

Assignment 1

Due Sunday September 20 at 11:59pm

Instructions: Make an R Notebook and in it answer the two questions below (one Notebook for both questions). When you are done, hand in on Quercus the *output* from Previewing (or Knitting) your Notebook. Do *not* hand in the Notebook itself. You want to show that you can (i) write code that will answer the questions, (ii) run that code and get some sensible output, (iii) write some words that show you know what is going on and that reflect your conclusions about the data. Your goal is to convince the grader that you *understand* what you are doing: not only doing the right thing, but making it clear that you know *why* it's the right thing.

Do *not* expect to get help on this assignment. The purpose of the assignments is for you to see how much *you* have understood. You will find that you also learn something from grappling with the assignments. The time to get help is after you watch the lectures and work through the problems from PASIAS, via tutorial and the discussion board. The only reason to contact the instructor while working on the assignments is to report something missing like a data file that cannot be read.

You have 4 hours to complete this assignment after you start it. (This will usually be 3 hours, but I wanted to give you some extra time while you get used to the process.)

My solutions to this assignment, with extra discussion, will be available after everyone has handed in their assignment.

1. In 1997, a company in Davis, California, had problems with odour in its wastewater facility. According to a company official, the problems were caused by “unprecedented weather conditions” and “because rainfall was at 170 to 180 percent of its normal level, the water in the holding ponds took longer to exit for irrigation, giving it more time to develop an odour.”

Annual rainfall data for the Davis area is in <http://ritsokiguess.site/STAC32/rainfall.txt>. Note that clicking on the link (in blue) will display the file, and *right*-clicking on the link will give you some options, one of which is Copy Link Address, which you can then paste into your R Notebook.

The rainfall is measured in inches.

- (a) Read in and display (some of) the data.

Solution:

Look at the data file, and see that the values are separated by a single space, so `read_delim` will do it. Read straight from the URL; the hint above tells you how to copy it, which would even work if the link spans two lines.

```
my_url <- "http://ritsokiguess.site/STAC32/rainfall.txt"
rain <- read_delim(my_url, " ")
```

```
## Parsed with column specification:
## cols(
##   Year = col_double(),
##   Rainfall = col_double()
## )
```

```
rain
```

```
## # A tibble: 47 x 2
##   Year Rainfall
##   <dbl>   <dbl>
## 1  1951    20.7
## 2  1952    16.7
## 3  1953    13.5
## 4  1954    14.1
## 5  1955    25.4
## 6  1956    12.0
## 7  1957    28.7
## 8  1958    11.0
## 9  1959    12.6
## 10 1960    12.8
## # ... with 37 more rows
```

Note for later that the `Year` and the `Rainfall` have Capital Letters. You can call the data frame whatever you like, but I think something descriptive is better than eg. `mydata`.

Extra: this works because there is exactly one space between the year and the rainfall amount. But the year is always four digits, so the columns line up, and there is a space all the way down between the year and the rainfall. That means that this will also work:

```
my_url <- "http://ritsokiguess.site/STAC32/rainfall.txt"
rain <- read_table(my_url)
```

```
## Parsed with column specification:
## cols(
##   Year = col_double(),
##   Rainfall = col_double()
## )
```

```
rain
```

```
## # A tibble: 47 x 2
##   Year Rainfall
##   <dbl>   <dbl>
## 1  1951    20.7
## 2  1952    16.7
## 3  1953    13.5
## 4  1954    14.1
## 5  1955    25.4
## 6  1956    12.0
## 7  1957    28.7
## 8  1958    11.0
## 9  1959    12.6
## 10 1960    12.8
## # ... with 37 more rows
```

This is therefore also good.

It also looks as if it could be tab-separated values, since the rainfall column always starts in the same place, but if you try it, you'll find that it doesn't work:

```
my_url <- "http://ritsokiguess.site/STAC32/rainfall.txt"
rain_nogood <- read_tsv(my_url)
```

```
## Parsed with column specification:
## cols(
##   `Year Rainfall` = col_character()
## )
```

```
rain_nogood
```

```
## # A tibble: 47 x 1
##   `Year Rainfall`
##   <chr>
## 1 1951 20.66
## 2 1952 16.72
## 3 1953 13.51
## 4 1954 14.1
## 5 1955 25.37
## 6 1956 12.05
## 7 1957 28.74
## 8 1958 10.98
## 9 1959 12.55
## 10 1960 12.75
## # ... with 37 more rows
```

This looks as if it worked, but it didn't, because there is only *one* column, of years and rainfalls smooshed together as text, and if you try to do anything else with them later it won't work.

Hence those values that might have been tabs actually were not. There's no way to be sure about this; you have to try something and see what works. An indication, though: if you have more than one space, and the things in the later columns are *left*-justified, that could be tab-separated; if the things in the later columns are *right*-justified, so that they finish in the same place but don't start in the same place, that is probably aligned columns.

- (b) Summarize the data frame (as done in Lecture 2).

Solution:

I almost gave the game away: this is `summary`.

```
summary(rain)
```

```
##      Year      Rainfall
##  Min.   :1951  Min.    : 6.14
## 1st Qu.:1962  1st Qu.:12.30
##  Median :1974  Median :16.72
##   Mean   :1974   Mean   :18.69
## 3rd Qu.:1986 3rd Qu.:25.21
##   Max.   :1997   Max.    :37.42
```

The summary of the years may not be very helpful, but the summary of the annual rainfall values might be. It's not clear yet why I asked you to do this, but it will become clearer later.

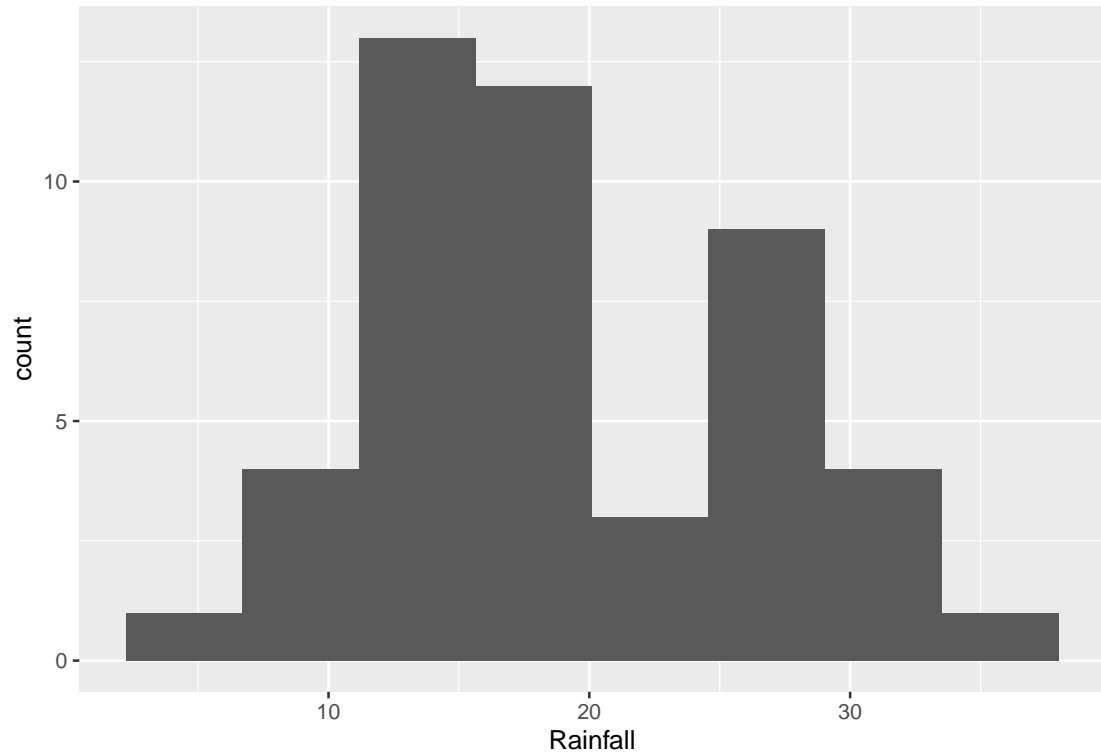
- (c) Make a suitable plot of the rainfall values. (We are not, for the time being, concerned about the

years.)

Solution:

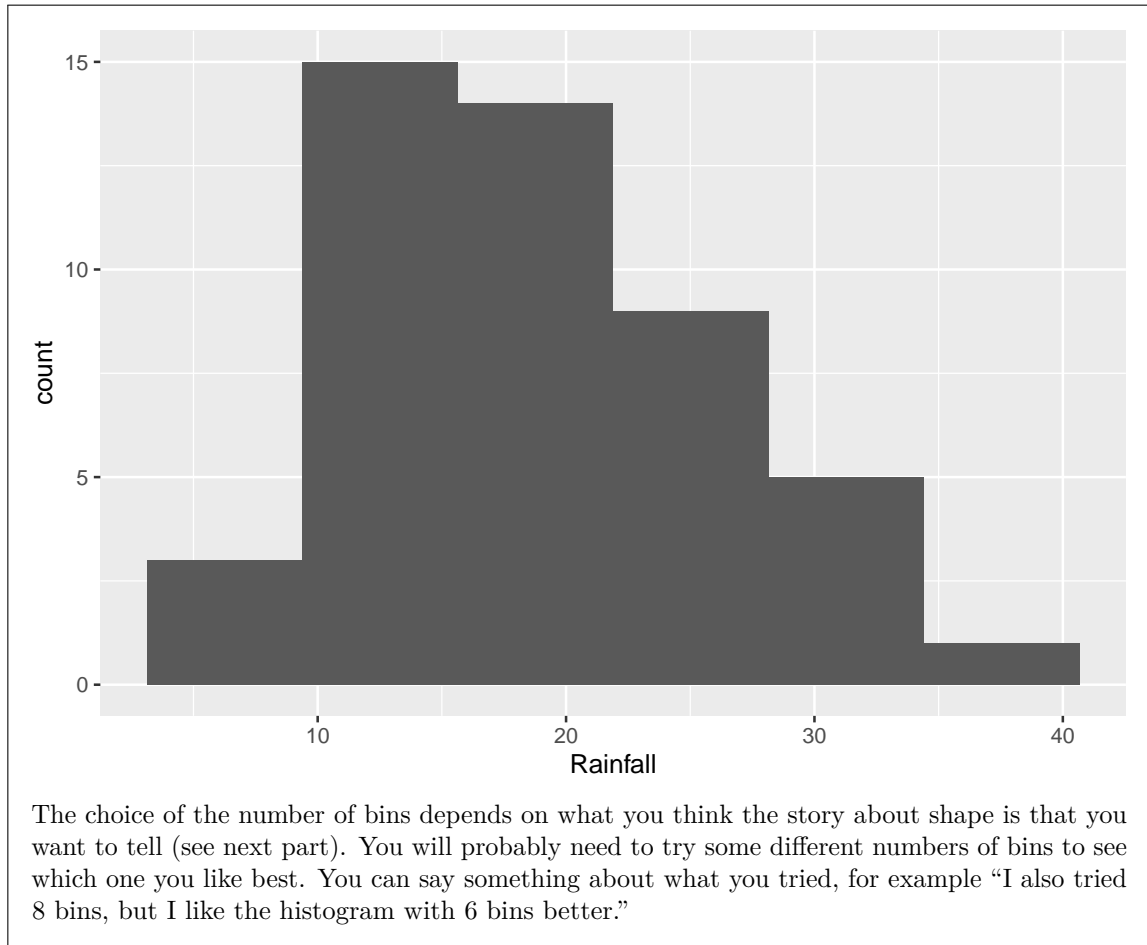
This is one quantitative variable, so a histogram is your first thought. This means picking a number of bins. Not too many, since you want a picture of the shape:

```
ggplot(rain, aes(x=Rainfall)) + geom_histogram(bins=8)
```



If you picked fewer bins, you'll get a different picture:

```
ggplot(rain, aes(x=Rainfall)) + geom_histogram(bins=6)
```



(d) How would you describe the shape of the distribution of rainfall values?

Solution:

This will depend on the histogram you drew in the previous part. If it looks like the first one, the best answer is “bimodal”: that is, it has two peaks with a gap between them. If it looks like the second one, you have an easier time; this is ordinary right-skewness.

(e) In the quote at the beginning of the question, where do you think the assertion that the 1997 rainfall was “at 170 to 180 percent of its normal level” came from? Explain briefly.

Solution:

First we need the 1997 rainfall. Go back and find it in the data. I am borrowing an idea from later in the course (because I am lazy):

```
rain %>% filter(Year==1997)
```

```
## # A tibble: 1 x 2
##   Year Rainfall
##   <dbl>    <dbl>
## 1  1997     29.7
```

29.7 inches.

Now, what would be a “normal level” of rainfall? Some kind of average, like a mean or a median, maybe. But we have those, from our summary that we made earlier, repeated here for (my) convenience:

```
summary(rain)
```

```
##      Year      Rainfall
## Min.   :1951   Min.    : 6.14
## 1st Qu.:1962   1st Qu.:12.30
## Median :1974   Median :16.72
## Mean   :1974   Mean    :18.69
## 3rd Qu.:1986   3rd Qu.:25.21
## Max.   :1997   Max.    :37.42
```

The mean is 18.69 and the median is 16.72 inches.

So divide the 1997 rainfall by each of the summaries, and see what happens, using your calculator, or using R as a calculator:

```
29.7/18.69
```

```
## [1] 1.589085
```

```
29.7/16.72
```

```
## [1] 1.776316
```

The 1997 rainfall was about 178 percent of the normal level if the normal level was the *median*.

- (f) Do you think the official’s calculation was reasonable? Explain briefly. (Note that this is not the same as asking whether the official’s calculation was *correct*. This is an important distinction for you to make.)

Solution: There are several approaches to take. Argue for yours.

If you came to the conclusion that the distribution was right-skewed, you can say that the sensible “normal level” is the median, and therefore the official did the right thing. Using the mean would have been the wrong thing.

If you thought the distribution was bimodal, you can go a couple of ways: (i) it makes no sense to use any measure of location for “normal” (in fact, the mean rainfall is almost in that low-frequency bar, and so is not really a “normal level” at all). Or, (ii) it looks as if the years split into two kinds: low-rainfall years with around 15 inches, and high-rainfall years with more than 25 inches. Evidently 1997 was a high-rainfall year, but 29.7 inches was not especially high for a high-rainfall year, so the official’s statement was an exaggeration. (I think (ii) is more insightful than (i), so ought to get more points.)

You could even also take a more conspiratorial approach and say that the official was trying to make 1997 look like a freak year, and picked the measure of location that made 1997 look more unusual.

“Normal level” here has nothing to do with a normal *distribution*; for this to make sense, the official would have needed to say something like “normal shape”. This is why language skills are also important for a statistician to have.

- (g) Do you think that the official was right to use the word “unprecedented” to describe the 1997 rainfall? Justify your answer briefly.

Solution:

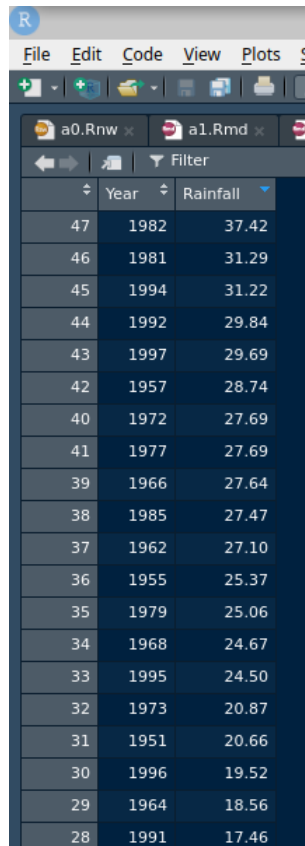
“Unprecedented” means “never seen before” or “never having happened or existed in the past”.¹ That came out of my head; [this link](#) has a very similar “never before known or experienced”).

If you look back at your histogram, there are several years that had over about 30 inches of rain: five or six, depending on your histogram. One of them was 1997, but there were others too, so 1997 was in no way “unprecedented”.

Another approach that you have seen is to **View** your dataframe:

```
View(rain)
```

That will come out as a separate tab in your R Studio and you can look at it (yourself; it won’t appear in the Preview). You can look at the 1997 rainfall (29.69 inches) and count how many were bigger than that, 4 of them. Or, save yourself some effort² and sort the rainfall values in descending order (with the biggest one first), by clicking on the little arrows next to Rainfall (twice). Mine looks like this:



The screenshot shows the R Studio interface with a dataframe named 'rain' sorted by 'Rainfall' in descending order. The dataframe has two columns: 'Year' and 'Rainfall'. The rows are numbered 28 to 47. The highest rainfall value is 37.42 inches in 1982, and the lowest shown is 17.46 inches in 1991.

	Year	Rainfall
47	1982	37.42
46	1981	31.29
45	1994	31.22
44	1992	29.84
43	1997	29.69
42	1957	28.74
40	1972	27.69
41	1977	27.69
39	1966	27.64
38	1985	27.47
37	1962	27.10
36	1955	25.37
35	1979	25.06
34	1968	24.67
33	1995	24.50
32	1973	20.87
31	1951	20.66
30	1996	19.52
29	1964	18.56
28	1991	17.46

Later, we learn how to sort in code, which goes like this (to sort highest first):

```
rain %>% arrange(desc(Rainfall))
```

```
## # A tibble: 47 x 2
##   Year Rainfall
```

```
##      <dbl>      <dbl>
## 1  1982      37.4
## 2  1981      31.3
## 3  1994      31.2
## 4  1992      29.8
## 5  1997      29.7
## 6  1957      28.7
## 7  1972      27.7
## 8  1977      27.7
## 9  1966      27.6
## 10 1985      27.5
## # ... with 37 more rows
```

A more sophisticated way that we learn later:

```
rain %>% summarize(max=max(Rainfall))
```

```
## # A tibble: 1 x 1
##   max
##   <dbl>
## 1  37.4
```

This is greater than the rainfall for 1997, ruling out “unprecedented”.

1997 was only the *fifth* highest rainfall, and two of the higher ones were also in the 1990s. Definitely not “unprecedented”. The official needs to get a new dictionary!

2. Some students in a Statistics class were asked how many minutes they typically exercised in a week. The data are shown in <http://ritsokiguess.site/STAC32/exercise.txt>.

Some of the students were male and some were female. Our concern is how the males and females compare in terms of the amount of exercise they do.

- (a) Take a look at the data file. (Click on the link, or paste the copied link into your web browser.) How is it laid out?

Solution: Aligned in columns. Or, separated by spaces, but a variable number of them. (The latter is a hint that `read_delim` will not work, and the former is a hint about what *will* work.)

- (b) Read in and display (some of) the data. (This means to display enough of what you read in to convince others that you read in the right kind of thing.)

Solution: “Aligned in columns” is the best way of saying it so that you know to use `read_table`. Hence:

```
my_url <- "http://ritsokiguess.site/STAC32/exercise.txt"
exercise <- read_table(my_url)
```

```
## Parsed with column specification:
## cols(
##   gender = col_character(),
##   minutes = col_double()
## )
```



```
exercise
```

```
## # A tibble: 29 x 2
##   gender minutes
##   <chr>     <dbl>
## 1 female      60
## 2 female     240
## 3 female      0
## 4 female     360
## 5 female     450
## 6 female     200
## 7 female     100
## 8 female      70
## 9 female     240
## 10 female      0
## # ... with 19 more rows
```

Just entering the data frame name displays the first ten rows, which is usually enough to convince anyone that you have the right thing.

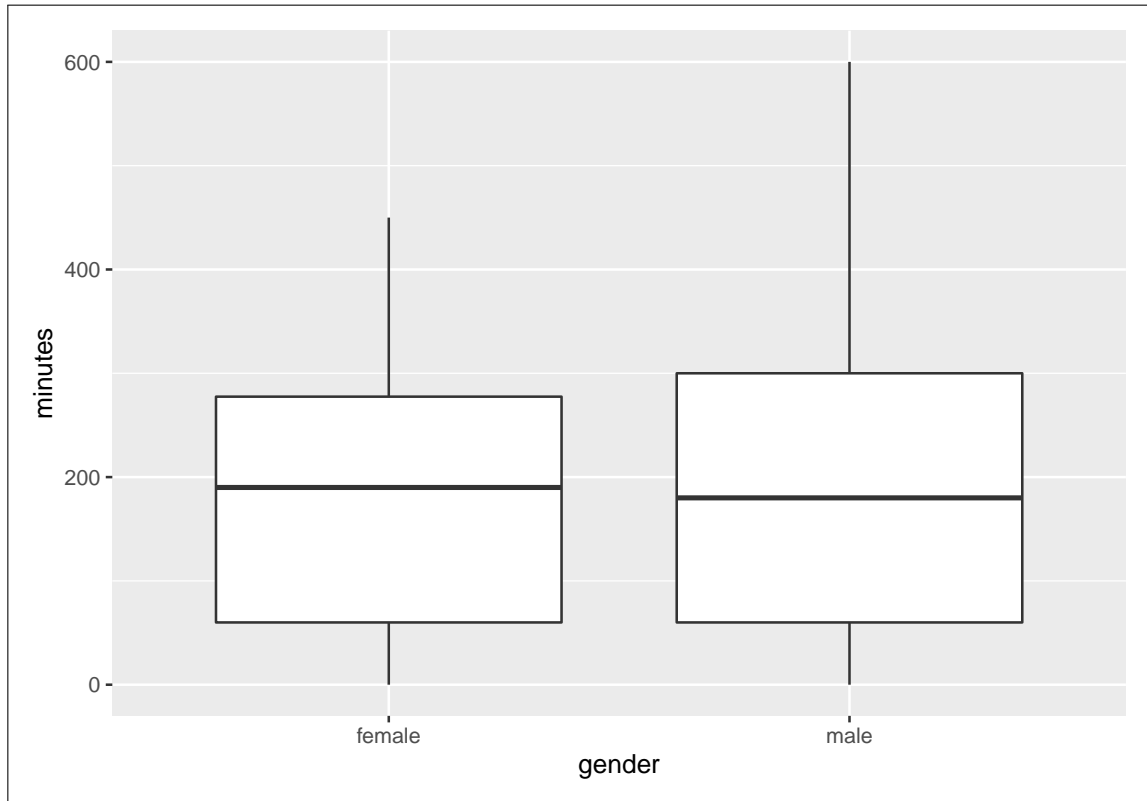
Call the data frame what you like, as long as it describes what's in the data frame in some way.

Note: Using `read.table` with a dot reveals that you have not been paying attention in this class. The instructions in this class (see course outline pages 14-16) say that I expect you to do things as they are done in this class, not as you may have learned elsewhere, so don't expect credit if you use `read.table` here or anywhere in the class; indeed, if you don't say where you got it from (since you certainly didn't get it from me) you are guilty of *plagiarism*, and you definitely don't want to get involved with that. See <https://governingcouncil.utoronto.ca/secretariat/policies/code-behaviour-academic-matters-july-1-2019> section B.I.1(d) and Appendix A, section 2(p).

- (c) Make a suitable plot of this data frame.

Solution: Two variables, one quantitative and one categorical, means that a boxplot is the thing:

```
ggplot(exercise, aes(x=gender, y=minutes)) + geom_boxplot()
```



- (d) Does there appear to be any substantial difference in the average amount of time that males and females spend exercising? Explain briefly. (“average” could be mean or median. Which is it here?)

Solution: A boxplot shows the median. So we learn here that the median time spent exercising per week is very slightly higher for females. However, there is a lot of variability (the height of the boxes), and so the difference between the medians is very small. Hence, there is definitely *no* substantial difference between males and females.

Less insightfully, the difference in median between males and females is very small (but that doesn’t rule out the possibility that the spread is very small also).

- (e) How do you know that both distributions, for males as well as females, are skewed to the right? Explain (very) briefly.

Solution: The upper whiskers, at the top of the box, are longer than the lower ones (at the bottom).

- (f) For data like this, why does it make practical sense that the distributions are skewed to the right?

Solution: Nobody can exercise less than 0 minutes per week, but there is no upper limit: a student in the class can exercise as much as they want. This means that there could be unusually high values, but not unusually low values.

Extra: Distributions that have a limit on one side are often skewed to the other side. This one has a limit on the left, so it is skewed to the *right*. This is most likely to be true if there are

observations close to the limit, such as the people in this data set that don't exercise at all (and there are some of those).

Notes

1. Searching for *define* followed by a word is a good way to find out exactly what a word means, if you are not sure, but you should at least say where you got the definition from if you had to look it up.
2. When you have a computer at your disposal, it's worth taking a few minutes to figure out how to use it to make your life easier.