Class 01: Introduction to R

Introduction

This class is designed to provide an introductory overview of working with data in R. We will cover the essentials of data manipulation, exploration, and regression analysis. By the end of this lecture, you should be comfortable setting up your R environment, loading data, exploring and modifying datasets, running regression models, and exporting your results.

Outline

- Setting up data
- Exploring data
- Editing data
- Estimating regressions
- Exporting regression output

1 Setting Up Data

Objective: Prepare the R environment by installing necessary packages, setting up the working directory, and loading the dataset.

1.1 Clearing the Workspace

It is important to start with a clean workspace to avoid conflicts with existing objects in your R session. This can be done using:

```
rm(list = ls())
```

This command clears all objects from the environment, ensuring that you start fresh.

1.2 Installing and Loading Packages

R packages extend the functionality of R. Here's how you would update and install packages, though in practice, these commands are often commented out to avoid reinstalling each time:

```
# update.packages()
# install.packages("tidyverse")
# install.packages("magrittr")
# install.packages("stargazer")
```

After installing the necessary packages, load them into your session:

```
library(tidyverse)
library(magrittr)
library(stargazer)
```

These packages are essential for data manipulation (tidyverse), pipe operations (magrittr), and generating summary tables (stargazer).

1.3 Setting the Working Directory

Setting the correct working directory is crucial as it tells R where to look for files and where to save output:

```
directory <- "/Users/taky/econometrics_and_r/class1_introduction_to_r/"
getwd()
# setwd("/Users/taky/econometrics_and_r/class1_introduction_to_r")
```

Ensure that the directory path is correctly set to where your data files are stored.

1.4 Loading the Dataset

Load the dataset into R using the read.csv() function:

```
data <- read.csv(paste0(directory, "wage1.csv"))
```

This command reads the data from a CSV file and stores it in the variable data for further analysis.

2 Exploring Data

Objective: Gain an understanding of the dataset's structure and content.

2.1 Understanding Data Structure

Use the str() function to inspect the structure of your data:

```
str(data)
data %>% select(wage, educ, exper) %>% str
```

This gives an overview of the data types and the dimensions of the dataset.

2.2 Previewing the Data

To get a quick look at the first few rows of your data:

```
data %>% select(wage, educ, exper) %>% head(10)
```

This helps you to see the values in your dataset and verify that the data was loaded correctly.

2.3 Generating Summary Statistics

Summarize the data using the stargazer package to produce clear, readable output:

```
data %>% stargazer(type = "text")
data %>% select(wage, educ, exper) %>% stargazer(type = "text")
```

These commands generate summary statistics for the entire dataset and for selected columns, respectively.

2.4 Exploring Data by Group

Understanding your data by groups (e.g., by gender) can reveal important patterns:

```
data %>% select(female) %>% table
data %>% filter(female == 1) %>% stargazer(type = "text")
split(data, data$female) %>% walk(~ stargazer(., type = "text"))
```

Here, we first tabulate the female variable, then filter the data to look at only female observations, and finally split the data by gender to compare statistics.

3 Editing Data

Objective: Modify the dataset by selecting, filtering, and creating new variables.

3.1 Selecting and Filtering Data

You can select specific columns and filter rows based on conditions:

```
data %<>%  select(wage, educ, exper, tenure, female, south, west)
data %<>% select(-tenure)
data %<>% filter(wage < 2)
```

The %<>% operator allows you to overwrite the existing data object with the modified version.

3.2 Creating New Variables

Creating new variables can add valuable information to your dataset:

```
data %<>% mutate(logwage = log(wage),
                 educsq = educ^2,
                 southwest = south + west)
summary(data)
```

In this example, we create logwage (the logarithm of wage), educsq (the square of education), and southwest (a combination of the south and west indicators).

4 Estimating Regressions

Objective: Perform regression analysis to explore relationships between variables.

4.1 Simple Regression Model

To explore the relationship between education and wages, we start with a simple linear regression:

```
linear.model.1 <- lm(wage ~ educ, data)
summary(linear.model.1)
```

This model predicts wage based on educ.

4.2 Multiple Regression Model

Extend the model by including more predictors to understand their combined effect on wages:

```
linear.model.2 <- lm(wage ~ educ + exper, data)
summary(linear.model.2)
```

Here, both educ and exper (experience) are used to predict wage.

5 Exporting Regression Output

Objective: Save the results of your regression analyses for reporting or further use.

5.1 Exporting to HTML

To save your regression output as an HTML file:

```
stargazer(linear.model.1, type = "html", out = "regtable.html")
```

5.2 Exporting to Word Document

To save multiple regression outputs to a Word document:

```
stargazer(linear.model.1, linear.model.2,
          type = "html",
          out = "regtable.doc")
```

This command exports the results of both regression models into a Word document, making it easy to include in reports.

6 Conclusion

In this session, we covered the essential steps for working with data in R: setting up your environment, exploring and editing data, estimating regression models, and exporting results. These skills form the foundation of econometric analysis in R, allowing you to handle real-world data effectively.