INTRODUCTION TO R

Carolyn Coberly

October 2017

GOALS

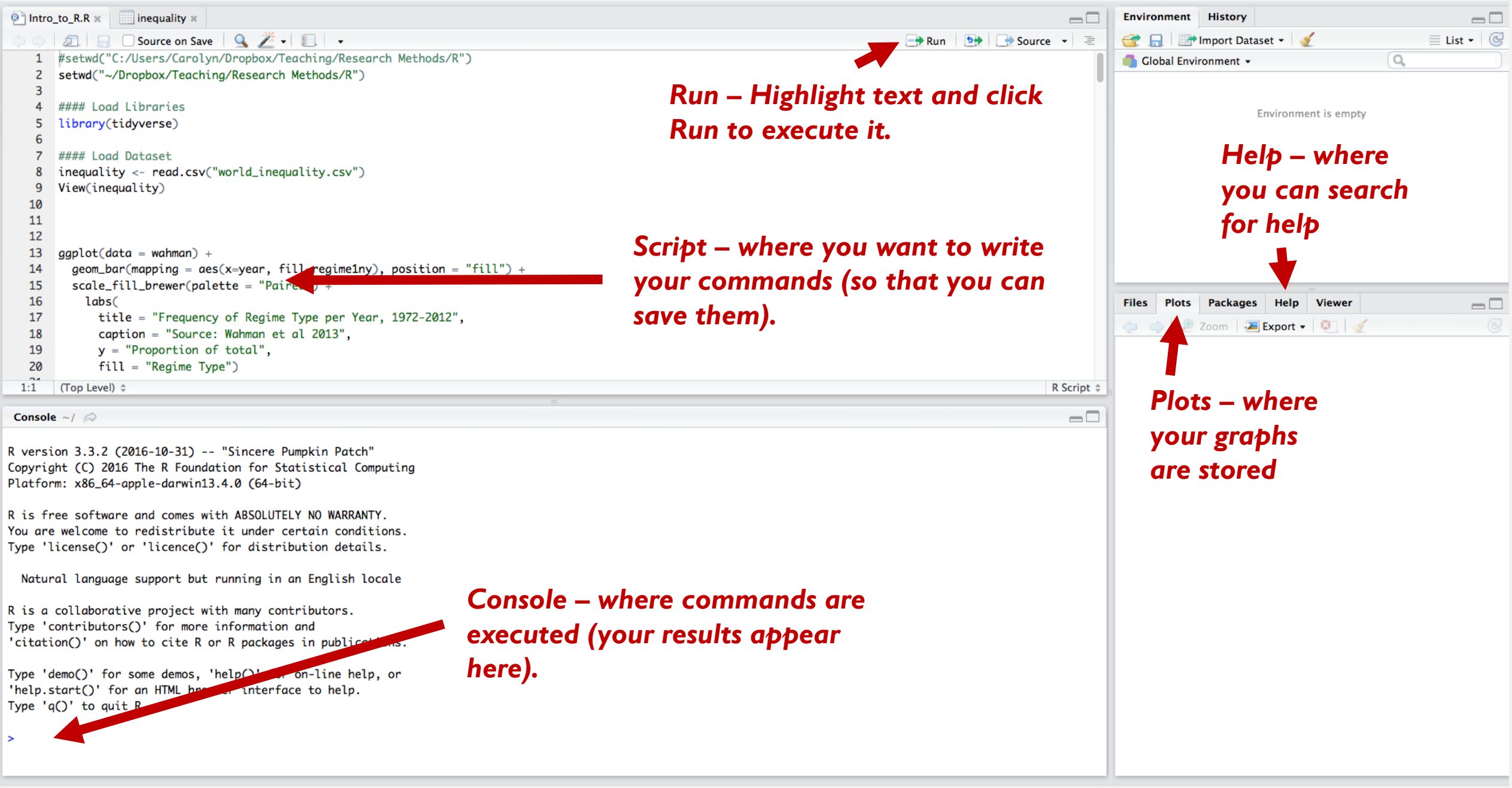
- Install R and learn its basic grammar
  - Libraries and packages
  - Assigning objects
  - Logical commands
- Perform basic statistics in R
  - Open a dataset
  - Produce summary statistics
  - Run a linear regression model
- Learn where to find more information

WHY R?

- Open Source and User-Generated
- Flexible
  - Can write programs to do advanced statistics, simulations, and text analysis
  - Work with more than one dataset at a time
  - Best graphics interface of all statistical programs
- Most common program used in the U.S. for political science

R AND R STUDIO

- R is a statistical software program. To use it, you need to install it on your computer.
- To install R (the actual program), go to:
  - https://cran.r-project.org/, select your operating system, and follow the instructions online.
- To install R Studio (which makes it easier to use R), go to:
  - https://www.rstudio.com/products/rstudio/download/, choose the open source license option, and follow the remaining instructions.
- To start using R, open R Studio.


Intro_to_R.R
inequality
Source on Save
Run
Source
#setwd("C:/Users/Carolyn/Dropbox/Teaching/Research Methods/R")
setwd("~/Dropbox/Teaching/Research Methods/R")
Load Libraries
library(tidyverse)
Load Dataset
inequality <- read.csv("world_inequality.csv")
View(inequality)
ggplot(data = wahman) +
geom_bar(mapping = aes(x=year, fill=regime1ny), position = "fill") +
scale_fill_brewer(palette = "Pair
labs(
title = "Frequency of Regime Type per Year, 1972-2012",
caption = "Source: Wahman et al 2013",
y = "Proportion of total",
fill = "Regime Type")
1:1
(Top Level)
R Script
Run – Highlight text and click Run to execute it.
Script – where you want to write your commands (so that you can save them).
Console ~/
R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin Patch"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Console – where commands are executed (your results appear here).
Environment
History
Import Dataset
List
Global Environment
Environment is empty
Help – where you can search for help
Files
Plots
Packages
Help
Viewer
Zoom
Export
Plots – where your graphs are stored

WRITING CODE

- To use R, you write commands that the program executes.
- The best way to do this is to write the commands as a script.
  - Open a new script (File/New File/New R Script…)
  - Save your script file (File/Save…)
- When you want to execute your commands, highlight them and click "Run."
  - You must highlight *all* the text you want to run.
- To add comments on your code, use the pound sign (#).
  - Any code appearing after the # will not run.
  - You must add a new # on each new line of comments (no multi-line commenting).

PACKAGES AND LIBRARIES

- Packages are like software programs for use within an R operating system.
- To use commands to perform functions, you must first load a library that contains that command.
- The first time you use a library, you must install the package that contains it.
  - *install_packages("tidyverse")*
- Then, and every time you open R, you must call up the libraries you want to use.
  - *library(tidyverse)*

OBJECTS

- R is an object-oriented programming language
- Objects can be anything – a datasheet, column within that datasheet (vector), a graph, etc.
- You need to assign your data and commands to an object in order to be able to use it later.
  - *data <- (commands)*
- You can call things whatever you want, but everything is case-sensitive
- If your object is a datasheet, and you only want to use one variable, you can call it using $:
  - *data$varname*

LOADING A DATASET

- Your data should be in .csv format before you try to use it in R (save as .csv if not already formatted).
- The command is read.csv(); remember to assign an object!
  - *data <- read.csv("data.csv")*
- Macintosh and PC systems register file names slightly differently.
  - *To load a PC file, use "C:/Users/Name/Directory/filename.csv"*
  - *To load on a Mac, use "~/Directory/filename.csv"*
- To take a look at your data use View()
  - *View(data)*

GRAMMAR OF COMMANDS

data <- read.csv("data.csv", header=TRUE, sep=",")

- <- assigns to an object
- General format: command(data, formula, options)
  - Insert the objects and formulae you want to use in the parentheses
  - Options typically come after the comma
  - Text in "", binary commands in CAPITALS
- Varies by package and command
  - *object <- lm(formula, data)*
  - *object <- ggplot(data, variables) + graphtype(options) + formatting(options)*
- The Help file for each command tells you the grammar and options available for each command.

LOGICAL COMMANDS

- If: if
- And: &
- Or: |
- Not: !
- Equal: A double equal sign == when used in a logical command only; otherwise use a single =
  - *if year == 2000 & country == "Armenia"*
  - *X = y*2*
- Greater/Less than: >, <, >=, <=

SUMMARY STATISTICS

- *summary()* – prints the mean, median, highest and lowest values for all variables
- *stat.desc()* – prints the N, min max, range, median, mean, variance, and standard deviation
  - Must load *library(pastecs)* to use

LINEAR REGRESSION

- Linear regression (OLS) provides a point estimate of the slope of the straight line that best fits your data (the coefficient) and a statistic estimating how accurate that coefficient is (the standard error).

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$$

- Command: *reg <- lm(y ~ x, data)*
- To view the coefficient and standard error, you need to use summary():
  - *summary(reg)*
- To add control variables, add them to your independent variable:
  - *mvreg <- lm(y ~ x+z, data)*

MULTIVARIATE REGRESSION - EXAMPLE

- *mvreg <- lm(wb_gdppc ~ cgv_dem+colbrit+colfra+wb_gini, inequality)*
- Going from dictatorship to democracy is associated with a $5651 increase in GDP per capita, holding colonial status and inequality constant. This result is statistically significant at the 0.05 level.

Coefficient

p value

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    5903.5     5013.0   1.178  0.24689
cgv_dem        5651.3     2399.3   2.355  0.02424 *
colbrit        9737.1     2934.1   3.319  0.00212 **
colfra         1273.8     3844.6   0.331  0.74238
wb_gini        -150.7      107.5  -1.401  0.16997
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

HELP!

- help()
- ?command
- Type command name into Help box
- R cookbook (www.cookbook-r.com)
- *R for Data Science* by Hadley Wickham & Garrett Grolemund (http://r4ds.had.co.nz/ and for purchase on Amazon)

CROSS-TABULATION

- A table comparing the frequency the values of two variables appear together is called a cross-tabulation.
- Can only be done using two categorical variables.
- crosstab <- table(data$y, data$x)
  - y will be rows, x columns
- CrossTable(data$y, data$x, expected=FALSE, prop.r=FALSE, prop.t= FALSE, prop.chisq=FALSE)
  - Requires library(gmodels)
  - This command also shows column percentages

DIFFERENCE OF MEANS

- The most common way to evaluate an experiment is to compare the mean response to the treatment to the mean response to the control – take the difference between the two means.
- You should report the actual difference and the statistic representing the statistical significance of the difference (a t-test).
- Command: t.test(y ~ x, dataset)
- Interpretation: If the p value of the t-test is less than 0.05, then the difference between the two means is statistically significant (aka, you can reject the null hypothesis that you have no difference in the means).

DIFFERENCE OF MEANS - EXAMPLE

- What is the effect of being a former British colony on the likelihood a country is democratic?
- t.test(cgv_dem ~ colbrit, inequality)

```
        Welch Two Sample t-test

data:  cgv_dem by colbrit
t = 0.91288, df = 136.18, p-value = 0.3629
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval
 -0.08041204  0.21830982
sample estimates:
mean in group 0 mean in group 1
      0.5983607       0.5294118
```

- 59.8% of non-former British colonies are democracies; 52.9% of former British colonies are democracies (these are the means in this example).
- The difference of means is 0.07 (or 6.9%). This result is not statistically significant (the p-value is greater than 0.05; the 95% confidence interval includes zero).