Data Visualization Using ggplot2

Developed by Tieanna Graphenreed and Colleen Nugent
Taught by Tieanna Graphenreed and Colleen Nugent
Economics of Financial Crisis
Professor Yaprak Tavman
Fall 2021

Northeastern University
NULab for Texts, Maps, and Networks

Feel free to ask questions at any point during the presentation!

Workshop Agenda

- Review how to work with data using R & RStudio
- Learn best practices for data cleaning using R & RStudio, and discuss some examples of best practices
- Learn how to download, install, and use the ggplot2 package for data visualization

Slides and handout available at: **http://bit.ly/diti-fall2021-tavman**

Feel free to ask questions at any point during the presentation!

A Quick Refresher on R: The Basics

R is an open-source, free software for data manipulation, computing and graphics display. **RStudio** is the integrated development environment (IDE) in which R operates.

R is quickly becoming the standard for statistical use because:

- It holds large data sets
- R makes it easy to transform files into HTML, CSV, Word, PDF and other accessible forms.
- R is run by its users, so it changes and adapts quickly to what people need

General Principles for R

- Some good **general principles** for R include:
    - keeping an eye on file paths
    - remembering to check the working directory
    - and verifying which project space you’re working in

You might see these tips come up again during this presentation because they often overlap with some best practices for **data cleaning** and **data visualization**.

Feel free to ask questions at any point during the presentation!

Best Practices for Data Cleaning in RStudio

Northeastern University
NULab for Texts, Maps, and Networks

Feel free to ask questions at any point during the presentation!

Best Practices for Data Cleaning

- **Read the documentation for any functions you are using** to make sure that you're using them appropriately.
- **Carefully familiarize yourself with the dataset** and make sure that you are considering its structure as you are manipulating it in R**.**
- **Saving:**
  - **Regularly save** your projects, code, and data.
  - Always **save an unmodified version** of your data before you begin making any changes.

Best Practices for Data Cleaning

- **Documentation:**
  - **Document all changes** that you make to your data. Different kinds of documentation will be necessary at different levels, but all documentation helps you prepare to share the results of your research.
  - **Establish consistent conventions for naming variables** and keep track of when you overwrite them or switch to a new data source**.** Failing to keep track might mean you lose a lot of ground.
  - **Make a plan to address irregularities in the data** by developing a documentation system for error-management and instructions for troubleshooting errors.
- **Review the results** after you make any global changes.

Feel free to ask questions at any point during the presentation!

Quick Tips to “Clean” Messy Data

False conclusions because of incorrect or “dirty” data can inform poor data management strategies and decision-making. You want your data to stand up to scrutiny. Here are some strategies for good data building and cleaning:

- **Standardized your content and naming conventions**.
- **Handle missing data/empty data cells.**
- **Fix structural errors**. Look for inconsistencies like unconventional naming conventions, typos, and incorrect capitalization.
- **Convert numbers** stored as text to numeric fields.
- **Ask yourself:** *Does your data make sense? Is it usable?*

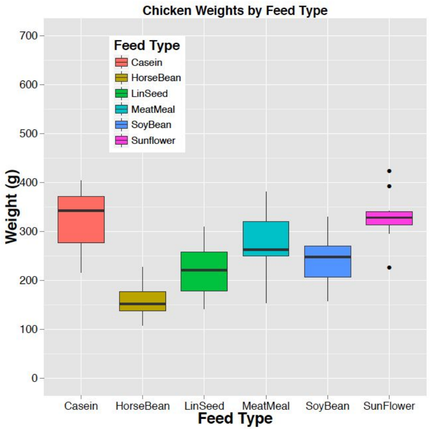
Data Visualization using ggplot2 in RStudio

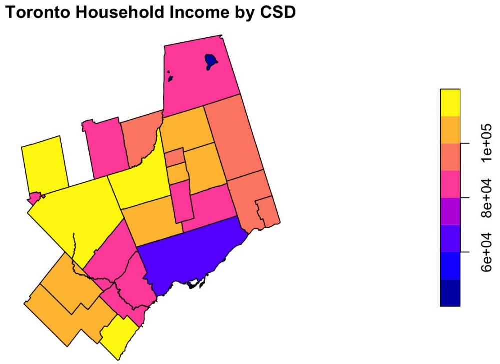
Northeastern University
NULab for Texts, Maps, and Networks

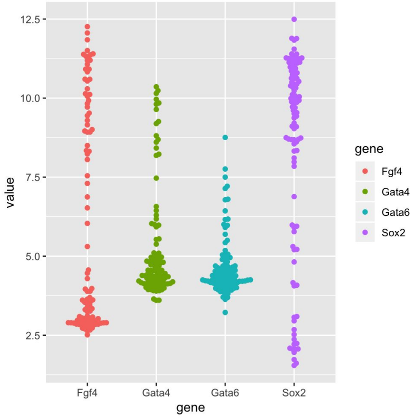
Feel free to ask questions at any point during the presentation!

Using R for Data Visualization

R can analyze or make data, but it can also create really effective data visualizations using its tidyverse package for data visualization **ggplot2**.







Many of the graphs and tables you see in *The Economist* are made with R!

Feel free to ask questions at any point during the presentation!

Basics of ggplot2

ggplot2 is a free, open-source data visualization package meant for use with the programming language R.

This software is known as a system for creating graphics, based on the Grammar of Graphics.

ggplot() is the specific function used in RStudio to turn data into compelling and persuasive visualizations.

Useful Vocabulary for ggplot2

Function: functions in R are code that performs a specific task.

Argument: the inputs provided to functions. Functions can have multiple arguments, or none at all.

Aesthetics: descriptions of how variables are mapped to visual properties or, “aesthetics”; uses the aes() function. Used to dictate the appearance of a geom.

Aesthetic Mapping: a set of aesthetic mappings. Usually a sequence of aesthetic-variable pairs—e.g., aes(x= , y=, colour=)

Useful Vocabulary for ggplot2

Geom: the function used to select the type of geometric object for data visualization. Geoms will always have aesthetics as parameters to control how the visualization is displayed. ggplot2 supplies numerous geom functions for almost every graphing need.

Layer: in ggplot2 layers are used to create graphs. Layers define geometry, define and set scales, change styles, and more.

Scale: scales control the details of how data values are translated to visual properties. You can use scale functions to tweak the appearance of x/y axes, legend keys, limits, and even color.

Feel free to ask questions at any point during the presentation!

Some questions to consider:

- **In what ways have you manipulated your data?**
- **What aspects of your dataset are best suited to data visualization?**
- **What type of plot is best suited to visualize your data?** Think about what your imagined audience would find most useful, and what will be most effective for your project long-term.

Feel free to ask questions at any point during the presentation!

Best Practices for Data Visualization

- **Clean your data.** Good data visuals start from well organized data (usually in tabular format). Be sure to choose clear, concise names for your columns that are friendly for exportation and easy to remember.
- **Make an accessibility-forward design.** Consider how font sizes, typeface, color choice, and other design elements might impact your user. Check your handout for resources from *Towards Data Science*.

Demo: Visualizing data using ggplot2

Follow along in the RMD file so you can practice the commands.

Feel free to ask questions at any point during the presentation!

Making a line graph using ggplot2

A line graph is a type of visualization that can be used to show **change over time**, and other relationships.

Line graphs consist of two axes: x-axis (horizontal) and y-axis (vertical).

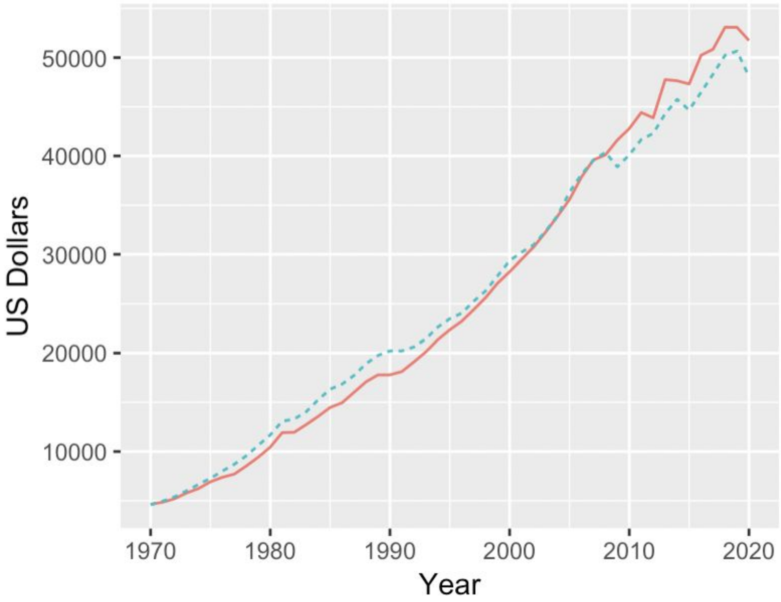
To make a line graph with ggplot2, you will use the function geom_line( ).

Feel free to ask questions at any point during the presentation!

RStudio demo: Making a Line Graph

Use the trial-dataset.csv file and **follow the demo instructions in the RMD file**.

1. To **start**, type the ggplot() function and select the appropriate axes.
   - Axes: x=TIME and y=Value
2. **Add** the geom_line() function for line graphs: ggplot(data, aes(x=TIME, y=Value) + geom_line()
3. In order to distinguish between the locations, add more arguments into the geom_line() function: geom_line(aes(linetype=LOCATION, color = LOCATION))



Northeastern University
NULab for Texts, Maps, and Networks

Feel free to ask questions at any point during the presentation!

Making a scatter plot using ggplot2

Scatter plots can be used to show density, trajectory, distribution across specific measurements, and other relationships.

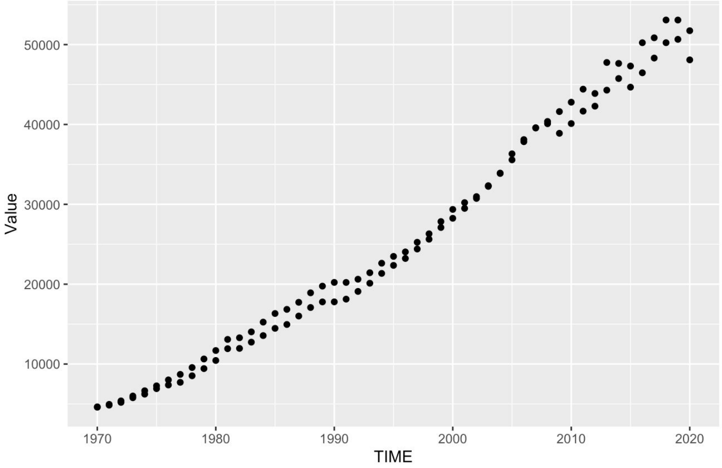
These graphs are great way to visualize the **correlation of different variables** (usually two numeric variables) and other **patterns in data sources**.

To make a scatter plot in RStudio, you will use the function geom_point( ).

RStudio demo: Making a Scatter Plot

Use the trial-dataset.csv file and **follow the demo instructions in the RMD file**. For the scatter plot, you will use the same x and y axes used for the line graph.

1. **Start** with the ggplot() function and axes
2. **Add** the geom_point() function for scatter plots:
   a. Try running this line of code in R: ggplot(data, aes(x=TIME, y=Value) + geom_point()
3. The scatter plot should generate something like the image to the right.
   a. But, as you see the graph is a mass of black dots making it difficult to differentiate between AUS and CAN.



Northeastern University
NULab for Texts, Maps, and Networks

Feel free to ask questions at any point during the presentation!

Setting scales for scatter plots

Setting the scale for your graphic means choosing what aspects of your data will be visualized.

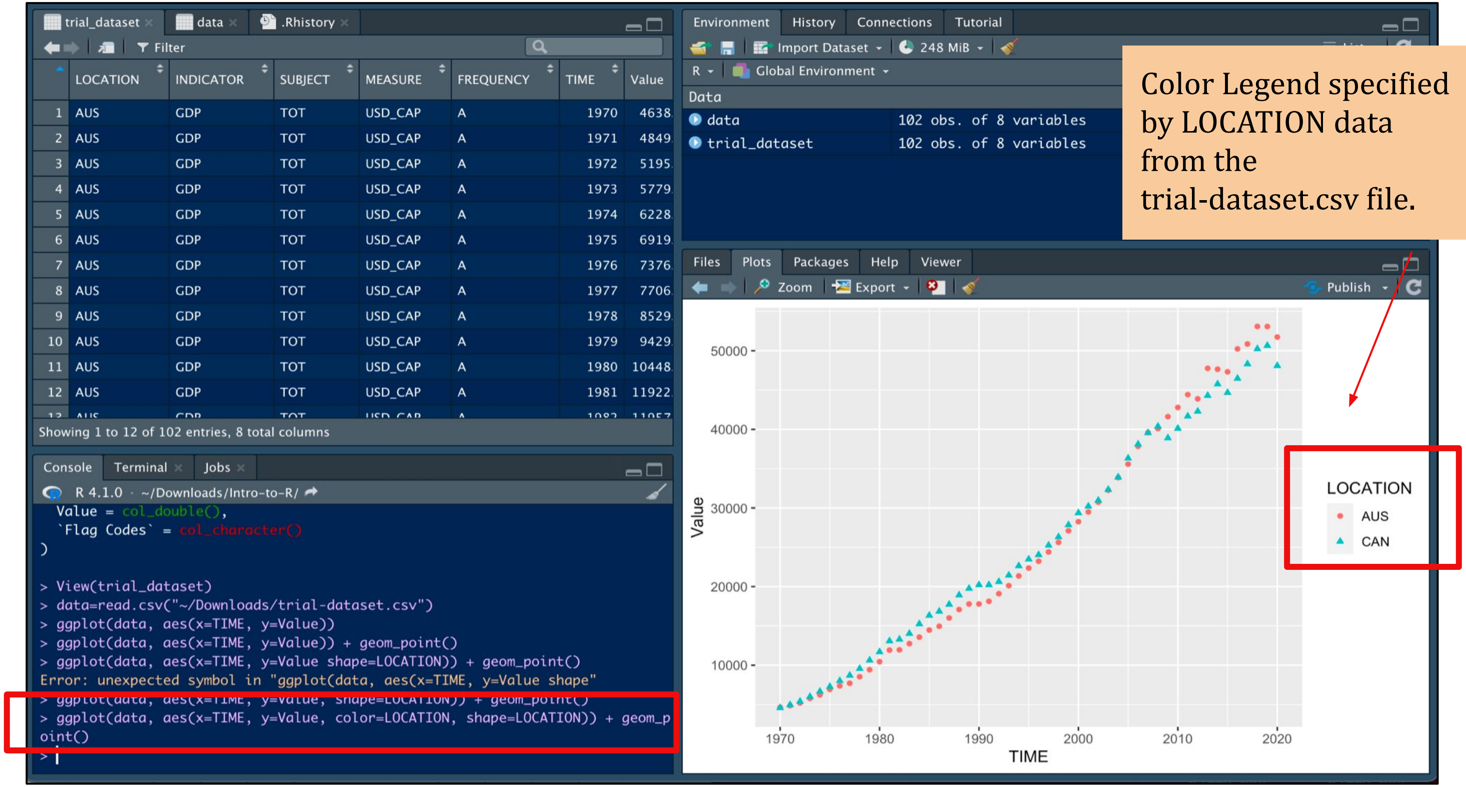
For graphs showing two sets of data, it might be helpful to use different colors and have an associated color legend. To add color to your plot points add the color= argument within the ggplot() function. To generate a legend for your chart, add the shape= argument within the ggplot() function. **Note: Be sure to separate all arguments with commas.**

Try running the following line in RStudio:

```
ggplot(data, aes(x=TIME, y=Value, color=LOCATION, shape=LOCATION)) + geom_point()
```

Setting scales for scatter plots

Scatter plot with color= and shape= arguments added to the ggplot() function.



	LOCATION	INDICATOR	SUBJECT	MEASURE	FREQUENCY	TIME	Value
1	AUS	GDP	TOT	USD_CAP	A	1970	4638.
2	AUS	GDP	TOT	USD_CAP	A	1971	4849.
3	AUS	GDP	TOT	USD_CAP	A	1972	5195.
4	AUS	GDP	TOT	USD_CAP	A	1973	5779.
5	AUS	GDP	TOT	USD_CAP	A	1974	6228.
6	AUS	GDP	TOT	USD_CAP	A	1975	6919.
7	AUS	GDP	TOT	USD_CAP	A	1976	7376.
8	AUS	GDP	TOT	USD_CAP	A	1977	7706.
9	AUS	GDP	TOT	USD_CAP	A	1978	8529.
10	AUS	GDP	TOT	USD_CAP	A	1979	9429.
11	AUS	GDP	TOT	USD_CAP	A	1980	10448.
12	AUS	GDP	TOT	USD_CAP	A	1981	11922.
13	AUS	GDP	TOT	USD_CAP	A	1982	11057

Showing 1 to 12 of 102 entries, 8 total columns

```
Data
data            102 obs. of 8 variables
trial_dataset   102 obs. of 8 variables
```

```
R 4.1.0 · ~/Downloads/Intro-to-R/
  Value = col_double(),
  `Flag Codes` = col_character()
)

> View(trial_dataset)
> data=read.csv("~/Downloads/trial-dataset.csv")
> ggplot(data, aes(x=TIME, y=Value))
> ggplot(data, aes(x=TIME, y=Value)) + geom_point()
> ggplot(data, aes(x=TIME, y=Value shape=LOCATION)) + geom_point()
Error: unexpected symbol in "ggplot(data, aes(x=TIME, y=Value shape"
> ggplot(data, aes(x=TIME, y=Value, shape=LOCATION)) + geom_point()
> ggplot(data, aes(x=TIME, y=Value, color=LOCATION, shape=LOCATION)) + geom_p
oint()
> 
```

Color Legend specified by LOCATION data from the trial-dataset.csv file.

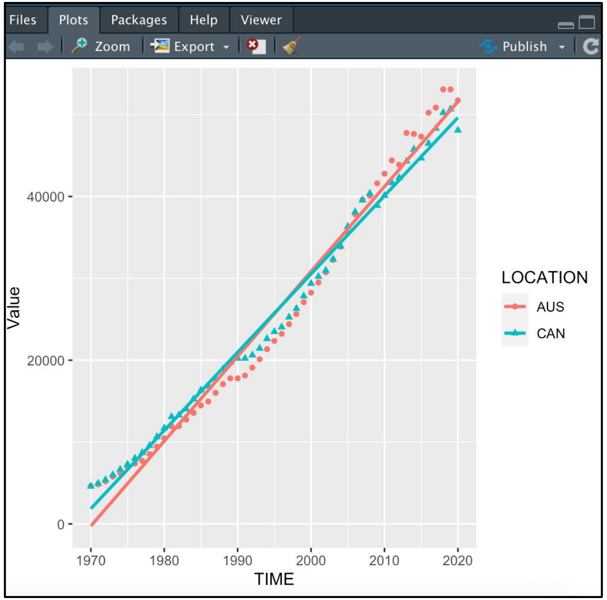
Feel free to ask questions at any point during the presentation!

Adding regression lines to scatter plots

For this course, you might choose to add regression lines to your scatter plot. To do so, you would add the function geom_smooth() to set the regression line. Input the method using method=lm to select a linear regression line.

Set the logical value (confidence interval) using the argument se=

- You should know: se= adjusts the **confidence interval** for your chart.
  A FALSE se can remove shading normally added around the regression line. (see image to the right)



Run this in RStudio with and without the se= argument.

Try this code in R! ggplot(data, aes(x=TIME, y=Value, color=LOCATION, shape=LOCATION)) + geom_point() + geom_smooth(method = lm, se=FALSE)

Northeastern University
NULab for Texts, Maps, and Networks

Feel free to ask questions at any point during the presentation!

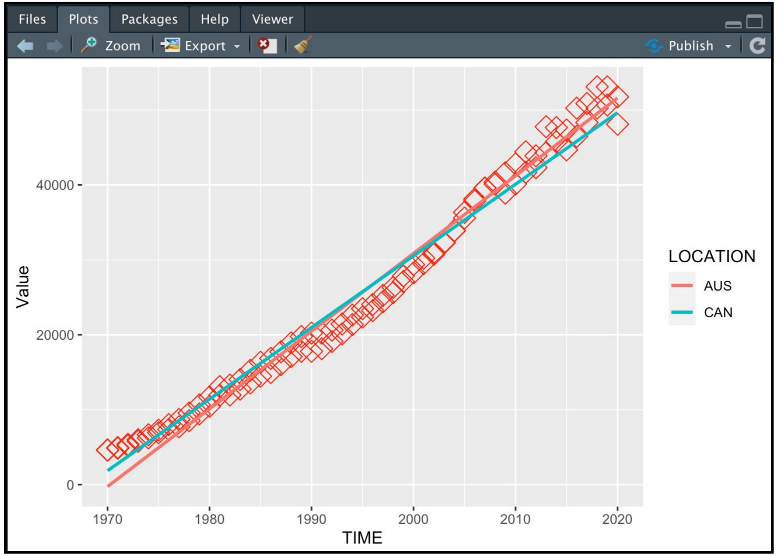
Changing Design Elements: Scatter plot

Typically the software will automatically select the size, shape, and other visual elements of your plot points. However, you can manually control these visual elements if you wish.

To manually change the point color/shape/size use the functions below:

- scale_shape_manual() for point shapes
- scale_color_manual() for point colors
- scale_size_manual() for point sizes

STHDA has a great resource for other adjustments you can make to your scatterplot.



Feel free to ask questions at any point during the presentation!

Thank you!

If you have any questions, contact DITI at nulab.info@gmail.com

Sign up for DITI office hours! https://calendly.com/diti-nu

Developed and Taught by

Tieanna Graphenreed **and** **Colleen Nugent**

Digital Integration Teaching Initiative
DITI Research Fellow

Digital Integration Teaching Initiative
DITI Research Fellow

Slides and handouts available at **http://bit.ly/diti-fall2021-tavman**

A special thanks to Ben Schmidt and his guide on ggplot2 "Visualizing Data: the basics"

Northeastern University
NULab for Texts, Maps, and Networks

Feel free to ask questions at any point during the presentation!