Pre-webinar instructions for
An Introduction to Data Visualization with RStudio

Stephanie A. Santorico, Ph.D.
Dep’t of Mathematical and Statistical Sciences, Human Medical Genetics and Genomics Program, and Dept of Biostatistics & Informatics
University of Colorado Denver
Stephanie.Santorico@ucdenver.edu
Installing R and RStudio

Step 1:
- Download base R for your computer at https://cran.rstudio.com/
- Follow directions to install for your computing system

Step 2:
- Go to the Rstudio download website: https://www.rstudio.com/products/rstudio/download/#download
- Download for your computer under “Installers for Supported Platforms”
- Follow directions to install for your computing system
Step 1:

- Download base R for your computer at https://cran.rstudio.com/

- Follow directions to install for your computing system
Step 2:

- Go to the Rstudio download website: https://www.rstudio.com/products/rstudio/download/#download
- Download for your computer under “Installers for Supported Platforms”
- Follow directions to install for your computing system
Download the R Script and data file if you would like to follow along with my coding.

An Introduction to Data Visualization with RStudio

Stephanie A. Santorico, Ph.D.
Dept of Mathematical and Statistical Sciences,
Human Medical Genetics and Genomics Program,
and Dept of Biostatistics & Informatics
University of Colorado Denver
Stephanie.Santorico@ucdenver.edu

Slides at: https://tinyurl.com/RtutorialSlides1
R Script at: https://tinyurl.com/RtutorialScript1
Data at: https://tinyurl.com/RtutorialData1
Abstract: With the data revolution well upon us, deciphering such a large amount of information can be daunting - requiring the use of statistical and visualization tools. In this webinar attendees will be introduced to the powerful statistical and visualization tool, RStudio. This will include a review of the basic features of RStudio with an demonstration on creating a range of simple plots, and a discussion of the advanced capabilities of RStudio.

Objectives: Participants will

1. Learn the value and function of R and RStudio
2. Read data into RStudio
3. Create a range of simple plots
4. Understand the more advanced capabilities of R and Rstudio
1. Learn the value and function of R and RStudio: *What is it?*

- **What is R?**
  - **FREE** Language and environment for statistical computing and graphics
  - Runs on UNIX, Linux, Windows, and MacOS
  - Allows for user created functions and libraries
  - Easy use of other’s functions and libraries (or packages)

- **What is R Studio?**
  - An integrated development environment (IDE) for R
  - **User perks:** tools for plotting, history, debugging and workspace management.
  - **Coding perks:** a console and syntax-highlighting editor that supports direct code execution

- "**One of R’s strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.**" (from https://www.r-project.org/about.html)
Overview of RStudio

The Global Environment gives access to data and objects that have been read in or created.

Access window to Help, Plots, and more!

Scripts and data sets open here.

Console: commands are executed here.
1. Learn the value and function of R and Rstudio: Compared to...?

- **R versus Excel:**
  - Excel allows for calculations and manipulation of tabular data
  - R provides allows for such calculations as well as more sophisticated analysis
  - Excel provides point and click access to data visualization
  - R, while powerful for statistical computing and graphics, has a bit of a learning curve
  - R can be run through a script, which allows for reproducibility of analyses
  - R provides a variety of functions that allow for user control and customization of data visualizations

- **R versus Tableau:**
  - Tableau provides quick data visualizations without the need for coding
  - R allows for customization of a larger array of data visualization tools
  - Tableau allows for basic statistical analyses
  - R allows access to sophisticated statistical methods through libraries/packages
  - Tableau allows for dashboards to be built on top of databases
  - R allows for dashboards to be developed through the Shiny package
2. Read data into RStudio

- Can read in different formats
- RStudio allows for data to be read in through a drop down menu
- Data can also be read in through a script or command line, e.g., through `read.csv()` or `read.table()`
An aside: a few R basics

• Scripts: We will use a script to write out commands that will be executed in R

• Packages: We will use some packages that need to be installed (once) and then loaded upon use in an R session

• Data: We will use data that is loaded and stored in a matrix format. For example, for a data set named \texttt{dat}:
  • \texttt{dat[i, j]} is the element in the \texttt{i}th row and the \texttt{j}th column
  • \texttt{dat[i, ]} is the \texttt{i}th row
  • \texttt{dat[ , j]} is the \texttt{j}th column

• Variables within a data frame can be accessed using the $ symbol, e.g., \texttt{dat$Age}

• Functions in R have the form of \texttt{functionName(options for function)}
We will use the adult data from “Marijuana Arrest Data” available at https://mpdc.dc.gov/node/1347766 for the purposes of illustrating plots.

**Background:**
In the District of Columbia, the laws related to the recreational use and possession of marijuana have changed at two milestones: the effective dates of the *Marijuana Possession Decriminalization Amendment Act of 2014* on July 17, 2014, and of Initiative 71 on February 26, 2015 (https://mpdc.dc.gov/marijuana).

**Data:**
This data includes arrests made by the Metropolitan Police Department (MPD). The data represents individuals arrested with a marijuana charge, regardless of whether there was a more serious secondary charge. If an arrestee was charged with multiple marijuana charges, the arrest is only counted once under the more serious charge type (Distribution > Possession with Intent to Distribute > Possession > Public Consumption).
3. Create a range of simple plots

- Histograms
- Boxplots
- Density plot
- Scatterplot
- 3D Scatterplot – Static
- 3D Scatterplot - Interactive
- 2D Hexagon
3. Create a range of simple plots: Histograms

# histogram with default options

hist(dat$Age)
3. Create a range of simple plots: Histograms

```r
# making a pretty histogram

hist(dat$Age,
     xlab = "Age",
     ylab = "Count",
     main = "Metropolitan Police Department Marijuana Arrests: 2012-2017",
     col = "darkgreen")

# see help on the hist function for more options
# try by typing help(hist) at the command prompt
```
3. Create a range of simple plots: Boxplots

```r
# boxplot with default options
boxplot(dat$Age)
```
3. Create a range of simple plots: Boxplots

# a grouped boxplot with default options

table(dat$Defendant.Sex)
boxplot(dat$Age ~ dat$Defendant.Sex)
3. Create a range of simple plots: Density plots

```r
# density plot with default options
plot(density(dat$Age, na.rm=T))
```
3. Create a range of simple plots: Density plots

# a Boolean vector is created for men and for women

men = (dat$Defendant.Sex == "M")

women = (dat$Defendant.Sex == "F")

plot(density(dat[men,]$Age, na.rm=T), col="red")

lines(density(dat[women,]$Age, na.rm=T), col="blue")
3. Create a range of simple plots: Density plots

```r
# making the density plot fancier
# col changes line color
# lty changes the line type
# lwd changes the line width

plot(density(dat[men,]$Age, na.rm=T), col="red",
     main="Density plots for Age by Sex",
     xlab="Age in Years", lty=1, lwd=2)
lines(density(dat[women,]$Age, na.rm=T),
      col="blue", lty=2, lwd=2)
legend(60,.05, legend=c("Men", "Women"),
       col=c("red","blue"), lty=c(1,2), lwd=2)
```
3. Create a range of simple plots: Scatterplots

```r
# Scatterplots with default options
```
3. Create a range of simple plots: Scatterplots

```r
# Scatterplot with fancier options
plot(dat$Arrest.Location.Block.GeoX, 
     dat$Arrest.Location.Block.GeoY, 
     xlab="Block level GEOX", 
     ylab="Block level GEOY", pch=20)
```
3. Create a range of simple plots: Scatterplots

```
# Scatterplot with more fancy options
# a vector for color is created based on arrest type

matype.col = rep("black", 9918)
matype.col[dat$Marijuana_Arrest.Type == "Distribution"] = "Blue"
matype.col[dat$Marijuana_Arrest.Type == "Possession"] = "Red"
matype.col[dat$Marijuana_Arrest.Type == "Possession with intent to distribute"] = "Green"
     xlab="Block level GEOX", ylab="Block level GEOY", pch=20,
     col=matype.col)

leg.text = c("Distribution", "Possession",
             "Possession with intent to distribute",
             "Public consumption")

legend(401000, 147000, leg.text, col=c("Blue", "Red", "Green", "Black"), pch=20)
```
3. Create a range of simple plots: Scatterplots

```r
# Create a scatterplot
plot(type = "p",
     data = data,
     col = c("Blue", "Red", "Green", "Black"),
     pch = 20,
     xlab = "Block level GEOX",
     ylab = "Block level GEOY")
```
3. Create a range of simple plots: 3D Scatterplot – Static

```r
# Creating a 3D plot
# Do this on the first time:
install.packages("scatterplot3d")

# do this everytime you have an R session
library("scatterplot3d")

scatterplot3d(dat$Arrest.Location.Block.GeoX, 
dat$Arrest.Location.Block.GeoY, dat$Age, 
xlab="Block Level GEOX", 
ylab="Block Level GEOY", zlab="Age in Years", 
color=matype.col)
```
3. Create a range of simple plots: 3D Scatterplot – Interactive

```r
# interactive 3D plots through the rgl library
# do this the first time you use the package:
install.packages("rgl")

library(rgl)
plot3d(dat$Arrest.Location.Block.GeoX,
       dat$Arrest.Location.Block.GeoY, dat$Age,
       xlab="Block Level GEOX",
       ylab="Block Level GEOY", zlab="Age in Years",
       col=mattype.col)

# adding on color to represent a fourth variable

plot3d(dat$Arrest.Location.Block.GeoX,
       dat$Arrest.Location.Block.GeoY, dat$Age,
       xlab="Block Level GEOX",
       ylab="Block Level GEOY", zlab="Age in Years",
       col=mattype.col)

# Extra: the library rmarkdown has a function that will allow
# an interactive html file to be created for the plot

library("rmarkdown")
writeWebGL(dir = "webGL",
           filename = file.path("YourDirectory", "index.html")))
```
3. Create a range of simple plots: 2D Hexagon

```r
# do this the first time: install.packages("ggplot2") and install.packages("hexbin")

# 2D Hexagon

library("ggplot2")


d + geom_hex(bins=15) + labs(x="Block Level GEOX", y="Block Level GEOY")
```
R provides many other plot options which build off the same framework and graphical options we have used. Potential plots of interest:

- Violin Plots
- Heatmaps
- Geospatial and thematic plots
- Hierarchical clustering trees

- Google is your friend here!
- I frequently find Quick-R to be a great resource: https://www.statmethods.net/

- The basics of plotting will take you a long ways, but with a little more time, R’s flexibility allows for the creation of informative and visually appealing plots.
More advanced options are available through:

- **ggplot2**: which is a data visualization package in R, [https://ggplot2.tidyverse.org/](https://ggplot2.tidyverse.org/)
- **Shiny**: which allows interactive web apps to be built using R, [https://shiny.rstudio.com/gallery/](https://shiny.rstudio.com/gallery/)
- **Tidyverse**: which is a collection of packages for data science, [https://www.tidyverse.org/](https://www.tidyverse.org/)
QUESTIONS?

• Now...

• Or later at Stephanie.Santorico@ucdenver.edu

• Have specific plots or graphical options you’d like more tutorial on? Send suggestions/requests to me at https://tinyurl.com/plotsForTutorial.