390R Data import and export notes
11/5/2020

When comparing 1 continuous and 1 categorical variables you can use histograms and boxplots to summarize the data
2 categoricals you can use boxplot or heatmap
2 continuous variables you can use scatter plots or discretize one variable to categorical
These tools are just examples of visualization.

We've finished the exploratory data analysis part. You can practice more by analyzing more data sets to get a better idea of data visualization.

Question: What variable in the diamonds dataset is most important for predicting the price of a diamond? (consider variables: carat, clarity, color, and cut) How is that variable correlated with cut? Why does the combination of those two relationships lead to lower quality diamonds being more expensive?
Based on the carat vs price graph, there's a positive association between carat and price, so when the carat increases so does price.
Let's look at color vs price. Color is categorical and price is continuous.
ggplot(diamonds)+geom_boxplot(aes(x=color,y=price)) this shows us that as the color quality decreases, the price increases.
Now let's look at clarity vs carat.
ggplot(diamonds)+geom_boxplot(aes(x=clarity,y=carat)) this shows that the better the clarity, the smaller the carat which would explain why the price is lower for better clarity.

New topic: Data import and export
I want to talk about how we import datasets from other data files. Using R you can import many types of data files like cvs and excel. Depending on the type of data you need different functions.
Today I will focus on read_csv and how R transforms the external data files to the data formats in R.
Ex: heights <- read_csv("data/heights.csv") this object will result as a tibble
Put the directory in quotes
The most important thing is to get the directory of your data files.

Working directory: To get the current working directory you can use the function getwd()
You can set your working directory by going to the files tab, and go to the folder you want to set as the working directory, then click more and then click working directory.
setwd() is the function to set the working directory

Lets talk about read_csv
Csv = comma separated files

```
read_csv("a,b,c
        1,2,3
```

4,5,6”)
The read_csv function will take the first line as variables and then the next lines are observations split by each new line.

read_csv(“a,b,c
1,2,3
4,5,6”, skip = 2) This will skip the first 2 lines, so the 3 line will be the variables and the 4th will be observations

read_csv(“a,b,c
1,2,3
4,5,6”, col_names = FALSE) will make all lines observations and provide preset variables names.

You can also set the column names
read_csv(“a,b,c
1,2,3
4,5,6”, col_names = c(“x”,”y”,”z”)

Question: What function would you use to read a file where fields were separated with “|”, e.g. "1|2|3\n4|5|6"? (check read_delim())
read_delime(“1|2|3\n4|5|6”, delime = “|”, col_names = FALSE)

Question: Identify what is wrong with each of the following inline CSV files. What happens when you run the code?
read_csv("a,b\n1,2,3\n4,5,6")
First line only has 2 variables and the next lines have 3. This will result in observations 3 and 6 being ignored.
read_csv("a,b,c\n1,2\n1,2,3,4")
It has 3 variables but not 3 observations per line.

To understand how read_csv works, we can break it into smaller parts which are the vectors.

R imports string vectors using the parse function

parse_integer(c(“1”,”123”,”456”)) takes a character vector as an input and outputs integers

If you try to put a character into parse_integer, it will result in an NA error

Data export

Similar to data import. You need to provide a directory you want to export the data to.

ggsave(“car.pdf”) this saves a cars.pdf file into the working directory.

Write_csv is the opposite of read_csv

write_csv(cars, “cars.csv”)   will write the cars data to the cars.csv file in the working directory