

Stat 120

Deepak Bastola

2023-01-01

Contents

About	7
1 Class Activity 1	9
1.1 Your Turn 1	9
1.2 Your Turn 2	11
2 Class Activity 2	13
2.1 Your Turn 2	13
2.2 Summary of article on It depends on how you ask!	13
2.3 Your Turn 3	13
2.4 Gettysburg random sample	13
2.5 Driving with a Pet on your Lap	16
3 Class Activity 3	19
3.1 Your Turn 1	19
3.2 Your turn 2	19
4 Class Activity 4	21
4.1 Your Turn 1	21
4.2 Your Turn 2	28
5 Class Activity 5	39
5.1 Your Turn 1	39
5.2 Hollywood Movies Domestic Gross	39

5.3	Your turn 2	47
5.4	Example 2: Sleep	47
5.5	Example 3: Z-scores for Test Scores	48
5.6	Example 4: 5 number summaries	49
5.7	Example 5: Hot dog	50
5.8	Examples 6: Hollywood Movies World Gross revisited	52
5.9	Example 8: Ants on a Sandwich	56
6	Class Activity 6	57
6.1	Your Turn 1	57
7	Class Activity 7	81
7.1	Your Turn 1	81
7.2	Few Instructions	81
7.3	Parameters and Statistics	81
7.4	Example 1: Parameters and Statistics	82
7.5	Example 2: Using Search Engines on the Internet	83
7.6	Example 3: Simulation of a Sample Proportion	84
7.7	Example 4: Simulation for a Sample Mean	90
7.8	Example 5: Effect of sample size	96
7.9	Example 6: Bootstrap Sampling	97
8	(PART*) Basics R	101
9	What is R?	103
9.1	What is RStudio?	103
9.2	R Studio Server	103
9.3	R Markdown Basics	104
9.4	Installing R/RStudio (not needed if you are using the maize server)	104
9.5	Install LaTeX (for knitting R Markdown documents to PDF): . .	105
9.6	Updating R/RStudio (not needed if you are using the maize server)	105
9.7	Instructions	105

<i>CONTENTS</i>	5
10 R Markdown	107
10.1 Including Plots	108
10.2 Read in data files	109
10.3 Hide the code	111

About

This is a *sample* book written in **Markdown**.

Chapter 1

Class Activity 1

- Try to knit the file at the present stage and see if it compiles.
- You can add `\vspace*{1in}` in the body of this file to produce a vertical space of 1 inches.

1.1 Your Turn 1

- a. Run the following chunk. Comment on the output.

```
example_data = data.frame(ID = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10),  
                           Greeting = c(rep("Hello", 5), rep("Goodbye",5)),  
                           Male = rep(c(TRUE, FALSE), 5),  
                           age = runif(n=10, 20,60))
```

Click for answer

```
example_data
```

	ID	Greeting	Male	age
1	1	Hello	TRUE	56.78755
2	2	Hello	FALSE	50.80770
3	3	Hello	TRUE	26.44464
4	4	Hello	FALSE	54.06676
5	5	Hello	TRUE	54.38210
6	6	Goodbye	FALSE	20.47676

```
7  7  Goodbye  TRUE 26.62443
8  8  Goodbye  FALSE 31.97642
9  9  Goodbye  TRUE 27.00487
10 10 Goodbye  FALSE 37.95879
```

Answer: We see a data frame with four columns, where the first column is an **identifier** for the cases. We have information on the greeting types, gender, and age on these cases in the remaining columns.

b. What is the dimension of the dataset called ‘example_data’?

Click for answer

```
dim(example_data)
```

```
[1] 10  4
```

```
nrow(example_data)
```

```
[1] 10
```

```
ncol(example_data)
```

```
[1] 4
```

Answer: There are 10 rows and 4 columns.

1.2 Your Turn 2

- a. Read the dataset `EducationLiteracy` from the Lock5 second edition book.

Click for answer

```
# read in the data
education_lock5 <- read.csv("https://www.lock5stat.com/datasets2e/EducationLiteracy.csv")
```

- b. Print the header (i.e. first 6 cases by default) of the dataset in part a.

Click for answer

```
head(education_lock5)
```

	Country	EducationExpenditure	Literacy
1	Afghanistan	3.1	31.7
2	Albania	3.2	96.8
3	Algeria	4.3	NA
4	Andorra	3.2	NA
5	Angola	3.5	70.6
6	Antigua and Barbuda	2.6	99.0

- c. What is the dimension of the dataset in a?

Click for answer

```
dim(education_lock5)
```

```
[1] 188  3
```

Answer: There are 188 rows and 3 columns.

- d. What type of variables are **Country**, **EducationExpenditure**, and **Literacy**?

Click for answer

Answer: **Country** is a categorical variable. **EducationExpenditure** and **Literacy** are both quantitative variables.

- e. If we would like to use education expenditure to predict the literacy rate of each countries, which variable is the explanatory variable and which one is the response?

Click for answer

Answer: The education expenditure is the explanatory variable, and the literacy rate is the response.

Chapter 2

Class Activity 2

2.1 Your Turn 2

2.2 Summary of article on It depends on how you ask!

Click for answer

Answer:

This study aimed to measure the effects of psychological biases on estimates of compliance with public health guidance regarding COVID-19. Results showed that compliance estimates were reduced when questions were framed negatively and anonymity was increased. Effect sizes were large, with compliance estimates diminishing by up to 17% points and 10% points, respectively. These findings suggest that standard tracking surveys pose questions in ways that lead to higher compliance estimates than alternative approaches.

2.3 Your Turn 3

2.4 Gettysburg random sample

Let's take a simple random sample (SRS) of Gettysburg words. The “population” is contained in the spreadsheet `GettysburgPopulationCounts.csv`. Carefully load this data into R:

```
pop <- read.csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/Gettysberg")
head(pop)
```

```
  position size word
1         1    4 Four
2         2    5 score
3         3    3  and
4         4    5 seven
5         5    5 years
6         6    3  ago,
```

The `position` variable enumerates the list of words in the population (address).

a. Sample

Run the following command to obtain a SRS of 10 words from the 268 that are in the population:

```
samp <- sample(1:268, size=10)
samp
```

```
[1] 43 173 72 177 212 70 205 142 226 98
```

This tells you the position (row number) of your sampled words. What are your sampled positions? Why are your sampled positions different from other folks in class?

b. Get words and lengths

We will *subset* the data set `pop` to obtain only the sampled rows listed in `samp`. We do this using **square bracket notation** `dataset[row number, column number/name]`. Run the following command to find your sampled words and sizes:

```
pop[samp,]
```

	position	size	word
43	43	6	nation,
173	173	3	the
72	72	2	of
177	177	4	they
212	212	2	to
70	70	1	a
205	205	5	these
142	142	3	will
226	226	2	we
98	98	4	that

What are your sampled words?

c. Compute your sample mean

The word lengths in part (b) are the data for your sample. You can compute your sample mean using a calculator, or using R. Let's try R (you will find it faster!). First save the quantitative variable `size` in a new variable called `mysize`:

```
mysize <- pop[samp, "size"]
mysize
```

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

Then find the mean of these values:

```
mean(mysize)
```

```
[1] 3.2
```

How does this sample mean (from a truly random sample) compare to your sample mean from the non-random sample?

Click for answer

Answer: The true mean is 4.29. Your two means will likely vary. Since the many non-random samples generally overestimated the population mean length, it is

possible (but not guaranteed) that *your* one non-random sample gave a mean length that is greater than the random sample's mean length.

2.5 Driving with a Pet on your Lap

Over 30,000 people participated in an online poll on `cnn.com` conducted in April 2012 asking: “Have you ever driven with a pet on your lap”? We see that 34% of the participants answered yes and 66% answered no.

- a. Can you conclude that a random sample was used from the description given? Explain.

Click for answer

Answer: No you can't make this conclusion from the info given. In fact, an online poll at a website like `cnn.com` is almost always reporting results from a non-random sample. The people who respond are individuals who visit `cnn.com`, then see the online poll and decide to respond.

- b. Explain why it is not appropriate to generalize these results to all drivers, or even to all drivers who visit `cnn.com`.

Click for answer

Answer: This is a volunteer sample, and volunteer samples are often biased and can't be generalized to *all drivers* (the population). It is likely that people who have driven with a pet on their lap are more likely to respond to the poll.

- c. How might we select a sample of people that would give us results that we can generalize to a broader population?

Click for answer

Answer: A random sample of individuals from all U.S. drivers would need to be selected and given the poll question. (There are many ways to do this, the most common being a variation of random digit dialing where phone numbers are randomly selected from known area codes.)

d. Is the variable measured in this study quantitative or categorical?

Click for answer

Answer: Categorical (yes or no answer to the question).

Chapter 3

Class Activity 3

3.1 Your Turn 1

Hello!

3.2 Your turn 2

Chapter 4

Class Activity 4

4.1 Your Turn 1

4.1.1 Flowers v. Mississippi

The data set `APM_DougEvansCases.csv` contains data from 1517 potential black and white jurors for 66 cases that he was primary prosecutor for between 1992 and 2017. These jurors were available for Doug Evans to strike using his “peremptory strikes” during the jury selection phase.

4.1.2 (a) Inspect data

Read in the data

```
jurors <- read.csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/APM_DougEvansCases.csv")
```

```
# Anything after a # is a comment in R  
# dimension of dataset  
dim(jurors)
```

```
[1] 1517    6
```

Look at the first **three rows** of the data set

```
jurors[c(1,2,3), ]
```

```

trial__id race      struck_state defendant_race
1         4 Black Not struck by State      White
2         4 Black   Struck by State      White
3         4 White Not struck by State      White
      same_race      struck_by
1 different race Juror chosen to serve on jury
2 different race   Struck by the state
3   same race Juror chosen to serve on jury

```

To get the data from one variable, we use the command `dataset$variable`. For example, `jurors$struck_state` gives us the data values from the `struck_state` variable, which tells us if a juror was struck by the state from the jury pool. Here we can see the first 10 entries in this variable:

```
jurors$struck_state[1:10]
```

```

[1] "Not struck by State" "Struck by State"
[3] "Not struck by State" "Not struck by State"
[5] "Struck by State"     "Not struck by State"
[7] "Struck by State"     "Not struck by State"
[9] "Not struck by State" "Not struck by State"

```

4.1.3 (b) Table of counts and proportions

The `summary` command used with a data frame gives summaries of each variable

```
summary(jurors)
```

```

trial__id      race      struck_state
Min.   : 4.0   Length:1517   Length:1517
1st Qu.: 52.0   Class :character   Class :character
Median : 82.0   Mode  :character   Mode  :character
Mean    :112.6
3rd Qu.:170.0
Max.    :301.0

defendant_race  same_race      struck_by
Length:1517     Length:1517   Length:1517
Class :character Class :character Class :character
Mode  :character Mode  :character Mode  :character

```

The `table` command gives the distribution of counts for a single categorical variable. To obtain the count table for `struck_state` you need to

```
counts <- table(jurors$struck_state)
counts
```

Not struck by State	Struck by State
1084	433

We can add the `prop.table` command to turn these counts into proportions:

```
prop.table(counts)
```

Not struck by State	Struck by State
0.7145682	0.2854318

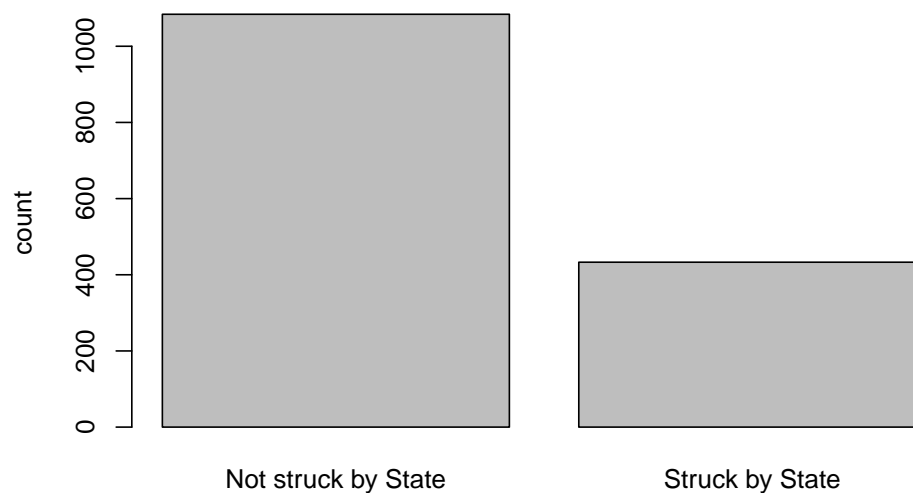
- What proportion of eligible jurors were struck by the state from the jury pool?

answer: about 28.5% of eligible jurors were struck by the state.

4.1.4 (c) Bar graph for one variable

You can create a simple bar graph for one categorical variable with the `barplot` command. Here we visualize the distribution of struck status for all eligible jurors:

```
barplot(counts, ylab = "count")
```



4.1.5 (d) Two-way tables

First 10 entries of `race` and `struck_state` variable is

```
jurors[(1:10), (2:3)]
```

```

      race      struck_state
1 Black Not struck by State
2 Black   Struck by State
3 White Not struck by State
4 White Not struck by State
5 Black   Struck by State
6 White Not struck by State
7 Black   Struck by State
8 White Not struck by State
9 White Not struck by State
10 White Not struck by State

```

The `table` command also gives two-way tables when two variables are included. Here is the two-way table for juror race and state struck status:

```
mytable <- table(jurors$race, jurors$struck_state)
mytable
```

```

      Not struck by State Struck by State
Black                225             310
White                859             123

```

- How many jurors were white and were not struck by the state?

answer: 859

4.1.6 (e) Conditional proportions: state strike status by juror race

The `prop.table` command gives conditional proportions for a two-way table. We plug our two-way table into `prop.table` with a `margin=1` to get proportions grouped by the `row` variable:

```
prop.table(mytable, margin = 1)
```


	Not struck by State	Struck by State
Black	0.4205607	0.5794393
White	0.8747454	0.1252546

Of all eligible black jurors, about 57.9% were struck by the state.

- What proportion of eligible white jurors were struck by the state?

answer: about 12.5%

- Is there evidence of an association between juror race and state strikes?

answer: Yes, there is an association because the rate of state strikes varies greatly by juror race with about 60% of black jurors were struck compared to only 13% of white jurors

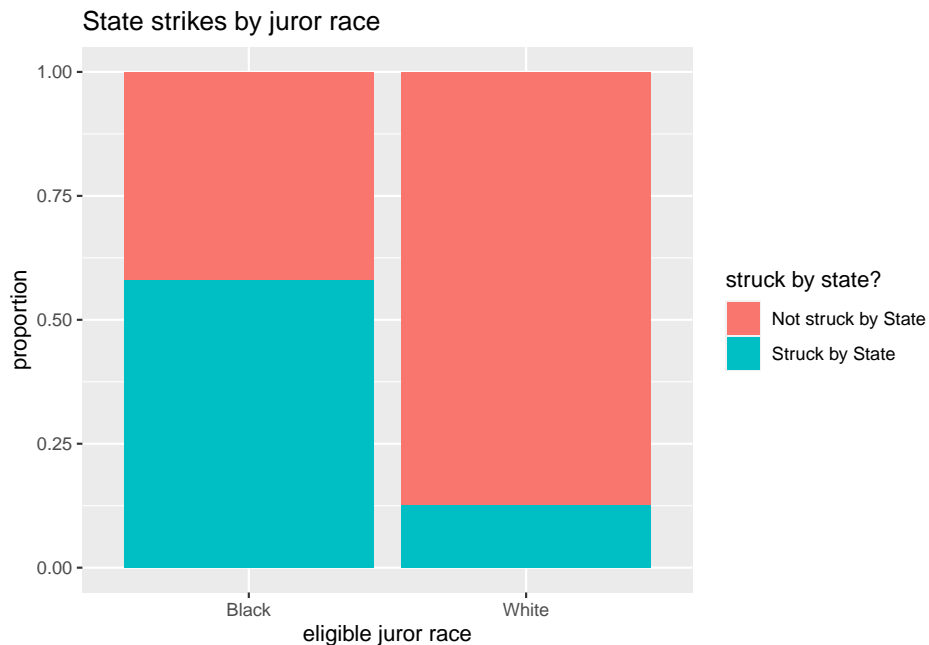
4.1.7 (f) Stacked bar graph for two variables

We can visualize the conditional distribution from part (e) with a stacked bar graph created using the `ggplot2` graphing package. First, load this package's functions with the `library` command:

```
library(ggplot2)
```

Now we can use the `geom_bar` command in this package. Here we get the conditional distribution of `struck_state` given `race`:

```
ggplot(jurors, aes(x = race, fill = struck_state)) +
  geom_bar(position = "fill") +
  labs(title = "State strikes by juror race", y = "proportion",
       x = "eligible juror race", fill = "struck by state?")
```



The basic syntax for this function is to let `ggplot` know your data set name (`jurors`), then specify the grouping or conditional variable on the x-axis (`race`) in the `aes` (aesthetic) argument. The `fill` variable is the response variable (`struck_state`). We add (+) the `geom_bar` geometry to get a bar graph with the `fill` position specified. Adding an informative label and title complete the graph.

4.1.8 (Non-Maize users) installing `ggplot2`

If you are using Rstudio on your **own computer**, you will first need to **install** the package but if you are using the Maize (online) Rstudio (or a lab computer) you do not. If you need to install the package:

- Click the **Packages** tab on the lower right Rstudio pane.
- Click **Install** and type `ggplot2` into the **Packages** box.
- Click the **Install** button. You should now see `ggplot2` in the list of packages.
- You only need to install the package once. After than, you run the `library` command to load the package functions into your current R session.

An alternate way way to install `ggplot2` from the R console is by using the following command:

```
install.packages("ggplot2", dependencies = TRUE)
```

4.1.9 (g) Conditional distribution of race grouped by strike status

We can “flip” our response and grouping variables easily (if we think it makes sense to do so). Here we specify the `margin=2` to get proportions grouped by the **column** variable:

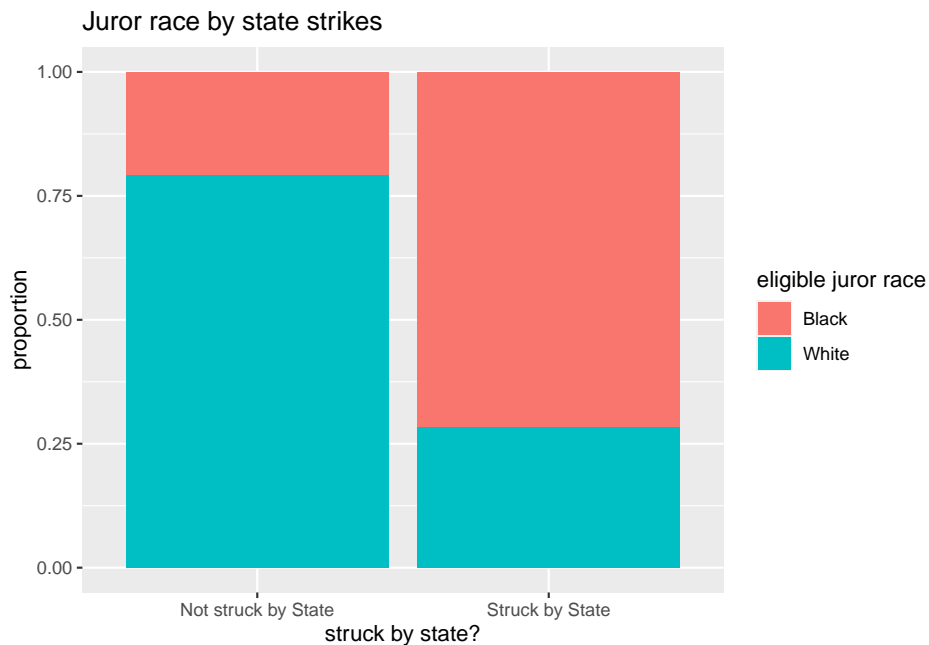
```
prop.table(mytable, margin = 2)
```

	Not struck by State	Struck by State
Black	0.2075646	0.7159353
White	0.7924354	0.2840647

Notice that the proportions add to one **down** each column. Of all eligible jurors struck by the state, about 71.6% were black.

The stacked bar graph for this distribution is

```
ggplot(jurors, aes(x = struck_state, fill = race)) +
  geom_bar(position = "fill") +
  labs(title = "Juror race by state strikes", y = "proportion",
       fill = "eligible juror race", x = "struck by state?")
```



- What proportion of eligible jurors who were not struck by the state were black? were white?

Answer: Of all jurors not struck by the state, about 20.8% were black

4.2 Your Turn 2

4.2.1 Graduate programs acceptance and sex

How are grad school program acceptance rates associated with sex? We will look at a classic data set from Berkeley grad school applications from 1973 (*Science*, 1975). The data cases are applicants to four graduate programs at Berkeley during 1973. The variable **result** tells us if the applicant was accepted to the graduate program, **sex** tells us the sex of the applicant (male or female), and **program** tells us program type (programs 1,2,3 or 4).

```
grad <- read.csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/Berkeley")
```

```
# dimension
dim(grad)
```

```
[1] 3014    3
```

```
# header
head(grad)
```

```
   program sex result
1 program1 male accept
2 program1 male accept
3 program1 male accept
4 program1 male accept
5 program1 male accept
6 program1 male accept
```

4.2.2 (a) Table of counts and proportions

```
prop.table(table(grad$result))
```

```
   accept    reject
0.4260119 0.5739881
```

- What proportion of applicants were accepted?

Answer: About 43% (1284/3014) of applicants were accepted.

4.2.3 (b) Two-way tables

The `table` command also gives two-way tables when two variables are included. Here is the two-way table for result and sex:

```
table(grad$sex, grad$result)
```

```
      accept reject
female    262    587
male     1022   1143
```

- How many applicants involved females who were accepted?

Answer: : 262 applicants involved females who were accepted.

4.2.4 (c) Conditional proportions: acceptance given sex

The `prop.table` command gives conditional proportions for a two-way table. First let's save the two-way table in an object named `mytable`:

```
mytable <- table(grad$sex, grad$result)
```

Then use `prop.table` to get the distribution of result conditioned (grouped) on applicant's sex:

```
prop.table(mytable, 1)
```

	accept	reject
female	0.3085984	0.6914016
male	0.4720554	0.5279446

The value of 1 in this command tell's R that you want *row* proportions (the denominator of the proportion is each row total).

- What proportion of female were accepted?

Answer: about 31% ($262/(262+587)$)

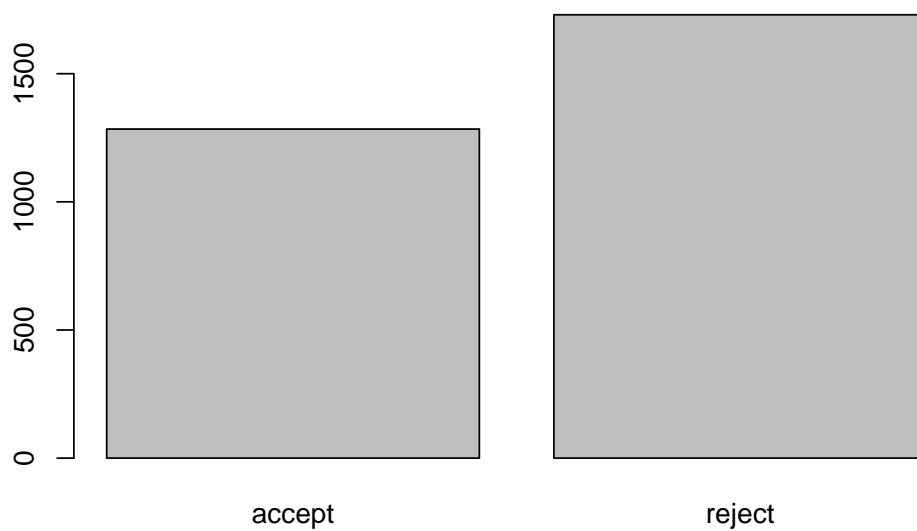
- What proportion of males were accepted?

Answer: about 47% ($1022/(1022+1143)$)

4.2.5 (d) Bar graph for one variable

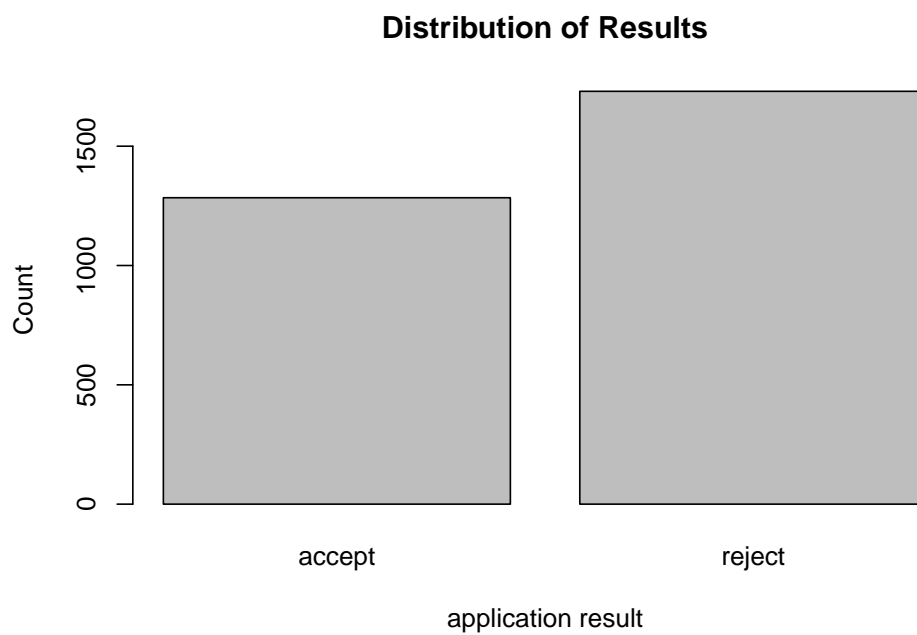
You can create a simple bar graph for one categorical variable with the `barplot` command. Here we visualize the distribution of result:

```
barplot(table(grad$result))
```



We can add in a title and x and y axis labels too:

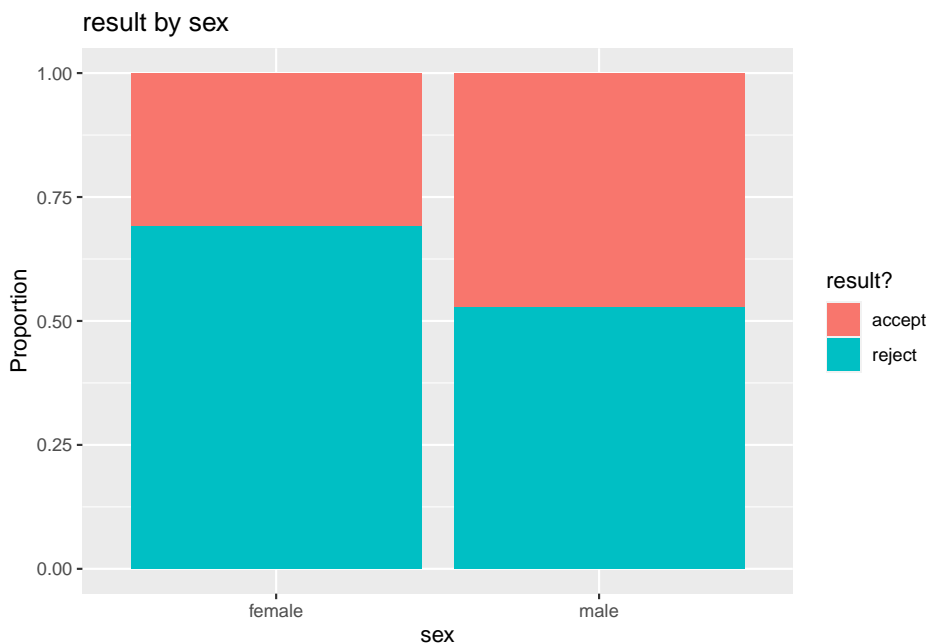
```
barplot(table(grad$result), xlab="application result",  
        ylab="Count", main = "Distribution of Results")
```



4.2.6 (e) Stacked bar graph for two variables

Now we can use the `geom_bar` command in this package. Here we get the conditional distribution of `result` given `sex`:

```
library(ggplot2) # don't need if you already entered it for example 1
ggplot(grad, aes(x = sex, fill = result)) +
  geom_bar(position = "fill") +
  labs(y="Proportion", title = "result by sex", fill = "result?", x = "sex")
```



The basic syntax for this function is to let `ggplot` know your data set name (`grad`), then specify the grouping or conditional variable on the x-axis (`sex`) in the `aes` (aesthetic) argument. The `fill` variable is the response variable (`result`). We add (+) the `geom_bar` geometry to get a bar graph with the `fill` position specified. Adding an informative label and title complete the graph.

- Verify that this graph is plotting the conditional proportions from part (c)

4.2.7 (f) Subsetting by program type

Finally, we will repeat the previous analysis of `result` and `sex`, but this time we will divide (or subset) the data set by program type. To do this we need to know how the values of `program` are coded:


```
table(grad$program)
```

```
program1 program2 program3 program4
      933      585      782      714
```

Here we use the `filter` command available from the `dplyr` package to get only the applicants to program 1:

```
library(dplyr)
grad.p1 <- filter(grad, program == "program1") # gets rows where program equal program1
head(grad.p1)
```

```
  program sex result
1 program1 male accept
2 program1 male accept
3 program1 male accept
4 program1 male accept
5 program1 male accept
6 program1 male accept
```

```
dim(grad.p1)
```

```
[1] 933  3
```

Verify that the number of rows in the subsetted program 1 data set matches the number of program 1 applicants shown in the `table` of counts above.

- Repeat the `filter` command to get a data set for program 2 and call the new data set `grad.p2`. Verify that the number of rows in this dataset matches the number of program 2 applicants in the original data set.

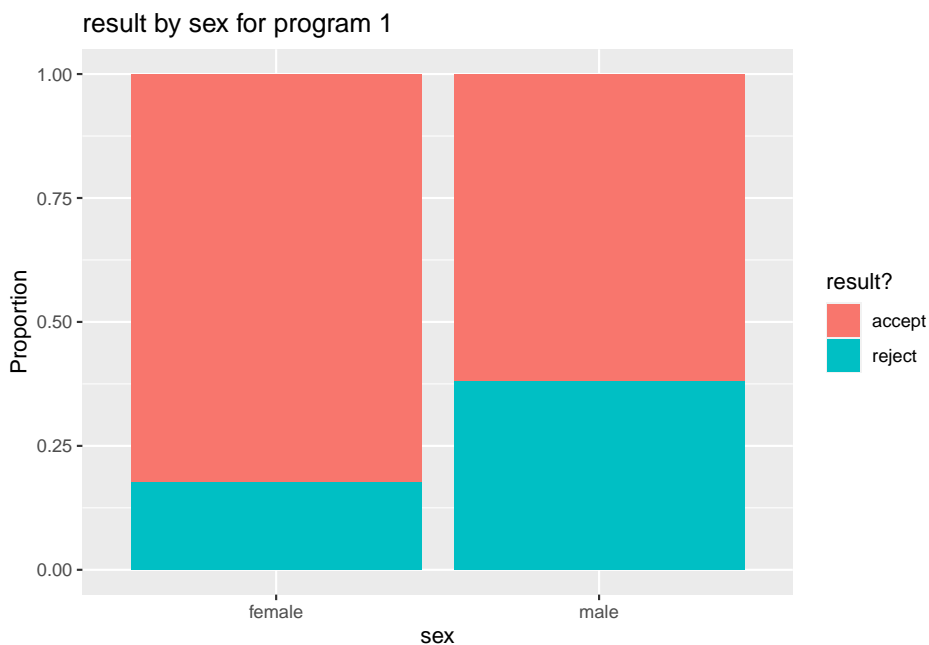
```
# enter R code for (f) here
grad.p2 <- filter(grad, program == "program2") # gets rows where program equal program1
head(grad.p2)
```

```
  program sex result
1 program2 male accept
2 program2 male accept
3 program2 male accept
4 program2 male accept
5 program2 male accept
6 program2 male accept
```

4.2.8 (g) Result by sex for program 1.

- Show the distribution of result conditioned on applicant's sex for the program 1 data set. Get both a table of conditional proportions (or percentages) and a stacked bar graph.

```
# enter R code for (g) here
ggplot(grad.p1, aes(x = sex, fill = result)) +
  geom_bar(position = "fill") +
  labs(y="Proportion", title = "result by sex for program 1",
       fill = "result?", x = "sex")
```



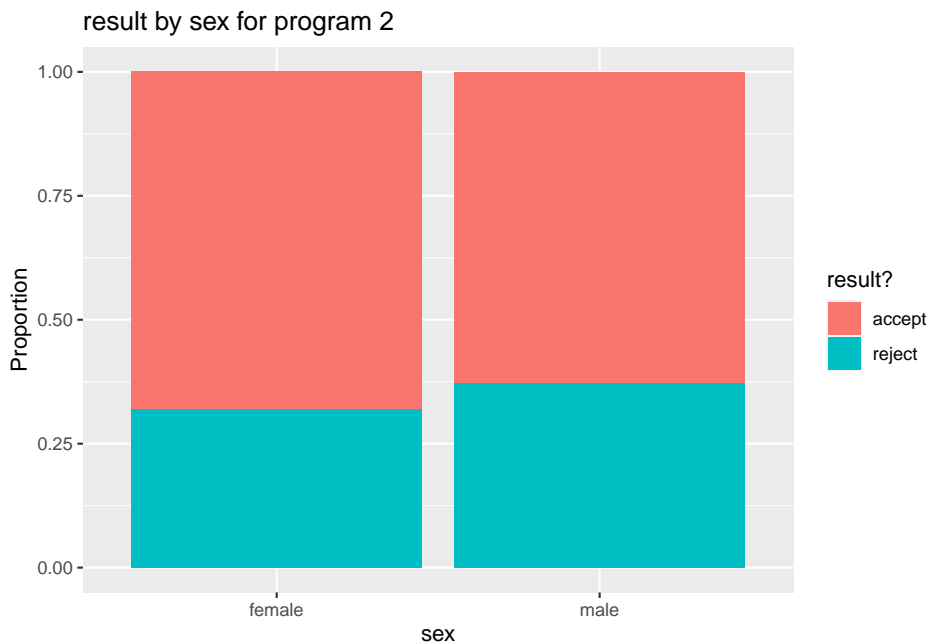
```
prop.table(table(grad.p1$sex, grad.p1$result),1)
```

	accept	reject
female	0.8240741	0.1759259
male	0.6193939	0.3806061

4.2.9 (h) Result by sex for program 2.

- Repeat part (g) but this time use the program 2 data set. Compare the two bar graphs for (g) and (h) and explain how they show that females have a higher acceptance rate after accounting for program type (1 or 2).

```
# enter R code for (h) here
ggplot(grad.p2, aes(x = sex, fill = result)) +
  geom_bar(position = "fill") +
  labs(y="Proportion", title = "result by sex for program 2",
       fill = "result?", x = "sex")
```



```
prop.table(table(grad.p2$sex, grad.p2$result),1)
```

```
      accept reject
female 0.6800000 0.3200000
male   0.6285714 0.3714286
```

Answer: For both programs 1 and 2, we see that female applicants have a slightly higher rate of acceptance than male applicants. After accounting for program type, we now see that black defendants have a higher rate of death penalty than white defendants. Without accounting for program type, the opposite was true (see parts (c) and (e)).

Why? the confounding affect of program type which is associated with both result and sex:

- females prefer to apply to programs 3 and 4 while males prefer programs 1 and 2 (more than 3 and 4).

- 44% of females applied to program 3 and 40% to program 4
- 38% of males applied to program 1 and 26% to program 2

```
prop.table(table(grad$sex, grad$program), 1)
```

	program1	program2	program3	program4
female	0.12720848	0.02944641	0.44169611	0.40164900
male	0.38106236	0.25866051	0.18799076	0.17228637

-Programs 3 and 4 were much harder to get into than programs 1 and 2 - 64% of applicants to program 1 were accepted and 63% of applicants to program 2 were accepted - 6% of applicants to program 4 were accepted and 34% of applicants to program 3 were accepted

```
prop.table(table(grad$program, grad$result), 1)
```

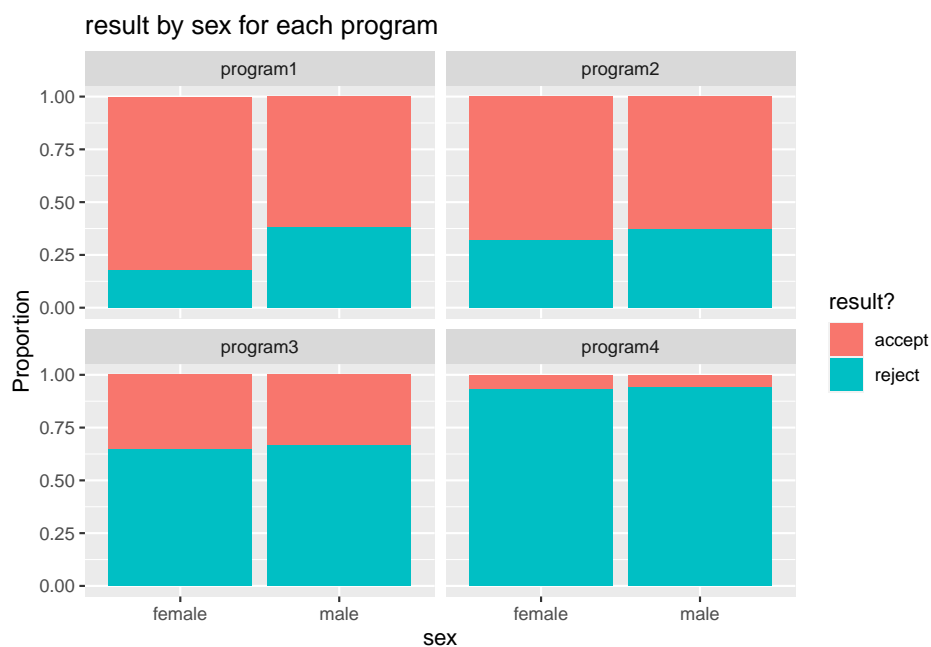
	accept	reject
program1	0.64308682	0.35691318
program2	0.63076923	0.36923077
program3	0.34398977	0.65601023
program4	0.06442577	0.93557423

So since the majority of females applied to the toughest programs (as measured by acceptance rates), there overall rate of acceptance was lower for females compared to males. But when we break down these rates by program type, we see that females have higher acceptance rates than males (see the visual in part (i)).

4.2.10 (i) A bar graph with three variables

If we simply want to graph the relationship between result and sex for each type of program, we can avoid subsetting the data by using the `facet_wrap` command in `ggplot2`. It is one simple addition to the stacked bar graph in part (e):

```
ggplot(grad, aes(x = sex, fill = result)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion",
       title = "result by sex for each program",
       fill = "result?",
       x = "sex") +
  facet_wrap(~program)
```



- Verify that this command creates side-by-side stacked bar graphs that match your graphs in parts (g) and (h) for programs 1 and 2.

Answer: The graphs match.

Chapter 5

Class Activity 5

5.1 Your Turn 1

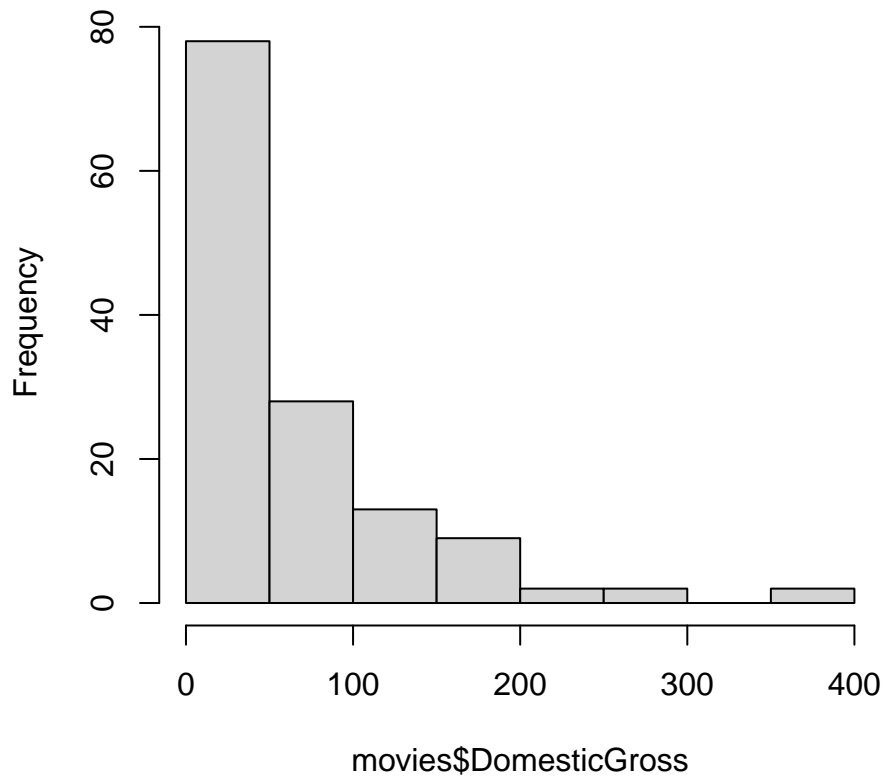
5.2 Hollywood Movies Domestic Gross

The dataset `HollywoodMovies2011` provides information on 136 movies that came out of Hollywood in 2011. We will look at the variable `DomesticGross`, which gives US domestic gross income for a movie from all viewers (in millions of dollars).

```
movies <- read.csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/HollywoodMovies2011.csv")
```

```
hist(movies$DomesticGross, main="Distribution of Domestic Gross")
```

Distribution of Domestic Gross



5.2.1 (a) Describe the shape of the distribution.

Answer: Skewed to the right

5.2.2 (b) Do there appear to be any outliers? If so, which values?

Answer: Yes, it looks like there are a few high outliers above 300 million.

5.2.3 (c) Finding outliers

We can find the row numbers of cases (movies) that have `DomesticGross` greater than 300 (300 million dollars):


```
which(movies$DomesticGross > 300)
```

```
[1] 4 14
```

Run the `which` command to verify that rows 4 and 14. Then find out which movies these are by subsetting the data frame:

```
movies[c(4,14), ]
```

	Movie				
4	Harry Potter and the Deathly Hallows Part 2				
14	Transformers: Dark of the Moon				
	LeadStudio	RottenTomatoes	AudienceScore	Story	
4	Warner Bros	96	92	Rivalry	
14	DreamWorks Pictures	35	67	Quest	
	Genre	TheatersOpenWeek	BOAverageOpenWeek	DomesticGross	
4	Fantasy	4375	38672	381.01	
14	Action	4088	23937	352.39	
	ForeignGross	WorldGross	Budget	Profitability	
4	947.10	1328.111	125	10.624888	
14	770.81	1123.195	195	5.759974	
	OpeningWeekend				
4	169.19				
14	97.85				

Note that the `c(4,14)` part of this command creates a **vector** of the numbers 4 and 14 (the `c` stands for combine). Which movies are the outliers?

Answer: Harry Potter and the Deathly Hallows Part 2 and Transformers: Dark of the Moon.

5.2.4 (d) Use the histogram to answer: Is the median less than 100 million, about 100 million, above 100 million?

Answer: It is the point with half the data to the left and half to the right. The median is less than 100 since 100 roughly 110 (80 + 30) cases below it which is well over half the movies in the data set.

5.2.5 (e) Do you expect the mean to be greater than or less than the median. Explain.

Answer: Because the distribution is skewed to the right, we expect the mean to be larger than the median. The large outliers will pull the mean up and won't

have much effect on the median.

5.2.6 (f) Computing the mean and median

You can get the mean and median a number of ways. Run these three commands:

```
mean(movies$DomesticGross)
```

```
[1] NA
```

```
median(movies$DomesticGross)
```

```
[1] NA
```

```
summary(movies$DomesticGross)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.02	19.03	37.35	63.22	80.46	381.01	2

What does NA stand for? How many movies have missing DomesticGross? You can subset the data to show you which cases have NA values for DomesticGross:

```
movies[is.na(movies$DomesticGross), ]
```

				Movie Lead	Studio
134				Hugo	Paramount
136	Never Back Down 2: The Beatdown				Sony
	RottenTomatoes	AudienceScore		Story	Genre
134		93		84	Adventure
136		NA		44 Rivalry	Action
	TheatersOpenWeek	BOAverageOpenWeek			DomesticGross
134		1277		8899	NA
136		NA		NA	NA
	ForeignGross	WorldGross	Budget		Profitability
134		NA	NA	NA	NA
136		NA	NA	3	0
	OpeningWeekend				
134		11.36			
136		8.60			

Answer: The NA value stands for “Not Available” which is used to code missing values. We can inspect the data frame and see that Hugo and Never Back Down 2 are the two movies that do not have domestic gross values.

5.2.7 (g) Missing data

There are some commands in R that “fail” as a default when missing data (NA) are present (`mean`, `median` and `sd` are examples). We can easily turn off this failure feature with the argument `na.rm=TRUE`

```
mean(movies$DomesticGross, na.rm=TRUE)
```

```
[1] 63.22276
```

```
median(movies$DomesticGross, na.rm=TRUE)
```

```
[1] 37.355
```

5.2.8 (h) Stats without outliers

There are a number of ways to “remove” outliers from an analysis. Here we use the square bracket `[]` notation along with a minus `-` to remove row 4 (Harry Potter) from the variable `DomesticGross` before our summary stat calculations:

```
summary(movies$DomesticGross[-4])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.02	18.88	37.30	60.83	80.36	352.39	2

Why does the mean change more than the median when this case is removed? (compare (g) and (h) mean and median values)

Answer: Both values go down after removing the highest grossing movie of the year, but the drop in the mean is more substantial. The mean drops by almost 4% when Harry Potter is removed while the median only drops by about 0.1%.

```
100*(60.83 - 63.22276)/63.22276 # percent change in the mean
```

```
[1] -3.78465
```

```
100*(37.30 - 37.355)/37.355 # percent change in the median
```

```
[1] -0.147236
```

5.2.9 (i) Computing standard deviation

The standard deviation command is `sd`. We need to add the `na.rm` argument to obtain the SD for `DomesticGross`:

```
sd(movies$DomesticGross, na.rm=TRUE)
```

```
[1] 69.41799
```

Look again at the distribution of `DomesticGross` shown in the histogram. Why is SD (variation around the mean) an inadequate measure of variation for this type of distribution?

Answer: There is much more variation (spread) to the data above the mean than below it. Because the distribution is strongly skewed right, we can't use one measure of variation when describing how `DomesticGross` values vary around some central value (like a mean).

5.2.10 (j) Stats by Genre

The `tapply(y, x, stat)` command gives the `stat` value of `y` for each level of `x`. Here we get the `summary` of `DomesticGross` for each type of `Genre`:

```
tapply(movies$DomesticGross, movies$Genre, summary)
```

`$Action`

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.54	24.96	40.26	91.02	161.53	352.39	1

`$Adventure`

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
NA	NA	NA	NaN	NA	NA	1

`$Animation`

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
21.39	51.41	115.67	104.62	142.86	191.45

`$Comedy`

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
5.79	23.21	37.41	56.51	69.75	254.46

`$Drama`

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.38	4.40	13.30	32.37	51.16	169.22

\$Fantasy

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.32	96.24	191.16	191.16	286.09	381.01

\$Horror

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.02	17.69	24.05	34.87	38.18	127.00

\$Romance

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.03	18.51	39.05	61.40	70.26	260.80

\$Thriller

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.02	31.18	40.49	41.44	62.50	79.25

- Which movies genre has the highest median domestic gross?
- Why are there no summary stats for the adventure genre?

Answer: To help answer these questions you really should explore the number of movies in each genre with the `table` command.

- The fantasy genre has the highest median domestic gross (\$381 million). But note that only two movies have this classification in 2011. The action genre was second highest at \$352 million and there were 12 movies in this category.
- The adventure genre only has one movie (Hugo) and this movie is also missing a value for `DomesticGross`!

```
table(movies$Genre)
```

Action	Adventure	Animation	Comedy	Drama	Fantasy
32	1	12	27	21	2
Horror	Romance	Thriller			
17	11	13			

```
which(movies$Genre == "Adventure")
```

```
[1] 134
```

```
movies[134, ]
```

```

      Movie LeadStudio RottenTomatoes AudienceScore Story
134  Hugo   Paramount           93           84
      Genre TheatersOpenWeek BOAverageOpenWeek
134 Adventure           1277           8899
      DomesticGross ForeignGross WorldGross Budget
134              NA           NA           NA      NA
      Profitability OpeningWeekend
134              NA           11.36

```

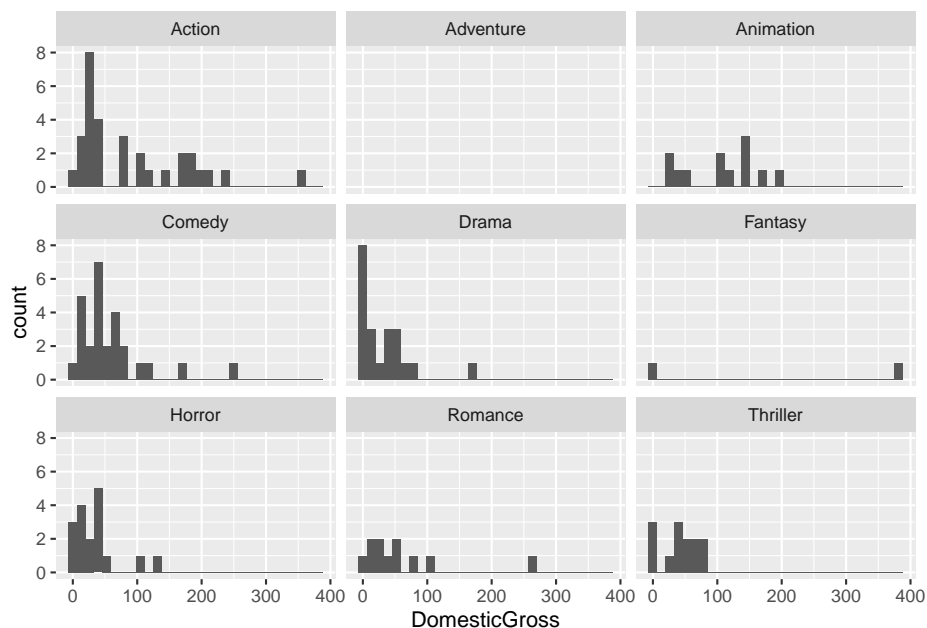
5.2.11 (k) Extra: Histogram of DomesticGross by Genre

(Not in Lab Manual) The `ggplot2` package allows you to create histograms separated by a categorical variable using the `facet_wrap` command. Assuming that `ggplot2` is already installed, all you need to do is load it with `library` then create your graph:

```

library(ggplot2)
ggplot(movies, aes(x=DomesticGross)) +
  geom_histogram() +
  facet_wrap(~Genre)

```



Which genre has the most variability in domestic gross?

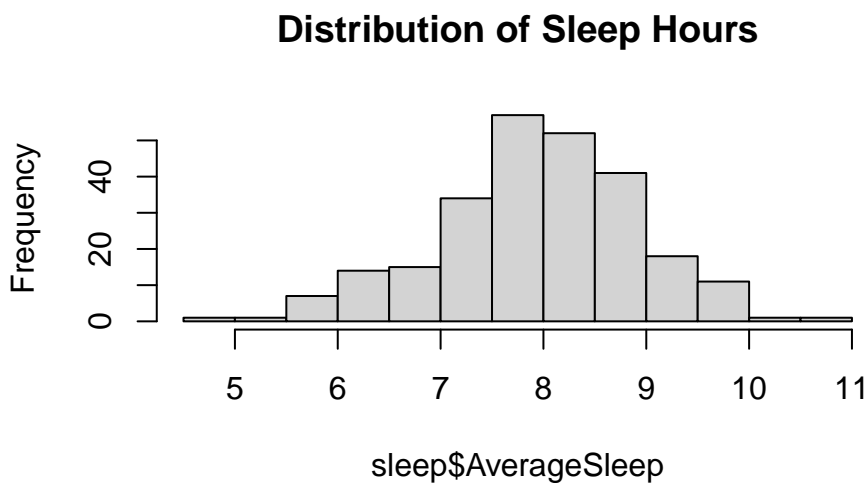
Answer: The action genre has the largest range of values.

5.3 Your turn 2

5.4 Example 2: Sleep

This histogram shows the distribution of hours or sleep per night for a large sample of students.

```
sleep <- read.csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/SleepStudy.csv")
hist(sleep$AverageSleep, main="Distribution of Sleep Hours")
```



5.4.1 (a) Estimate the average hours of sleep per night.

Answer: The mean is around 8 hours

5.4.2 (b) Use the 95% rule to estimate the standard deviation for this data.

Answer: Most of the data is between about 6 and 10, with a mean around 8 (due to the roughly symmetric distribution). So two standard deviations is about 2 hours of sleep, making one standard deviation about 1 hours of sleep.

Let's check the rule. Here are the actual mean and SD:

```
mean(sleep$AverageSleep)
```

```
[1] 7.965929
```

```
sd(sleep$AverageSleep)
```

```
[1] 0.9648396
```

5.5 Example 3: Z-scores for Test Scores

The ACT test has a population mean of 21 and standard deviation of 5. The SAT has a population mean of 1500 and a standard deviation of 325. You earned 28 on the ACT and 2100 on the SAT.

5.5.1 (a) Which test did you do better on?

Answer:

- ACT: The z-score for the score of 28 is $z = (28 - 21)/5 = 1.4$.
- SAT: The z-score for the score of 2100 is $z = (2100 - 1500)/325 = 1.85$.
- The SAT score is 1.85 standard deviations above average while the ACT score is only 1.4 standard deviations above. You did better on the SAT.

5.5.2 (b) For each test, find the interval that is likely to contain about 95% of all test scores.

Answer:

- ACT: Two standard deviations is $2(5) = 10$. About 95% of ACT scores are between $28 - 10 = 18$ and $28 + 10 = 38$. This claim assumes that ACT scores follow a bell-shaped distribution.
 - SAT: Two standard deviations is $2(325) = 650$. About 95% of SAT scores are between $1500 - 650 = 850$ and $1500 + 650 = 2150$. This claim assumes that SAT scores follow a bell-shaped distribution.
-

5.6 Example 4: 5 number summaries

For each five number summary below, indicate whether the data appear to be symmetric, skewed to the right, or skewed to the left.

5.6.1 (a) (2, 10, 15, 20, 69)

```
my_vector1 <- c(1, 10, 15, 20, 69)
summary(my_vector1)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	10	15	23	20	69

Answer: Skewed right. It has a longer right tail than left since $max - Q3 \gg Q1 - min$

5.6.2 (b) (10, 57, 85, 88, 93)

```
my_vector2 <- c(10, 57, 85, 88, 93)
summary(my_vector2)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
10.0	57.0	85.0	66.6	88.0	93.0

Answer: Skewed left since mean is less than median.

5.6.3 (c) (200, 300, 400, 500, 600)

```
my_vector3 <- c(200, 300, 400, 500, 600)
summary(my_vector3)
```

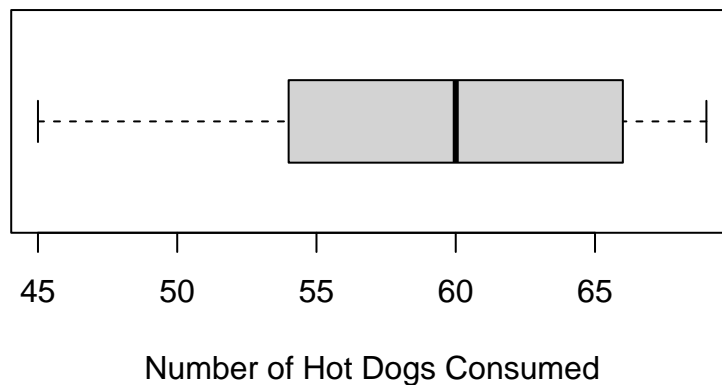
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
200	300	400	400	500	600

Answer: Symmetric since mean is same as median.

5.7 Example 5: Hot dog

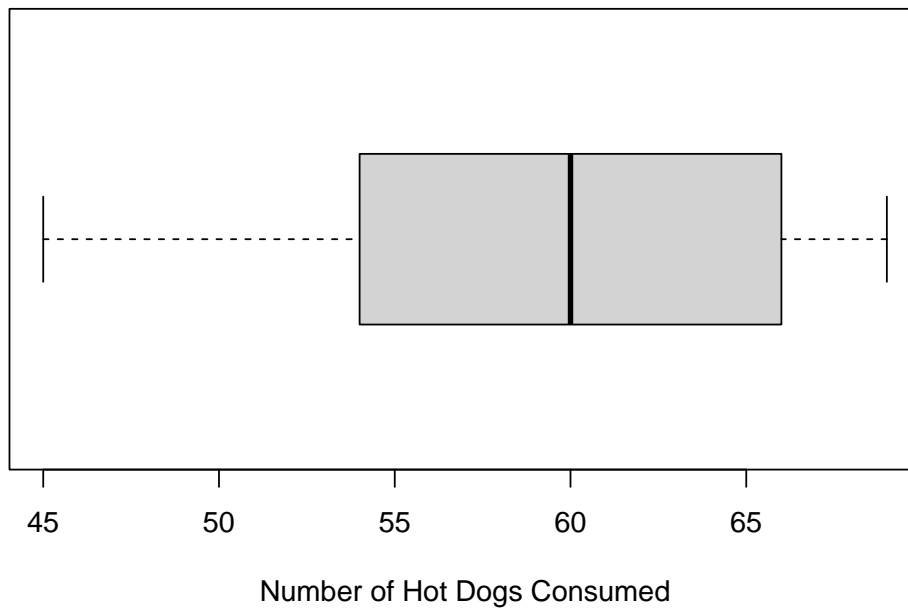
This boxplot shows the number of hot dogs eaten by the winners of Nathan's Famous hot dog eating contests from 2002-2011.

```
hotdogs <- read.csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/HotDogs")  
boxplot(hotdogs$HotDogs, xlab="Number of Hot Dogs Consumed", horizontal=T)
```



5.7.1 (a) Use the boxplot to estimate the 5 number summary and IQR for this data.

```
boxplot(hotdogs$HotDogs, xlab="Number of Hot Dogs Consumed", horizontal=T)
```



Answer: min = 45, Q1 = 50, m = 54, Q3 = 62, max = 67. IQR is about 62-50 or 12 hotdogs

5.7.2 (b) Computing 5 number summaries

R doesn't have '5 number summary' command, but `summary` gives you a "6" number summary by adding the mean to the 5 number summary. You can also use `IQR` to get the IQR:

```
summary(hotdogs$HotDogs)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
45.00	54.00	60.00	58.64	65.00	69.00

```
IQR(hotdogs$HotDogs)
```

```
[1] 11
```

How close were your guesses from the boxplot to the values given by this command?

(Answers will vary) Within one hotdog of the R values.

5.7.3 (c) Use the boxplot outlier rule to verify that there are no outliers in this data.

Answer:

- $1.5IQR = 18$ hotdogs.
- Lower fence: $Q1 - 1.5IQR = 50 - 18 = 32 < \min$ so there are no low outliers.
- Upper fence: $Q3 + 1.5IQR = 62 + 18 = 80 > \max$ so there are no high outliers.

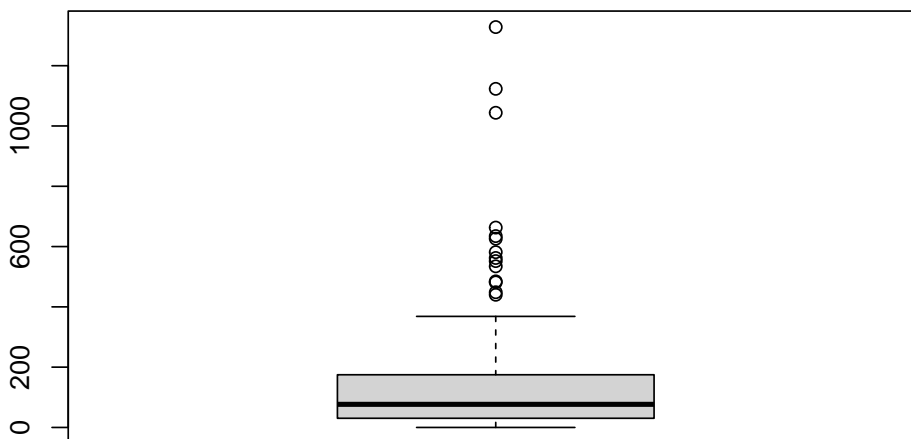
5.8 Examples 6: Hollywood Movies World Gross revisited

Let's revisit the WorldGross analysis from the Hollywood movies data set:

```
movies <- read.csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/Hollyw")
```

5.8.1 (a) Draw a boxplot of WorldGross.

```
boxplot(movies$WorldGross)
```



How many movies are identified as outliers for world gross?

Answer: Just using the boxplot, there looks to be about 10 movies that are high outliers

5.8.2 (b) Calculating boxplot values

Use the boxplot outlier rule to find the “fence” (cutoff) between an outlier and non-outlier for `WorldGross`. Then determine the value (of `WorldGross`) that the upper “whisker” (non-outlier) extends to.

```
summary(movies$WorldGross)
```

```

      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
0.025    30.706    76.659   150.742   173.691  1328.111
NA's
      2

```

```
IQR(movies$WorldGross, na.rm = TRUE)
```

```
[1] 142.985
```

- $1.5IQR = 1.5(142.985) = 214.48$ hundred million dollars
- Lower fence: $Q1 - 1.5IQR = 30.710 - 214.48 = -183.8 < min$ so there are no low outliers.
- Upper fence: $Q3 + 1.5IQR = 173.7 + 214.48 = 388.18 < max$ so there are high outliers.
- The upper whisker extends to the largest movie value that is below the fence of 388.18. You could look at the data spreadsheet and find which movie comes closest to this fence, but a quicker way is to use R. First we can use `which` to find out the row numbers of the movies with less than 388.18 in `WorldGross`. Then use this set to find out the max of the `WorldGross` within this group of movies, which turns out to be 368.404 hundred million dollars.

```
1.5*IQR(movies$WorldGross, na.rm = TRUE)
```

```
[1] 214.4775
```

```
30.710 - 214.48
```

```
[1] -183.77
```

```
173.7 + 214.48
```

```
[1] 388.18
```

```
notoutliers <- which(movies$WorldGross < 388.18)
max(movies$WorldGross[notoutliers])
```

```
[1] 368.404
```

```
which(movies$WorldGross == 368.404)
```

```
[1] 49
```

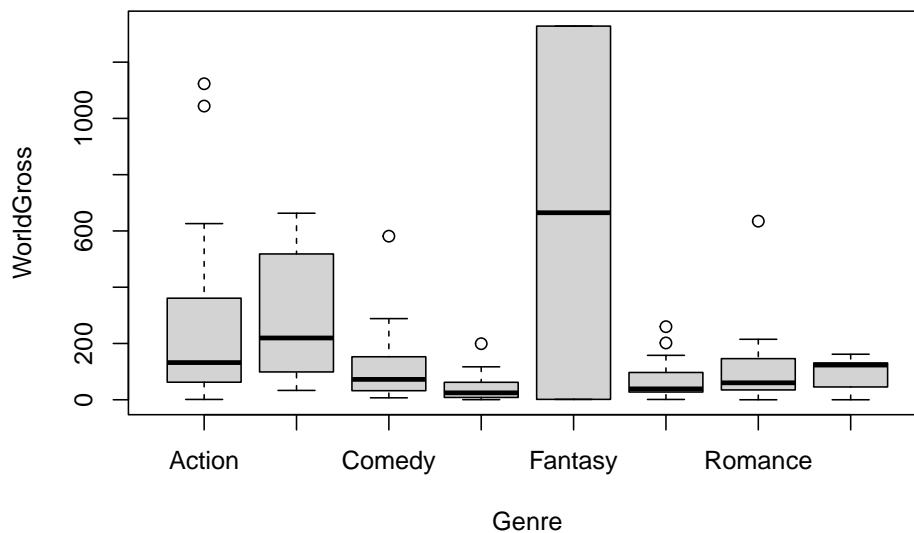
```
movies[49,]
```

	Movie	LeadStudio
49	Captain America: The First Avenger	Disney
	RottenTomatoes	AudienceScore
49	78	75
	Metamorphosis	Action
	TheatersOpenWeek	BOAverageOpenWeek
49	3715	17512
	DomesticGross	
49	176.65	
	ForeignGross	WorldGross
49	191.75	368.404
	Budget	Profitability
49	140	2.631457
	OpeningWeekend	
49	65.06	

5.8.3 (c) Side-by-side boxplot

We can compare boxplots of WorldGross across Genre categories:

```
boxplot(WorldGross ~ Genre, data=movies)
```



5.8. EXAMPLES 6: HOLLYWOOD MOVIES WORLD GROSS REVISITED55

- What does this type of graph illustrate well about the relationship between **WorldGross** and **Genre**?

Answer: Does a good job comparing median values and extremes

- What does this type of graph not illustrate well about the relationship between **WorldGross** and **Genre**?

Answer: It doesn't illustrate sample sizes well, e.g. the fantasy genre only has 2 movies in it

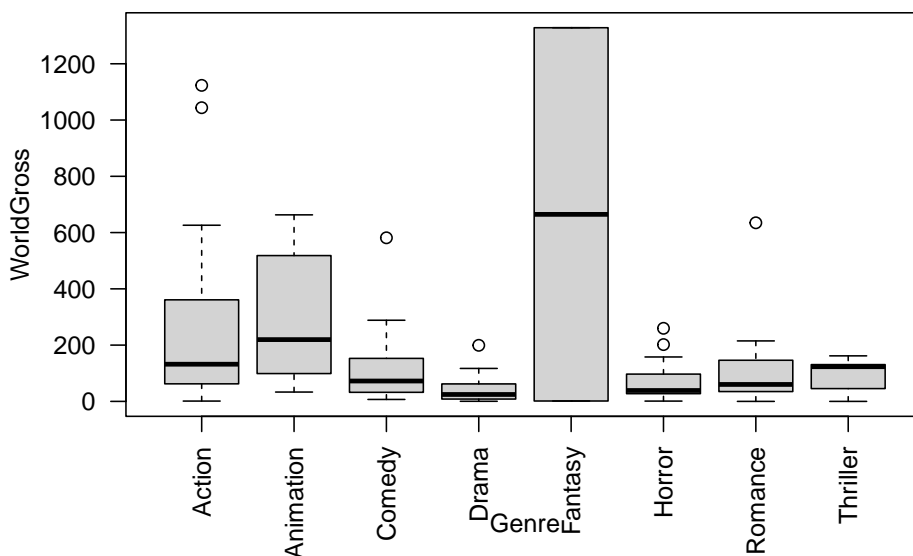
- What is one issue with the default version of this graph?

Answer: The genre labels are not all present.

5.8.4 (d) Improving the default boxplot

There are many values in **Genre** for this data and their values (levels) have longer names. This can cause issues when using these names to label graphs, like the x-axis in your boxplot. There are many (many, many) ways to modify graphs in R. Here is one way to change the label orientation on your x-axis.

```
boxplot(WorldGross ~ Genre, data=movies, las=2)
```



The `las` arguments let's you change the orientation of the axis labels relative to the axis. The value of 2 makes the labels perpendicular to the axis.

5.9 Example 8: Ants on a Sandwich

The number of ants climbing on a piece of a peanut butter sandwich left on the ground near an anthill for a few minutes was measured 7 different times and the results are: 43, 59, 22, 25, 36, 47, 19

5.9.1 (a) Calculate the mean number of ants.

Answer: $\bar{x} = 35.857$

5.9.2 (b) Calculate the median number of ants.

Answer: Order data then find middle value: 19, 22, 25, 36, 43, 47, 59. Then $m = 36$

5.9.3 (c) Calculate the quartiles for the number of ants.

Answer: Since $m = 36$, the first quartile will be the median of 19, 22, 25 : $Q1 = 22$. The third quartile will be the median of 43, 47, 59 : $Q3 = 47$.

Chapter 6

Class Activity 6

6.1 Your Turn 1

6.1.1 Beer Example

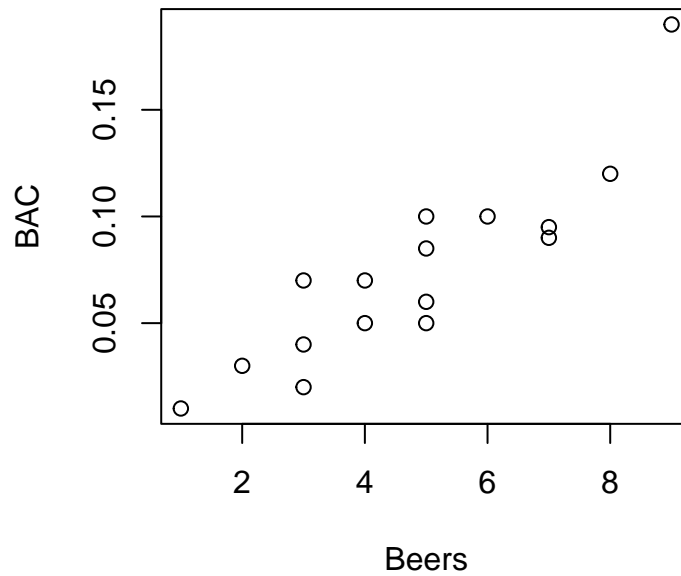
A study of 16 Ohio State University students looked at the relationship between the number of beers a student consumes and their blood alcohol content (BAC) 30 minutes after their last beer. The regression information from R to predict BAC from number of beers consumed is given below.

```
bac <- read.csv("http://math.carleton.edu/kstclair/data/BAC.csv")
```

6.1.2 (a) Always start with a visual!!!!

Plot the response (BAC) on the y-axis and the explanatory (“predictor”) on the x-axis.

```
plot(BAC ~ Beers, data=bac)
```

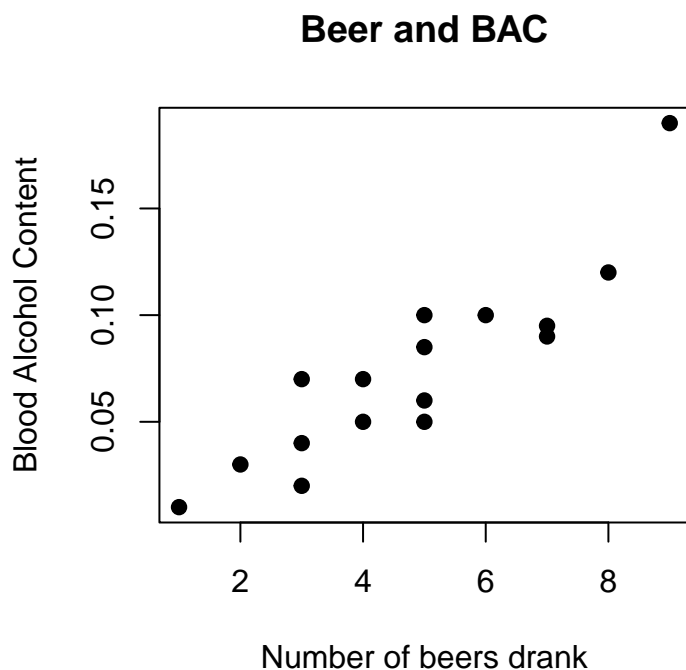


- Is there a relationship?

- direction?
- strength?
- form?

You can modify this basic graph by adding a title and changing the plotting symbol. The `pch=19` argument changes the symbols to filled circles.

```
plot(BAC ~ Beers, data=bac, pch=19,  
     main="Beer and BAC", xlab="Number of beers drank", ylab = "Blood Alcohol Content")
```



6.1.3 (b) Computing correlation

Since the *form* of the relationship is linear, we can use **correlation** to measure its strength:

```
cor(bac$BAC, bac$Beers)
```

```
[1] 0.8943381
```

6.1.4 (c) Fitting a regression line

We use the `lm(y ~ x, data=mydata)` function to fit a linear (regression) **model** for a response y given an explanatory variable x . This command creates a **linear model object** that needs to be assigned a name, here we call it `bac.lm`. You can get the slope and intercept by typing out the object name:

```
bac.lm <- lm(BAC ~ Beers, data=bac)
bac.lm
```

Call:

```
lm(formula = BAC ~ Beers, data = bac)
```

Coefficients:

(Intercept)	Beers
-0.01270	0.01796

- After running the `lm` command above in your R console, check the **Environment** tab to see that the object `bac.lm` is now one of the objects stored in R's memory (for this session of Rstudio).
- Write down the fitted regression equation to predict BAC from number of beers.

Answer: $\hat{y} = \dots$

- You can add this regression line to your scatterplot from part (a) by creating the plot and using the `abline` command:

```
# Need to call the plot function again!!
```

```
plot(BAC ~ Beers, data=bac, pch=19,  
     main="Beer and BAC", xlab="Number of beers drank", ylab = "Blood Alcohol Content",  
     abline(bac.lm) # adds regression line to the plot above
```



6.1.5 (d) Interpret the slope in context.

Answer: Drinking one more beer is associated with a 0.0180 unit increase in predicted BAC.

6.1.6 (e) Interpret the intercept in context, if it makes sense to do so.

Answer: The intercept is -0.0127. A student who drinks 0 beers would be predicted to have a negative blood alcohol content. This is not possible so the intercept does not make sense in this context, but the intercept is included in the model to get the best fit line for the data collected.

6.1.7 (f) If your friend at Ohio State drank 2 beers, what would you predict their BAC to be?

Answer: The predicted BAC is

$$\widehat{BAC} = -0.0127 + 0.0180(2) = 0.0233.$$

```
y.hat <- -0.0127 + 0.0180*(2)
y.hat
```

```
[1] 0.0233
```

6.1.8 (g) Find the residual for the student in the dataset who drank 2 beers and had a BAC of 0.03.

Answer: The residual is

$$BAC - \widehat{BAC} = .03 - .0233 = 0.0067$$

```
0.03 - (-0.0127 + 0.0180*(2))
```

```
[1] 0.0067
```

6.1.9 (h) Getting residuals in R

We can use the `resid` command to get the residuals for each case in the data set:

```
# part h
resid(bac.lm)
```

```
      1      2      3      4
0.022881795 0.006773080 0.041026747 -0.011009491
      5      6      7      8
-0.001190682 -0.018045729 0.028809318 -0.017118205
      9     10     11     12
-0.021190682 -0.027118205 0.010845557 0.004918033
     13     14     15     16
0.007881795 -0.023045729 0.004736842 -0.009154443
```

Notice that case 2 in the data drank 2 beers and had a BAC recorded as 0.03. We can see that their residual value matches our answer to (g) up to some rounding error.

```
# part h
bac$BAC[2]
```

```
[1] 0.03
```

```
bac$Beers[2]
```

```
[1] 2
```

```
resid(bac.lm)[2]
```

```
      2
0.00677308
```

6.1.10 (i) Getting R^2 value

You can use the `summary` command on an `lm` object to get a more detailed print out of your linear model, along with the R^2 value for your model:

```
summary(bac.lm)
```

Call:

```
lm(formula = BAC ~ Beers, data = bac)
```

```

Residuals:
      Min       1Q   Median       3Q      Max
-0.027118 -0.017350  0.001773  0.008623  0.041027

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.012701   0.012638  -1.005   0.332
Beers        0.017964   0.002402   7.480 2.97e-06 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02044 on 14 degrees of freedom
Multiple R-squared:  0.7998,    Adjusted R-squared:  0.7855
F-statistic: 55.94 on 1 and 14 DF,  p-value: 2.969e-06

```

6.1.11 (j) Making a residuals plot

The regression of BAC on Beers has a residuals plot that plots the model's residuals on the y-axis and the explanatory ("predictor") on the x-axis. We add a horizontal reference line (the detrended regression line) with the `abline(h=0)` command:

```

# code for residual plot
plot(resid(bac.lm) ~ Beers, data=bac, pch=19, main = "residuals plot")
abline(h=0)

```



Interpret: There is one case of 9 beers with a large residual (much higher BAC than predicted), but since there is no clear pattern (trend) in this plot it looks like our regression model adequately describes the relationship between number of beers and BAC.

- Is the magnitude of the scatter around the horizontal 0-line in the residuals plot greater than, less than, or the same as the magnitude of the scatter around the regression line in the scatterplot?

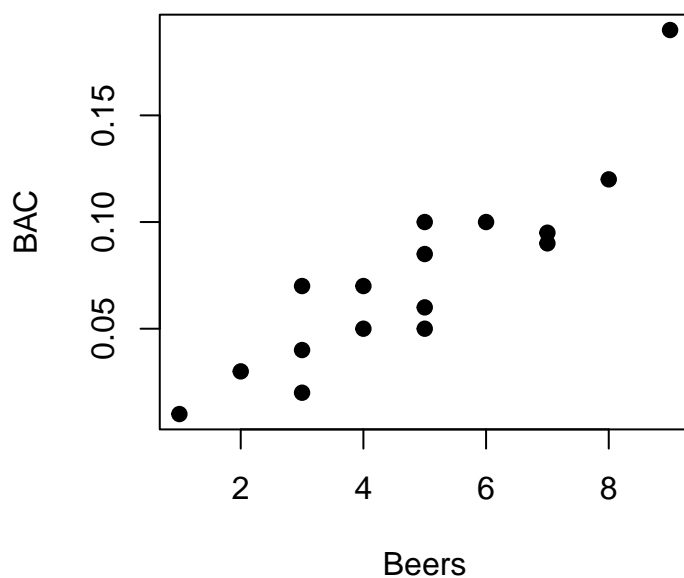
Answer: The same! The residuals plot is only a “detrended” scatterplot, meaning the vertical distances between a point and the regression line on the scatterplot or a point and the 0-line on the residuals plot are exactly the same. The residual plot looks more scattered because the trend is removed and the scale of the y-axis compressed.

6.1.12 (k) Identifying points

The `which` command can be used to identify points by their row number in a scatterplot.

We can use `==` to see which case drank exactly 9 beers. Which is the row number of the case that drank 9 beers?


```
plot(BAC ~ Beers, data=bac, pch=19)
```



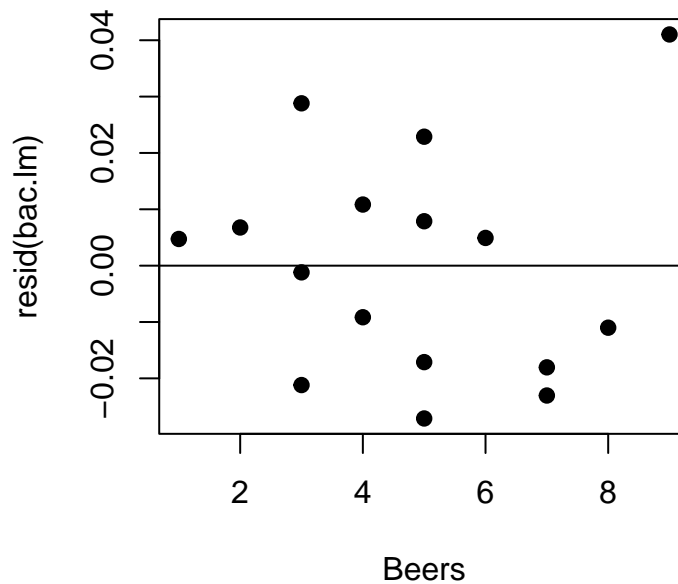
```
which(bac$Beers == 9)
```

```
[1] 3
```

Answer: Row 3.

What is the row number of the case with the most negative residual?

```
plot(resid(bac.lm) ~ Beers, data=bac, pch=19)  
abline(h=0)
```



We could eyeball the graph to see that the most negative residual is less than -0.02:

```
# which case has resid less than -0.02?
resid(bac.lm)[which(resid(bac.lm) < -0.02)]
```

```
          9          10          14
-0.02119068 -0.02711821 -0.02304573
```

But this identifies 3 cases. We also can see that the lowest residual drank 5 beers. We can add this statement to the original one using the “and” sign &:

```
# which case had resid less than -0.02 AND drank 5 beers
resid(bac.lm)[which(resid(bac.lm) < -0.02 & bac$Beers == 5)]
```

```
          10
-0.02711821
```

6.1.13 (l) Checking outlier influence

Will the regression line slope increase, decrease or stay the same if we remove case 3, the 9 beer case, from our model?

Check your answer by adding `subset = -3` to the `lm` command (this removes row 3):

```
# define a different linear model with row 3 removed
bac.lm2 <- lm(BAC ~ Beers, data=bac, subset = -3)
```

```
# Compare the two models
summary(bac.lm2)
```

```
Call:
lm(formula = BAC ~ Beers, data = bac, subset = -3)

Residuals:
    Min       1Q   Median       3Q      Max
-0.023685 -0.010068 -0.003685  0.011985  0.027208

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.481e-05  1.088e-02   0.002   0.998
Beers        1.455e-02  2.216e-03   6.568  1.8e-05 ***
---
Signif. codes:
  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01624 on 13 degrees of freedom
Multiple R-squared:  0.7684,    Adjusted R-squared:  0.7506
F-statistic: 43.14 on 1 and 13 DF,  p-value: 1.802e-05
```

```
summary(bac.lm)
```

```
Call:
lm(formula = BAC ~ Beers, data = bac)

Residuals:
    Min       1Q   Median       3Q      Max
-0.027118 -0.017350  0.001773  0.008623  0.041027

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.012701   0.012638  -1.005   0.332
Beers        0.017964   0.002402   7.480 2.97e-06 ***
---
Signif. codes:
  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 0.02044 on 14 degrees of freedom
 Multiple R-squared: 0.7998, Adjusted R-squared: 0.7855
 F-statistic: 55.94 on 1 and 14 DF, p-value: 2.969e-06

- After removing case 3, how has the slope changed? Explain the why the change occurred.

Answer: The slope drops from 0.0180 to 0.0146. Explanation given above.

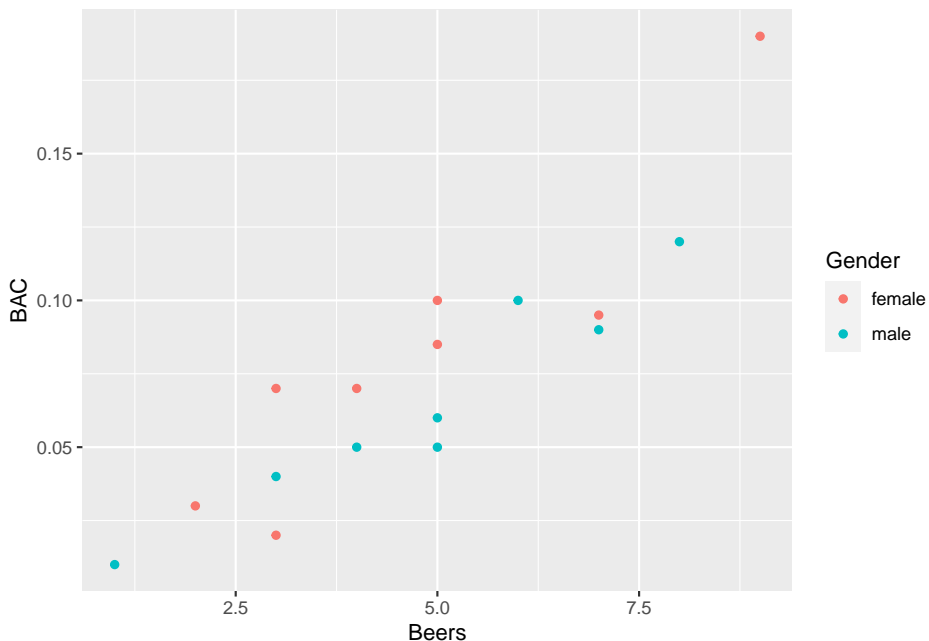
- After removing case 3, how has the R^2 changed? Explain the why the change occurred.

Answer: The R^2 decreases from 79.9% to 76.8%. This small decrease happens because case 3 actually enhances the overall linear trend and removing it results is a slight decrease to correlation and R^2 .

6.1.14 (m) Adding a categorical variable to your plot

We can create a scatterplot with plotting symbols color coded by a categorical grouping variable using `ggplot2` package. We use the `geom_point()` plot geometry to get a scatterplot with the `x`, `y`, and `color` aesthetics specified. Here we look at the BAC vs. Beers plot with `Gender` added:

```
library(ggplot2)
ggplot(bac, aes(x=Beers, y=BAC, color=Gender)) + geom_point()
```



- Are the associations similar? (form, strength, direction)

Answer: Both females and males have similar strong, positive linear associations.

6.1.15 (n) Regression lines by groups

A quick way to get the male and female regression line formulas for part (c) is to add a `subset` argument to the `lm` command:

```
bac.lm.female <- lm(BAC ~ Beers, data=bac, subset = Gender == "female")
bac.lm.female
```

Call:

```
lm(formula = BAC ~ Beers, data = bac, subset = Gender == "female")
```

Coefficients:

```
(Intercept)      Beers
   -0.01567      0.02067
```

```
# enter code for the male model
```

```
bac.lm.male <- lm(BAC ~ Beers, data=bac, subset = Gender == "male")
bac.lm.male
```

Call:

```
lm(formula = BAC ~ Beers, data = bac, subset = Gender == "male")
```

Coefficients:

```
(Intercept)      Beers
   -0.009785      0.015341
```

- What is the regression line for females? for males?

Answer: For females: $\widehat{BAC} = -0.016 + 0.021(BAC)$ and for males: $\widehat{BAC} = -0.01 + 0.015(BAC)$

- Which gender has the largest slope? What does this suggest about the relationship between number of beers and BAC for this gender?

Answer: The slope for females is slightly higher. This shows that the effect of one more beer on predicted BAC in females is larger than males (a 0.021 increase vs. a 0.015 increase).

Another way to obtain regression models by **Gender** is to split the data set in a female and male data set, then run your `lm` on these two data sets. The benefit of this method is you can then create a residuals plot for your model much easier than the quicker method above:

```
bac.female <- subset(bac, sub = Gender == "female")
lm(BAC ~ Beers, data=bac.female)
```

Call:

```
lm(formula = BAC ~ Beers, data = bac.female)
```

Coefficients:

(Intercept)	Beers
-0.01567	0.02067

```
bac.male <- subset(bac, sub = Gender == "male")
lm(BAC ~ Beers, data=bac.male)
```

Call:

```
lm(formula = BAC ~ Beers, data = bac.male)
```

Coefficients:

(Intercept)	Beers
-0.009785	0.015341

6.1.16 Mice Mass Example

The time of day in which calories are consumed can affect weight gain. At least, that appears to be true in mice. Mice normally eat all their calories at night, but when mice ate some of their calories during the day (when mice are supposed to be sleeping), they gained more weight even though all the mice ate the same total amount of calories. Here we look at the regression of body mass gain in grams, **BMGain**, against the percent of calories eaten during the day, **DayPct** for a study involving 27 mice. The R commands needed to answer the questions below are:

```
mice <- read.csv("http://math.carleton.edu/Stats215/Textbook/LightatNight4Weeks.csv")

plot(BMGain ~ DayPct, data=mice, pch=19)
mice.lm <- lm(BMGain ~ DayPct, data=mice)
mice.lm
```

Call:

```
lm(formula = BMGain ~ DayPct, data = mice)
```

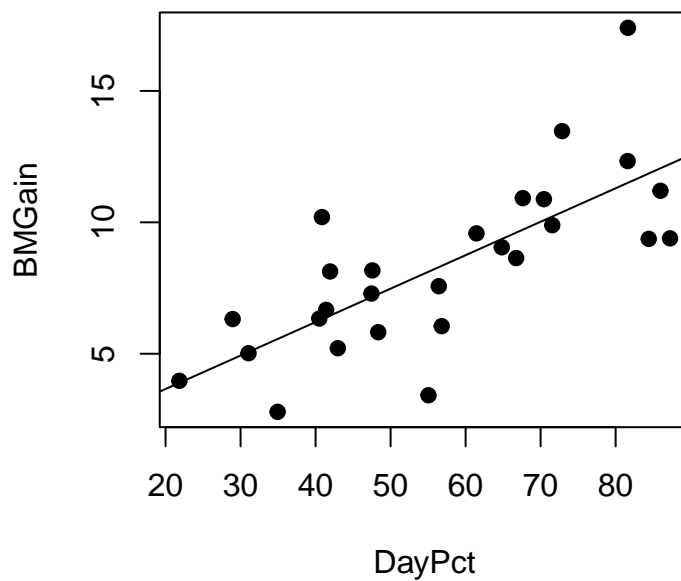
Coefficients:

(Intercept)	DayPct
1.1128	0.1273

```
cor(mice$BMGain, mice$DayPct)
```

```
[1] 0.7398623
```

```
abline(mice.lm) # adds regression line to previously created scatterplot
```



6.1.17 (a) What are the coordinates (roughly) of the case with the largest positive residual?

```
mice[which(resid(mice.lm) == max(resid(mice.lm))),]
```

```

      Light BMGain Corticosterone DayPct Consumption
25    LL   17.4           66.679 81.636          7.177
      GlucoseInt  GTT15  GTT120 Activity
25           Yes 435.644 405.941      6702

```

Answer: The case with the largest residual is located at about 80% calories and 17g body mass gain. We can find which row this corresponds to using the `which` command shown below.

6.1.18 (b) What are the coordinates (roughly) of the case with the most negative residual?

```
mice[which(resid(mice.lm) == min(resid(mice.lm))),]
```

```

      Light BMGain Corticosterone DayPct Consumption
10    DM   3.42           208.26 55.051          3.857
      GlucoseInt  GTT15  GTT120 Activity
10           No 271.717 148.485      1084

```

Answer: The case with the most negative residual is located at about 55% calories and 3g body mass gain. We can find which row this corresponds to using the `which` command shown below. The code below also highlights the cases in (a) with a circle and (b) with a square.

6.1.19 (c) What is the predicted body mass gain for a mouse that eats 50% of its calories during the day?

$$\widehat{BMGain} = 1.1128 + 0.1273(50) = 7.48$$

```
1.1128 + .1273*50
```

```
[1] 7.4778
```

Answer: A mouse that eats 50% of its calories during the day is predicted to gain 7.48 grams.

6.1.20 (d) Find the residual for the mouse who ate 48.3% of its calories during the day and gained 5.82 grams.

Answer: We first find the predicted body mass gain:

$$\widehat{BMGain} = 1.1128 + 0.1273(48.3) = 7.26$$

The residual is then:

$$Residual = BMGain - \widehat{BMGain} = 5.82 - 7.26 = -1.44.$$

```
1.1128 + .1273*48.3
```

```
[1] 7.26139
```

```
5.82 - (1.1128 + .1273*48.3)
```

```
[1] -1.44139
```

6.1.21 (e) Interpret the slope of the regression line in context.

Answer: The slope is 0.1273. When a mouse eats one more percent of its calories during the day, its predicted body mass gain goes up by 0.1273 grams.

6.1.22 (f) Interpret the intercept of the line in context, if it makes sense to do so.

Answer: The intercept is 1.1128. A mouse who eats 0% of its calories during the day (and all of them at night when a mouse normally eats all its food) is predicted to gain 1.11 grams. But this would be **extrapolation** because the range of observed percents is, roughly, 20-90. It does not make sense to interpret the intercept in this context.

6.1.23 (g) Use the correlation value to compute R^2 , then interpret (in context) the R^2 value for this model.

```
r <- 0.7398623
r^2
```

```
[1] 0.5473962
```

- 6.1.24 (h)** Get the value of R^2 from the regression output, then interpret (in context) the R^2 value for this model.

```
summary(mice.lm)
```

Call:

```
lm(formula = BMGain ~ DayPct, data = mice)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-4.6990	-1.1694	0.0728	0.9174	5.8975

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.11280	1.38211	0.805	0.428
DayPct	0.12727	0.02315	5.499	1.03e-05 ***

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.231 on 25 degrees of freedom

Multiple R-squared: 0.5474, Adjusted R-squared: 0.5293

F-statistic: 30.24 on 1 and 25 DF, p-value: 1.032e-05

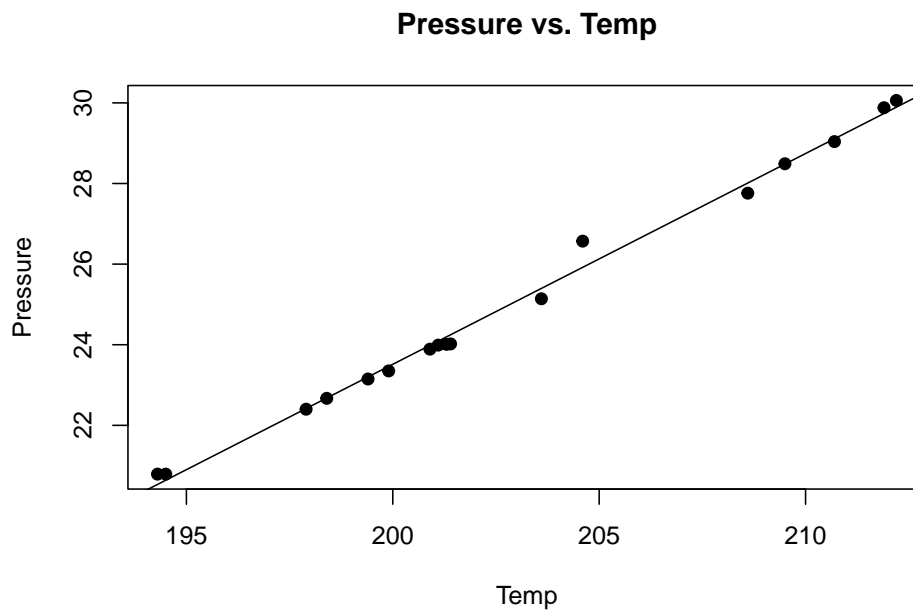
Answer: From Multiple R-squared, we get $R^2 = 0.547$. The percent of calories that a mouse eats during the day explains about 55% of the variability in weight gain for this study.

6.1.25 Forbes Example

In the mid 1800s, James D. Forbes conducted a experiments designed to determine if the atmospheric pressure at a given location can just be determined by the boiling temp of water at that location.

- 6.1.26 (a)** Fit the linear regression of Pressure on Temp:

```
forbes <- read.csv("http://math.carleton.edu/kstclair/data/forbes.csv")
plot(Pressure ~ Temp, data=forbes, pch=19, main = "Pressure vs. Temp")
forbes.lm <- lm(Pressure ~ Temp, data=forbes)
abline(forbes.lm)
```



```
summary(forbes.lm)
```

Call:

```
lm(formula = Pressure ~ Temp, data = forbes)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.25717	-0.11246	-0.05102	0.14283	0.64994

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-81.06373	2.05182	-39.51	<2e-16 ***
Temp	0.52289	0.01011	51.74	<2e-16 ***

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2328 on 15 degrees of freedom

Multiple R-squared: 0.9944, Adjusted R-squared: 0.9941
 F-statistic: 2677 on 1 and 15 DF, p-value: < 2.2e-16

- Describe the relationship between pressure and temp (strength, form, direction).

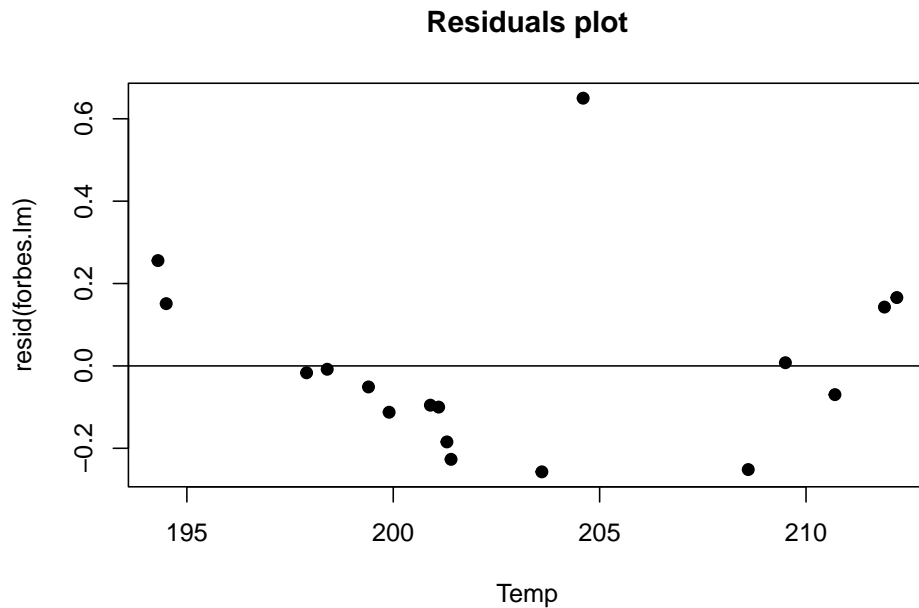
Answer: This is a strong, positive relationship that looks linear.

- Interpret the value of R^2

Answer: About 99.4% of the variation observed in pressure can be explained by the boiling point temps.

6.1.27 (b) Check the residuals plot

```
plot(resid(forbes.lm) ~ Temp, data=forbes, pch=19, main = "Residuals plot")
abline(h=0)
```



- Is the relationship between pressure and temp linear?

Answer: No! There is curvature, which means the linear model is systematically underestimating pressure at low and high temps and overestimating pressure at mid-range temps.

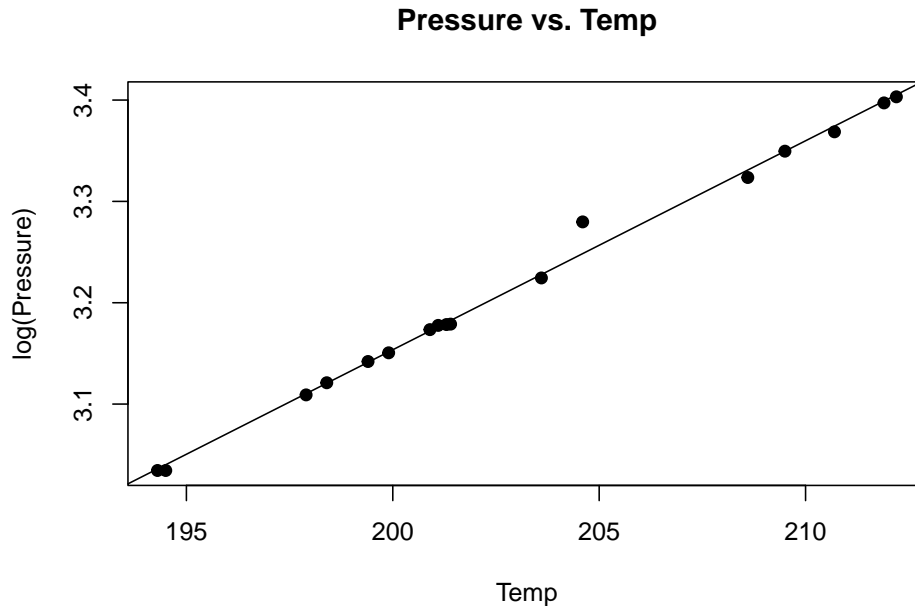
- Does the residual plot highlight an unusual case? Explain.

Answer: Yes, there is one case that has an unusually high pressure value given its temp.

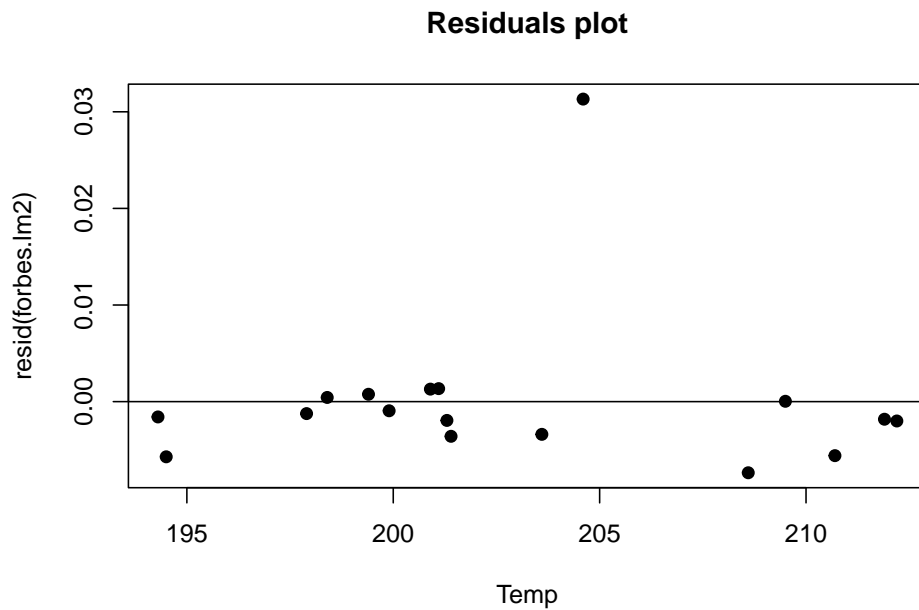
6.1.28 (c) “Fixing” the model

A linear model can be used with this data if we **transform** the response variable to the logarithmic scale. Here $\log(y)$ gives the natural log of the variable y .

```
plot(log(Pressure) ~ Temp, data=forbes, pch=19, main = "Pressure vs. Temp")
forbes.lm2 <- lm(log(Pressure) ~ Temp, data=forbes)
abline(forbes.lm2)
```



```
plot(resid(forbes.lm2) ~ Temp, data=forbes, pch=19, main = "Residuals plot")
abline(h=0)
```



- Has the curvature in the scatterplot and residuals plots been reduced by logging the variables?

Answer: Yes, there is less curvature

- Has the outlier been eliminated by logging the variables?

Answer: No, the outlier is still present.

6.1.29 (d) Removing bad measurement

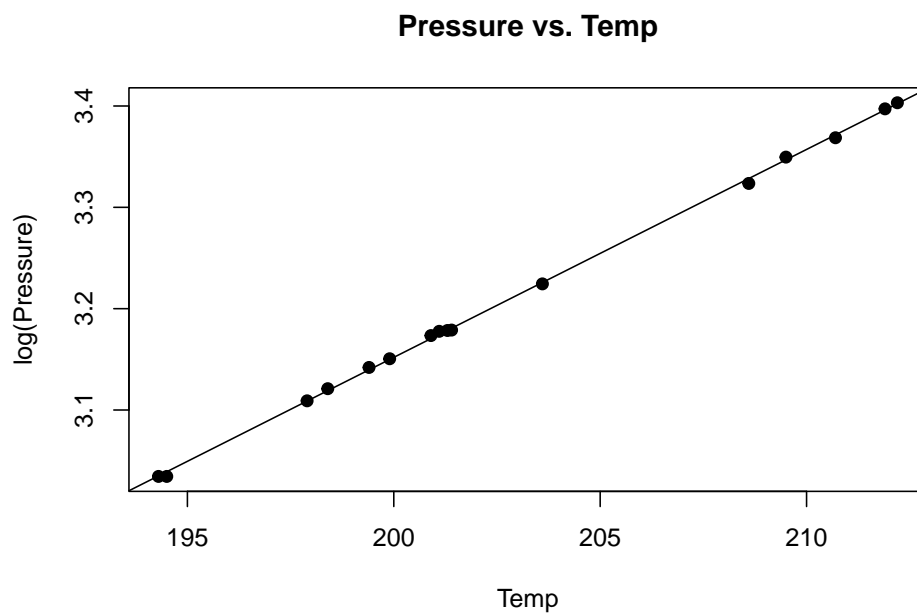
Identify which case has the large residual value around 0.03.

```
resid(forbes.lm2)[which(resid(forbes.lm2) > 0.02)]
```

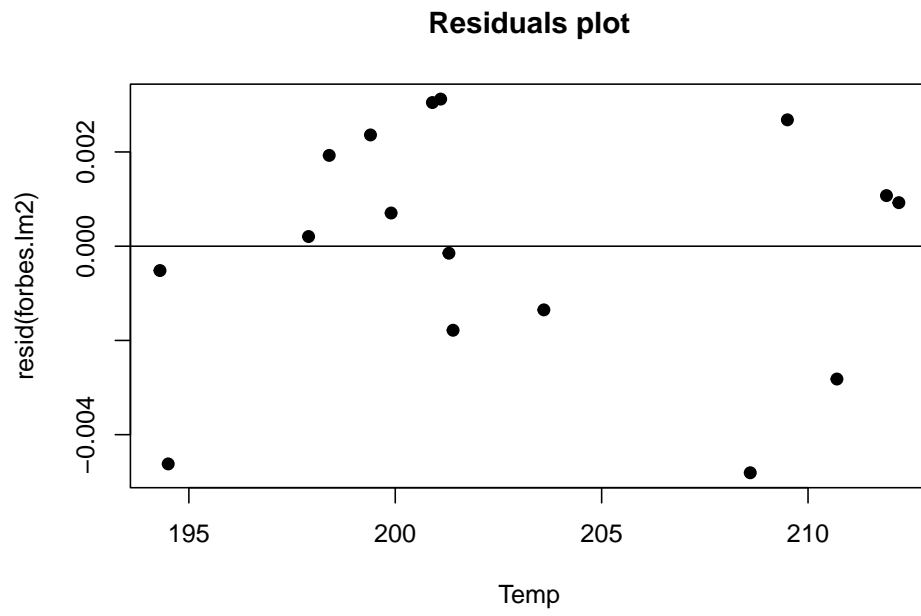
```
12
0.03131388
```

Repeat part (c) but this time remove the case you identified. The easiest way to do this is to create a new version of the data with row 12 removed:

```
forbes2 <- forbes[-12, ]  
plot(log(Pressure) ~ Temp, data=forbes2, pch=19, main = "Pressure vs. Temp")  
forbes.lm2 <- lm(log(Pressure) ~ Temp, data=forbes2)  
abline(forbes.lm2)
```



```
plot(resid(forbes.lm2) ~ Temp, data=forbes2, pch=19, main = "Residuals plot")  
abline(h=0)
```



Chapter 7

Class Activity 7

7.1 Your Turn 1

7.2 Few Instructions

The default setting in Rstudio when you are running chunks is that the “output” (numbers, graphs) are shown **inline** within the Markdown Rmd. If you prefer to have your plots appear on the right of the console and not below the chunk, then change the settings as follows:

1. Select Tools > Global Options.
2. Click the R Markdown section and uncheck (if needed) the option Show output inline for all R Markdown documents.
3. Click OK.

Now try running R chunks in the .Rmd file to see the difference. You can recheck this box if you prefer the default setting.

7.3 Parameters and Statistics

Here are some notations that will be useful for you. Look for the codes to produce this in the associated Rmd file.

	Parameter	Statistic
Mean	μ	\bar{x}
Proportion	p	\hat{p}

	Parameter	Statistic
Std. Dev.	σ	s
Correlation	ρ	r
Slope	β	b

7.4 Example 1: Parameters and Statistics

For each of the following, state whether the quantity described is a parameter or a statistic, and give the correct notation.

7.4.1 a. Average household income for all houses in the US, using data from the US census

Answer: This is a parameter since the mean is for all houses in the US, and the notation is μ .

7.4.2 b. The proportion of all residents in a county who voted in the last presidential election.

Answer: This is a parameter since we have information on all the residents, and the notation is p .

7.4.3 c. The difference in proportion who have ever smoked cigarettes, between a sample of 500 people who are 60 years old and a sample of 200 people who are 25 years old.

Answer: We use statistics since the proportions are from samples. The notation for the difference in sample proportions is $\hat{p}_1 - \hat{p}_2$

7.4.4 d. The correlation between weight and height for 5-year old kids.

Answer: If we are looking at all 5-year old kids it is a parameter, and the notation for correlation is ρ .

7.4.5 e. The mean number of extracurricular activities from a random sample of 50 students at your school.

Answer: This is a statistic since the mean is from a sample, and the notation is μ .

7.5 Example 2: Using Search Engines on the Internet

A 2012 survey of a random sample of 2253 US adults found that 1,329 of them reported using a search engine (such as Google) every day to find information on the Internet.

7.5.1 a). Find the relevant proportion and give the correct notation with it.

Answer: $\hat{p} = 1329/2253$

```
p.hat <- 1329/2253
p.hat
[1] 0.5898802
```

7.5.2 b). Is your answer to part (a) a parameter or a statistic?

Answer: Statistic

7.5.3 c). Give notation for and define the population parameter that we estimate using the result of part (a).

Answer: p = the proportion of all US adults that would report that they use an Internet search engine every day

7.6 Example 3: Simulation of a Sample Proportion

According to a PEW survey, 66% of U.S. adult citizens casted a ballot in the 2020 election. Suppose we take a random sample of $n = 100$ eligible U.S. voters and computed the sample proportion who voted.

```
# Call the library
library(ggplot2)
```

```
# Define parameters
pop.prop <- .66 # Population proportion
n.size <- 100 # sample size
```

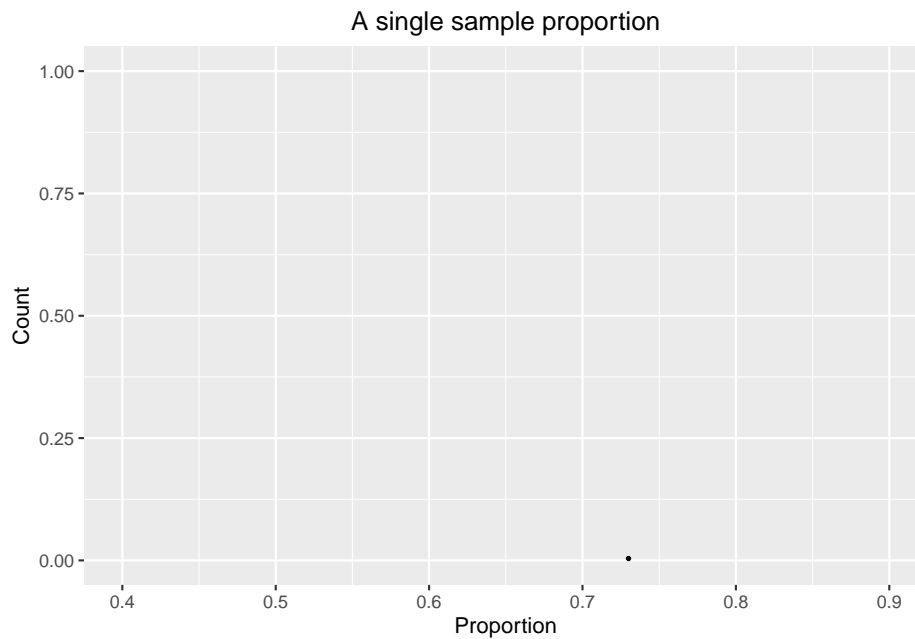
7.6.1 (a) Generate a random sample of size $n = 100$ and plot its sample proportion.

```
# Generate 1 sample
sample1 <- rbinom(n = 1, size = n.size, p = pop.prop) # R simulates the samples
sample.prop1 <- sample1/n.size # Proportion = No. of Success / Sample Size
```

```
# define a data frame
mydata <- data.frame(x = sample.prop1)

# Plot a dot plot of the sample proportion
ggplot(mydata, aes(x = sample.prop1)) +
  geom_dotplot(dotsize=0.25, stackratio=0.75, binwidth=0.01) +
```

```
ggtitle("A single sample proportion") + xlab("Proportion") + ylab("Count")+
scale_x_continuous(limits = c(0.4, 0.9))+
theme(plot.title = element_text(hjust = 0.5))
```

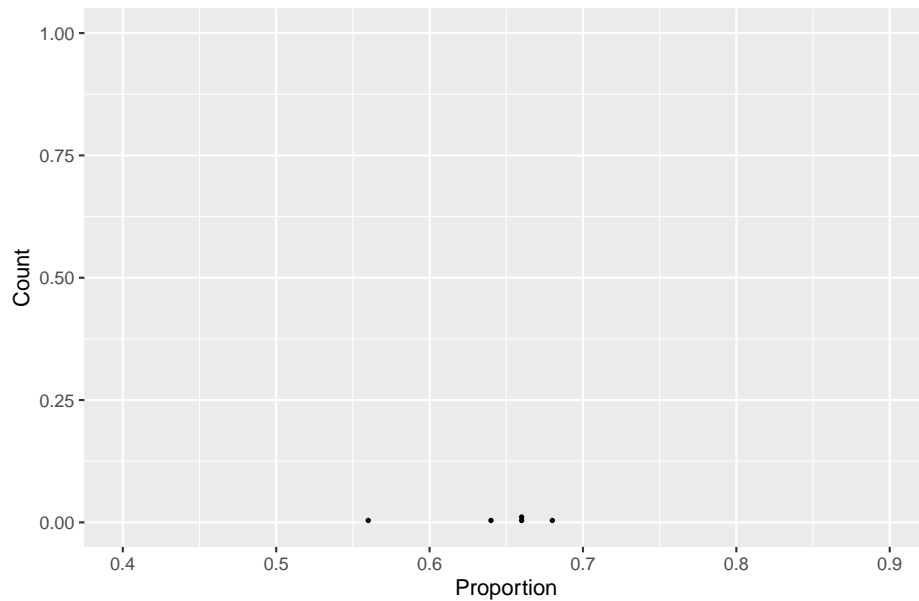


7.6.2 (b) Generate 5 random samples of size $n = 100$ and plot the sample proportions.

```
# generate 5 random samples of size 100
sample5 <- rbinom(n = 5, size = n.size, p = pop.prop)
sample.prop5 <- sample5/n.size

data <- data.frame(x = sample.prop5)

ggplot(data, aes(x = sample.prop5)) +
  geom_dotplot(dotsize=0.25, stackratio=0.9, binwidth=0.01 )+
  ggtitle("") + xlab("Proportion") + ylab("Count")+
  scale_x_continuous(limits = c(0.4, 0.9))+
  theme(plot.title = element_text(hjust = 0.5))
```

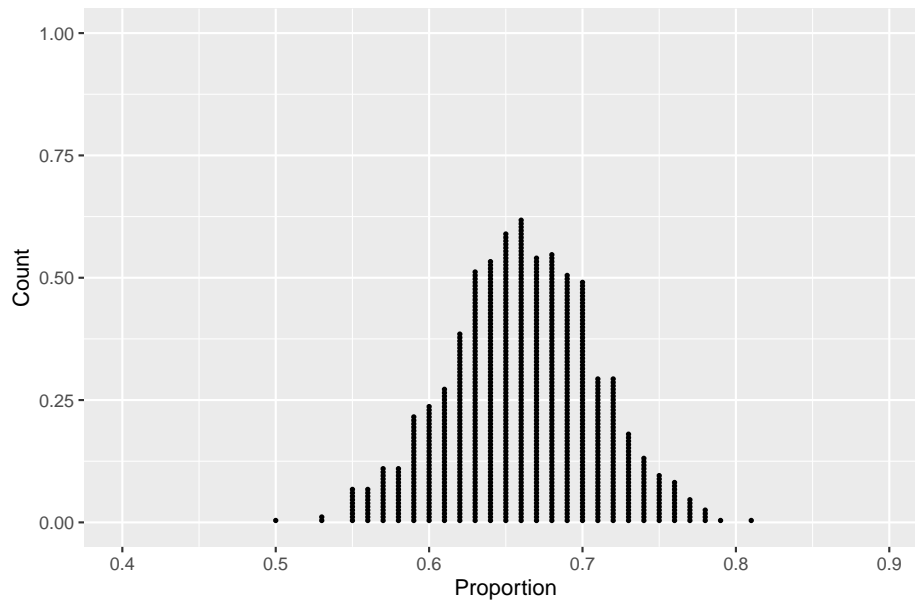


7.6.3 (c) Generate 1000 random samples of size $n = 100$ and plot the sample proportions.

```
# Generate 1000 samples
sample1000 <- rbinom(n = 1000, size = n.size, p = pop.prop)
sample.prop1000 <- sample1000/n.size

data <- data.frame(x = sample.prop1000)

ggplot(data, aes(x = sample.prop1000)) +
  geom_dotplot(dotsize=0.25, method = "histodot", stackratio=0.9, binwidth=0.01) +
  ggtitle("") + xlab("Proportion") + ylab("Count") +
  scale_x_continuous(limits = c(0.4, 0.9))+
  theme(plot.title = element_text(hjust = 0.5))
```



Question: What does each dot represent?

Answer: One sample proportion from a sample of $n=100$ eligible voters.

Question: What is the shape of your sampling distribution?

Answer: Roughly symmetric.

Question: Where is your distribution centered?

Answer: About 0.66, which is the population proportion.

Question: The distribution should be centered at the population proportion. Verify that the distribution is centered around the population proportion, $p = 0.66$.

Answer:

```
# r-code
mean(sample.prop1000)
[1] 0.66109
```

Question: What is the standard deviation of this distribution? (Hint: use the 95% rule.)

Answer: About 0.03, it looks like most sample proportions are between 0.55 to 0.75 so 2 standard deviations is about 0.10. This makes the SD about 0.05.

Question: The standard deviation of sampling distribution has a separate name. It is called the **Standard Error**. Verify the standard deviation of this distribution using R-code.

Answer:

```
# r-code
sd(sample.prop1000)
[1] 0.04705025
```

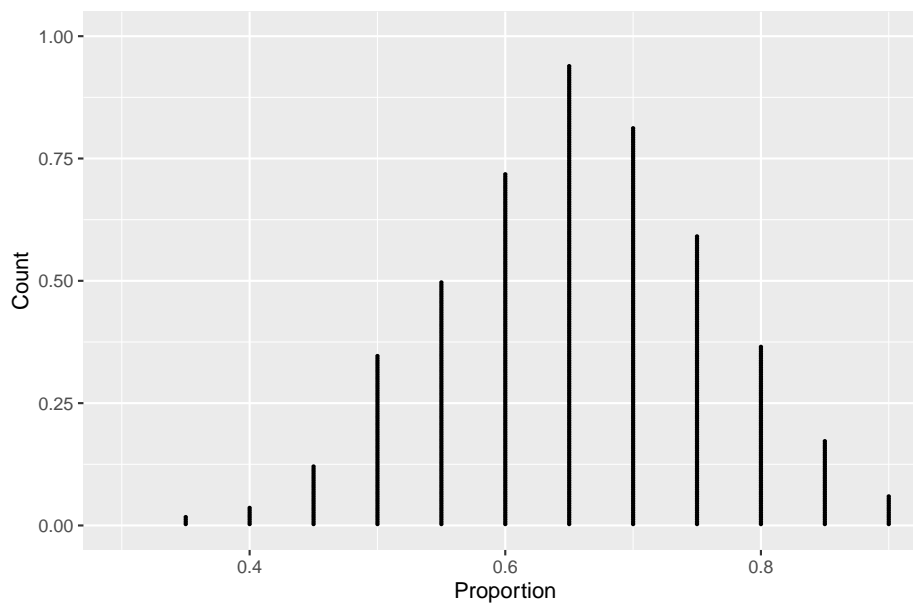
7.6.4 (d) Repeat part(c) with sample size 20 instead of 100. Generate 1000 samples.

```
# Generate 1000 samples
n.size <- 20

sample1000 <- rbinom(n = 1000, size = n.size, p = pop.prop)
sample.prop1000 <- sample1000/n.size

data <- data.frame(x = sample.prop1000)

ggplot(data, aes(x = sample.prop1000)) +
  geom_dotplot(dotsize=0.225, method = "histodot", stackratio=0.8, binwidth=0.01) +
  ggtitle("") + xlab("Proportion") + ylab("Count") +
  scale_x_continuous(limits = c(0.3, 0.9))+
  theme(plot.title = element_text(hjust = 0.5))
```



Question: How has the sampling distribution changed? (Shape? Center? Variability?)

Answer: The shape is slightly left skewed, still centered at 0.66 but with more variability than before (SD of about 0.10). This distribution is more discrete looking because there are just a few sample proportions possible with $n=20$ (e.g. 20/20, 19/20, 18/20, etc).

```
mean(sample.prop1000)
[1] 0.65645
sd(sample.prop1000)
[1] 0.1043877
```

7.6.5 (d) Now suppose the population proportion is $p = 0.90$ instead of $p = 0.66$ in part (e). Keep $n.size=20$.

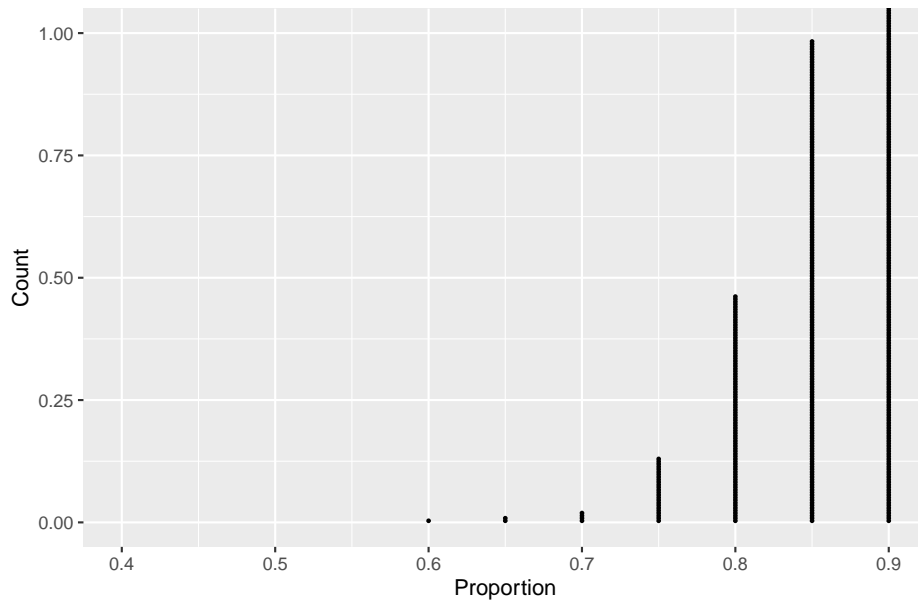
```
# Generate 1000 samples

pop.prop <- 0.90
n.size <- 20
n.size <- 20

sample1000 <- rbinom(n = 1000, size = n.size, p = pop.prop)
sample.prop1000 <- sample1000/n.size

data <- data.frame(x = sample.prop1000)

ggplot(data, aes(x = sample.prop1000)) +
  geom_dotplot(dotsize=0.21, method = "histodot", stackratio=0.8, binwidth=0.01) +
  ggtitle("") + xlab("Proportion") + ylab("Count") +
  scale_x_continuous(limits = c(0.4, 0.9))+
  theme(plot.title = element_text(hjust = 0.5))
```



Question: How has the sampling distribution changed? (Shape? Center? Variability?)

Answer: The shape is much more left skewed than when $p=0.66$. Center is around 0.90 and SD is around 0.07. Note that increasing the population proportion closer to 1 results in a decrease in the SD because most samples give proportion near 1.

```
mean(sample.prop1000)
[1] 0.9035
sd(sample.prop1000)
[1] 0.06528409
```

7.7 Example 4: Simulation for a Sample Mean

We'll look at sampling movies from the population of 134 Hollywood movies made in 2011 and measuring their budget (millions of dollars).

```
# import dataset
library(Lock5Data)
movies <- HollywoodMovies2011
```

7.7.1 (a) What is the population mean of the Budget?

```
# r-code
mean(movies$Budget, na.rm = TRUE)
[1] 53.48134
```

7.7.2 (b) Generate a random sample of size $n = 10$ and plot the sample proportion.

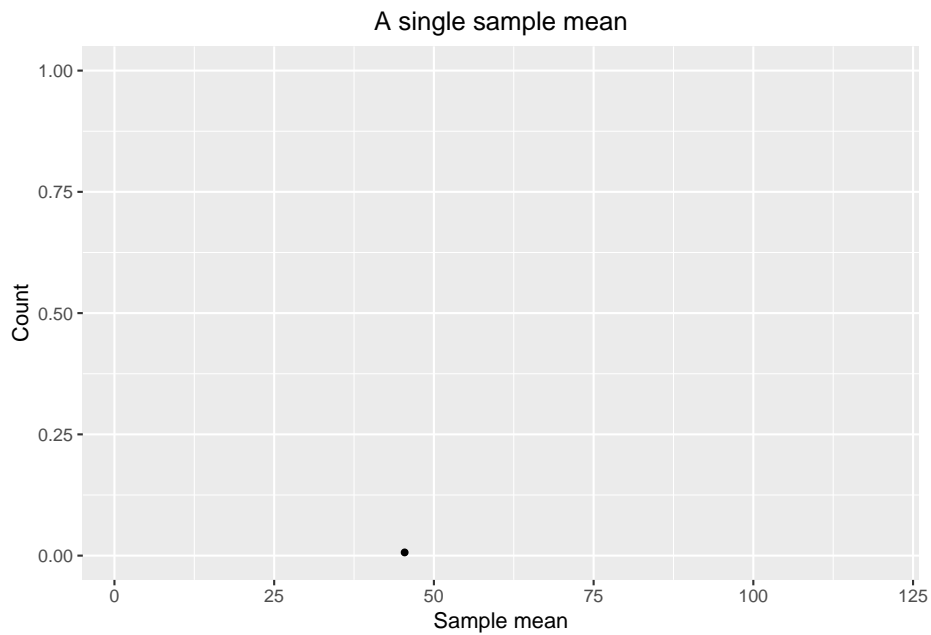
```
# define a data frame
n.size <- 10

Budget <- movies$Budget[!is.na(movies$Budget)] # remove NAs

sample1 <- sample(Budget, size = n.size)
sample.mean1 <- mean(sample1)

mydata <- data.frame(x = sample.mean1)

# Plot a dot plot of the sample proportion
ggplot(mydata, aes(x = sample.mean1)) +
  geom_dotplot(dotsize=1, stackratio=0.9, binwidth=1) +
  ggtitle("A single sample mean") + xlab("Sample mean") + ylab("Count")+
  scale_x_continuous(limits = c(1,120))+
  theme(plot.title = element_text(hjust = 0.5))
```



7.7.3 (c) Generate 5 random samples of size $n = 10$ and plot the sample means.

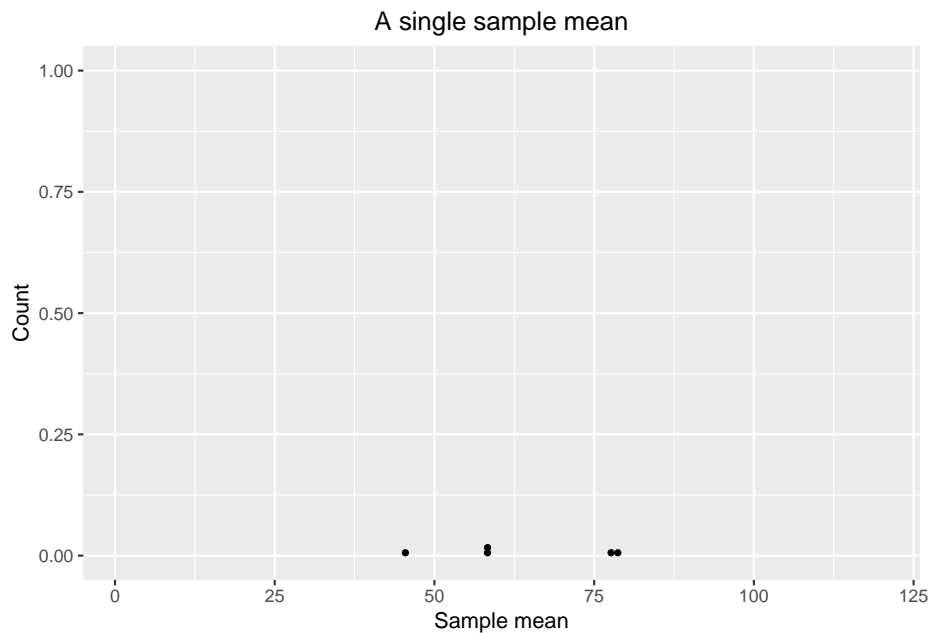
```
n.size <- 10
n.rep <- 5

Budget <- movies$Budget[!is.na(movies$Budget)] # remove NAs

sample5 <- lapply(1:5, function(i) sample(Budget, size = n.size))
sample.mean5 <- lapply(sample5, function(x) mean(x))
sample.mean5 <- unlist(sample.mean5)

mydata <- data.frame(x = sample.mean5)

# Plot a dot plot of the sample proportion
ggplot(mydata, aes(x = sample.mean5)) +
  geom_dotplot(dotsize=0.9, stackratio=0.9, binwidth=1) +
  ggtitle("A single sample mean") + xlab("Sample mean") + ylab("Count")+
  scale_x_continuous(limits = c(1,120))+
  theme(plot.title = element_text(hjust = 0.5))
```



7.7.4 (d) Generate 1000 random samples of size $n = 10$ and plot the sample means.

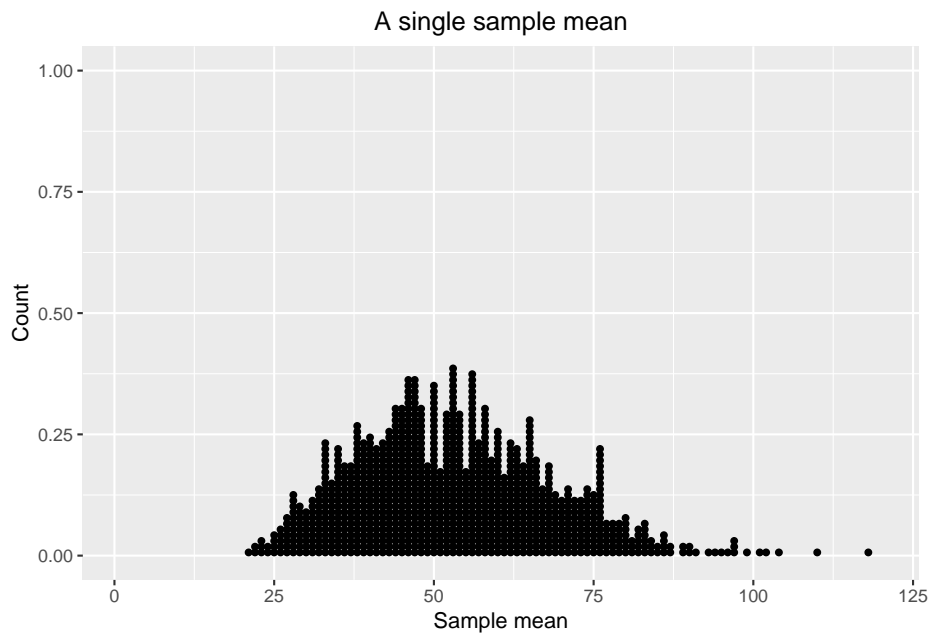
```
# Generate 1000 samples
n.size <- 10
n.rep <- 1000

Budget <- movies$Budget[!is.na(movies$Budget)] # remove NAs

sample1000 <- lapply(1:n.rep, function(i) sample(Budget, size = n.size))
sample.mean1000 <- lapply(sample1000, function(x) mean(x))
sample.mean1000 <- unlist(sample.mean1000)

mydata <- data.frame(x = sample.mean1000)

# Plot a dot plot of the sample proportion
ggplot(mydata, aes(x = sample.mean1000)) +
  geom_dotplot(dotsize=1, method = "histodot", stackratio=0.9, binwidth=1) +
  ggtitle("A single sample mean") + xlab("Sample mean") + ylab("Count")+
  scale_x_continuous(limits = c(1,120))+
  theme(plot.title = element_text(hjust = 0.5))
```



Question: What does each dot represent?

Answer: A sample mean budget from a sample of $n=10$

Question: What is the shape of your sampling distribution?

Answer: Slightly right skewed.

Question: Where is your distribution centered?

Answer: About \$53 million, which is the population mean budget.

```
mean(movies$Budget, na.rm = TRUE)
[1] 53.48134
```

Question: The distribution should be centered at the population mean. Verify that the distribution is centered around the population mean, $\mu = 53.48$.

Answer: It is very close to the population mean.

```
# r-code
mean(sample.mean1000)
[1] 53.21977
```

Question: What is the standard deviation of this distribution? (Hint: use the 95% rule.)

Answer: About 15 million.

Question: The standard deviation of sampling distribution has a separate name. It is called the **Standard Error**. Verify the standard deviation of this distribution using R-code.

Answer: It is 14.80 million quite close to our previous informed guess.

```
# r-code
sd(sample.mean1000)
[1] 15.37391
```

7.7.5 (e) Repeat part(d) with sample size 50 instead of 10. Generate 1000 samples.

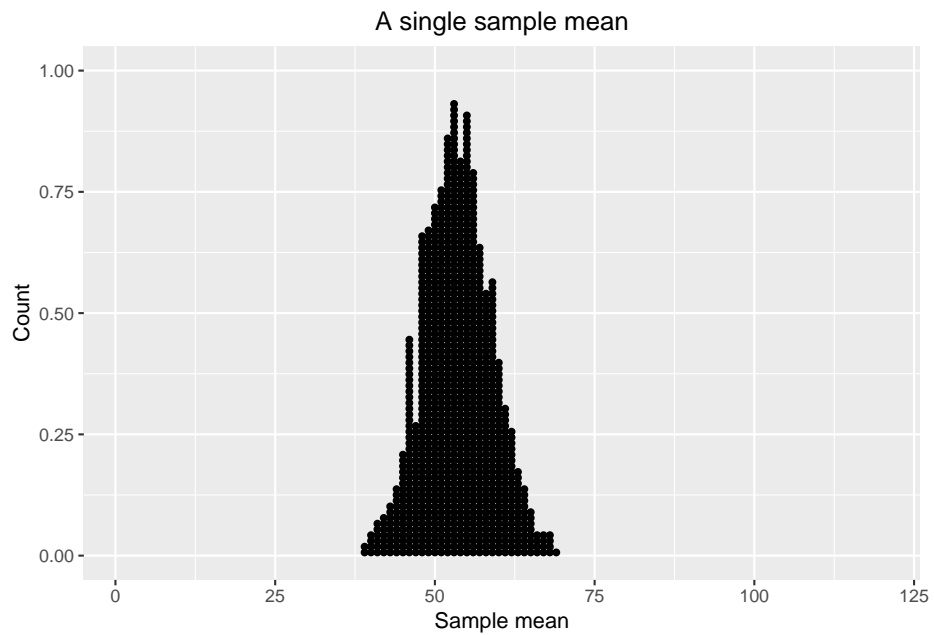
```
# Generate 1000 samples
n.size <- 50
n.rep <- 1000

Budget <- movies$Budget[!is.na(movies$Budget)] # remove NAs

sample1000 <- lapply(1:n.rep, function(i) sample(Budget, size = n.size))
sample.mean1000 <- lapply(sample1000, function(x) mean(x))
sample.mean1000 <- unlist(sample.mean1000)

mydata <- data.frame(x = sample.mean1000)

# Plot a dot plot of the sample proportion
ggplot(mydata, aes(x = sample.mean1000)) +
  geom_dotplot(dotsize=1, method = "histodot", stackratio=0.9, binwidth=1) +
  ggtitle("A single sample mean") + xlab("Sample mean") + ylab("Count")+
  scale_x_continuous(limits = c(1,120))+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(plot.title = element_text(hjust = 0.5))
```

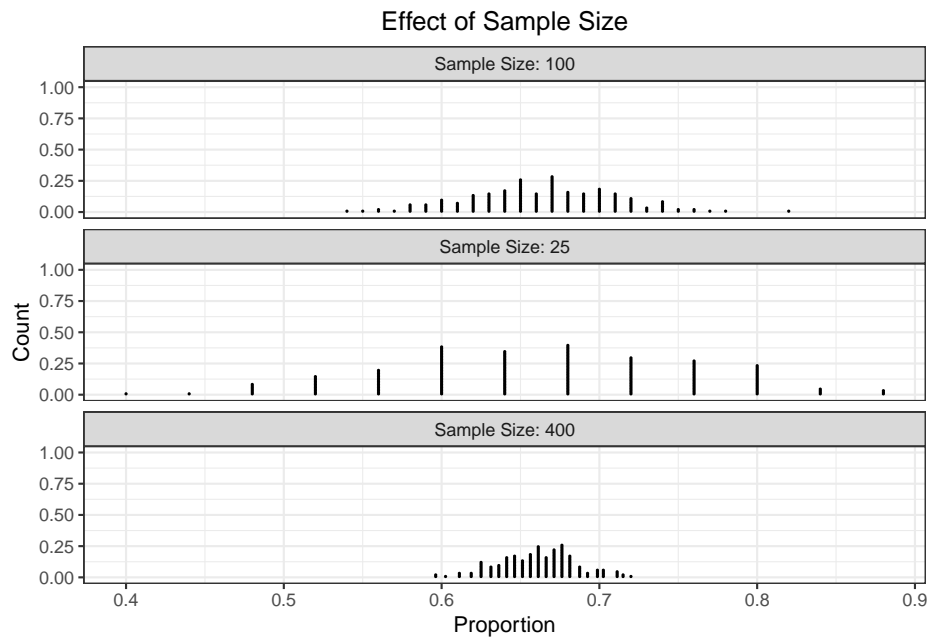


Question Is this sampling distribution more or less symmetric compared to the distribution when $n = 10$?

Answer The distribution is more symmetric with $n=50$ than when $n=10$.

7.8 Example 5: Effect of sample size

Let's investigate the effect of sample size in the sampling distribution using the same setting as in Exercise 1 with $p = 0.66$. The following are three sampling distributions corresponding to different sample sizes.



Question What happens if we increase the sample size?

Answer When we increase the sample size, the variability of the sampling distribution becomes smaller.

Question Estimate the standard error of each and verify your answer to the previous question.

Answer The standard errors are

```
sd(data.size.25$x)
[1] 0.0957207
sd(data.size.100$x)
[1] 0.04704213
sd(data.size.400$x)
[1] 0.02386724
```

As the sample size increases, the variability as measured by the standard error of the sampling distribution does indeed decrease.

7.9 Example 6: Bootstrap Sampling

```
# Movies Example Again!
```

```

Budget <- movies$Budget[!is.na(movies$Budget)]

# Bootstrap samples
n.size <- 10
boot.sample1 <- sample(Budget, 10, replace = TRUE) # sampling with replacement

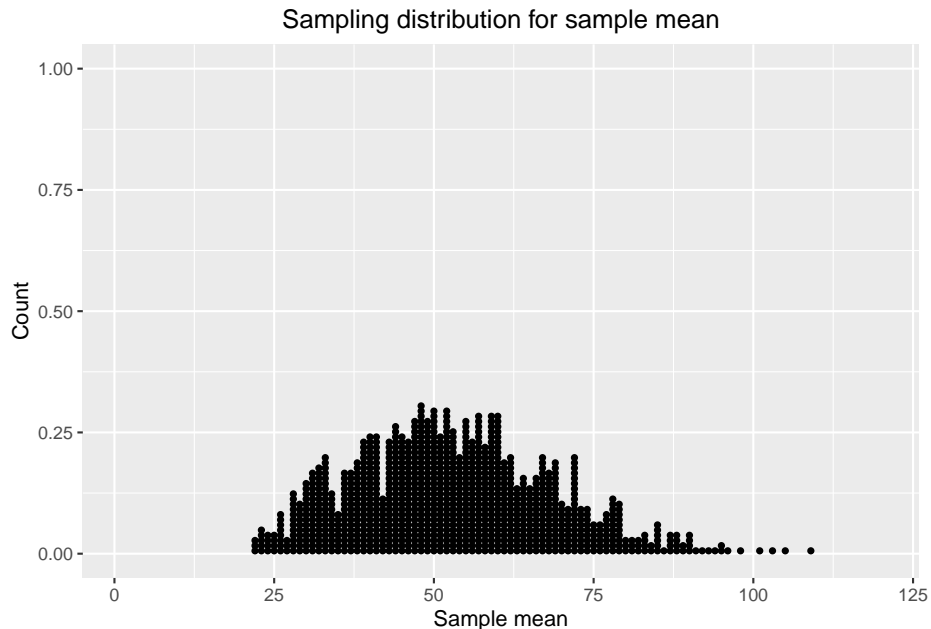
n.rep <- 1000
boot.sample1000 <- lapply(1:n.rep, function(i) sample(Budget, 10, replace = TRUE))
boot.samplemean1000 <- lapply(boot.sample1000, function(x) mean(x))
boot.samplemean1000 <- unlist(boot.samplemean1000)

# Plot the bootstrap distribution

mydata <- data.frame(x = boot.samplemean1000)

# Plot a dot plot of the sample proportion
ggplot(mydata, aes(x = boot.samplemean1000)) +
  geom_dotplot(dotsize=0.9, stackratio=0.9, binwidth=1, method = "histodot") +
  ggtitle("Sampling distribution for sample mean") + xlab("Sample mean") + ylab("Count")
  scale_x_continuous(limits = c(1,120))+
  theme(plot.title = element_text(hjust = 0.5))

```



- (a) Compare the center/spread/shape of the bootstrap distribution to the distribution computed in Ex. 4 (d). Answer all the questions in Ex. 4(d).

Answer: The shape/center and variability of this bootstrap distribution is very similar to that of Ex 4 (d)

```
mean(mydata$x)
[1] 52.99095
sd(mydata$x)
[1] 15.47923
```


Chapter 8

(PART*) Basics R

Chapter 9

What is R?

R is a free and open source statistical programming language that facilitates statistical computation. There are a myriad of application that can be done in R, thanks to a huge online support community and dedicated packages. However, R has no graphical user interface and it has to be run by typing commands into a text interface.

9.1 What is RStudio?

RStudio provides graphical interface to R! You can think of RStudio as a graphical front-end to R that that provides extra functionality. The use of the R programming language with the RStudio interface is an essential component of this course.

9.2 R Studio Server

The quickest way to get started is to go to <https://maize.mathcs.carleton.edu>, which opens an R Studio window in your web browser. Once logged in, I recommend that you do the following:

- Step 1: Create a folder for this course where you can save all of your work. In the Files window, click on New Folder.
- Step 2: Click on Tools -> Global Options -> R Markdown. Then uncheck the box that says “Show output inline...”

(It is also possible to download RStudio on your own laptop. Instructions may be found at the end of this document.)

9.3 R Markdown Basics

An R Markdown file (.Rmd file) combines R commands and written analyses, which are ‘knit’ together into an HTML, PDF, or Microsoft Word document.

An R Markdown file contains three essential elements:

- Header: The header (top) of the file contains information like the document title, author, date and your preferred output format (pdf_document, word_document, or html_document).
- Written analysis: You write up your analysis after the header and embed R code where needed. The online help below shows ways to add formatting details like bold words, lists, section labels, etc to your final pdf/word/html document. For example, adding ****** before and after a word will bold that word in your compiled document.
- R chunks: R chunks contain the R commands that you want evaluated. You embed these chunks within your written analysis and they are evaluated when you compile the document.

9.3.1 R Markdown example:

- Simple R Markdown example
 - compiled pdf

The following handouts, written by Prof Katie St Clair, contain useful information for making the figures and tables in your compiled documents look nice:

- Graph Formatting: Markdown .Rmd file and pdf
- Table Formatting: Markdown .Rmd file and pdf

9.4 Installing R/RStudio (not needed if you are using the maize server)

- Download the latest version of R:
 - Windows: <http://cran.r-project.org/bin/windows/base/>
 - Mac: <http://cran.r-project.org/bin/macosx/>
- Download the free Rstudio desktop version (Windows or Mac): <https://www.rstudio.com/products/rstudio/download/>

Use the default download and install options for each.

9.5 Install LaTeX (for knitting R Markdown documents to PDF):

If you want to compile R Markdown to .pdf files, you also need a LaTeX distribution (Note: this is not necessary if you choose to compile as a Word document.) Click instructions for Windows or instructions for Mac, depending on your operating system to complete the installation.

9.6 Updating R/RStudio (not needed if you are using the maize server)

If you have used a local version of R/RStudio before and it is still installed on your machine, then you should make sure that you have the most recent versions of each program.

- To check your version of R, run the command `getRversion()` and compare your version to the newest version posted on <https://cran.r-project.org/>. If you need an update, then install the newer version using the installation directions above.
- In RStudio, check for updates with the menu option **Help > Check for updates**. Follow directions if an update is needed.

9.7 Instructions

If using Rstudio on your computer, using the **File>Open File** menu to find and open this .Rmd file.

If using Maize Rstudio from your browser:

- In the Files tab, select **Upload** and **Choose File** to find the .Rmd that you downloaded. Click *OK* to upload to your course folder/location in the maize server account.
- Click on the .Rmd file in the appropriate folder to open the file.

Extra notes:

- You can run a line of code by placing your cursor in the line of code and clicking **Run Selected Line(s)**

- You can run an entire chunk by clicking the green triangle on the right side of the code chunk.
- After each small edit or code addition, **Knit** your Markdown. If you wait until the end to Knit, it will be harder to find errors in your work.
- Format output type: You can use any of pdf_document, html_document type, or word_document type.
- **Maize users:** You may also need to allow for “pop-up” in your web browser when knitting documents.

Chapter 10

R Markdown

This is a R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

You can use asterisk mark to provide emphasis, such as **italics** or ****bold****.

You can create lists with a dash:

```
- Item 1
- Item 2
- Item 3
  + Subitem 1
* Item 4
```

- Item 1
- Item 2
- Item 3
 - Subitem 1
- Item 4

You can embed Latex equations in-line, $\frac{1}{n} \sum_{i=1}^n x_i$ or in a new line as

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

Embed an R code chunk:

Use

```
```r
Use back ticks to
create a block of code
```
```

to produce:

```
Use back ticks to
create a block of code
```

You can also evaluate and display the results of R code. Each task can be accomplished in a suitably labeled chunk like the following:

```
summary(cars)
```

| speed | dist |
|--------------|----------------|
| Min. : 4.0 | Min. : 2.00 |
| 1st Qu.:12.0 | 1st Qu.: 26.00 |
| Median :15.0 | Median : 36.00 |
| Mean :15.4 | Mean : 42.98 |
| 3rd Qu.:19.0 | 3rd Qu.: 56.00 |
| Max. :25.0 | Max. :120.00 |

```
fit <- lm(dist ~ speed, data = cars)
fit
```

Call:

```
lm(formula = dist ~ speed, data = cars)
```

Coefficients:

| | |
|-------------|-------|
| (Intercept) | speed |
| -17.579 | 3.932 |

10.1 Including Plots

You can also embed plots. See Figure 10.1 for example:

```
par(mar = c(0, 1, 0, 1))
pie(
  c(280, 60, 20),
  c('Sky', 'Sunny side of pyramid', 'Shady side of pyramid'),
```

```
col = c('#0292D8', '#F7EA39', '#C4B632'),  
init.angle = -50, border = NA  
)
```

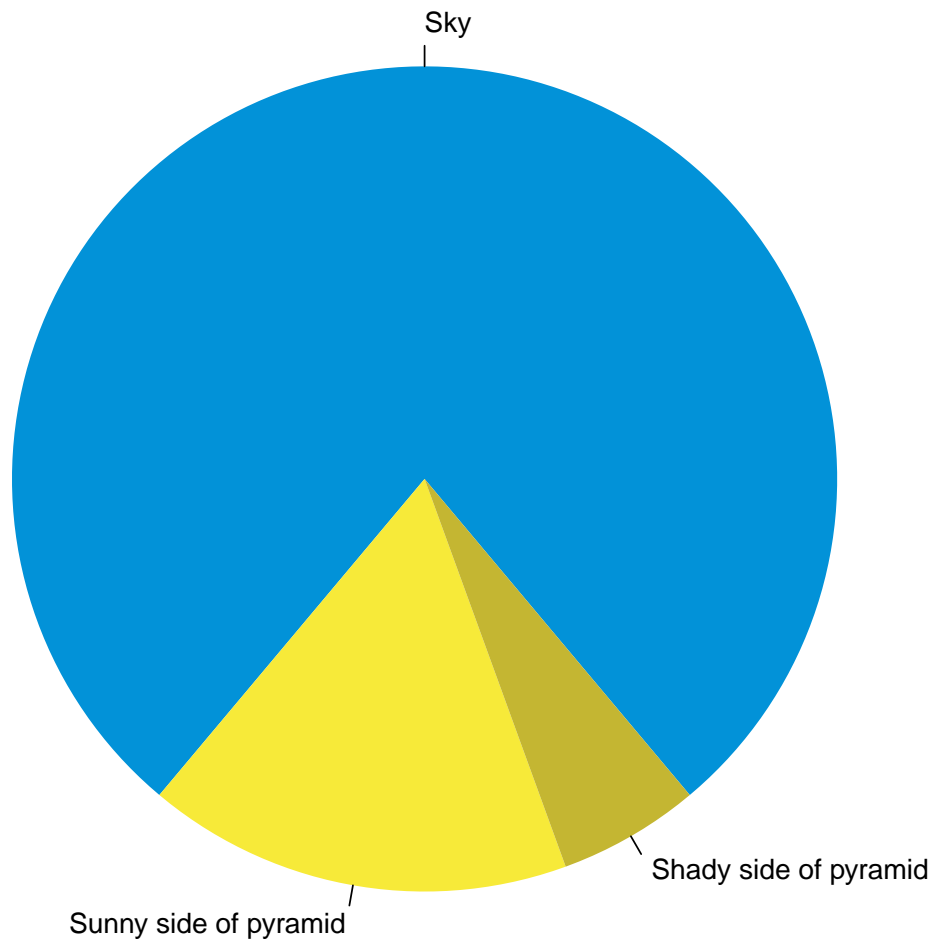


Figure 10.1: A fancy pie chart.

(Credit: Yihui Xie)

10.2 Read in data files

```
simple_data <- read.csv("https://deepbas.io/data/simple-1.dat", )  
summary(simple_data)
```

```

      initials      state      age
Length:3      Length:3      Min.   :45.0
Class :character Class :character 1st Qu.:47.5
Mode  :character Mode  :character Median :50.0
                                   Mean  :52.0
                                   3rd Qu.:55.5
                                   Max.   :61.0

      time
Length:3
Class :character
Mode  :character

```

```
knitr::kable(simple_data, format = "html")
```

```
initials
```

```
state
```

```
age
```

```
time
```

```
vib
```

```
MA
```

```
61
```

```
6:01
```

```
adc
```

```
TX
```

```
45
```

```
5:45
```

```
kme
```

```
CT
```

```
50
```

```
4:19
```

10.3 Hide the code

If we enter the `echo = FALSE` option in the R chunk (see the .Rmd file). This prevents the R code from being printed to your document; you just see the results.

initials

state

age

time

vib

MA

61

6:01

adc

TX

45

5:45

kme

CT

50

4:19