14.310x: Data Analysis for Social Scientists
Joint, Marginal, and Conditional distributions & Functions of Random Variables

Welcome to your third homework assignment!We encourage you to get an early start, particularly if you still feel you need more experience using R. We have provided this PDF copy of the assignment so that you can print and work through the assignment offline. You can also go online directly to complete the assignment. If you choose to work on the assignment using this PDF, please go back to the online platform to submit your answers based on the output produced.

Good luck ☺!

In this problem set we will guide you through different ways of accessing real data sets and how to summarize and describe it properly. First we will go through some of the data that is collected by the World Bank. We will do some cleaning on the data before we start analyzing it. Then, we will try to do a simple web scraping exercise where we will analyze the data as well.

Let's start with the data sets of the World Bank. Please go to the World Bank Datasets website: http://data.worldbank.org/. Once you are there use the data catalogue to find the Gender Stats data. Please download the file in csv format. Save the file in your computer in a folder where you can get it easily. In my case, I have saved the files in the following directory: *"/Users/raz/Dropbox/14.31 edX Building the Course/Problem Sets/PSET 3/Gender_Stats_csv"*. It is important to work in the same directory that the files are or to use the whole path when you specify opening a data set. To know which directory you are currently working in, you can use the command getwd(). Similarly, in order to set a different directory, you can use the command setwd(). For the purpose of analyzing the data, we are going to use the packages "utils" and "tidyverse". Once you have uploaded the data to R you are going to see there are multiple indicators of gender, countries and years in the data. In this case we are just interested in analyzing the data for one indicator that is the *Adolescent Fertility Rate,* in the data the indicator code for this variable is called `SP.ADO.TFRT` . This indicator measures the annual number of births to women 15 to 19 years of age per 1,000 women in that age group. It represents the risk of childbearing among adolescent women 15 to 19 years of age. It is also referred to as the age-specific fertility rate for women aged 15-19. Once you have completed this problem set you'll have more information of how this rate has evolved over time and how it varies across different groups of countries.

Take a look at the following lines of code, whose main purpose is to upload the data in a data frame and to choose the proper indicator. Please, try to understand the code and then run it in your computer. Remember to set the directory accordingly to the folder where you saved the files.

```
#Preliminaries
rm(list = ls())
library("utils")
library("tidyverse")

setwd("/Users/raz/Dropbox/14.31 edX Building the Course/Problem
Sets/PSET 3/Gender_Stats_csv")
```

```
#Getting the data
gender_data <- as_tibble(read.csv("Gender_StatsData.csv"))
```

1. What is the purpose of the line rm(list = ls())?
    a. To remove all the current existing objects in R
    b. To change the current directory path
    c. To list all the files in the current directory
    d. To look in the web for the World Bank dataset.

2. The first thing you want to figure out when you look at a new dataset, is how it is organized. If your dataset is stored as a tibble, you can simply print the object, and it will print in a nice-looking format. Alternatively, you can also use the built-in R commands such as: `str()` which allows you to see the structure of an object in R. Likewise, the commands `head()` and `tail()`, will allow you to see the first six and last six observations of your data frame, respectively. Another useful function is the function `dim()` , which will give you the number of rows and columns in your dataset. Take the time to explore the data using these commands and others. Which of the following statements best describes how your data is organized?
    a. The unit of observation is a country/region for a given year .
    b. Each row corresponds to a country/region and an indicator.
    c. Each row corresponds to an indicator for a given year.
    d. The unit of observation is country-indicator-year.

3. Now, generate a tibble called `"teenager_fr"`, which contains only the adolescent fertility rate indicator for each country-year. Your code should look something like this:
`teenager_fr <- □ (gender_data, Indicator.Code == "SP.ADO.TFRT")`
What dplyr function belongs in the missing blank? And what is the equivalent base-R function ? (select one)
    a. Filter(); and the equivalent base-R function is match().
    b. Select(); and the equivalent base-R function is which().
    c. Filter(); and the equivalent base-R function is subset().
    d. Select(); and the equivalent base-R function is subset().

4. Since we are not interested in any other variables and the gender_data dataset is quite large, you might want to get rid of it instead of asking R to keep it stored in memory. What statement do you need to run in R to get rid of the object gender_data? Enter the statement.

Now that you have loaded the data we want to analyze and have familiarized yourself with the structure, it is time to get our hands dirty!

A second exploratory thing to do once we have organized a data set is to get basic summary statistics of the data. Now let's do this! To print summary statistics directly in your console, you

can use any of the basic summary functions in R (`mean()`, `sd()`, `min()`, `max()`, `sum()`…). The basic summary functions take vectors as an input, and output a single value.

For example, if you were interested in obtaining the sample mean of the Adolescent Fertility Rate in 1975, one way of doing this is as follows:

```
mean(teenager_fr$X1975, na.rm = TRUE)
```

5. Why it is necessary to add the option "na.rm = TRUE" to the command? [select all that apply]

    a. The default option of na.rm is set to FALSE. Thus, in case we don't specify this R will try to calculate the mean using all the observations in the data.
    b. This part is necessary since otherwise R would duplicate some of the observations in the data set when it calculates the sample mean. In particular, the observations with missing values would have higher weights than the observations without missing values.
    c. It is not necessary to add this option to the command to obtain the mean of this variable.
    d. Otherwise we will obtain a missing value since not all the countries in the data have information on the adolescent fertility rate in 1975.
    e. This option is necessary since there are missing values in the data set. Thus, when R tries to calculate the mean it assumes that the result is not a number.

To calculate summary statistics for a group of variables there are different commands. The command `mean()` was just to introduce you to the different options available. Now, we invite you to go through the R documentation and explore different commands by yourself.

If you want to store the output as values in your dataset, or if you want to do something more complicated (ex. Generate these by group, or use one of the dplyr summary functions (ex. n_distinct()), you can use any of the basic summary functions as well as others, in combination with `mutate()` and `summarise()` to generate variables in your dataset containing summary values.

Now that you've learnt how to look at and generate summary statistics, answer the following questions.

6. What is the sample mean and standard deviation of the Adolescent fertility rate in 1960? (round to 2 decimal places)

7. What is the sample mean and median of the Adolescent fertility rate in 2000? (round to 2 decimal places)

8. **True or False?** From the values that you have calculated above we can conclude that the Adolescent Fertility Rate has had a permanent decreasing trend from 1960-2000, and that the dispersion of this variable has decreased over time.

    a. True
    b. False

Now, we are interested in plotting the evolution of the Adolescent Fertility Rate from 1960 to 2015. In addition, we are interested in having different information in the same plot. First, we want to plot the sample mean of all the data set, but also we want to add more information such as the rate for low, middle and high income countries (an indicator for country code is stored in the variable "Country.Code").

Inspect this variable to get a sense of what it contains. Note that it includes indicators for both countries, regions, and income group. Since we are only interested in the trends by income group, we want to filter the data to contain only the fertility rate for high, middle, and low income countries.

9. Use the dplyr `filter()` command and the logical `%in%`, to keep only the relevant Country.code observations in `teenager_fr`. Make sure you name the new dataset "byincomelevel". Enter your line of code below, without any spaces:

Notice, there are still two problems with the resulting data:

1. It contains additional variables that we don't need or are meaningless at this level of aggregation.
2. It is not organized in a very intuitive way. A more natural way to organize this data, and prepare it for plotting, is to have each observation represent either a year or a country group-year, and each of the columns represent either the fertility rate for a given group, or if the data is at the country-group year level , then just the fertility rate.

10. Suppose you decide you prefer to have one observation income group-year. The dplyr command `gather()` can help you achieve this. Look up the command in the help files. Select the set of arguments that belong in the blanks below:

```
plotdata_bygroupyear <- gather(byincomelevel, □, □,
                               □) %>%
  select(Year, Country.Name, Country.Code, FertilityRate)
```

    a. Country.Code; FertilityRate; X1960:X2015
    b. Year; FertilityRate; Country.Code
    c. Year; FertilityRate; X1960:X2015
    d. Country.Code; Year; FertilityRate
    e. Year; Country.Code; FertilityRate

    (NOTE: Depending on your operating system, you may encounter an encoding error when trying to reference the variable "Country.Name", if that is the case, add the following line before running the code:

```
byincomelevel<-colnames(byincomelevel)[1]="Country.Name")
```

11. Suppose you take a look at the data, and change your mind- you decided you prefer to look at the data at the year level, and have the fertility rates for each income-group as separate variables. The dplyr command `spread()` can help you achieve this. Look up the command in the help files. Select the set of arguments that belong in the blanks below:

```
plotdata_byyear <- select(plotdata_byyear, Country.Code,Year , FertilityRate) %>%
  spread(□,□)
```

    a. Year; FertilityRate
    b. Country.Code; FertilityRate
    c. Year; Country.Code
    d. FertilityRate, Country.Code
    e. Country.Code; Year
    f. FertilityRate, Year

12. True or False? The select statement in the code for question 11 is redundant, since we already selected the variables we wanted in generating "`plotdata_byyear`".
    a. True
    b. False

13. Good news. We are finally ready to plot the data! Let's begin by plotting the fertility rate over time, separately for each income level. To do this, we can use the basic ggplot syntax Prof. Duflo explained in lecture.
    Let's start by trying to generate this plot using the `plotdata_bygroupyear` tibble we generated earlier. Here is the code to generate this plot:

```
ggplot(plotdata_bygroupyear, aes(x=□ , y=□ ,
                                 group=□)) +
  geom_line()
```

    As you can see, it's missing some arguments. Select the set of arguments that belong in the blanks to generate the desired plot.
    a. Year; FertilityRate; Country.Code
    b. Year; Country.Code; FertilityRate
    c. Country.Code; Year; FertilityRate
    d. Country.Code; FertilityRate; Year

14. It would be nicer if the different plot lines had different colors. You can adding the argument , `color=Country.Code` to the code you generated in question 13. Where do you need to specify this argument? Select one of the roman numbered blanks in the code below, to replace with ",`color=Country.Code`" or "`color=Country.Code` " in order for each of the lines to have a different color:

```
ggplot(plotdata_bygroupyear, aes(x=□ , y=□ ,

                                 group=□ I) II) + III

  geom_line(IV)V
```

a. I
b. II
c. III
d. IV
e. V

15. It is good practice to include titles in your plot, to do this, look up the ggplot "`labs()`". Select one of the roman numbered blanks in the code below, to replace with (possibly preceded by "`.`" or "`+`")"`labs(title='Fertility Rate by Country-Income-Level over Time')`":

```
ggplot(plotdata_bygroupyear, aes(x=□ , y=□ ,

                                 group=□ I) II) + III

  geom_line(IV)V
```

a. I
b. II
c. III
d. IV
e. V

16. One more thing we could improve in this plot is the x-axis labels, first, by removing the leading "X", and second by storing them as numeric so ggplot can use it's "optimal" scaling to make a prettier plot, instead of having a label for each year. To do this, we can transform the Year variable using dplyr's mutate function, and a combination of the functions `as.numeric()` and the stringr package. Try to figure out a few ways to do this. Which of the following statements are equivalent, and can be used in the missing blank below in the code below, to complete it?

```
plotdata_bygroupyear <- mutate(plotdata_bygroupyear, Year=
as.numeric(str_sub(Year,□) ))
```

a. str_sub(Year,-4)
b. str_sub(Year,2,5)
c. str_replace(Year,"X","")

d. All of the above

.

17. Which of the following statements can you conclude from the plot?

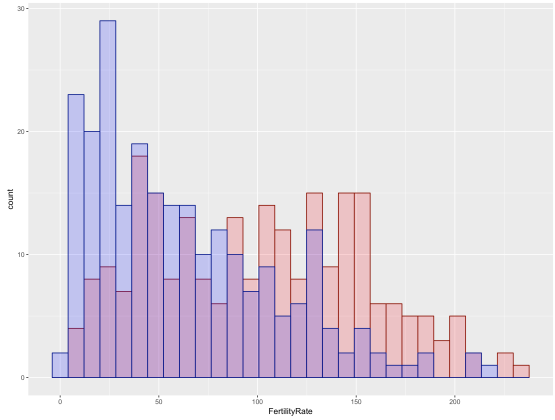
a. The average of the rate using all the data is always below the rate of high and middle income countries, and below the one for low income countries.
b. While the rate for high income countries has presented a decreasing trend in all the period, the rate for low income countries is barely steady until the mid-nineties. From there onwards the rate has decreased significantly.
c. The gap between high and middle income countries is lower in 2014 than in 1960, while the gap between low and middle income countries is actually larger.
d. Since the mid-nineties the rate for low income countries has decreased more than for high and middle income countries.

Now, we are not going to consider the trends of the different categories over the years. Instead, we are going to compare how the distribution of the Adolescent Fertility Rate is different between 1960 and 2000. To do this, we want to plot a histogram of the two variables.

The following code in R plots the histogram of these two variables in the same graph. Please take a look at the code and try to understand what it is doing.

```
ggplot(histdata_twoyears, aes(x=FertilityRate)) +
  geom_histogram(data=subset(histdata_twoyears,   Year=="X1960"),
color= "darkred", fill="red", alpha=0.2)+
  geom_histogram(data=subset(histdata_twoyears,   Year=="X2000"),
color= "darkblue", fill="blue", alpha=0.2)
ggsave("hist.png")
```

Here is the figure that this code has produced:



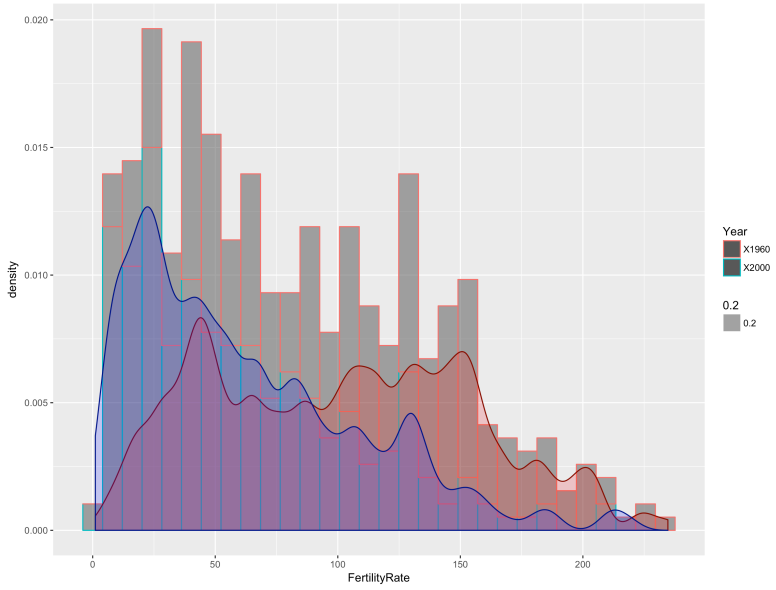
18. What does the argument “alpha” dictate?
    a. The width of the bins.
    b. The width of the outline of the bins.
    c. The extent to which the plot colors are different.
    d. The level of transparency in the color of bins.

19. As you can see we have certain number of bins in the figure, go to the R documentation and look for the option in the command hist, that will allow you to change the number of bins in the figure.

Now, we are going to add some kernels to the histogram. The kernels were done using the command density, and all the default options in R. Again, take a look at the code, run it on your computer and try to understand what it is doing.

```
ggplot(histdata_twoyears, aes(x=FertilityRate, group=Year,  color=Year,
alpha=0.2)) +
  geom_histogram(aes(y=..density..))+
  geom_density(data=subset(histdata_twoyears,  Year=="X1960"),  color=
"darkred",  fill="red" , alpha=0.2, bw=5)+
  geom_density(data=subset(histdata_twoyears,  Year=="X2000"),  color=
"darkblue",  fill="blue", alpha=0.2, bw=5)
```

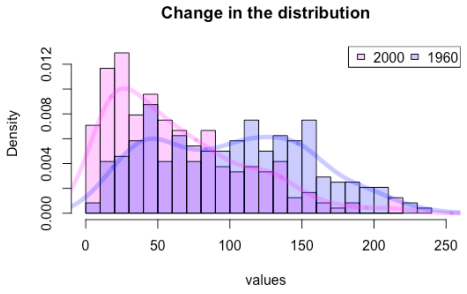
The figure that is produced by running this code is presented next:



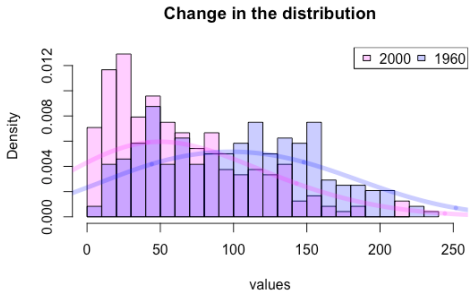
20. As it was stated before, the plot was done using the default options in R. For the kernel, the default option is to use gaussian. There are other options that the user can state when running the *density* command in R. Of the following list, which of the following weighting function is <u>not</u> bell-shaped? In other words, which one doesn't underweight observations at the boundaries of each bin.
    a. gaussian
    b. epanechnikov
    c. rectangular
    d. triangular
    e. biweight
    f. cosine
    g. optcosine

21. The following plots were done changing the bandwidth of the kernel function in R. Which one of them was done with the largest bandwidth?

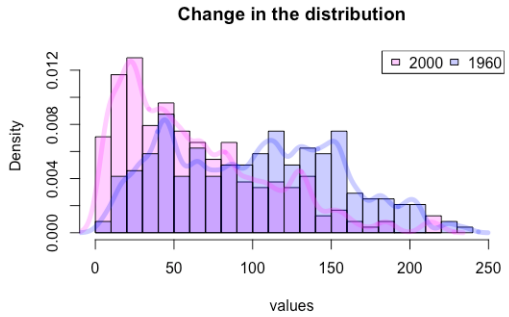
    a. It is not possible to tell just by looking at the figure.
    b.



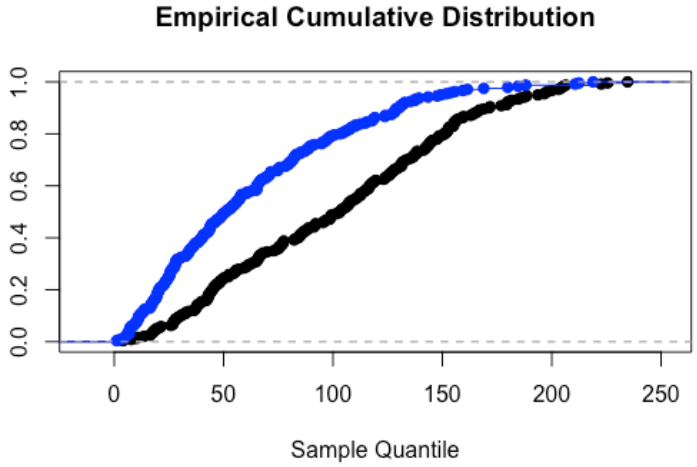
    i.
    c.



    i.
    d.



One of the things that professor Duflo also discussed in the lecture, was the construction of the ECD. The following figures shows the ECD for the Adolescent Fertility Rate in the World in 1960 and in 2000. However, as you can see the person who made the graph forgot to properly label it.



22. Can you infer from the histograms that were plotted before, which one corresponds to the Adolescent Fertility Rate in 2000 and which one to the same indicator in 1960. [select all that apply]

    a. Blue corresponds to 2000
    b. Black corresponds to 2000
    c. Blue corresponds to 1960
    d. Black corresponds to 1960
    e. It is not possible to tell from the plot

23. Can you infer from the figure, whether the distribution used to construct the black series satisfy the First Order Stochastic Dominance property over the distribution used to construct the blue series?

    a. Yes

b. No