ECON-320-Lab-1-2

Sonan Memon

Introduction

- I am a second year PhD Econ student; interested in international finance and monetary policy.
- I thank Micaela Wood for some material in these slides.
- My office hours are on Monday, 3 to 5 pm in PLC 514; email: smemon@uoregon.edu and course github link is: github.

Installation of R, R Studio and Quarto.

- In this first lab, we will talk about basics regarding installing R, R Studio and working with Quarto.
- Download R, RStudio and Quarto from these sites.
- RStudio is the IDE; Quarto is for rendering presentations and papers. R markdown is a substitute for Quarto.

Common R Packages

Below, I have some common libraries loaded in R.

```
library(tinytex)

library(tidyverse)

library(dslabs)

library(dplyr)

library(ggplot2)


library(tibble)

#install.packages("devtools")
#devtools::install_github("username/repo")
```

Basics of R/R Studio

- Why R? Many economists think it dominates STATA and R is open-source.
- Large community for debugging and discussions: stackoverflow etc.
- After Python, it is the second most common language used in industry for data science.

Basics of R/R Studio

- Console: type code for trial without saving, bugs will display here.
- Render: Used for executing quarto; knitting is used for R Markdown.
- Environment: Shows data, variables and functions.
- Help: searching functions and syntax.
- .r (script file), .qmd or .rmd files are file types.

Basic Vector Operations

- The data below is fictitious.

```
runs <- c(12, 65, 8, 55, 27, 15, 4, 5, 22, 30, 17,
          10, 40, 7, 29, 38, 5,
          23, 18, 20, 9, 25, 22, 0, 0)


summary(runs)

average <- mean(runs)
which.max(runs)

# sorting:
sort(runs, decreasing = TRUE)

# Position of Sorted Numbers:
order(runs, decreasing = TRUE)
```

Basic Vector Operations

- Computing statistics:

```
# element vise operation:

runs[c(1:20)]/20

average_first_20 <- mean(runs[c(1:20)])
average_first_5 <- mean(runs[c(1:5)])

# Percentiles:

quantile(runs, c(0.1, 0.5, 0.9, 0.99))

cat("Batting Average In First 20 Matches:",
    average_first_20, "\n")
cat("Batting Average In First 5 Matches:",
    average_first_5, "\n")
```

Basic Data Wrangling

In the code below, I import data, create new variables and export data.

```
#setwd("C:/users/sonan/Documents/ECON-320-Fall-2025-GE")



data <- read.csv("US_GDP_NX.csv")


head(data)

tail(data)


data <- ts(data, start = c(1947,1), frequency = 3)
```

Basic Data Wrangling

```r
data <- data.frame(data)



data$Quarter <- seq(from = as.Date("1947-01-01"),
                    to = as.Date("2020-12-01"),
                    by = 'quarter')

colnames(data) <- c("GDP", "NX", "index", "Quarter")

# Create new variable with natural log
data$log_gdp <- log(data$GDP)

#write.csv(data, "output.csv", row.names = FALSE)
```

Equations

- Quarto allows for basic LaTeX syntax to produce equations.

$$\begin{aligned}\text{Strike Rate} &= \frac{\text{Total Runs}}{\text{Total Balls}} \times 100 \\ &= \frac{590}{1154} \times 100 \approx 51.1 \\ \text{Mean} &= \frac{1}{25} \sum_{i=1}^{25} R_i\end{aligned}$$

Binding 1

```
country <- c("USA", "India", "Argentina", "Sudan")

WB_classification <- c("High Income",
                       "Lower Middle Income",
"Upper Middle Income", "Low Income")

rbind(country, WB_classification)

cbind(country, WB_classification)

cc <- cbind(country, WB_classification)

chdi <- cbind(country, hdi_rank = c(17, 130, 47, 176))

cbind(cc, chdi)

rbind(cc, chdi) #doesn't capture column names of latter.
```

Binding 2

```
country_class1 <- tribble(
  ~Country,      ~Income_Classification,
  "USA",         "High income",
  "India",       "Lower-middle income",
  "Argentina",   "Upper-middle income",
  "Sudan",       "Low income"
)

country_class2 <- tribble(
  ~Country,   ~Income_Classification,
  "USA",      "High income",
  "China",    "Upper-middle income",
  "Germany",  "High income"
)
```

Binding 2

```
cc_tib <- as_tibble(cc)

chdi_tib <- as_tibble(chdi)

bind_cols(cc_tib, chdi_tib)

bind_rows(cc_tib, chdi_tib)

union(country_class1, country_class2)
# similar to bind rows but unions will remove duplicates.

intersect(country_class1, country_class2)

setdiff(country_class1, country_class2)
# unique of first
```

Mutating Joins

```
country_class3 <- tribble(
  ~Country,      ~HDI,
  "USA",         17,
  "India",       130,
  "Argentina",   47,
  "Ethiopia",     180
)


left_join(country_class1, country_class3, by = "Country")

left_join(country_class3, country_class1, by = "Country")

inner_join(country_class1, country_class3, by = "Country")

full_join(country_class1, country_class3, by = "Country")
```

Filtering Joins

```
country_class3 <- tribble(
  ~Country,      ~HDI,
  "USA",         17,
  "India",       130,
  "Argentina",   47,
  "Ethiopia",     180
)


semi_join(country_class1, country_class3)
#keeps all rows of first where key matches with second.

anti_join(country_class1, country_class3)
#keeps rows of first which don't match with the second.
```

Outliers

```
data <- read.csv("output.csv")


stats <- summary(data$log_gdp)
stats

low <- 1.25
high <- 5.5

filtered_gdp <- filter(data, log_gdp > low,
                       log_gdp < high)
```