Psychological Statistics Lab 2

PSYC 2020-A01 / PSYC 6022-A01 | 2025-08-29 | Descriptive Statistics I

Jessica Helmer

Outline

- Assignment 1 Review
- Central Tendency Review
- Central Tendency in R
- Variable Assignment
- R Functions

Learning objectives: Stats: Review measures of central tendency R: Variable assignment, functions, histogram, boxplot

Assignment 1 Review

Placeholder for common mistakes on Assignment 1

Review of Central Tendency!

Mean: Sum of all values divided by the total number of values

Median: When sorted lowest to highest, the middle value

Mode: The value that appears most often

Central Tendency Practice

Given this dataset:

```
1 c(0, 2, 2, 4)
```

```
[1] 0 2 2 4
```

What is the mean?

What is the median?

What is the mode?

Central Tendency Practice

Given this dataset:

```
1 c(0, 1, 2, 4)
```

```
[1] 0 1 2 4
```

What is the mean?

What is the median?

What is the mode?

R Functions

A *function* performs some operation on an *input* and produces some *output*

Saw this last week

```
1 head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

What is the function? Input? Output?

Central Tendency in R: Mean

We can calculate central tendencies in two ways:

Given this dataset, calculate the mean

```
1  c(2, 3, 12, 4, 4)
```

```
[1]  2  3 12  4  4
```

By hand (computer)

R Code | Start Over | Run Code

```
1  # let's calculate the mean!
```

With the `mean()` function

R Code | Start Over | Run Code

```
1  # let's calculate the mean!
```

Central Tendency in R: Median

Given this dataset, calculate the median

```
1 c(2, 3, 12, 4, 4)
```

```
[1]  2  3 12  4  4
```

By hand (computer)

R Code | Start Over | Run Code

```
1 # let's calculate the median!
```

With the `median()` function

R Code | Start Over | Run Code

```
1 # let's calculate the median!
```

Central Tendency in R: Mode

Given this dataset, calculate the mode

```
1 c(2, 3, 12, 4, 4)
```

```
[1]  2  3 12  4  4
```

With the `mode()` function

R Code | Start Over | Run Code

```
1 # let's calculate the mode!
```

Doesn’t work :(

Have to create our own

R Functions

We've seen some built-in R functions (e.g., `mean()`, `median()`), but we can also make our own

```
function_name <- function(argument) {
  do some stuff
  return(this stuff)
}
```

ⓘ Don't actually need to call `return()`; R will automatically return the last expression

Then, you can call the function

```
function_name(specific_argument)
```

To keep the results, make sure to assign them to some variable

```
very_important_results <- function_name(specific_argument)
```

R Functions

R Code | Start Over | Run Code

```
1  # write a function that takes in two numbers, adds t
```

R Code | Start Over | Run Code

```
1  # write a function that takes in two vectors, puts t
```

Let’s go back to finding the mode

Central Tendency in R: Mode

Given this dataset, calculate the mode

```r
c(2, 3, 12, 4, 4)
```

```
[1]  2  3 12  4  4
```

```r
my_mode <- function(x) {
  values <- unique(x)
  counts <- tabulate(match(x, values))
  max_index <- which.max(counts)
  values[max_index]
}
```

How does this work?

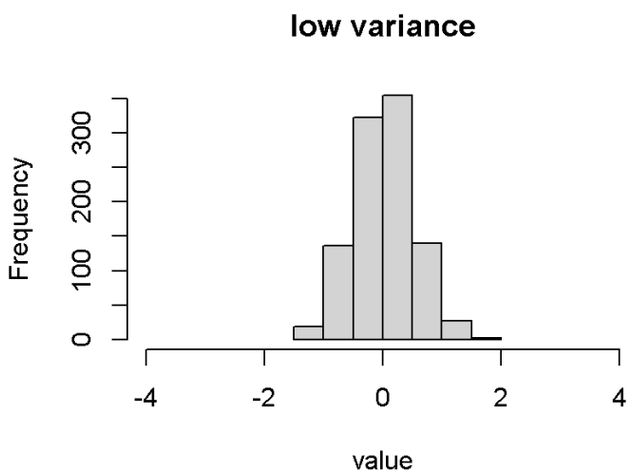
R Code | Start Over | Run Code

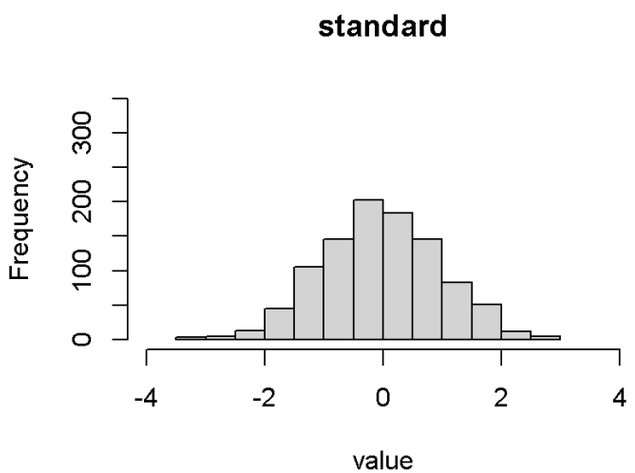
```r
# let's calculate the mode!
```

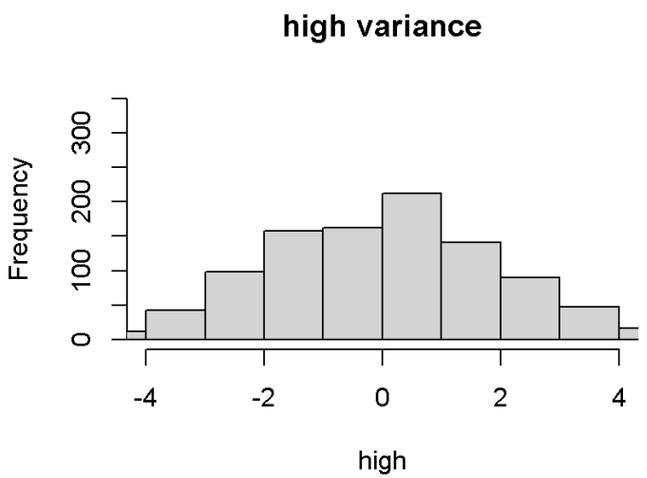
Measures of Variability Intro

Will go over in class in detail next week

Describe the “spread” or “dispersion” of the data







Measures of Variability Functions

```
1  testscores <- c(88, 93, 92, 99, 96)
```

Variance: `var()` function

Standard Deviation: `sd()` function

Interquartile Range: `IQR()` function

- Difference between the 3rd and 1st quantile (so 50% of the data within this range)
  - 25% of the data lower than the 1st quantile
  - 75% of the data lower than the 3rd quantile
  - IQR = 3rd - 1st

R Code | Start Over | Run Code

```
1  # let's look at some variances!
```

Quantiles: `quantile()` function

Descriptive Statistics in R

Takes time to look at all these for a lot of variables, even with functions

The `summary()` function provides us a quick overview of this information

R Code | Start Over | Run Code

```
1  # let's get descriptive statistics for the iris dataset
```

What all do we get?

- Minimum and maximum
- 1st quantile, median, 3rd quantile
- Mean

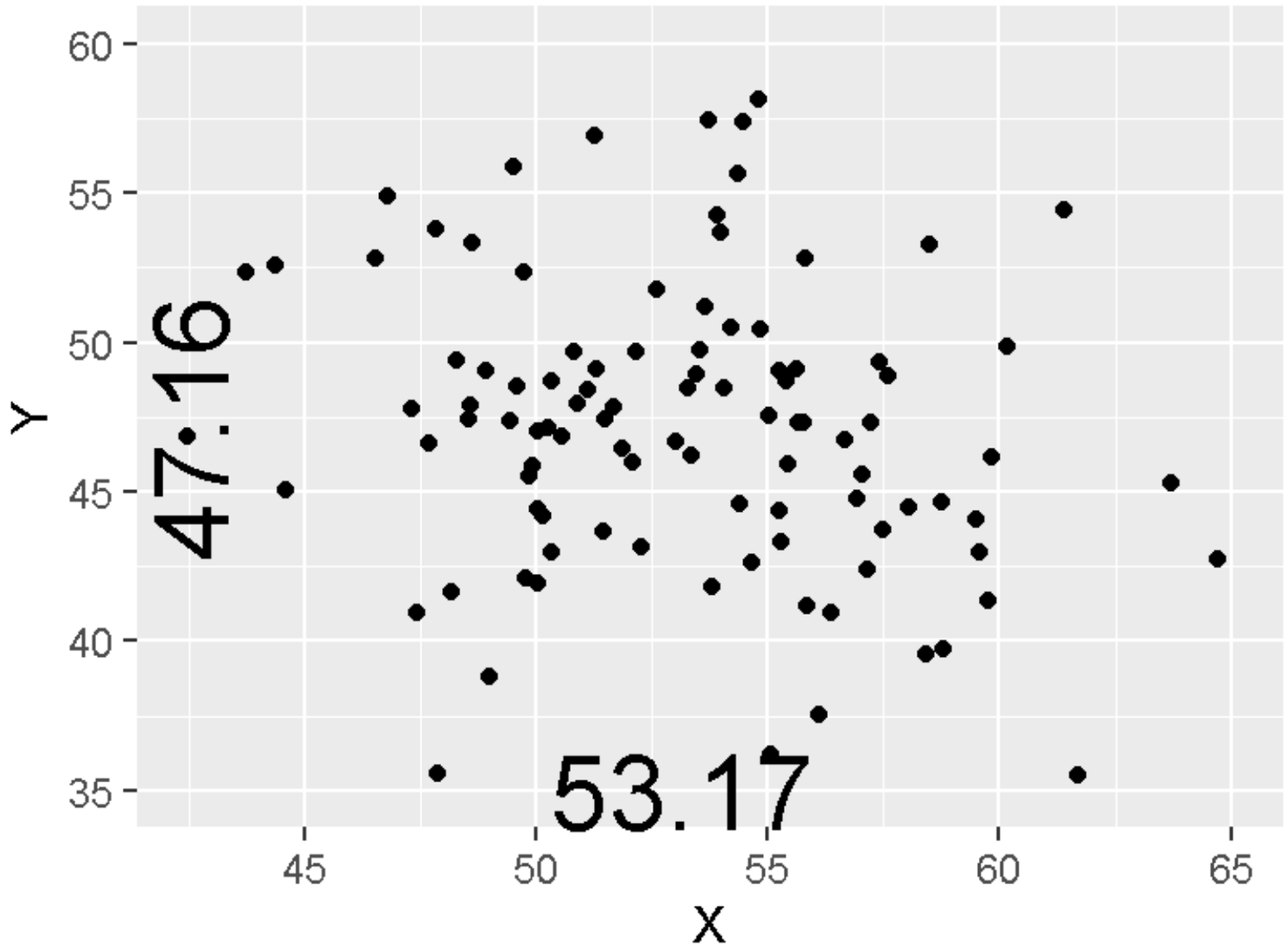
Visualizations!

Summary statistics are great, but don't trust them alone!

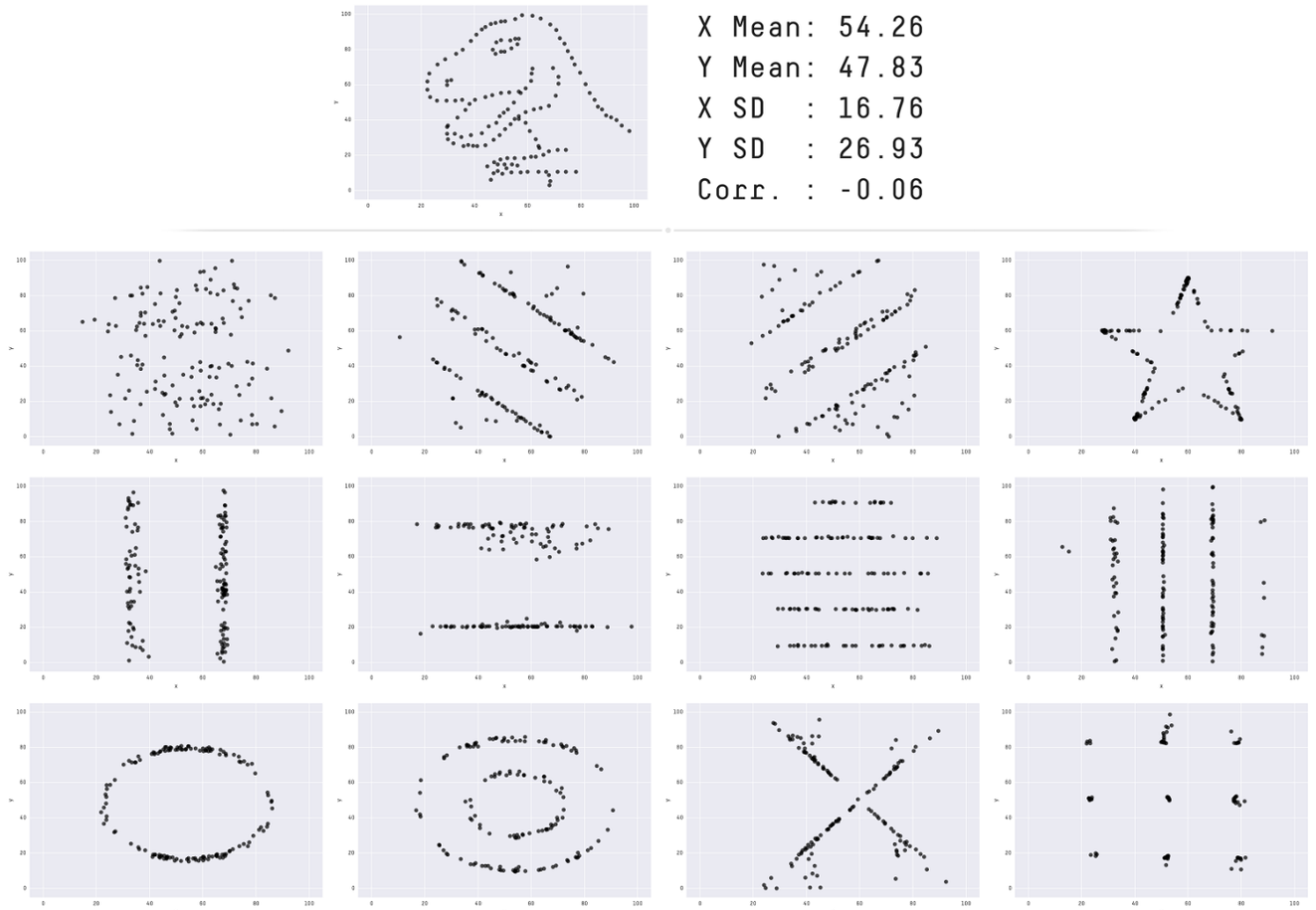
What do you think a dataset with these descriptives would look like?

```
X_mean <- 54.26
Y_mean <- 47.83

X_sd <- 16.76
Y_sd <- 26.93

cor <- -0.06
```

Visualizations!



Visualizations!



Datasaurus Dozen

Visualizations

Don’t rush: graph your data!

What should graphs do?

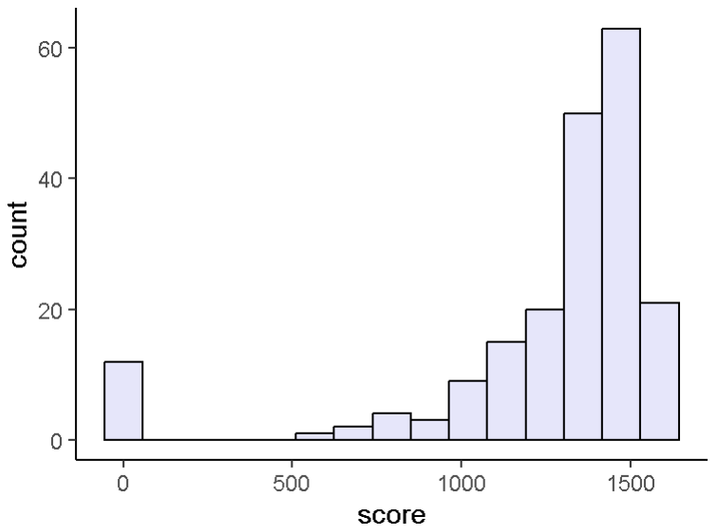
- Show the data
- Draw the reader primarily to the data (not the graphical effects)
- Avoid distorting the data
- Present many numbers with minimum ink
- Make large data sets coherent
- Encourage the reader to compare different pieces of data

Visualizations: Histograms

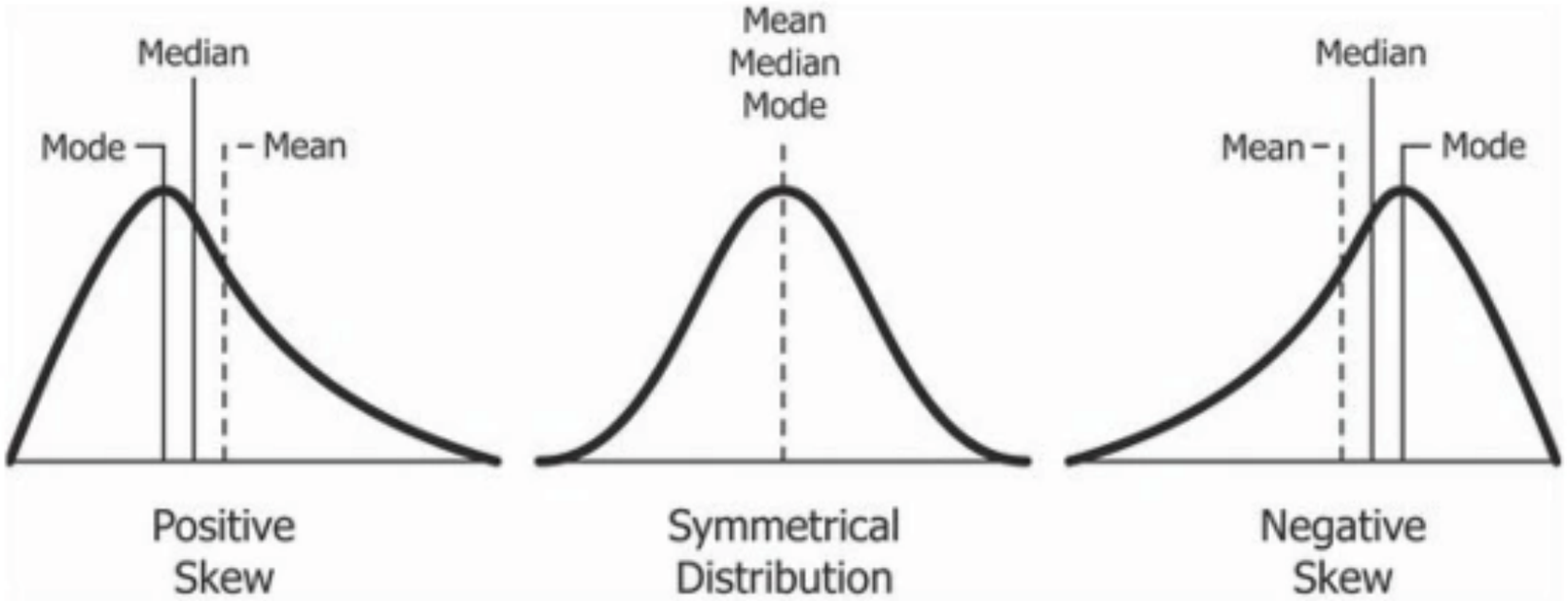
An example of (simulated) SAT scores

What do we see here?

- Outliers at zero! Not a possible SAT score
- Negatively skewed: more data on the left than on the right



Skew



Positive Skew, right-tailed

The mass of the distribution is concentrated on the left of the figure

Negative Skew, left-tailed

The mass of the distribution is concentrated on the right of the figure

Skewness Demonstration

Full screen version here

Skewness demonstration!

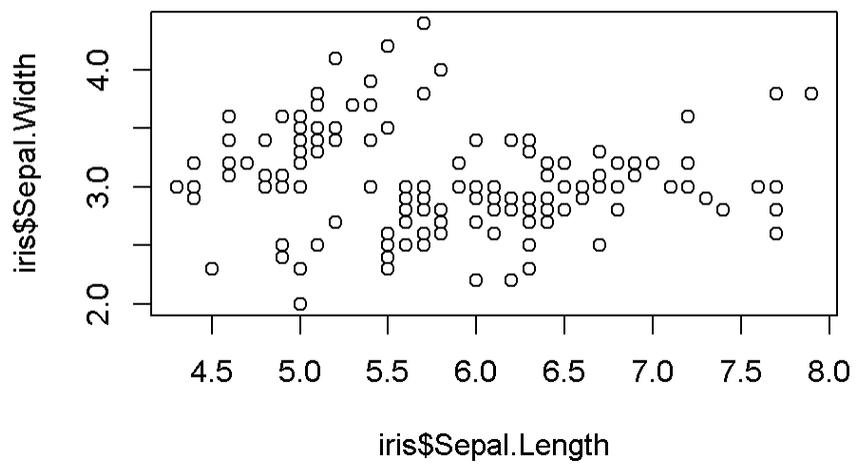
Credits to Fabio Setti

Let’s Do Some Visualization

Base R Graphics

R has some plotting features built in—we saw this last week

```
1  plot(iris$Sepal.Length, iris$Sepal.Width)
```

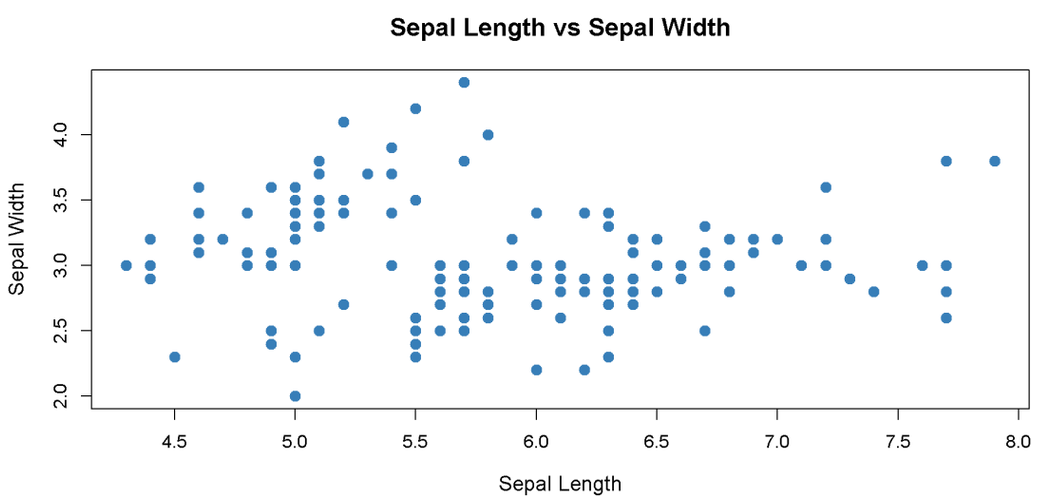


But we did not love this!

Base R Graphics

Better… (thanks, ChatGPT)

Plot | Code



We will learn a few plots in base R plotting, and then we will learn a *better* way of making plots:

`ggplot2`

R Graph Gallery

Let’s Do Some Visualization

Base R Graphics: Histogram

`hist()` function

Required arguments:

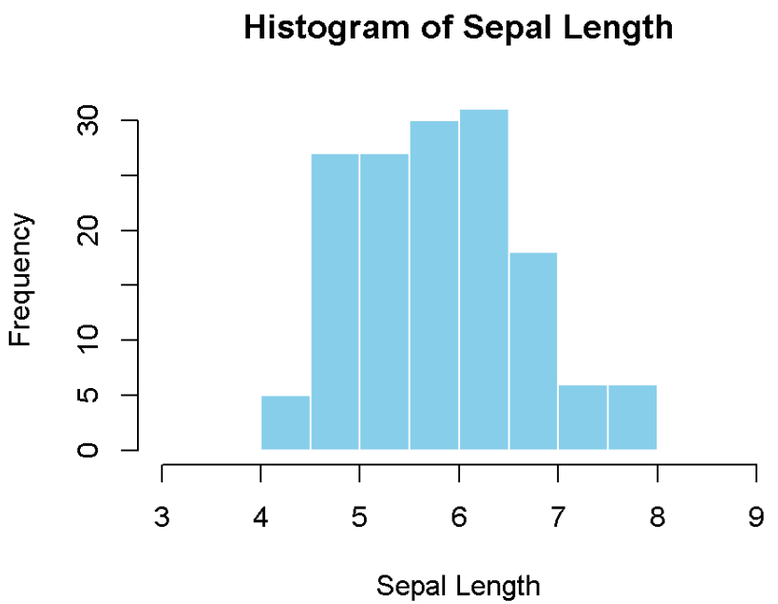
- `x` = vector (variable) you want to plot (remember the $ function!)

Optional arguments:

- `breaks`: number of bins
- `main`: title
- `xlab`: label for x-axis
- `ylab`: label for y-axis
- `col`: color for bars
- `xlim`: range for x-axis
- `ylim`: range for y-axis
- `prob`: `T`/`F`, y-axis proportion instead of frequency

If you do not set specific values for non-essential subarguments, it will use the default

Plot | Code

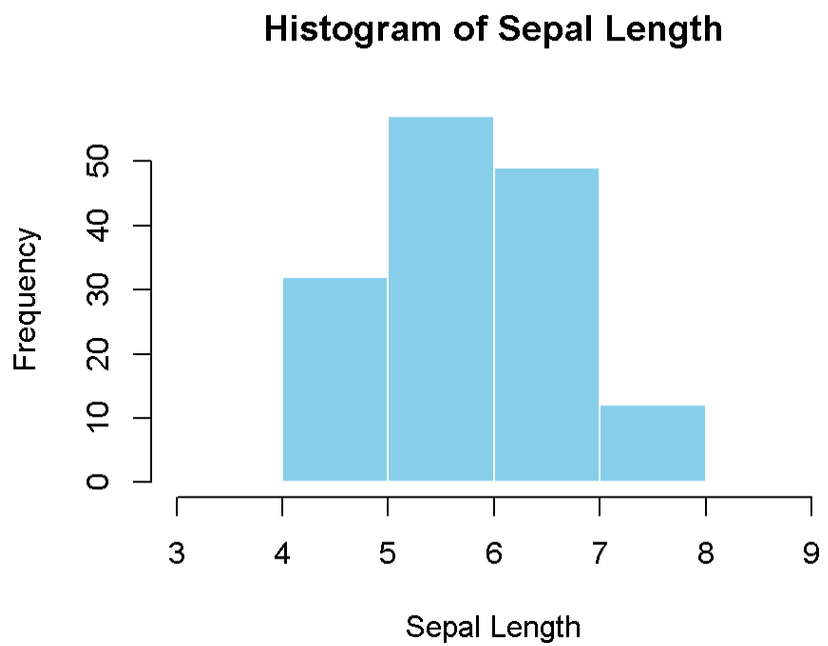


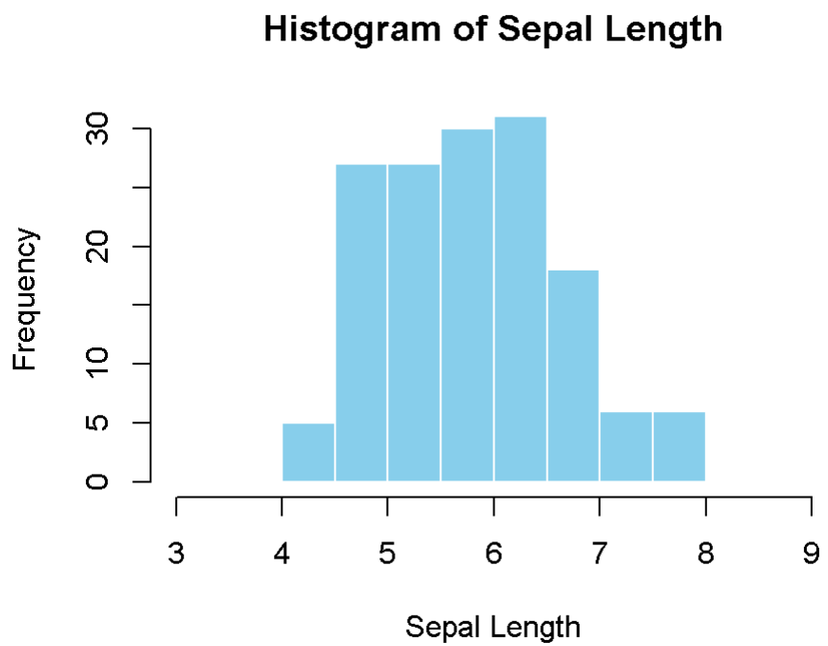
Let’s Do Some Visualization

Base R Graphics: Histogram

An important decision for histograms is this number (or width) of bins

Specified with the `breaks` argument



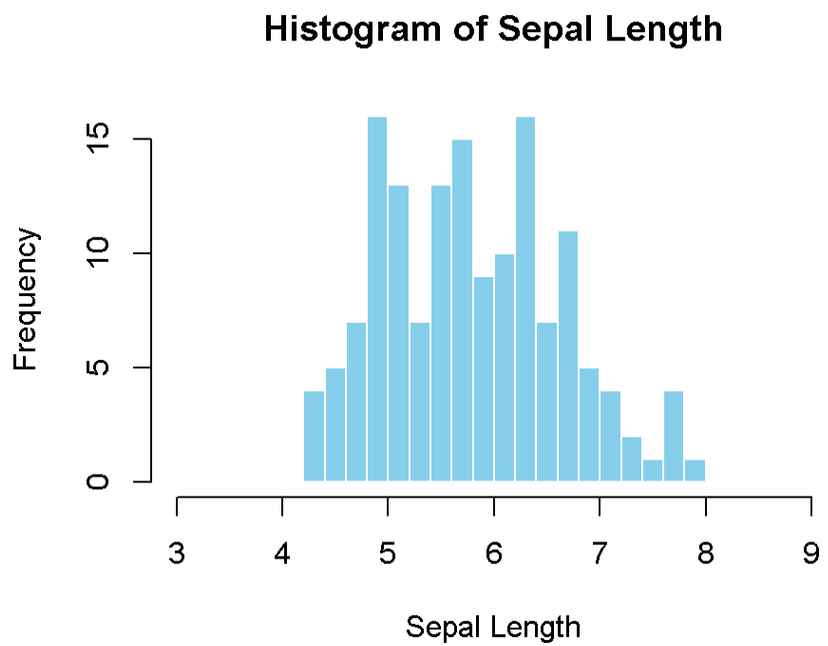


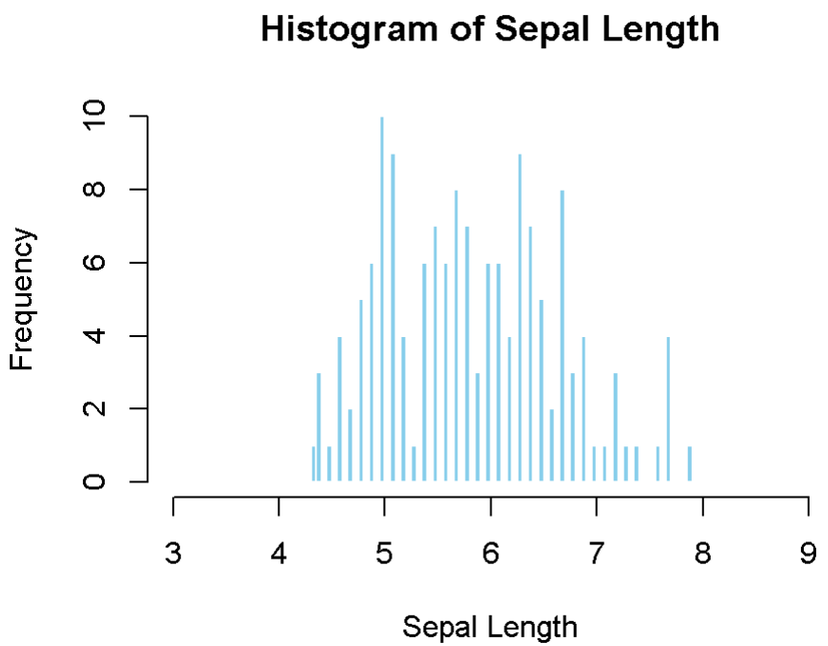
Let’s Do Some Visualization

Base R Graphics: Histogram

An important decision for histograms is this number (or width) of bins

Specified with the `breaks` argument



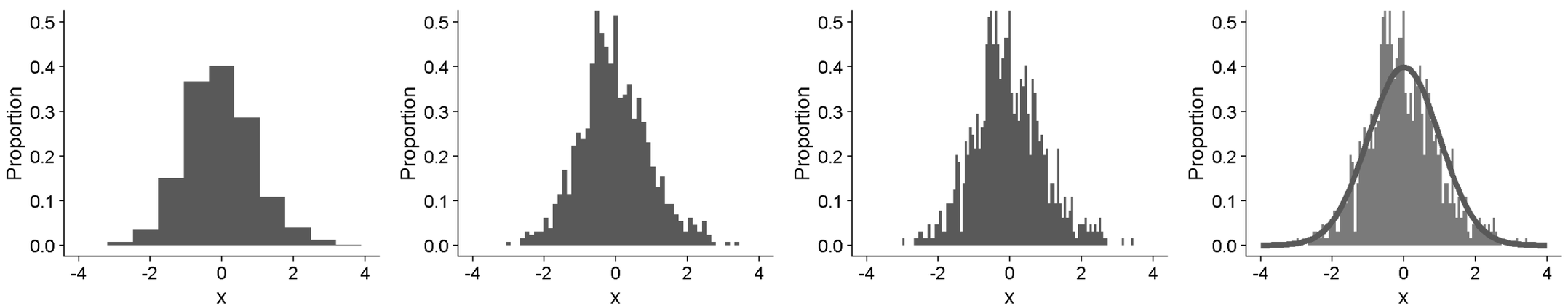


Let’s Do Some Visualization

Base R Graphics: Histogram

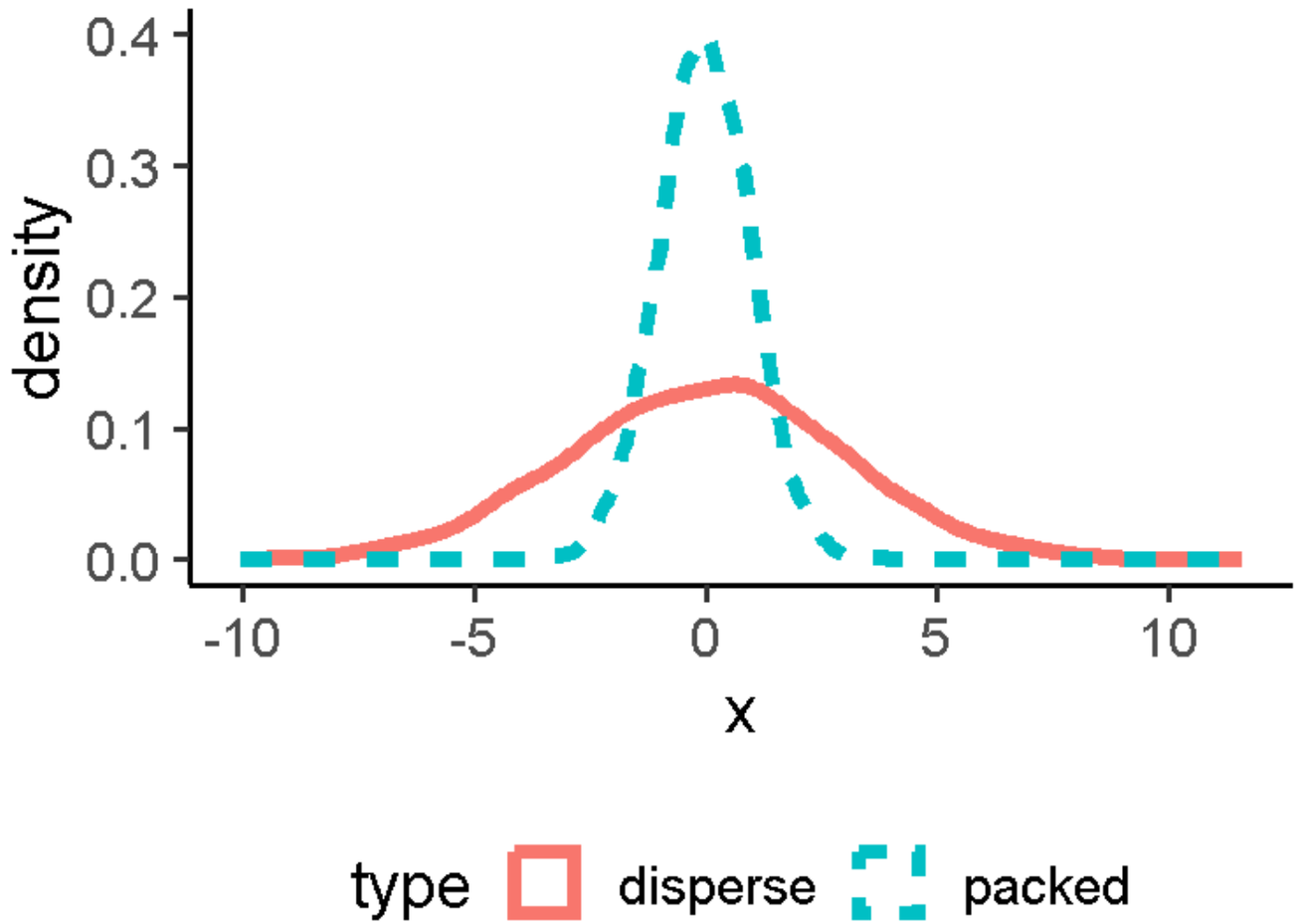
If we could make the bins infinitesimally small, we could get a probability density function (PDF)

Plot Code



Visualizations: Histogram

Can describe a distribution by its “dispersion”

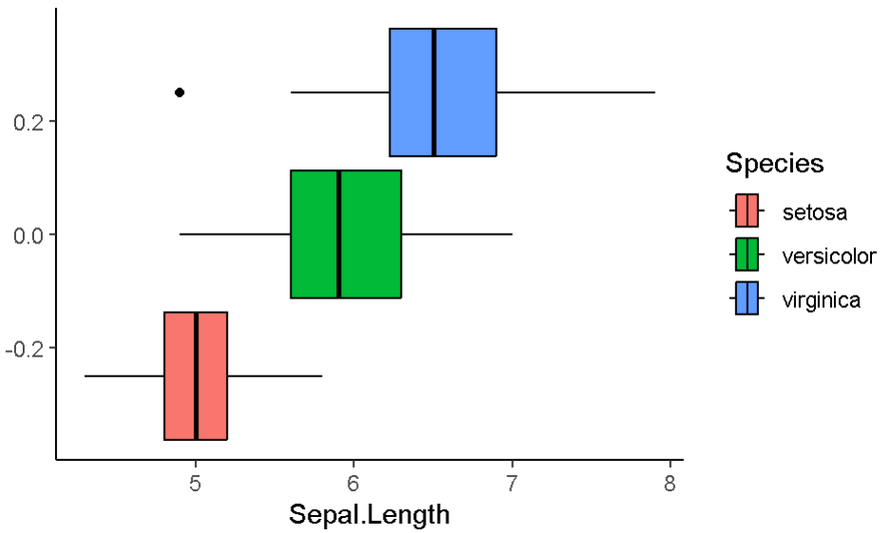


Visualizations: Boxplots

Back to the `iris` dataset, distribution of Sepal Length by species

Anatomy of a boxplot:

- “Minimum”
- 25th Quantile (Q1)
- Median
- 75th Quantile (Q3)
- “Maximum”
- Points representing outliers



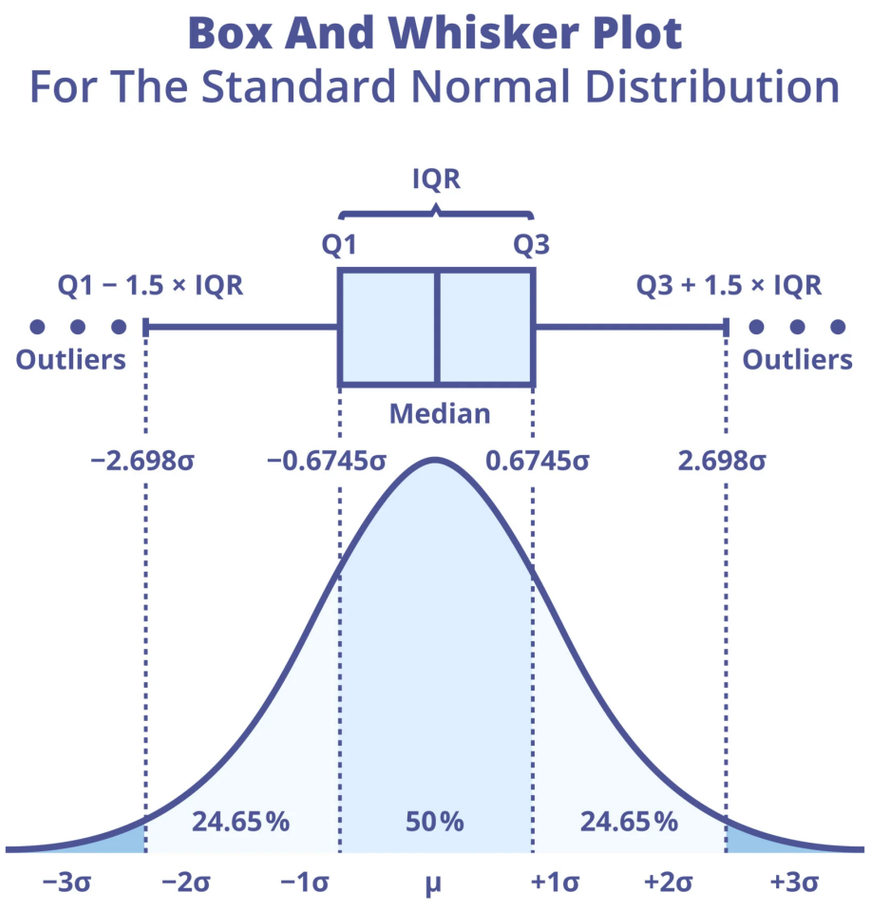
“Minimum” and “maximum” are not the *true* min and max

- Minimum: $Q1 - 1.5 * IQR$
- Maximum: $Q3 + 1.5 * IQR$

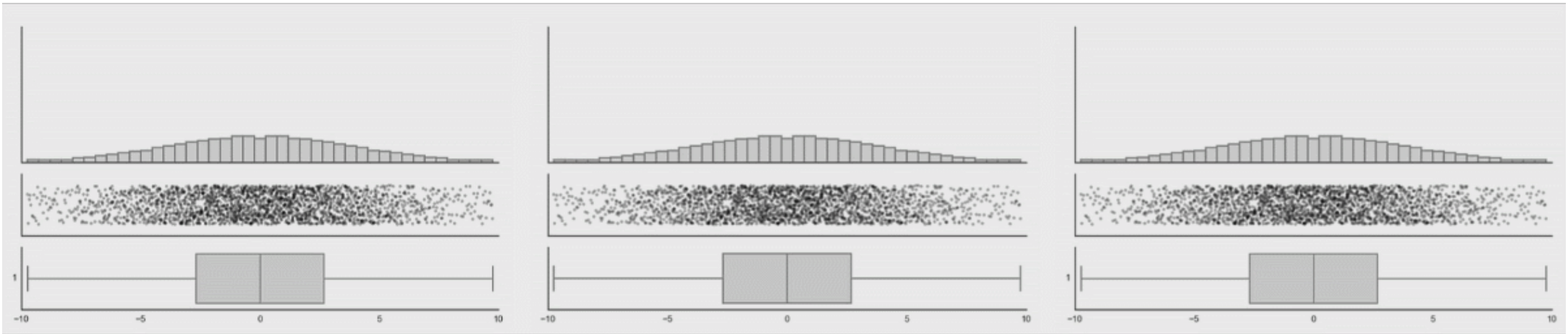
Means that the whiskers contain ~99% of the data, rest are outliers

Visualizations: Boxplots

One more resource for boxplot anatomy



Visualizations: Boxplots…



Datasaurus Dozen Boxplots

Let’s Do Some Visualization

Base R Graphics: Boxplot

`boxplot()` function

Its arguments are:

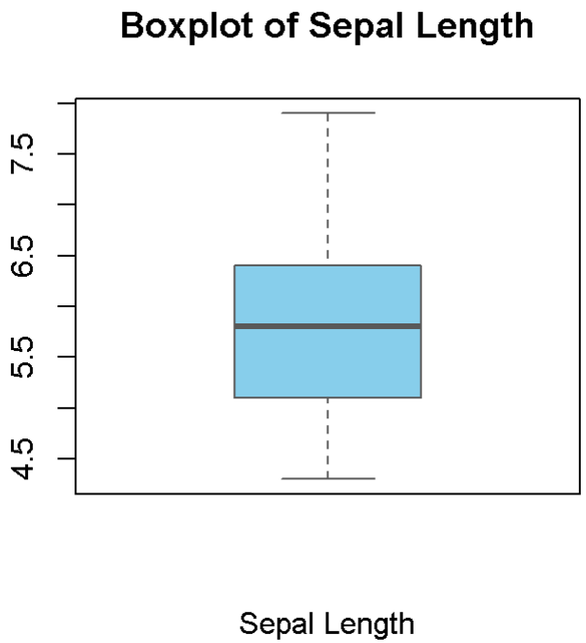
Required arguments:

- `x` = vector (variable) you want to plot

Optional arguments:

- `main`: title
- `xlab`: label for x-axis
- `ylab`: label for y-axis
- `border`: color for bar borders
- `col`: color for bars
- `horizontal`: `T`/`F` to switch

Plot | Code



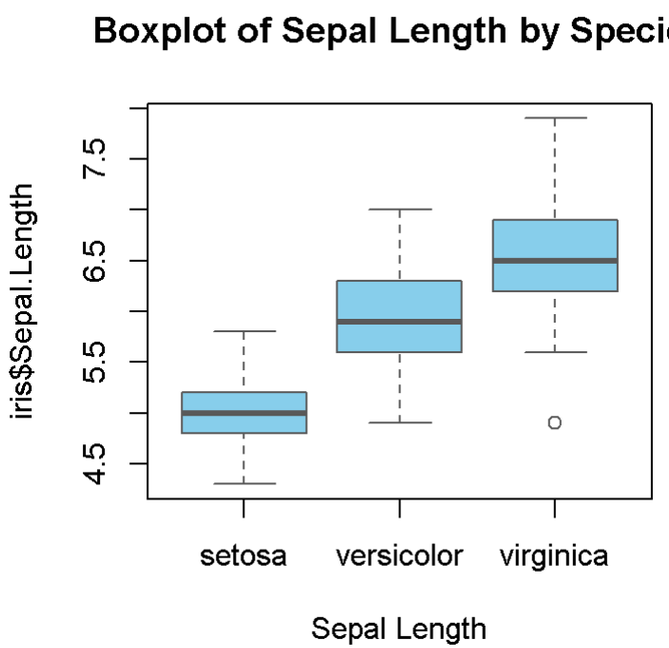
Let’s Do Some Visualization

Base R Graphics: Boxplot

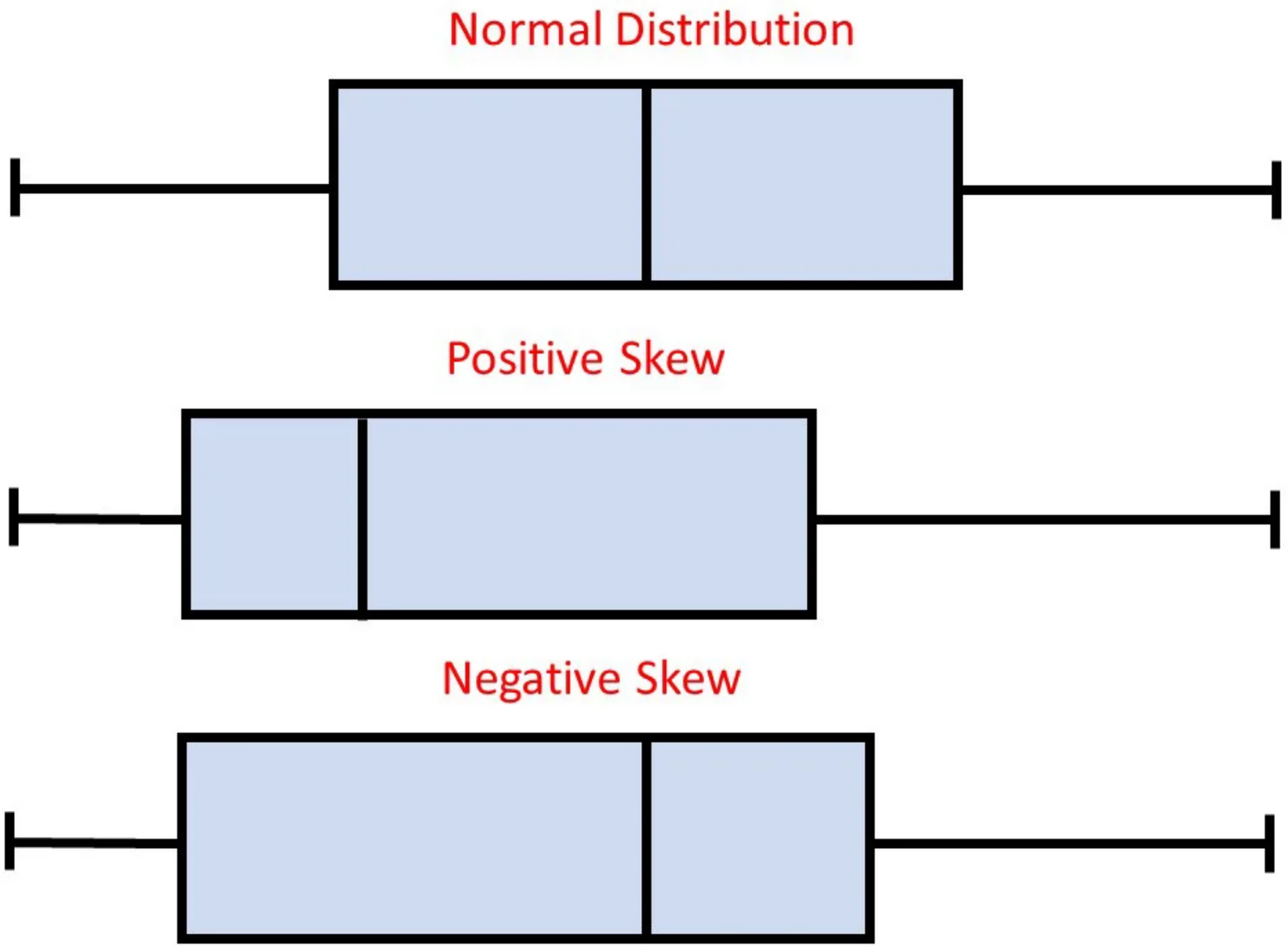
To group them, you can change the `x` to a “formula”

```
outcome_var ~ group_var
```

Plot | Code



Visualizations: Boxplot



Assignment 2