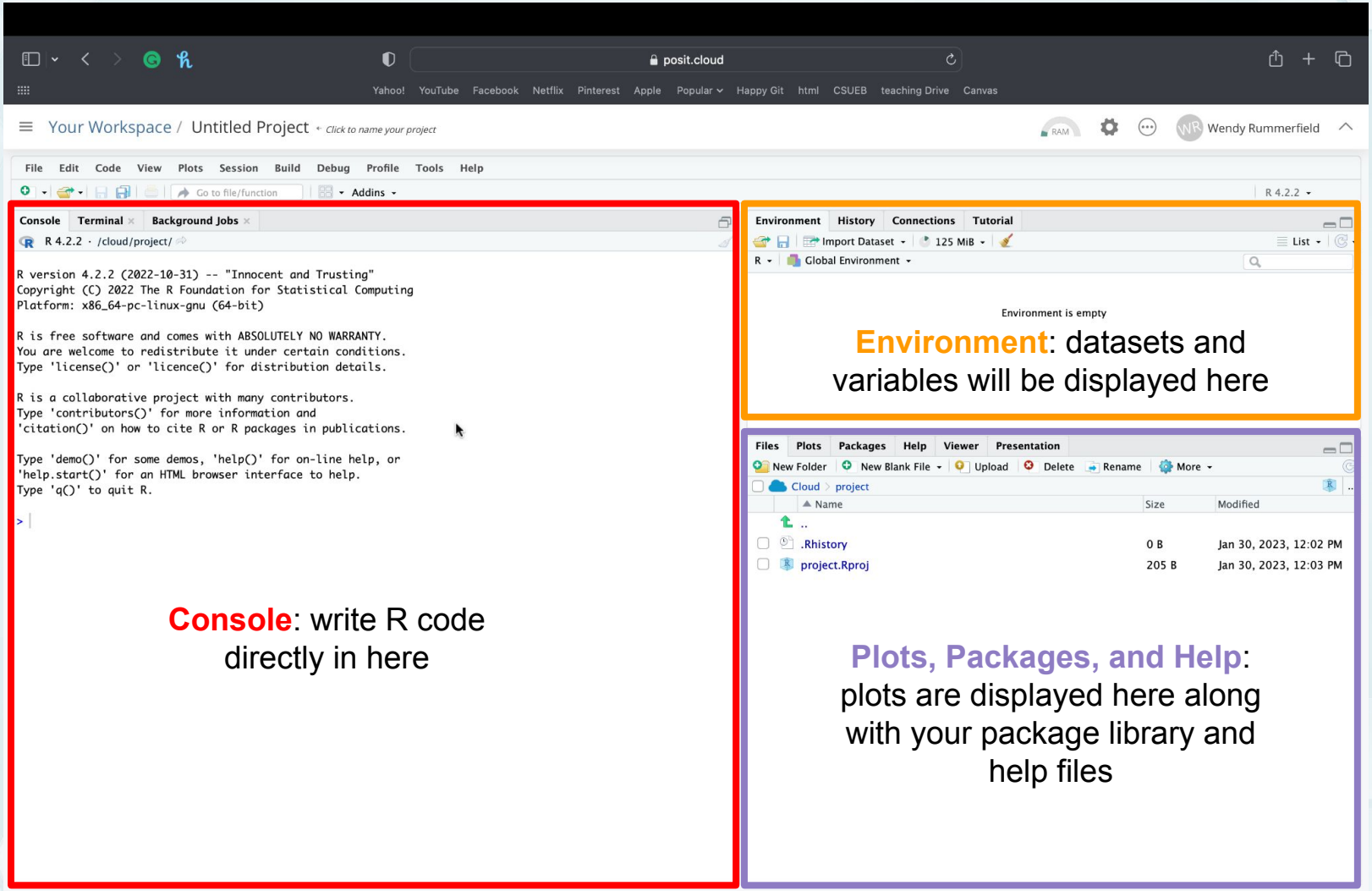
Lecture 3

R

ouR agenda

Announcements	Canvas updates, hw 1, other resources available soon (e.g., Git)
Mini-lecture	Let's get comfy with R terms
RStudio	Open RStudio on your desktop or on posit cloud and let's get coding!
Wrap-up	Any questions? Reminders for the weekend


posit.cloud
Yahoo! YouTube Facebook Netflix Pinterest Apple Popular Happy Git html CSUEB teaching Drive Canvas
Your Workspace / Untitled Project Click to name your project
Wendy Rummerfield
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
R 4.2.2
Console Terminal Background Jobs
R 4.2.2 · /cloud/project/
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Console: write R code directly in here
Environment History Connections Tutorial
Import Dataset 125 MiB List
R Global Environment
Environment is empty
Environment: datasets and variables will be displayed here
Files Plots Packages Help Viewer Presentation
New Folder New Blank File Upload Delete Rename More
Cloud project
Name Size Modified
..
.Rhistory 0 B Jan 30, 2023, 12:02 PM
project.Rproj 205 B Jan 30, 2023, 12:03 PM
Plots, Packages, and Help: plots are displayed here along with your package library and help files

R Variables

objects in R that make it easier for us to refer to them in the code

```
name <- stuff
```

name your variables something descriptive

assignment operator (save the `stuff` on the right using the `name` on the left)

- Number
- Word
- Equation/Formula
- Vector: list of numbers
- Dataframe: what R calls at dataset

R Variables

objects in R that make it easier for us to refer to them in the code

Examples

```
x <- 3
y <- 4
z <- (2*x - 3*y) / 5
name <- “Wendy”
```

More Examples

```
ages <- c(18, 35, 24)
weight <- ChickWeight$weight
avg_weight <- mean(weight)
med_weight <- median(weight)
```

The do's & don'ts of naming variables

Rule of thumb: name your variables and datasets something meaningful

Do:

- use *unique* names for different objects
  ```
  > x <- 3 and y <- 4
  ```
- use CamelCase or under_scores
  ```
  > MeanAge <- mean(ages)
  ```
- use names that make sense
  ```
  > mean_age <- mean(ages)
  ```

Don't:

- use the same name for different objects
  ```
  > x <- 3 and x <- 4
  ```
- leave any spaces
  ```
  > mean age <- mean(ages)
  ```
- use only numbers or symbols
  ```
  > 3 <- 4
  ```
- use function names
  ```
  > mean <- mean(ages)
  ```

dataframes & tibbles (datasets)

dataframes and **tibbles** (yes, that's a real word): they are just fancy names for datasets

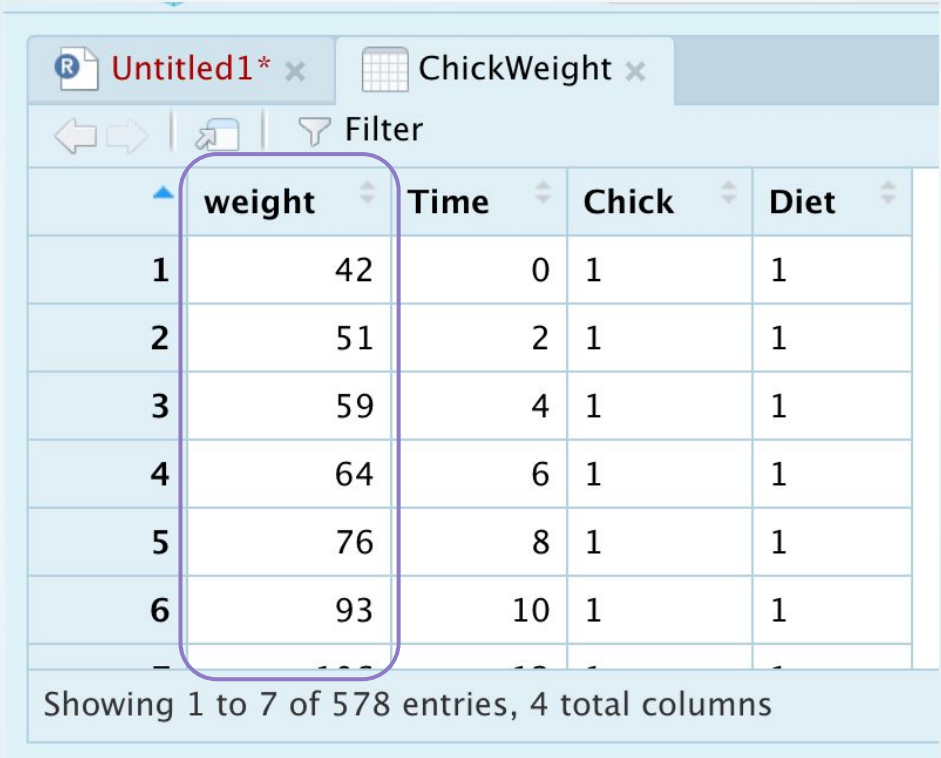
```
dataframe$colname
```

name of the dataset

how to access the columns in the dataframe

column name (make sure you type it EXACTLY as it is written)

ChickWeight$weight



	weight	Time	Chick	Diet
1	42	0	1	1
2	51	2	1	1
3	59	4	1	1
4	64	6	1	1
5	76	8	1	1
6	93	10	1	1

Showing 1 to 7 of 578 entries, 4 total columns

functions

Function: a block of code that performs something and outputs a result

```
function(arg1, arg2,...)
```

name of the function

type of input

another type of input

function examples

Some functions work right away any time you want to use RStudio
Other function work only after you install and load an R package.

Examples

```
mean(x)
```

*where x = any numeric R object

More Examples

```
> median(x)
> min(x)
> max(x)
> range(x)
> sd(x)
> var(x)
```

comments

Think of comments as a way for us to write notes to explain what our code is doing.

Syntax

```
# this is a comment
```

Start comments with a "#"

*Comments are ignored by R so they don't run like regular code.

Example

```
> mean(1:5)
[1] 3

> mean(1:5) # calculate the mean
[1] 3
```

ignored by R!

Let's try it!

Open RStudio or go to posit.cloud

Exploratory Data Analysis (EDA)

obtaining **descriptive statistics** and using **data visualization** to learn about your dataset before modeling

Summarizing Categorical Variables

Frequency

A count (must be a whole number)

Relative Frequency

The count *relative* to a total (can be a fraction, decimal, or percentage)

Bar plots

A visual way to represent categorical data (height of bars is frequency or relative frequency)

Tables

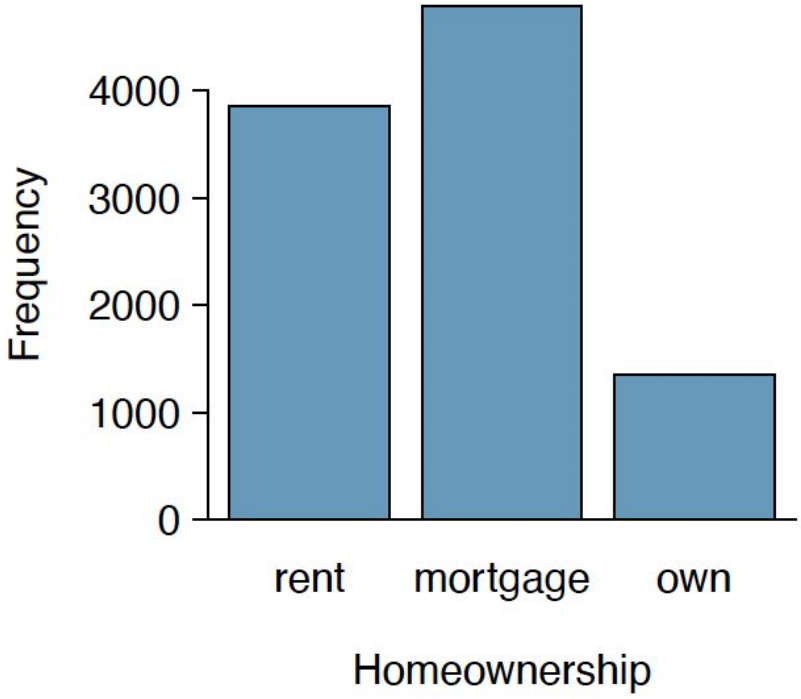
		homeownership			
		rent	mortgage	own	Total
app_type	individual	3496	3839	1170	8505
	joint	362	950	183	1495
	Total	3858	4789	1353	10000

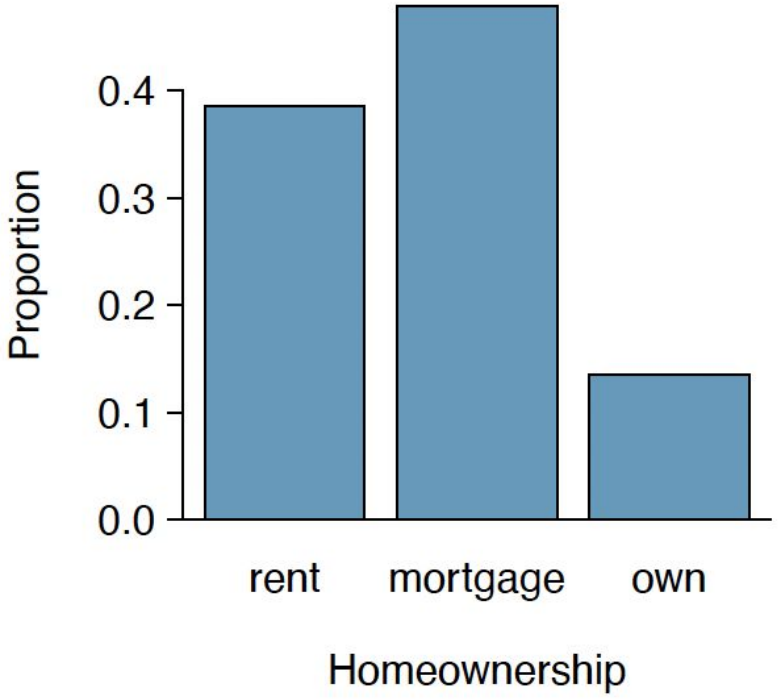
Figure 2.17: A contingency table for app_type and homeownership.

homeownership	Count
rent	3858
mortgage	4789
own	1353
Total	10000

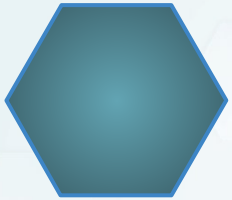
Figure 2.18: A table summarizing the frequencies of each value for the homeownership variable.

Bar plots


Frequency
4000
3000
2000
1000
0
rent
mortgage
own
Homeownership


Proportion
0.4
0.3
0.2
0.1
0.0
rent
mortgage
own
Homeownership

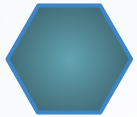
Making tables



table()

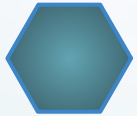
Input: 1 or more categorical (factor) variables

e.g.,
table(ChickWeight$Diet)



addmargins()

Input: a table
e.g.,
addmargins(table(ChickWeight$diet))



prop.table()

Input: a table
e.g.,
prop.table(table(ChickWeight$diet))

Distribution

A function/table that shows all the possible values for a **variable** and how often they occur

Summarizing *quantitative* variables

Center

roughly speaking, the middle of the data

- mean
- median
- mode

Spread

the amount of **variability** in the data

- standard deviation
- variance
- range (max - min)

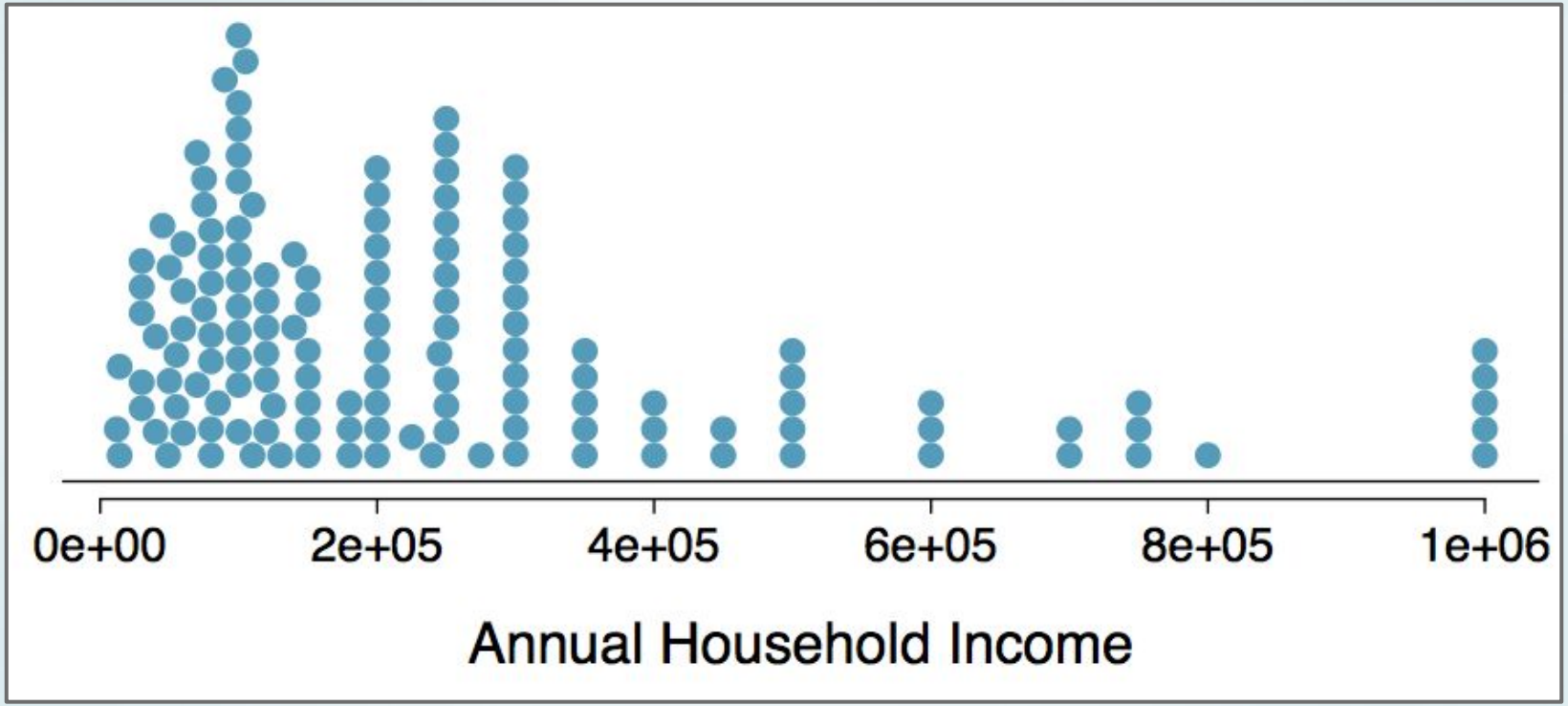
Shape

the literal shape of a **histogram**/**dot plot**/density plot

- modality (peaks)
- skewness

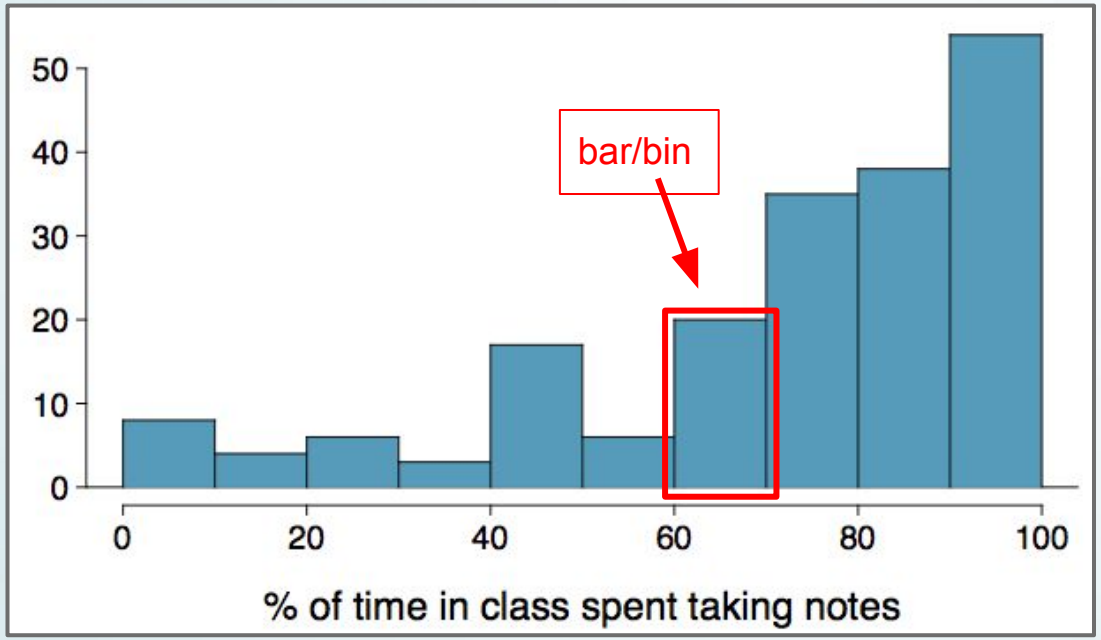
(Stacked) Dot plots

Distribution of Annual Household Income



Histogram

Percentage of time spent taking notes versus doing activities in class

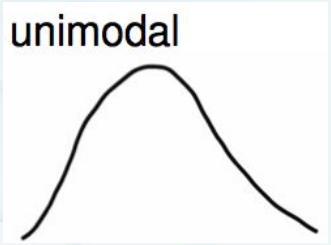


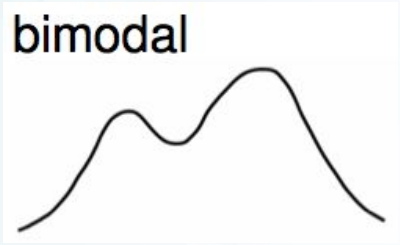
* Each bin contains all the data points that fall **within the interval**

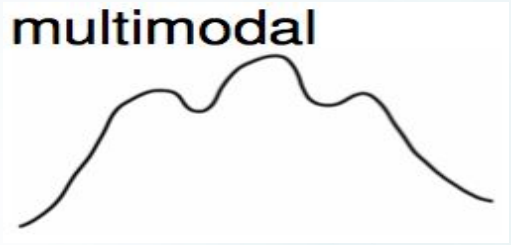
* Bin widths must be the **same** for all bins

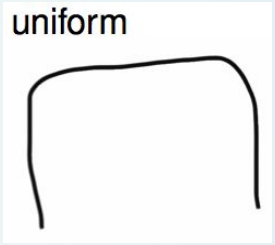
Shape (of histograms)

Modality

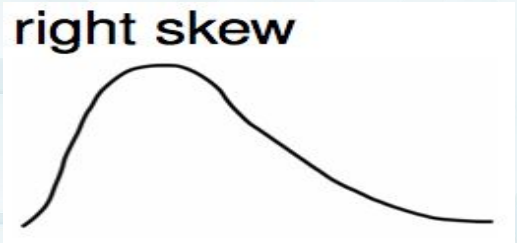


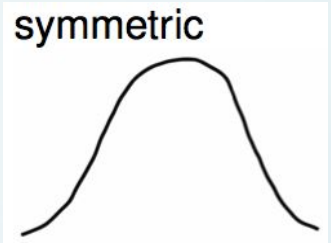




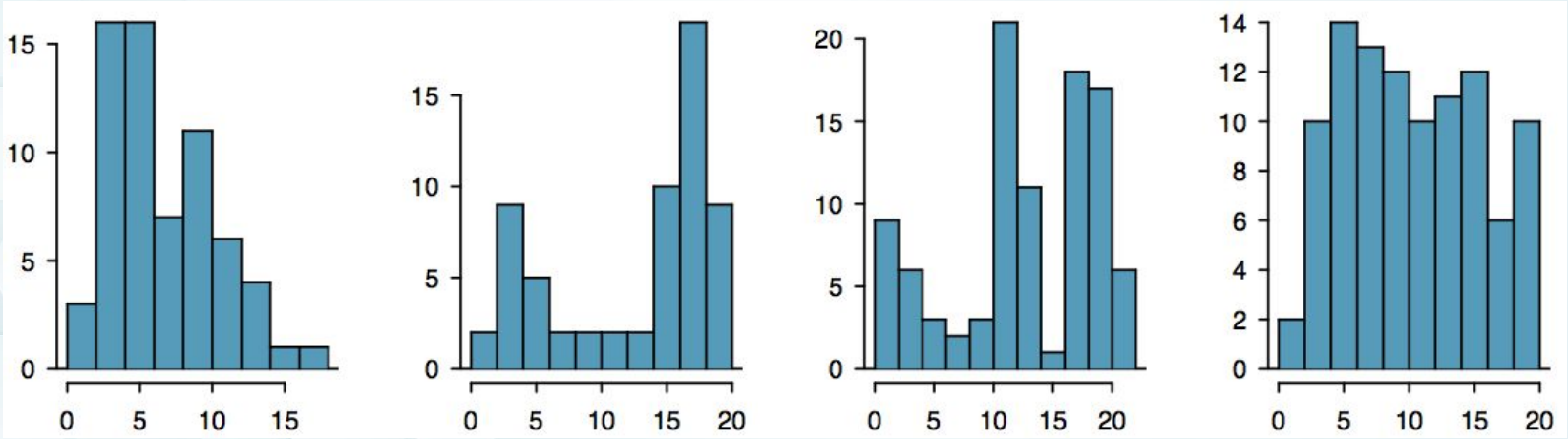


Skewness

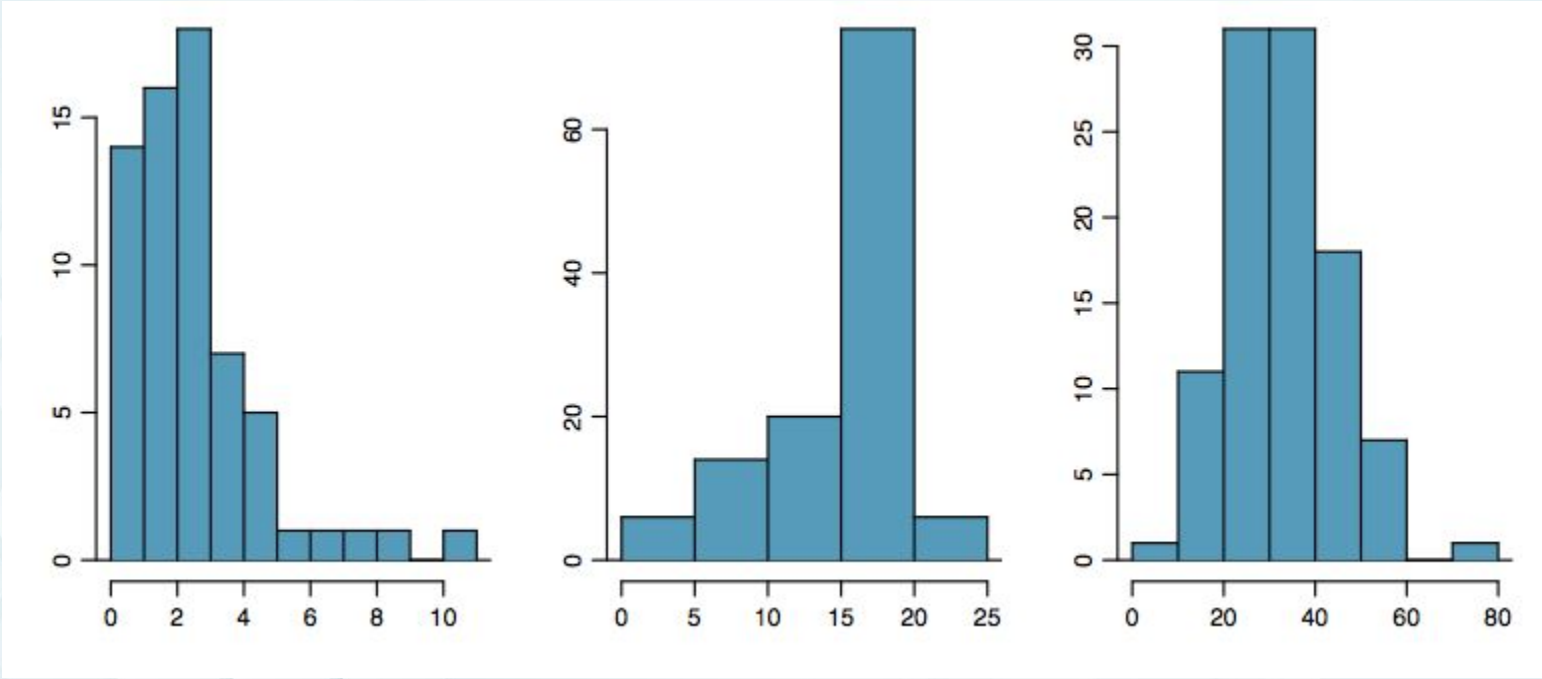




Modality


15
10
5
0
0 5 10 15
15
10
5
0
0 5 10 15 20
20
15
10
5
0
0 5 10 15 20
14
12
10
8
6
4
2
0
0 5 10 15 20

Skewness


0
5
10
15
0
2
4
6
8
10
0
20
40
60
0
5
10
15
20
25
0
5
10
15
20
25
30
0
20
40
60
80

Measures of center

Mean

The “balancing point” of a distribution (the arithmetic average)

$\bar{x},\ \mu$

Median

the “middle” of a distribution 50% below, 50% above

med or $median$

Mode

the highest point of a distribution (number that occurs most often)

m or $mode$

Measures of center

Mean

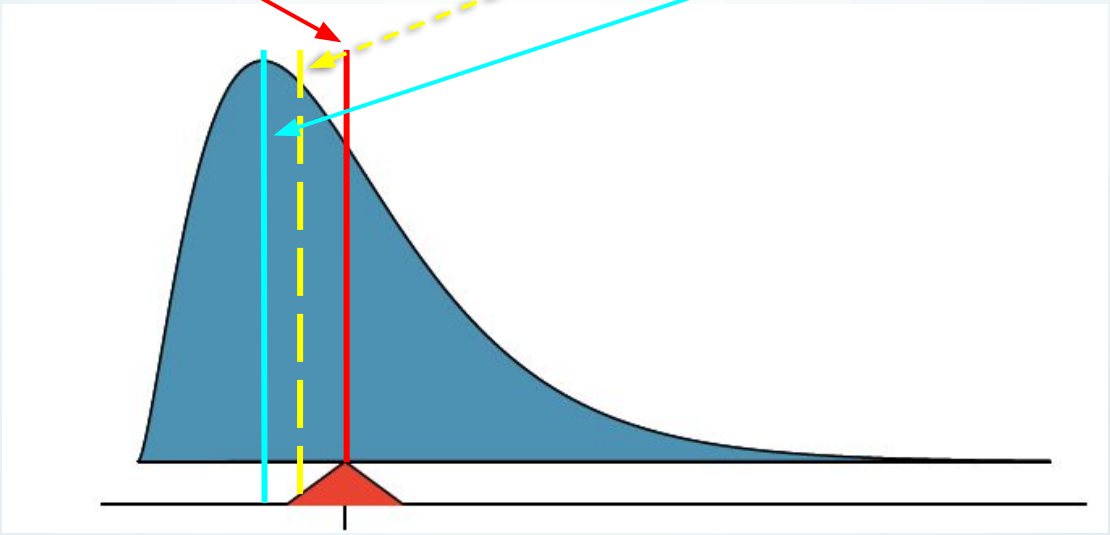
$\bar{x},\ \mu$

Median

med or $median$

Mode

m or $mode$



Calculating the 3 m's

mean

add up all the numbers and divide by the total

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

median

order data values from smallest to largest and pick the number in the middle (if there are two numbers, take the average)

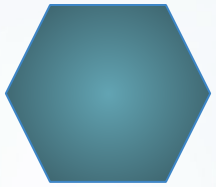
mode

See which value occurs the most often or what is the highest point in the distribution

Calculating the 3 m's

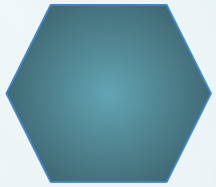
mean

mean()



median

median()



mode

Less obvious in R… I would just look at a plot

Measures of spread

Range

distance between the minimum and maximum value of a set of numbers

$$range = max - min$$

Variance

average squared deviations from the mean

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

Standard Deviation

square root of variance

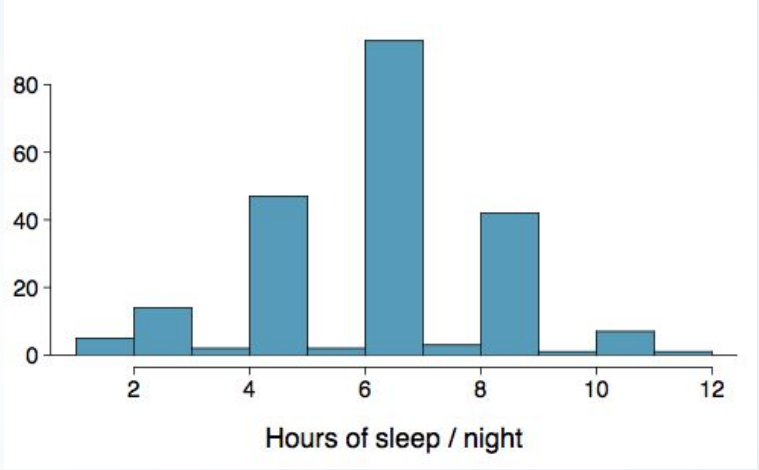
$$s = \sqrt{s^2}$$

Visualizing Variance

Calculating by hand

$$\bar{x} = 6.71$$

$$n = 217$$



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

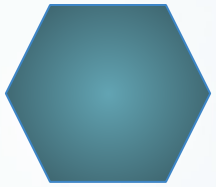
$$s^2 = \frac{(5-6.71)^2 + (9-6.71)^2 + \cdots + (7-6.71)^2}{217-1} = 4.11 \text{ hours}^2$$

$$s = \sqrt{4.11} = 2.03 \text{ hours}$$

Calculating in R

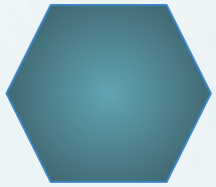
Range

`range( )`



Variance

`var( )`



Standard Deviation

`sd( )`

Enough lecturing

Let's keep coding!