UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

ASSIGNMENT 1

The goal of this assignment is to familiarize you with the fundamentals of R, data visualization, basic data wrangling, and exploration. The first part of this assignment is quite easy to complete because you will have a chance to check the solution at each step, interactively. However, I strongly recommend you do **not** check a solution before you attempt to answer a question. Otherwise, you will not learn, and you will struggle with the rest of the course. So, please!

If you don't cut corners by checking solutions even before you attempt, this assignment will take quite some time depending on how familiar you are with R. Please plan ahead and start early.

Assignment Instructions (Part I)

Please complete the following RStudio Cloud Primers and take the screenshots showing both the completion of each Primer and your username together. You can take screenshots on a PC using the Snipping Tool (Mac users, see this). A sample submission for Part I is on the last page.

Before you click on the following links and start working on the assignment, please visit rstudio.cloud and log in first. If you don't log in, your name will not come up in the screenshots.

This assignment has five mandatory parts:

1. Data Visualization Basics
2. Programming Basics
3. Working with Tibbles
4. Isolating Data with dplyr
5. Deriving Information with dplyr

and two highly recommended sections[1]:

1. Visualize Data
2. Tidy Your Data

and two more recommended sections:

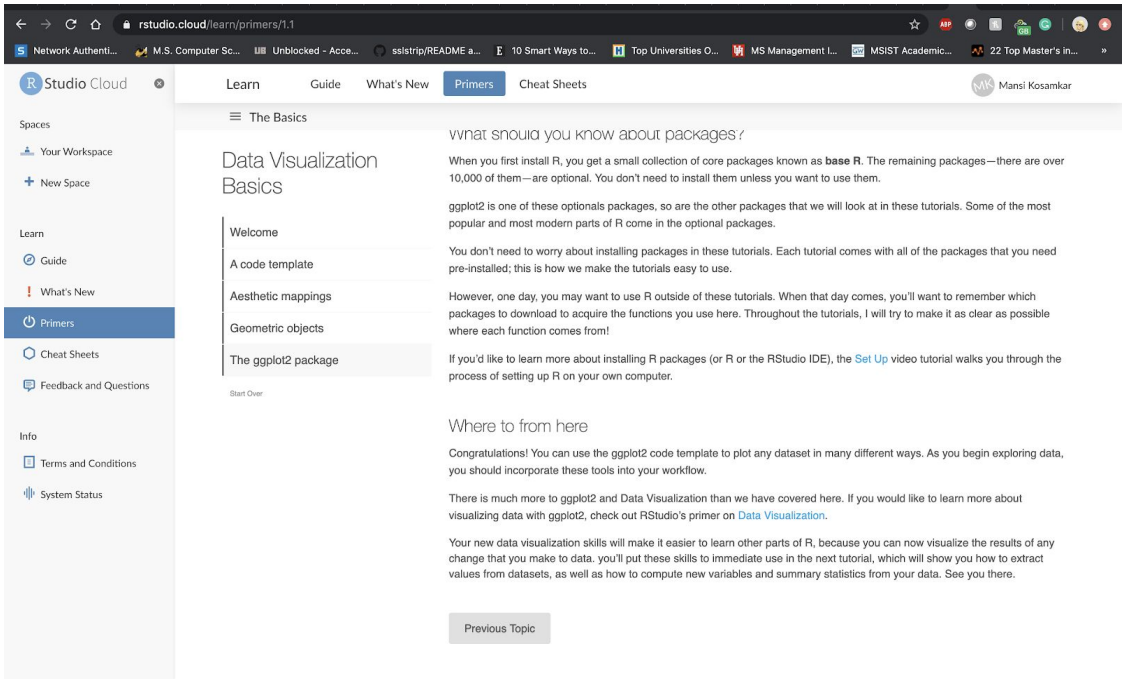
1. Iterate
2. Write Functions

[1] In the past, students who completed the recommended sections increased the probability of getting an A by 39%, on average. Don't attempt to finish them all at once. You will **not** submit the optional sections in your assignment.

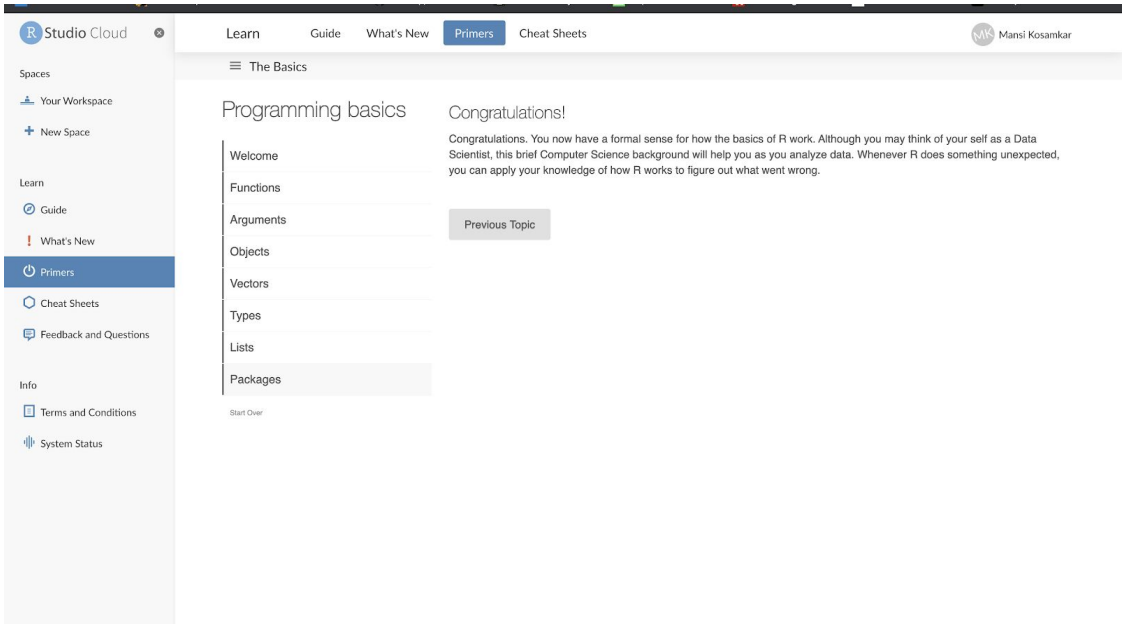
Submission for Part I

Your name and UID: Mansi Kosamkar (116904988)

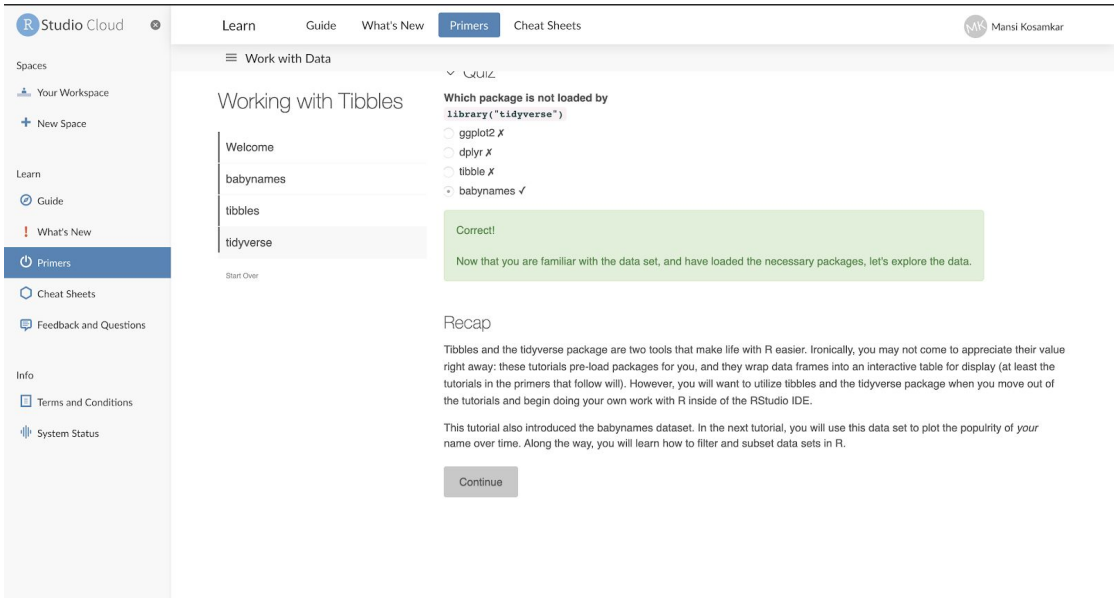
1. Data Visualization Basics



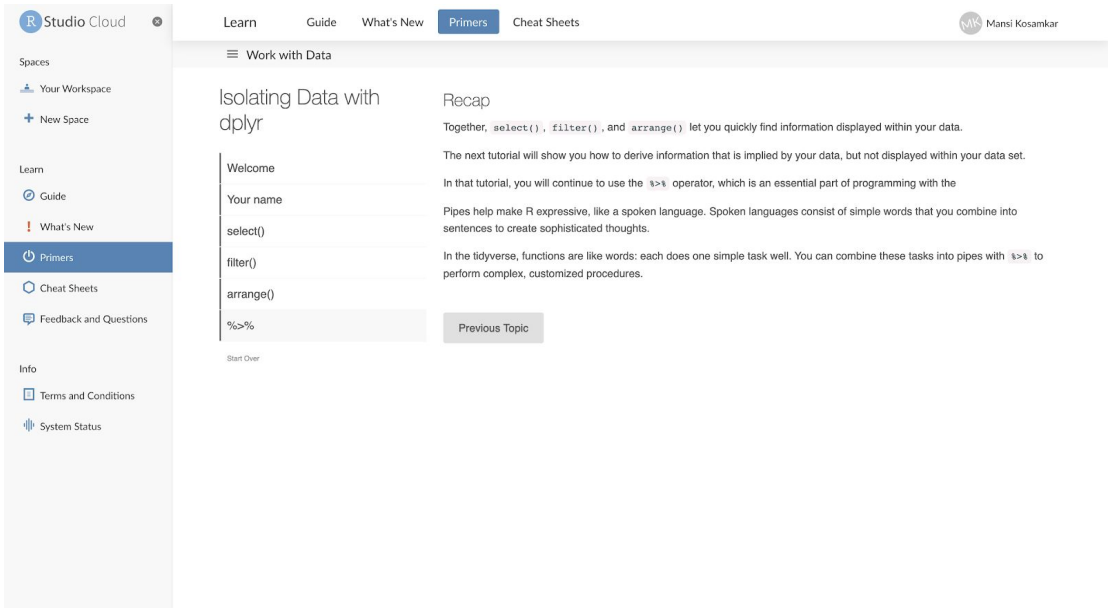
2. Programming Basics



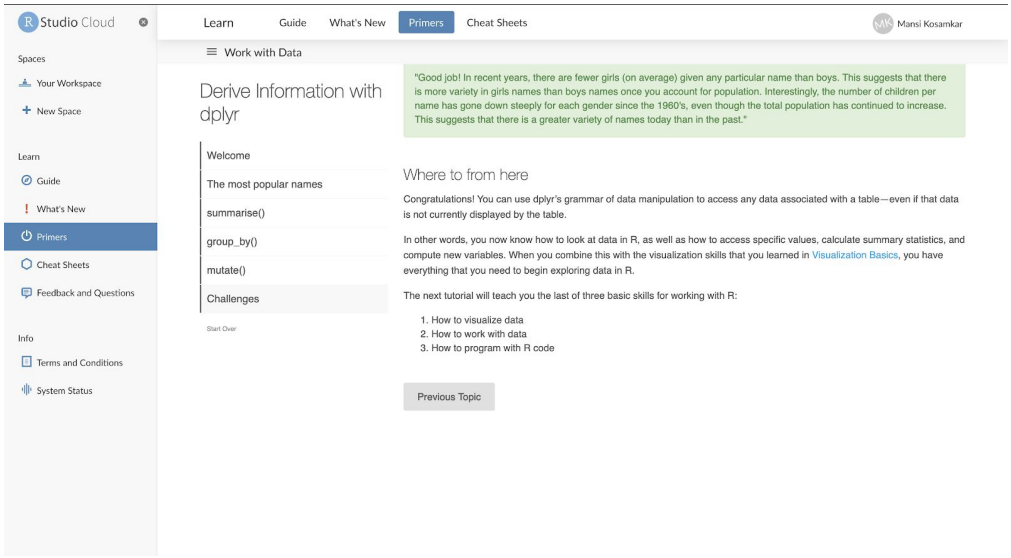
3. Working with Tibbles



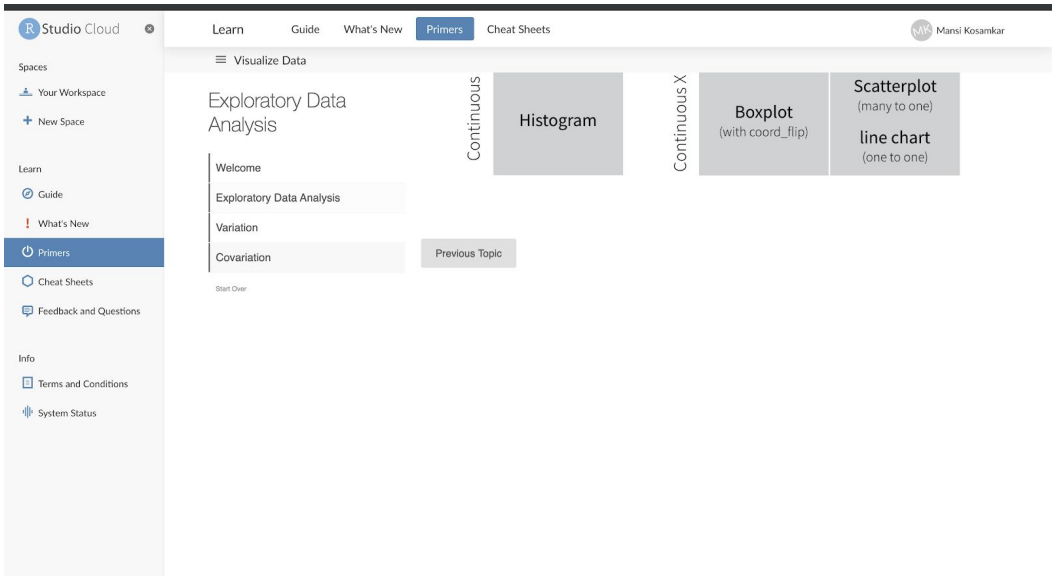
4. Isolating Data with dplyr

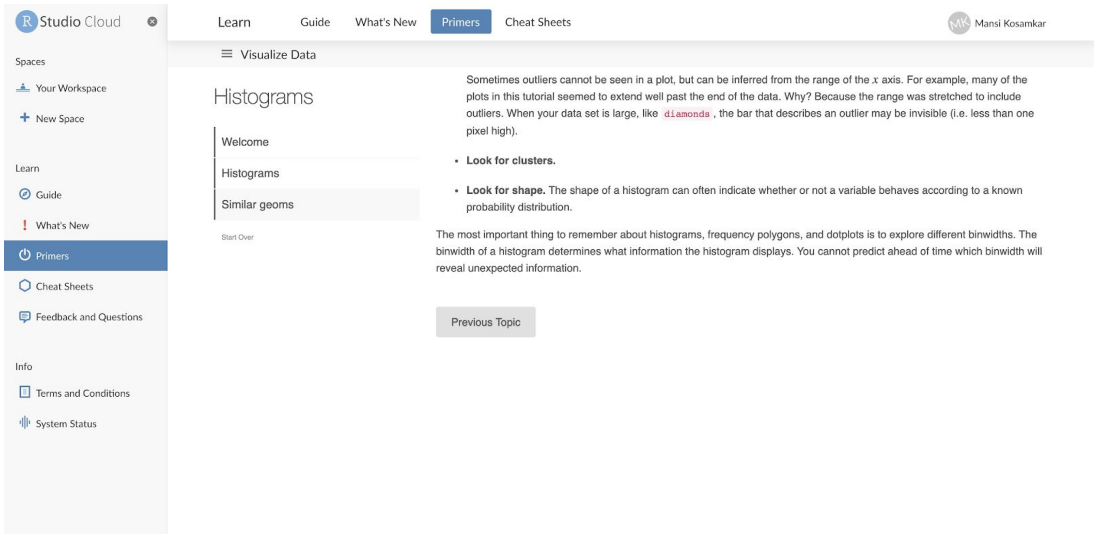


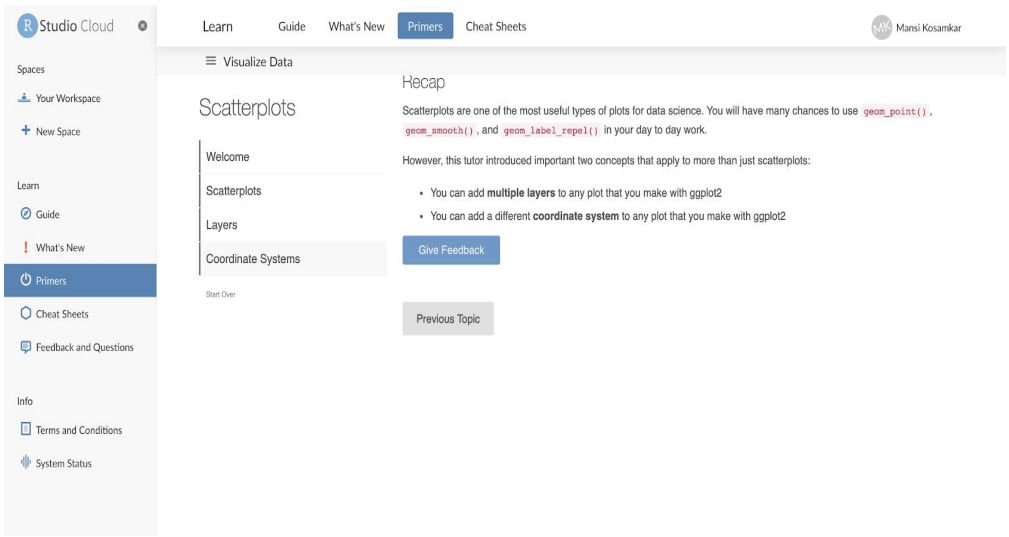
5. Deriving Information with dplyr

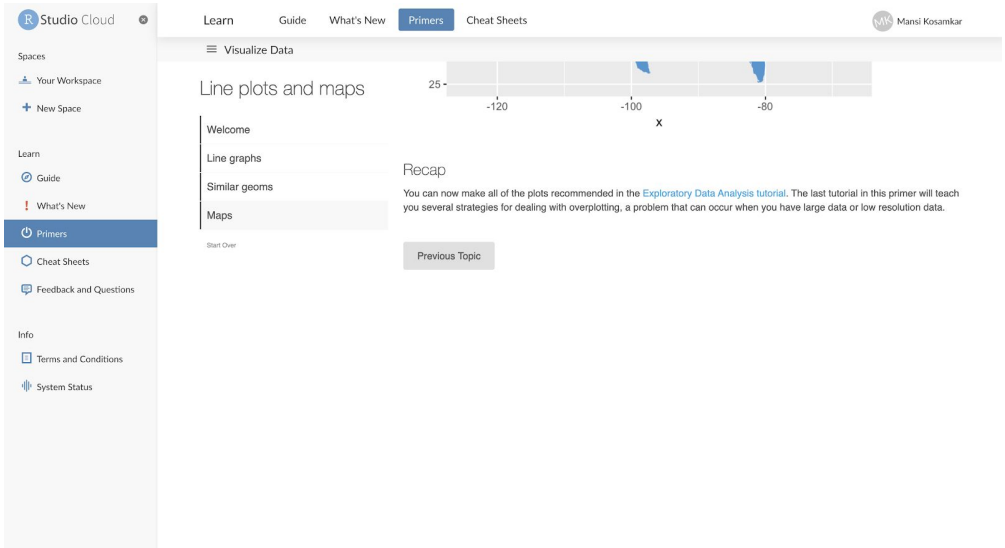


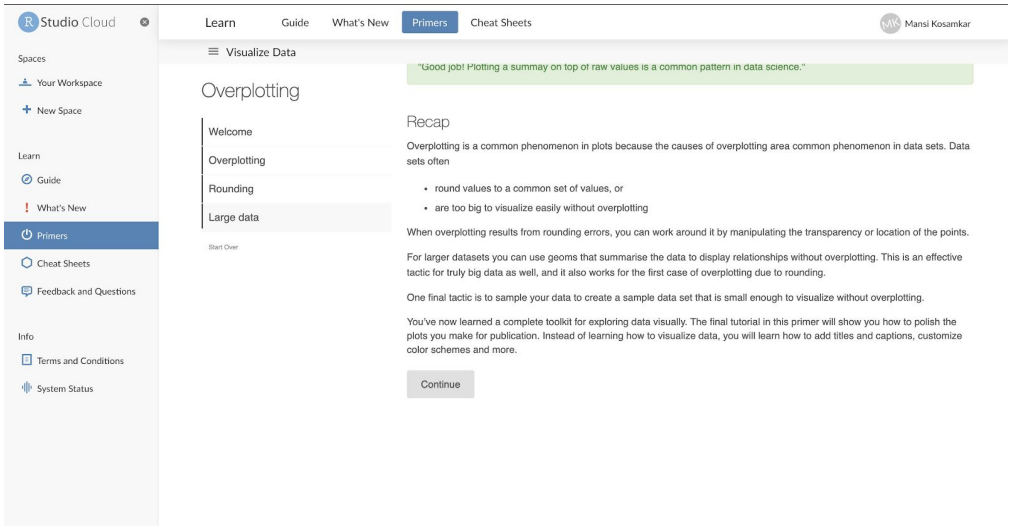
6. Visualize data

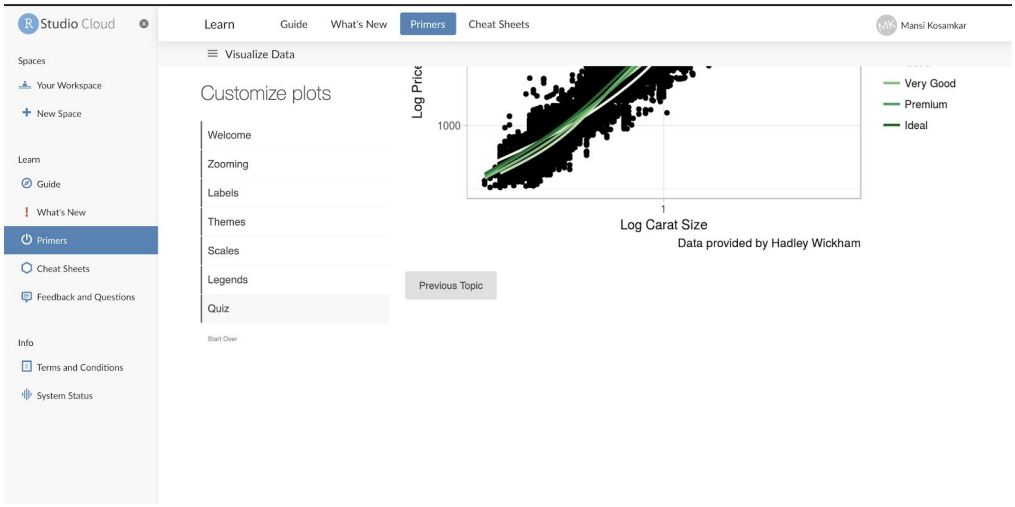



Studio Cloud
Learn
Guide
What's New
Primers
Cheat Sheets
Mansi Kosamkar
Spaces
Your Workspace
New Space
Learn
Guide
What's New
Primers
Cheat Sheets
Feedback and Questions
Info
Terms and Conditions
System Status
Visualize Data
Histograms
Welcome
Histograms
Similar geoms
Start Over
Sometimes outliers cannot be seen in a plot, but can be inferred from the range of the x axis. For example, many of the plots in this tutorial seemed to extend well past the end of the data. Why? Because the range was stretched to include outliers. When your data set is large, like diamonds, the bar that describes an outlier may be invisible (i.e. less than one pixel high).
Look for clusters.
Look for shape. The shape of a histogram can often indicate whether or not a variable behaves according to a known probability distribution.
The most important thing to remember about histograms, frequency polygons, and dotplots is to explore different binwidths. The binwidth of a histogram determines what information the histogram displays. You cannot predict ahead of time which binwidth will reveal unexpected information.
Previous Topic


Studio Cloud
Learn
Guide
What's New
Primers
Cheat Sheets
Mansi Kosamkar
Spaces
Your Workspace
New Space
Learn
Guide
What's New
Primers
Cheat Sheets
Feedback and Questions
Info
Terms and Conditions
System Status
Visualize Data
Scatterplots
Welcome
Scatterplots
Layers
Coordinate Systems
Start Over
Recap
Scatterplots are one of the most useful types of plots for data science. You will have many chances to use geom_point(), geom_smooth(), and geom_label_repel() in your day to day work.
However, this tutor introduced important two concepts that apply to more than just scatterplots:
You can add multiple layers to any plot that you make with ggplot2
You can add a different coordinate system to any plot that you make with ggplot2
Give Feedback
Previous Topic


Studio Cloud
Learn
Guide
What's New
Primers
Cheat Sheets
Mansi Kosamkar
Spaces
Your Workspace
New Space
Learn
Guide
What's New
Primers
Cheat Sheets
Feedback and Questions
Info
Terms and Conditions
System Status
Visualize Data
Line plots and maps
Welcome
Line graphs
Similar geoms
Maps
Start Over
25
-120
-100
-80
x
Recap
You can now make all of the plots recommended in the Exploratory Data Analysis tutorial. The last tutorial in this primer will teach you several strategies for dealing with overplotting, a problem that can occur when you have large data or low resolution data.
Previous Topic





Assignment Instructions (Part II)

In this part of the assignment, you will apply what you learned in the first part. You will work on the Gapminder dataset, which is available as a library in R (how convenient!). You can read more about the source of the dataset here, and enjoy this talk from 2007 if you have not yet done so. See How to submit labs/assignments to learn how to submit the second part of the assignment.

Use the following R libraries in this assignment: tidyverse, tidymodels, plotly, skimr, gapminder

Data Overview

1. country: Name of the country
2. continent: Name of the continent a country belongs in
3. year: Year for which the data is collected

4. lifeExp: Average life expectancy of the people in a country in a year
5. pop: Total population of a country in a year
6. gdpPercap: GDP per capita of a country in a year

Questions

1. **Create an R Notebook**
    a. Start with the assignment template and set your working directory using `setwd`
    b. From here on, use a new chunk for each question to make your code readable
2. **Load the data**
    a. In a new chunk, load the `gapminder` library, and use this line to create dfGap: **`dfGap <- gapminder`**
3. **Explore the data**
    a. Use the `skim` function on the `dfGap` dataframe to get summary statistics in a nice format. I suggest you use the widest screen possible for the best reading.
    b. Filter `dfGap` for the year 2007 and sort it in descending order of life expectancy. Don't forget to use pipes!
        i. What are the names of the countries with a life expectancy over 81?

    Ans. Japan, Hongkong, Iceland, Switzerland, Australia.
    c. Add a calculated column `totalGDP` to `dfGap` showing the total GDP per country, filter the dataframe for 2007, and sort in descending order for `totalGDP`. If you like, save the new dataframe as a new one for repeated use.
        i. What are some names of the countries with the top levels of total GDP?

    Ans. United States, China, Japan, India, Germany
        ii. Which ones of these countries overlap with the countries from **3-b**?

    Ans. Japan
        iii. What if you selected only the two columns country and gdpPercap and sorted the dataframe in descending order for `gdpPercap`? Do you observe more of an overlap now? What do you infer from this difference?

    Ans. Yes. Hong Kong, Switzerland and Iceland. It shows that the higher the GDP, the higher the life expectancy and there is a high positive correlation.
    d. Filter `dfGap` for 2007, group it by continent, and then calculate the median life expectancy and *median* total GDP (so you need to have totalGDP already). Remember, you will pipe the filtered and grouped dataframe into `summarize()` to get the medians. Then, sort it in descending order for the median life expectancy. Before you sort it, don't forget to use `ungroup()` to ungroup.
        i. What continent has the highest median of life expectancy?

    Ans. Oceania
        ii. Does it seem to be correlated with the median total GDP

    Ans. No, it doesn't seem to be correlated. This can be concluded from the similar values of Asia and America for median life expectancy but significantly different values for Median GDP.
4. **Visualize the data**
    a. Now that you have explored the relationship between life expectancy and totalGDP in a table format, let's also visualize it to see a bigger picture.

i. Create a scatter plot to understand the relationship between life expectancy (y-axis) and totalGDP (x-axis) in 2007. Does this plot help?

Ans. No, it does not help explain any relationship between life expectancy and totalGDP

ii. Copy the same code, but this time also filter for countries with a totalGDP of over a billion (use the scientific notation 1e+12). What about now?

Ans. It shows that countries with around the same GDP has similar life expectancy with a few anomalies.

iii. Copy the same code, and add labels this time. Do you see a cluster now? What are the names of the countries that are outside of the cluster?

Ans. Yes, I see a cluster now. The names of the countries that are outside of the cluster are India, China, United States.

iv. Here is a pro tip. The labels you used in (iii) overlap and hide the points. This causes poor visibility. Install and load the `ggrepel` library. After that, copy the same code and use `geom_label_repel()` function instead of `geom_label()`. Does it look better now? Describe what has changed.

Ans. The labels correlating to a point are pointed to by a straight line so as to clearly show the points.

v. Copy the same code. This time, add a color for the continent. What are the continents that are missing from your visual? Why do you think so?

Ans. Australia, Africa and Antarctica are missing from the visual. This maybe because these countries are not big economic power houses.

b. You have an idea about the relationship between life expectancy and totalGDP even though you have not tested it statistically. Now, let's examine a more realistic relationship between life expectancy and gdpPercap (GDP per capita). Plot life expectancy (y-axis) against gdpPercap (x-axis) for 2007, add a smoothed line (no need to define any parameters, use the defaults). What do you observe about the overall relationship? Don't use any labels, just focus on the aggregate.

Ans. The life expectancy keeps increasing with GDP per capita.

c. Now let's find out the variations in life expectancy across different continents. Create box plots for each continent (in the same plot) and add a title this time. What do you observe? Describe your observations and answer the questions:

i. Which continent has the highest median life expectancy?

Ans. Oceania has the highest median life expectancy.

ii. Which continent has the largest range of life expectancy?

Ans. Asia has the largest range of life expectancy.

iii. Save your plot as `boxPlotsForAll` and put it into the `ggplotly()` function. More useful, right? Report the actual medians per continent by reading from the new interactive plot `ggplotly()` has created for you.

Ans. Africa-47.79 , Americas- 67.05, Asia- 61.79, Europe- 72.24, Oceania-73.66.

d. Finally, it is time to create a more advanced (and likely more helpful) plot. Create a line plot to show how **median GDP per capita** by continent changes **over time**. [Hint: For the continents, use the color parameter]. Describe what you observe.

i. What continents have a clearer trend than others? Why do you think so?

Ans. Oceania, Europe and Africa have a clearer trend than others. This is because the median GDP per capita increases faster with the year than others.

ii. Change the summary metric from median to mean. What has changed? Why do you think so?

Ans. Asia overtook the Americas when we changed the summary metric from median to mean. This is because taking the mean skews the result because of outliers or extreme values. Whereas, the median is a more centralized value.

iii. Finally, don't you think these plots would be much more useful in plotly? Pick one and save it as `gdpOverTime` and call `ggplotly()` on it. You can now read the actual GDP values per year. What are some of the breakthrough years (steep changes) for GDP in different continents?

Ans. The 1990s and 2000s show significant increase in the GDP per capita of Oceania and Europe. Asia saw some growth in the period between late 1960s and early 1970s and the 2000s.