Descriptive statistics/plots continued and data transformations

Overview

Review

- Plots and statistics of categorical data
- Measures of central tendency

Continuation of statistics and plots of quantitative data

- Measures of spread
- Two quantitative variables

Start on data transformations using dplyr

Reminder: Homework 2

It is due on Gradescope by 11pm on Monday July 14th

- Question 4 involves reading a short article and commented on it, so you can get started on this right away

How is the homework going so far?

Review: Categorical data

Categorical variables take on one of a fixed number of possible values

For categorical variables we usually want to view:

- **Frequency table**: How many items are each category   or
- **Relative frequency table**: The proportion (or percentage) of items in each category

```
# Vector of drinking behavior
> drinking_vec <- profiles$drinks

# Frequency and relative frequency tables
> drinks_table <- table(drinking_vec)
> prop.table(drinks_table)
```

	age	body_type	diet	drinks	drugs	education
1	22	a little extra	strictly anything	socially	never	working on college/university
2	35	average	mostly other	often	sometimes	working on space camp
3	38	thin	anything	socially	NA	graduated from masters program
4	23	thin	vegetarian	socially	NA	working on college/university
5	29	athletic	NA	socially	never	graduated from college/university
6	29	average	mostly anything	socially	NA	graduated from college/university

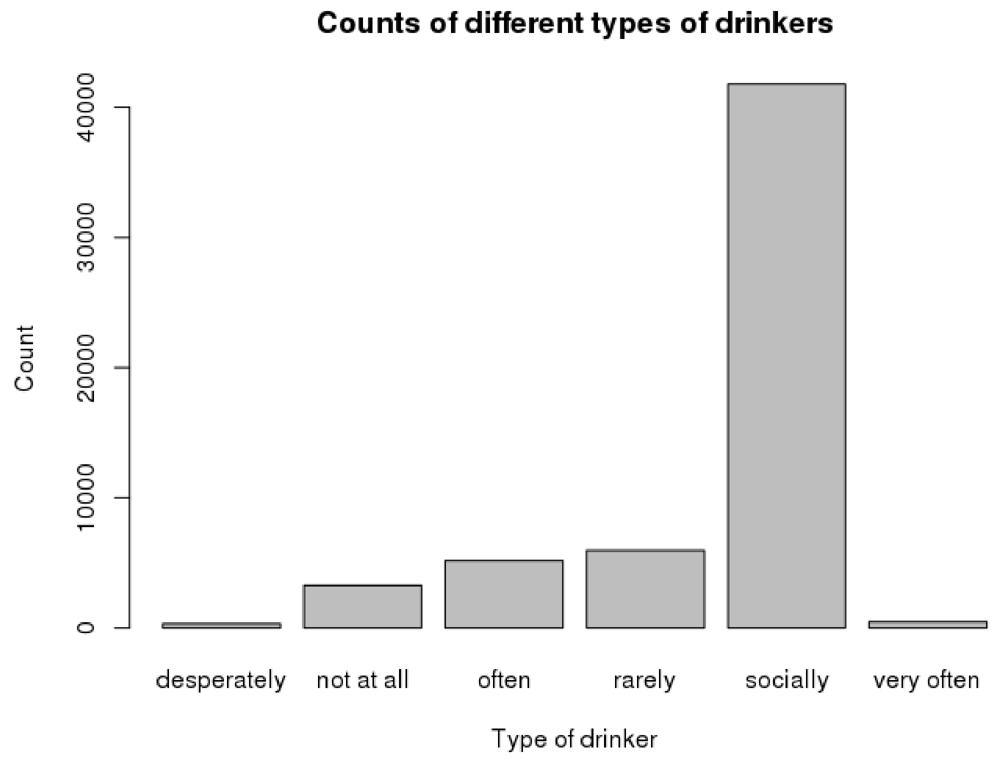
Review: Visualizing categorical data

We can plot the number of items in each category using a bar plot

```
barplot(drinks_table,
        ylab = "Count",
        xlab = "Type of drinker")
```

We can also use the pie() function to create pie charts

```
pie(drinks_table)
```

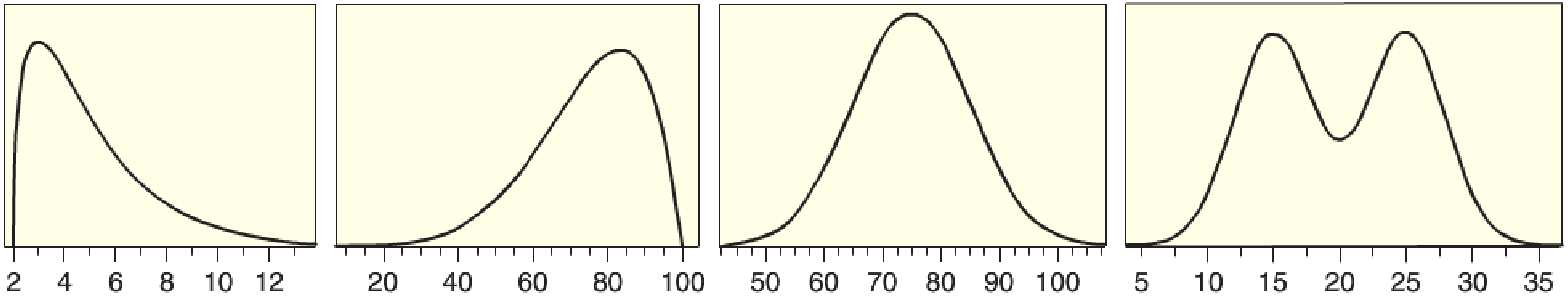


Review Visualizing quantitative data

We can visualize quantitative data using histograms

```
hist(profiles$height, breaks = 50)
```

Common shapes of histograms are:



(a) Skewed to the right (b) Skewed to the left (c) Symmetric and bell-shaped (d) Symmetric but not bell-shaped

Review Measures of central tendency

One common measure of central tendency is the **mean**

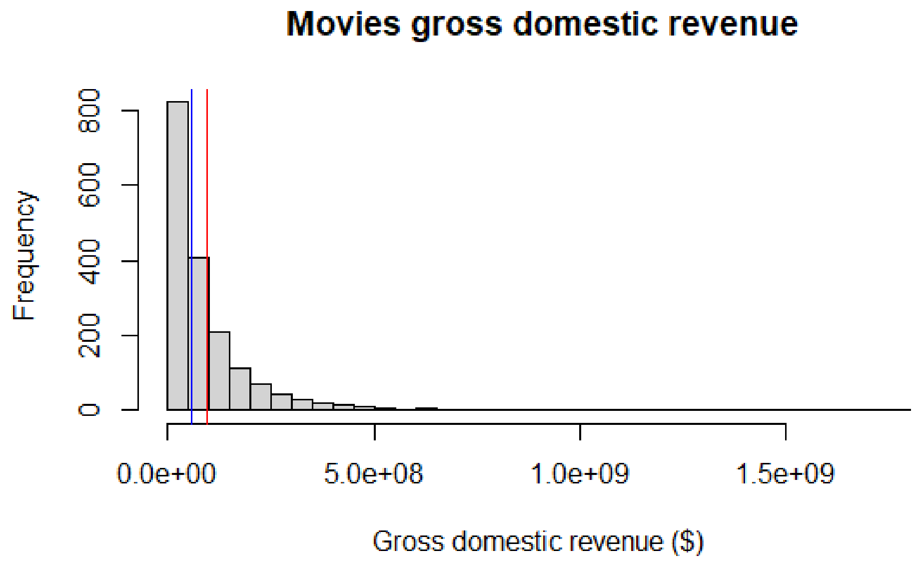
```
mean(x, na.rm = TRUE)
```

$$\frac{1}{n}\sum_{i=1}^{n} x_i$$

The **median** is the value such that half of the data is less than the median and half are greater than the median

```
median(v, na.rm = TRUE)
```

The median is resistant to extreme values while the mean is not



Example:

Mean US salary = $72,641

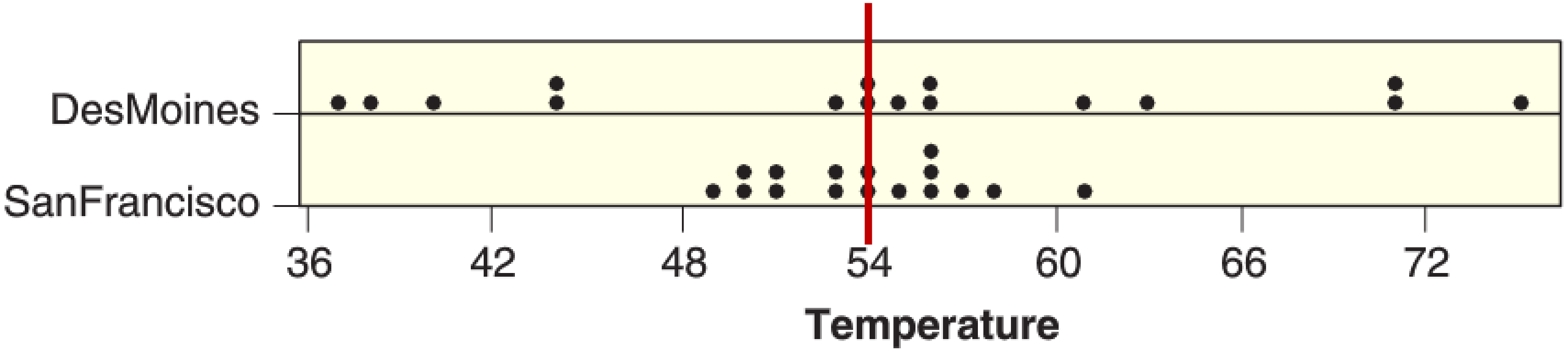
Median US salary = $51,939

Measures of spread

Measure of spread 1: standard deviation

The **standard deviation** is a statistic that quantifies how far the data is spread

Data measured on April 14th from 1997 to 2010:



Mean temperature (°F): Des Moines = 54.49 San Francisco = 54.01

Standard deviation (°F): Des Moines = 11.73 San Francisco = 3.38

Example: computing the standard deviation

Suppose we had a sample with n = 4 points:

$x_1 = 8$, $x_2 = 2$, $x_3 = 6$, $x_4 = 4$,

We can compute the mean using the formula:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad = \tfrac{1}{4}\cdot(x_1 + x_2 + x_3 + x_4) \quad = \tfrac{1}{4}\cdot(8+2+6+4) \ = 5$$

The standard deviation can be computed using the formula:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad s = \sqrt{\frac{1}{4-1}\sum_{i=1}^{n}(x_i - 5)^2}$$

(remember order of operations!)

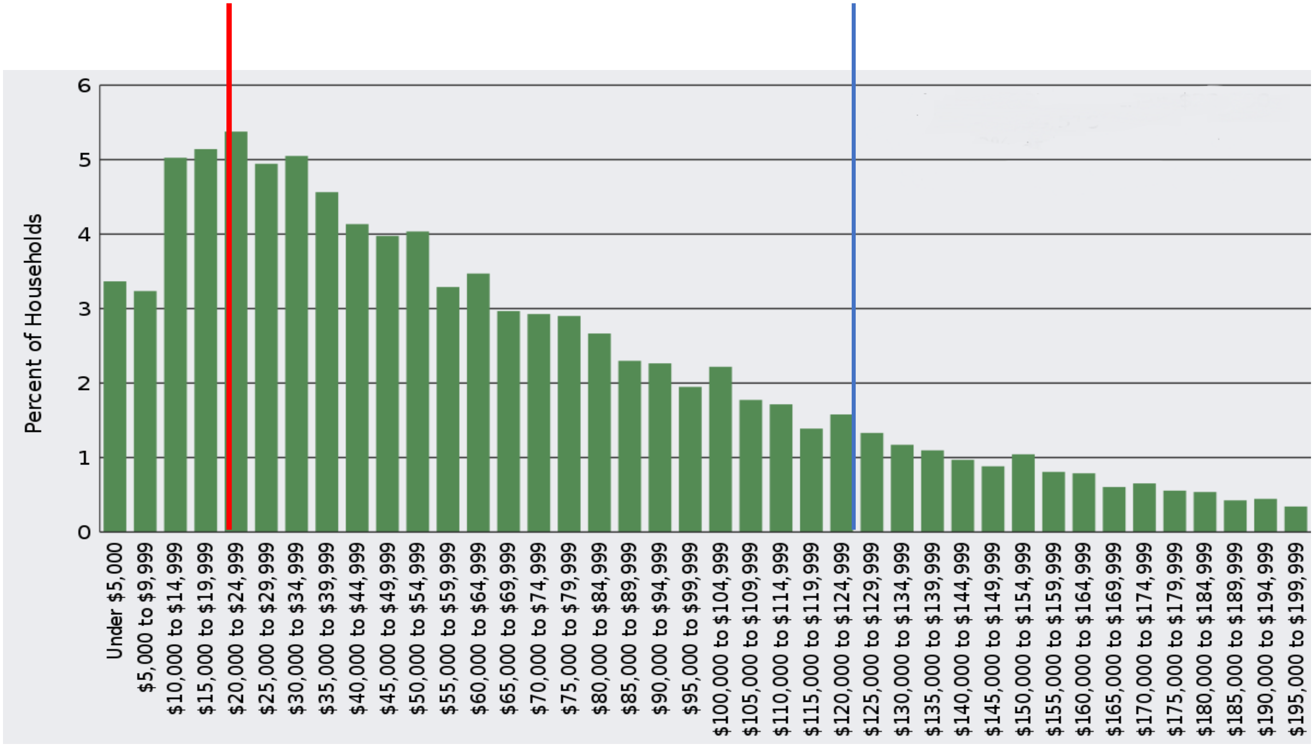
Percentiles

The **Pth percentile** is the value of a quantitative variable which is greater than P percent of the data

20th percentile = \$21,430

80th percentile = \$112, 254

For the US income distribution what are the 20th and 80th percentiles?



R: `quantile(v, .95)`

Five Number Summary

A **five-number summary** is a set of five descriptive statistics that provides a concise overview of a dataset's distribution

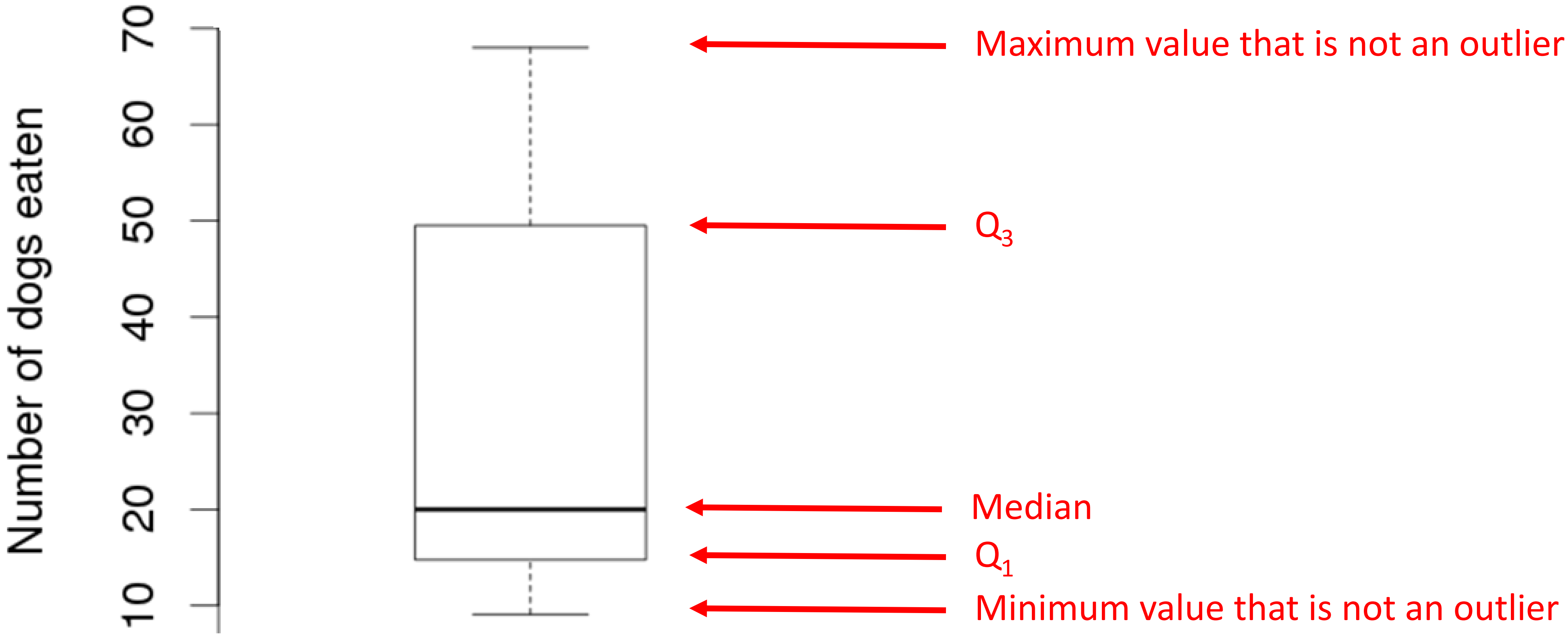
Five Number Summary = (minimum, Q_1, median, Q_3, maximum)

Q_1 = 25th percentile (also called 1st quartile)
Q_3 = 75th percentile (also called 3rd quartile)

Roughly divides the data into fourths

Measure of spread 2: Interquartile range (IQR) = $Q_3 - Q_1$

Box plots can also visualize quantitative data



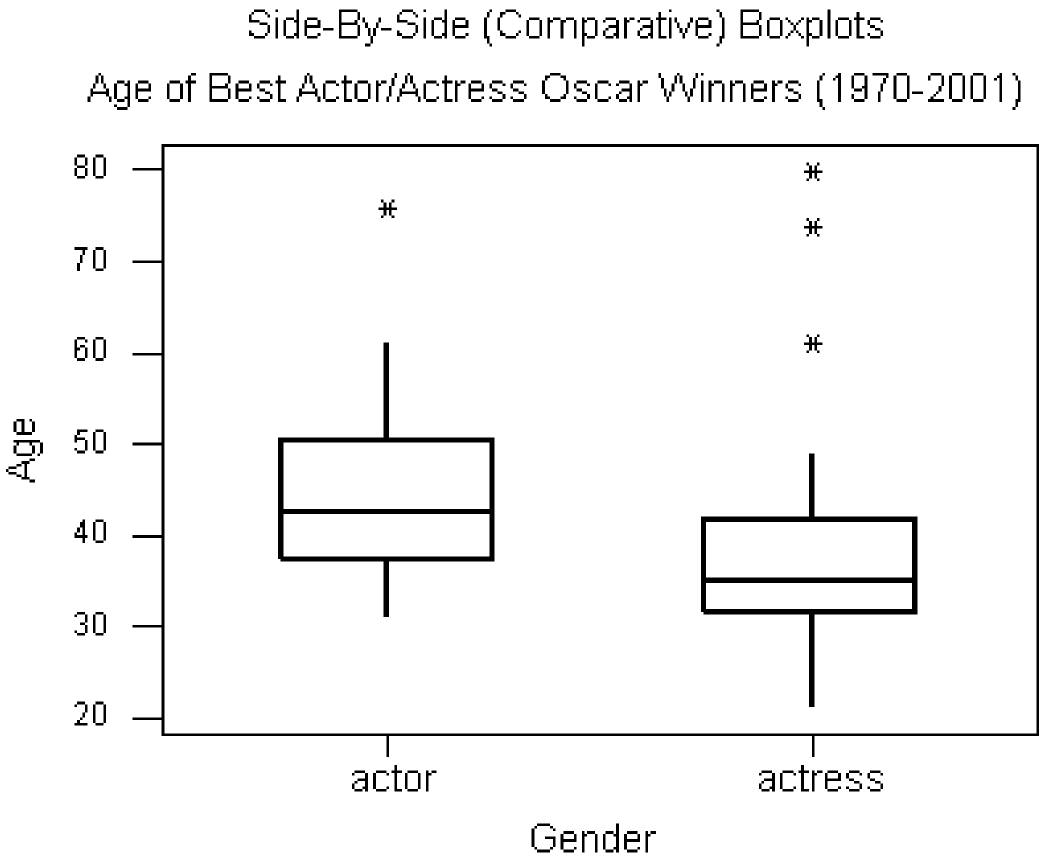
R: `boxplot(v)`

Side-by-side boxplots

Boxplots are particularly useful for comparing distributions!

Let’s look at the ages that people won the best actor/actress Oscar

What does this figure tell us?



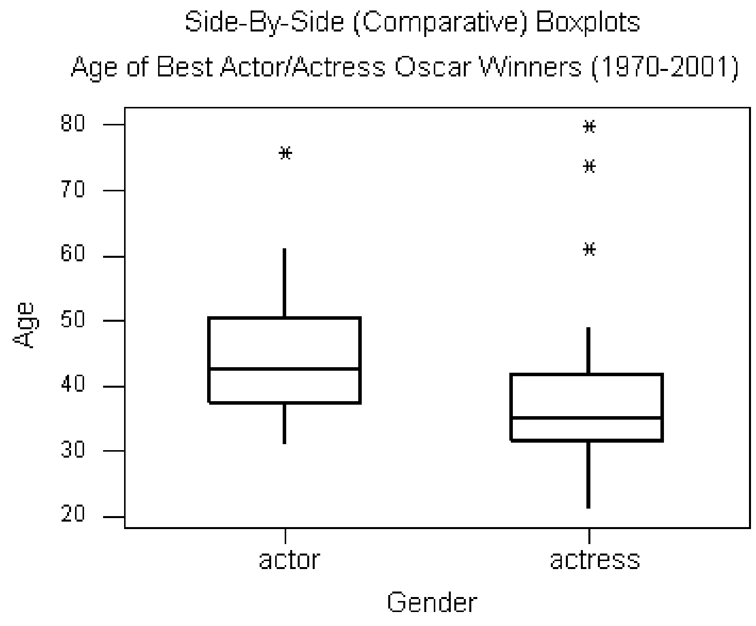
Outliers

Outliers on boxplots are values that are more than 1.5 * IQR

What should we do if we have outliers?

Investigate:

- If there are due to an error, remove them
- If not, need to account for them



Questions?

Let’s try it in RStudio!

Visualizing two quantitative variables

CitiBike data

Let’s look at the bike share data from NYC

> load('daily_bike_totals.rda’)

CitiBike analysis

	date	trips	precipitation	snow_depth	snowfall	max_temperature	min_temperature
1	2013-07-01	16650	0.8385830	0	0	77.00	71.96
2	2013-07-02	22745	0.0787402	0	0	82.04	71.96
3	2013-07-03	21864	0.5314960	0	0	82.94	73.04
4	2013-07-04	22326	0.0000000	0	0	87.08	75.02
5	2013-07-05	21842	0.0000000	0	0	89.96	75.92
6	2013-07-06	20467	0.0000000	0	0	91.94	78.08
7	2013-07-07	20477	0.0000000	0	0	91.94	78.08
8	2013-07-08	21615	0.2204720	0	0	89.06	73.04

What does each case correspond to?

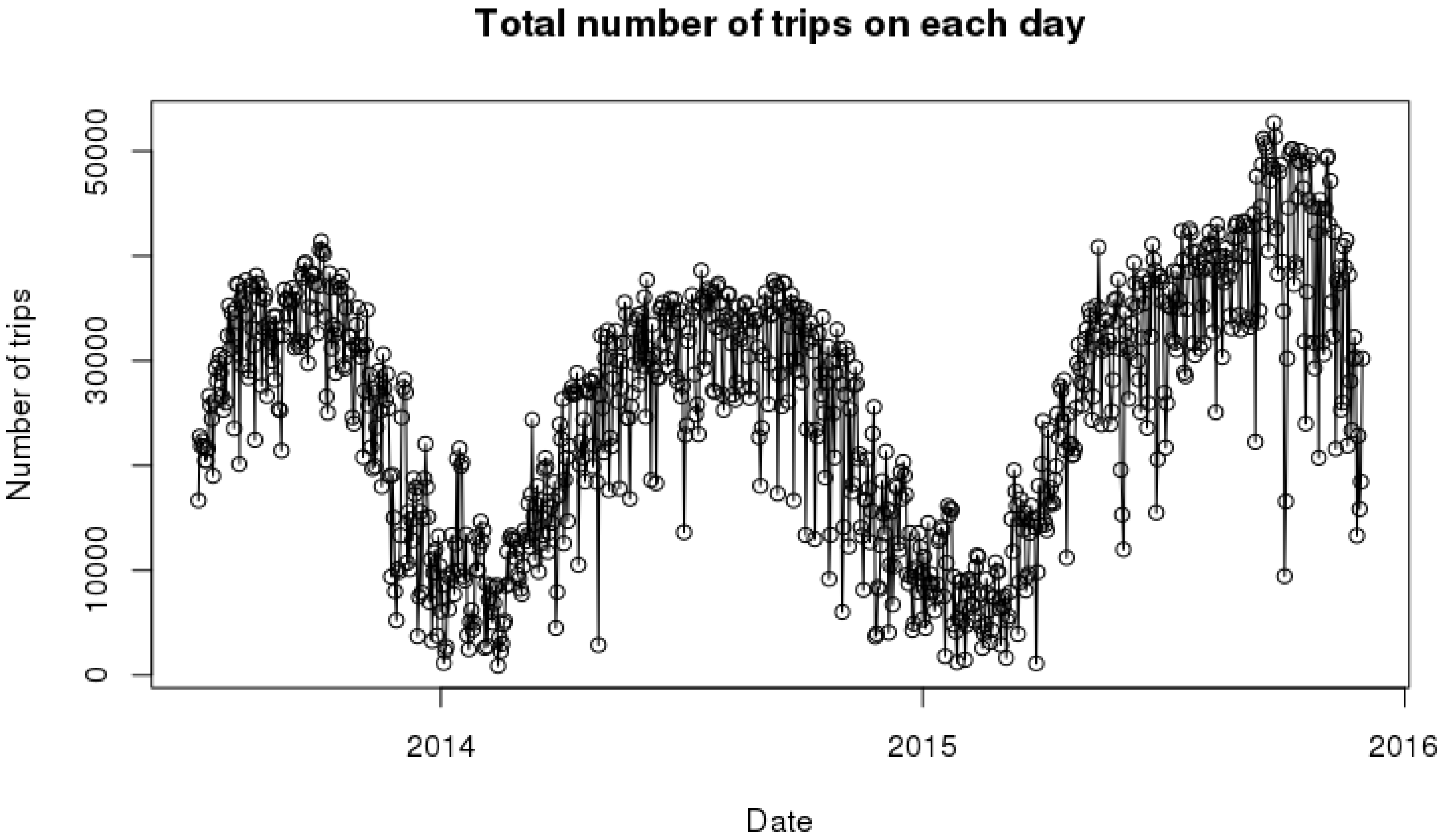
Line plots

We can use the plot(x, y) function to create line plots

```
# we can connect the points in a plot using
  plot(x, y, type = 'l')    # line graph
  plot(x, y, type = 'o')   # both points and a line

 plot(bike_daily_data$date,  bike_daily_data$trips,
            type = 'o',
            xlab = "Date",
            ylab = "Number of trips",
            main = "Total number of trips on each day")
```

Line plots



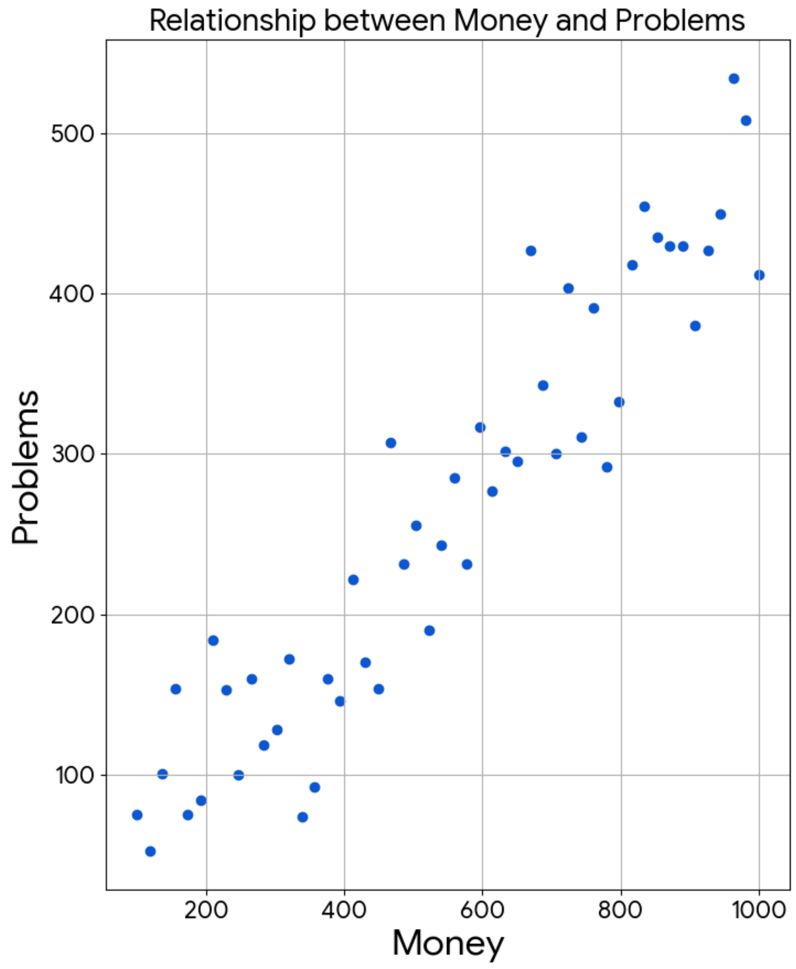
Total number of trips on each day
Number of trips
0
10000
30000
50000
2014
2015
2016
Date

Scatterplot

A **scatterplot** graphs the relationship between two variables

- Each axis represents the value of one variables
- Each point the plot shows the value for the two variables for a single data case

If there is an explanatory and response variable, then the explanatory variable is put on the x-axis and the response variable is put on the y-axis



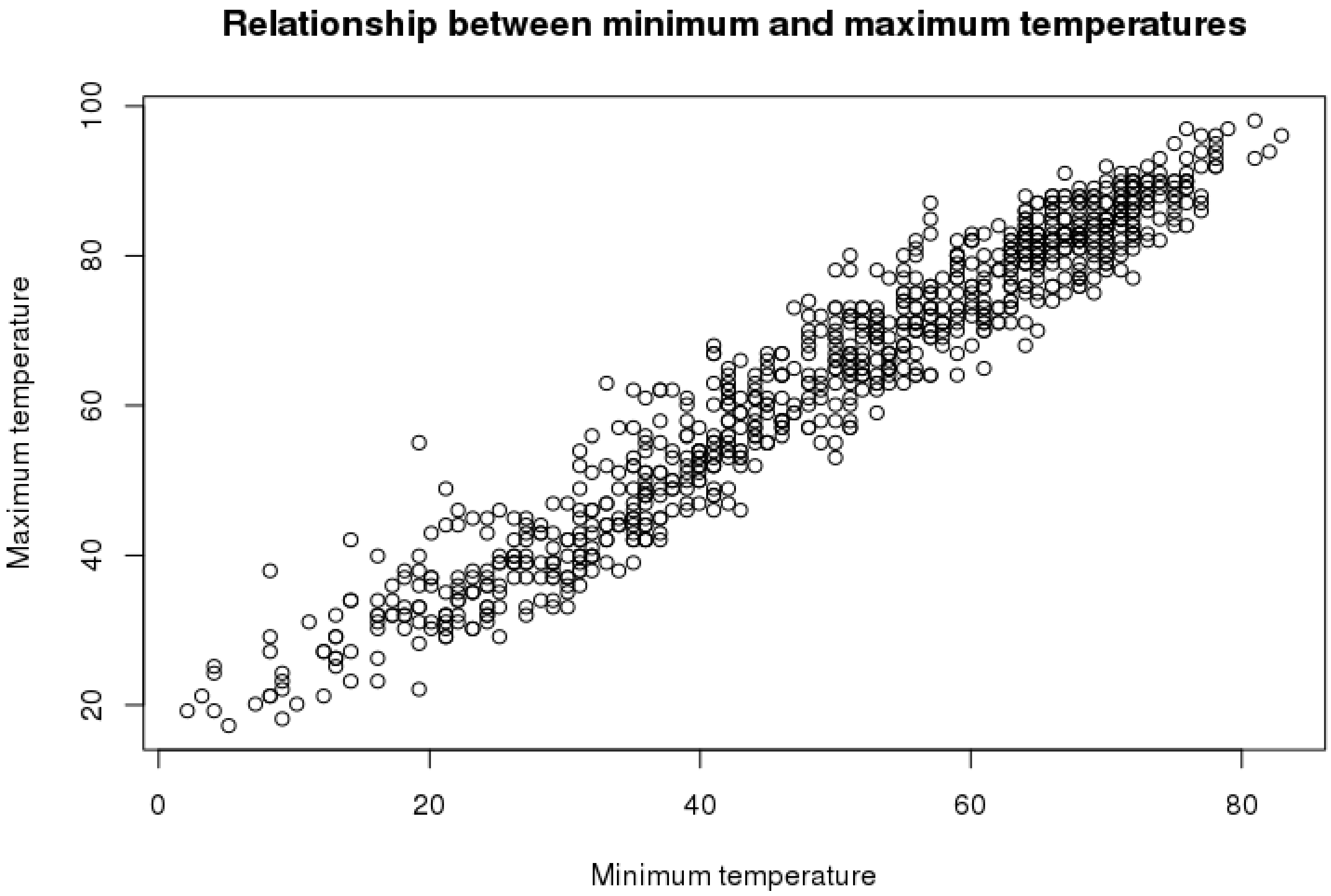
Scatter plots

We can use the plot(x, y) function to create scatter plots

Can you create a scatter plot of the relationship between the minimum and maximum temperatures?

```
plot(bike_daily_data$min_temperature,
     bike_daily_data$max_temperature,
     xlab = "Minimum temperature",
     ylab = "Maximum temperature",
     main = "Relationship between min and temp")
```

Scatter plots



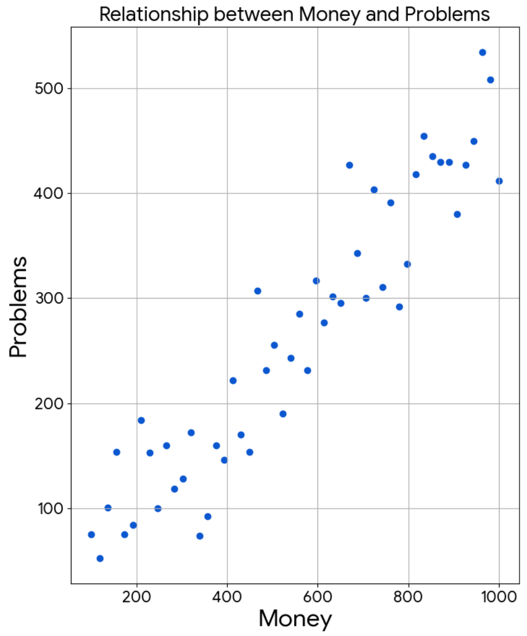
The correlation coefficient

The **correlation** is measure of the strength and direction of a linear association between two variables

$$r = \frac{1}{(n-1)} \sum_{i=1}^{n} \left(\frac{x_i - \overline{x}}{s_x} \right) \left(\frac{y_i - \overline{y}}{s_y} \right)$$

R: `cor(x, y)`

Properties of the correlation



Correlation as always between -1 and 1: $-1 \leq r \leq 1$

The sign of r indicates the direction of the association

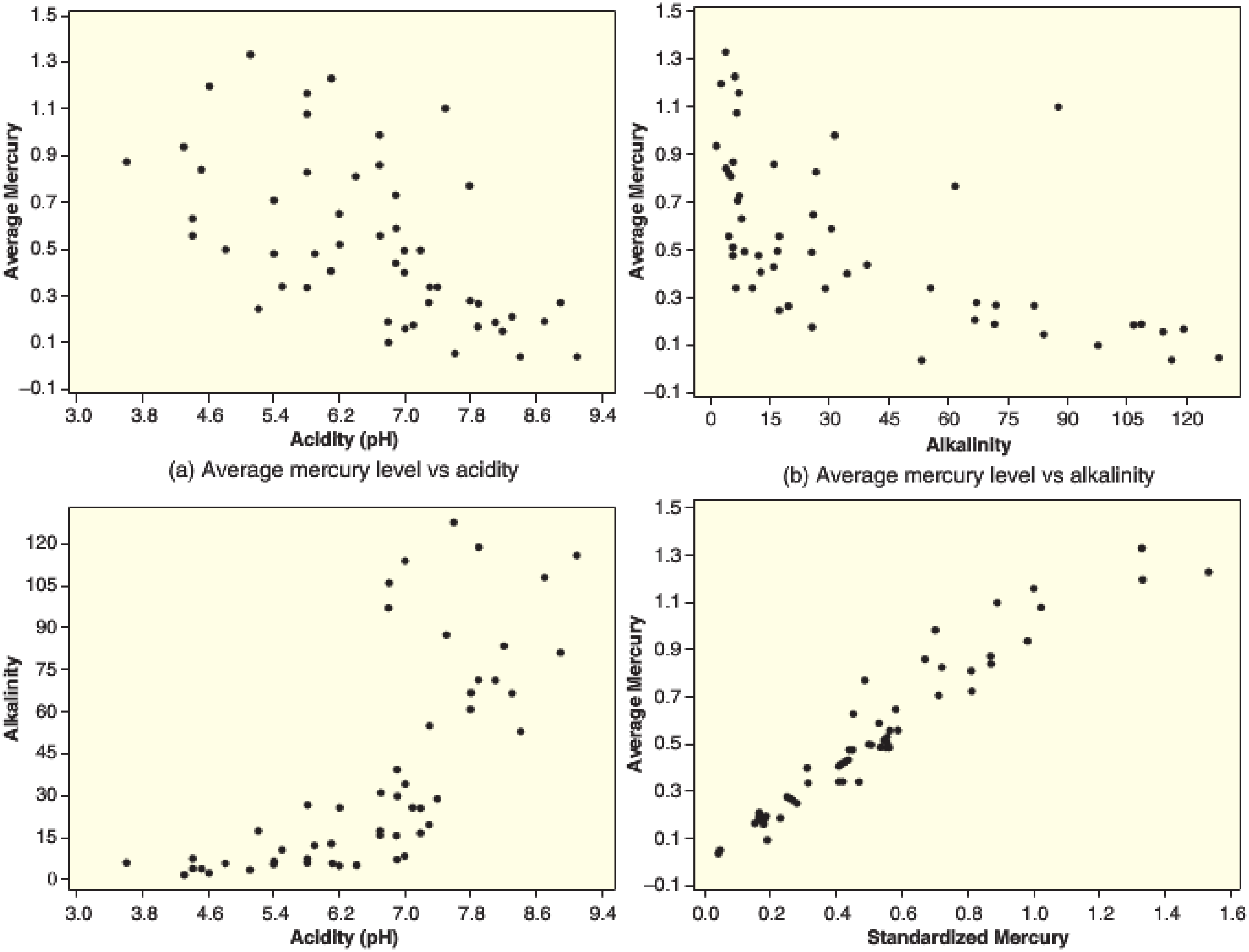
Values close to ± 1 show strong linear relationships, values close to 0 show no linear relationship

Correlation is symmetric: r = cor(x, y) = cor(y, x)

$$r = \frac{1}{(n-1)} \sum_{i=1}^{n} \left(\frac{x_i - \overline{x}}{s_x}\right)\left(\frac{y_i - \overline{y}}{s_y}\right)$$

Florida lakes

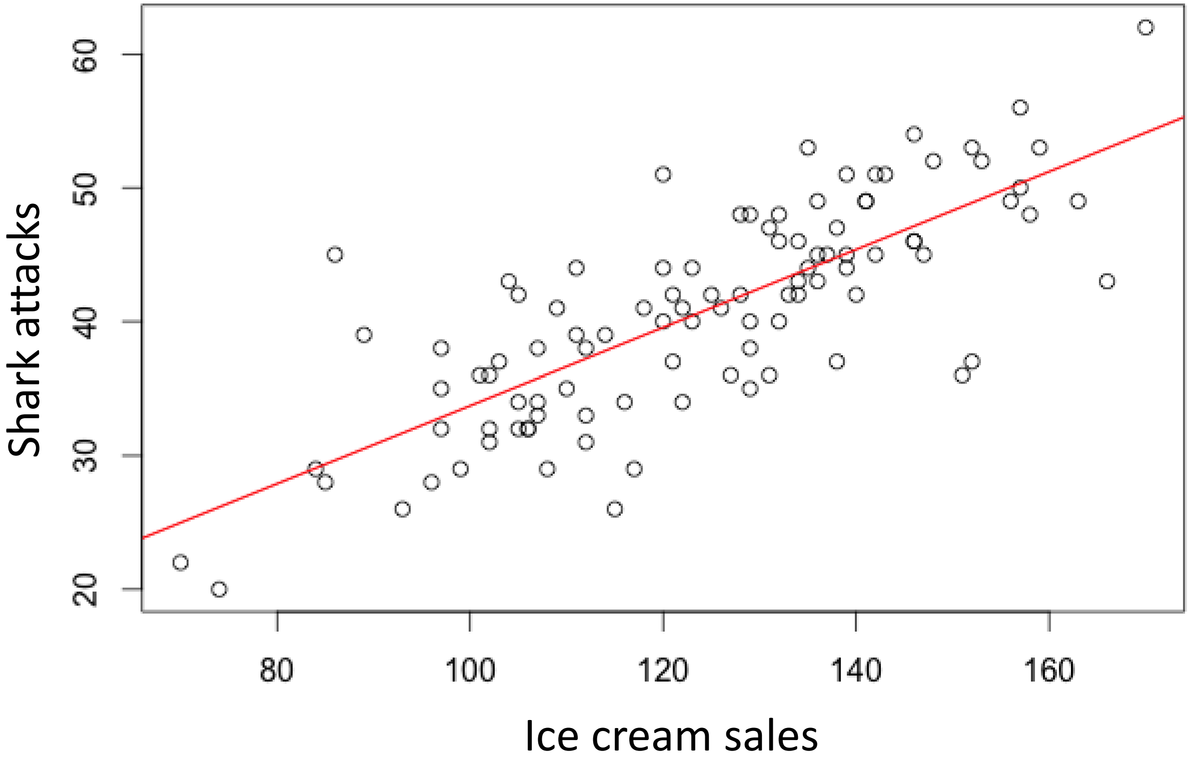
[Correlation game]()



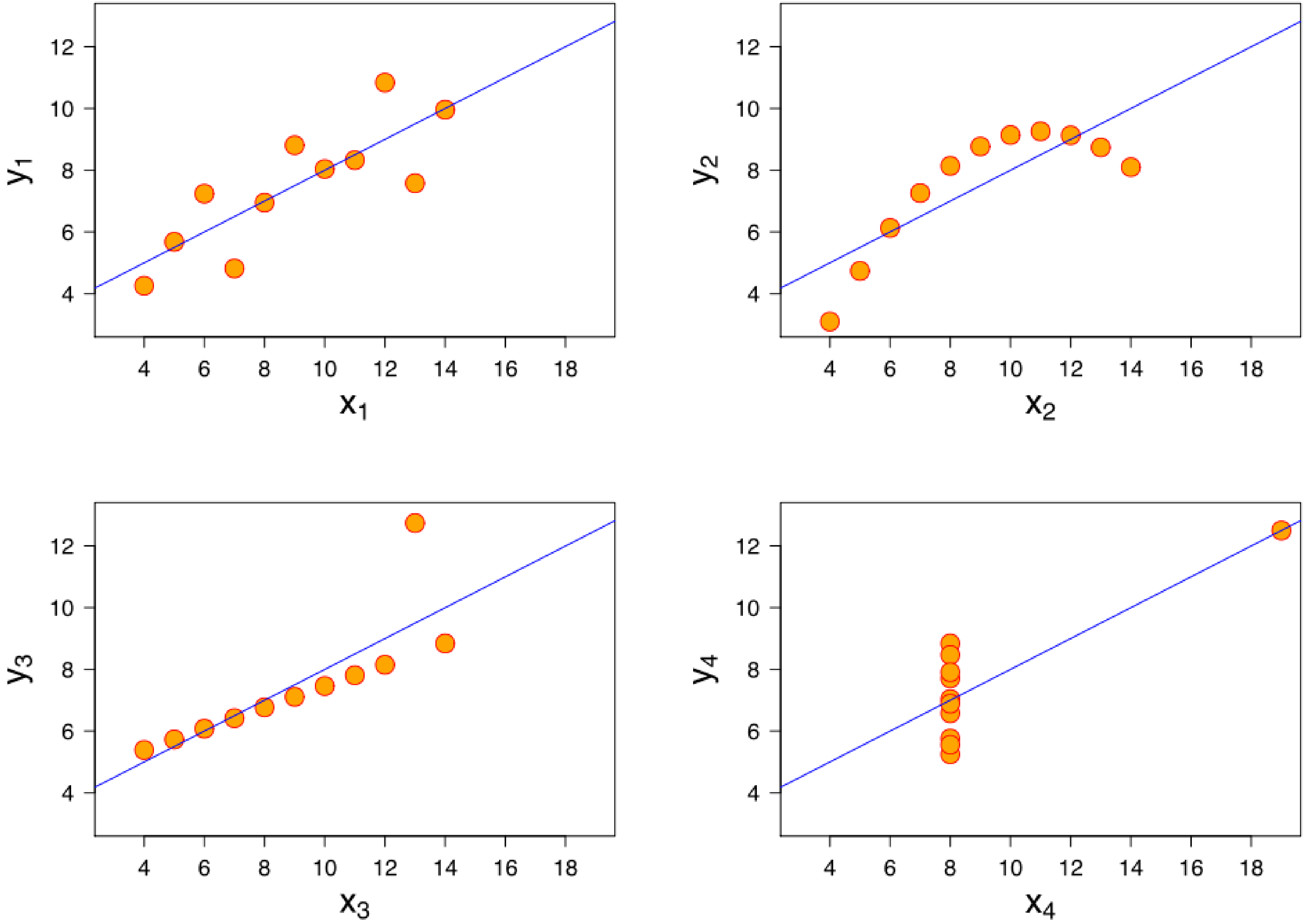
(a) Average mercury level vs acidity
(b) Average mercury level vs alkalinity
(c) Alkalinity vs acidity
(d) Average vs standardized mercury levels

Correlation caution #1

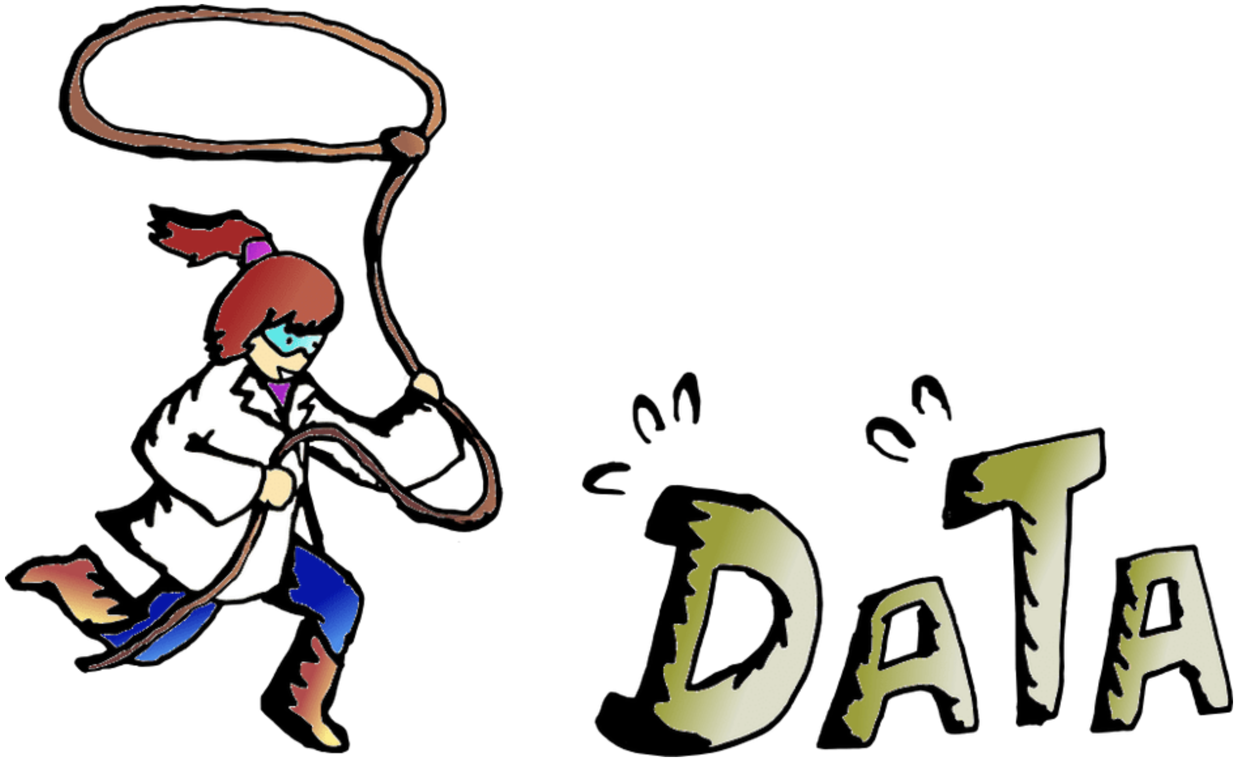
A strong positive or negative correlation does not (necessarily) imply a cause and effect relationship between two variables



Anscombe's quartet (r = 0.81)



Data wrangling/transformation using dplyr



The tidyverse and dplyr

The ‘tidyverse’

The tidyverse is set of R packages that operate ‘tidy data’

- i.e., that operate on data frames (or tibbles)

Tidy data is data where:

- Each variable must have its own column
- Each observation must have its own row
- Each value must have its own cell

variables observations values

Messy data…

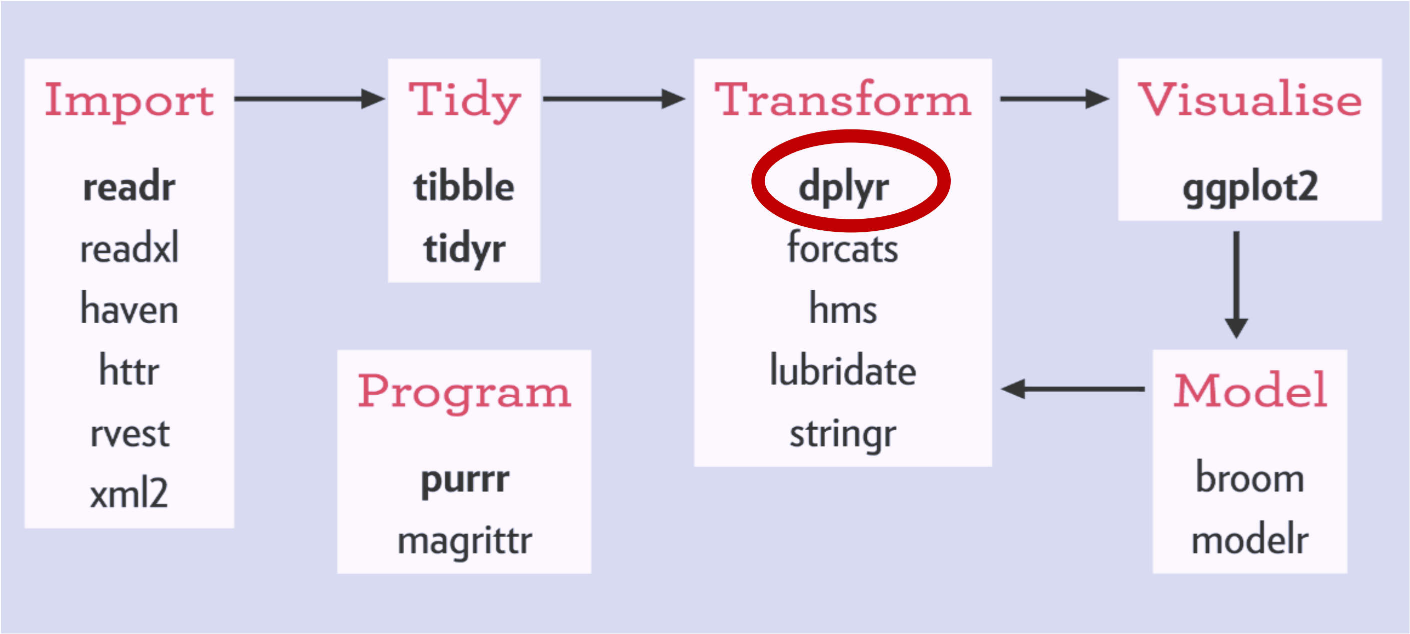
What would be an example of data that is not tidy?

Curve information - Curve quality data															
Name	Formula	Slope at	Intercept	ED-20	ED-50	ED-80	Correlati	Forced through origo							
Standard	Calc 1: C	standard	standard	3792394	27752	0.2	0.5	0.8	1	No					
Plate information															
Plate	Repeat	Barcode	Measured	Chamber	Chamber	Humidity	Humidity	Ambient	Ambient	Formula	Measurement date				
1	1		N/A	N/A	N/A	N/A	N/A	N/A	N/A	Calc 1: C	standard	standard	10.12.2013 10:23:33		
Background information															
Plate	Label	Result	Signal	Flashes/T	Meastime	MeasInfo									
1	PicoGree	0	110307	10	0	De=1st Ex=Top Em=Top Wdw=N/A									
Calculate	standard	standards on each plate) where Label: PicoGreenFilterTop(1) channel 1													
	1	2	3	4	5	6	7	8	9	10	11	12			
A	-0.0011	-0.0011	-0.001	-0.001	-0.0011	-0.0012	-0.0011	-0.0011	-0.0012	-0.0012	0.9973	1.0026			
B	0.0012	0.0014	0.0013	0.0012	0.0013	0.0012	0.0014	0.0003	-0.0011	-0.0011	0.0981	0.103			
C	0.0016	0.0013	0.0013	0.0011	0.0012	0.0015	0.0016	-0.0004	-0.0011	-0.0011	0.0104	0.0095			
D	0.0019	0.0024	0.0018	0.0015	-0.001	-0.001	-0.001	-0.001	-0.0011	-0.0011	0.0008	0.0009			
E	-0.001	-0.0011	-0.0011	-0.0011	-0.001	-0.0012	-0.0011	-0.001	-0.0009	-0.0011	-0.0001	-0.0002			
F	-0.001	-0.0011	-0.001	-0.001	-0.0012	-0.0011	-0.0011	-0.0009	-0.001	-0.001	-0.0003	-0.0002			
G	-0.0011	-0.0011	-0.0011	-0.001	-0.001	-0.0012	-0.0011	-0.001	-0.001	-0.0011	-0.0002	0.0012			
H	-0.0011	-0.0012	-0.0011	-0.001	-0.0011	-0.0011	-0.0012	-0.0011	-0.0011	-0.001	-0.0003	-0.0003			

The ‘tidyverse’

The tidyverse is a set of packages share a common design philosophy

- Most written by Hadley Wickham



dplyr: A grammar for data wrangling

Grammar: a set of components that can be combined to achieve a goal

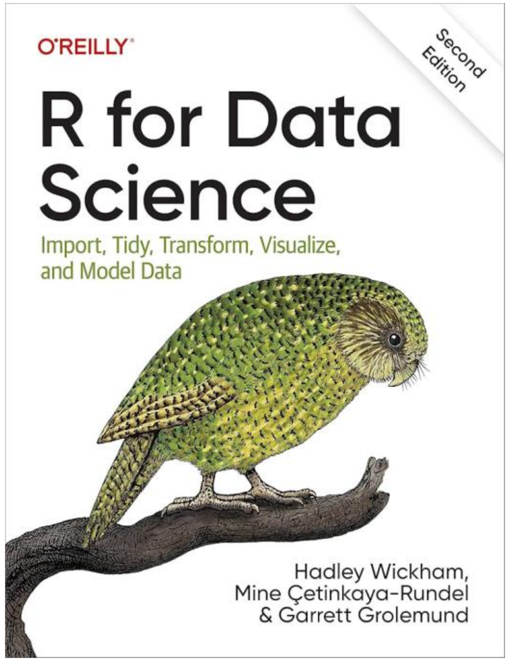
dplyr is a package that has a set of verbs that are useful for transformations data:

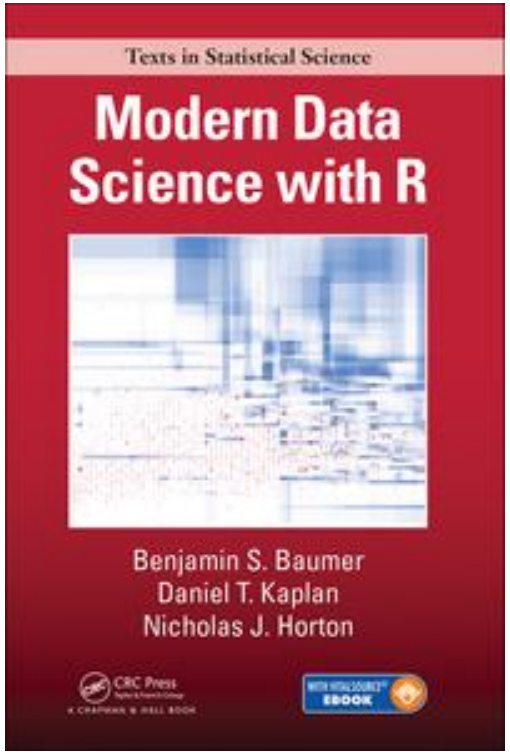
1. filter()
2. select()
3. mutate()
4. arrange()
5. group_by()
6. summarize()

All these function **take a data frame** and other arguments and **return a data frame**

```
> library(dplyr)   # load the dplyr package
```

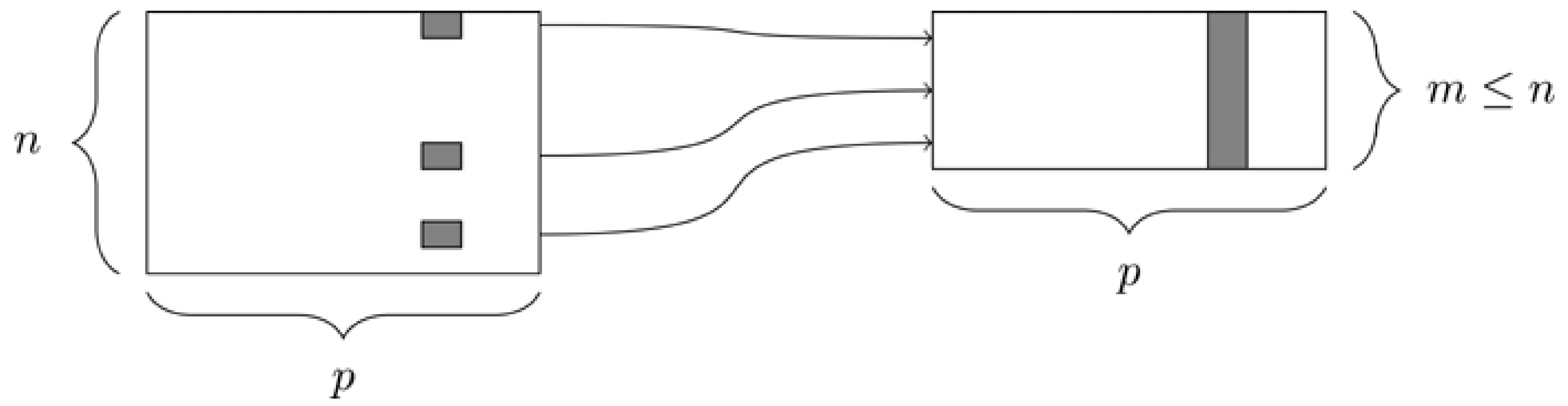
Quick overview of the dplyr functions





1. filter()

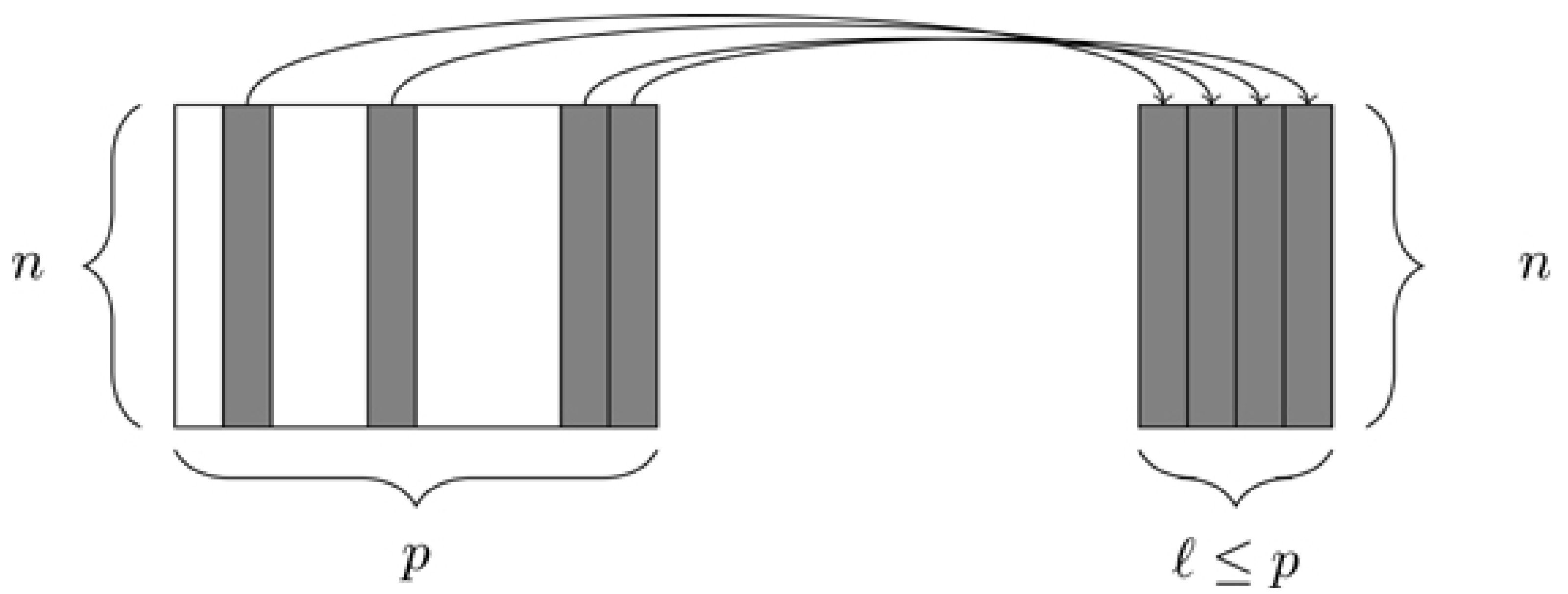
The filter() function allows you to select a subset of rows in data frame



filter(profiles, height == 77)

2. select()

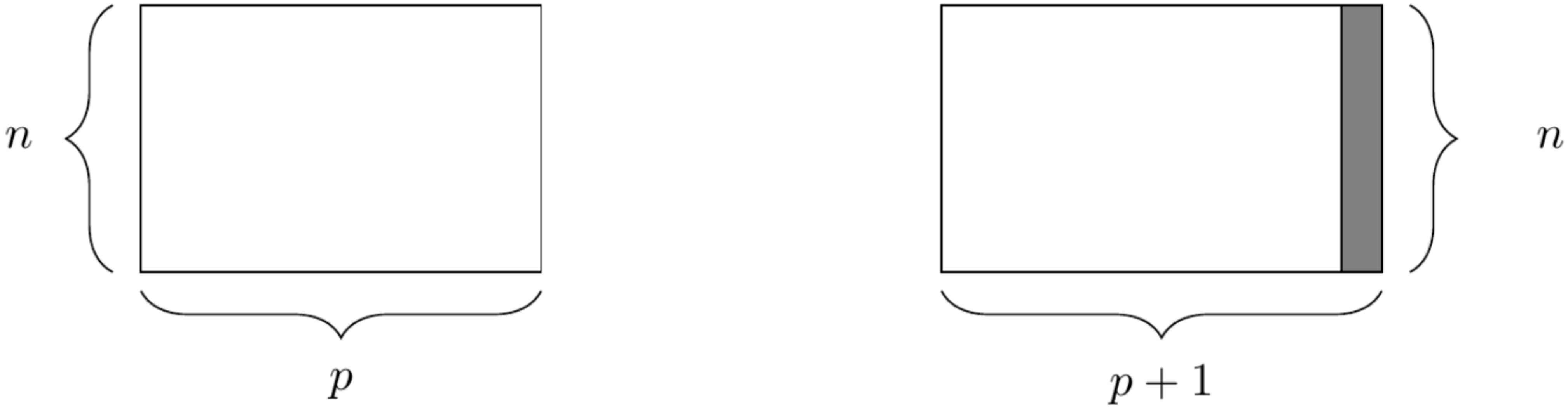
The select() function allows you to select a subset of columns



select(profiles, age, height)

3. mutate()

The mutate() function allows you to create new columns that are functions of existing columns

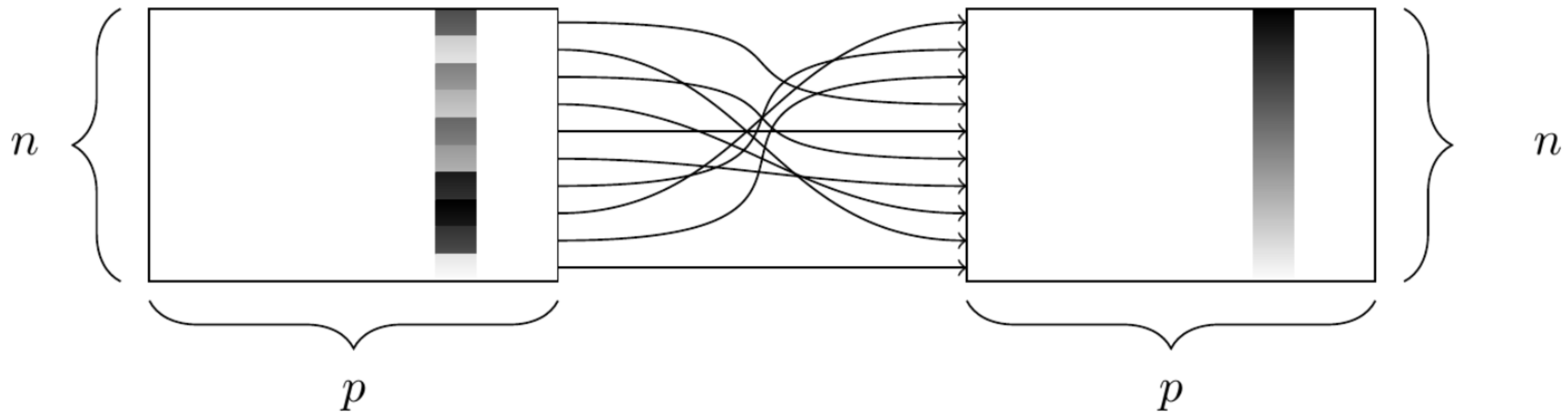


mutate(profiles, height_feet = height/12)

4. arrange()

The arrange() function arranges the rows based values in a column

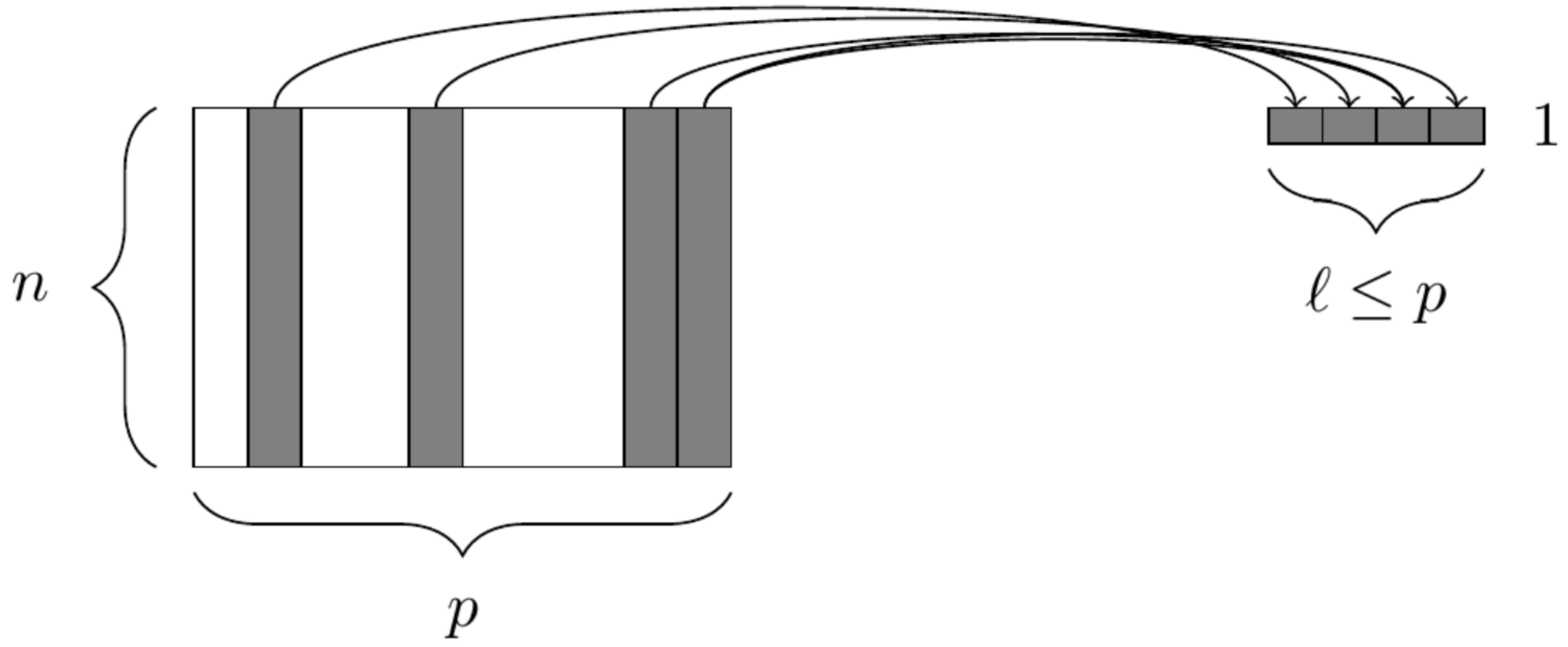
- arrange(desc()) arranges from largest to smallest



arrange(profiles, desc(height))

5. summarize()

The summarize() function reduces values in many rows into single values

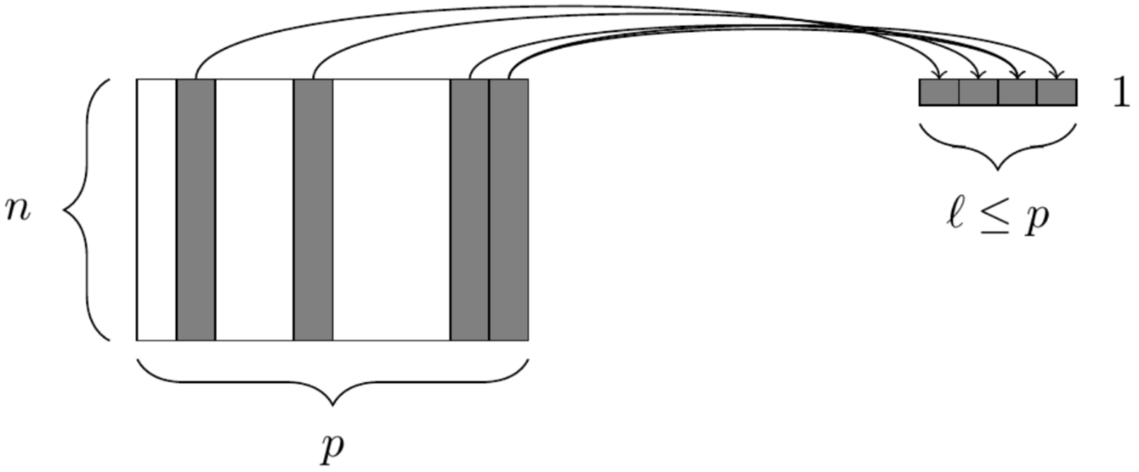


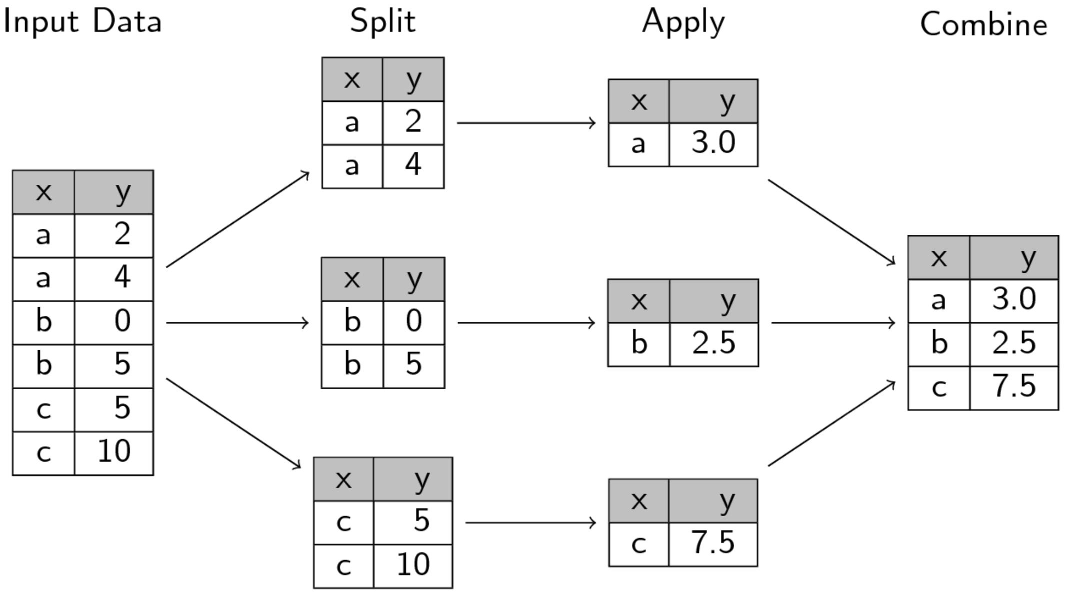
summarize(mean_age = mean(age))

6. The group_by() function

The group_by() function groups variables for future operations

- It works in conjunction with summarize() and mutate() to do split, apply, combine

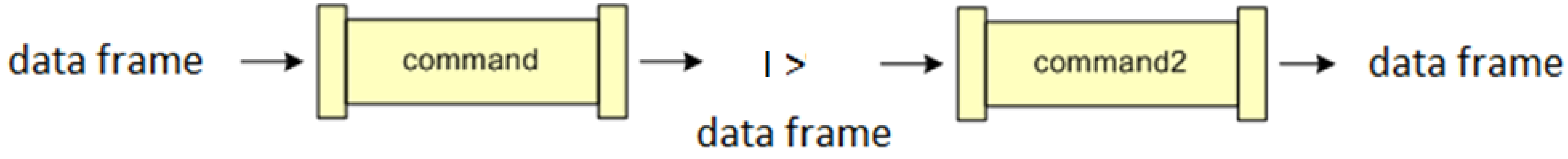




group_by(profiles, sex)

The pipe operator

The pipe operator |> allows us to chain commands together




```
profiles|>
        group_by(sex) |>
        summarize(mean_age = mean(age))
```

Let’s try it out!