Cookbook Data Analysis with Stata and R

Manuel Oliveira

August, 2024

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

```
cat(" This is a test again")
```

```
 This is a test again
```

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

2 Chapter 1: Getting Started

2.1 Introduction

This chapter provides a quick tutorial on how to install and set up R and Stata on both Windows and Mac computers. By the end of this chapter, you'll have the necessary tools ready to begin your analysis.

2.2 Installing R

2.2.1 Windows

1. **Download R**:

    - Go to the R Project website.
    - Click on "Download R for Windows."
    - Click on "base" to download the base R package.

2. **Install R**:

    - Run the downloaded `.exe` file.
    - Follow the installation instructions, accepting the default settings.

3. **Install RStudio** (Optional but recommended):

    - Download RStudio from the RStudio website.
    - Run the installer and follow the setup instructions.

2.2.2 Mac

1. **Download R**:

    - Visit the R Project website.
    - Click on "Download R for macOS."

2. **Install R**:

    - Open the downloaded `.pkg` file.

- Follow the installation instructions.

3. **Install RStudio** (Optional but recommended):

   - Download RStudio from the RStudio website.
   - Open the `.dmg` file and drag RStudio to your Applications folder.

2.3 Installing Stata

2.3.1 Windows

1. **Obtain a License**:

   - Stata is commercial software. Ensure you have a valid license.

2. **Download Stata**:

   - Go to the Stata website and log in to your account to download the installer.

3. **Install Stata**:

   - Run the downloaded `.exe` file.
   - Follow the installation instructions, entering your license information when prompted.

2.3.2 Mac

1. **Obtain a License**:

   - Make sure you have a valid license for Stata.

2. **Download Stata**:

   - Visit the Stata website and log in to your account to download the installer.

3. **Install Stata**:

   - Open the downloaded `.dmg` file.
   - Drag the Stata application to your Applications folder.
   - Launch Stata and enter your license information.

2.4 Setting Up Your Environment

2.4.1 R Setup

1. **Open RStudio** (or R GUI if not using RStudio).
2. **Install Essential Packages**:

   - Open the Console and run:

```
install.packages(c("tidyverse", "lme4", "ggplot2"))
```

3. **Create a New Project** (Optional but recommended in RStudio):

   - Go to "File" > "New Project" > "New Directory" > "New Project."
   - Choose a location and name for your project, then click "Create Project."

2.4.2 Stata Setup

1. **Open Stata**.
2. **Set a Working Directory**:

   - Use the command:

```
cd "path/to/your/directory"
```

Replace `"path/to/your/directory"` with the path where you want to save your files.

3. **Creating Do-Files**:

   - Go to "File" > "New Do-file Editor."
   - Save the Do-file in your working directory.

2.5 Verification

2.5.1 R

1. **Test Installation**:

   - In RStudio or R GUI, type:

```
print("R is working!")
```

- If you see the output `[1] "R is working!"`, your installation is successful.

2. **Load a Package**:

    - Run:

```
library(ggplot2)
print("ggplot2 is loaded!")
```

2.5.2 Stata

1. **Test Installation**:

    - In the Command window, type:

```
display "Stata is working!"
```

- If you see the output `Stata is working!`, your installation is successful.

2. **Check Version**:

    - Type:

```
about
```

- This will display the version of Stata installed.

With your environment set up, you're now ready to start performing analyses using R and Stata!

3 ANOVA

3.1 Introduction

This chapter covers ANOVA (Analysis of Variance), used to compare the means across multiple groups. We will use an example dataset to investigate whether the design of a user interface (UI) affects the time users spend on a website.

3.2 Example Question

Does the design of a user interface (UI) influence the time users spend on a website?

3.3 Dataset Simulation in R

```
# Load necessary package
set.seed(123)

# Simulate data
n_groups <- 3  # Number of UI designs
n_per_group <- 50  # Number of users per group

ui_design <- factor(rep(1:n_groups, each = n_per_group))
time_spent <- rnorm(n_groups * n_per_group, mean = rep(c(20, 25, 22), each = n_per_group),

# Create a data frame
data <- data.frame(ui_design, time_spent)

# View the first few rows
head(data)
```

3.4 Dataset Simulation in Stata

```
* Set seed for reproducibility
set seed 123

* Simulate data
set obs 150
gen ui_design = ceil(_n/50)
gen time_spent = rnormal(20 + (ui_design==2)*5 + (ui_design==3)*2, 5)

* Convert ui_design to a factor
egen ui_design_factor = group(ui_design)

* View the first few rows
list in 1/10
```

3.5 Performing ANOVA

3.5.1 R

```
# Fit the ANOVA model
model <- aov(time_spent ~ ui_design, data = data)

# View the summary
summary(model)

# Post-hoc test (Tukey's HSD)
TukeyHSD(model)
```

3.5.2 Stata

```
* Fit the ANOVA model
anova time_spent ui_design

* Post-hoc test (Bonferroni)
oneway time_spent ui_design, bonferroni
```

3.6 Assumptions

- **Independence**: Observations should be independent of each other.
- **Normality**: The residuals of the model should be normally distributed.
- **Homoscedasticity**: Variances across the groups should be equal.
- **Random Sampling**: The data should be randomly sampled from the population.

4 Linear Regression

4.1 Introduction

This chapter covers how to perform linear regression to study the relationship between variables. We'll use an example dataset that simulates the relationship between study time and performance on an online learning platform.

4.2 Example Question

How does the amount of time spent on an e-learning platform (in hours) affect the test scores of users?

4.3 Dataset Simulation in R

```r
# Load necessary package
set.seed(123)

# Simulate data
n <- 100
study_time <- rnorm(n, mean = 10, sd = 2)  # Average 10 hours
test_score <- 50 + 5 * study_time + rnorm(n, mean = 0, sd = 5)  # Linear relationship with

# Create a data frame
data <- data.frame(study_time, test_score)

# View the first few rows
head(data)
```

4.4 Dataset Simulation in Stata

```
* Set seed for reproducibility
set seed 123

* Simulate data
set obs 100
gen study_time = rnormal(10, 2)
gen test_score = 50 + 5 * study_time + rnormal(0, 5)

* View the first few rows
list in 1/10
```

4.5 Performing Linear Regression

4.5.1 R

```
# Fit the linear regression model
model <- lm(test_score ~ study_time, data = data)

# View the summary
summary(model)
```

4.5.2 Stata

```
* Fit the linear regression model
regress test_score study_time
```

4.6 Assumptions

- **Linearity**: The relationship between the independent and dependent variable should be linear.
- **Independence**: Observations should be independent of each other.
- **Homoscedasticity**: The residuals should have constant variance at every level of the independent variable.
- **Normality**: The residuals should be normally distributed.

5 Multilevel Regression

5.1 Introduction

This chapter covers multilevel regression, where data is nested. We will explore how user satisfaction with a mobile app is affected by time spent on the app, considering that users are nested within different age groups.

5.2 Example Question

Does time spent on a mobile app influence user satisfaction, and does this effect differ across age groups?

5.3 Dataset Simulation in R

```r
# Load necessary package
set.seed(123)

# Simulate data
n_groups <- 5  # Number of age groups
n_per_group <- 50  # Number of users per group

age_group <- factor(rep(1:n_groups, each = n_per_group))
time_spent <- rnorm(n_groups * n_per_group, mean = 30, sd = 10)
satisfaction <- 3 + 0.2 * time_spent + as.numeric(age_group) + rnorm(n_groups * n_per_grou

# Create a data frame
data <- data.frame(age_group, time_spent, satisfaction)

# View the first few rows
head(data)
```

5.4 Dataset Simulation in Stata

```
* Set seed for reproducibility
set seed 123

* Simulate data
set obs 250
gen group = ceil(_n/50)  // Age group
gen time_spent = rnormal(30, 10)
gen satisfaction = 3 + 0.2 * time_spent + group + rnormal(0, 2)

* Convert group to a factor
egen group_factor = group(group)

* View the first few rows
list in 1/10
```

5.5 Performing Multilevel Regression

5.5.1 R

```
# Load necessary package
library(lme4)

# Fit the multilevel model
model <- lmer(satisfaction ~ time_spent + (1 | age_group), data = data)

# View the summary
summary(model)
```

5.5.2 Stata

```
* Fit the multilevel model
mixed satisfaction time_spent || group:
```

5.6 Assumptions

- **Normality of residuals**: The residuals at each level of the model should be normally distributed.
- **Linearity**: The relationship between predictors and the outcome should be linear at each level of the model.
- **Independence**: Observations within each group should be independent.
- **Homoscedasticity**: The variance of residuals should be consistent across all levels of the hierarchy.

6 Logistic Regression

6.1 Introduction

This chapter covers logistic regression, which is used when the outcome variable is binary. The example dataset will examine whether the frequency of technical support contact predicts whether a user continues to use a software product.

6.2 Example Question

Does the frequency of contacting technical support predict whether a user will continue using a software product?

6.3 Dataset Simulation in R

```r
# Load necessary package
set.seed(123)

# Simulate data
n <- 200
support_contact <- rpois(n, lambda = 2)  # Number of contacts with support
continued_use <- rbinom(n, size = 1, prob = 1 / (1 + exp(-(-1 + 0.5 * support_contact))))

# Create a data frame
data <- data.frame(support_contact, continued_use)

# View the first few rows
head(data)
```

6.4 Dataset Simulation in Stata

```
* Set seed for reproducibility
set seed 123

* Simulate data
set obs 200
gen support_contact = rpoisson(2)
gen continued_use = rbinomial(1, 1 / (1 + exp(-(-1 + 0.5 * support_contact))))

* View the first few rows
list in 1/10
```

6.5 Performing Logistic Regression

6.5.1 R

```
# Fit the logistic regression model
model <- glm(continued_use ~ support_contact, data = data, family = "binomial")

# View the summary
summary(model)
```

6.5.2 Stata

```
* Fit the logistic regression model
logit continued_use support_contact
```

6.6 Assumptions

- **Binary Outcome**: The dependent variable should be binary.
- **Independence**: Observations should be independent of each other.
- **Linearity of logit**: The logit (log-odds) of the outcome should be linearly related to the predictors.
- **No multicollinearity**: The predictors should not be highly correlated with each other.
- **Large sample size**: Logistic regression typically requires a large sample size to provide reliable estimates.

7 Summary

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```

References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.