래프

42



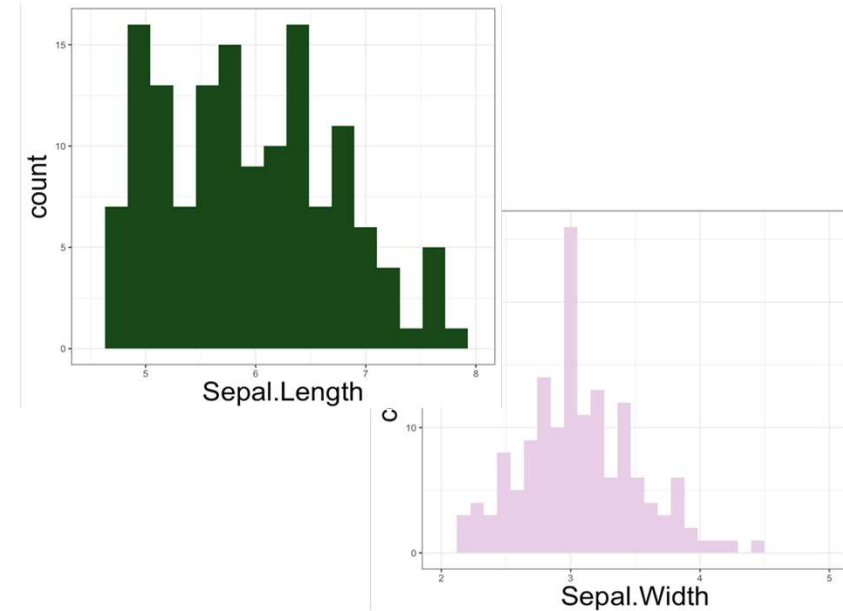
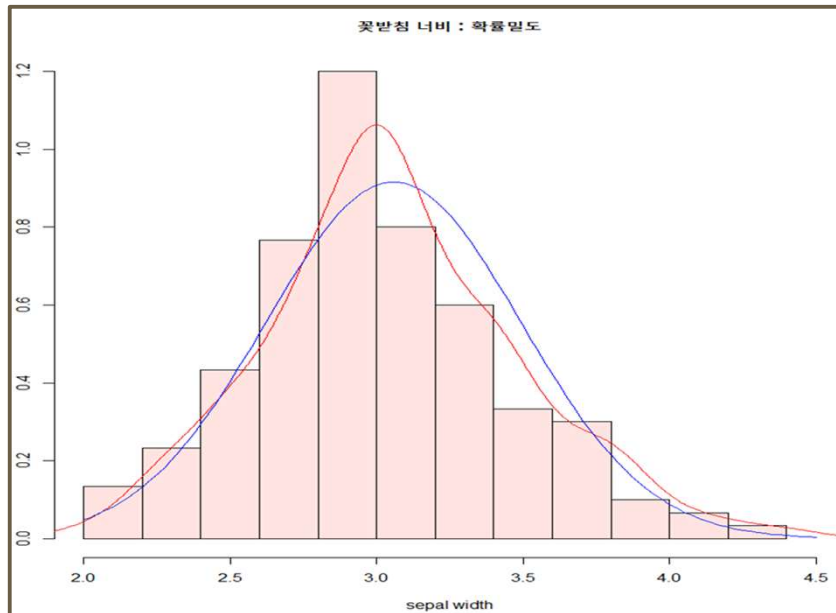
5. 상자 그래프

기본 내장 함수 **boxplot** 함수를 사용하였고, 비교 시각화는 **esquisse** 패키지의 **ggplot2**를 활용하여 **boxplot** 함수 사용으로 결과가 동일하게 나왔다.

앞서 다른 **esquisse** 패키지에 대한 아쉬운 점이 있었으나 상자 그래프는 미적 요소 보다는 다양한 정보(중앙값, 분위수, 최대.최솟값)를 표현하는 그래프이기에 간단하고 빠르게 시각화 할 수 있는 **esquisse** 패키지가 적합 하다.

4. 시각화 비교 분석 - 히스토그램

43

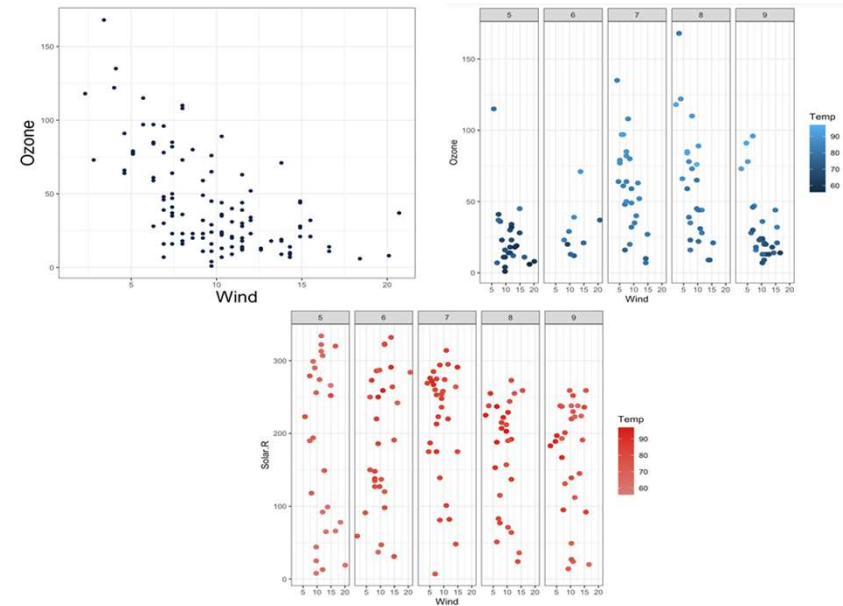
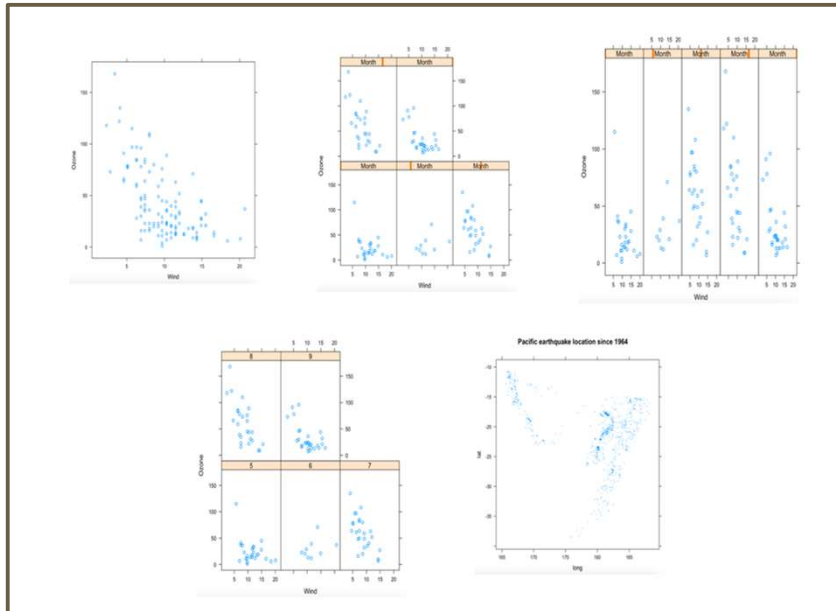


6. 히스토그램

기본 내장 함수 `hist` 함수를 사용하였고, 비교 시각화는 `esquisse` 패키지의 `ggplot2`를 활용하여 `geom_histogram` 함수 사용 하였다. 기본 내장 함수는 밀도 곡선 표현 할 수 있었지만 비교 시각화는 `factor` 컨트롤에 어려움이 있어 간단한 히스토그램을 시각화만 가능 하였다.

4. 시각화 비교 분석 - 산점도

44



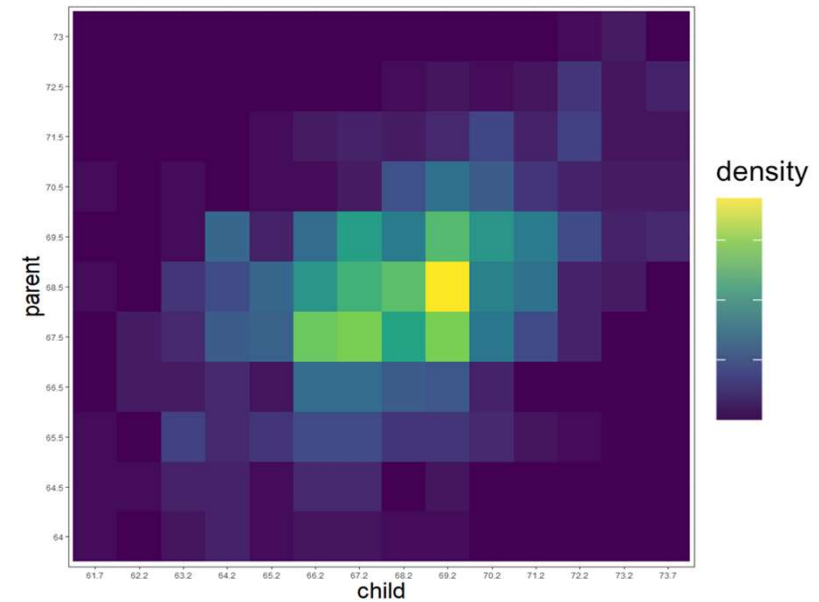
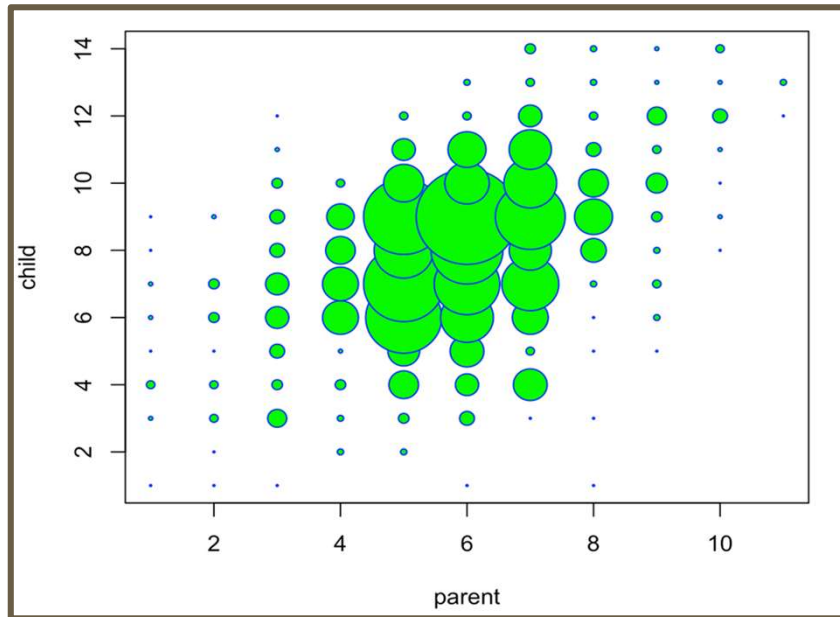
7. 산점도

교과에서 **lattice** 패키지로 다양한 조건에서의 공기의 질을 산점도로 표현하였다. 비교한 패키지는 **esquisse** 패키지의 **ggplot2**를 활용하여 **geom_point** 함수를 사용하여 표현했다.

정보 전달에 큰 차이는 없지만 **ggplot2**의 특성상 미적 요소 맵핑에 더 편리한 점이 눈에 띈다.

4. 시각화 비교 분석 - 중첩자료

45



8. 중첩 자료

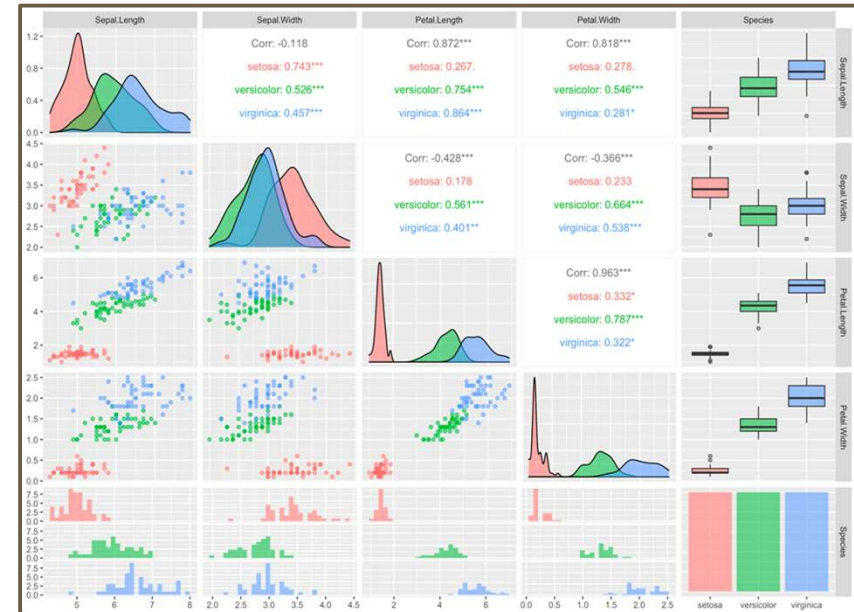
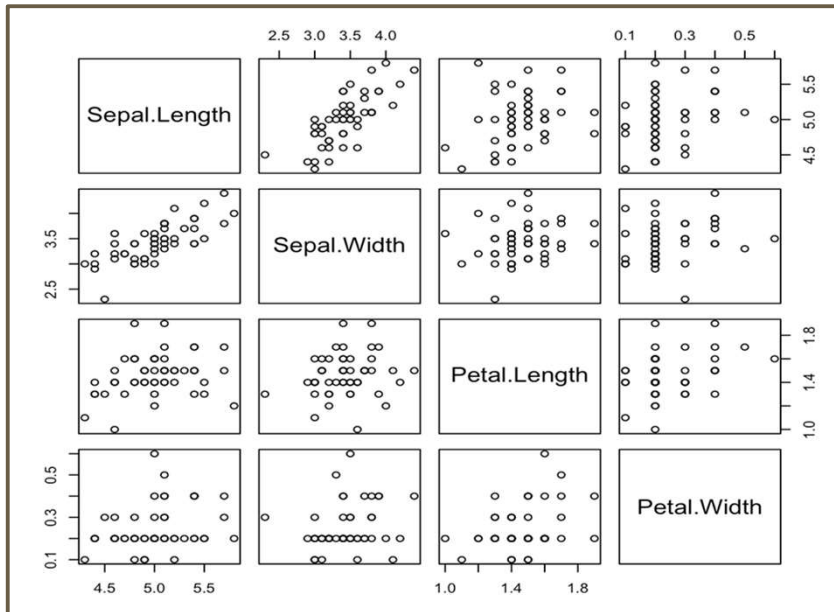
기본 내장함수와 **esquisse** 패키지의 **ggplot2(scale_fill_viridis)**를 비교 하여 시각화를 구성해 보았다.

기본 내장함수는 중첩될수록 동그라미가 커짐에 따라 시각적으로 도드라져 보였고 비교 시각화 색상만으로 중첩성을 표현했다.

분석 대상의 데이터의 성격에 따라서 두 가지 시각화 모두 장 단점이 있을 수 있지만, 본 데이터에서는 색보다는 크기가 더 직관적으로 데이터를 파악하기 유리하단 것을 알 수 있다.

4. 시각화 비교 분석 - 변수간의 비교

46



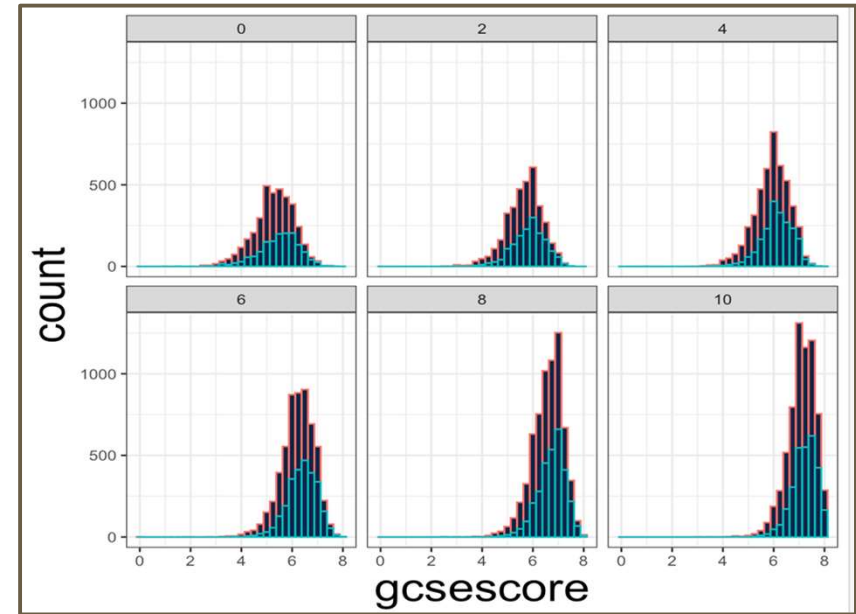
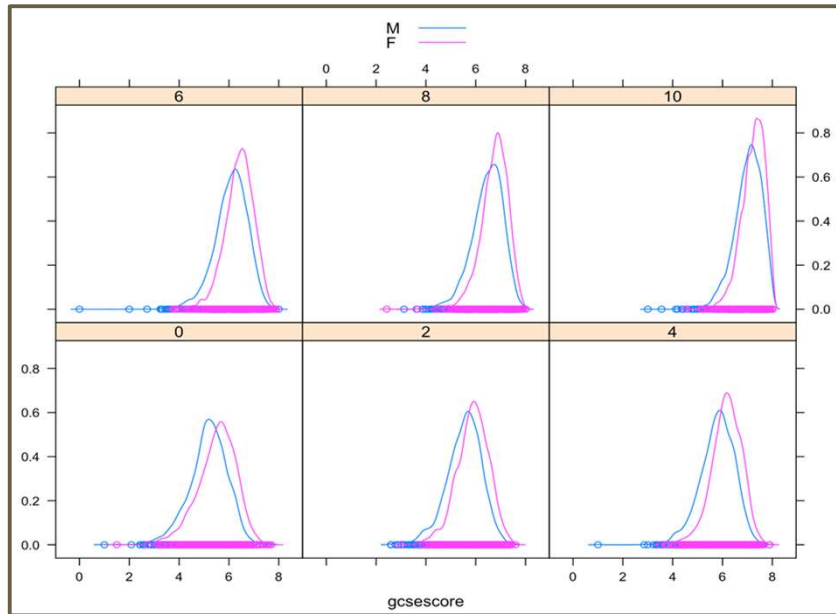
9. 변수간의 비교

기본 내장함수와 **GGally** 패키지의 **ggpairs** 함수를 비교 하였다.

GGally 패키지는 변수간의 관계 뿐만 아니라 해당 변수의 정보 까지 보여준다. 게다가 변수 간의 차이를 다른 색상으로 표현되어 시각화 활용도가 높다.

4. 시각화 비교 분석 - 밀도 그래프

47

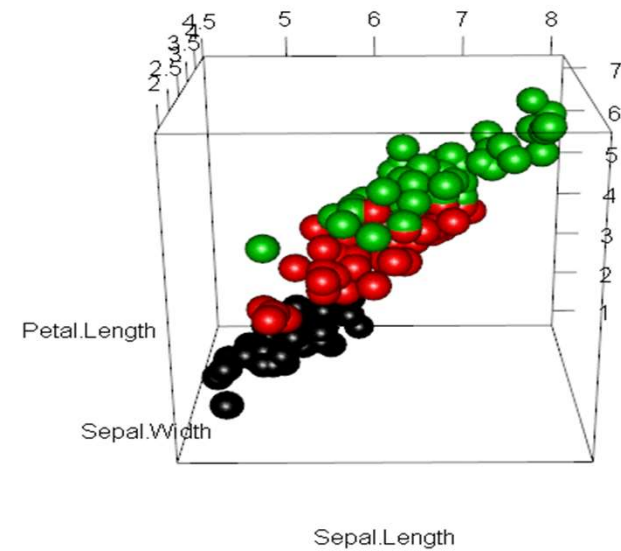
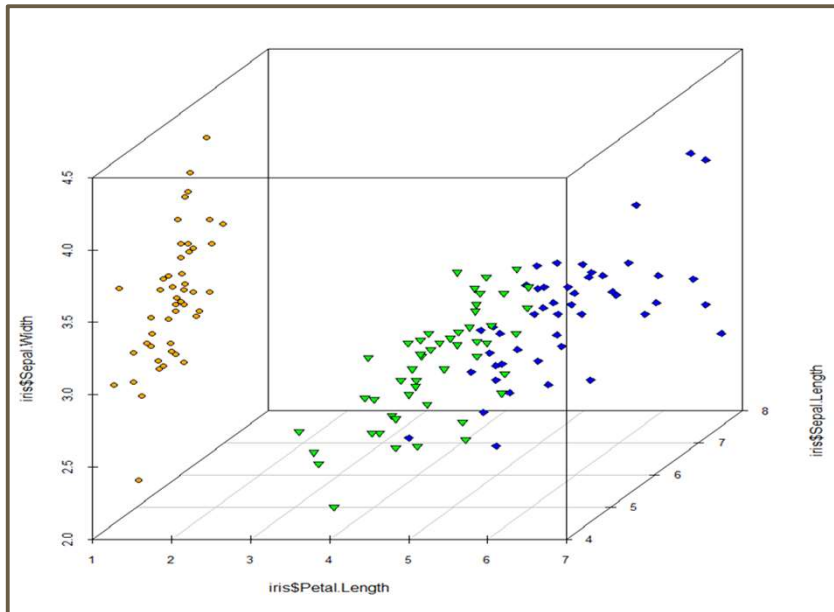


10. 밀도 그래프

lattice 패키지의 **densityplot** 함수와 **ggplot2** 패키지의 **geom_histogram + geom_density** 함수를 비교하여 시각화 하였다. 그래프의 높낮이로 밀도의 정도를 알 수 있다. 시각화 정보 전달에 있어 어려움이 없다.

4. 시각화 비교 분석 - 3차원 산점도

48



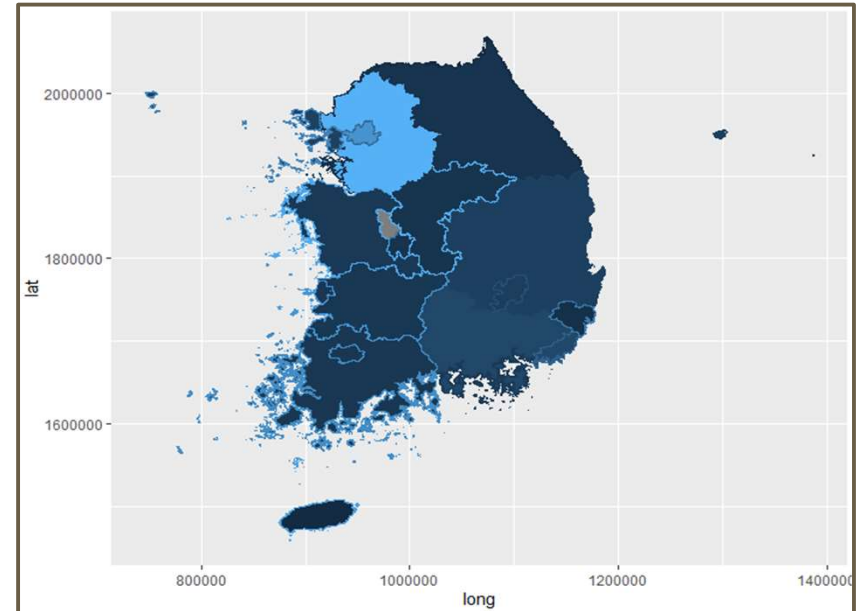
11. 3차원 산점도

3차원 산점도는 다양한 패키지가 있어 여러가지를 비교해 보았다.

변수에 각각 다른 색으로 차이를 두어 시각화 하고 마우스로 인터렉션 기능으로 가려지는 부분을 돌려서 볼 수 있도록 하는 **rgl** 패키지가 제일 적합하다.

4. 시각화 비교 분석 - 지도

49



12. 지도

교과는 기본 내장함수 비교 시각화는 **ggmap** 패키지의 **ggmap** 함수를 사용 했다.

ggplot2 패키지는 **Basic package** 보다 대중적인 그래픽을 제공한다. 인구 밀집을 점으로 표현하는 **ggmap** 과 다르게, 해당 지역을 색 채우기로 표현함으로 직관적인 시각화가 가능하다.

가장 활용도가 높을 것 같은 패키지는 ?

실제 가장 많이 사용 되는 'ggplot2'를 제외하고는 'rgl' , 'GGally' 이 두 가지 패키지가 가장 많이 활용 될 수 있을 것 같다.
간단한 코드로 정보를 효과적으로 전달할 수 있는 패키지이다.

rgl은 간단한 두 줄 코드임에도 입체적으로 모든면을 볼수 있고 시각적으로 정보를 얻을 수 있다.

GGally 또한 한줄 코드임에도 각 함수의 상관관계 데이터의 정보 등을 직관적으로 알 수 있다.

데이터 분석에서 시각화란 ?

분석에서 가장 중요한건 결과를 전달 하는 것이다.

결과를 전달하는 가장 보편화된 방식 중 하나가 시각화 하는 방법이다.

시각화는 어떠한 말보다 전달력이 높고 결과 제안 중 가장 중요한 **empathy**를 가져올 도구 이다.

시각화(결과) = 정보(데이터)