Feel free to use these exercises to test your data manipulation and visualization skills on your own, or submit them to `amjad_dabi@unc.edu` for feedback. Submissions in an Rscrpit or Markdown format are acceptable.

Take a look at the `tidyverse` teaching document. It should give you a great outline of the basics of working with `tidyverse`. In the next few questions you will apply these concepts to the toy dataset attached.

# Data Loading and Manipulation

1. Load the dataset and store it in the variable `data`. Remember that if you are working on a local Rstudio distribution can check the current working directory of your script with `getwd()` and set it with `setwd()`. For Rstudio Cloud, you will need to upload the `toy_dataset.csv` to the project directory before loading it. Once you have loaded the data, use the function `head()` to get the first part of the data to print out. That is usually always a good step to look at what you are working with.

2. Now let's play around with subsetting, filtering, and doing some basic operations on the data. First of all let's clean the data a bit by dropping that `Number` column using the `select()` function in `tidyverse`, as we do not need it. Store the cleaned data in the new dataframe `clean_data`. It is highly encouraged that you use the pipe assignment `%>%`. (Tip: you can use the shortcut Ctrl + Shift ⇧ + M for typing `%>%`). Remember that you can either specify the columns to keep by passing their names as a vector to `select()` or which columns to drop by adding a minus sign `-` right before that vector. You can also drop or keep columns by passing their indices as a vector to the `filter()` function.

3. Say we are interested in statistics about income for each city. Create a new dataframe `income_by_city` from `clean_data` which summarizes the income as a mean in an `income_mean` column and the median in an `income_median` column. Remember the useful functions `group_by()` and `summarise()`. Sort this dataframe by `income_mean` to make it easier to rank these cities. Which city has the highest income and which has the lowest? Is there a large difference between the mean and the median for each city? What does that tell you?

4. We may be interested in the number of specific data points in the dataset. For instance, what are the number of data points for each of the two sexes contained in the dataset? Which function would you use to achieve this?

5. Now we want to know about average income, but by illness status rather than city. First, filter the data so it only contains those aged 40 and up, and then calculate the mean and median income by illness status similar to what you did in question 3. Do you notice a large difference between the two illness statuses?

6. If we wanted to build some classifier to predict illness status from this data, it may be useful to turn our categorical variable into a binary one. Create a new column `Illness_binary` in `clean_data` that contains a 0 for No and a 1 for Yes in the `Illness` column.

# Data Plotting with ggplot

1. Create a boxplot of the data for income vs city. Compare what you see on this plot for what you found out in question 3 from the previous section.

2. Create a line-graph for age vs average income from the data. You will first need to calculate the average income for each age point before using `ggplot`. Remember that `geom_point()` will plot the data points and `geom_line()` will connect these points for you to create a neat line-graph.

3. (Bonus Question)

   Repeat the last question, but now create two line-graphs, in two different colors, one for each illness status. Notice that to do this, you will need to modify how you group the data before calculating the mean. Do you see any interesting trends?