**R 로 데이터 분석을 할 때(분석에 대한 가설을 만들어 놔야 한다. – 약 5개 정도)**

1. **데이터 수집(가설에 합당한 데이터)**
2. **데이터를 수집했으면 데이터를 읽어 들여야 한다. (읽기 전에 전처리가 가능하면 전처리 먼저 수행)**
3. **데이터 내부 구조 분석(head, tail, summary 등으로 파악)**
4. **칼럼에 대한 설명이 별도 파일로 모두 나와있다. (이 내용을 파악)**
5. **4번 정보를 토대로 NA값/이상치가 있는지 없는지 검증해야 한다.  
   (이 때도 마찬가지로 결측값과 이상값을 처리(전처리 작업)) – 정제작업**
6. **어떤 통계 기법을 이용해서 분석했다. 내용이 보고 서 내부에 포함되어야 한다.**
7. **시각화 작업**

**3. 데이터 분석을 위한 연장 챙기기**

**변수(Variable)**

• 다양한 값을 지니고 있는 하나의 속성

• 변수는 데이터 분석의 대상



여러 값으로 구성된 변수 만들기

1. **c() (동일한 데이터타입만 들어갈 수 있다.)**

var1 <- c(1, 2, 5, 7, 8) # 숫자 다섯 개로 구성된 var1 생성

var1

## [1] 1 2 5 7 8

var2 <- c(1:5) # 1~5 까지 연속값으로 var2 생성

var2

## [1] 1 2 3 4 5

벡터를 생성시켜 주는 함수 : c(), seq(), rep() – 값을 반복하는 함수

1. **seq(시작값, 끝값) - by옵션을 이용해서 사이의 간격 조정 가능**

var3 <- seq(1, 5) # 1~5 까지 연속값으로 var3 생성

var3

## [1] 1 2 3 4 5

var4 <- seq(1, 10, by = 2) # 1~10 까지 2 간격 연속값으로 var4 생성

var4

## [1] 1 3 5 7 9

var5 <- seq(1, 10, by = 3) # 1~10 까지 3 간격 연속값으로 var5 생성 ( 간격에 따라 10이 안나올 수 있다. )

var5

## [1] 1 4 7 10

1. **연속값 변수로 연산하기**

var1

## [1] 1 2 5 7 8

var1+2 # 변수에 어떤 값을 더하게 되면 각 항목에 모두 연산

## [1] 3 4 7 9 10

var1

## [1] 1 2 5 7 8

var2

## [1] 1 2 3 4 5

var1+var2

## [1] 2 4 8 11 13

1. **문자로 된 변수 만들기**

str1 <- "a" str1

## [1] "a"

str2 <- "text" str2

## [1] "text"

str3 <- "Hello World!" str3

## [1] "Hello World!"

1. **연속 문자 변수 만들기**

str4 <- c("a", "b", "c") str4

## [1] "a" "b" "c"

str5 <- c("Hello!", "World", "is", "good!") str5

## [1] "Hello!" "World" "is" "good!"

**문자로 된 변수로는 연산할 수 없다**

str1+2

## Error in str1 + 2: non-numeric argument to binary operator

**03-2. 마술 상자 같은 '함수' 이해하기**

**함수**

• 값을 넣으면 특정한 기능을 수행해 처음과 다른 값이 출력됨

1. **숫자를 다루는 함수 이용하기**

# 변수 만들기

x <- c(1, 2, 3)

x

## [1] 1 2 3

# 함수 적용하기

mean(x)

## [1] 2

max(x)

## [1] 3

min(x)

## [1] 1

**추가적인 함수 : 데이타 셋 요약값 만들어주는 함수 : summary()**

**중앙값 : median(), 합 : sum()**

변수의 타입을 알아내는 함수 : class() – 문자면 character, 숫자형은 numeric

1. **문자를 다루는 함수 이용하기 (문자 합치기)-문자는 연산불가**

str5

## [1] "Hello!" "World" "is" "good!"

paste(str5, collapse = ",") # 쉼표를 구분자로 str5 의 단어들 하나로 합치기(collapse 옵션)

## [1] "Hello!,World,is,good!"

1. **함수의 옵션 설정하기 – 파라미터**

paste(str5, collapse = " ")

## [1] "Hello! World is good!"

1. **함수의 결과물로 새 변수 만들기**

x\_mean <- mean(x)

x\_mean

## [1] 2

str5\_paste <- paste(str5, collapse = " ")

str5\_paste

## [1] "Hello! World is good!"

**03-3. 함수 꾸러미, '패키지' 이해하기**

1. **패키지(packages)**
   1. **기본 패키지 : base**
   2. **추가 패키지 : CRAN(미러사이트에서 다운받아서 사용하는 패키지)**

* **install.packages(“패키지명”) 함수를 이용해서 패키지 설치 --> library(패키지명) 으로 로드해야 한다.**
* **R 스튜디오에서는 패키지 탭에서 install 을 눌러서 패키지명 입력 후 다운 가능**
* **cascade : install 메시지가 출력되고, download 메시지가 출력. 다 끝나면 ‘>’ 프롬프트가 나와야 완료.**

• 함수가 여러 개 들어 있는 꾸러미

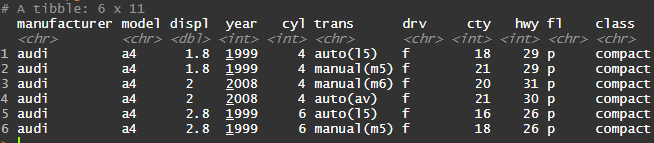
• 하나의 패키지 안에 다양한 함수가 들어있음

• 함수를 사용하려면 패키지 설치 먼저 해야함

1. **ggplot2 패키지 설치하기, 로드하기**

install.packages("ggplot2") # ggplot2 패키지 설치

library(ggplot2) # ggplot2 패키지 로드

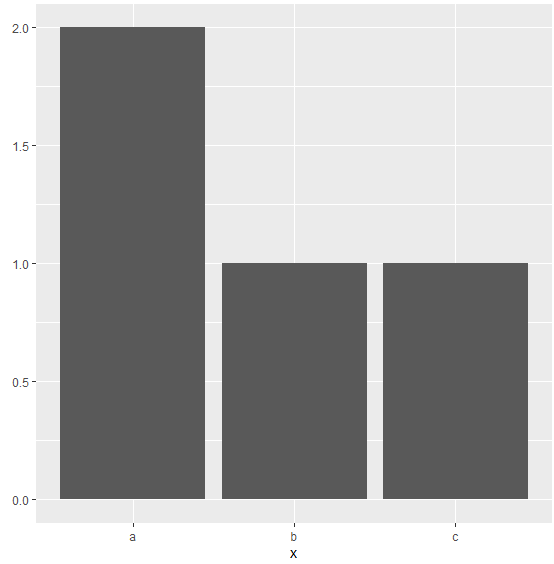
**head(mpg), cty : 고속도로 연비, hwy : 고속도로 연비, drv: 기어 단수  
**

예제)

x <- c("a", "a", "b", "c")

# 빈도 그래프 출력

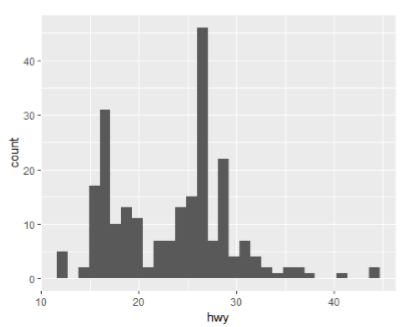
qplot(x) # 대표 이름을 찾는다.(각 항목이 몇개인지 파악해서 차트로 만들어준다.)

****

1. **ggplot2의 mpg 데이터로 그래프 만들기**

# data 에 mpg, x 축에 hwy 변수 지정하여 그래프 생성 (mpg는 ggplot이 가지고 있는 데이터셋)

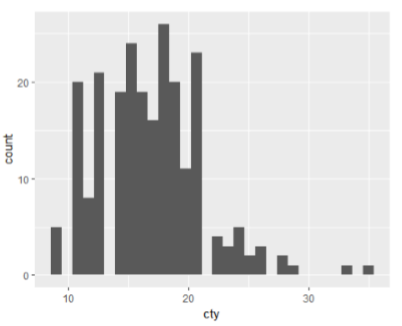
qplot(data = mpg, x = hwy)

****

1. **qplot() 파라미터 바꿔보기**

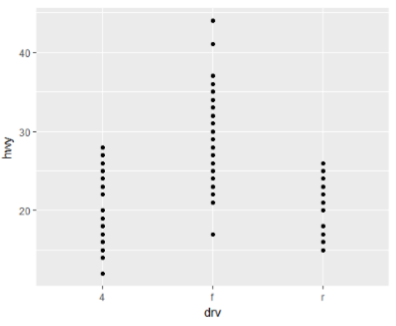
# x 축 cty

qplot(data = mpg, x = cty)



# x 축 drv, y 축 hwy

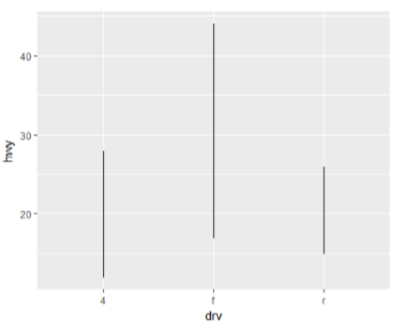
qplot(data = mpg, x = drv, y = hwy)



# x 축 drv, y 축 hwy, 선 그래프 형태

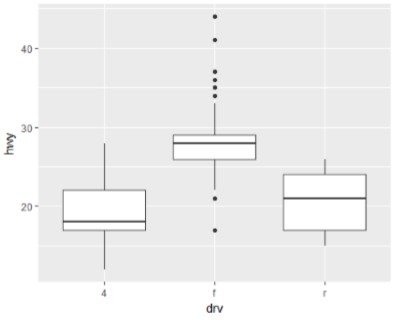
* qplot 은 geom 옵션으로 차트 형태를 바꿀 수 있음

qplot(data = mpg, x = drv, y = hwy, geom = "line") # geom : 모양을 선 모양으로 바꿔달라고 요청



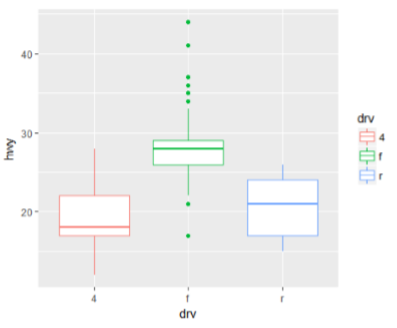
# x 축 drv, y 축 hwy, 상자 그림 형태

qplot(data = mpg, x = drv, y = hwy, geom = "boxplot") # boxplot 차트로 변경



# x 축 drv, y 축 hwy, 상자 그림 형태 , drv 별 색 표현

qplot(data = mpg, x = drv, y = hwy, geom = "boxplot", colour = drv) # colour 옵션으로 범례 지정 가능



Q1. 시험 점수 변수 만들고 출력하기

score <- c(80, 60, 70, 50, 90) score

## [1] 80 60 70 50 90

Q2. 전체 평균 구하기

mean(score)

## [1] 70

Q3. 전체 평균 변수 만들고 출력하기

mean\_score <- mean(score) mean\_score

## [1] 70

**4. 데이터 프레임의 세계로**

**04-1. 데이터는 어떻게 생겼나? - 데이터 프레임 이해하기**

**데이터 프레임**

**벡터 : 1차원, 다차원 c()함수로 생성**

**배열 : array()함수로 생성, 행렬 만들어주는 전용함수 : metrix() 함수(행 번호와 열 번호, 인덱스 값(0번)이 없다.)**

**ex) (A,B,D열만 꺼내오려면 array[ , -3 ] or array[ , c(1,4) ]와 같이 combine 으로 묶으면 된다.**

* **배열과 벡터는 반드시 데이터가 동일해야 한다.**

**리스트 : 데이터 유형이 다양하게 들어갈 수 있다.**

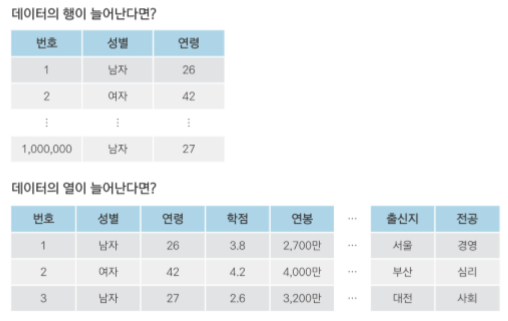
**데이터프레임 : 리스트를 여러개 쌓은 형태 ( 학생$성별 과 같이 학생 DF에 성별 컬럼을 추출)**



• '열'은 속성

• '행'은 한 사람의 정보

데이터가 크다 = 행이 많다 또는 열이 많다



**04-2. 데이터 프레임 만들기 - 시험 성적 데이터를 만들어 보자!**

1. **데이터 입력해 데이터 프레임 만들기**

english <- c(90, 80, 60, 70) # 영어 점수 변수 생성

english

## [1] 90 80 60 70

math <- c(50, 60, 100, 20) # 수학 점수 변수 생성

math

## [1] 50 60 100 20

# english, math 로 데이터 프레임 생성해서 df\_midterm 에 할당

df\_midterm <- data.frame(english, math)

df\_midterm

## english math

## 1 90 50

## 2 80 60

## 3 60 100

## 4 70 20

class <- c(1, 1, 2, 2) class

## [1] 1 1 2 2

df\_midterm <- data.frame(english, math, class)

df\_midterm

## english math class

## 1 90 50 1

## 2 80 60 1

## 3 60 100 2

## 4 70 20 2

mean(df\_midterm$english) # df\_midterm 의 english 로 평균 산출

## [1] 75

mean(df\_midterm$math) # df\_midterm 의 math 로 평균 산술

## [1] 57.5

1. **데이터 프레임 한 번에 만들기**

df\_midterm <- data.frame(english = c(90, 80, 60, 70),

math = c(50, 60, 100, 20),

class = c(1, 1, 2, 2))

df\_midterm

## english math class

## 1 90 50 1

## 2 80 60 1

## 3 60 100 2

## 4 70 20 2

**Q1. data.frame()과 c()를 조합해서 표의 내용을 데이터 프레임으로 만들어 출력해보세요.**

# 데이터 프레임 만들기

sales <- data.frame(fruit = c("사과", "딸기", "수박"),

price = c(1800, 1500, 3000),

volume = c(24, 38, 13))

# 데이터 프레임 출력하기

sales

## fruit price volume

## 1 사과 1800 24

## 2 딸기 1500 38

## 3 수박 3000 13

**Q2. 앞에서 만든 데이터 프레임을 이용해서 과일 가격 평균, 판매량 평균을 구해보세요.**

mean(sales$price) # 가격 평균

## [1] 2100

mean(sales$volume) # 판매량 평균

## [1] 25

**04-3. 외부 데이터 이용하기 - 축적된 시험 성적 데이터를 불러오자**

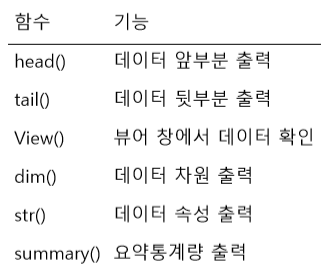
엑셀 파일 불러오기

# readxl 패키지 설치

install.packages("readxl")

# readxl 패키지 로드

library(readxl)

 x <- c("a", "a", "b", "c") --> str(x) --> chr [1:4] "a" "a" "b" "c"

**5. 데이터 분석 기초!**

**데이터 파악하기, 다루기 쉽게 수정하기**

1. **exam 데이터 파악하기**

데이터 준비

exam <- read.csv("csv\_exam.csv")

head(exam) # 앞에서부터 6행까지 출력

head(exam, 10) # 앞에서부터 10행까지 출력

tail(exam) # 뒤에서부터 6행까지 출력

tail(exam, 10) # 뒤에서부터 10행까지 출력

View(exam) # 데이터 뷰어 창에서 exam 데이터 확인

dim(exam) # 행, 열 출력

str(exam) # 데이터 속성 확인

summary(exam) # 요약 통계량 출력

View() - 뷰어 창에서 데이터 확인하기

View(exam) **[유의] View()에서 맨 앞의 V는 대문자**

dim() - 몇 행 몇 열로 구성되는지 알아보기

dim(exam) # 행 , 열 출력

## [1] 20 5

str() - 속성 파악하기

str(exam) # 데이터 속성 확인

## 'data.frame': 20 obs. of 5 variables: # 내부 구조가 데이터프레임 형태다, 몇개의 행과 몇개의 열로 구성되어있다.

# 열의 컬럼명들은 이와같이 붙어있다. 해당 컬럼값을 꺼내고 싶다면 exam$id 로 꺼내올 수 있다.

## $ id : int 1 2 3 4 5 6 7 8 9 10 ...

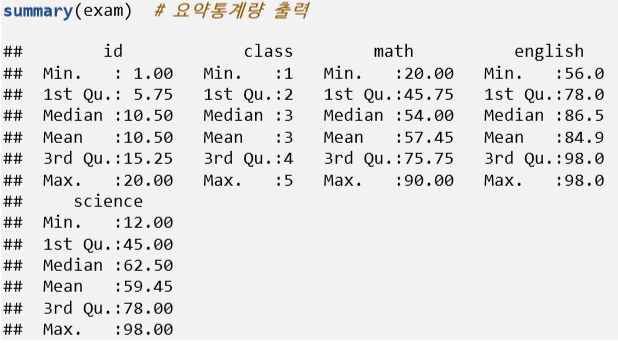
## $ class : int 1 1 1 1 2 2 2 2 3 3 ...

## $ math : int 50 60 45 30 25 50 80 90 20 50 ...

## $ english: int 98 97 86 98 80 89 90 78 98 98 ...

## $ science: int 50 60 78 58 65 98 45 25 15 45 ...

**summary() - 요약통계량 산출하기**

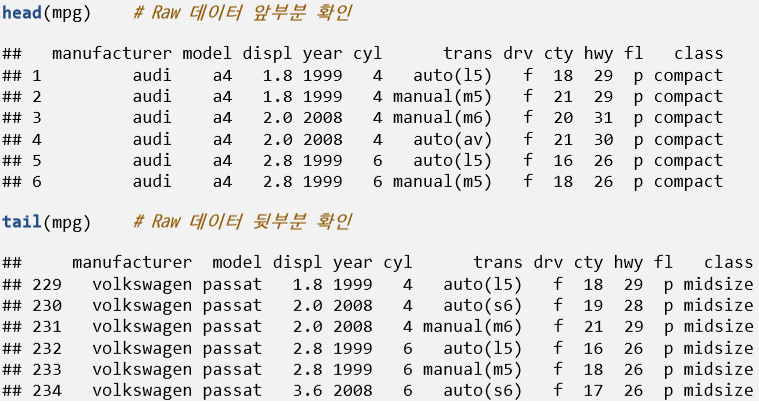


**mpg 데이터 파악하기**

# ggplo2 의 mpg 데이터를 데이터 프레임 형태로 불러오기

mpg <- as.data.frame(ggplot2::mpg)

**mpg 데이터 파악하기**

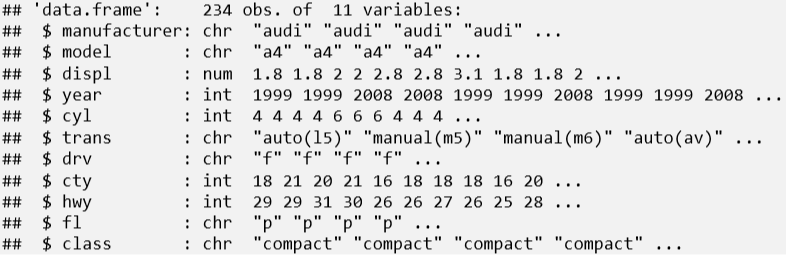


View(mpg) # Raw 데이터 뷰어 창 확인

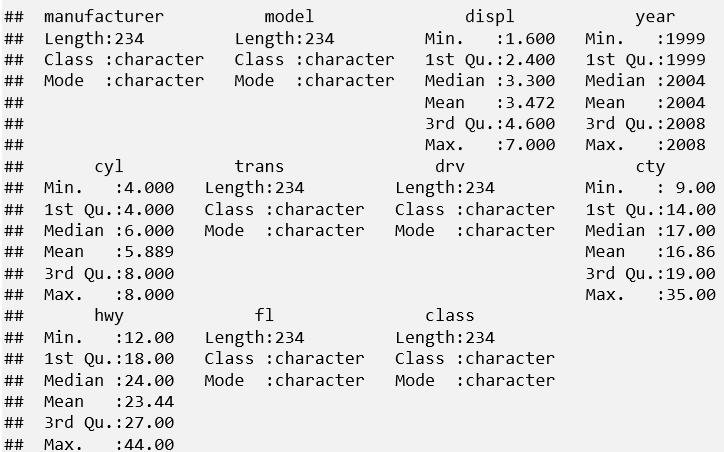
dim(mpg) # 행 , 열 출력

## [1] 234 11

str(mpg) # 데이터 속성 확인



summary(mpg) # 요약통계량 출력



**2. 데이터 수정하기 - 변수명 바꾸기**

**1) dplyr 패키지 설치 & 로드**

install.packages("dplyr") # dplyr 설치

library(dplyr) # dplyr 로드

데이터 프레임 생성

df\_raw <- data.frame(var1 = c(1, 2, 1),

var2 = c(2, 3, 2))

df\_raw

## var1 var2

## 1 1 2

## 2 2 3

## 3 1 2

df\_new <- df\_raw # 복사본 생성

df\_new # 출력

## var1 var2

## 1 1 2

## 2 2 3

## 3 1 2

**2) 변수명 바꾸기**

df\_new <- rename(df\_new, v2 = var2) # var2 를 v2 로 수정

df\_new

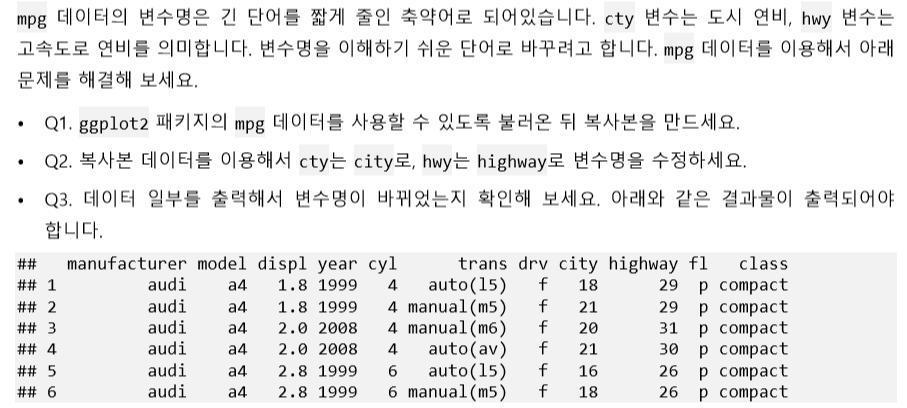
## var1 v2

## 1 1 2

## 2 2 3

## 3 1 2

[유의] rename()에 '새 변수명 = 기존 변수명' 순서로 입력



**Q1. ggplot2 패키지의 mpg 데이터를 사용할 수 있도록 불러온 뒤 복사본을 만드세요.**

mpg <- as.data.frame(ggplot2::mpg) # mpg 데이터 불러오기

mpg\_new <- mpg # 복사본 만들기

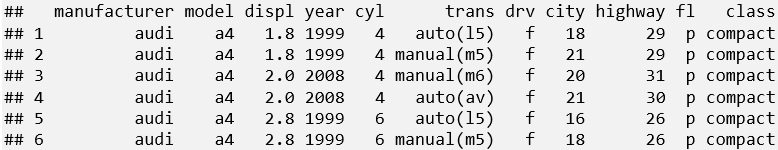
**Q2. 복사본 데이터를 이용해서 cty는 city로, hwy는 highway로 변수명을 수정하세요.**

mpg\_new <- rename(mpg\_new, city = cty) # cty 를 city 로 수정

mpg\_new <- rename(mpg\_new, highway = hwy) # hwy 를 highway 로 수정

**Q3. 데이터 일부를 출력해서 변수명이 바뀌었는지 확인해 보세요. 아래와 같은 결과물이 출력되어야 합니다.**

head(mpg\_new) # 데이터 일부 출력



1. **변수 조합해 파생변수 만들기**

**데이터 프레임 생성**

df <- data.frame(var1 = c(4, 3, 8),

var2 = c(2, 6, 1))

df

## var1 var2

## 1 4 2

## 2 3 6

## 3 8 1

**파생변수 생성 (데이터 컬럼명을 새로 추가하는 방법)**

df$var\_sum <- df$var1 + df$var2 # var\_sum 파생변수 생성

df

## var1 var2 var\_sum

## 1 4 2 6

## 2 3 6 9

## 3 8 1 9

**파생변수 생성**

df$var\_mean <- (df$var1 + df$var2)/2 # var\_mean 파생변수 생성( mean 함수는 컬럼에 대한 평균, 행에 대한평균은 안됨)

df

## var1 var2 var\_sum var\_mean

## 1 4 2 6 3.0

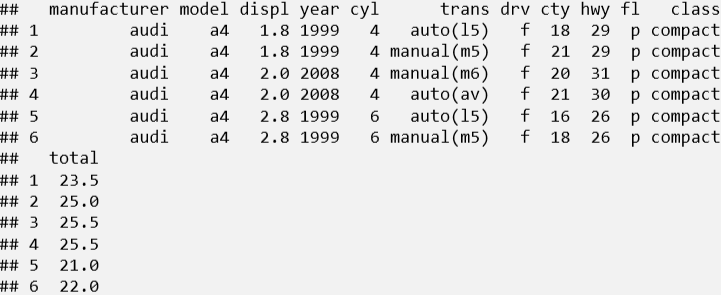
## 2 3 6 9 4.5

## 3 8 1 9 4.5

**mpg 통합 연비 변수 만들기**

mpg$total <- (mpg$cty + mpg$hwy)/2 # 통합 연비 변수 생성

head(mpg)



mean(mpg$total) # total 컬럼의 평균 구하기

## [1] 20.14957

**조건문을 활용해 파생변수 만들기**

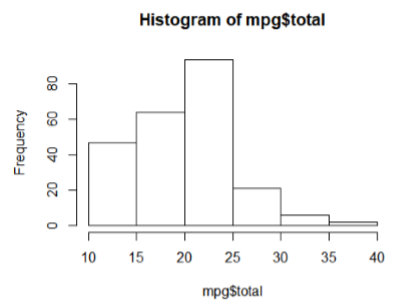
**1.기준값 정하기**

summary(mpg$total) # 요약 통계량 산출

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 10.50 15.50 20.50 20.15 23.50 39.50

hist(mpg$total) # 히스토그램 생성



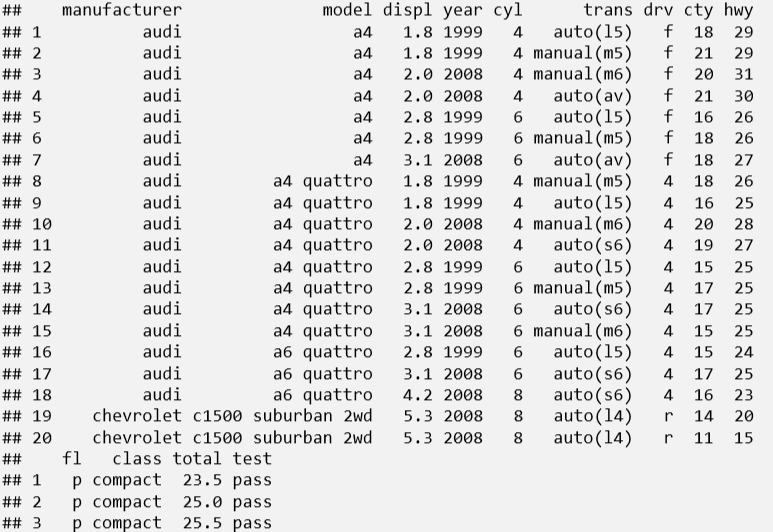
**2. 조건문으로 합격 판정 변수 만들기**

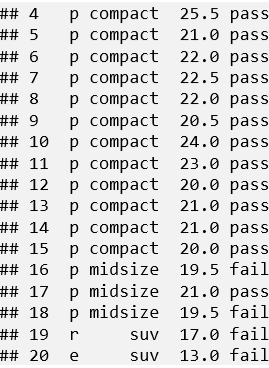


# 20 이상이면 pass, 그렇지 않으면 fail 부여

mpg$test <- ifelse(mpg$total >= 20, "pass", "fail")

head(mpg, 20) # 데이터 확인





**3. 빈도표로 합격 판정 자동차 수 살펴보기**

table(mpg$test) # 연비 합격 빈도표 생성

# fail 이 몇 개 인지, pass가 몇 개 인지 확인

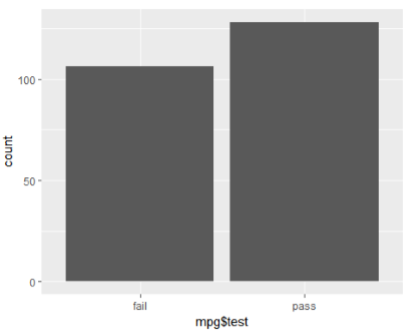
## fail pass

## 106 128

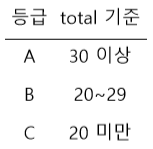
**4. 막대 그래프빈 빈도 표현하기**

library(ggplot2) # ggplot2 로드

qplot(mpg$test) # 연비 합격 빈도 막대 그래프 생성



**중첩 조건문 활용하기 - 연비 등급 변수 만들기**

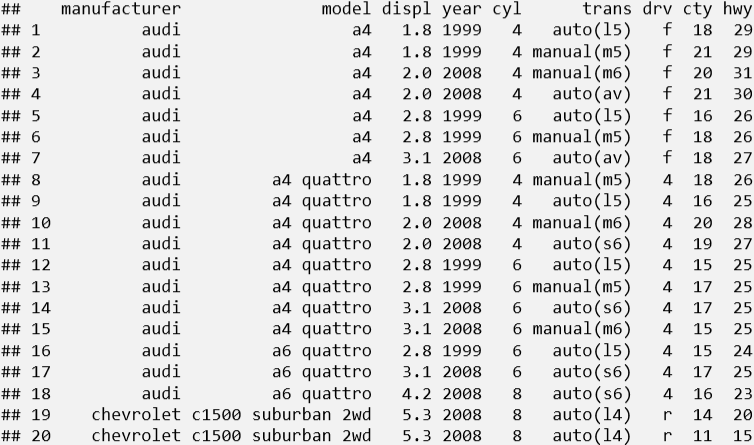
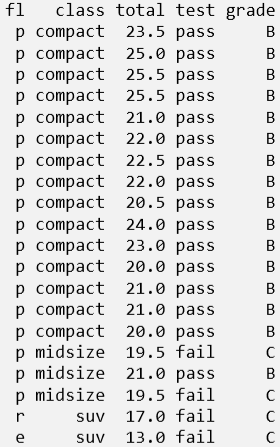


# total 을 기준으로 A, B, C 등급 부여

mpg$grade <- ifelse(mpg$total >= 30, "A",

ifelse(mpg$total >= 20, "B", "C"))

head(mpg, 20) # 데이터 확인

 **[유의] ifelse()가 두 번 반복되므로 열리는 괄호와 닫히는 괄호가 각각 두 개, 쉼표도 각각 두 개**

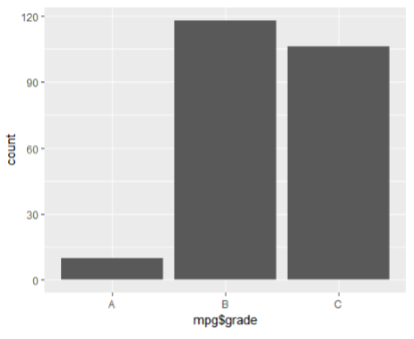
**빈도표, 막대 그래프로 연비 등급 살펴보기**

table(mpg$grade) # 등급 빈도표 생성

## A B C

## 10 118 106

qplot(mpg$grade) # 등급 빈도 막대 그래프 생성



**원하는 만큼 범주 만들기**

# A, B, C, D 등급 부여

mpg$grade2 <- ifelse(mpg$total >= 30, "A",

ifelse(mpg$total >= 25, "B",

ifelse(mpg$total >= 20, "C", "D")))

**정리하기**

**# 1. 데이터 준비 , 패키지 준비**

mpg <- as.data.frame(ggplot2::mpg) # 데이터 불러오기

library(dplyr) # dplyr 로드

library(ggplot2) # ggplot2 로드

**# 2. 데이터 파악**

head(mpg) # Raw 데이터 앞부분

tail(mpg) # Raw 데이터 뒷부분

View(mpg) # Raw 데이터 뷰어 창에서 확인

dim(mpg) # 차원

str(mpg) # 속성

summary(mpg) # 요약 통계량

분석 도전!

ggplot2 패키지에는 미국 동북중부 437개 지역의 인구통계 정보를 담은 midwest라는 데이터가 포함되어 있습니다. midwest 데이터를 사용해 데이터 분석 문제를 해결해보세요.

• 문제 1. ggplot2 의 midwest 데이터를 데이터 프레임 형태로 불러와서 데이터의 특성을 파악하세요.

• 문제 2. poptotal(전체 인구)을 total 로, popasian(아시아 인구)을 asian 으로 변수명을 수정하세요.

• 문제 3. total, asian 변수를 이용해 '전체 인구 대비 아시아 인구 백분율' 파생변수를 만들고, 히스토그램을 만들어 도시들이 어떻게 분포하는지 살펴보세요.

• 문제 4. 아시아 인구 백분율 전체 평균을 구하고, 평균을 초과하면 "large", 그 외에는 "small"을 부여하는 파생변수를 만들어 보세요.

• 문제 5. "large"와 "small"에 해당하는 지역이 얼마나 되는지, 빈도표와 빈도 막대 그래프를 만들어 확인해 보세요.

midwest <- as.data.frame(ggplot2::midwest)

midwestNew <- midwest

head(midwestNew)

str(midwestNew)

dim(midwestNew) # 437 28

summary(midwestNew)

names(midwestNew)

midwestNew <- rename(midwestNew, c(total = poptotal,asian = popasian))

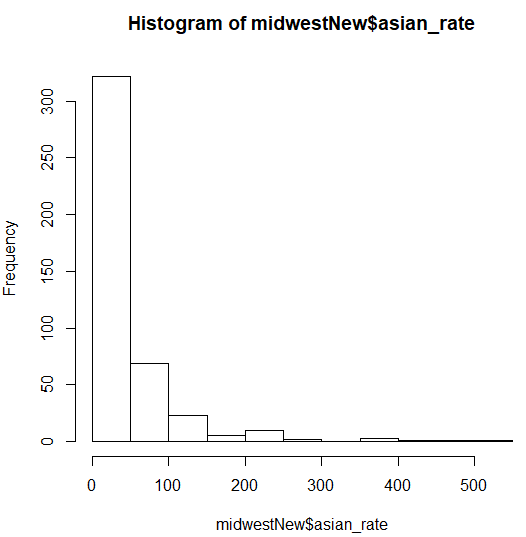
names(midwestNew) # total, asian 변경완료 확인

midwestNew$asian\_rate <- midwestNew$asian / midwestNew$total # ? \*100왜안먹힘

midwestNew$asian\_rate <- midwestNew$asian\_rate\*100

head(midwestNew$asian\_rate) # 아시아인 인구 비율 확인

hist(midwestNew$asian\_rate) # 히스토그램 생성



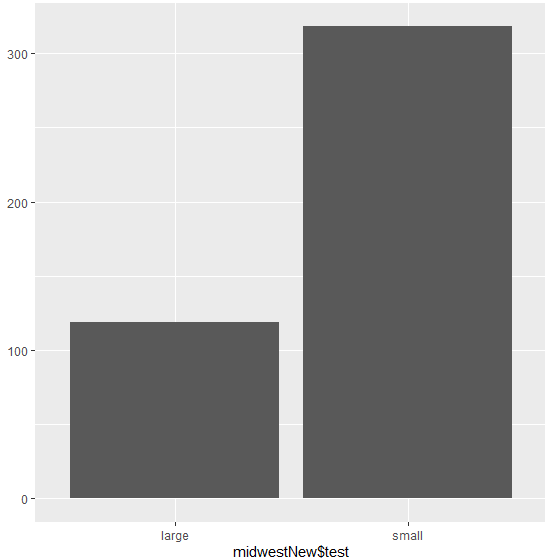
asian\_mean <- mean(midwestNew$asian\_rate) # 아시아 인구 백분율 평균 : 48.72%

midwestNew$test <- ifelse(midwestNew$asian\_rate > asian\_mean, "large", "small") # 조건문에 맞는 파생변수 삽입

library(ggplot2) #라이브러리 로드

table(midwestNew$test) # large 119, small 318

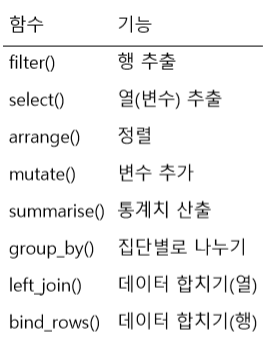
qplot(midwestNew$test)



**6. 자유자재로 데이터 가공하기**

**06-1. 데이터 전처리 - 원하는 형태로 데이터 가공하기**

**데이터 전처리(Preprocessing) - dplyr 패키지**

****

**함수 기능**

filter() 행 추출

select() 열(변수) 추출

arrange() 정렬

mutate() 변수 추가

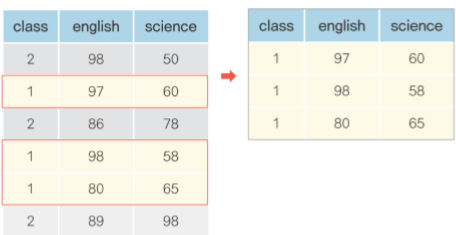
summarise() 통계치 산출

group\_by() 집단별로 나누기

left\_join() 데이터 합치기(열)

bind\_rows() 데이터 합치기(행)

**06-2. 조건에 맞는 데이터만 추출하기**

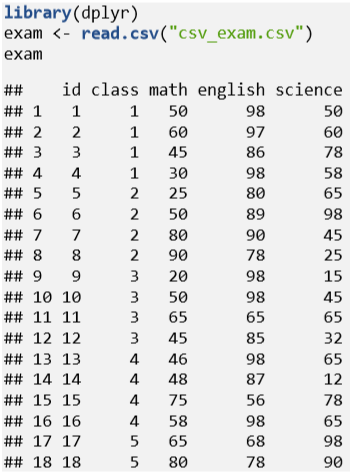


**dplyr 패키지 로드 & 데이터 준비**

**library(dplyr)**

**exam <- read.csv(file.choose(), header = T)**

**exam**

****

# exam 에서 class 가 1 인 경우만 추출하여 출력

exam %>% filter(class == 1)

**[참고] 단축키 [Ctrl+Shit+M]으로 %>% 기호 입력**

## id class math english science

## 1 1 1 50 98 50

## 2 2 1 60 97 60

## 3 3 1 45 86 78

## 4 4 1 30 98 58

# 2 반인 경우만 추출

exam %>% filter(class == 2)

## id class math english science

## 1 5 2 25 80 65

## 2 6 2 50 89 98

## 3 7 2 80 90 45

## 4 8 2 90 78 25

# 1 반이 아닌 경우

exam %>% filter(class != 1)

**초과, 미만, 이상, 이하 조건 걸기**

# 수학 점수가 50 점을 초과한 경우

exam %>% filter(math > 50)

**여러 조건을 충족하는 행 추출하기**

# 1 반 이면서 수학 점수가 50 점 이상인 경우

exam %>% filter(class == 1 & math >= 50)

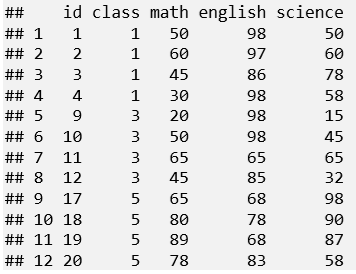
## id class math english science

## 1 1 1 50 98 50

## 2 2 1 60 97 60

**목록에 해당되는 행 추출하기**

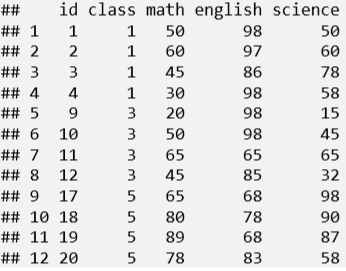
exam %>% filter(class == 1 | class == 3 | class == 5) # 1, 3, 5 반에 해당되면 추출



**%in% 기호 이용하기**

exam %>% filter(class %in% c(1,3,5)) # 1, 3, 5 반에 해당하면 추출

ex ) seq(1,100, by=2) 와 같이 적어서 긴 형태의 데이터 처리 가능



**추출한 행으로 데이터 만들기**

class1 <- exam %>% filter(class == 1) # class 가 1 인 행 추출 , class1 에 할당

class2 <- exam %>% filter(class == 2) # class 가 2 인 행 추출 , class2 에 할당

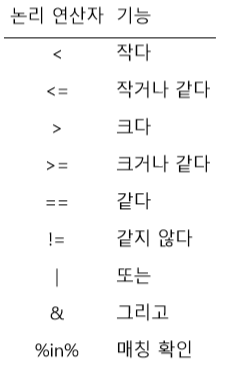
mean(class1$math) # 1 반 수학 점수 평균 구하기

## [1] 46.25

mean(class2$math) # 2 반 수학 점수 평균 구하기

## [1] 61.25

**R에서 사용하는 기호들**



**논리 연산자 기능**

< 작다

<= 작거나 같다

> 크다

>= 크거나 같다

== 같다

!= 같지 않다

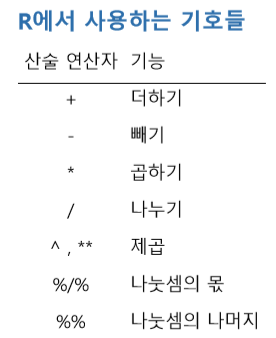
│ 또는

& 그리고

%in% 매칭 확인

**R에서 사용하는 기호들**

**산술 연산자 기능**

****

+ 더하기

- 빼기

\* 곱하기

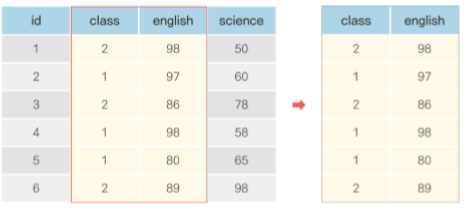
/ 나누기

^ , \*\* 제곱

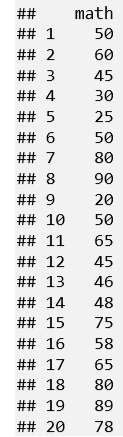
%/% 나눗셈의 몫

%% 나눗셈의 나머지

**06-3. 필요한 변수만 추출하기**



exam %>% select(math) # math 추출 ( select 함수는 열을 추출, filter 함수는 행 추출 )

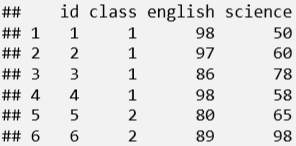


**여러 변수 추출하기**

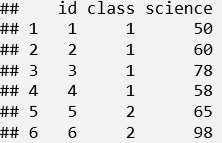
exam %>% select(class, math, english)

**변수 제외하기**

exam %>% select(-math) # math 제외



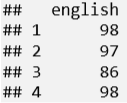
exam %>% select(-math, -english) # math, english 제외



**dplyr 함수 조합하기**

# class 가 1 인 행만 추출한 다음 english 추출

exam %>% filter(class == 1) %>% select(english) # exam 데이터를 filter로 넘기고 처리된 연산을 select로 넘긴다.



**가독성 있게 줄 바꾸기**

exam %>%

filter(class == 1) %>% # class 가 1 인 행 추출

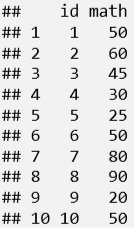
select(english) # english 추출

**일부만 출력하기**

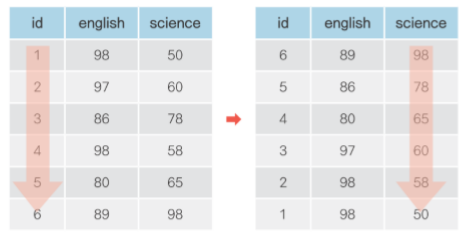
exam %>%

select(id, math) %>% # id, math 추출

head(10) # 앞부분 10 행까지 추출

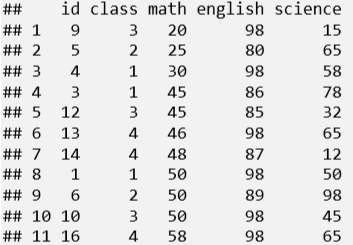


**06-4. 순서대로 정렬하기**



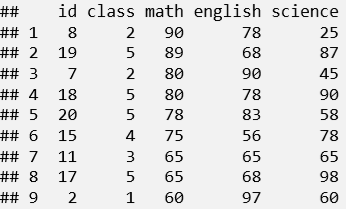
**오름차순으로 정렬하기**

exam %>% arrange(math) # math 오름차순 정렬



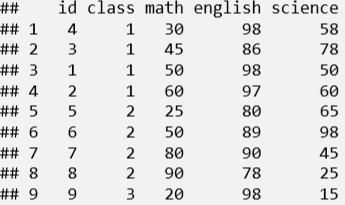
**내림차순으로 정렬하기**

exam %>% arrange(desc(math)) # math 내림차순 정렬



**정렬 기준 변수 여러개 지정**

exam %>% arrange(class, math) # class 및 math 오름차순 정렬

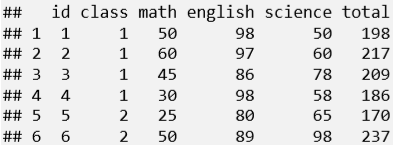


**06-5. 파생변수 추가하기**

exam %>%

mutate(total = math + english + science) %>% # 총합 변수 추가(컬럼 추가하는 함수)

head # 일부 추출



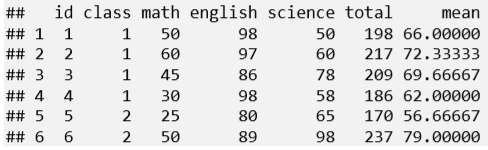
**여러 파생변수 한 번에 추가하기**

exam %>%

mutate(total = math + english + science, # 총합 변수 추가

mean = (math + english + science)/3) %>% # 총평균 변수 추가

head # 일부 추출

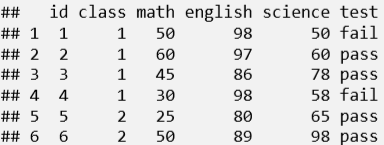


**mutate()에 ifelse() 적용하기**

exam %>%

mutate(test = ifelse(science >= 60, "pass", "fail")) %>%

head



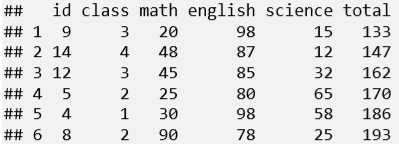
**추가한 변수를 dplyr 코드에 바로 활용하기**

exam %>%

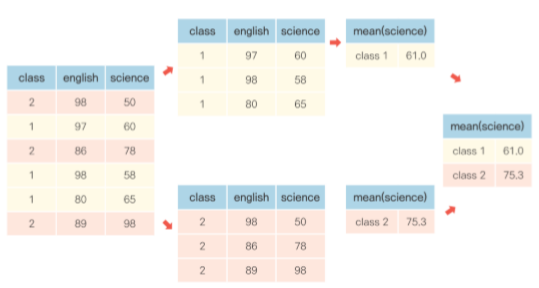
mutate(total = math + english + science) %>% # 총합 변수 추가

arrange(total) %>% # 총합 변수 기준 정렬

head # 일부 추출



**06-6. 집단별로 요약하기**



**집단별로 요약하기**

**요약하기**

exam %>% summarise(mean\_math = mean(math)) # math 평균 산출

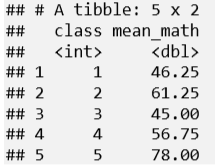
## mean\_math

## 1 57.45

**집단별로 요약하기**

exam %>%

group\_by(class) %>% # class 별로 분리 (1, 2, 3, 4 별로 묶어준 값을 summarise에 전달한다.) summarise(mean\_math = mean(math)) # math 평균 산출 (그걸로 각 매스 값들의 평균값을 넣어준다.)

dbl 은 더블의 약자(소수점 표현)

**여러 요약통계량 한 번에 산출하기**

exam %>%

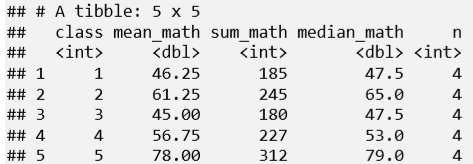
group\_by(class) %>% # class 별로 분리

summarise(mean\_math = mean(math), # math 평균

sum\_math = sum(math), # math 합계

median\_math = median(math), # math 중앙값

n = n()) # 학생 수

5x5 의미는 종류와 컬럼이 5개 라는 의미

**자주 사용하는 요약통계량 함수**

함수 의미

mean() 평균

sd() 표준편차

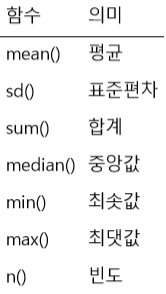
sum() 합계

median() 중앙값

min() 최솟값

max() 최댓값

n() 빈도



**각 집단별로 다시 집단 나누기**

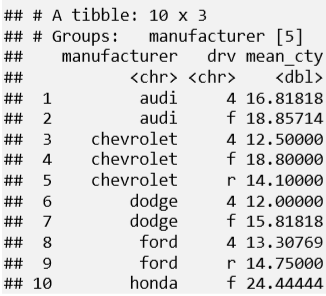
mpg %>%

group\_by(manufacturer, drv) %>% # 회사별 , 구동방식별 분리

summarise(mean\_cty = mean(cty)) %>% # cty 평균 산출

head(10) # 일부 출력

* 전체 데이터를 group\_by 함수로 넘겨서 제조사 별로 그룹을 만들어주고, 동일 제조사들이 있을텐데 그 제조사에 따라 구동방식 별로 다시 재 그룹화를 시킨다.( 4는 4륜구동, f는 전륜)
* #Groups : manufacturer(전체그룹은 제조사별로), 그 제조사 내에서 구동방식 별로 재 그룹화
* 그 데이터를 다시 summarise 함수로 전달해서 cty(도시연비)에 대한 평균값을 저장
* 데이터 일부 출력



**dplyr 조합하기**

**문제) 회사별로 "suv" 자동차의 도시 및 고속도로 통합 연비 평균을 구해 내림차순으로 정렬하고, 1~5위까지 출력하기**

**분석 절차 생각해보기**



절차 기능 dplyr 함수

1 회사별로 분리 group\_by()

2 suv 추출 filter() # 행 추출을 위해 select 가 아닌 filter 사용

3 통합 연비 변수 생성 mutate() # 통합 연비 컬럼을 생성

4 통합 연비 평균 산출 summarise() # 계산은 서머라이즈로 추출

5 내림차순 정렬 arrange() # 내림차순 정렬 후 출력

6 1~5위까지 출력 head()

**dplyr 조합하기**

mpg %>%

group\_by(manufacturer) %>% # 회사별로 분리

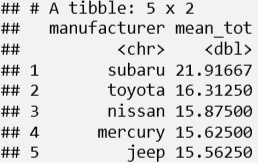
filter(class == "suv") %>% # suv 추출

mutate(tot = (cty+hwy)/2) %>% # 통합 연비 변수 생성

summarise(mean\_tot = mean(tot)) %>% # 통합 연비 평균 산출

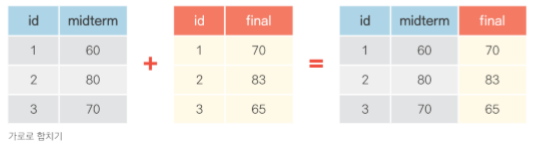
arrange(desc(mean\_tot)) %>% # 내림차순 정렬

head(5) # 1~5 위까지 출력

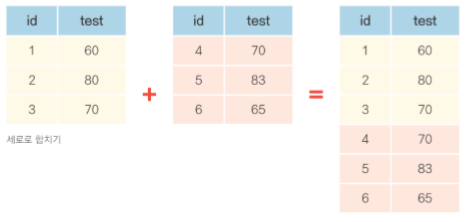
 상위 5개 연비만 출력

**06-7. 데이터 합치기**

**가로로 합치기**



**세로로 합치기**



**가로로 합치기**

**데이터 생성**

# 중간고사 데이터 생성

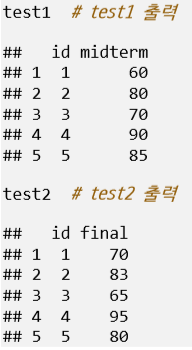
test1 <- data.frame(id = c(1, 2, 3, 4, 5),

midterm = c(60, 80, 70, 90, 85))

# 기말고사 데이터 생성

test2 <- data.frame(id = c(1, 2, 3, 4, 5),

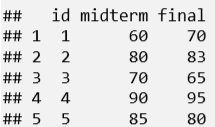
final = c(70, 83, 65, 95, 80))



**id 기준으로 합치기**

total <- left\_join(test1, test2, by = "id") # id 기준으로 합쳐 total 에 할당 (옵션값으로 지정하려면 따옴표로 묶어야 함)

total # total 출력



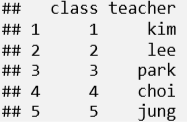
**[주의] by에 변수명을 지정할 때 변수명 앞 뒤에 쌍따옴표 입력**

**다른 데이터 활용해 변수 추가하기**

**반별 담임교사 명단 생성**

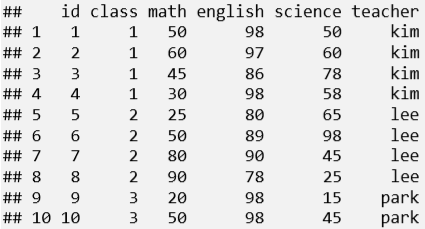
name <- data.frame(class = c(1, 2, 3, 4, 5),

teacher = c("kim", "lee", "park", "choi", "jung"))

****

**class 기준 합치기**

exam\_new <- left\_join(exam, name, by = "class") exam\_new



**세로로 합치기**

**데이터 생성**

# 학생 1~5 번 시험 데이터 생성

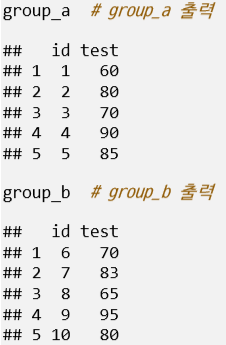
group\_a <- data.frame(id = c(1, 2, 3, 4, 5),

test = c(60, 80, 70, 90, 85))

# 학생 6~10 번 시험 데이터 생성

group\_b <- data.frame(id = c(6, 7, 8, 9, 10),

test = c(70, 83, 65, 95, 80))



**세로로 합치기**

group\_all <- bind\_rows(group\_a, group\_b) # 데이터 합쳐서 group\_all 에 할당

group\_all # group\_all 출력



**7. 데이터 정제**

**빠진 데이터, 이상한 데이터 제거하기**

**07-1. 빠진 데이터를 찾아라! - 결측치 정제하기**

**결측치(Missing Value)**

• 누락된 값, 비어있는 값

• 함수 적용 불가, 분석 결과 왜곡

• 제거 후 분석 실시

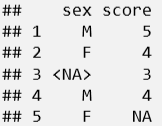
결측치 만들기

• 결측치 표기 - 대문자 NA

df <- data.frame(sex = c("M", "F", NA, "M", "F"),

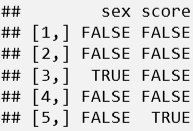
score = c(5, 4, 3, 4, NA))

df

**[유의] NA 앞 뒤에 겹따옴표 없음**

**결측치 확인하기**

is.na(df) # 결측치 확인



table(is.na(df)) # 결측치 빈도 출력



table(is.na(df$sex)) # sex 결측치 빈도 출력



table(is.na(df$score)) # score 결측치 빈도 출력



**변수별로 결측치 확인하기**

table(is.na(df$sex)) # sex 결측치 빈도 출력

## FALSE TRUE

## 4 1

table(is.na(df$score)) # score 결측치 빈도 출력

## FALSE TRUE

## 4 1

**결측치 포함된 상태로 분석**

mean(df$score) # 평균 산출

## [1] NA

sum(df$score) # 합계 산출

## [1] NA

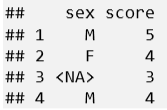
**결측치 있는 행 제거하기**

library(dplyr) # dplyr 패키지 로드

df %>% filter(is.na(score)) # score 가 NA 인 데이터만 출력



df %>% filter(!is.na(score)) # score 결측치 제거



**결측치 제외한 데이터로 분석하기**

df\_nomiss <- df %>% filter(!is.na(score)) # score 결측치 제거

mean(df\_nomiss$score) # score 평균 산출

## [1] 4

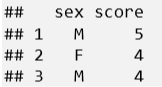
sum(df\_nomiss$score) # score 합계 산출

## [1] 16

**여러 변수 동시에 결측치 없는 데이터 추출하기**

# score, sex 결측치 제외

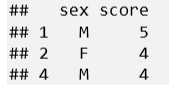
df\_nomiss <- df %>% filter(!is.na(score) & !is.na(sex)) df\_nomiss



**결측치가 하나라도 있으면 제거하기**

df\_nomiss2 <- na.omit(df) # 모든 변수에 결측치 없는 데이터 추출

df\_nomiss2 # 출력



• 분석에 필요한 데이터까지 손실 될 가능성 유의

• ex) 성별-소득 관계 분석하는데 지역 결측치까지 제거

**함수의 결측치 제외 기능 이용하기 - na.rm = T**

mean(df$score, na.rm = T) # 결측치 제외하고 평균 산출

## [1] 4

sum(df$score, na.rm = T) # 결측치 제외하고 합계 산출

## [1] 16

**summarise()에서 na.rm = T사용하기 • 결측치 생성**

exam <- read.csv("csv\_exam.csv") # 데이터 불러오기

exam[c(3, 8, 15), "math"] <- NA # 3, 8, 15 행의 math 에 NA 할당

• 평균 구하기

exam %>% summarise(mean\_math = mean(math)) # 평균 산출

## mean\_math

## 1 NA

exam %>% summarise(mean\_math = mean(math, na.rm = T)) # 결측치 제외하고 평균 산출

## mean\_math

## 1 55.23529

**다른 함수들에 적용**

exam %>% summarise(mean\_math = mean(math, na.rm = T), # 평균 산출

sum\_math = sum(math, na.rm = T), # 합계 산출

median\_math = median(math, na.rm = T)) # 중앙값 산출



**결측치 대체하기**

**• 결측치 많을 경우 모두 제외하면 데이터 손실 큼**

**• 대안: 다른 값 채워넣기**

**결측치 대체법(Imputation)**

**• 대표값(평균, 최빈값 등)으로 일괄 대체**

**• 통계분석 기법 적용, 예측값 추정해서 대체**

**평균값으로 결측치 대체하기**

**평균 구하기**

mean(exam$math, na.rm = T) # 결측치 제외하고 math 평균 산출

## [1] 55.23529

**평균으로 대체하기**

exam$math <- ifelse(is.na(exam$math), 55, exam$math) # math 가 NA 면 55 로 대체

* 만약 exam 데이터 프레임 안에 math 컬럼 값 중에, 결측값이 발견 되면(TRUE), 그 부분은 55 셋팅, 그렇지 않으면 exam$math (원래 값 셋팅)

table(is.na(exam$math)) # 결측치 빈도표 생성

##

## FALSE

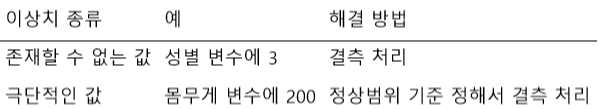
## 20

**07-2. 이상한 데이터를 찾아라! - 이상치 정제하기**

**이상치(Outlier) - 정상범주에서 크게 벗어난 값**

**• 이상치 포함시 분석 결과 왜곡 (왜곡점이 만들어 진다.)**

**• 결측 처리 후 제외하고 분석**



**이상치 제거하기 - 1. 존재할 수 없는 값**

**• 논리적으로 존재할 수 없으므로 바로 결측 처리 후 분석시 제외**

이상치 포함된 데이터 생성 - sex 3, score 6

outlier <- data.frame(sex = c(1, 2, 1, 3, 2, 1),

score = c(5, 4, 3, 4, 2, 6))

outlier



**이상치 확인하기**

table(outlier$sex) #빈도수 확인

##

## 1 2 3

## 3 2 1

table(outlier$score)

##

## 2 3 4 5 6

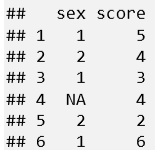
## 1 1 2 1 1

**결측 처리하기 - sex**

# sex 가 3 이면 NA 할당

outlier$sex <- ifelse(outlier$sex == 3, NA, outlier$sex)

outlier

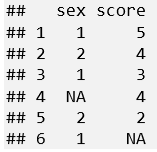


**결측 처리하기 - score**

# sex 가 1~5 아니면 NA 할당

outlier$score <- ifelse(outlier$score > 5, NA, outlier$score)

outlier



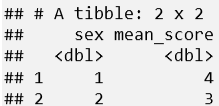
**결측치 제외하고 분석**

outlier %>%

filter(!is.na(sex) & !is.na(score)) %>%

group\_by(sex) %>%

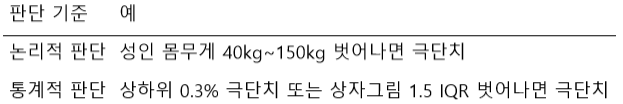
summarise(mean\_score = mean(score))

 결측값을 제외하여 연산처리

* 과정별로 실행하는 방법 : 원하는 부분까지만 선택해서 실행시키면 단계별 결과값을 추출할 수 있다.
* 아니면 중간중간 변수처리 하여 결과값 확인 가능

**이상치 제거하기 - 2. 극단적인 값**

**• 정상범위 기준 정해서 벗어나면 결측 처리**

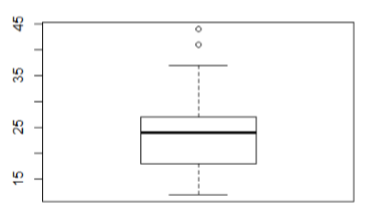
****

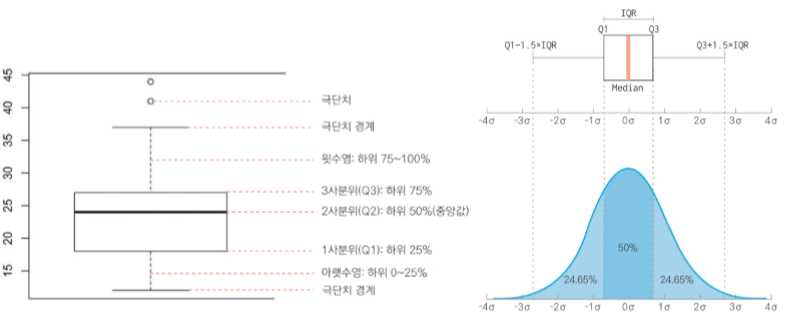
**상자그림으로 극단치 기준 정해서 제거하기 (base 패키지)**

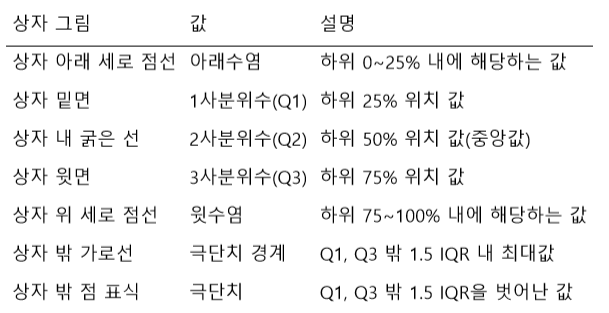
**상자그림 생성**

mpg <- as.data.frame(ggplot2::mpg) # ggplot2 는 qplot 함수

boxplot(mpg$hwy) # 이건 기본적으로 설치되어있는 차트

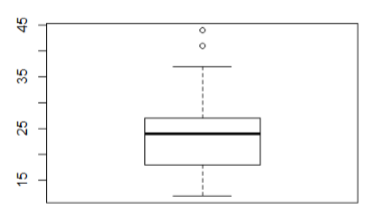


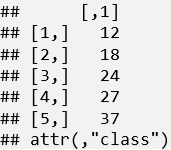




**상자그림 통계치 출력**

boxplot(mpg$hwy)$stats # 상자그림 통계치 출력





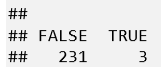


**결측 처리하기**

# 12~37 벗어나면 NA 할당

mpg$hwy <- ifelse(mpg$hwy < 12 | mpg$hwy > 37, NA, mpg$hwy)

table(is.na(mpg$hwy))

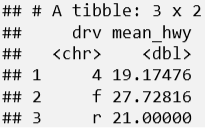


결측치 제외하고 분석하기

mpg %>%

group\_by(drv) %>%

summarise(mean\_hwy = mean(hwy, na.rm = T))



**정리하기**

**# 1. 결측치 정제하기**

# 결측치 확인

table(is.na(df$score))

# 결측치 제거

df\_nomiss <- df %>% filter(!is.na(score))

# 여러 변수 동시에 결측치 제거

df\_nomiss <- df %>% filter(!is.na(score) & !is.na(sex))

# 함수의 결측치 제외 기능 이용하기

mean(df$score, na.rm = T) exam %>% summarise(mean\_math = mean(math, na.rm = T))

**# 2. 이상치 정제하기**

# 이상치 확인

table(outlier$sex)

# 결측 처리

outlier$sex <- ifelse(outlier$sex == 3, NA, outlier$sex)

# boxplot 으로 극단치 기준 찾기

boxplot(mpg$hwy)$stats

# 극단치 결측 처리

mpg$hwy <- ifelse(mpg$hwy < 12 | mpg$hwy > 37, NA, mpg$hwy)

**08. 그래프 만들기**

**08-1. R로 만들 수 있는 그래프 살펴보기**

• 2차원 그래프, 3차원 그래프

• 지도 그래프 • 네트워크 그래프

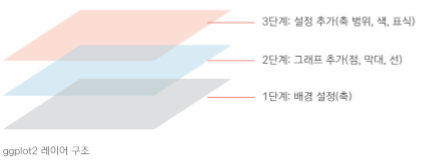
• 모션 차트

• 인터랙티브 그래프

**쉽게 그래프를 만들 수 있는 ggplot2 패키지**

**08-2. 산점도 - 변수 간 관계 표현하기**

**ggplot2 레이어 구조 이해하기**



**산점도 만들기**

• 산점도(Scater Plot) : 데이터를 x축과 y축에 점으로 표현한 그래프

• 나이와 소득처럼, 연속 값으로 된 두 변수의 관계를 표현할 때 사용

**ggplot2 로드**

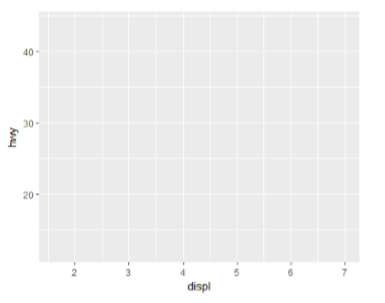
library(ggplot2)

**1. 배경 설정하기**

# x 축 displ, y 축 hwy 로 지정해 배경 생성

ggplot(data = mpg, aes(x = displ, y = hwy))

# aes : 축 만들어주는 함수(x축 값, y축 값 지정, data = mpg 데이터프레임 사용하겠다고 지정

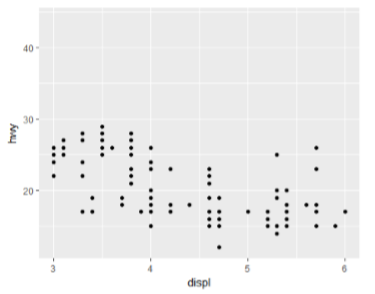


**2. 그래프 추가하기**

# 배경에 산점도 추가

ggplot(data = mpg, aes(x = displ, y = hwy)) + geom\_point()

# 지오메트릭 포인트 함수 추가



**3. 축 범위를 조정하는 설정 추가하기**

# x 축 범위 3~6, y 축 범위 10~30 으로 지정

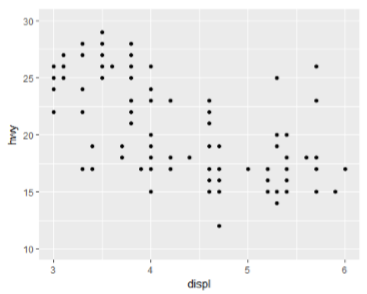
ggplot(data = mpg, aes(x = displ, y = hwy)) +

geom\_point() +

xlim(3, 6) +

ylim(10, 30)

# x, y축에 대한 시작 값과 끝 값을 지정하여 산점도 정렬



**ggplot 함수 구조**



**ggplot2 코드 가독성 높이기**

**• 한 줄로 작성**

ggplot(data = mpg, aes(x = displ, y = hwy)) + geom\_point() + xlim(3, 6) + ylim(10, 30)

**• + 뒤에서 줄 바꾸기**

ggplot(data = mpg, aes(x = displ, y = hwy)) +

geom\_point() +

xlim(3, 6) +

ylim(10, 30)

**ggplot() vs qplot()**

• qplot() : 전처리 단계 데이터 확인용 문법 간단, 기능 단순

• ggplot() : 최종 보고용. 색, 크기, 폰트 등 세부 조작 가능

**08-3. 막대 그래프 - 집단 간 차이 표현하기**

• 막대 그래프(Bar Chart) : 데이터의 크기를 막대의 길이로 표현한 그래프

• 성별 소득 차이처럼 **집단 간 차이**를 표현할 때 주로 사용

**막대 그래프 1 - 평균 막대 그래프 만들기**

**• 각 집단의 평균값을 막대 길이로 표현한 그래프**

**1. 집단별 평균표 만들기**

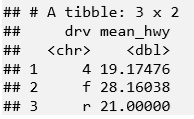
library(dplyr)

df\_mpg <- mpg %>% # 데이터 프레임 mpg 로부터

group\_by(drv) %>% # 구동 방식을 그룹핑 시켰다

summarise(mean\_hwy = mean(hwy)) # 그 중 고속도로 연비의 평균을 mean\_hwy 에 새로 만들어 넣고 통계치 추출

df\_mpg

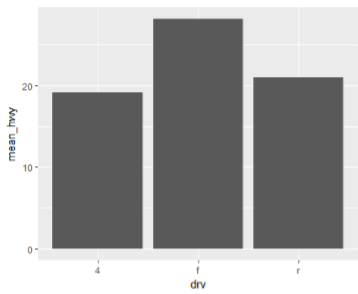


4륜의 고속도로 평균값, 전륜, 후륜의 고속도로 평균값 추출

**2. 그래프 생성하기**

ggplot(data = df\_mpg, aes(x = drv, y = mean\_hwy)) + geom\_col()

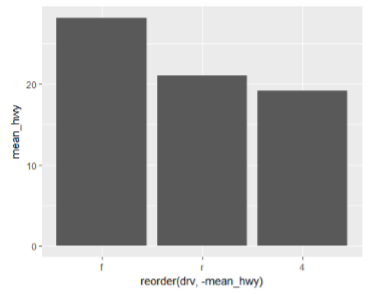
# 점이 아닌 막대기로 찍어야 하니, geom\_col() 으로 설정한다.



**3. 크기 순으로 정렬하기 (가장 큰 데이터부터 작은 데이터 순으로 정렬) – reorder 함수**

ggplot(data = df\_mpg, aes(x = reorder(drv, -mean\_hwy), y = mean\_hwy)) + geom\_col()

**? –mean\_hwy 에서 –를 제외하고 실행하면 작은순으로 정렬된다.**



**막대 그래프 2 - 빈도 막대 그래프**

• 값의 개수(빈도)로 막대의 길이를 표현한 그래프

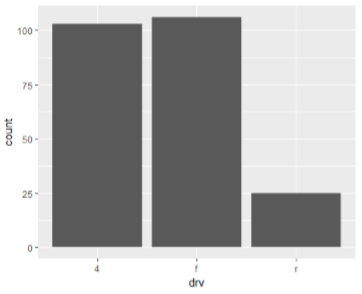
# x 축 범주 변수 , y 축 빈도 ( 각 구동방식 별로 몇개가 있는지 확인 )

ggplot(data = mpg, aes(x = drv)) + geom\_bar()

# geom\_col() 이 아닌 geom\_bar() 를 사용해 막대그래프 작성

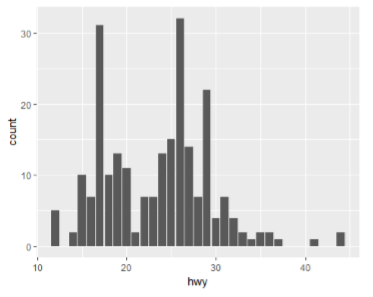
# geom\_col() 은 데이터 갯수만큼 막대기를 그려준다.

# geom\_bar() 는 동일한 데이터가 몇개 있는지 막대그래프를 그려준다.



# x 축 연속 변수 , y 축 빈도

ggplot(data = mpg, aes(x = hwy)) + geom\_bar()



**08-4. 선 그래프 - 시간에 따라 달라지는 데이터 표현하기**

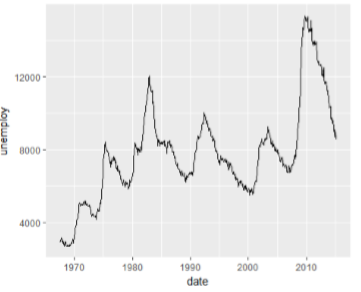
**• 선 그래프(Line Chart) : 데이터를 선으로 표현한 그래프**

**• 시계열 그래프(Time Series Chart) : 일정 시간 간격을 두고 나열된 시계열 데이터(Time Series Data)를 선으로 표현한 그래프. 환율, 주가지수 등 경제 지표가 시간에 따라 어떻게 변하는지 표현할 때 활용**

**시계열 그래프 만들기**

ggplot(data = economics, aes(x = date, y = unemploy)) + geom\_line()

# x축은 날짜 컬럼, y축은 실업자, 데이터와 데이터 사이는 선으로 연결시키겠다 : geom\_line()



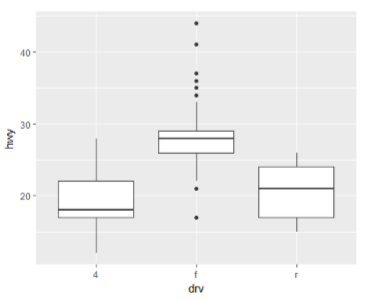
**08-5. 상자 그림 - 집단 간 분포 차이 표현하기**

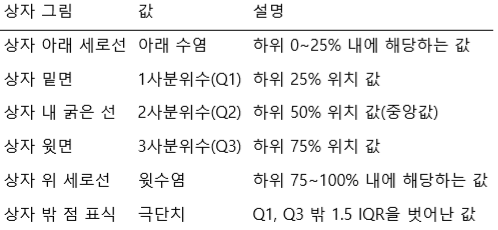
• 상자 그림(Box Plot) : 데이터의 분포(퍼져 있는 형태)를 직사각형 상자 모양으로 표현한 그래프

• 분포를 알 수 있기 때문에 평균만 볼 때보다 데이터의 특성을 좀 더 자세히 이해할 수 있음

**상자 그림 만들기**

ggplot(data = mpg, aes(x = drv, y = hwy)) + geom\_boxplot()





**참고 1.5 IQR: 사분위 범위(Q1~Q3간 거리)의 1.5배**

앞에서 다룬 ggplot2 함수들

값 내용

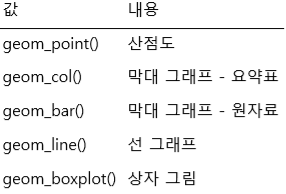
geom\_point() 산점도

geom\_col() 막대 그래프 - 요약표

geom\_bar() 막대 그래프 - 원자료

geom\_line() 선 그래프

geom\_boxplot() 상자 그림



**정리하기**

**# 1. 산점도**

ggplot(data = mpg, aes(x = displ, y = hwy)) + geom\_point()

# 축 설정 추가

ggplot(data = mpg, aes(x = displ, y = hwy)) + geom\_point() + xlim(3, 6) + ylim(10, 30)

**# 2. 평균 막대 그래프**

# 1 단계 . 평균표 만들기

df\_mpg <- mpg %>% group\_by(drv) %>% summarise(mean\_hwy = mean(hwy))

# 2 단계 . 그래프 생성하기 , 크기순 정렬하기

ggplot(data = df\_mpg, aes(x = reorder(drv, -mean\_hwy), y = mean\_hwy)) + geom\_col()

**# 3. 빈도 막대 그래프**

ggplot(data = mpg, aes(x = drv)) + geom\_bar()

**# 4. 선 그래프**

ggplot(data = economics, aes(x = date, y = unemploy)) + geom\_line()

**# 5. 상자 그림**

ggplot(data = mpg, aes(x = drv, y = hwy)) + geom\_boxplot()

09-1. ‘한국복지패널데이터’ 분석 준비하기

**데이터 분석 준비하기**

**패키지 준비하기**

install.packages(“foreign”) # foreign 패키지 설치

install.packages("readxl") # readxl 패키지 설치

library(foreign) # SPSS 파일 로드

library(dplyr) # 전처리

library(ggplot2) # 시각화

library(readxl) # 엑셀파일 불러오기

데이터 준비하기

**# 데이터 불러오기**

raw\_welfare <- read.spss(file = “Koweps\_hpc10\_2015\_beta1.sav”,

to.data.frame =T)

**# 복사본 만들기**

welfare <- raw\_welfare

**# 데이터 검토하기**

head(welfare)

tail(welfare)

View(welfare)

str(welfare)

summary(welfare)

dim(welfare) # 16664 행, 957 열

sum(is.na(welfare)) # 9733535

**#변수명 바꾸기**

welfare <- rename(welfare,

sex = h10\_g3, #성별

birth = h10\_g4, #태어난 연도

marrige = h10\_g10, # 혼인 상태

religion = h10\_g11, # 종교

income = p1002\_8aq1, # 월급

code\_job = h10\_eco9, # 직종 코드

code\_region = h10\_reg7) # 지역 코드

names(welfare)# 변경된 컬럼명 확인

**데이터 분석 절차**

* 1단계. 변수 검토 및 전처리
* 코드북에 보면 성별은 1과 2로 정렬해야 한다(3이 들어간 경우 이상치 값으로 정제 필요)
* 2단계. 변수 간 관계 분석

**09-2. 성별에 따른 월급 차이**

* “성별에 따라 월급이 다를까?”

**분석 절차**

1. **변수 검토 및 전처리**

* 성별
* 월급

1. **변수 간 관계 분석**

* 성별 월급 평균표 만들기
* 그래프 만들기

**성별 변수 검토 및 전처리**

1. **변수 검토하기**

class(welfare$sex) # numeric 타입 확인

table(welfare$sex) # 1: 7578, 2: 9086 확인(남녀 빈도수)

**# 이상치 결측 처리**

welfare$sex <- ifelse(welfare$sex == 0, NA, welfare$sex)

**# 결측치 확인**

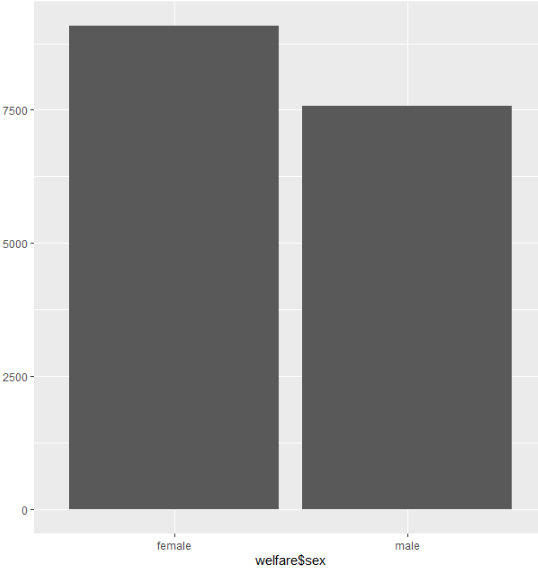
table(is.na(welfare$sex)) # False : 16664

**#성별 항목 이름 부여**

welfare$sex <- ifelse(welfare$sex ==1, "male", "female")

table(welfare$sex)

qplot(welfare$sex) # 간단한 전처리 확인( ggplot2 패키지 내부에 존재 )



**월급 변수 검토 및 전처리**

1. **변수 검토하기**

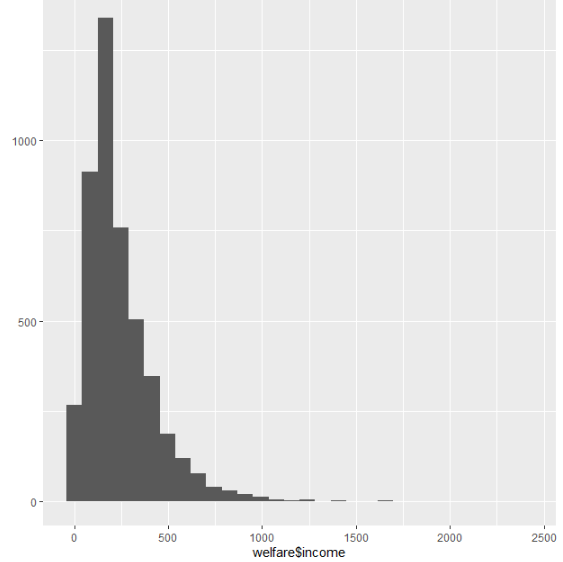
class(welfare$income) # 데이터 타입 numeric 값 확인

summary(welfare$income)

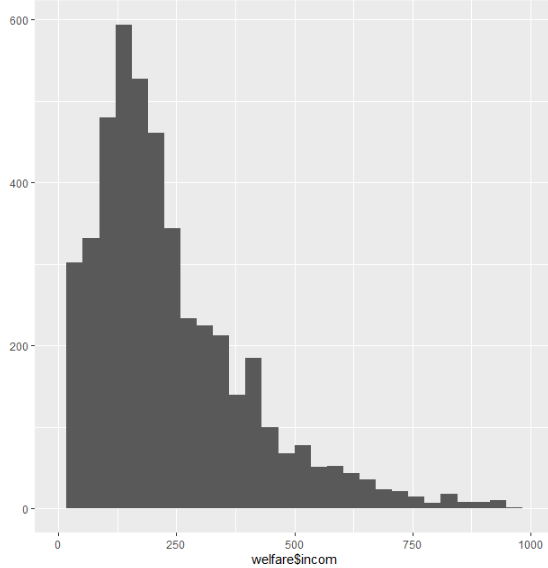


* NA 값 전처리 필요

qplot(welfare$income)

**결측값이 있기 때문에 표현이 안 된다.(2400이 없음)**

qplot(welfare$incom) + xlim(0, 1000)



**# 이상치 결측 처리**

welfare$income <- ifelse(welfare$income %in% c(0, 9999), NA, welfare$income)

--> welfare$income 에 0이 있거나 9999라는 값이 있으면 NA 처리해라 아니면 welfare$income 값을 그대로 넣어라

**# 결측치 확인**

table(is.na(welfare$income)) # False : 4620, True : 12044

1. **성별 월급 평균표 만들기**

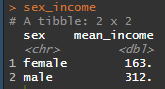
sex\_income <- welfare %>%

filter(!is.na(income)) %>% # 월급 컬럼값이 na가 아닌애들만 걸러낸다.

group\_by(sex) %>% # 그 데이터만 다시 그룹바이 함수에 넘긴다.(남자, 여자로 묶어준다)

summarise(mean\_income = mean(income)) # 그룹화된 값들의 평균을 구해서(남녀) 그 결과를 sex\_income 에 할당시킨다.

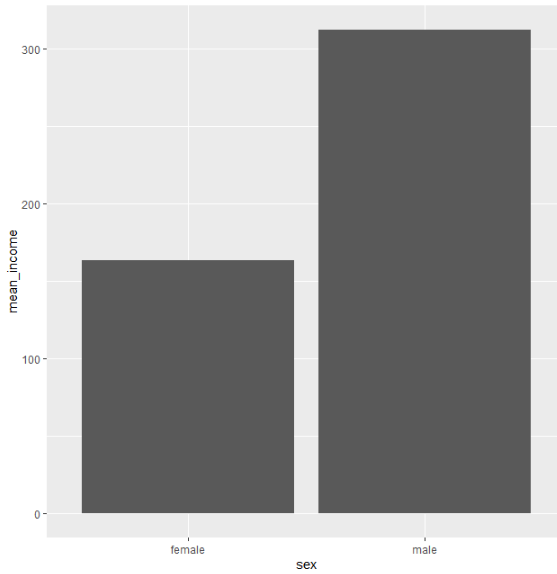
sex\_income # female : 163, male : 312 (sex\_mean)



1. **그래프 만들기(ggplot2 패키지)**

ggplot(data = sex\_income, aes(x=sex, y=mean\_income)) + geom\_col()

* data는 sex\_income 데이터, aes 함수로 x값, y 값 지정, geom\_col 로 막대그래프 그리기



**09-2. 성별에 따른 월급 차이**

* “몇살 때 월급을 가장 많이 받을까?”

분석 절차

1. 변수 검토 및 전처리

* 나이
* 월급

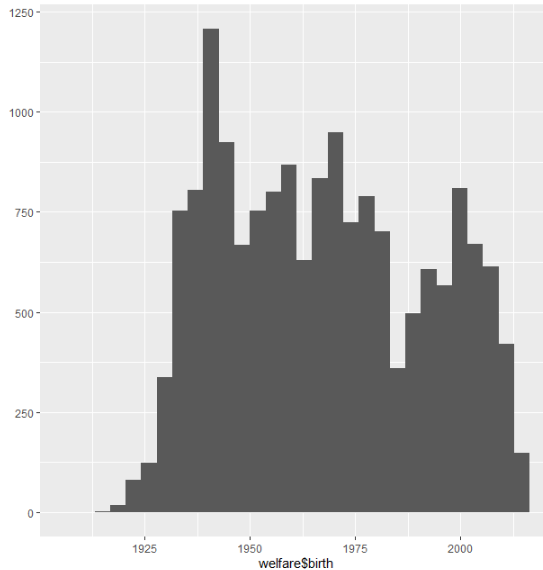
1. 변수 간 관계 분석
2. 변수 검토하기

class(welfare$birth) # numeric

summary(welfare$birth)



qplot(welfare$birth)



**# 결측치 확인**

table(is.na(welfare$birth)) # False 16664

**#이상치 결측 처리**

welfare$birth <- ifelse(welfare$birth == 9999, NA, welfare$birth)

table(is.na(welfare$birth)) # False : 16664

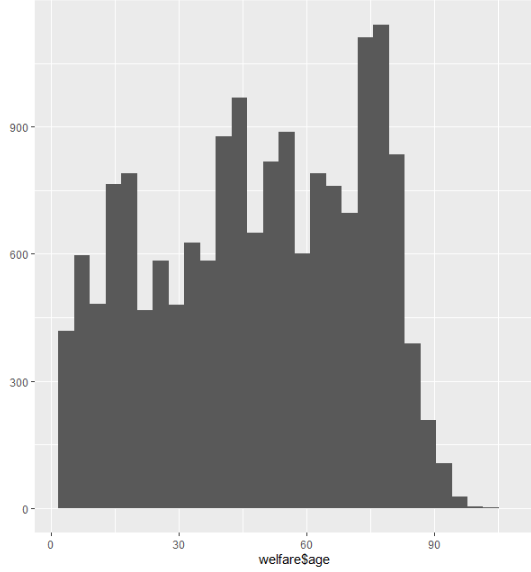
1. **파생변수 만들기 – 나이**

welfare$age <- 2015 - welfare$birth +1

summary(welfare$age)



qplot(welfare$age)



**나이와 월급의 관계 분석하기**

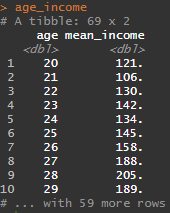
1. **나이에 따른 월급 평균표 만들기**

age\_income <- welfare %>%

filter(!is.na(income)) %>% # na 값이 아닌애들만 걸러낸다.

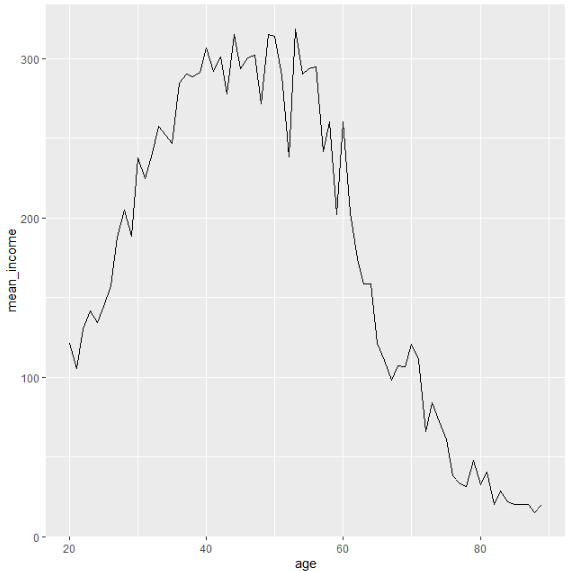
group\_by(age) %>% # 걸러낸 내용을 토대로 age 별로 그룹화한다.

summarise(mean\_income = mean(income)) # summarise 함수로 income의 평균값을 age\_income 변수에 할당



1. **그래프 만들기**

ggplot(data = age\_income, aes(x=age, y=mean\_income)) + geom\_line()



**09-4. 연령대에 따른 월급 차이**

* **“어떤 연령대의 월급이 가장 많을까?”**

**분석 절차**

1. **변수 검토 및 전처리**

* 연령대
* 월급

1. **변수 간 관계 분석**

* 연령대별 월급 평균표 만들기
* 그래프 만들기

**파생변수 만들기 – 연령대**

welfare <- welfare %>%

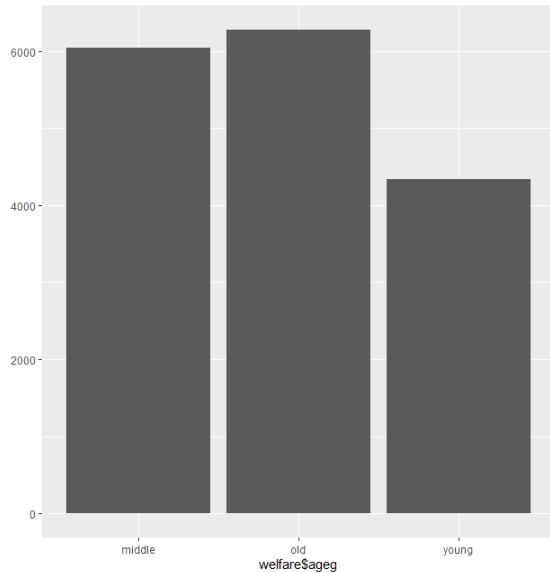
mutate(ageg = ifelse(age < 30, "young",

ifelse(age <= 59, "middle", "old")))

table(welfare$ageg)



qplot(welfare$ageg)



1. **연령대별 월급 평균표 만들기**

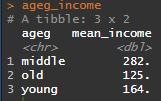
ageg\_income <- welfare %>%

filter(!is.na(income)) %>%

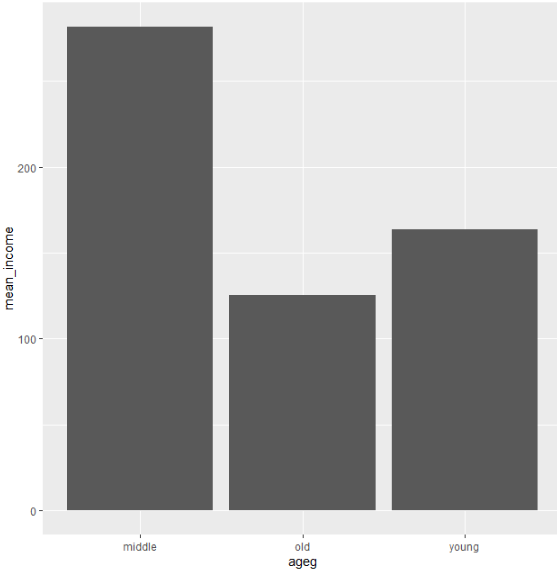
group\_by(ageg) %>%

summarise(mean\_income = mean(income))

ageg\_income



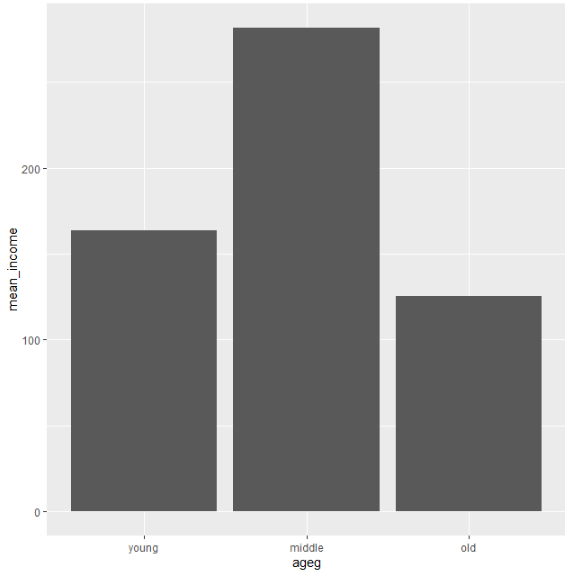
1. **그래프 만들기**

ggplot(data = ageg\_income, aes(x=ageg, y=mean\_income)) + geom\_col()

**막대 정렬 : 초년, 중년, 노년 나이 순**

ggplot(data = ageg\_income, aes(x = ageg, y= mean\_income)) + geom\_col() +

scale\_x\_discrete(limits = c("young", "middle", "old"))



**09-5. 연령대 및 성별 월급 차이**

* **“성별 월급 차이는 연령대별로 다를까?”**

**분석 절차**

1. **변수 검토 및 전처리**

* 연령대 (ageg)
* 성별 (sex)
* 월급 (income)

1. **변수 간 관계 분석**

* 연령대 및 성별 월급 평균표 만들기
* 그래프 만들기

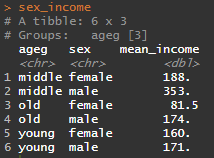
1. **연령대 및 성별 월급 평균표 만들기**

sex\_income <- welfare %>%

filter(!is.na(income)) %>% # 필터함수에서 월급을 기준으로 na 값이 아닌 것을 걸러낸다.

group\_by(ageg, sex) %>% # 그 결과를 그룹바이로 넘겨서 연령대별로 먼저 그룹을 지어주고(초년 중년 장년), 이후

summarise(mean\_income = mean(income)) # 그룹별로, 성별 별로 그룹을 지어준다.

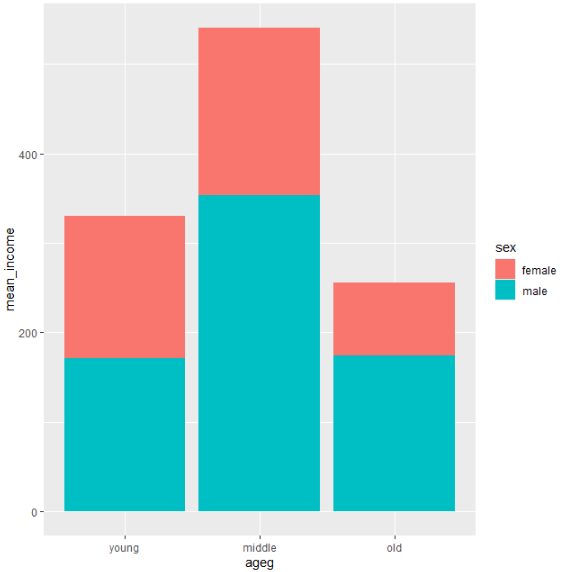


1. **그래프 만들기(스택 차트 생성)**

ggplot(data = sex\_income, aes(x = ageg, y=mean\_income, fill=sex)) +

geom\_col() +

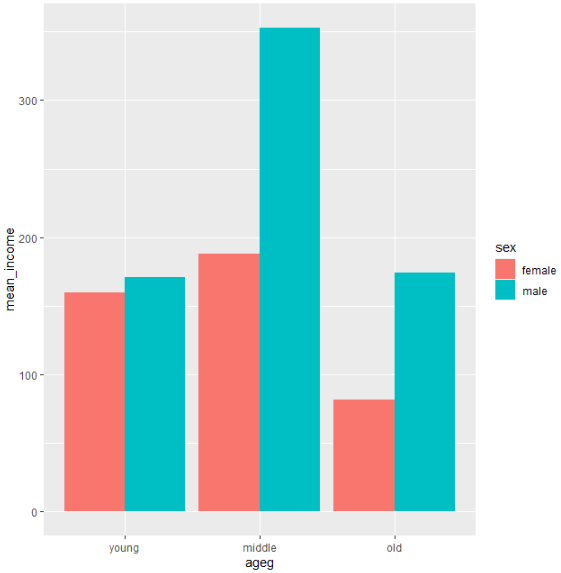
scale\_x\_discrete(limits = c("young", "middle", "old"))



ggplot(data = sex\_income, aes(x = ageg, y=mean\_income, fill=sex)) +

geom\_col(position="dodge") **+ # 동일코드지만 옵션만 추가(스택차트가 아닌 그룹차트 생성)**

scale\_x\_discrete(limits = c("young", "middle", "old"))



**나이 및 성별 월급 차이 분석하기**

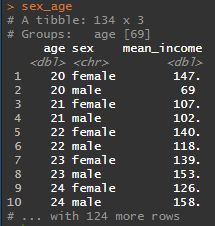
**# 성별 연령별 월급 평균표 만들기**

sex\_age <- welfare %>%

filter(!is.na(income)) %>%

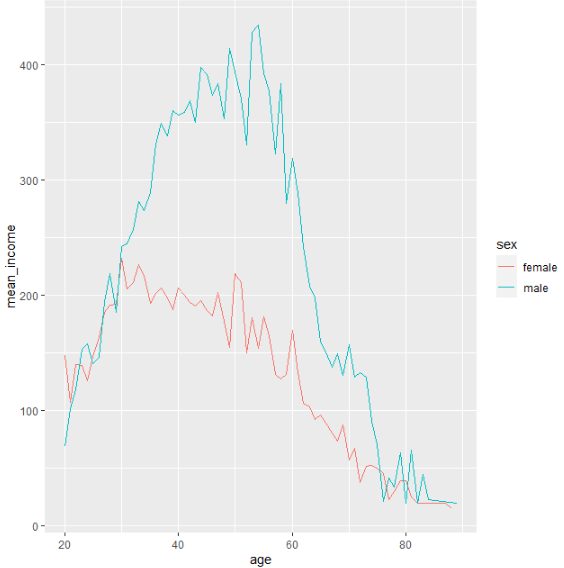
group\_by(age, sex) %>%

summarise(mean\_income = mean(income))



1. **그래프 만들기**

ggplot(data = sex\_age, aes(x=age, y=mean\_income, col=sex)) + geom\_line()



저장할 때 이미지 값 저장해야한다.