Assignment 1

Instructions: Make an R Notebook and in it answer the questions below. 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, that is *before* you start work on the assignment. The only reasons to contact the instructor while working on an assignment are to report (i) something missing like a data file that cannot possibly be read, (ii) something *beyond your control* that makes it impossible to finish the assignment in time after you have started it.

You have 4 hours to complete this assignment after you start it.

1. One way to assess climate change is to look at weather records over a number of years and to identify any changes.

   Annual rainfall data for the Davis, California is in http://ritsokiguess.site/STAC32/rainfall.txt. (Right-click on the URL and select Copy Link Address, then paste into your R Notebook.) The rainfall is measured in inches.

   (a) Read in and display (some of) the data. Briefly justify why you coded it as you did.

Solution:

Look at the data file, and see that the values are separated by a single space, so `read_delim` will do it. (This is the brief justification.) You *need to have some words* in this part explaining your choice of `read_delim` (or `read_table`: see below).

Read straight from the URL rather than copying and pasting the data or doing anything like that:[1]

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

```
##
## -- 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 1: `read_delim` 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 is to say, another possibility for your "brief justification" is that the columns are lined up all the way down. That means that this will also work:

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

```
##
## -- 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)
```

```
##
## -- 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.[2]

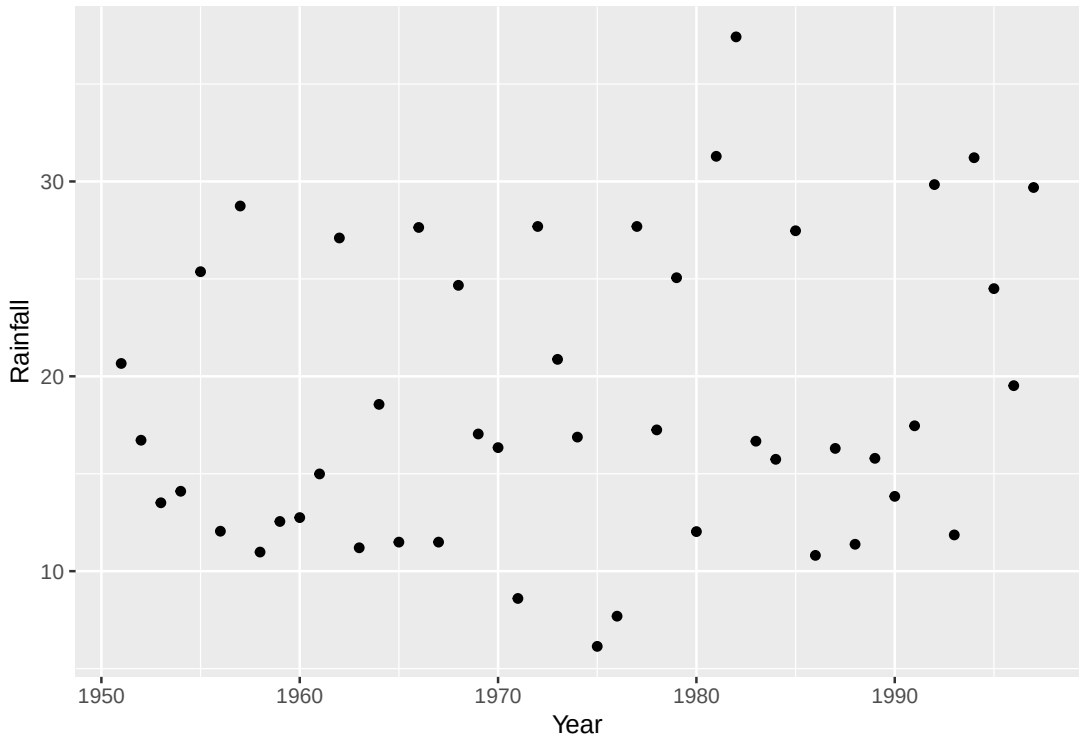
Hence those values that might have been tabs actually were not. There's no way to be sure about this; you have to try something and see what works.[3]

Extra 2: `read.table` with a dot will work, but it is in this course *wrong*. As I stated in the course outline, I expect you to do things *as they are done in this course*. Strictly speaking, if you use `read.table`, which you did not learn from me, and you say nothing about where you got it from, you are guilty of plagiarism, which is an academic offence. I will probably not be that strict, but you should certainly expect to lose marks for doing things differently from how they are done in this course.

(b) Make a suitable plot of the rainfall values and years. Explain briefly why you drew the plot you did.

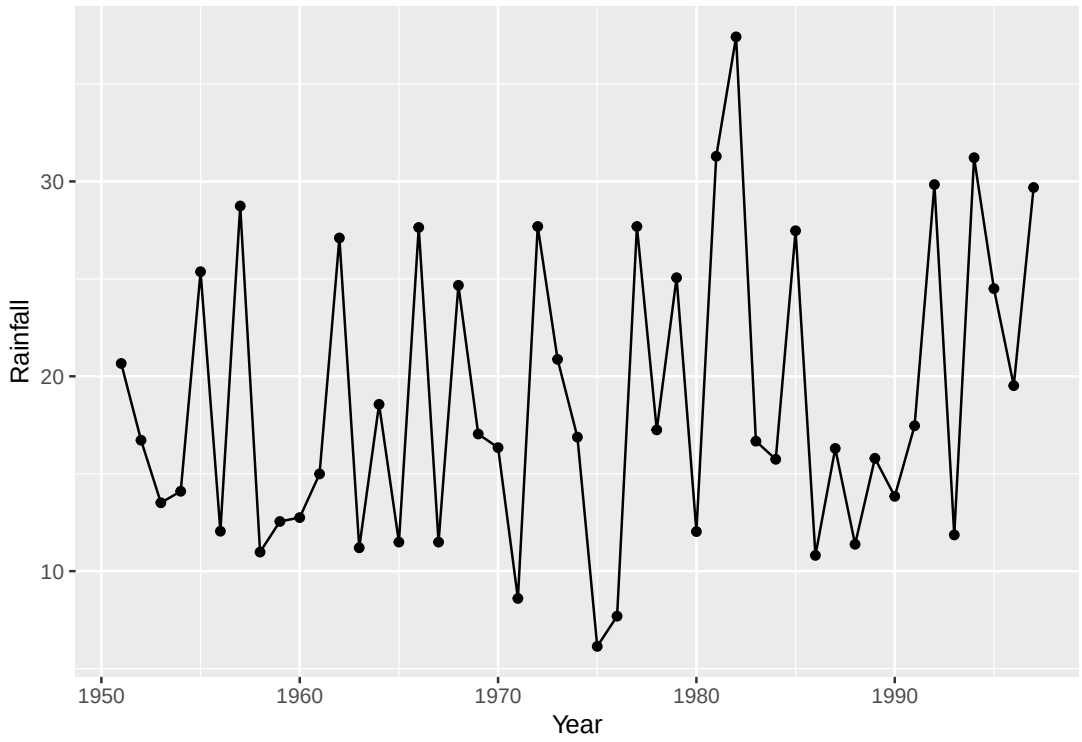
Solution: This is two quantitative variables, so a scatterplot makes sense. (That's the brief explanation.) To decide which variable goes on which axis, think of rainfall as a response to the explanatory variable year, or note that time goes by tradition on the horizontal axis:

```
ggplot(rain, aes(x=Year, y=Rainfall)) + geom_point()
```



This is a time trend, so it would also be reasonable to join the points with lines:

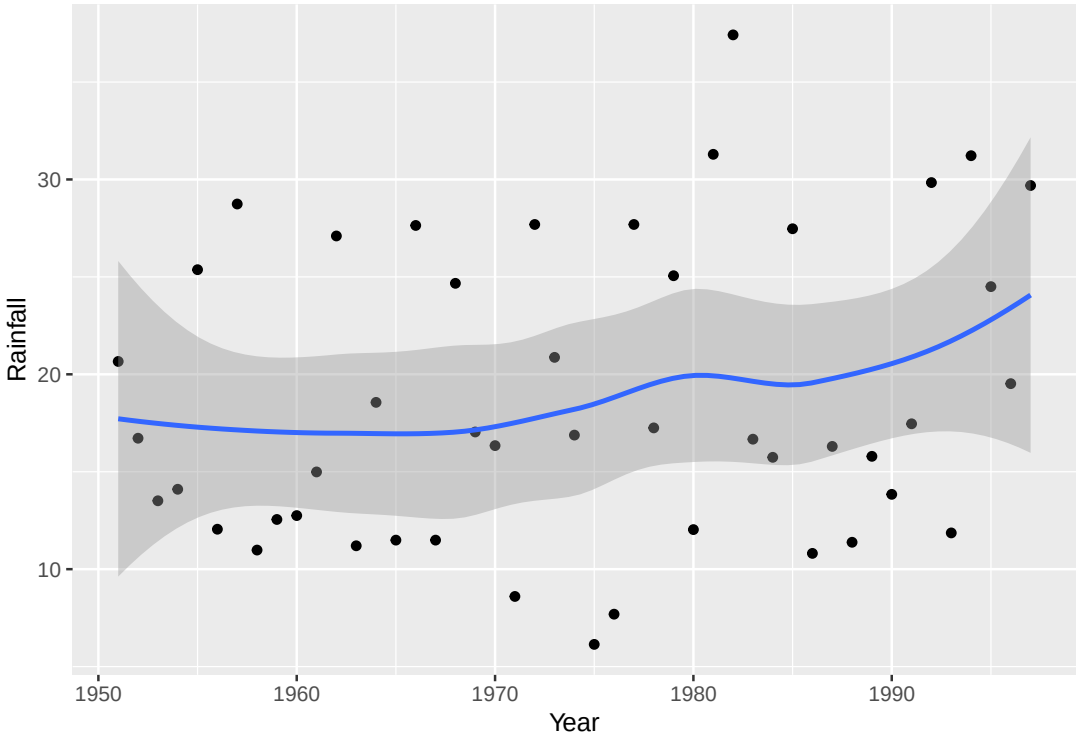
```
ggplot(rain, aes(x=Year, y=Rainfall)) + geom_point() + geom_line()
```



and because any trend looks irregular, you could also justify putting a smooth trend through the points:

```
ggplot(rain, aes(x=Year, y=Rainfall)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

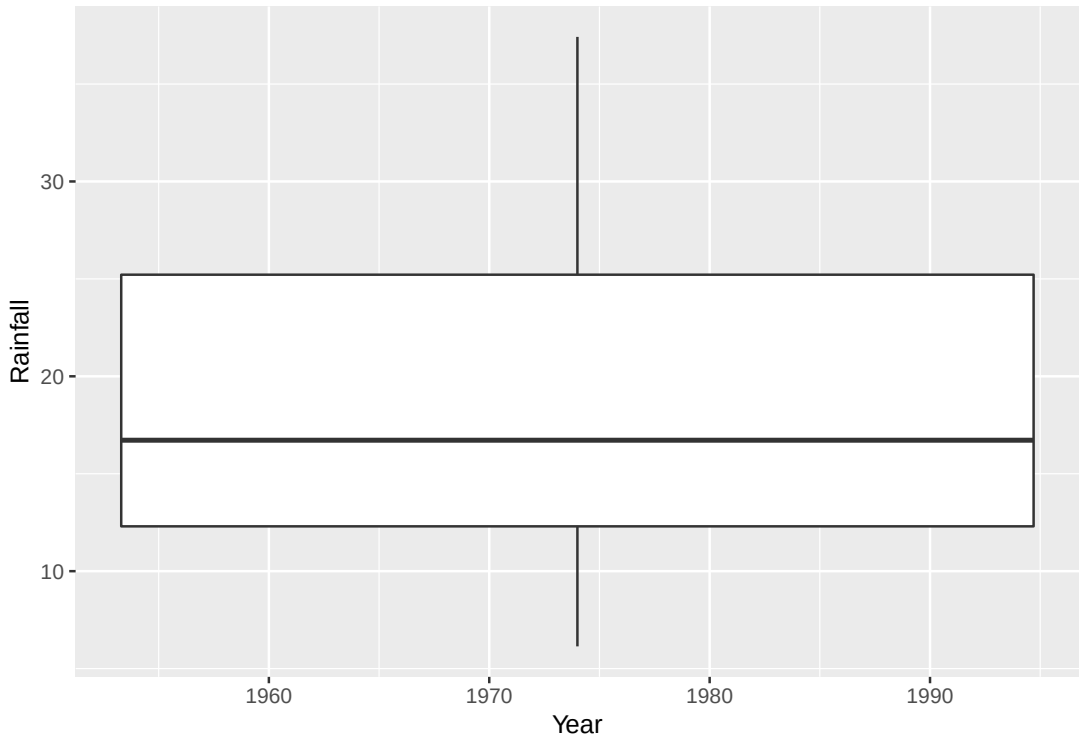


This one is rather interesting (as an aside): there is a more noticeable upward trend at the end, after about 1985. If you study environmental science, you may have seen that a lot of time plots of climate data show a bigger change after about 1990, and this is one of those.

If you treat the year as categorical, and try to draw a boxplot, it won't work out so well:

```
ggplot(rain, aes(x=Year, y=Rainfall)) + geom_boxplot()
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



This is actually a boxplot of rainfall in all the years together; it ignored year because it couldn't figure out what you wanted to do with it.

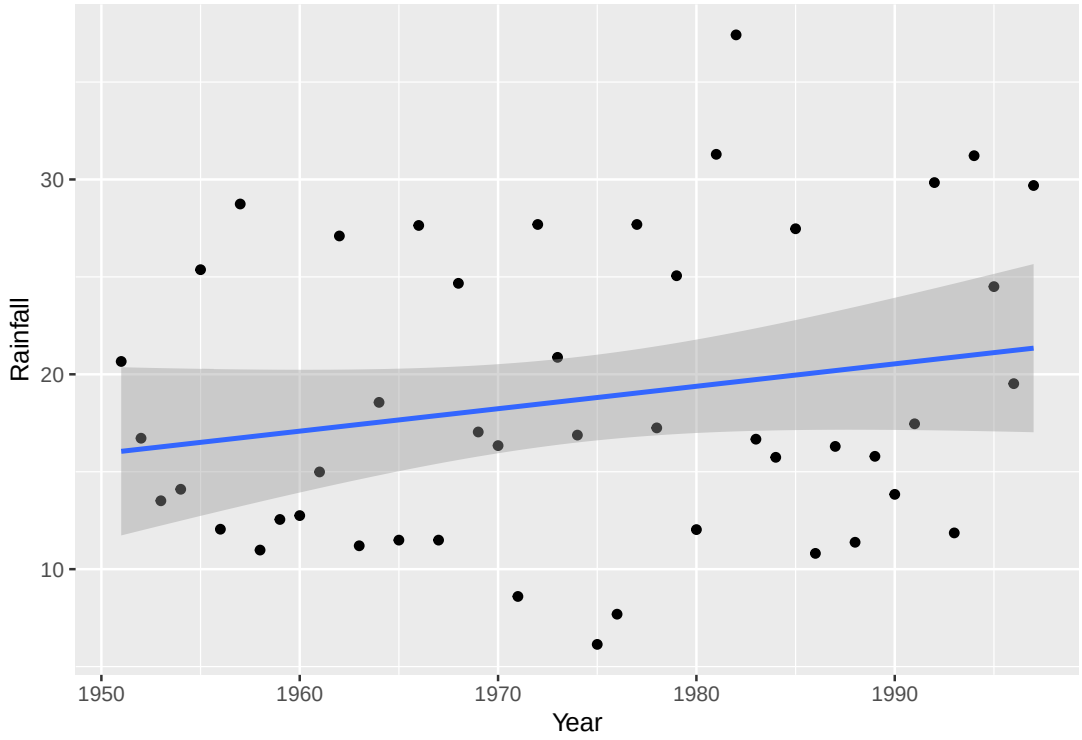
The kind of situation where this approach does work is if you have something like monthly rainfall over a number of years, and then you have multiple values for January over all the years that you have data for.

(c) Add a regression line to your plot. Is there any convincing indication of a trend, upward or downward, in annual rainfall for Davis, California? Discuss briefly.

Solution: Thus:

```
ggplot(rain, aes(x=Year, y=Rainfall)) + geom_point() +
    geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The regression line goes uphill, but this is not convincing evidence of an overall upward trend for several reasons (pick one):

- the increase, if there is one, is very small
- the data points are mostly far away from the regression line
- there are points within the grey envelope all the way across (meaning, for example, that the mean annual rainfall could be 20 inches for all the years).

Further evidence comes from a regression (which we haven't looked at in the course yet):

```
r.1 <- lm(Rainfall~Year, data=rain)
summary(r.1)
```

```
##
## Call:
## lm(formula = Rainfall ~ Year, data = rain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.670  -5.252  -1.905   7.088  17.804
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -208.57325  158.42281  -1.317    0.195
## Year           0.11513    0.08025   1.435    0.158
##
## Residual standard error: 7.463 on 45 degrees of freedom
```

```
## Multiple R-squared:  0.04373,    Adjusted R-squared:  0.02248
## F-statistic: 2.058 on 1 and 45 DF,  p-value: 0.1583
```

The slope is indeed positive, but it is not significantly different from zero. Also, the R-squared is *very* small, 4.4%. Thus the apparent upward trend is no more than chance.

Extra: looking at the plot from earlier with the smooth trend suggests that there may not have been any real increase since 1950, but if you look only since about 1985 it may be different. There is a statistical question mark over that, though, because we started looking at 1985 only because there seemed to be an upward trend starting from about there.

2. At a high school in New Jersey, teachers were interested in what might help students to learn algebra. One idea was laptops as a learning aid, to see whether having access to one helped with algebra scores. (This was some time ago.) The 20 students in one class were given laptops to use in school and at home, while the 27 students in another class were not given laptops. For all of these students, the final exam score in algebra was recorded. The data are in http://ritsokiguess.site/STAC33/algebra.txt, with two columns, one indicating whether the student received a laptop or not, and the other giving their score on the algebra final exam.

   (a) Read in and display (some of) the data. Do you have (i) the correct number of observations, and (ii) the correct *type* of columns? Explain briefly.

Solution:

Take a look at the data file first: the data values are *aligned in columns* with variable numbers of spaces between, so `read_table` is the thing. Read directly from the URL, rather than trying to copy the data from the website:

```
my_url <- "http://ritsokiguess.site/STAC33/algebra.txt"
algebra <- read_table(my_url)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   laptop = col_character(),
##   score = col_double()
## )
```

```
algebra
```

```
## # A tibble: 47 x 2
##    laptop score
##    <chr>  <dbl>
##  1 yes       98
##  2 yes       84
##  3 yes       97
##  4 yes       93
##  5 yes       88
##  6 yes       57
##  7 yes      100
##  8 yes       84
##  9 yes      100
## 10 yes       81
## # ... with 37 more rows
```

There were $20+27=47$ students altogether in the two classes, and we do indeed have 47 rows, one per student. So we have the right number of rows. This is two independent samples; each student was in only one of the two classes, either the class whose students got laptops or not. The values in the `laptop` column are text (see the `chr` at the top), and the values in the `score` column are numbers (`dbl` at the top). Alternatively, you can look at the R Console output in which you see that `laptop` is `col_character()` (text) and `score` is `col_double()` (numerical, strictly a decimal number).

Extra 1: as in the first question, `read.table` works but it is *wrong* in this course.

Extra 2: with more than one space between the values, `read_delim` will not work. Or, perhaps more confusing, it will appear to work and then fail later, which means that you need to pay attention:

```
d <- read_delim(my_url, " ")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   laptop = col_character(),
##   score = col_character()
## )
```

```
d
```

```
## # A tibble: 47 x 2
##    laptop score
##    <chr>  <chr>
##  1 yes    "    98"
##  2 yes    "    84"
##  3 yes    "    97"
##  4 yes    "    93"
##  5 yes    "    88"
##  6 yes    "    57"
##  7 yes    "   100"
##  8 yes    "    84"
##  9 yes    "   100"
## 10 yes    "    81"
## # ... with 37 more rows
```

This *looks* all right, but look carefully: the `laptop` column is correctly text, but the `score` column, which should be numbers (`dbl`), is actually text as well. An easier way to see this is to look at the output from the console, which is the descriptions of the columns: they are *both* `col_character` or text, while `score` should be numbers. On mine, though possibly not on yours, you can see exactly what went wrong: with more than one space separating the values, the remaining spaces went into `score`, which then becomes a piece of text with some spaces at the front and then numbers.

This will actually work for a while, as you go through the question, but will come back to bite you the moment you need `score` to be numerical (eg. when you try to draw a boxplot), because it is actually not numerical at all.

Extra 3: this is the standard R way to lay out this kind of data, with all the outcome values in one column and a second (categorical) column saying which group each observation was in. In other places you might see two separate columns of scores, one for the students with laptops

and one for the students without, as below (you won't understand the code below now, but you will by the end of the course):

```
algebra %>%
  mutate(row = c(1:20, 1:27)) %>%
  pivot_wider(names_from = laptop, values_from = score)
```

```
## # A tibble: 27 x 3
##      row   yes    no
##    <int> <dbl> <dbl>
##  1     1    98    63
##  2     2    84    83
##  3     3    97    97
##  4     4    93    93
##  5     5    88    52
##  6     6    57    74
##  7     7   100    83
##  8     8    84    63
##  9     9   100    88
## 10    10    81    86
## # ... with 17 more rows
```

A column of `yes` and a column of `no`. The classes were of different sizes, so the `yes` column, with only 20 observations, has some `NA` ("missing") observations at the end (scroll down to see them) to enable the dataframe to keep a rectangular shape.

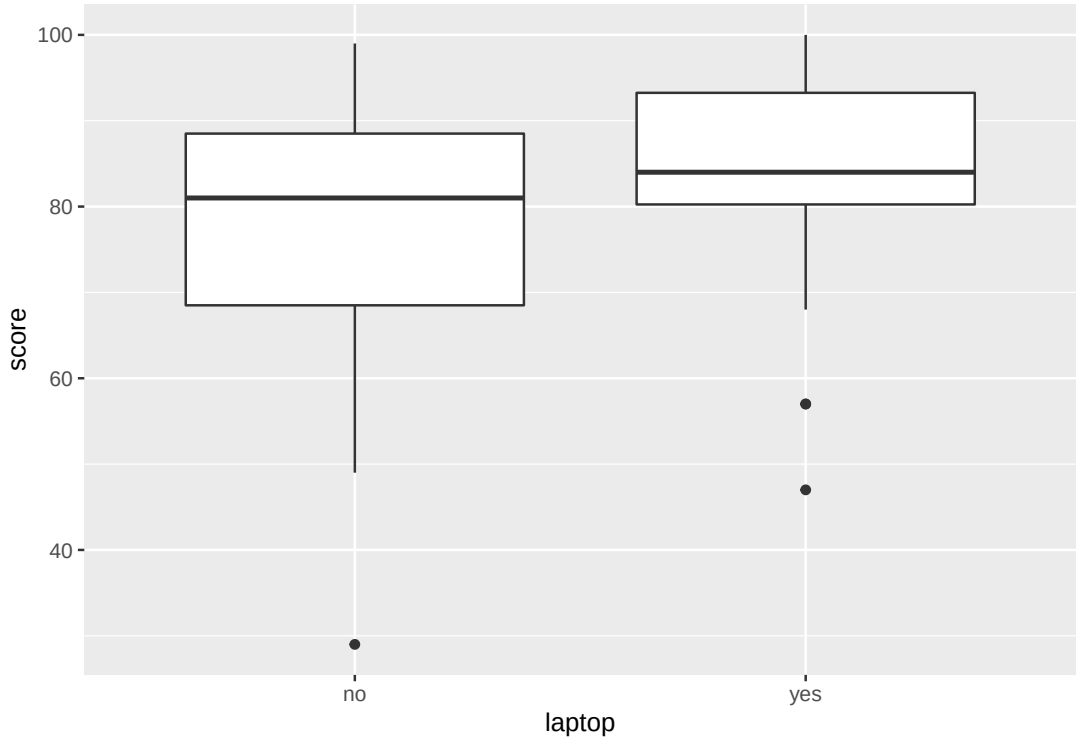
We will learn later that these layouts of data are called "longer" and "wider" (respectively), and how to convert between them. R usually likes "longer" data, as in the data file, but you will often see data sets displayed wider because it takes up less space.

(b) Make a suitable graph of these data.

Solution:

The teachers were hoping to see how the laptop-yes and the laptop-no groups compared in terms of algebra scores, so side-by-side boxplots would be helpful. More simply, we have one quantitative and one categorical variable, which is a boxplot according to the table in the notes:

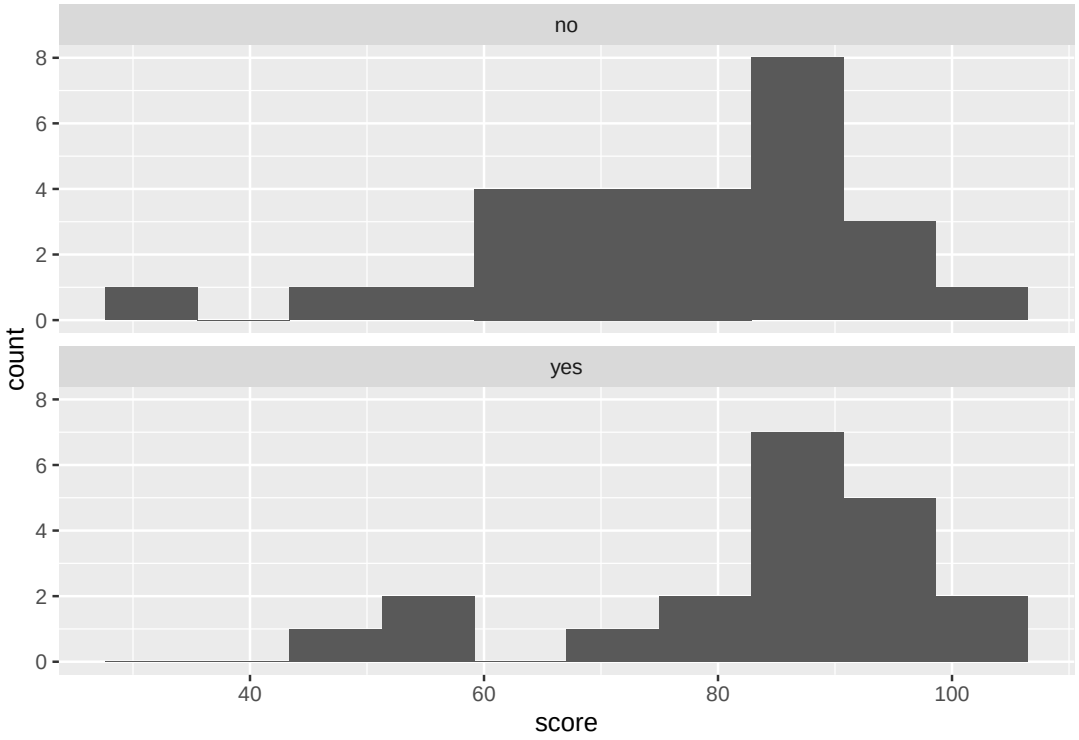
```
ggplot(algebra, aes(x = laptop, y = score)) + geom_boxplot()
```



Extra: as you will note below, the median score for the students with laptops is a little higher for the students who had laptops. This is easy to see on a boxplot because that is what a boxplot does. (That was what Tukey, who we will meet later, *designed* the boxplot to do.)

Another plot you might have drawn is a histogram for each group, side by side, or, as they come out here, above and below. This works using facets:

```
ggplot(algebra, aes(x = score)) +
  geom_histogram(bins = 10) +
  facet_wrap(~laptop, ncol = 1)
```



Looking at those, can you *really* say that the median is slightly higher for the `yes` group? I really don't think you can. Certainly it is clear from the histograms that the spread for the `yes` group is less, but comparing the medians is much more easily done from the boxplot. The teachers were interested in whether the laptops were associated with higher scores on average, so the kind of comparison that the boxplot affords is clearly preferred here.

If you are interested in the code: you imagine you're going to make a histogram of scores regardless of group, and then at the end you facet by your grouping variable. I added the `ncol = 1` to make the plots come out in one column (that is, one above the other). If you don't do this, they come out left and right, which makes the distributions even harder to compare.

(c) Comment briefly on your graph, thinking about what the teachers would like to know.

Solution:

There are three things to say something about, the first two of which would probably interest the teachers:

- comparison of centre: the *median* score for the group that had laptops is (slightly) higher than for the group that did not.
- comparison of spread: the scores for the group that had laptops are less spread out (have smaller *interquartile range*) than for the group that did not.
- assessment of shape: both groups have low outliers, or are skewed to the left in shape.

Some comments from me:

- boxplots say *nothing* about mean and standard deviation, so don't mention those here. You should say something about the measures of centre (median) and spread (IQR) that they *do* use.

- I think of skewness as a property of a whole distribution, but outlierness as a property of individual observations. So, when you're looking at this one, think about where the evidence about shape is coming from: is it coming from those one or two low values that are different from the rest (which would be outliers), or is it coming from the whole distribution (would you get the same story if those maybe-outliers are taken away)? My take is that if you take the outliers away, both distributions are close to symmetric, and therefore what you see here is outliers rather than skewness. If you see something different, make the case for it.

One reason to suspect skewness or something like it is that test scores have an upper limit (100) that some of the scores got close to, and no effective lower limit (the lower limit is 0 but no-one got very close to that). In this sort of situation, you'd expect the scores to be skewed away from the limit: that is, to the left. Or to have low outliers rather than high ones.

(d) Work out the median and inter-quartile range for the students who did and who did not have laptops, and compare with the boxplot. (In R, the inter-quartile range is `IQR` in uppercase.)

Solution:

This is easy to make way harder than it needs to be: `group_by` and `summarize` will do it. Put the two summaries in one `summarize`:

```
algebra %>%
  group_by(laptop) %>%
  summarize(med = median(score), iqr = IQR(score))
```

```
## # A tibble: 2 x 3
##   laptop   med   iqr
## * <chr>  <dbl> <dbl>
## 1 no        81    20
## 2 yes       84    13
```

Then relate these to the information on the boxplot: the centre line of the box is the median. For the `no` group this is just above 80, so 81 makes sense; for the `yes` group this is not quite halfway between 80 and 90, so 84 makes sense.

The inter-quartile range is the height of the box for each group. Estimate the top and bottom of the two boxes from the boxplot scale, and subtract. For the `no` group this is something like $88 - 68$ which *is* 20, and for the `yes` group it is something like $93 - 80$ which is indeed 13.

Extra: I didn't ask you here about whether the difference was likely meaningful. The focus here was on getting the graph and summaries. If I had done so, you would then need to consider things like whether a three-point difference in medians could have been chance, and whether we really had random allocation of students to groups.

To take the second point first: these are students who chose to take two different classes, rather than being randomly allocated to classes as would be the case in a true experiment. What we have is really in between an experiment and an observational study; yes, there was a treatment (laptop or not) that was (we hope) randomly allocated to one class and not the other, but the classes could have been different for any number of other reasons that had nothing to do with having laptops or not, such as time of day, teacher, approach to material, previous ability at algebra, etc.

So even if we are willing to believe that the students were as-if randomized to laptop or not,

the question remains as to whether that three-point difference in medians is reproducible or indicative of a real difference or not. This is the kind of thing we would try a two-sample t-test with. In this case, we might doubt whether it will come out significant (because of the small difference in medians and presumably means, compared to the amount of variability present), and, even then, there is the question of whether we *should* be doing a t-test at all, given the outliers.

Notes

1. Reading from the URL is "reproducible" in that somebody else doing what you did will get exactly what results you did. Copying and pasting data is not in general reproducible because somebody else might do it differently from you, such as missing a line of data. Copying and pasting a URL, especially by right-clicking and Copy Link Address, has much less to go wrong. Physically selecting the URL text by selecting all the letters in the URL can go wrong, especially if the printed URL goes over two lines on the page.

2. It is actually possible to disentangle data like this and then work with it (we will see how later in the course), but the best way to do it is to make it easiest for yourself and anyone reading your code, and read the data file in a way that is appropriate for the layout you have in the file.

3. 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.