Introduction to R

Nguyen Bich Ngoc

23 June, 2024

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]  0.62281568  0.58312750  0.45451709 -1.30398666 -0.20625839 -0.38723622
##   [7]  0.05363540 -0.23120599  0.10264344 -1.07033902 -0.64301441 -1.49024138
##  [13]  0.30070059  0.54294127  2.00620339 -0.07553674  0.24909706 -0.62481508
##  [19]  0.75506293  0.40261526 -1.24692878  0.37569940 -1.20052858 -0.54919173
##  [25] -1.63335694 -0.30875777  0.14017027  0.16543059 -0.94810365  1.93467224
##  [31]  0.70508869 -0.28612001 -1.09814973  0.78877065  0.70268473  0.44572027
##  [37] -0.62932193  1.14563841  1.17187555  0.22454399 -0.95150041 -0.06167269
##  [43] -0.25461019 -1.34199286 -0.03926203  0.22337837  0.84730062  0.84234388
##  [49]  1.32837159 -0.24006422 -1.98395329 -0.65537980  1.19814457  0.35016478
##  [55]  0.58980138  1.02746977 -1.81368906 -0.16279765  0.97070077 -0.13813345
##  [61]  0.40930744 -1.51940150 -0.12214393 -1.40978620 -0.28139357  0.12052454
##  [67]  1.27325488 -0.79555762 -0.67051741  0.19580915  1.88415442  0.75654959
##  [73] -1.26955162 -0.24816868 -1.00414862  0.40683283  0.62956576  0.15345164
##  [79] -0.45274413  0.52377891  0.64769480 -0.81047883 -2.29873541 -0.72140768
##  [85]  1.37845707 -0.31119335  0.45463541 -1.23089107  1.57157843 -0.25877931
##  [91] -0.36331943 -0.54891519  0.64038306 -0.61385345  2.39075781 -0.07292934
##  [97] -0.19158072 -1.52475813 -0.81254342  1.18029156
```

```
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

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

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

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

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

- same type of data

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

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

```
str(a)
```

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

```
str(x)
```

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

```
str(y)
```

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

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

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

- creating vector

```
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

```
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
```

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

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

```
x[-2]
```

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

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

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

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

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

2.2.2 Matrices

- same type of data
- columns & rows

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

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

```
dim(x)
```

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

```
dimnames(x)
```

```
## NULL
```

```
attributes(x)
```

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

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

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

```
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
```

```
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

```
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"
```

```
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"
```

```
summary(students)
```

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

```
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
```

```r
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
```

```r
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

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

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

```r
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

```
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

```
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

```
# open excel file
# 'C:\Users\nbngo\OneDrive\Work\[C]_ResearchMethods\quant_rm\Data\1
# Raw\Dataset_environmental_sustainability.xlsx'

# env <- read.delim('clipboard')
```

- From csv

```
env <- read.csv("Data/1 Raw/Dataset_environmental_sustainability.csv",
    sep = ",", header = T)
```

- From xlsx

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

env <- read_excel("Data/1 Raw/Dataset_environmental_sustainability.xlsx")
```

- From Rdata

```
load("Data/1 Raw/ntl_joined_avg.Rdata")
```

2.4 Export data

- as Rdata

```
save(env, file = "Data/2 Processed/environment_survey.Rdata")
```

- to clipboard

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

- to csv

```
write.csv(ntl_joined_avg, file = "Data/2 Processed/ice_cover_lake.csv",
    row.names = F)
```

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(mtcars)
```

```
##       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/quantitative variables

```
summary(mtcars$mpg)
```

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

```
mean(mtcars$mpg)
```

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

```
median(mtcars$mpg)
```

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

```
sd(mtcars$mpg)
```

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

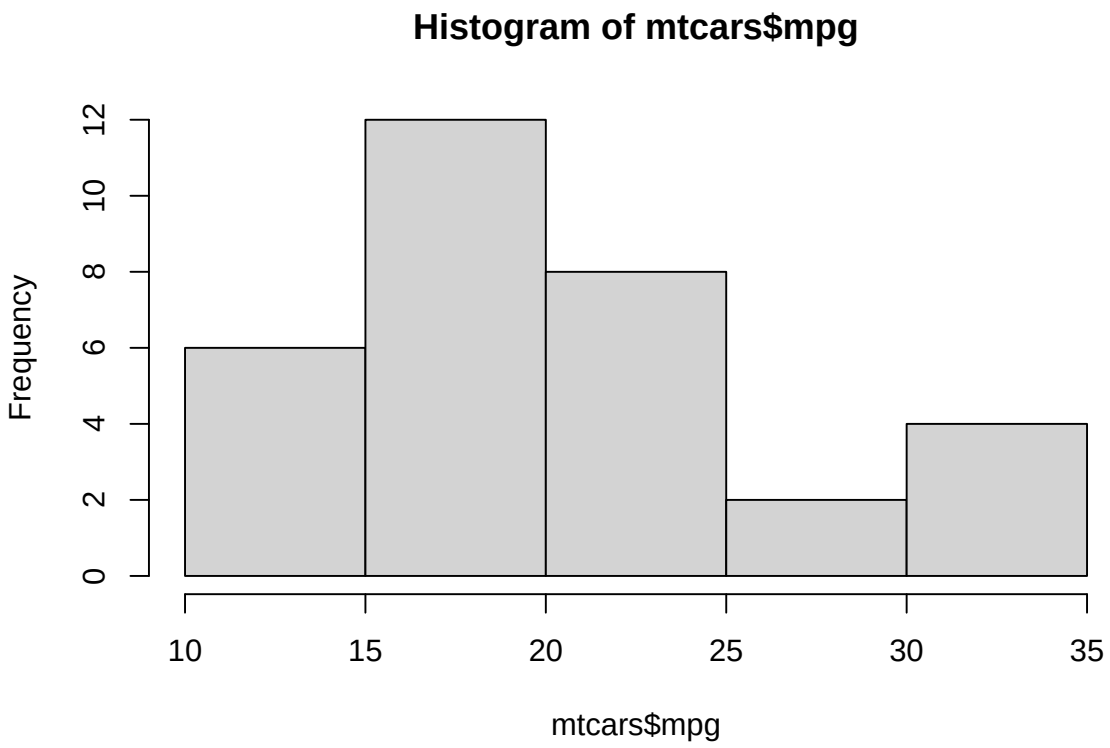
```
var(mtcars$mpg)
```

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

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

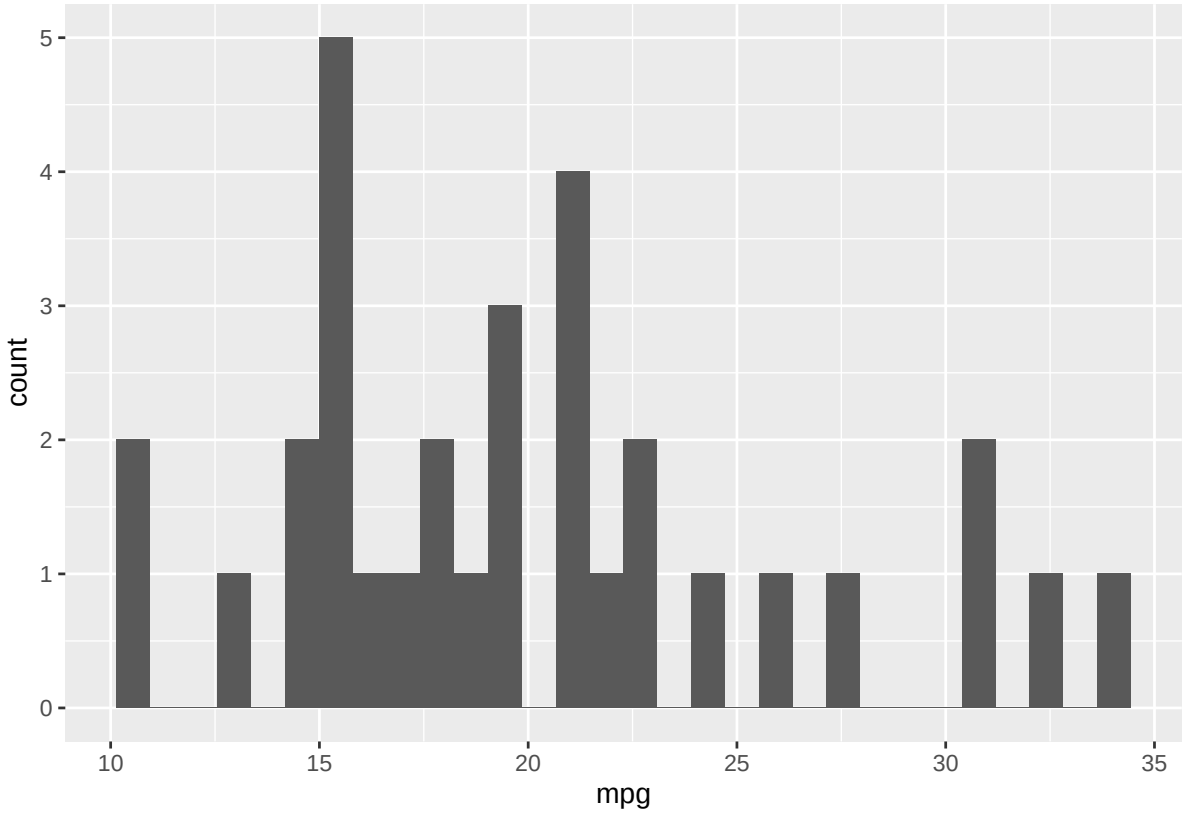
```
##    0%   20%   40%   60%   80%  100%
## 10.40 15.20 17.92 21.00 24.08 33.90
```

```r
hist(mtcars$mpg)
```

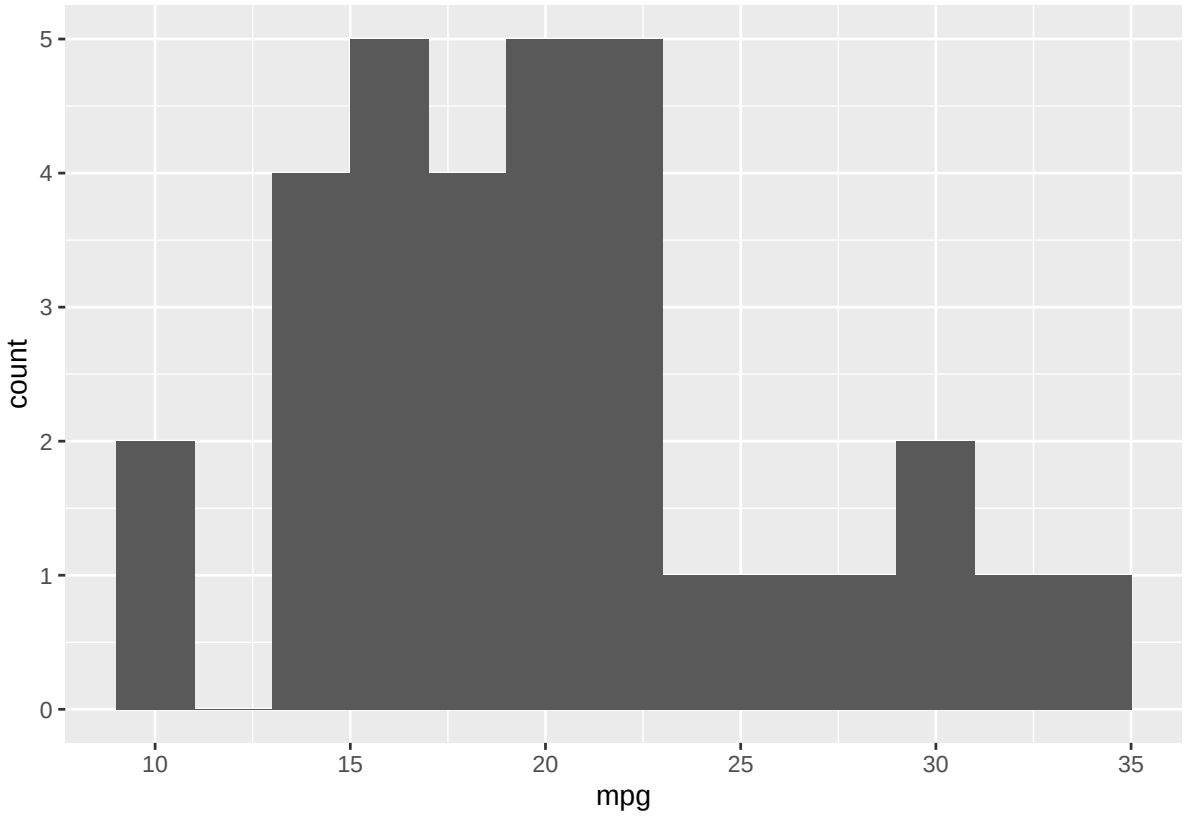



```r
ggplot(mtcars, aes(x = mpg)) + geom_histogram()
```

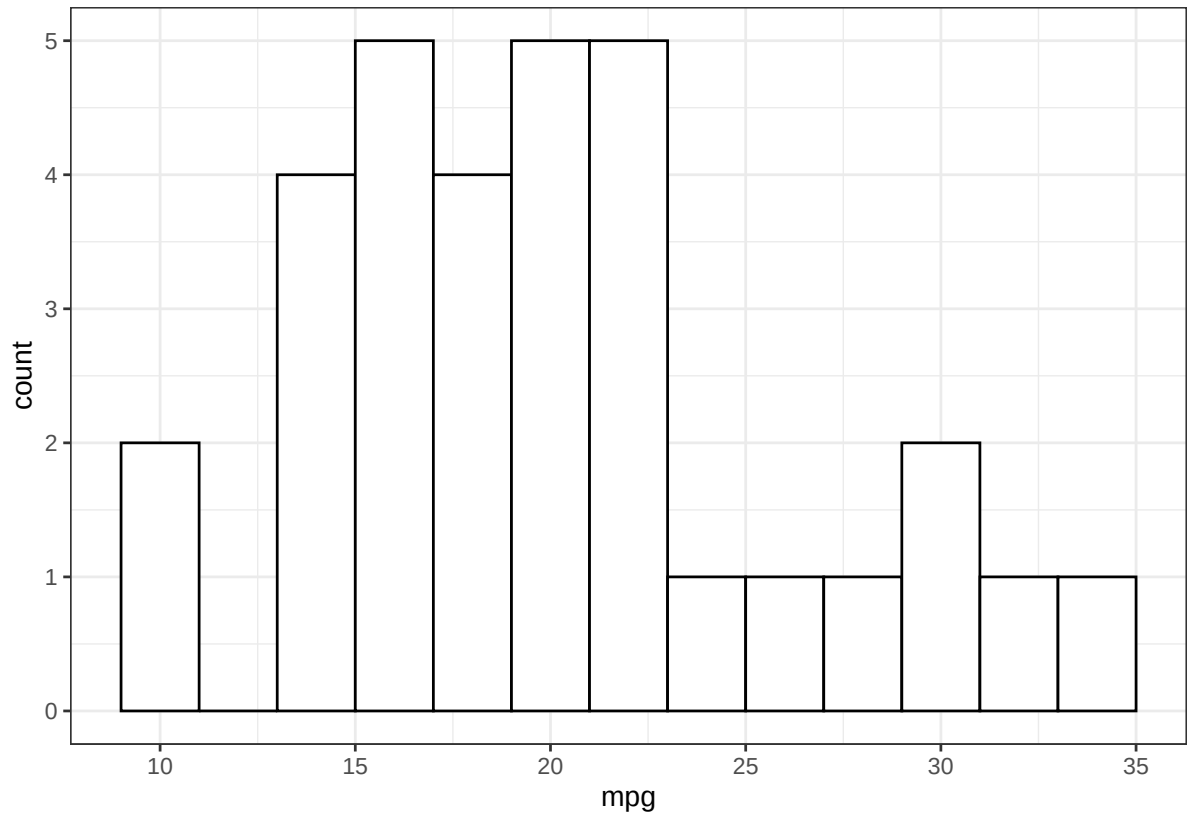
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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



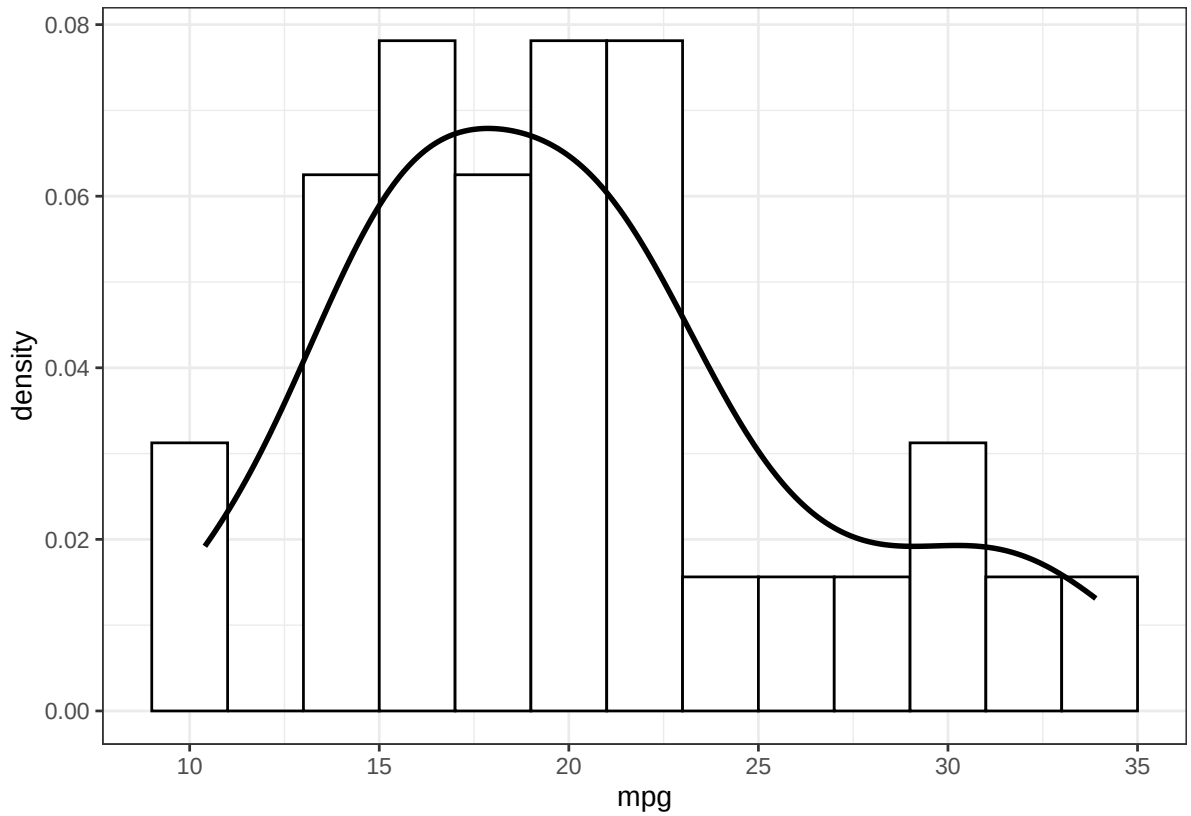

```
ggplot(mtcars, aes(x = mpg)) + geom_histogram(binwidth = 2, fill = "white",
    color = "black") + theme_bw()
```



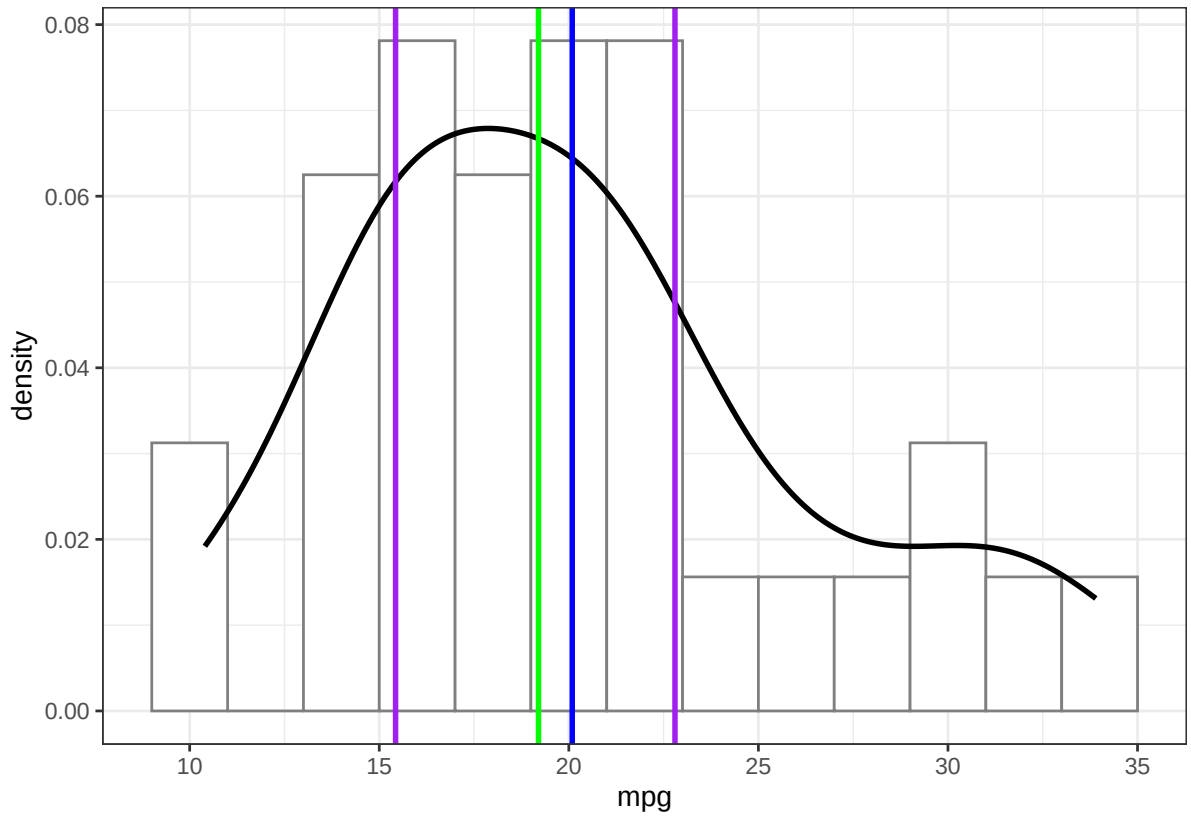

```
`?`(`?`(theme_bw))

ggplot(mtcars, aes(x = mpg)) + geom_histogram(aes(y = ..density..),
    binwidth = 2, fill = "white", color = "black") + geom_density(linewidth = 1) +
    theme_bw()
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
ggplot(mtcars, aes(x = mpg)) + geom_histogram(aes(y = ..density..),
    binwidth = 2, fill = "white", color = "grey50") + geom_density(linewidth = 1) +
    geom_vline(xintercept = 19.2, color = "green", linewidth = 1) +
    geom_vline(xintercept = 20.09, color = "blue", linewidth = 1) +
    geom_vline(xintercept = c(15.43, 22.8), color = "purple",
        linewidth = 1) + theme_bw()
```



Character/Factor (Nominal/Order/Categorical)

```
table(mtcars$cyl)
```

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

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

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

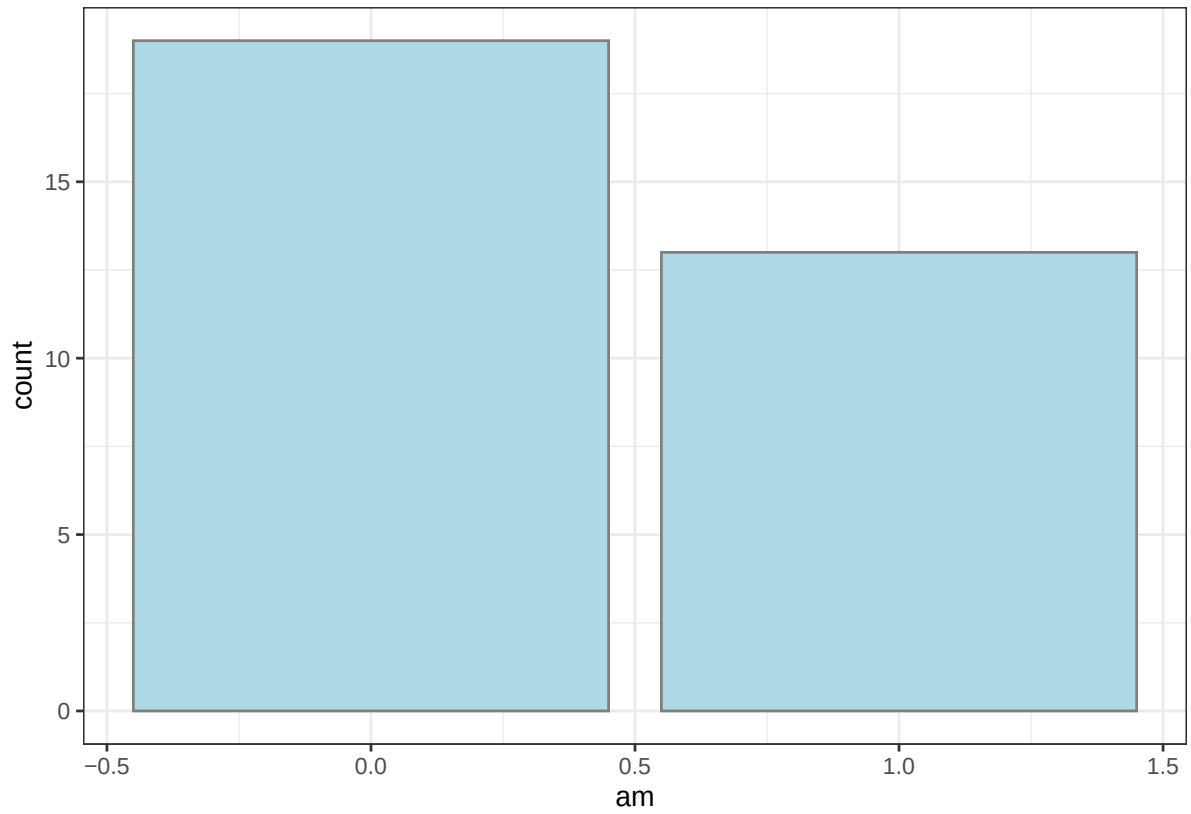
```
table(mtcars$am)
```

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

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

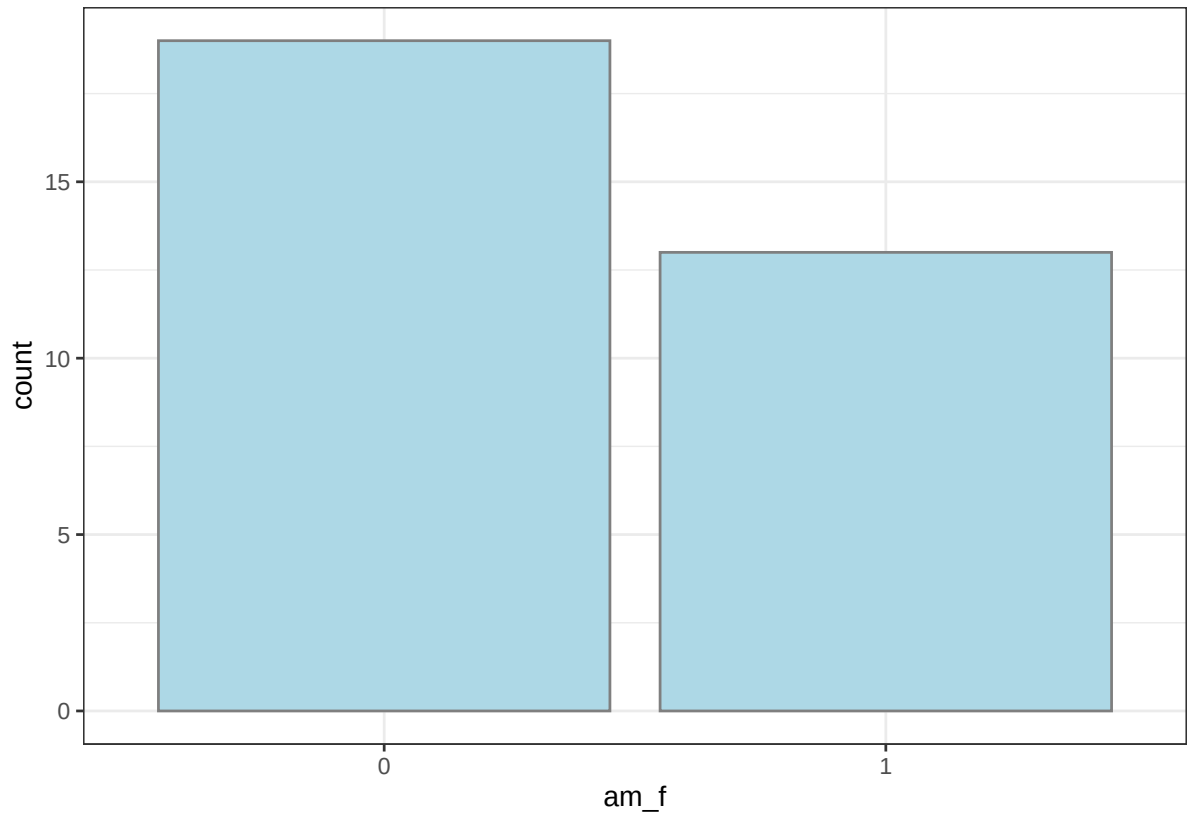
```
##
##       0       1
## 0.59375 0.40625
```

```
ggplot(mtcars, aes(x = am)) + geom_bar(fill = "lightblue", color = "grey50") +
    theme_bw()
```



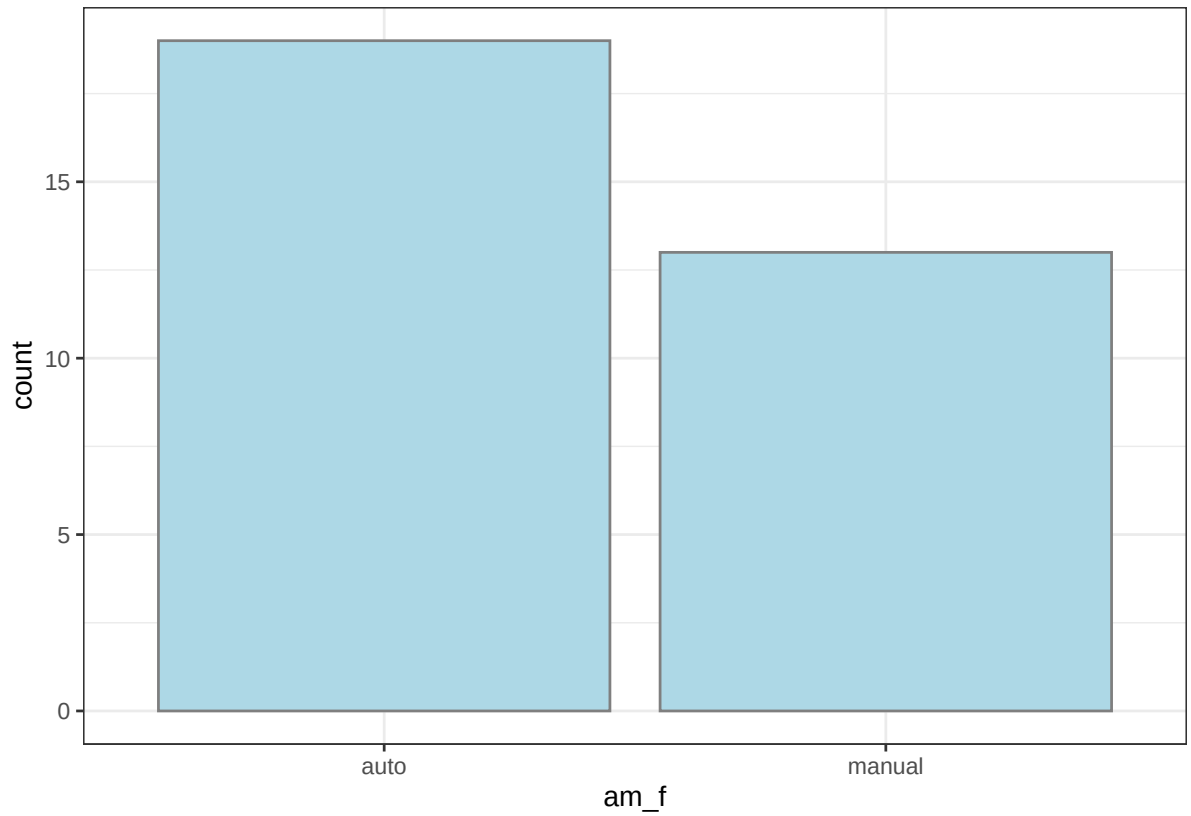

```
mtcars$am_f <- factor(mtcars$am)


ggplot(mtcars, aes(x = am_f)) + geom_bar(fill = "lightblue",
    color = "grey50") + theme_bw()
```

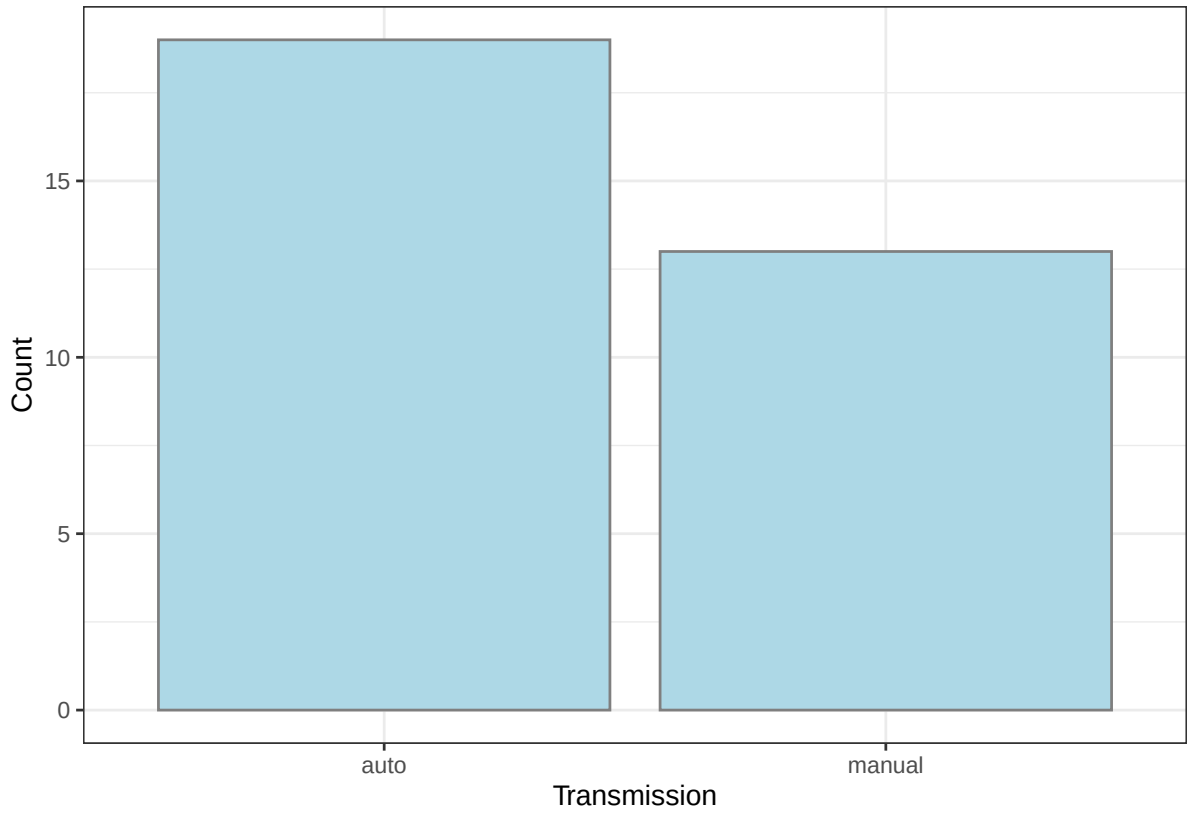



```
mtcars$am_f <- factor(mtcars$am, labels = c("auto", "manual"))


ggplot(mtcars, aes(x = am_f)) + geom_bar(fill = "lightblue",
    color = "grey50") + theme_bw()
```



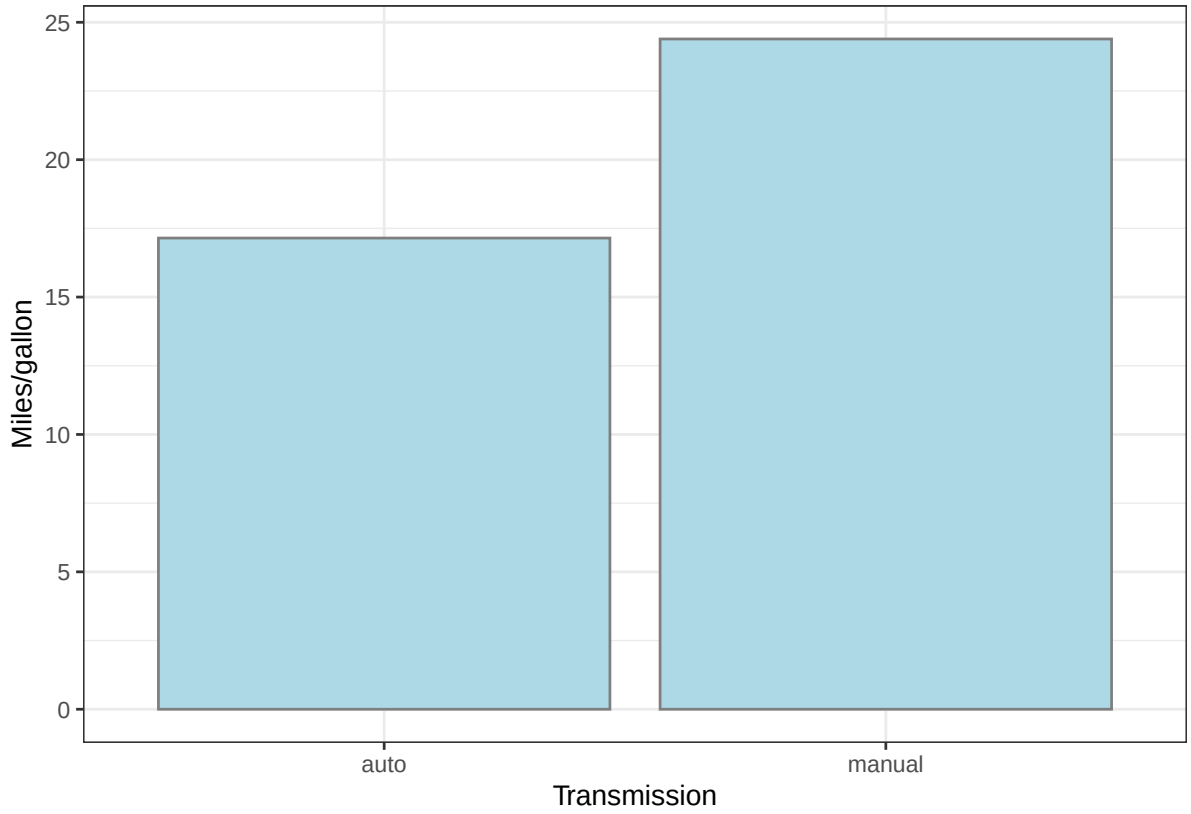

```
ggplot(mtcars, aes(x = am_f)) + geom_bar(fill = "lightblue",
    color = "grey50") + theme_bw() + labs(x = "Transmission",
    y = "Count")
```



3.3 Bivariate

- continuous & discrete

```
ggplot(mtcars, aes(x = am_f, y = mpg)) + geom_bar(stat = "summary",
    fun = "mean", fill = "lightblue", color = "grey50") + theme_bw() +
    labs(x = "Transmission", y = "Miles/gallon")
```

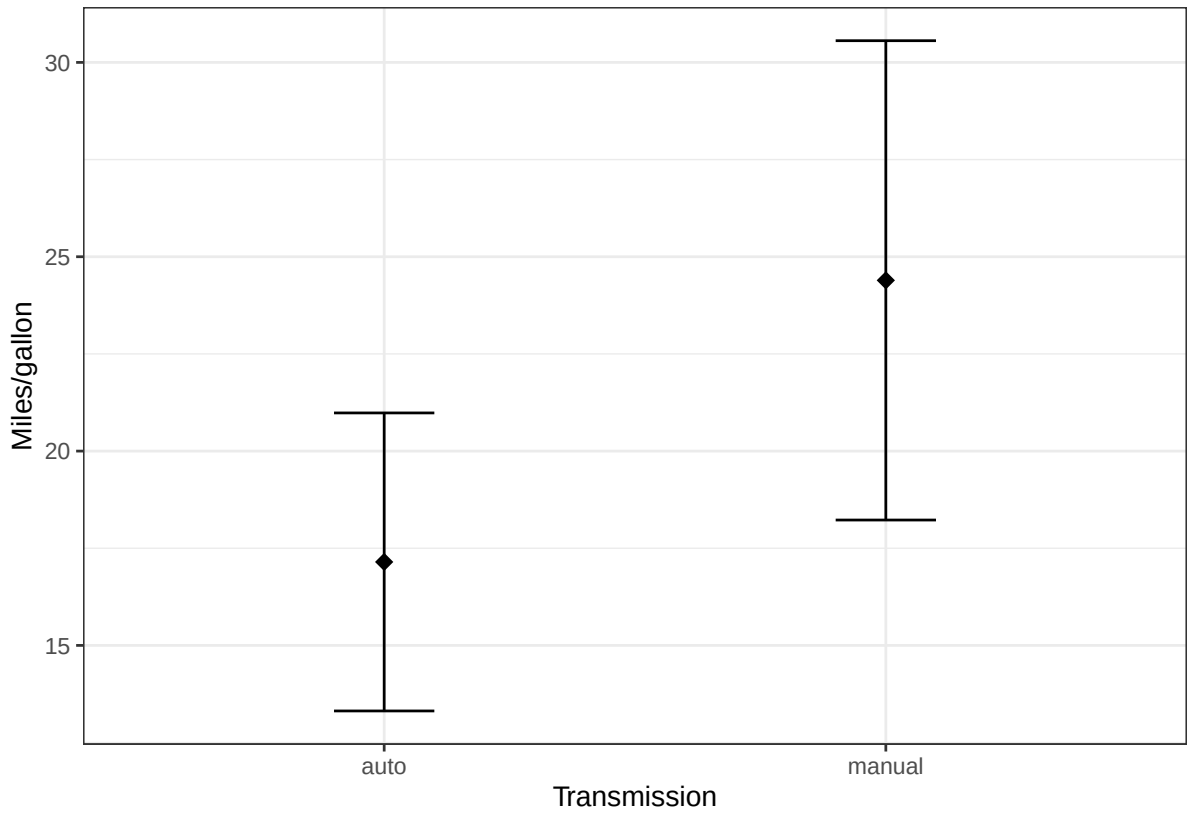



```
# install.packages('tidyverse')
library(tidyverse)
```

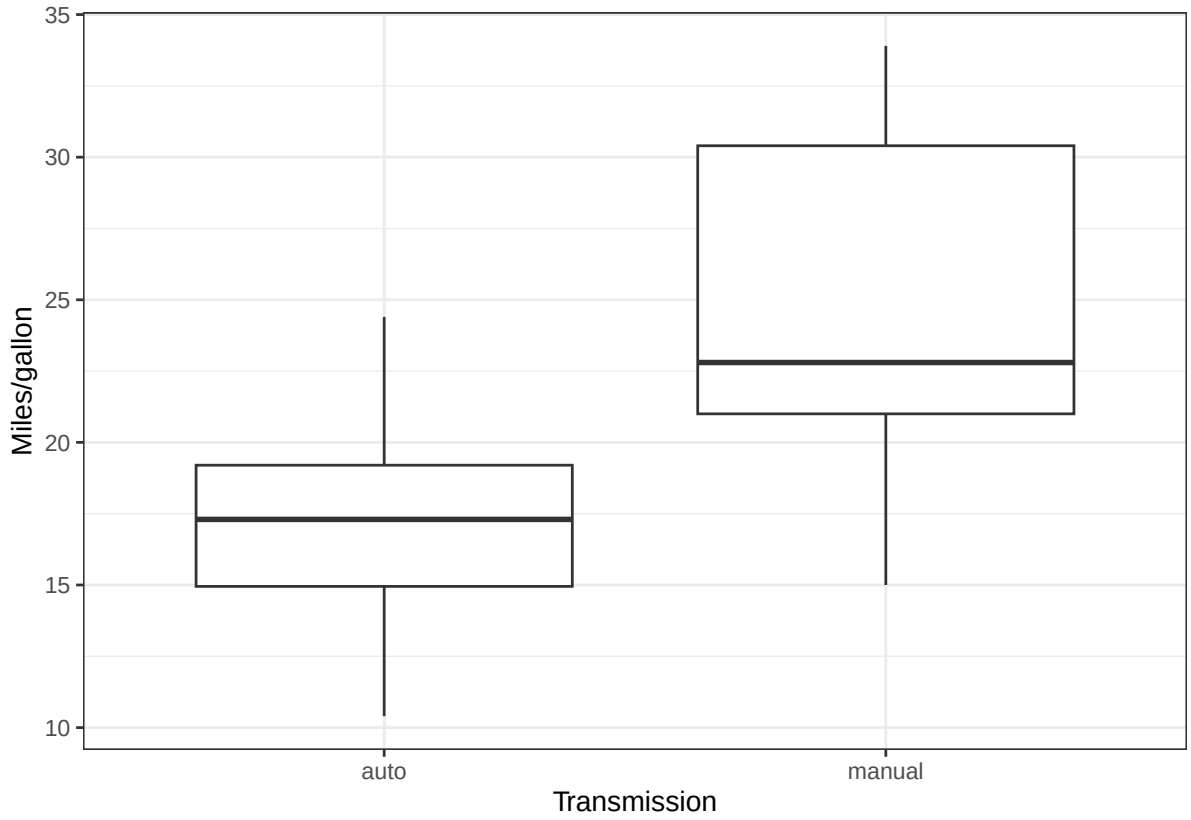
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become er
```

```
mpg_am <- mtcars %>%
    group_by(am_f) %>%
    summarise(mpg_mean = mean(mpg), mpg_sd = sd(mpg), mpg_min = min(mpg),
        mpg_max = max(mpg), mpg_q1 = quantile(mpg, 0.25), mpg_median = median(mpg),
        mpg_q3 = quantile(mpg, 0.75))

ggplot(mpg_am, aes(x = am_f, y = mpg_mean)) + geom_point(size = 3,
    shape = 18) + geom_errorbar(aes(ymin = mpg_mean - mpg_sd,
    ymax = mpg_mean + mpg_sd), width = 0.2) + theme_bw() + labs(x = "Transmission",
    y = "Miles/gallon")
```

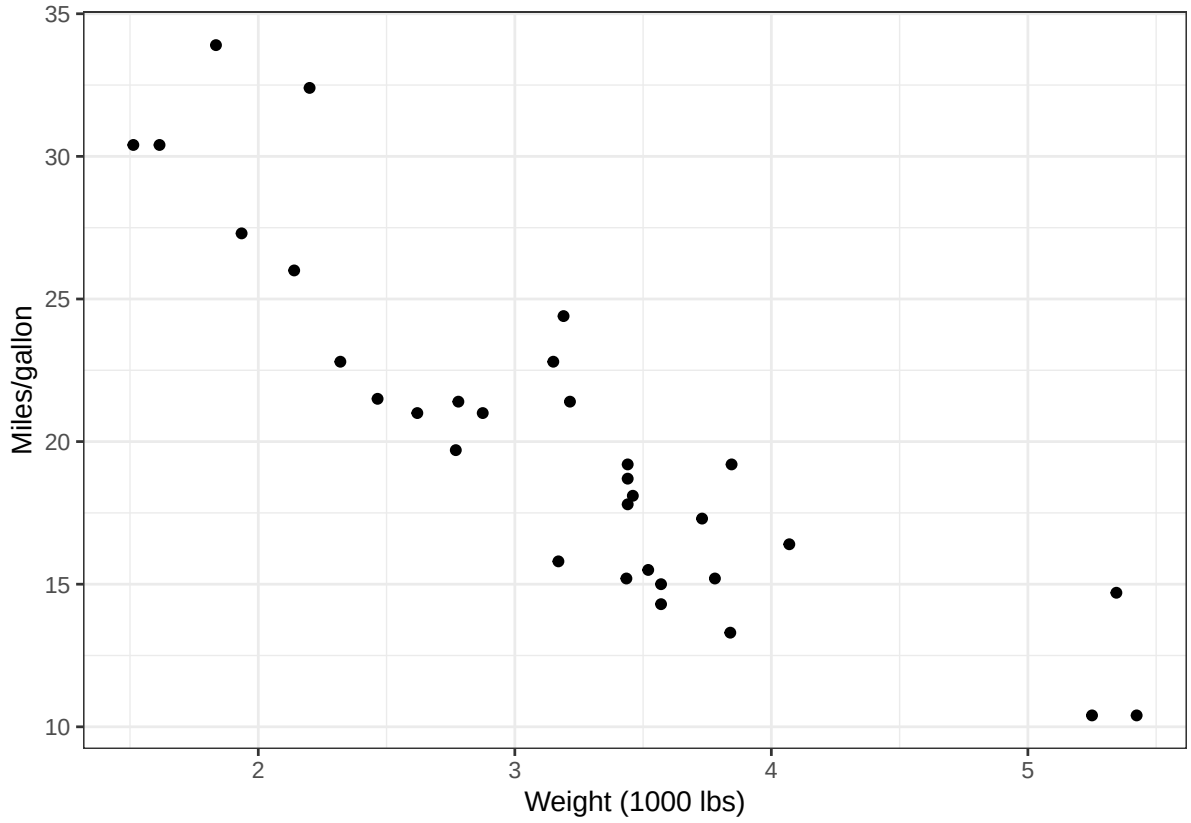



```
ggplot(mtcars, aes(x = am_f, y = mpg)) + geom_boxplot() + theme_bw() +
    labs(x = "Transmission", y = "Miles/gallon")
```



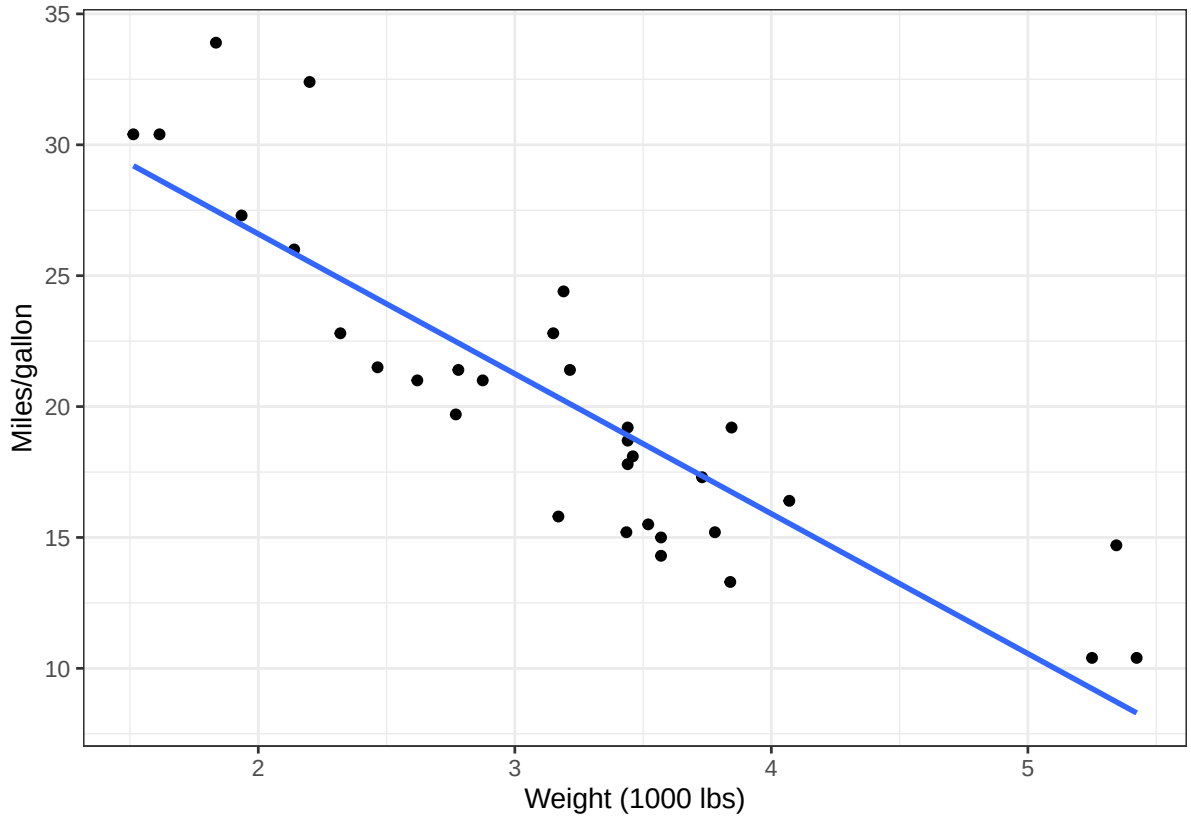
- Continuous & continuous

```
ggplot(mtcars, aes(x = wt, y = mpg)) + geom_point() + theme_bw() +
    labs(x = "Weight (1000 lbs)", y = "Miles/gallon")
```




```
ggplot(mtcars, aes(x = wt, y = mpg)) + geom_point() + geom_smooth(method = "lm",
    se = F) + theme_bw() + labs(x = "Weight (1000 lbs)", y = "Miles/gallon")
```

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




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

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

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

```
##
##  Pearson's product-moment correlation
##
## data:  mtcars$wt and mtcars$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(mtcars$cyl, mtcars$am_f)
```

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

```
prop.table(table(mtcars$cyl, mtcars$am_f), 1)
```

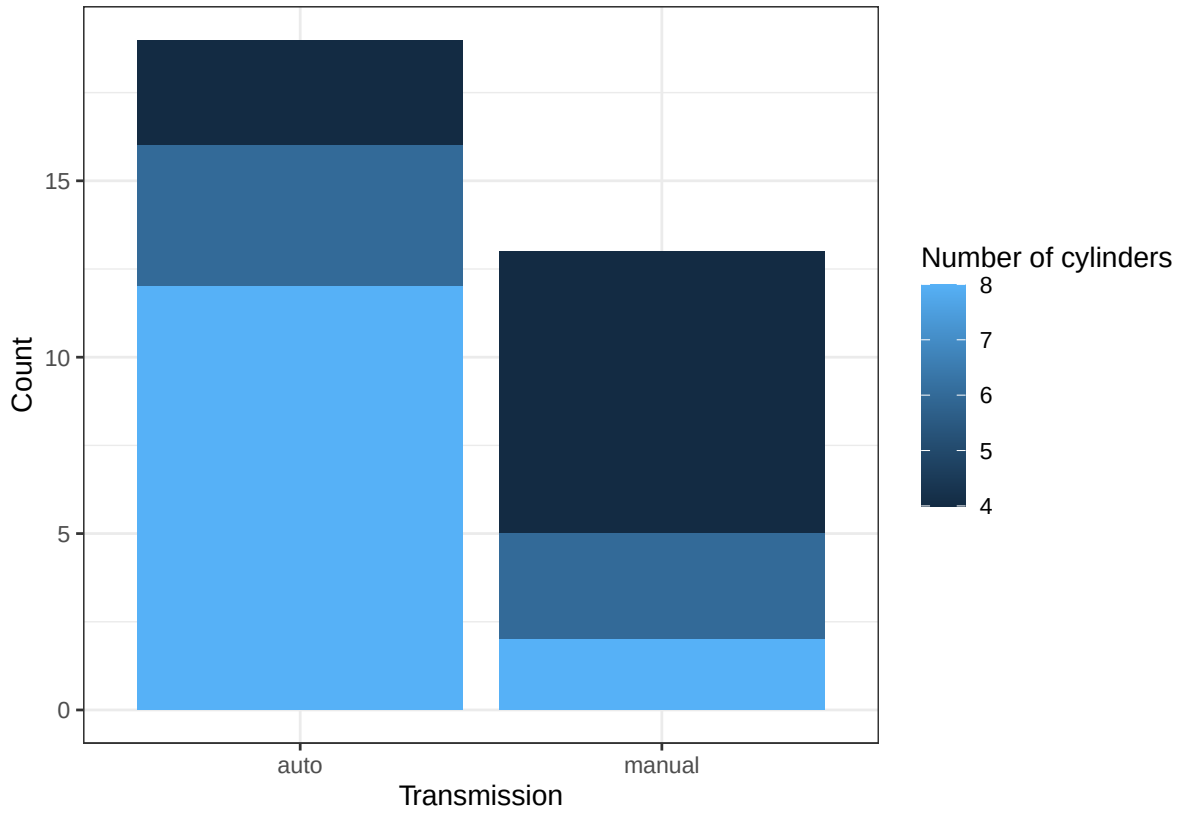
```
##
##          auto    manual
##   4 0.2727273 0.7272727
##   6 0.5714286 0.4285714
```

```
##   8 0.8571429 0.1428571
```

```
prop.table(table(mtcars$cyl, mtcars$am_f), 2)
```

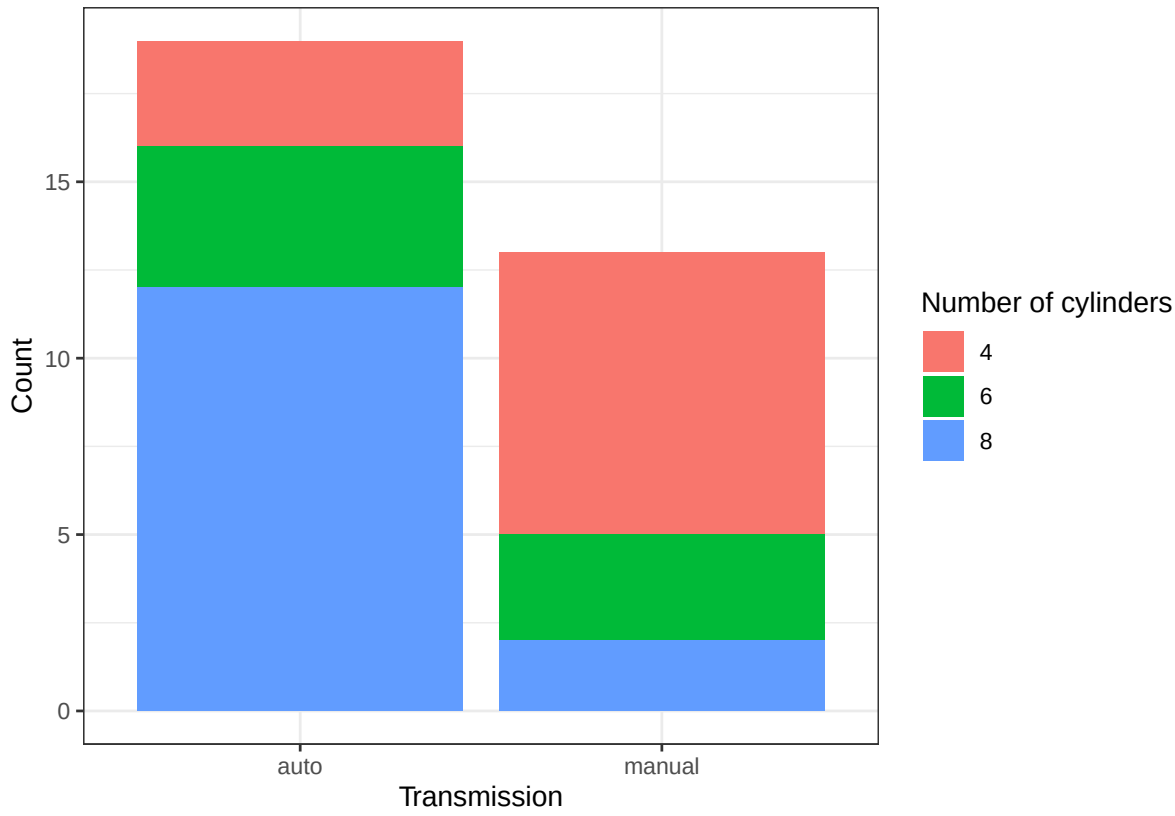
```
##
##          auto    manual
##   4 0.1578947 0.6153846
##   6 0.2105263 0.2307692
##   8 0.6315789 0.1538462
```

```
ggplot(mtcars, aes(x = am_f, fill = cyl, group = cyl)) + geom_bar() +
    theme_bw() + labs(x = "Transmission", y = "Count", fill = "Number of cylinders")
```

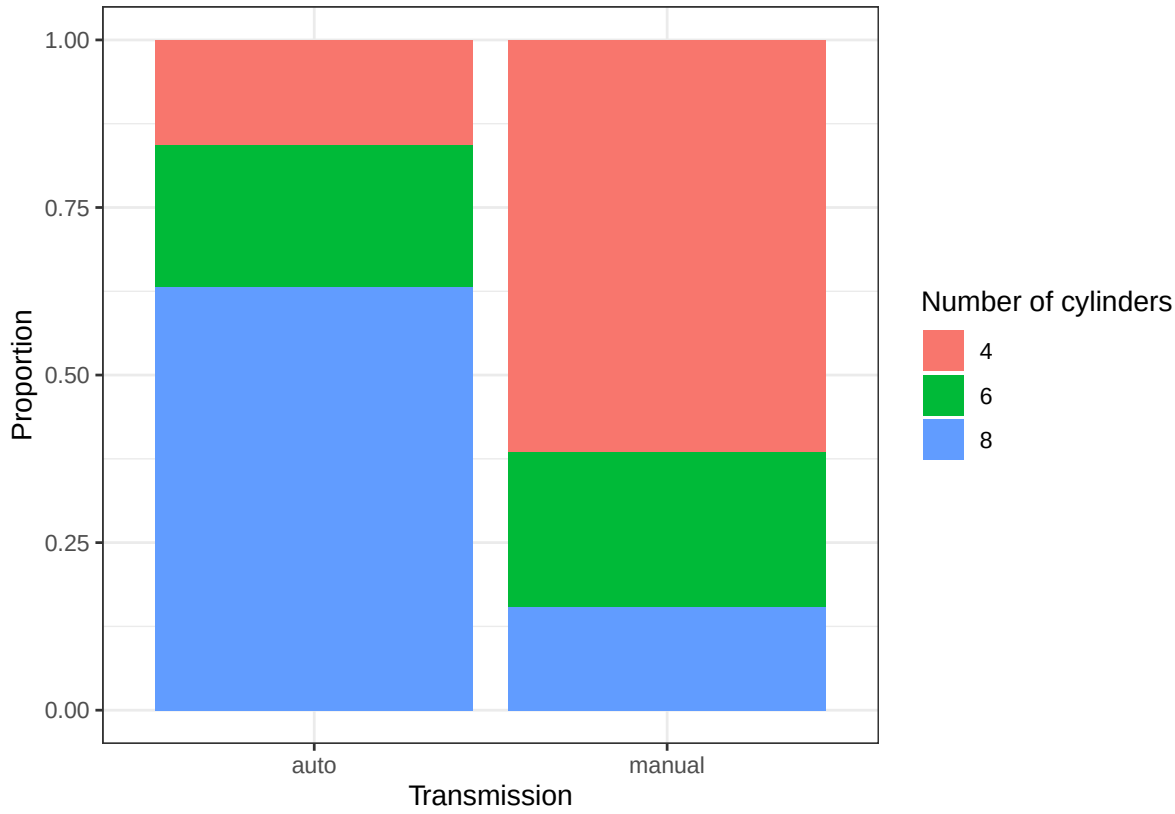



```
mtcars$cyl_f <- factor(mtcars$cyl)

ggplot(mtcars, aes(x = am_f, fill = cyl_f, group = cyl)) + geom_bar() +
    theme_bw() + labs(x = "Transmission", y = "Count", fill = "Number of cylinders")
```

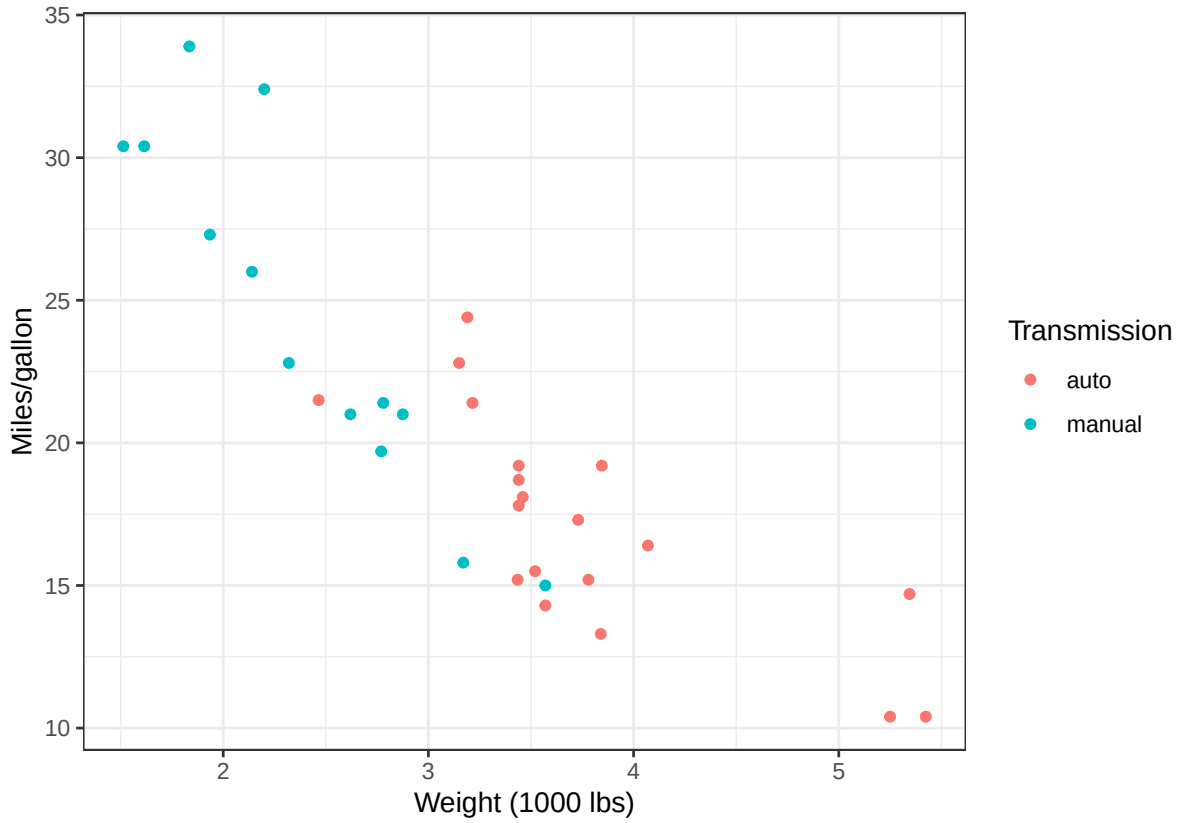



```
ggplot(mtcars, aes(x = am_f, fill = cyl_f, group = cyl)) + geom_bar(position = "fill") +
    theme_bw() + labs(x = "Transmission", y = "Proportion", fill = "Number of cylinders")
```



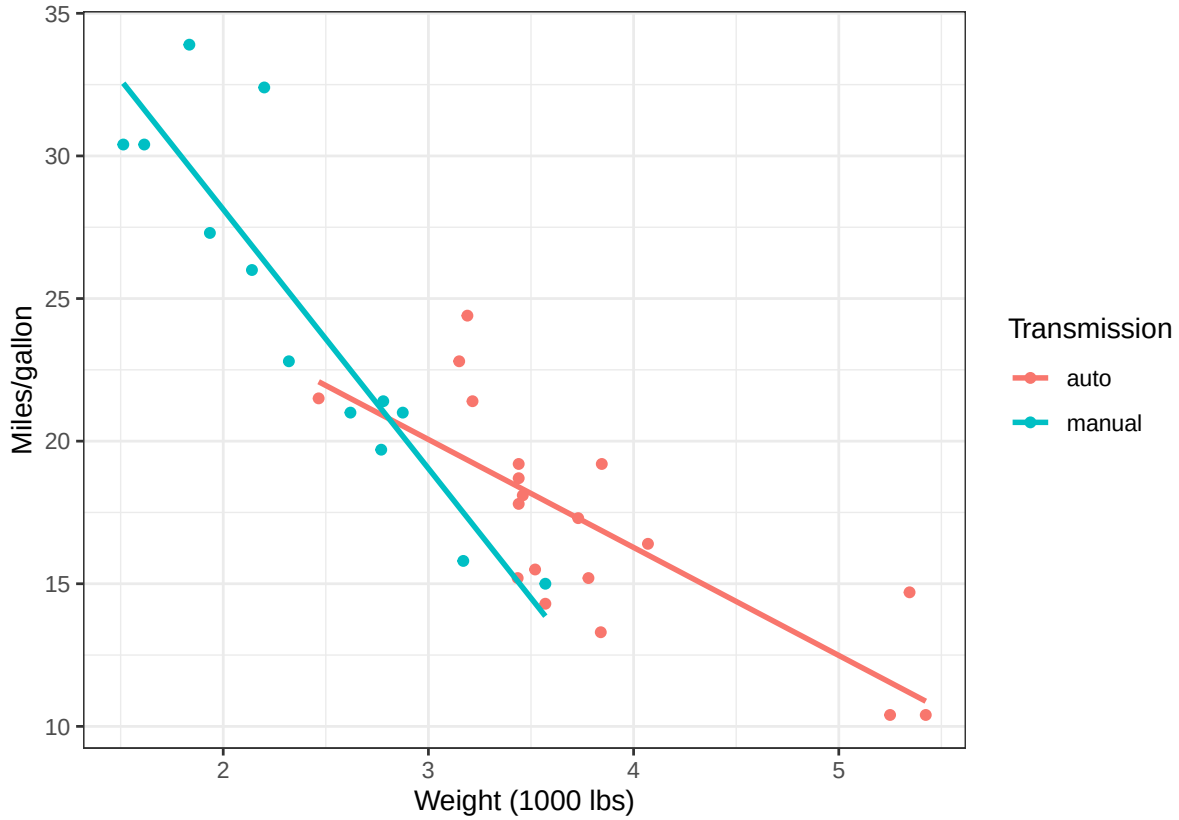
3.4 Add dimension

```
ggplot(mtcars, aes(x = wt, y = mpg, col = am_f)) + geom_point() +
    theme_bw() + labs(x = "Weight (1000 lbs)", y = "Miles/gallon",
    col = "Transmission")
```

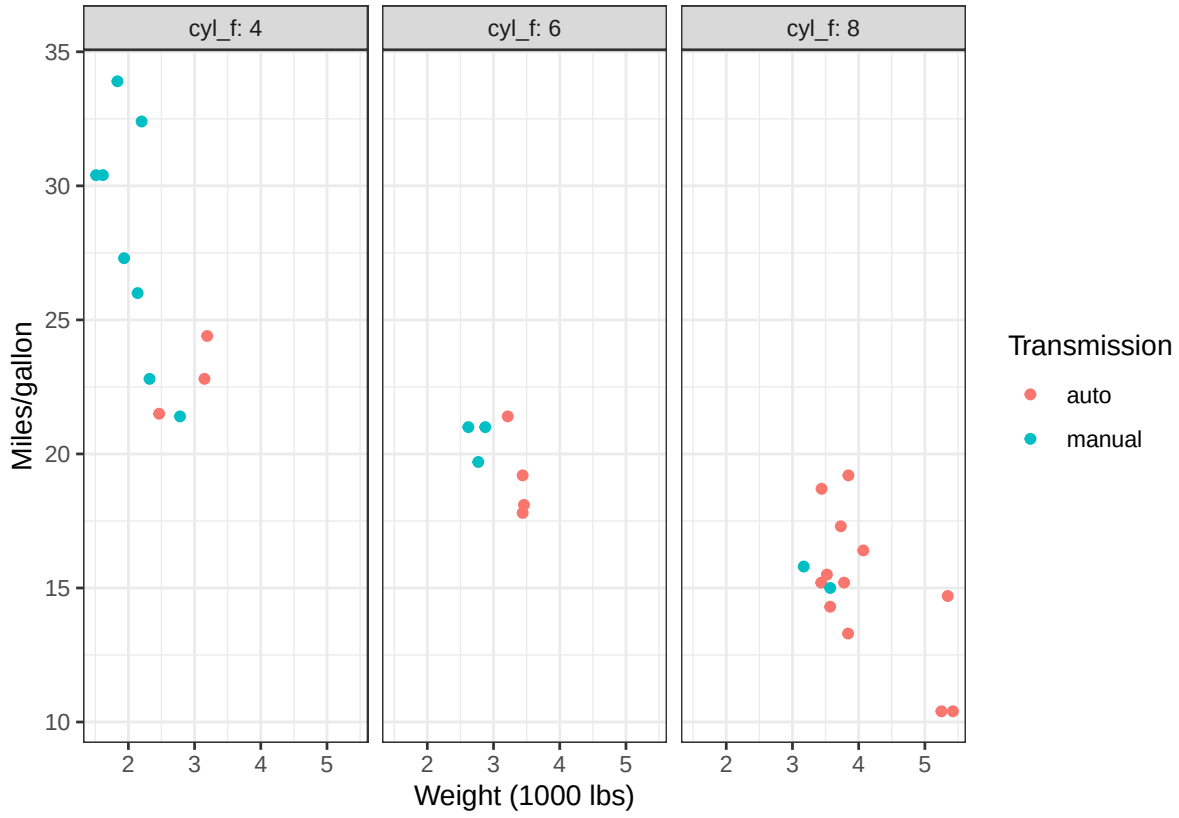



```
ggplot(mtcars, aes(x = wt, y = mpg, col = am_f)) + geom_point() +
    geom_smooth(method = "lm", se = F) + theme_bw() + labs(x = "Weight (1000 lbs)",
    y = "Miles/gallon", col = "Transmission")
```

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




```
ggplot(mtcars, aes(x = wt, y = mpg, col = am_f)) + geom_point() +
    facet_grid(. ~ cyl_f, labeller = labeller(.cols = label_both)) +
    theme_bw() + labs(x = "Weight (1000 lbs)", y = "Miles/gallon",
    col = "Transmission")
```



4 Linear regression

4.1 Compare means

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

```
##
## Call:
## lm(formula = mpg ~ am_f, data = mtcars)
##
## 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
```

4.2 Simple linear regression

```
fit2 <- lm(mpg ~ wt, mtcars)
summary(fit2)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## 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

```
fit3 <- lm(mpg ~ wt + am_f, mtcars)
summary(fit3)
```

```
##
## Call:
## lm(formula = mpg ~ wt + am_f, data = mtcars)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5295 -2.3619 -0.1317  1.4025  6.8782
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.32155    3.05464  12.218 5.84e-13 ***
## wt          -5.35281    0.78824  -6.791 1.87e-07 ***
## am_fmanual  -0.02362    1.54565  -0.015    0.988
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.098 on 29 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7358
## F-statistic: 44.17 on 2 and 29 DF,  p-value: 1.579e-09
```

```
fit4 <- lm(mpg ~ wt + I(wt^2), mtcars)
summary(fit4)
```

```
##
## Call:
## lm(formula = mpg ~ wt + I(wt^2), data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.483 -1.998 -0.773  1.462  6.238
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  49.9308     4.2113  11.856 1.21e-12 ***
## wt          -13.3803     2.5140  -5.322 1.04e-05 ***
## I(wt^2)       1.1711     0.3594   3.258  0.00286 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.651 on 29 degrees of freedom
## Multiple R-squared:  0.8191, Adjusted R-squared:  0.8066
## F-statistic: 65.64 on 2 and 29 DF,  p-value: 1.715e-11
```

```
AIC(fit2)
```

```
## [1] 166.0294
```

```
AIC(fit3)
```

```
## [1] 168.0292
```

```
AIC(fit4)
```

```
## [1] 158.0484
```

```
anova(fit3, fit2)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + am_f
## Model 2: mpg ~ wt
##   Res.Df    RSS Df  Sum of Sq     F Pr(>F)
## 1     29 278.32
## 2     30 278.32 -1 -0.0022403 2e-04 0.9879
```

```r
anova(fit4, fit2)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + I(wt^2)
## Model 2: mpg ~ wt
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     29 203.75
## 2     30 278.32 -1   -74.576 10.615 0.00286 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# plot(fit4)
```

5 Factor analysis

```r
# install.packages('lavaan')
library(lavaan)
```

```
## This is lavaan 0.6-18
## lavaan is FREE software! Please report any bugs.
```

```r
round(cor(env[, 9:20], method = "spearman"), 2)
```

```
##        Q1   Q2    Q3    Q4    Q5    Q6    Q7   Q8    Q9  Q10   Q11   Q12
## Q1   1.00 0.28  0.49  0.54  0.62 -0.32  0.40 0.24  0.30 0.21  0.21  0.35
## Q2   0.28 1.00  0.30  0.29  0.24  0.07  0.40 0.39  0.43 0.23  0.40  0.42
## Q3   0.49 0.30  1.00  0.47  0.50 -0.24  0.31 0.31  0.34 0.24  0.30  0.39
## Q4   0.54 0.29  0.47  1.00  0.63 -0.26  0.43 0.36  0.38 0.33  0.36  0.47
## Q5   0.62 0.24  0.50  0.63  1.00 -0.30  0.39 0.25  0.31 0.26  0.25  0.35
## Q6  -0.32 0.07 -0.24 -0.26 -0.30  1.00 -0.04 0.04 -0.05 0.01 -0.02 -0.09
## Q7   0.40 0.40  0.31  0.43  0.39 -0.04  1.00 0.49  0.47 0.32  0.51  0.51
## Q8   0.24 0.39  0.31  0.36  0.25  0.04  0.49 1.00  0.53 0.36  0.47  0.56
## Q9   0.30 0.43  0.34  0.38  0.31 -0.05  0.47 0.53  1.00 0.39  0.54  0.60
## Q10  0.21 0.23  0.24  0.33  0.26  0.01  0.32 0.36  0.39 1.00  0.45  0.45
## Q11  0.21 0.40  0.30  0.36  0.25 -0.02  0.51 0.47  0.54 0.45  1.00  0.75
## Q12  0.35 0.42  0.39  0.47  0.35 -0.09  0.51 0.56  0.60 0.45  0.75  1.00
```

```r
att_bhv <- env[, 9:20]


model1 <- "
  att =~ Q1 + Q2 + Q3 + Q4 + Q5 + Q6 + Q7
  bhv =~ Q8 + Q9 + Q10 + Q11 + Q12
"

cfa_result <- cfa(model1, data = att_bhv, std.lv = TRUE)
summary(cfa_result, fit.measures = T, standardized = T)
```

```
## lavaan 0.6-18 ended normally after 25 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        25
##
##   Number of observations                           312
##
## Model Test User Model:
##
```

```
##   Test statistic                                226.563
##   Degrees of freedom                                 53
##   P-value (Chi-square)                            0.000
##
## Model Test Baseline Model:
##
##   Test statistic                               1690.447
##   Degrees of freedom                                 66
##   P-value                                         0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                     0.893
##   Tucker-Lewis Index (TLI)                        0.867
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)               -4957.378
##   Loglikelihood unrestricted model (H1)       -4844.096
##
##   Akaike (AIC)                                 9964.755
##   Bayesian (BIC)                              10058.330
##   Sample-size adjusted Bayesian (SABIC)        9979.039
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                           0.102
##   90 Percent confidence interval - lower          0.089
##   90 Percent confidence interval - upper          0.116
##   P-value H_0: RMSEA <= 0.050                     0.000
##   P-value H_0: RMSEA >= 0.080                     0.996
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                            0.098
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   att =~
##     Q1                0.625    0.040   15.478    0.000    0.625    0.771
##     Q2                0.525    0.064    8.224    0.000    0.525    0.464
##     Q3                0.638    0.045   14.207    0.000    0.638    0.725
##     Q4                0.801    0.047   16.862    0.000    0.801    0.817
##     Q5                0.703    0.041   17.356    0.000    0.703    0.833
##     Q6               -0.239    0.114   -2.087    0.037   -0.239   -0.125
##     Q7                0.663    0.065   10.211    0.000    0.663    0.559
##   bhv =~
##     Q8                0.756    0.064   11.793    0.000    0.756    0.630
##     Q9                0.756    0.059   12.869    0.000    0.756    0.675
##     Q10               0.626    0.069    9.067    0.000    0.626    0.507
##     Q11               0.947    0.056   17.032    0.000    0.947    0.827
##     Q12               1.017    0.055   18.520    0.000    1.017    0.875
```

```
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   att ~~
##     bhv               0.635    0.042   15.179    0.000    0.635    0.635
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Q1                0.267    0.026   10.186    0.000    0.267    0.406
##    .Q2                1.004    0.083   12.066    0.000    1.004    0.784
##    .Q3                0.367    0.034   10.756    0.000    0.367    0.475
##    .Q4                0.319    0.034    9.302    0.000    0.319    0.332
##    .Q5                0.218    0.025    8.896    0.000    0.218    0.306
##    .Q6                3.611    0.290   12.466    0.000    3.611    0.984
##    .Q7                0.966    0.082   11.785    0.000    0.966    0.687
##    .Q8                0.866    0.076   11.411    0.000    0.866    0.603
##    .Q9                0.683    0.061   11.112    0.000    0.683    0.545
##    .Q10               1.134    0.095   11.932    0.000    1.134    0.743
##    .Q11               0.414    0.048    8.686    0.000    0.414    0.316
##    .Q12               0.316    0.046    6.950    0.000    0.316    0.234
##     att               1.000                              1.000    1.000
##     bhv               1.000                              1.000    1.000
```

```
# install.packages('psych')
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:lavaan':
##
##     cor2cov
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
att_bhv_sc <- scale(att_bhv)

fa <- fa(r = att_bhv_sc, nfactors = 2)
```

```
## Loading required namespace: GPArotation
```

```
summary(fa)
```

```
##
## Factor analysis with Call: fa(r = att_bhv_sc, nfactors = 2)
##
## Test of the hypothesis that 2 factors are sufficient.
## The degrees of freedom for the model is 43  and the objective function was  0.26
## The number of observations was  312  with Chi Square =  78.74  with prob <  0.00072
##
## The root mean square of the residuals (RMSA) is  0.03
## The df corrected root mean square of the residuals is  0.04
##
## Tucker Lewis Index of factoring reliability =  0.965
## RMSEA index =  0.052  and the 10 % confidence intervals are  0.033 0.07
## BIC =  -168.21
##  With factor correlations of
##      MR1  MR2
## MR1 1.00 0.53
```

```
## MR2 0.53 1.00
```

```
fa$loadings
```

```
##
## Loadings:
##     MR1    MR2
## Q1          0.841
## Q2   0.506  0.111
## Q3          0.670
## Q4   0.172  0.704
## Q5          0.856
## Q6   0.370 -0.391
## Q7   0.594  0.153
## Q8   0.689
## Q9   0.662
## Q10  0.455
## Q11  0.856
## Q12  0.763
##
##                  MR1   MR2
## SS loadings    3.224 2.601
## Proportion Var 0.269 0.217
## Cumulative Var 0.269 0.485
```

```
model2 <- "
  att =~ Q1 + Q3 + Q4 + Q5 + Q7
  bhv =~ Q8 + Q9 + Q11 + Q12
"

cfa_result <- cfa(model2, data = env, std.lv = TRUE)
summary(cfa_result, fit.measures = T, standardized = T)
```

```
## lavaan 0.6-18 ended normally after 23 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        19
##
##   Number of observations                           312
##
## Model Test User Model:
##
##   Test statistic                               128.369
##   Degrees of freedom                                26
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              1441.337
##   Degrees of freedom                                36
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.927
##   Tucker-Lewis Index (TLI)                       0.899
##
## Loglikelihood and Information Criteria:
##
```

```
##   Loglikelihood user model (H0)              -3397.504
##   Loglikelihood unrestricted model (H1)      -3333.320
##
##   Akaike (AIC)                                6833.009
##   Bayesian (BIC)                              6904.126
##   Sample-size adjusted Bayesian (SABIC)       6843.864
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.112
##   90 Percent confidence interval - lower         0.093
##   90 Percent confidence interval - upper         0.132
##   P-value H_0: RMSEA <= 0.050                    0.000
##   P-value H_0: RMSEA >= 0.080                    0.997
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.087
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   att =~
##     Q1                0.629    0.040   15.597    0.000    0.629    0.775
##     Q3                0.636    0.045   14.129    0.000    0.636    0.723
##     Q4                0.801    0.048   16.856    0.000    0.801    0.818
##     Q5                0.713    0.040   17.661    0.000    0.713    0.844
##     Q7                0.644    0.065    9.854    0.000    0.644    0.543
##   bhv =~
##     Q8                0.748    0.064   11.628    0.000    0.748    0.624
##     Q9                0.751    0.059   12.735    0.000    0.751    0.671
##     Q11               0.944    0.056   16.845    0.000    0.944    0.825
##     Q12               1.029    0.055   18.659    0.000    1.029    0.885
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   att ~~
##     bhv               0.606    0.044   13.799    0.000    0.606    0.606
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Q1                0.262    0.026   10.052    0.000    0.262    0.399
##    .Q3                0.370    0.034   10.741    0.000    0.370    0.478
##    .Q4                0.318    0.035    9.193    0.000    0.318    0.331
##    .Q5                0.205    0.024    8.466    0.000    0.205    0.288
##    .Q7                0.990    0.084   11.829    0.000    0.990    0.705
##    .Q8                0.877    0.077   11.412    0.000    0.877    0.610
##    .Q9                0.690    0.062   11.096    0.000    0.690    0.550
##    .Q11               0.419    0.049    8.490    0.000    0.419    0.320
##    .Q12               0.293    0.048    6.162    0.000    0.293    0.217
##     att               1.000                              1.000    1.000
##     bhv               1.000                              1.000    1.000
```

```r
model3 <- "
  att =~ Q1 + Q3 + Q5
  bhv =~ Q8 + Q9 + Q10 + Q11 + Q12
"

cfa_result <- cfa(model3, data = env, std.lv = TRUE)
summary(cfa_result, fit.measures = T, standardized = T)
```

```
## lavaan 0.6-18 ended normally after 20 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        17
##
##   Number of observations                           312
##
## Model Test User Model:
##
##   Test statistic                                41.165
##   Degrees of freedom                                19
##   P-value (Chi-square)                           0.002
##
## Model Test Baseline Model:
##
##   Test statistic                              1081.055
##   Degrees of freedom                                28
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.979
##   Tucker-Lewis Index (TLI)                       0.969
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3110.560
##   Loglikelihood unrestricted model (H1)      -3089.977
##
##   Akaike (AIC)                                6255.120
##   Bayesian (BIC)                              6318.751
##   Sample-size adjusted Bayesian (SABIC)       6264.833
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.061
##   90 Percent confidence interval - lower         0.035
##   90 Percent confidence interval - upper         0.087
##   P-value H_0: RMSEA <= 0.050                    0.216
##   P-value H_0: RMSEA >= 0.080                    0.120
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.042
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
```

```
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   att =~
##     Q1               0.650    0.041   15.688    0.000    0.650    0.802
##     Q3               0.654    0.046   14.256    0.000    0.654    0.744
##     Q5               0.708    0.043   16.631    0.000    0.708    0.839
##   bhv =~
##     Q8               0.751    0.064   11.690    0.000    0.751    0.627
##     Q9               0.752    0.059   12.765    0.000    0.752    0.672
##     Q10              0.624    0.069    9.017    0.000    0.624    0.505
##     Q11              0.952    0.056   17.106    0.000    0.952    0.832
##     Q12              1.018    0.055   18.433    0.000    1.018    0.875
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   att ~~
##     bhv              0.532    0.050   10.636    0.000    0.532    0.532
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Q1               0.235    0.029    8.100    0.000    0.235    0.357
##    .Q3               0.346    0.036    9.592    0.000    0.346    0.447
##    .Q5               0.211    0.031    6.841    0.000    0.211    0.296
##    .Q8               0.872    0.076   11.404    0.000    0.872    0.607
##    .Q9               0.689    0.062   11.100    0.000    0.689    0.549
##    .Q10              1.137    0.095   11.923    0.000    1.137    0.745
##    .Q11              0.404    0.048    8.378    0.000    0.404    0.308
##    .Q12              0.316    0.047    6.744    0.000    0.316    0.234
##     att              1.000                              1.000    1.000
##     bhv              1.000                              1.000    1.000
```