Introduction to R

Nguyen Bich Ngoc

07 May, 2023

1 Introduction

1.1 R

- Developed from S - interpreted language
- mostly use in statistics and data analysis
- free & open-source
- extendable with packages and new functions
- https://cran.r-project.org/

1.2 RStudio

- Interface
- only work with R installed
- free & open-source
- start and save scripts
- Ctrl+Enter to run
- http://rstudio.com/

1.3 Let's start with some simple calculations

```
2+3
```

```
## [1] 5
```

```
4^2
```

```
## [1] 16
```

```
sqrt(25)
```

```
## [1] 5
```

```
8^(1/3)
```

```
## [1] 2
```

```
pi
```

```
## [1] 3.141593
```

```
exp(1)
```

```
## [1] 2.718282
```

```
log(exp(2))
```

```
## [1] 2
```

```
log10(100)
```

```
## [1] 2
```

1.4 Packages & Functions

- packages add additional functions
- packages need to be installed once
- sometime will need to be installed again after update
- load before every run

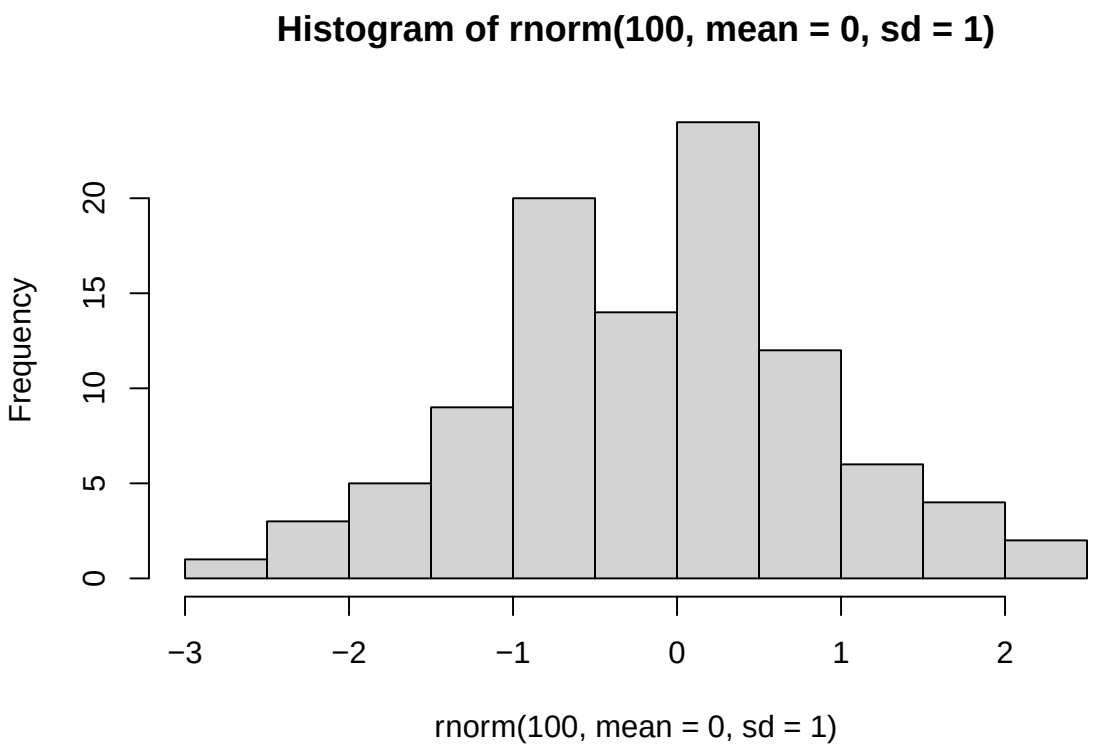
```
# install.packages("ggplot2")
library(ggplot2)
```

- functions receive arguments => return results

```
rnorm(100, mean = 0, sd = 1)
```

```
##   [1]  0.054209096  0.531046567 -0.157954288 -1.981092275  0.229515472
##   [6] -0.680109027 -1.002627436  0.257009673 -1.323219681  0.005142566
##  [11] -0.259881707  0.311233721  1.182304627  0.453578724 -1.745887192
##  [16]  0.056632291 -1.259901578 -1.020271979  0.366893517 -0.949322769
##  [21]  0.760504685  0.182387332  2.480698953 -0.025514654 -0.570868631
##  [26] -1.121065020  0.690357669  0.426095251  0.382635665 -0.322909990
##  [31]  1.181651647 -0.447733392 -0.345711265  1.242518993  1.224990977
##  [36] -0.185786563  0.481606517 -1.710824277  0.122615488  0.092660670
##  [41] -0.811281679  0.418018416  0.786843330 -0.660249965  1.132837650
##  [46]  0.199798477 -0.057521099 -0.107900931  0.891157612 -0.057098067
##  [51]  2.386015617 -1.622982599 -0.048898892 -0.484497162 -0.894929703
##  [56]  0.793543529 -1.043269890  0.524224518  1.132887286 -0.543336194
##  [61]  2.238841638  2.017643833 -2.212186971  1.223293219  0.211390838
##  [66] -0.628673177  0.467721296 -0.020148750  0.696674048 -2.271607772
##  [71] -1.517209038  0.310480463 -1.302990316 -1.389708490  0.046398015
##  [76]  0.142183019  0.037675083 -1.223124552 -0.068040312 -0.672492133
##  [81]  0.262546470 -0.366969427 -1.124507284  0.281024690  0.084014828
##  [86]  1.231403595 -1.181001229 -0.879457758 -0.113846853 -0.022758251
##  [91] -0.385425298 -0.255306073 -0.905051011 -1.209053832 -1.252645559
##  [96]  1.283169766 -0.234667609 -1.538795364 -0.551323286  1.155290045
```

```
hist(rnorm(100, mean = 0, sd = 1))
```



1.5 Operators

```
x <- 2+3
y = 6-5

x == 3
```

```
## [1] FALSE
```

```
y != 2
```

```
## [1] TRUE
```

```
x < 0
```

```
## [1] FALSE
```

```
y < 4
```

```
## [1] TRUE
```

```
x >=5
```

```
## [1] TRUE
```

```
y <= 10
```

```
## [1] TRUE
```

```
is.na(x)
```

```
## [1] FALSE
```

```
x < 0 & y < 4
```

```
## [1] FALSE
```

```
x < 0 | y < 4
```

```
## [1] TRUE
```

1.6 Getting help

```
?sqrt
```

```
## starting httpd help server ... done
```

```
# x + 3
x <- 2
x + 3
```

```
## [1] 5
```

2 Data

2.1 Data types

- numeric: e.g. `1`, `45.3`
- integer: e.g. `2L`, `53L`
- logical: e.g. `TRUE`, `T`, `FALSE`, `F`
- character: e.g. `"orange"`, `"female"`, `"Totally agree"`

2.2 Data structure

- Vectors
- Matrix
- Data frame
- List

2.2.1 Vectors

- simplest type

```r
x <- c(1, 8, 23, -7, 13)
x
```

```
## [1]  1  8 23 -7 13
```

```r
y <- c("a", "b", "c", "d", "e")
y
```

```
## [1] "a" "b" "c" "d" "e"
```

- same type of data

```r
a <- c(1, "a", 3, T)
a
```

```
## [1] "1"    "a"    "3"    "TRUE"
```

```r
str(a)
```

```
##  chr [1:4] "1" "a" "3" "TRUE"
```

```r
str(x)
```

```
##  num [1:5] 1 8 23 -7 13
```

```r
str(y)
```

```
##  chr [1:5] "a" "b" "c" "d" "e"
```

```r
b = c(1L, 8L, 23L, -7L, 13L)
str(b)
```

```
##  int [1:5] 1 8 23 -7 13
```

- creating vector

```r
x <- c(1, 8, 23, -7, 13)
x <- 1:20
y <- seq(from = 3, to = 8, by = 0.2)


rep("Female", 10)
```

```
##  [1] "Female" "Female" "Female" "Female" "Female" "Female" "Female" "Female"
##  [9] "Female" "Female"
```

- Vectorization

```r
x <- c(3, 7, 6, 3, 5, 2)
x + 1
```

```
## [1] 4 8 7 4 6 3
```

```
x*2
```

```
## [1]  6 14 12  6 10  4
```

```
sqrt(x)
```

```
## [1] 1.732051 2.645751 2.449490 1.732051 2.236068 1.414214
```

- logical vector

```
x
```

```
## [1] 3 7 6 3 5 2
```

```
z <- x > 4
z
```

```
## [1] FALSE  TRUE  TRUE FALSE  TRUE FALSE
```

```
y <- c("a", "b", "a", "d", "e")

t <- y == "A"
t
```

```
## [1] FALSE FALSE FALSE FALSE FALSE
```

```
t <- y == "a"
t
```

```
## [1]  TRUE FALSE  TRUE FALSE FALSE
```

- useful functions for vectors

```
x
```

```
## [1] 3 7 6 3 5 2
```

```
length(x)
```

```
## [1] 6
```

```
sum(x)
```

```
## [1] 26
```

```
max(x)
```

```
## [1] 7
```

```
min(x)
```

```
## [1] 2
```

```
sort(x)
```

```
## [1] 2 3 3 5 6 7
```

```
order(x)
```

```
## [1] 6 1 4 5 3 2
```

```
unique(x)
```

```
## [1] 3 7 6 5 2
```

```
mean(x)
```

```
## [1] 4.333333
```

```
sd(x)
```

```
## [1] 1.966384
```

```
y
```

```
## [1] "a" "b" "a" "d" "e"
```

```
length(y)
```

```
## [1] 5
```

```
unique(y)
```

```
## [1] "a" "b" "d" "e"
```

- subset vector

```
x
```

```
## [1] 3 7 6 3 5 2
```

```
x[1]
```

```
## [1] 3
```

```
x[3:5]
```

```
## [1] 6 3 5
```

```r
x[c(1,3:5)]
```

```
## [1] 3 6 3 5
```

```r
x[-2]
```

```
## [1] 3 6 3 5 2
```

```r
x[x > 4]
```

```
## [1] 7 6 5
```

```r
v <- 3
v[1]
```

```
## [1] 3
```

2.2.2 Matrices

- same type of data
- columns & rows

```r
x <- rbind(c(1:4), c(5:8))
x
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
```

```r
dim(x)
```

```
## [1] 2 4
```

```r
dimnames(x)
```

```
## NULL
```

```r
attributes(x)
```

```
## $dim
## [1] 2 4
```

```r
x <- cbind(c(1:4), c(5:8))
attributes(x)
```

```
## $dim
## [1] 4 2
```

```r
x <- matrix(1:8, nrow = 2, ncol = 4, byrow = T)
x
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
```

```r
x <- matrix(1:8, nrow = 2, ncol = 4, byrow = F)
x
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    5    7
## [2,]    2    4    6    8
```

2.2.3 Data frame

- extension of matrix
- columns of different data types

```r
id <- 1:6
gender <- rep(c("F", "M"), each = 3)
age <- rep(8, 6)

students <- cbind(id, gender, age)
students
```

```
##      id  gender age
## [1,] "1" "F"    "8"
## [2,] "2" "F"    "8"
## [3,] "3" "F"    "8"
## [4,] "4" "M"    "8"
## [5,] "5" "M"    "8"
## [6,] "6" "M"    "8"
```

```r
str(students)
```

```
##  chr [1:6, 1:3] "1" "2" "3" "4" "5" "6" "F" "F" "F" "M" "M" "M" "8" "8" "8" ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "id" "gender" "age"
```

```r
summary(students)
```

```
##       id                gender              age
##  Length:6           Length:6           Length:6
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
```

```r
students <- cbind.data.frame(id, gender, age)
students
```

```
##   id gender age
## 1  1      F   8
## 2  2      F   8
## 3  3      F   8
## 4  4      M   8
## 5  5      M   8
## 6  6      M   8
```

```
str(students)
```

```
## 'data.frame':    6 obs. of  3 variables:
##  $ id    : int  1 2 3 4 5 6
##  $ gender: chr  "F" "F" "F" "M" ...
##  $ age   : num  8 8 8 8 8 8
```

```
summary(students)
```

```
##        id          gender               age
##  Min.   :1.00   Length:6           Min.   :8
##  1st Qu.:2.25   Class :character   1st Qu.:8
##  Median :3.50   Mode  :character   Median :8
##  Mean   :3.50                      Mean   :8
##  3rd Qu.:4.75                      3rd Qu.:8
##  Max.   :6.00                      Max.   :8
```

```
id <- 1:6
math <- c(9, 6, 7, 8, 10, 5)
english <- c(8, 5, 9, 8, 9, 7)

results <- cbind.data.frame(id, math, english)

df <- merge(students, results)
df
```

```
##   id gender age math english
## 1  1      F   8    9       8
## 2  2      F   8    6       5
## 3  3      F   8    7       9
## 4  4      M   8    8       8
## 5  5      M   8   10       9
## 6  6      M   8    5       7
```

- subset data frame

```
sub1 <- df[1:3, 1:2]
sub1
```

```
##   id gender
## 1  1      F
## 2  2      F
## 3  3      F
```

```
sub2 <- df[, c(1, 4:5)]
sub2
```

```
##   id math english
## 1  1    9       8
## 2  2    6       5
## 3  3    7       9
## 4  4    8       8
## 5  5   10       9
## 6  6    5       7
```

```r
sub3 <- df$gender
sub3
```

```
## [1] "F" "F" "F" "M" "M" "M"
```

```r
sub4 <- subset(df, math > 7, select = c(id, english))
sub4
```

```
##   id english
## 1  1       8
## 4  4       8
## 5  5       9
```

```r
sub5 <- subset(df, english <= 7, select = -age)
sub5
```

```
##   id gender math english
## 2  2      F    6       5
## 6  6      M    5       7
```

- reorder the data frame

```r
df1 <- df[order(df$math),]
df1
```

```
##   id gender age math english
## 6  6      M   8    5       7
## 2  2      F   8    6       5
## 3  3      F   8    7       9
## 4  4      M   8    8       8
## 1  1      F   8    9       8
## 5  5      M   8   10       9
```

2.2.4 List

- contain different type of data
- can contain data frame

```r
cls1 <- data.frame(id = 1:5, names = c("Lan", "Hung", "Tuan", "Mai", "Long"))
cls2 <- data.frame(id = 6:10, names = c("Thanh", "Son", "Nghia", "Hanh", "Thuy"))

ls <- list(cls1 = cls1, cls2 = cls2)
ls
```

```
## $cls1
##   id names
## 1  1   Lan
## 2  2  Hung
## 3  3  Tuan
## 4  4   Mai
## 5  5  Long
##
## $cls2
##   id names
```

```
## 1  6 Thanh
## 2  7   Son
## 3  8 Nghia
## 4  9  Hanh
## 5 10  Thuy
```

- subset list

```
ls[1]
```

```
## $cls1
##   id names
## 1  1   Lan
## 2  2  Hung
## 3  3  Tuan
## 4  4   Mai
## 5  5  Long
```

```
ls[[1]]
```

```
##   id names
## 1  1   Lan
## 2  2  Hung
## 3  3  Tuan
## 4  4   Mai
## 5  5  Long
```

```
ls$cls1
```

```
##   id names
## 1  1   Lan
## 2  2  Hung
## 3  3  Tuan
## 4  4   Mai
## 5  5  Long
```

2.3 Import data

- Direct typing
- From clipboard

```
df <- read.delim("clipboard")
```

- From csv

```
df <- read.csv("C:/Users/nbngo/OneDrive/Work/[C] Quantitative research methods/quant_rm/Data - Quant
```

- From xlsx

```
# install.packages("readxl")
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.3
```

```
df <- read_excel("C:/Users/nbngo/OneDrive/Work/[C] Quantitative research methods/quant_rm/Data - Qua
```

- From Rdata

```
load("C:/Users/nbngo/OneDrive/Work/[C] Quantitative research methods/quant_rm/Data - Quant Methods/n
```

2.4 Export data

- as Rdata

```
save(df, file = "C:/Users/nbngo/OneDrive/Work/[C] Quantitative research methods/quant_rm/Data - Quan
```

- to clipboard

```
write.table(ntl_joined_avg, "clipboard", sep = "\t", row.names = F)
```

- to csv

```
write.csv(ntl_joined_avg, file = "C:/Users/nbngo/OneDrive/Work/[C] Quantitative research methods/qua
```

3 Exploring and plotting data

3.1 Data

- mtcars data in R

```
data <- mtcars

head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

- explaining variables
- summary() gives overall information of the data

```
summary(data)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
```

```
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

3.2 Numerical/continuous (ratio/interval) variables

```
summary(data$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.43   19.20   20.09   22.80   33.90
```

```
mean(data$mpg)
```

```
## [1] 20.09062
```

```
median(data$mpg)
```

```
## [1] 19.2
```

```
sd(data$mpg)
```

```
## [1] 6.026948
```

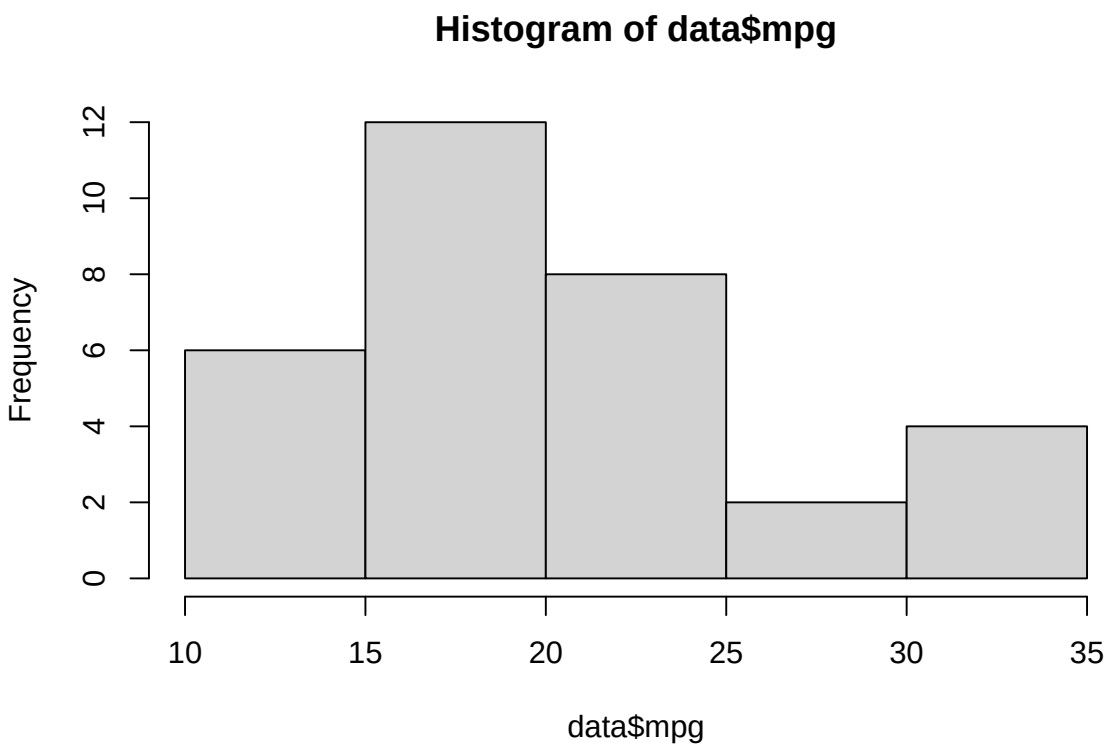
```
var(data$mpg)
```

```
## [1] 36.3241
```

```
quantile(data$mpg, seq(0, 1, 0.2))
```

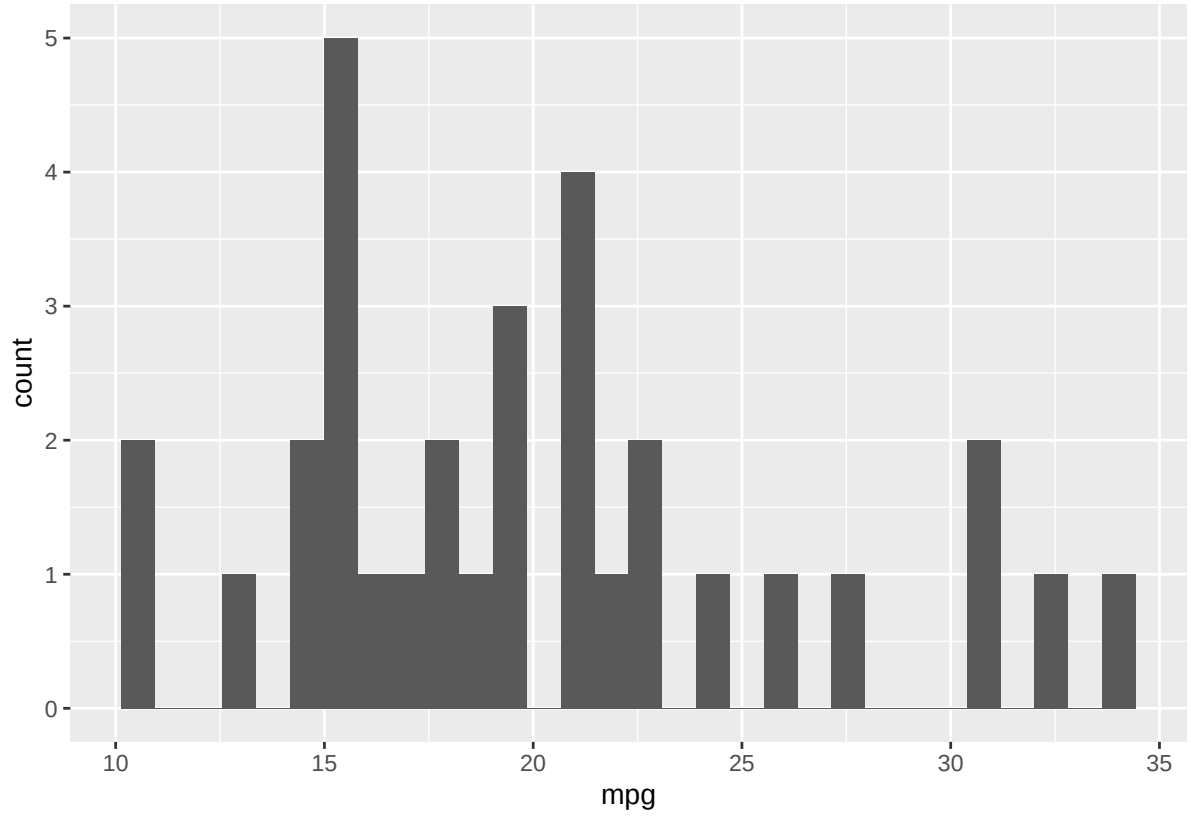
```
##    0%   20%   40%   60%   80%  100%
## 10.40 15.20 17.92 21.00 24.08 33.90
```

```
hist(data$mpg)
```

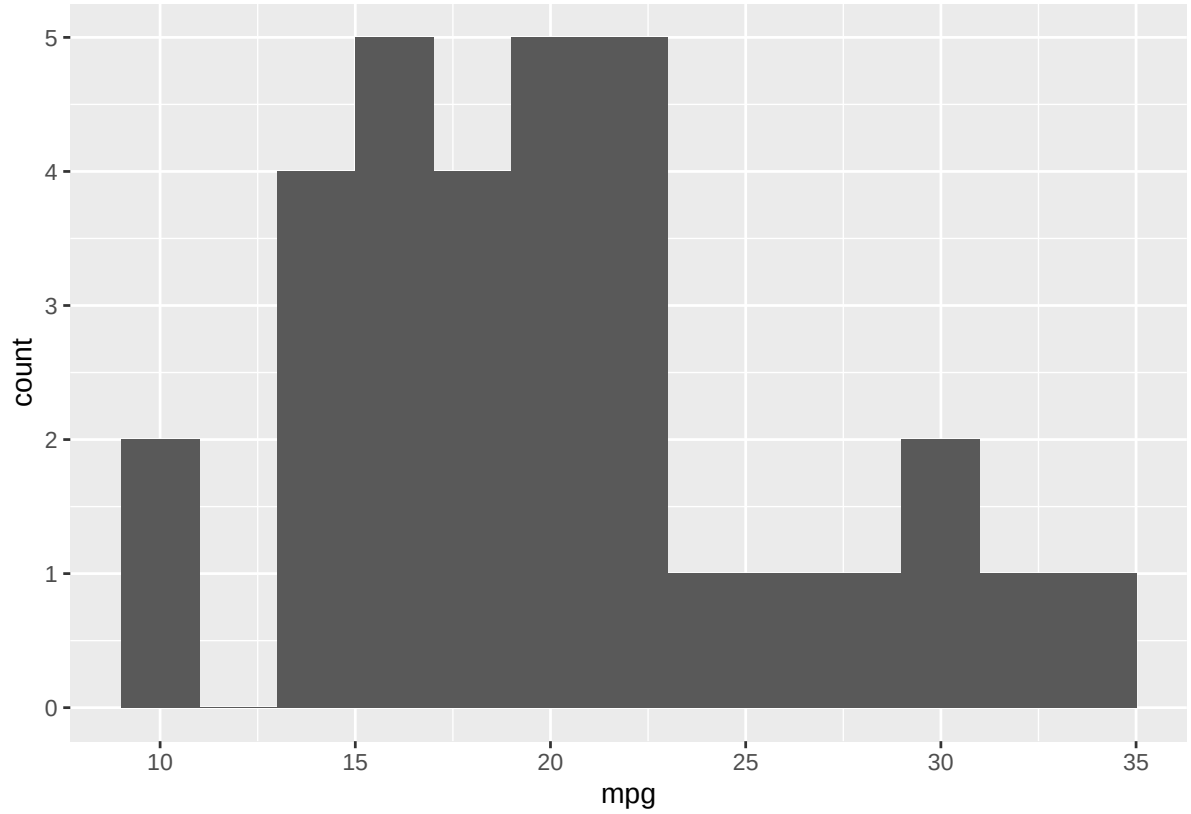



```
ggplot(data, aes(x = mpg)) +
  geom_histogram()
```

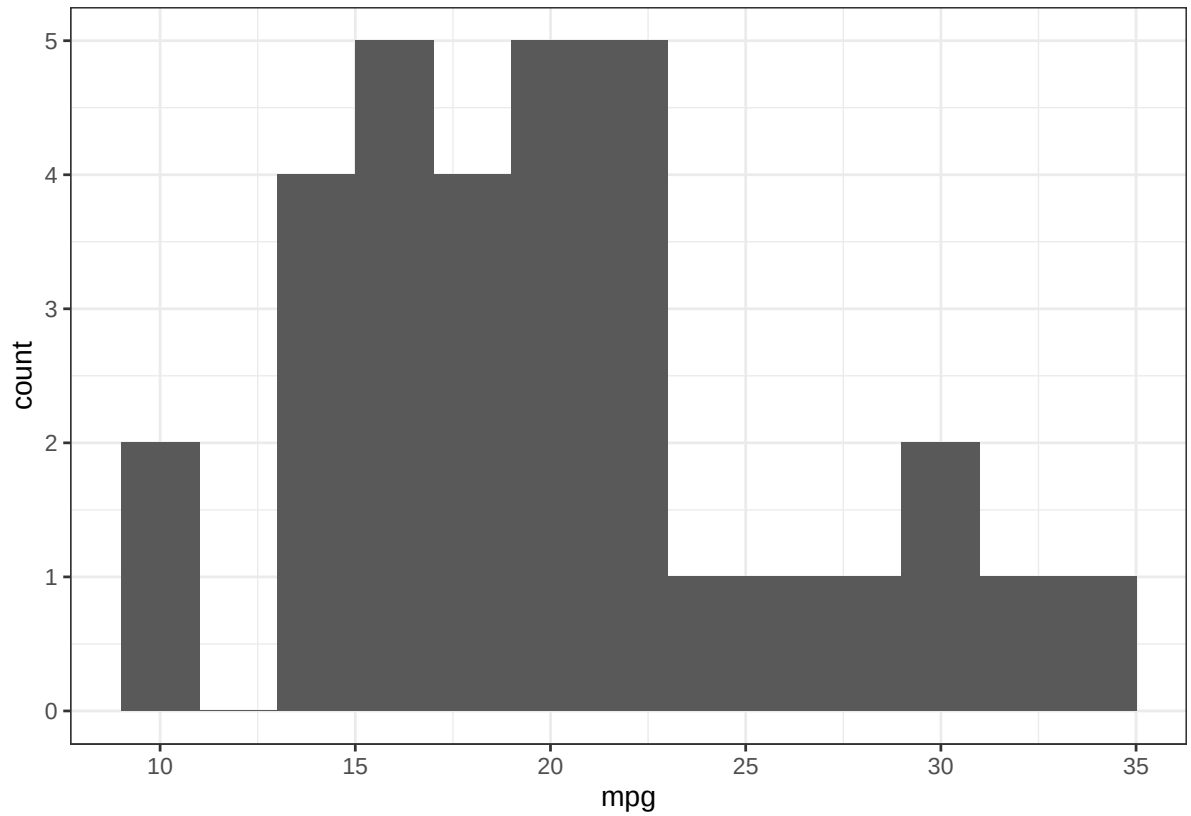
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



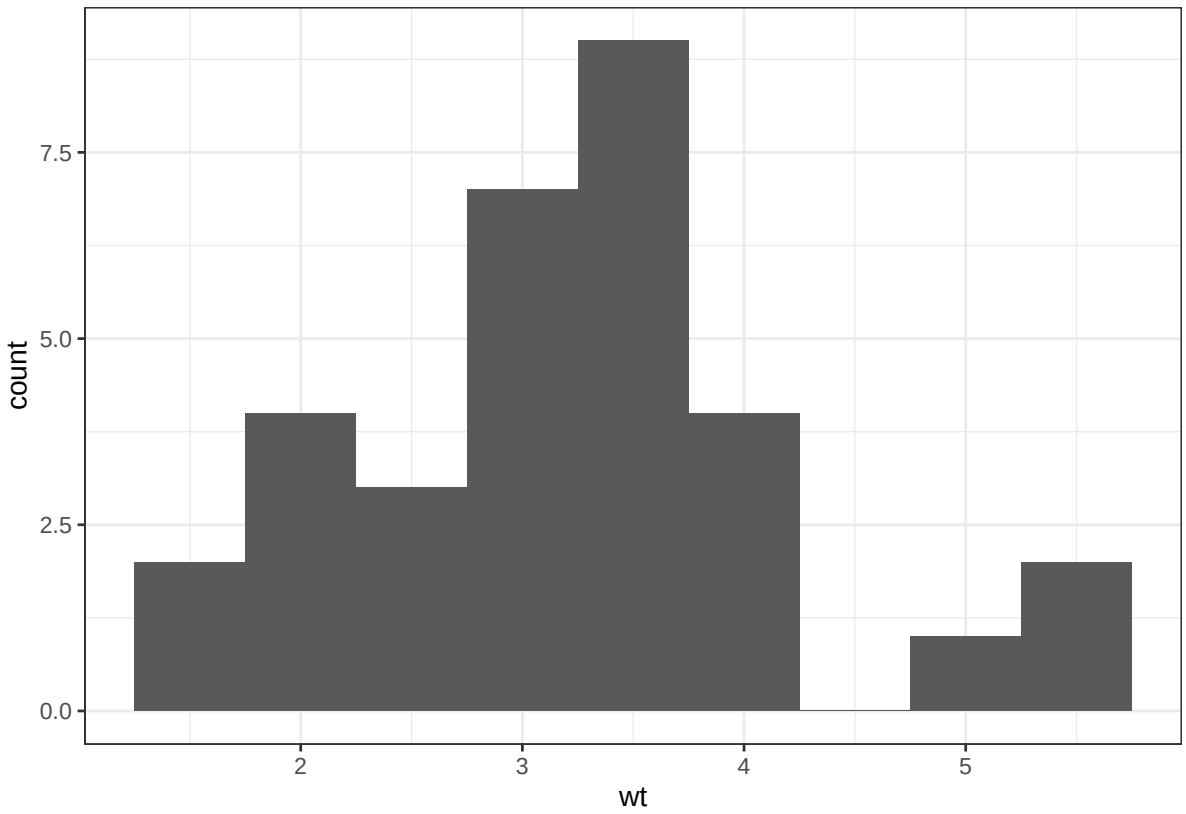
```
ggplot(data, aes(x = mpg)) +
  geom_histogram(binwidth = 2)
```




```
ggplot(data, aes(x = mpg)) +
  geom_histogram(binwidth = 2) +
  theme_bw()
```




```
??theme_bw

ggplot(data, aes(x = wt)) +
  geom_histogram(binwidth = 0.5) +
  theme_bw()
```



Character/Factor (Nominal/Order/Categorical)

```
table(data$cyl)
```

```
##
##  4  6  8
## 11  7 14
```

```
prop.table(table(data$cyl))
```

```
##
##       4       6       8
## 0.34375 0.21875 0.43750
```

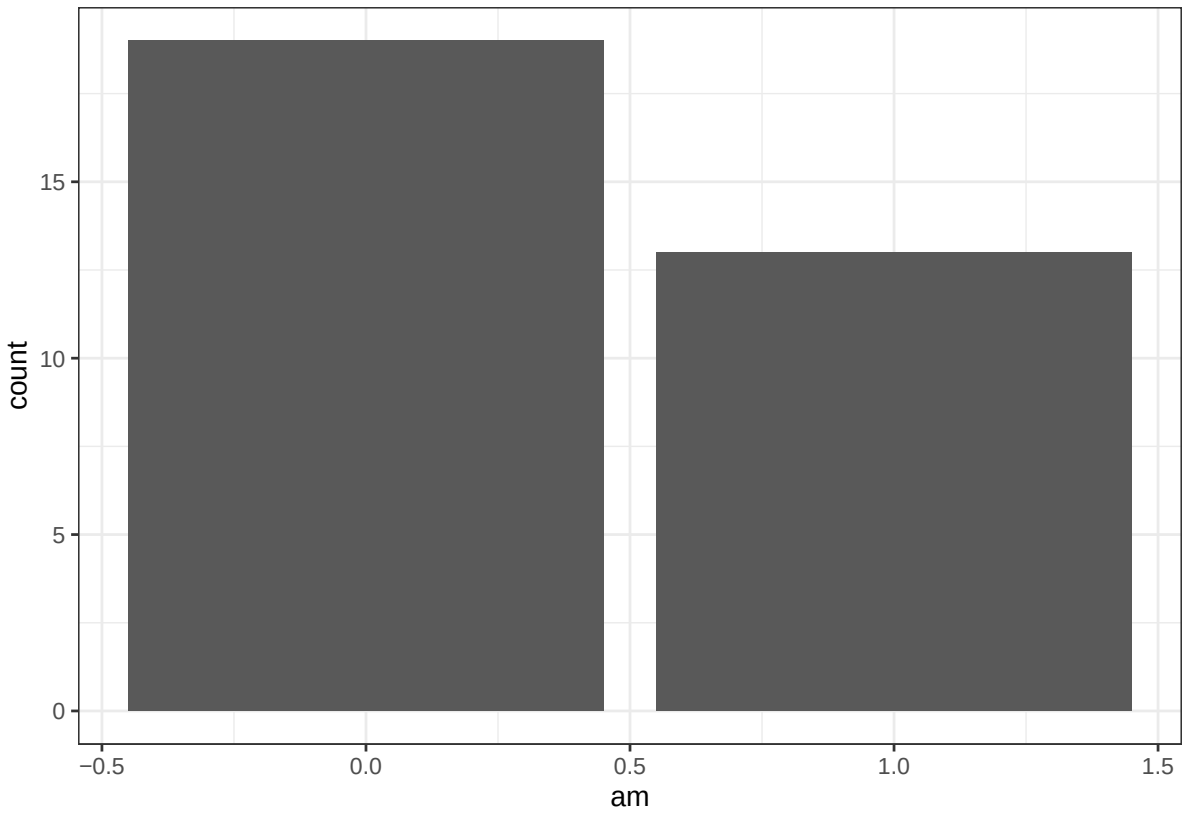
```
table(data$am)
```

```
##
##  0  1
## 19 13
```

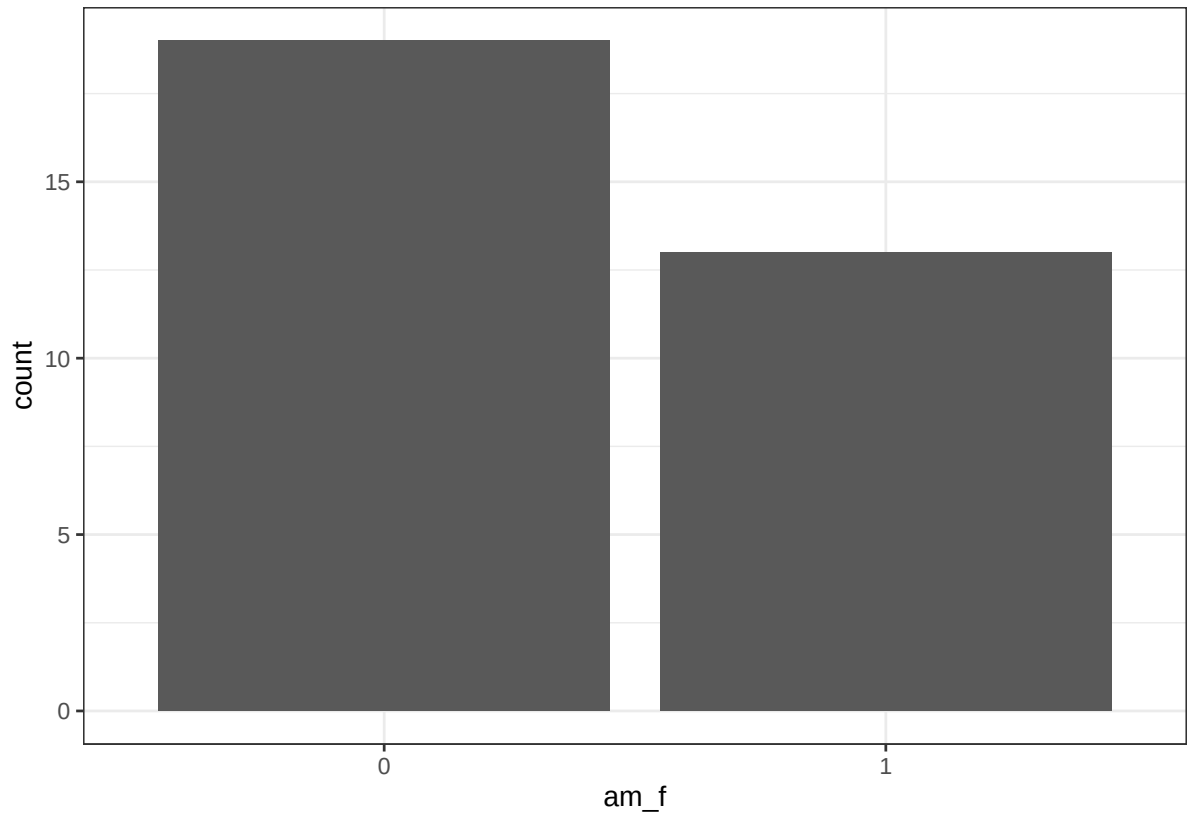
```
prop.table(table(data$am))
```

```
##
##       0       1
## 0.59375 0.40625
```

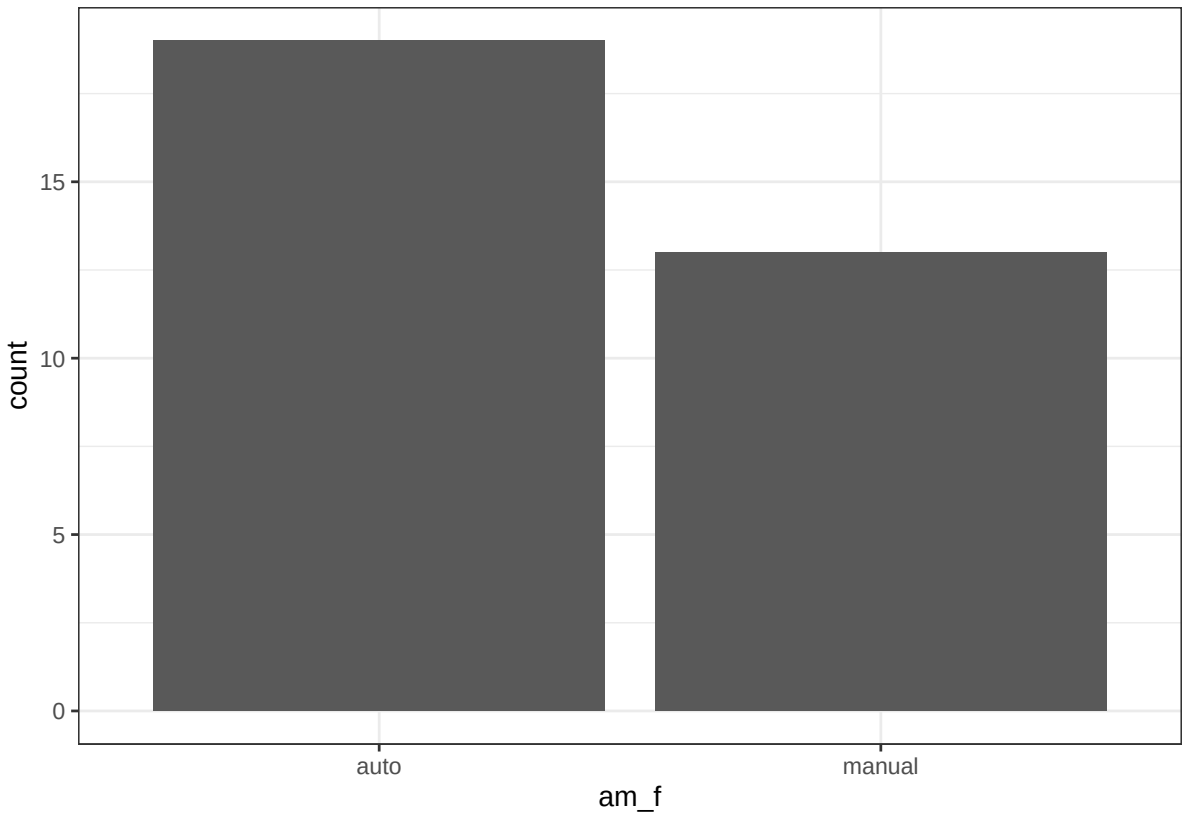
```
ggplot(data, aes(x = am)) +
  geom_bar() +
  theme_bw()
```




```
data$am_f <- factor(data$am)


ggplot(data, aes(x = am_f)) +
  geom_bar() +
  theme_bw()
```



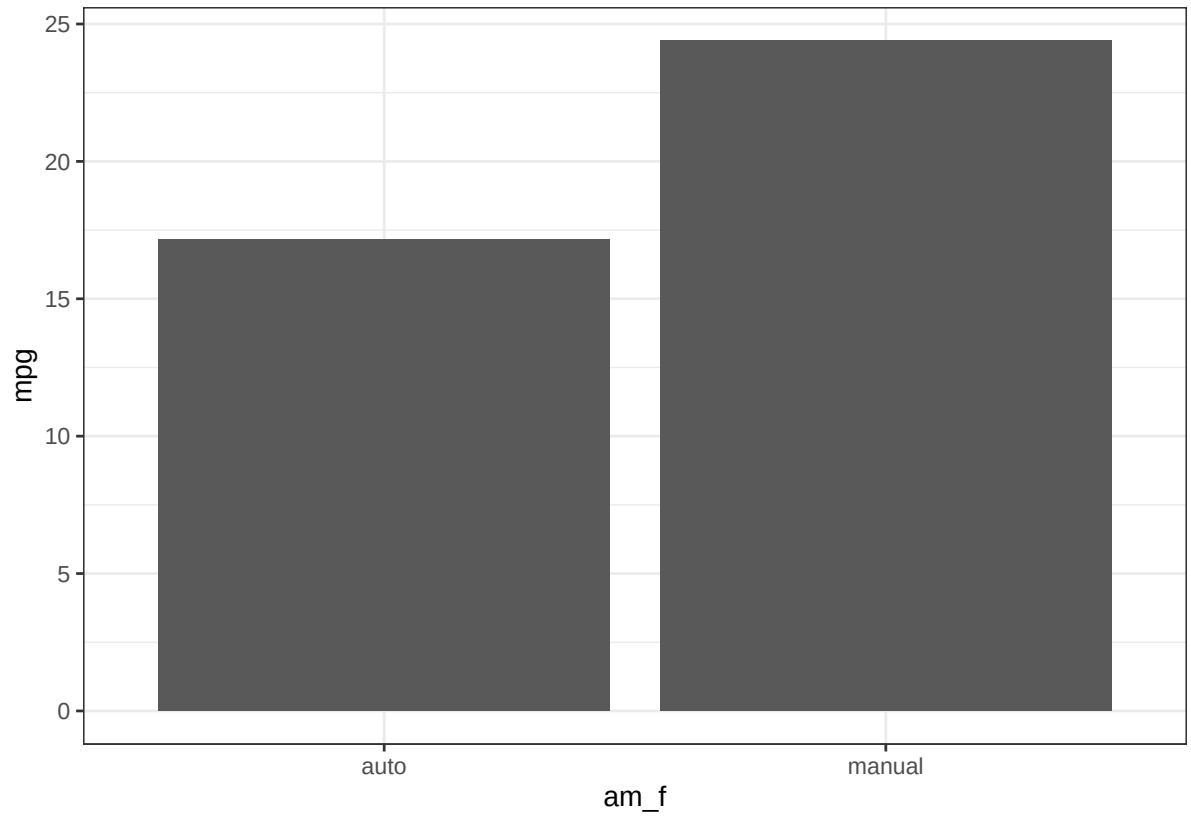

```
data$am_f <- factor(data$am, labels = c("auto", "manual"))


ggplot(data, aes(x = am_f)) +
  geom_bar() +
  theme_bw()
```



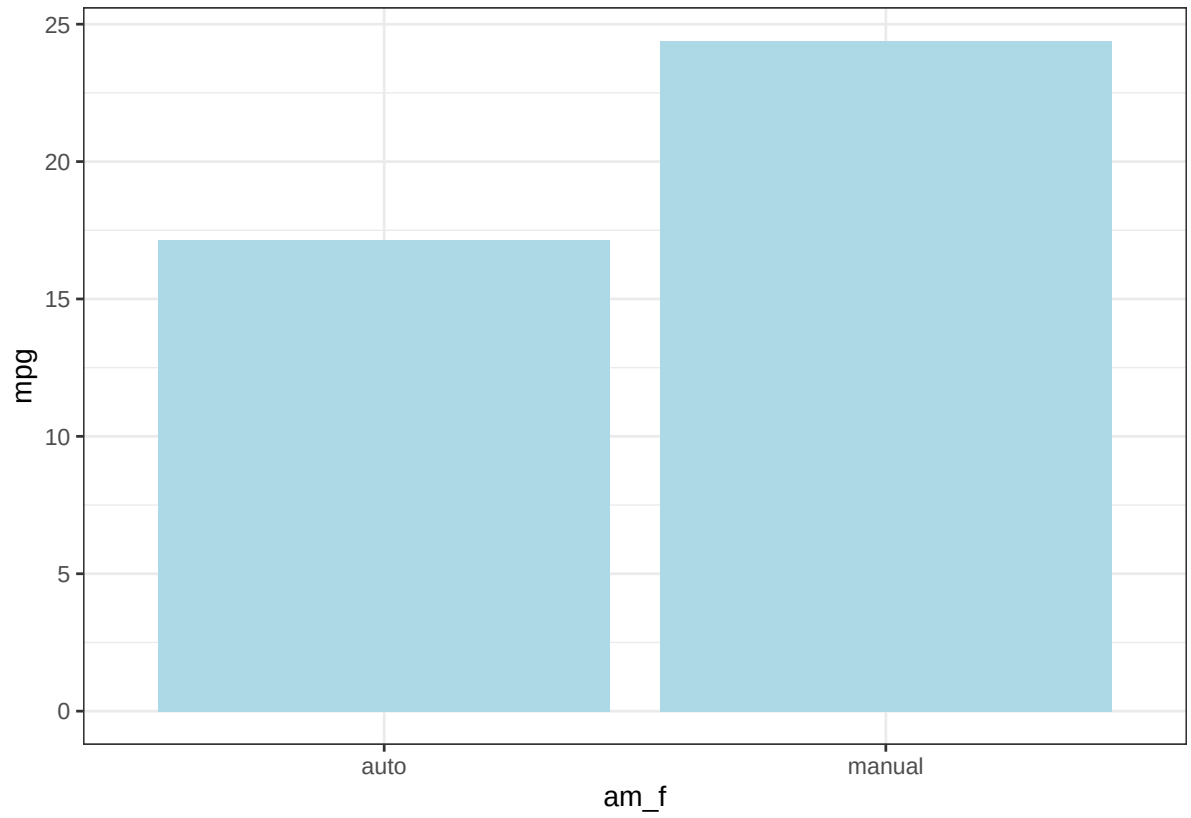
3.3 Bivariate

- continuous & discrete

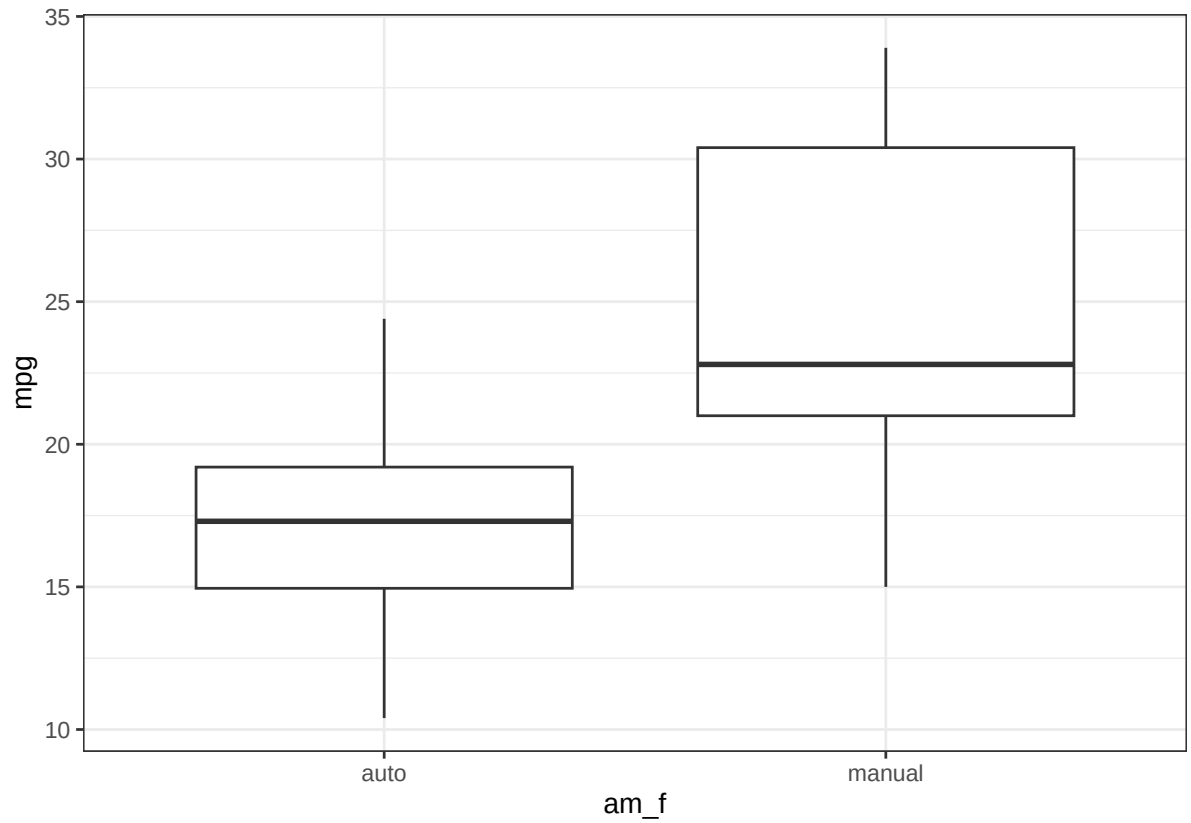
```
ggplot(data, aes(x = am_f, y = mpg)) +
  geom_bar(stat = "summary", fun = "mean") +
  theme_bw()
```




```
ggplot(data, aes(x = am_f, y = mpg)) +
  geom_bar(stat = "summary", fun = "mean", fill = "lightblue") +
  theme_bw()
```



```
ggplot(data, aes(x = am_f, y = mpg)) +
  geom_boxplot() +
  theme_bw()
```




```
# install.packages("dplyr")
library("dplyr")
```

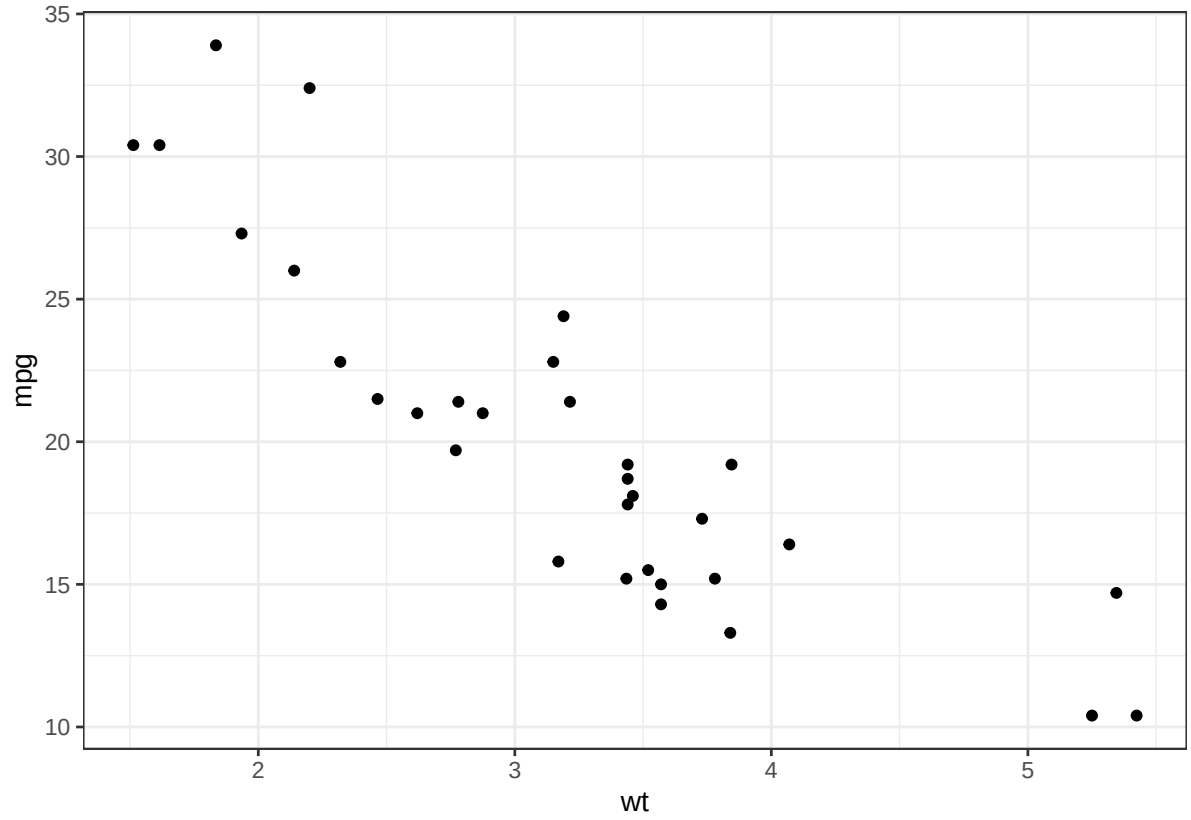
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data %>%
  group_by(am_f) %>%
  summarize(min = min(mpg),
            sd = sd(mpg),
            min = min(mpg),
            q1 = quantile(mpg, 0.25),
            median = median(mpg),
            q3 = quantile(mpg, 0.75),
            max = max(mpg))
```

```
## # A tibble: 2 x 7
##   am_f     min    sd    q1 median    q3   max
##   <fct>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1 auto    10.4  3.83  15.0   17.3  19.2  24.4
## 2 manual  15    6.17  21     22.8  30.4  33.9
```

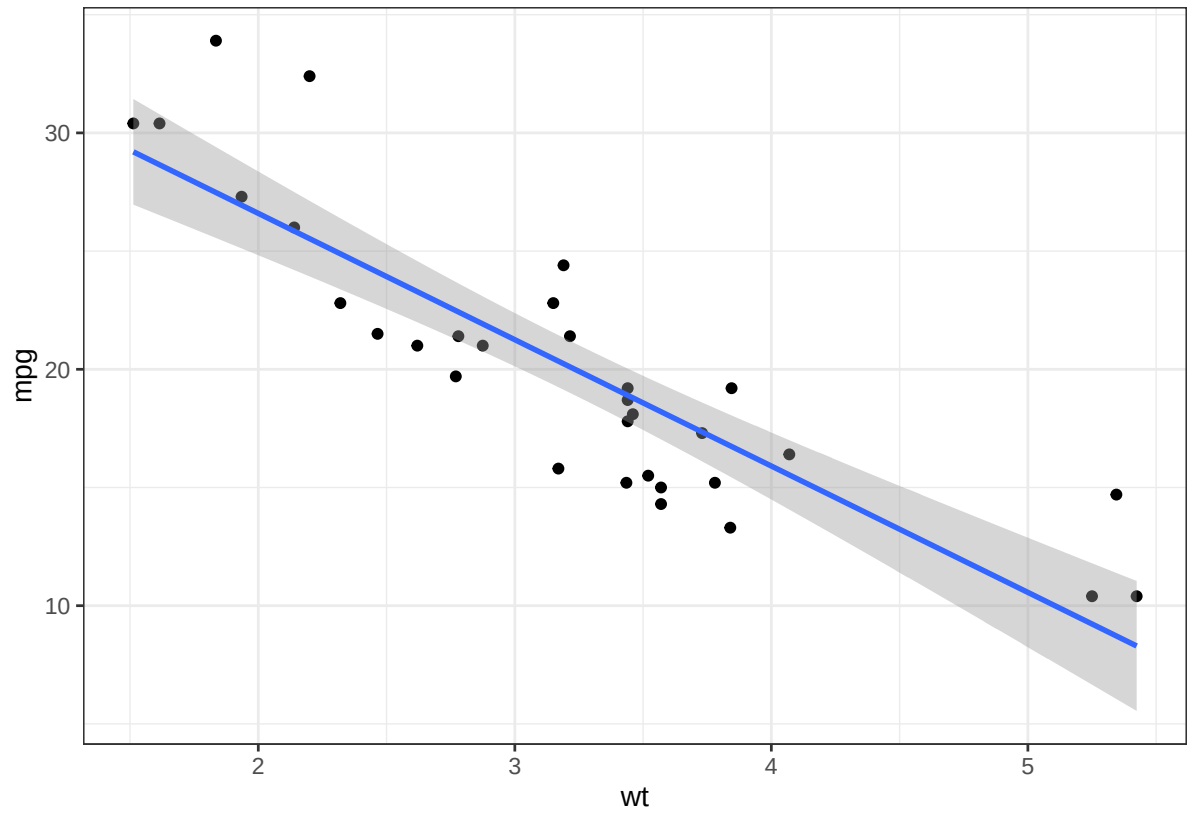
- Continuous & continuous

```
ggplot(data, aes(x = wt, y = mpg)) +
  geom_point() +
  theme_bw()
```



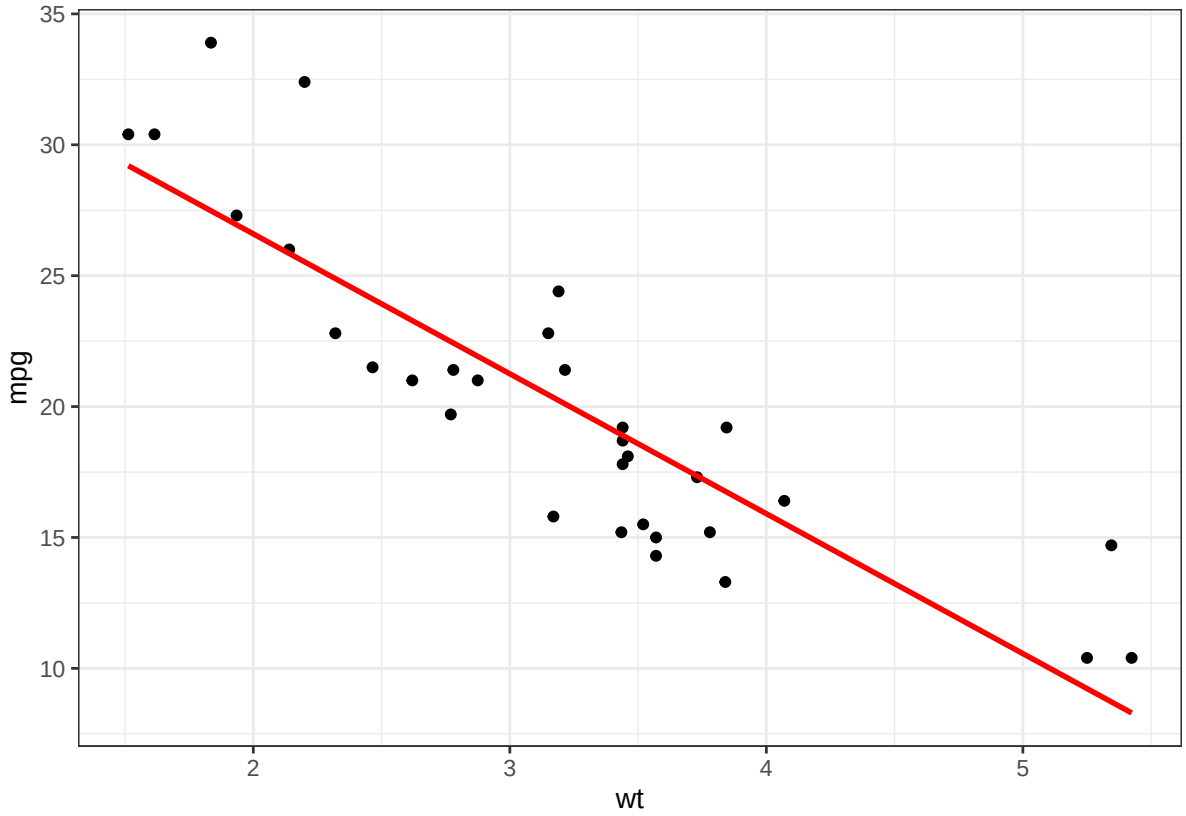

```
ggplot(data, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red", se = F) +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
cor(data$wt, data$mpg)
```

```
## [1] -0.8676594
```

```
cor.test(data$wt, data$mpg)
```

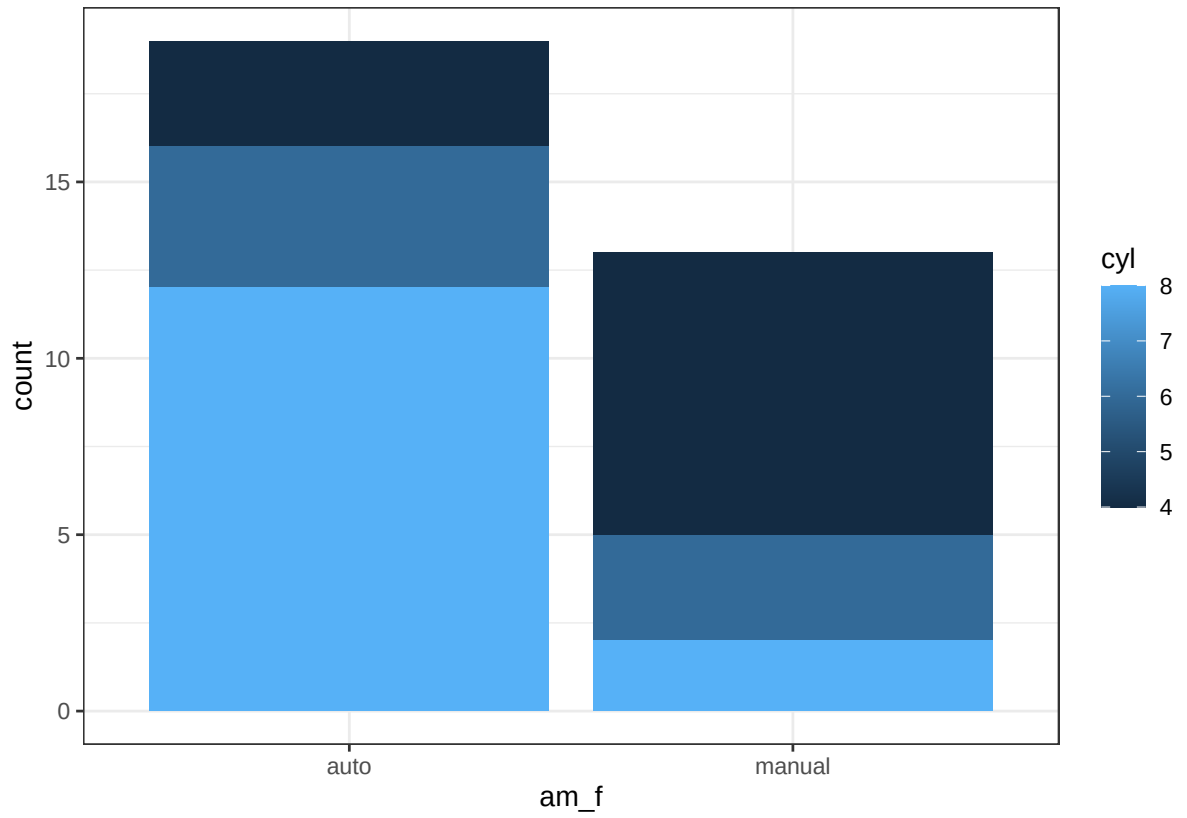
```
##
##  Pearson's product-moment correlation
##
## data:  data$wt and data$mpg
## t = -9.559, df = 30, p-value = 1.294e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9338264 -0.7440872
## sample estimates:
##        cor
## -0.8676594
```

- Discrete & discrete

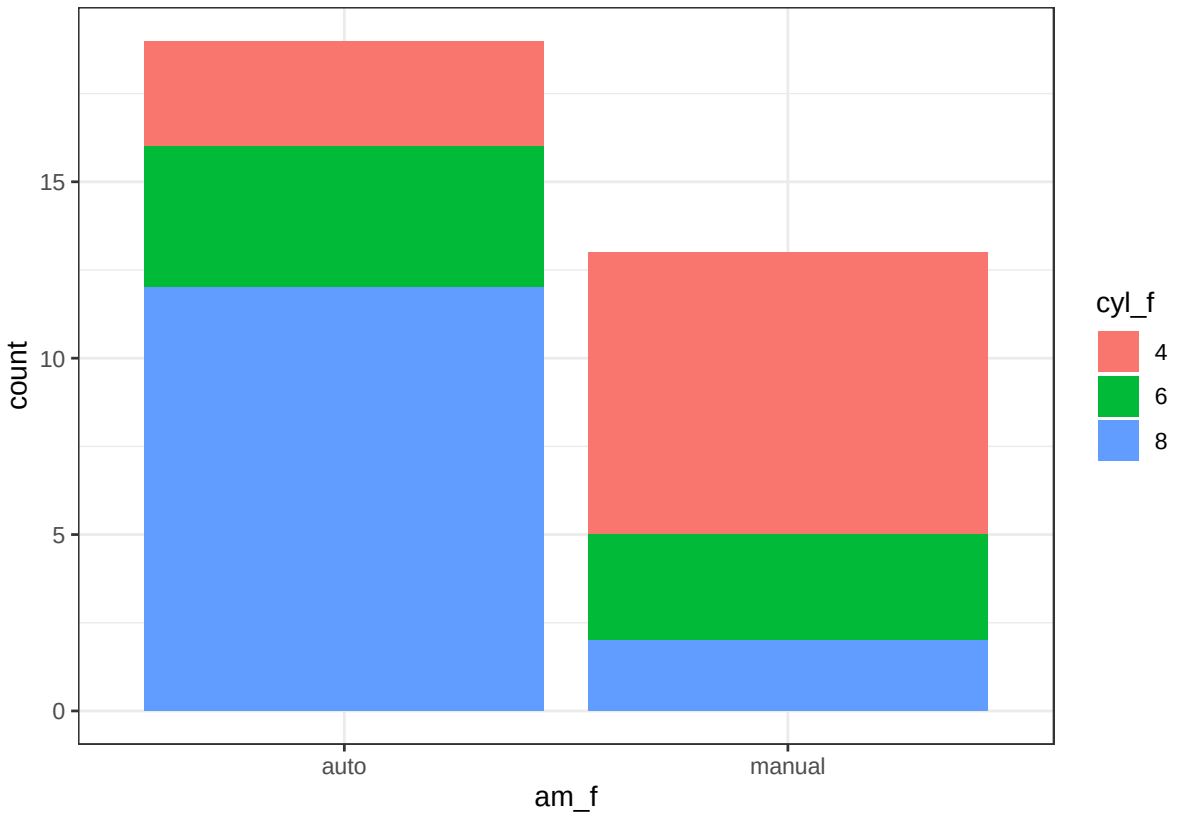
```
table(data$cyl, data$am_f)
```

```
##
##     auto manual
##   4    3      8
##   6    4      3
##   8   12      2
```

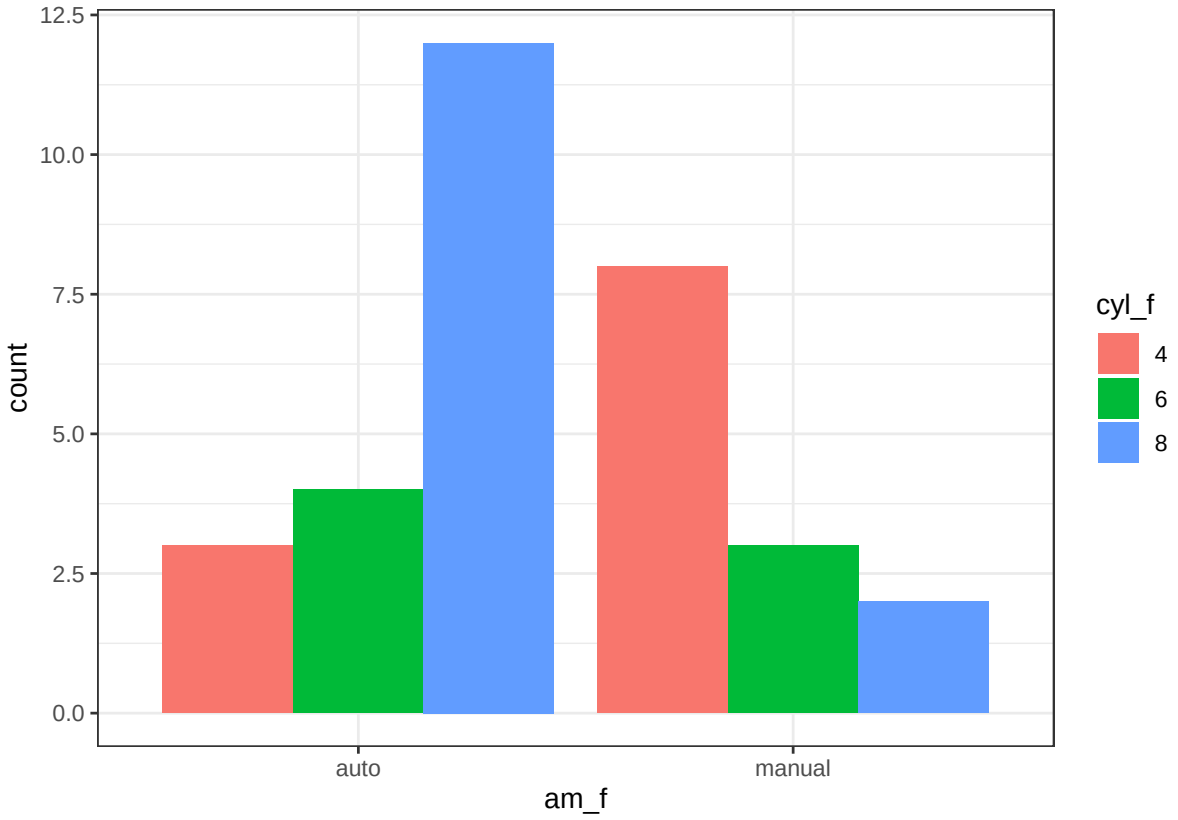
```
ggplot(data, aes(x = am_f, fill = cyl, group = cyl)) +
  geom_bar() +
  theme_bw()
```




```
data$cyl_f <- factor(data$cyl)

ggplot(data, aes(x = am_f, fill = cyl_f, group = cyl)) +
  geom_bar() +
  theme_bw()
```

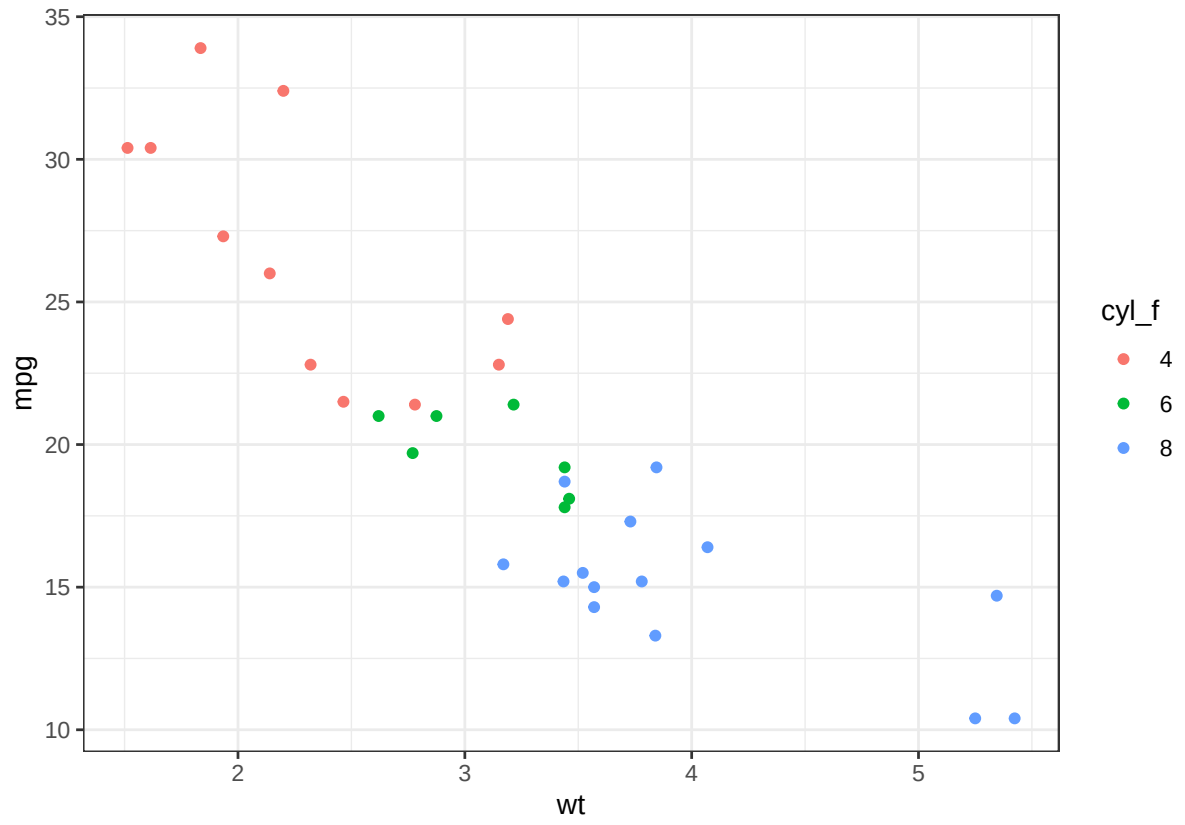



```
ggplot(data, aes(x = am_f, fill = cyl_f, group = cyl)) +
  geom_bar(position = "dodge") +
  theme_bw()
```



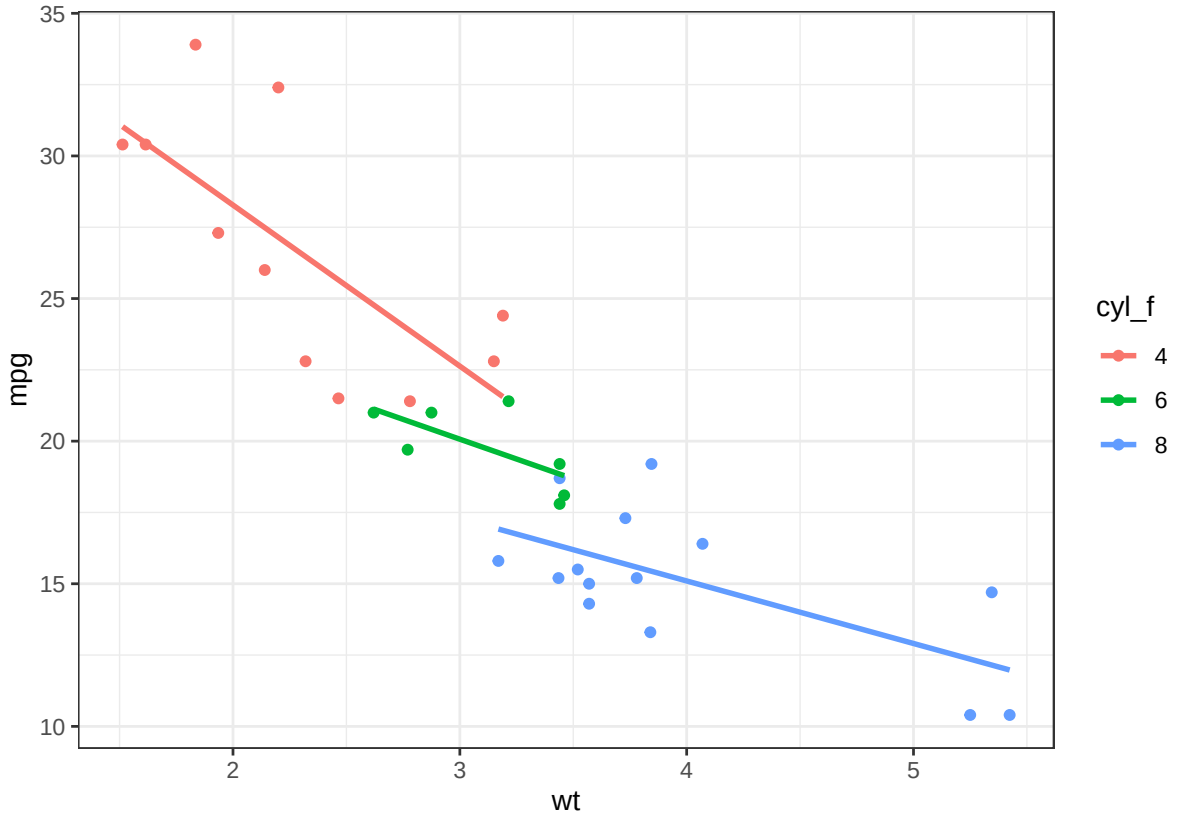
3.4 Add dimension

```
ggplot(data, aes(x = wt, y = mpg, col = cyl_f)) +
  geom_point()  +
  theme_bw()
```

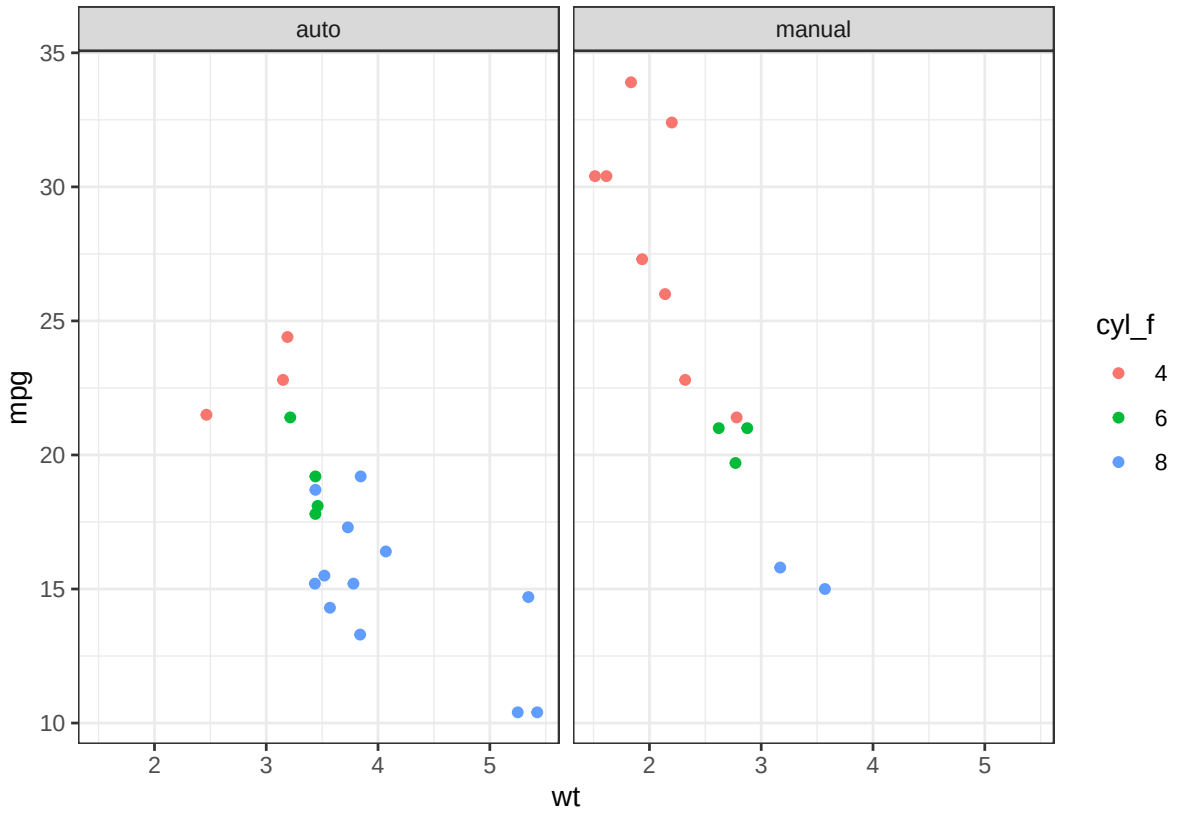



```
ggplot(data, aes(x = wt, y = mpg, col = cyl_f)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)  +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data, aes(x = wt, y = mpg, col = cyl_f)) +
  geom_point() +
  facet_grid(.~am_f) +
  theme_bw()
```



4 Linear regression

4.1 Compare means

```
fit1 <- lm(mpg ~ am_f, data)
summary(fit1)
```

```
##
## Call:
## lm(formula = mpg ~ am_f, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3923 -3.0923 -0.2974  3.2439  9.5077
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   17.147      1.125  15.247 1.13e-15 ***
## am_fmanual     7.245      1.764   4.106 0.000285 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.902 on 30 degrees of freedom
## Multiple R-squared:  0.3598, Adjusted R-squared:  0.3385
## F-statistic: 16.86 on 1 and 30 DF,  p-value: 0.000285
```

```
fit2 <- lm(mpg ~ cyl_f, data)
summary(fit2)
```

```
##
## Call:
## lm(formula = mpg ~ cyl_f, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2636 -1.8357  0.0286  1.3893  7.2364
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  26.6636     0.9718  27.437  < 2e-16 ***
## cyl_f6       -6.9208     1.5583  -4.441 0.000119 ***
## cyl_f8      -11.5636     1.2986  -8.905 8.57e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.223 on 29 degrees of freedom
## Multiple R-squared:  0.7325, Adjusted R-squared:  0.714
## F-statistic:  39.7 on 2 and 29 DF,  p-value: 4.979e-09
```

4.2 Simple linear regression

```
fit3 <- lm(mpg ~ wt, data)
summary(fit3)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

4.3 Multiple linear regression

```
fit4 <- lm(mpg ~ wt + cyl_f + am_f, data)
summary(fit4)
```

```
##
## Call:
## lm(formula = mpg ~ wt + cyl_f + am_f, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4898 -1.3116 -0.5039  1.4162  5.7758
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.7536     2.8135  11.997  2.5e-12 ***
## wt           -3.1496     0.9080  -3.469  0.00177 **
## cyl_f6       -4.2573     1.4112  -3.017  0.00551 **
## cyl_f8       -6.0791     1.6837  -3.611  0.00123 **
## am_fmanual    0.1501     1.3002   0.115  0.90895
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.603 on 27 degrees of freedom
## Multiple R-squared:  0.8375, Adjusted R-squared:  0.8134
## F-statistic: 34.79 on 4 and 27 DF,  p-value: 2.73e-10
```

```
fit5 <- lm(mpg ~ wt + cyl_f , data)
summary(fit5)
```

```
##
## Call:
## lm(formula = mpg ~ wt + cyl_f, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -4.5890 -1.2357 -0.5159  1.3845  5.7915
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.9908     1.8878  18.006  < 2e-16 ***
## wt           -3.2056     0.7539  -4.252 0.000213 ***
## cyl_f6       -4.2556     1.3861  -3.070 0.004718 **
## cyl_f8       -6.0709     1.6523  -3.674 0.000999 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.557 on 28 degrees of freedom
## Multiple R-squared:  0.8374, Adjusted R-squared:   0.82
## F-statistic: 48.08 on 3 and 28 DF,  p-value: 3.594e-11
```

```
AIC(fit4)
```

```
## [1] 158.6065
```

```
AIC(fit5)
```

```
## [1] 156.6223
```

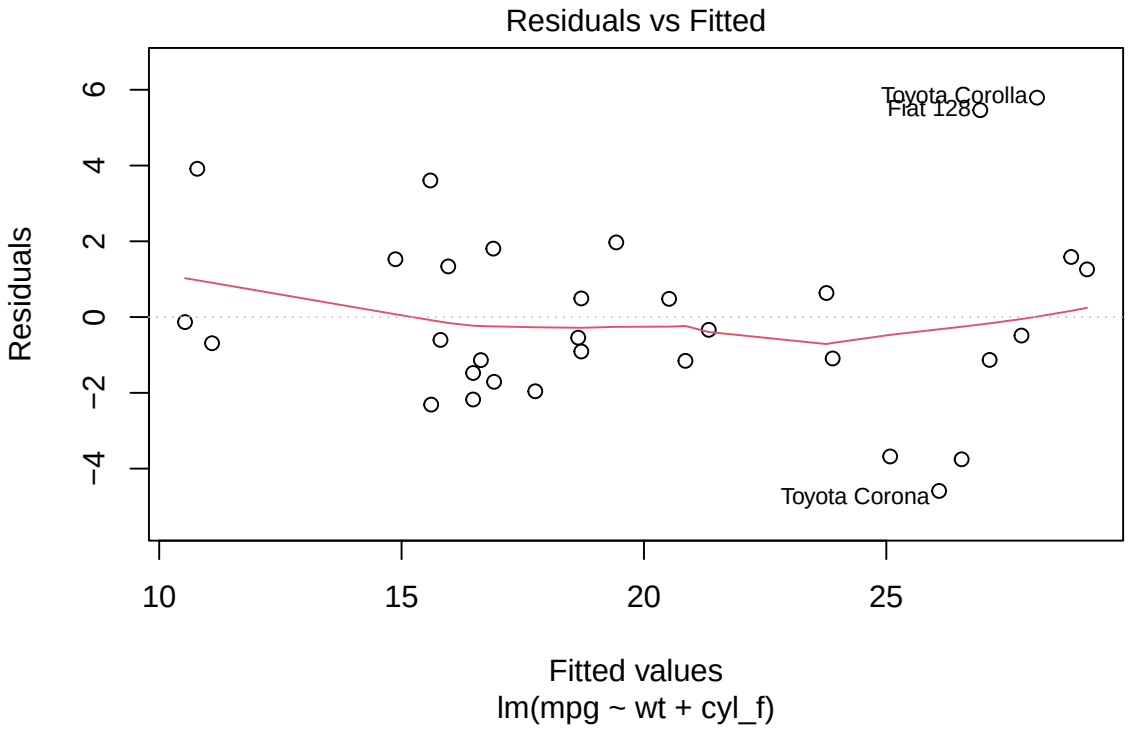
```
anova(fit4, fit5)
```

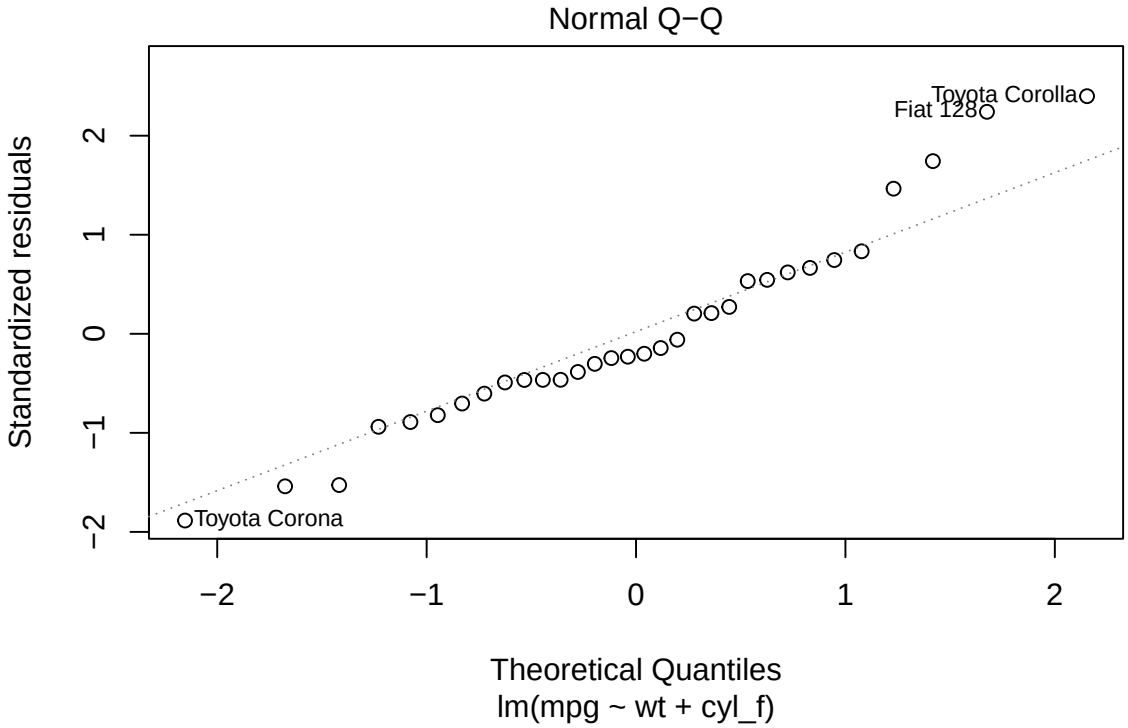
```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + cyl_f + am_f
## Model 2: mpg ~ wt + cyl_f
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     27 182.97
## 2     28 183.06 -1 -0.090314 0.0133 0.9089
```

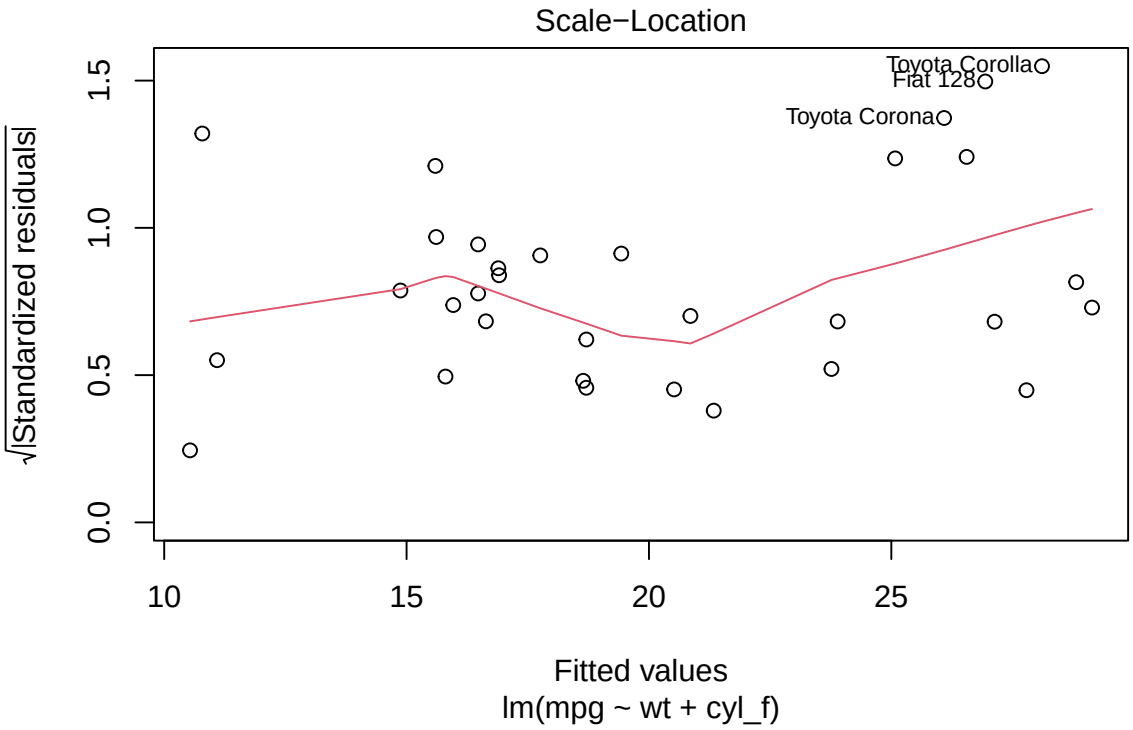
```
anova(fit4, fit3)
```

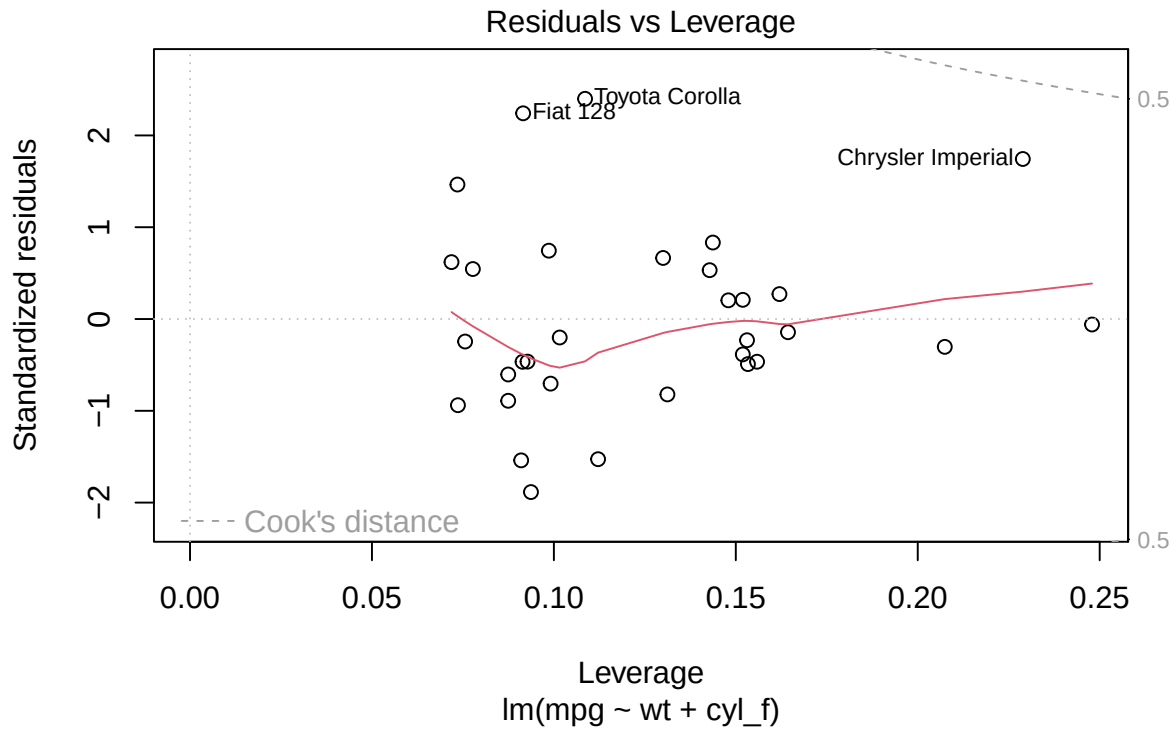
```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + cyl_f + am_f
## Model 2: mpg ~ wt
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     27 182.97
## 2     30 278.32 -3   -95.354 4.6903 0.009202 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(fit5)
```


Residuals vs Fitted
Residuals
Fitted values
lm(mpg ~ wt + cyl_f)
Toyota Corolla
Fiat 128
Toyota Corona


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(mpg ~ wt + cyl_f)
Toyota Corolla
Fiat 128
Toyota Corona


Scale−Location
√|Standardized residuals|
Toyota Corolla
Fiat 128
Toyota Corona
0.0
0.5
1.0
1.5
10
15
20
25
Fitted values
lm(mpg ~ wt + cyl_f)


Residuals vs Leverage
Standardized residuals
Toyota Corolla
Fiat 128
Chrysler Imperial
0.5
Cook's distance
0.5
−2
−1
0
1
2
0.00
0.05
0.10
0.15
0.20
0.25
Leverage
lm(mpg ~ wt + cyl_f)