Programming for Professional Research Using R

Session 1

March 30, 2023

Introduction

Purpose | Learning R

Purpose of this course: Put you on the right track to use R for professional research.

- **The vast majority** of research assistant/analyst (RA) work consists of cleaning and constructing datasets for analysis
- **Entry-level RA positions rarely require complex econometric/regression skills**

Introduction

Purpose | Learning R

You should think of learning R like learning a language.

- Taking a six-hour course won't make you proficient in it
- If you don't practice it, you'll forget it
- Solution -- **Find ways to use R in your life, either personally or professionally**

Why R?

Jobs | Beautiful Tables | Beautiful Graphs | Beautiful Maps

- Many entry-level research jobs in policy, economic, development, or political science institutions now expect quantitative work using Stata, R, or Python
- Coding skills make you more valuable in any position -- data adds value to nearly every kind of research!

Why R?

Jobs | **Beautiful Tables** | Beautiful Graphs | Beautiful Maps

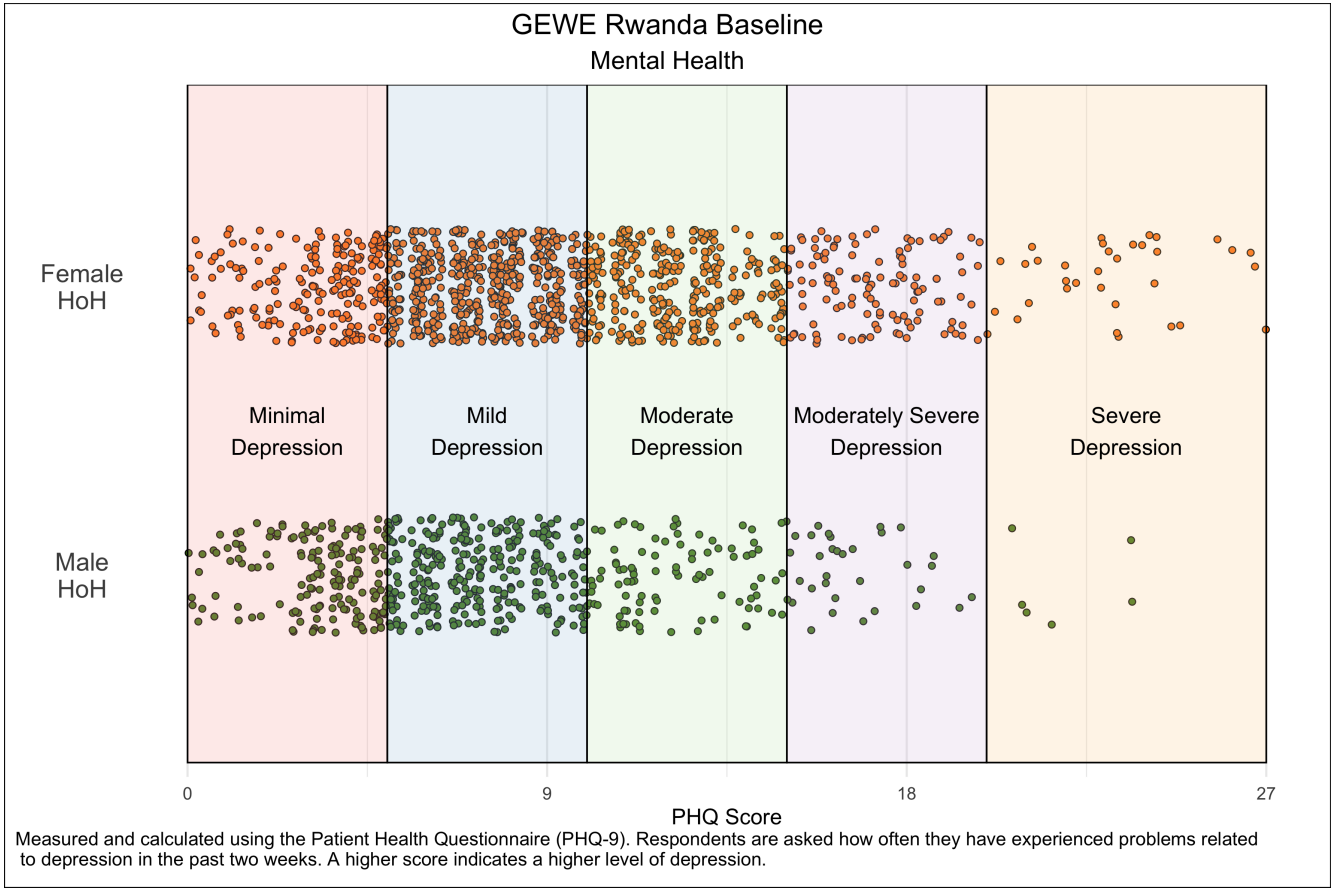
2021 Expected vs. Actual Fantasy Points

Top 40 Running Backs

Player	Team	Expected FP	Expected FP Rank	Actual FP	Actual FP Rank
Jonathan Taylor	IND	301.20	RB 1	324.20	RB 1
Najee Harris	PIT	292.67	RB 2	228.10	RB 4
Joe Mixon	CIN	251.89	RB 3	254.80	RB 3
Leonard Fournette	TB	232.12	RB 4	221.10	RB 5
Ezekiel Elliott	DAL	225.03	RB 5	214.76	RB 6
Austin Ekeler	LAC	217.88	RB 6	263.90	RB 2
Antonio Gibson	WAS	216.49	RB 7	186.70	RB 11

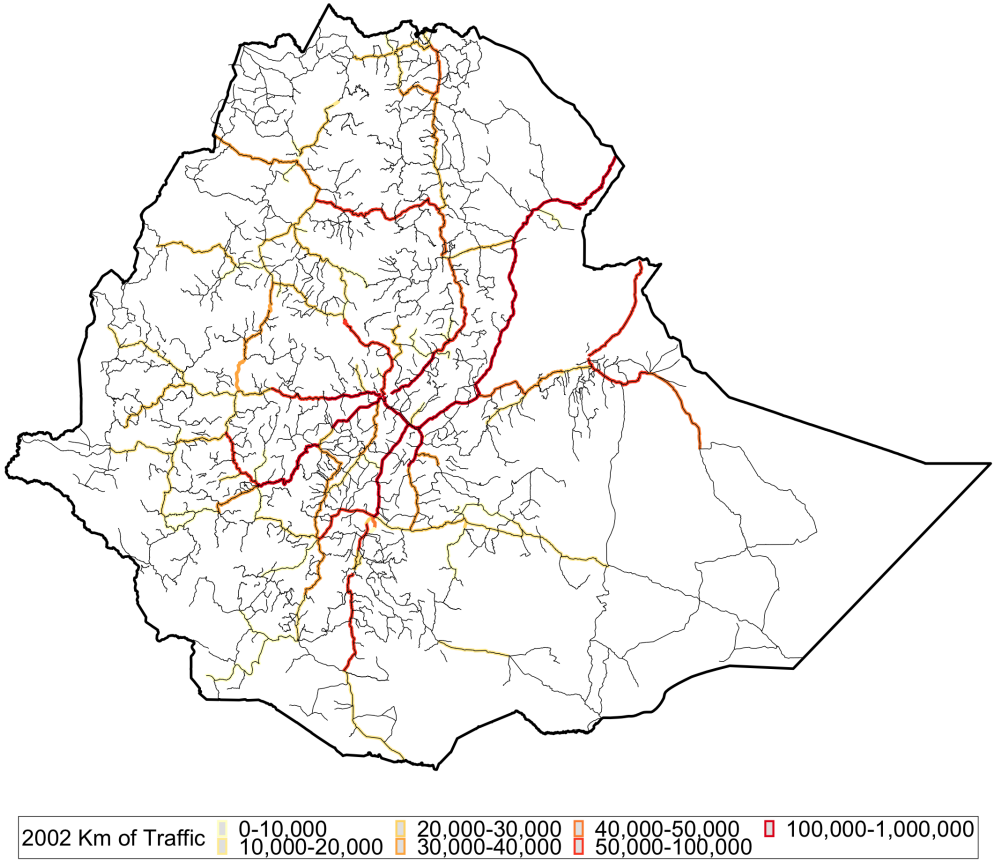
Why R?

Jobs | Beautiful Tables | **Beautiful Graphs** | Beautiful Maps



Why R?

Jobs | Beautiful Tables | Beautiful Graphs | **Beautiful Maps**



Today

- Learn how to:
  - Import data in an efficient and reproducible manner
  - Filter, mutate, group, and summarize data using Tidyverse functions
  - Reshape data using Tidyverse functions
- Be introduced to:
  - Code and data collaboration -- GitHub and data project reproducibility
- Practice the above!

R Basics

RStudio Layout | Creating Vectors | Selecting Vector Elements

- You write your code in a **script**
- When you run the code, it runs in the **console**
- The objects that you use/create (datasets, vectors, values) appear in your **environment**

R Basics

RStudio Layout | **Creating Vectors** | Selecting Vector Elements

```
vector <- c(1, 2, 5)
vector
```

```
## [1] 1 2 5
```

```
vector <- 2:6
vector
```

```
## [1] 2 3 4 5 6
```

```
vector <- seq(2, 3, by = 0.5)
vector
```

```
## [1] 2.0 2.5 3.0
```

R Basics

RStudio Layout | Creating Vectors | **Selecting Vector Elements**

```r
## By Position
x[4]        # Fourth element
x[-4]       # Everything but the fourth element
x[2:4]      # Elements two to four
x[-(2:4)]   # Everything but elements two to four
x[c(1, 5)] # Elements one and five
```

```r
# By Value
x[x == 10]            # Elements which are equal to 10
x[x < 0]              # Elements that are less than zero
x[x %in% c(1, 2, 5)] # Elements in the set 1, 2, 5
```

Coding Set Up

Easy

```
install.packages("tidyverse")

library(tidyverse)
```

Better

The `pacman` package installs packages if they aren't installed yet, loads them otherwise

```
if(!require(pacman)) install.packages("pacman")

pacman::p_load(tidyverse)
```

You want your code to be **reproducible** and **easy to use by other people**

Solution:

```
# Set User (this allows us to use fixed file paths but to adapt them
# for multiple possible users)

      # 1 -- Marc-Andrea Fiorina

      # 2 -- Enter here if needed

user <- 1

if(user == 1) {

    # Absolute file path

    main_filepath <- "/Users/marc-andreafiorina/Dropbox/SAIS R Course/"

}

# Notice the relative file paths

data_filepath <- paste0(main_filepath, "data/")
```

Importing Data

Easiest file type to import into R is a .csv file. But you can also import .xlsx, .dta (Stata), etc.

- Easy -> `read.csv()`
- Harder (faster) -> `data.table::fread()`

```
norms_values_data <- data.table::fread(
    paste0(data_filepath, "session_1/wvs_values_norms_data.csv"),
    na.strings = ""

)
```

Data 'Wrangling'

Tidyverse Introduction

Base R Layout | Tidyverse Layout

```
names(mtcars)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

```
str_replace(str_to_upper(names(mtcars)), "MPG", "HELLO")
```

```
##  [1] "HELLO" "CYL"   "DISP"  "HP"    "DRAT"  "WT"    "QSEC"  "VS"    "AM"
## [10] "GEAR"  "CARB"
```

Tidyverse Introduction

Base R Layout | **Tidyverse Layout**

Tidyverse functions introduce a 'cleaner' method to write code out, using what is called the 'pipe operator': **%>%**. It's almost like writing a recipe, step by step.

```
names(mtcars)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

```
mtcars %>%
    names() %>%
    str_to_upper() %>%
    str_replace("MPG", "HELLO")
```

```
##  [1] "HELLO" "CYL"   "DISP"  "HP"    "DRAT"  "WT"    "QSEC"  "VS"    "AM"
## [10] "GEAR"  "CARB"
```

mutate() filter() select()

```r
mutate_example <- mtcars %>%
    mutate(
        heavy = case_when(
            wt > 3 ~ "Yes",
            TRUE   ~ "No"
        )
    ) %>%
    select(wt, heavy)

mutate_example %>% head()
```

```
##                      wt heavy
## Mazda RX4         2.620    No
## Mazda RX4 Wag     2.875    No
## Datsun 710        2.320    No
## Hornet 4 Drive    3.215   Yes
## Hornet Sportabout 3.440   Yes
## Valiant           3.460   Yes
```

mutate() **filter()** select()

```r
filter_example <- mtcars %>%
    filter(wt > 3)

filter_example %>% head()
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
```

mutate() filter() **select()**

```r
select_example <- mtcars %>%
    select(
        matches("mpg|carb")
    )
select_example %>% head()
```

```
##                    mpg carb
## Mazda RX4         21.0    4
## Mazda RX4 Wag     21.0    4
## Datsun 710        22.8    1
## Hornet 4 Drive    21.4    1
## Hornet Sportabout 18.7    2
## Valiant           18.1    1
```

group_by() and summarize() pivot_longer() pivot_wider()

```r
group_by_summarize_example <- mtcars %>%
    group_by(cyl) %>%
    summarize(
        mpg = mean(mpg, na.rm = TRUE)
    )

group_by_summarize_example
```

```
## # A tibble: 3 × 2
##     cyl   mpg
##   <dbl> <dbl>
## 1     4  26.7
## 2     6  19.7
## 3     8  15.1
```

```r
relig_income[1:6] %>% head(n = 2)
```

```
## # A tibble: 2 × 6
##   religion `<$10k` `$10-20k` `$20-30k` `$30-40k` `$40-50k`
##   <chr>      <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
## 1 Agnostic      27        34        60        81        76
## 2 Atheist       12        27        37        52        35
```

```r
relig_income_long <- relig_income %>%
    pivot_longer(
        cols      = !religion, # Everything but religion
        names_to  = "levels",
        values_to = "num"
    )

relig_income_long %>% head(n = 4)
```

```
## # A tibble: 4 × 3
##   religion levels    num
##   <chr>    <chr>   <dbl>
## 1 Agnostic <$10k      27
## 2 Agnostic $10-20k    34
## 3 Agnostic $20-30k    60
## 4 Agnostic $30-40k    81
```

```r
fish_encounters %>% head(n = 4)
```

```
## # A tibble: 4 × 3
##   fish  station  seen
##   <fct> <fct>   <int>
## 1 4842  Release     1
## 2 4842  I80_1       1
## 3 4842  Lisbon      1
## 4 4842  Rstr        1
```

```r
fish_encounters_wide <- fish_encounters %>%
    pivot_wider(
        names_from  = station,
        values_from = seen
    )

fish_encounters_wide[1:6] %>% head(n = 2)
```

```
## # A tibble: 2 × 6
##   fish  Release I80_1 Lisbon  Rstr Base_TD
##   <fct>   <int> <int>  <int> <int>   <int>
## 1 4842        1     1      1     1       1
## 2 4843        1     1      1     1       1
```

Code and Data Collaboration

Important Points

Professional settings are collaborative settings

Most R courses teach you to code in isolation. But **professional use of R often happens within teams of researchers.**

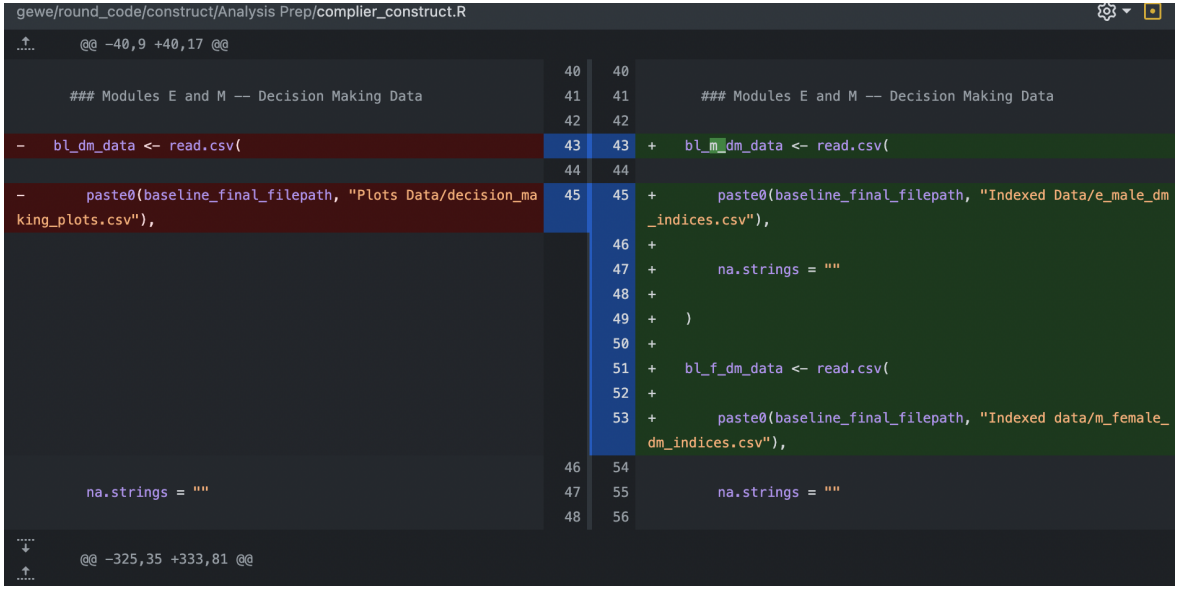
Reproducibility and shareability of your code are key to ensure:

- Others can use and understand your work
- Accountability

The ultimate collaborative tool: GitHub

GitHub is the preferred code storing platform for data teams because of the following attributes:

- **Easy sharing** of scripts between team members
- **Tracked changes** -- every changed line of code is attributed to a specific person
- **'Commit' structure** -- allows you to see how your changes affect other people's work on the same scripts



Practical Exercise -- Using the World Values Survey Dataset

World Values Survey

Background

"The survey, which started in 1981, seeks to use the most rigorous, high-quality research designs in each country. The WVS consists of nationally representative surveys conducted in almost 100 countries which contain almost 90 percent of the world's population, using a common questionnaire. [...] WVS seeks to help scientists and policy makers understand changes in the beliefs, values and motivations of people throughout the world."

Survey Contents

- Social values, attitudes & stereotypes
- Societal well-being
- Social capital, trust and organizational membership
- Economic values
- Corruption
- Migration
- Post-materialist index
- Science & technology
- Religious values
- Security
- Ethical values & norms
- Political interest and political participation
- Political culture and political regimes
- Demography

Today's practical component

1. Successfully run the code in the `session_1_template.R` script
2. Create your own script and do one or more of the following:
    - Find mean values for 'importance in life' variables (Q1-6) for countries in another region than Europe
    - Calculate average 'enthusiasm' for these life subjects in countries in another region than Europe
    - Perform the same analysis, either on European countries or other countries, for another group of indicators in the dataset:
        - Important child qualities: Q7-18
        - Neighbors: Q19-26
        - Statements to agree with: Q27-41

NOTE You should refer to documentation for the dataset, which can be found in `Dropbox/SAIS R Course/documentation/`, for details on the variables and their given values.

Links

Syllabus:
https://mfiorina.github.io/sais_r_course/spring_2023/syllabus/r_course_syllabus.html

Thomas Mock, “A Gentle Introduction to Tidy Statistics in R” (blog post and video)

Dominic Royé, “A very short introduction to Tidyverse”

tidyr, “Pivoting”

Hadley Wickham, “dplyr 1.0.0: working across columns”

Hadley Wickham & Garrett Grolemund, R for Data Science

RStudio, RStudio Cheatsheets