Cookbook Data Analysis with Stata and R

Manuel Oliveira

August, 2024

Table of contents

Preface **5**

1 Introduction **6**

2 Getting Started **7**
2.1 Introduction 7
2.2 Installing R 7
2.2.1 Windows 7
2.2.2 Mac 7
2.3 Installing Stata 8
2.3.1 Windows 8
2.3.2 Mac 8
2.4 Setting Up Your Environment 9
2.4.1 R Setup 9
2.4.2 Stata Setup 9
2.5 Verification 9
2.5.1 R 9
2.5.2 Stata 10

3 ANOVA **11**
3.1 Introduction 11
3.2 Example Question 11
3.3 Required Packages (R) 11
3.4 Simulating the Dataset in R 12
3.5 Simulating the Dataset in Stata 13
3.6 Visualizing the Descriptives in R 13
3.7 Visualizing the Descriptives in Stata 14
3.8 Running the ANOVA in R 14
3.9 Running the ANOVA in Stata 15
3.10 Interpreting the Output 15
3.10.1 In R 15
3.10.2 In Stata 15
3.11 Post-hoc Testing in R 15
3.12 Post-hoc Testing in Stata 16
3.13 Plotting the Results in R 16

3.14 Plotting the Results in Stata . . . 17
3.15 Assumptions . . . 17
3.16 Syntax Comparison: R vs Stata . . . 18

4 Linear Regression **19**
4.1 Introduction . . . 19
4.2 Example Question . . . 19
4.3 Dataset Simulation in R . . . 19
4.4 Dataset Simulation in Stata . . . 20
4.5 Performing Linear Regression . . . 20
4.5.1 R . . . 20
4.5.2 Stata . . . 20
4.6 Assumptions . . . 20

5 Multilevel Regression **21**
5.1 Introduction . . . 21
5.2 Example Question . . . 21
5.3 Dataset Simulation in R . . . 21
5.4 Dataset Simulation in Stata . . . 22
5.5 Performing Multilevel Regression . . . 22
5.5.1 R . . . 22
5.5.2 Stata . . . 22
5.6 Assumptions . . . 23

6 Logistic Regression **24**
6.1 Introduction . . . 24
6.2 Example Question . . . 24
6.3 Required Packages (R) . . . 24
6.4 Simulating the Dataset in R . . . 25
6.5 Simulating the Dataset in Stata . . . 25
6.6 Visualizing the Descriptives in R . . . 26
6.7 Visualizing the Descriptives in Stata . . . 26
6.8 Running the Logistic Regression in R . . . 27
6.9 Running the Logistic Regression in Stata . . . 27
6.10 Interpreting the Output . . . 28
6.10.1 In R . . . 28
6.10.2 In Stata . . . 28
6.11 Plotting the Results in R . . . 28
6.12 Plotting the Results in Stata . . . 29
6.13 Assumptions . . . 29
6.13.1 In R and Stata . . . 29
6.14 Syntax Comparison: R vs Stata . . . 30

7 Summary 31

References 32

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 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 Required Packages (R)

```r
# Load the necessary packages
library(tidyverse) # used for data manipulation and visualization
```

```
Warning: package 'ggplot2' was built under R version 4.2.3
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.1     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(car) # provides tools for ANOVA and regression diagnostics
```

```
Loading required package: carData

Attaching package: 'car'

The following object is masked from 'package:dplyr':

    recode

The following object is masked from 'package:purrr':

    some
```

```r
# to install any missing packages go to the Terminal and run the command: install.packages
```

3.4 Simulating the Dataset in R

```r
# Setting a seed for reproducibility
set.seed(123)

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

# Creating a factor variable for UI design
ui_design <- factor(rep(1:n_groups, each = n_per_group))

# Simulating time spent data with different means for each UI design
time_spent <- rnorm(n_groups * n_per_group, mean = rep(c(20, 25, 22), each = n_per_group),

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

# Viewing the first few rows of the dataset
head(data)
```

```
  ui_design time_spent
```

```
1          1   17.19762
2          1   18.84911
3          1   27.79354
4          1   20.35254
5          1   20.64644
6          1   28.57532
```

3.5 Simulating the Dataset in Stata

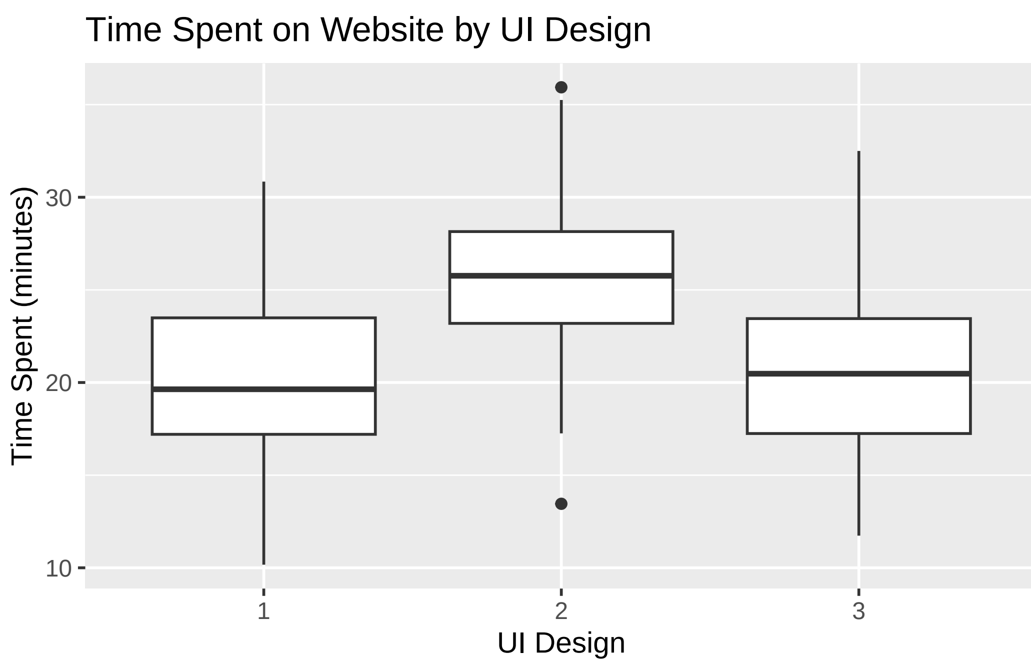
```
* Set seed for reproducibility
set seed 123

* 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)

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

3.6 Visualizing the Descriptives in R

```
# Plotting the distribution of time spent across different UI designs
ggplot(data, aes(x = ui_design, y = time_spent)) +
  geom_boxplot() +
  labs(title = "Time Spent on Website by UI Design",
       x = "UI Design",
       y = "Time Spent (minutes)")
```



3.7 Visualizing the Descriptives in Stata

```
* Box plot of time spent by UI design
graph box time_spent, over(ui_design) title("Time Spent on Website by UI Design") ///
    ytitle("Time Spent (minutes)") xtitle("UI Design")
```

3.8 Running the ANOVA in R

```
# Performing ANOVA
anova_model <- aov(time_spent ~ ui_design, data = data)

# Viewing the summary of the ANOVA model
summary(anova_model)
```

```
             Df Sum Sq Mean Sq F value   Pr(>F)    
ui_design     2    937   468.7   21.18 8.29e-09 ***
Residuals   147   3253    22.1                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.9 Running the ANOVA in Stata

```
* Perform ANOVA
anova time_spent ui_design
```

3.10 Interpreting the Output

3.10.1 In R

The ANOVA table provides the following key pieces of information: - **Df**: Degrees of freedom associated with the sources of variance. - **Sum Sq**: Sum of squares, which measures the total variation for each source. - **Mean Sq**: Mean square, calculated as Sum Sq divided by Df. - **F value**: The F-statistic, calculated as the ratio of mean square values. - **Pr(>F)**: The p-value associated with the F-statistic.

3.10.2 In Stata

The output of the ANOVA in Stata provides similar information: - **Source**: Lists the sources of variance. - **Partial SS**: Partial sum of squares for each source. - **df**: Degrees of freedom associated with each source. - **MS**: Mean square for each source, calculated as SS/df. - **F**: The F-statistic for each source. - **Prob > F**: The p-value associated with the F-statistic.

If the p-value is less than the significance level (typically 0.05), we reject the null hypothesis that all group means are equal.

3.11 Post-hoc Testing in R

```
# Performing Tukey's Honest Significant Difference test
tukey_test <- TukeyHSD(anova_model)

# Viewing the Tukey test results
tukey_test
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = time_spent ~ ui_design, data = data)
```

```
$ui_design
          diff       lwr       upr     p adj
2-1  5.5600236  3.332272  7.787775 0.0000001
3-1  0.5584801 -1.669271  2.786231 0.8237890
3-2 -5.0015435 -7.229295 -2.773792 0.0000012
```

3.12 Post-hoc Testing in Stata

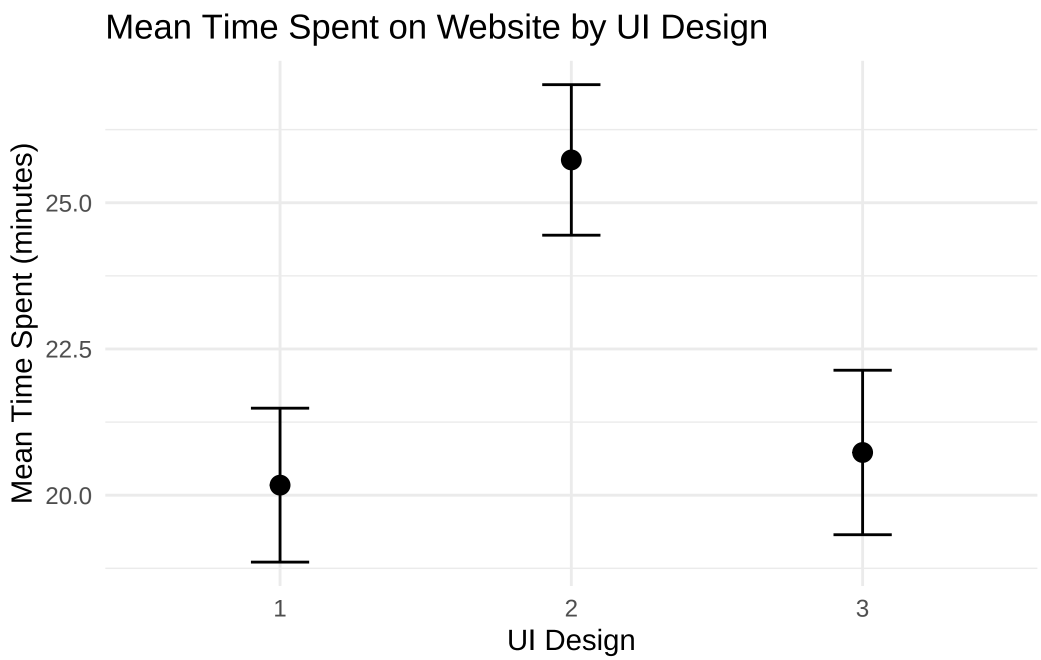
```
* Perform Bonferroni post-hoc test
oneway time_spent ui_design, bonferroni
```

3.13 Plotting the Results in R

```
# Plotting the results of the Tukey HSD test
plot(tukey_test, las = 1)
```


```
# Creating a plot to visualize group means with confidence intervals
ggplot(data, aes(x = ui_design, y = time_spent)) +
```

```
stat_summary(fun.data = mean_cl_normal, geom = "errorbar", width = 0.2) +
stat_summary(fun = mean, geom = "point", size = 3) +
labs(title = "Mean Time Spent on Website by UI Design",
     x = "UI Design",
     y = "Mean Time Spent (minutes)") +
theme_minimal()
```



3.14 Plotting the Results in Stata

```
* Plot group means with confidence intervals
means time_spent, over(ui_design) ci
```

3.15 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.

These assumptions should be checked to ensure the validity of the ANOVA results.

3.16 Syntax Comparison: R vs Stata

This table summarizes the main differences between R and Stata in terms of syntax for performing ANOVA analyses.

Task	R Command	Stata Command
Simulating Data	`rnorm()` for simulating normal distribution	`rnormal()` for simulating normal distribution
Setting Seed for Reproducibility	`set.seed(123)`	`set seed 123`
Creating a Factor Variable	`factor()`	`gen variable` and `egen group`
Visualizing Descriptives	`ggplot()` with `geom_boxplot()`	`graph box`
Running ANOVA	`aov()` and `summary()`	`anova`
Post-hoc Testing	`TukeyHSD()`	`oneway` with `bonferroni` option
Plotting Group Means with Confidence Intervals	`ggplot()` with `stat_summary()`	`means` with `ci` option

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

```
# 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

```
# 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. We will use an example dataset to investigate 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 Required Packages (R)

```
# Load the necessary packages
library(tidyverse) # used for data manipulation and visualization
```

```
Warning: package 'ggplot2' was built under R version 4.2.3
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.1     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(broom) # for tidying the model output, making it easier to work with

# to install any missing packages go to the Terminal and run the command: install.packages
```

6.4 Simulating the Dataset in R

```r
# Setting a seed for reproducibility
set.seed(123)

# Simulating 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))))

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

# Viewing the first few rows of the dataset
head(data)
```

```
  support_contact continued_use
1               1             0
2               3             0
3               2             1
4               4             1
5               4             1
6               0             1
```

6.5 Simulating the Dataset in Stata

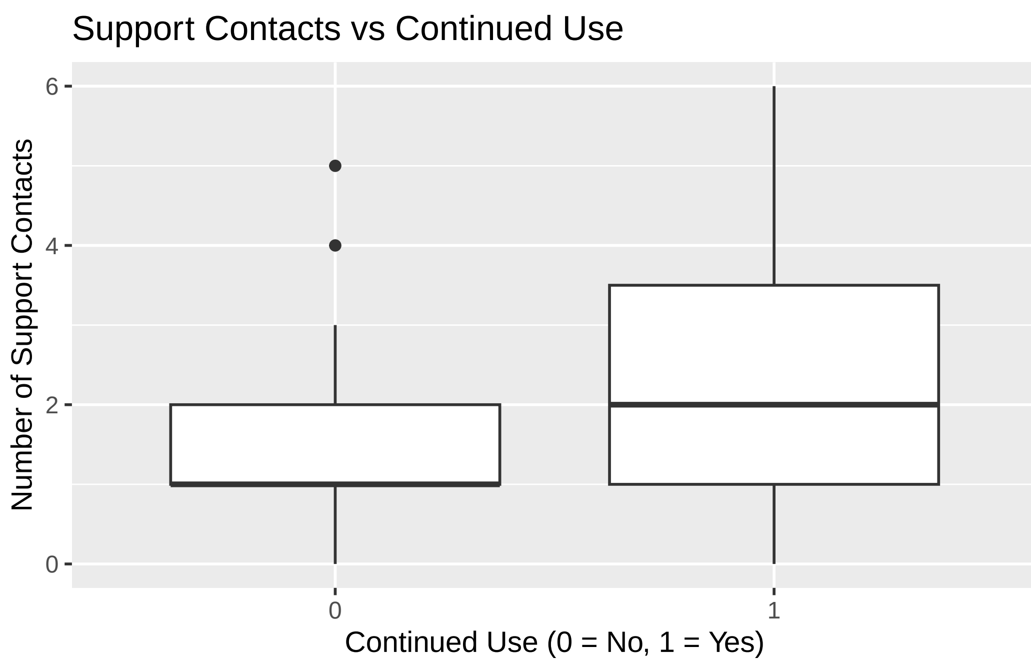
```stata
* Set seed for reproducibility
set seed 123

* 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
```

6.6 Visualizing the Descriptives in R

```
# Plotting the distribution of support contacts for users who continued vs those who didn'
ggplot(data, aes(x = factor(continued_use), y = support_contact)) +
  geom_boxplot() +
  labs(title = "Support Contacts vs Continued Use",
       x = "Continued Use (0 = No, 1 = Yes)",
       y = "Number of Support Contacts")
```



6.7 Visualizing the Descriptives in Stata

```
* Box plot of support contacts by continued use
graph box support_contact, over(continued_use) title("Support Contacts vs Continued Use")
    ytitle("Number of Support Contacts") xtitle("Continued Use (0 = No, 1 = Yes)")
```

6.8 Running the Logistic Regression in R

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

# Viewing the summary of the logistic regression model
summary(logistic_model)
```

```
Call:
glm(formula = continued_use ~ support_contact, family = "binomial",
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2697  -0.9883   0.5643   1.0621   1.7118

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.2023     0.3046  -3.947 7.90e-05 ***
support_contact   0.7398     0.1453   5.092 3.54e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 274.83  on 199  degrees of freedom
Residual deviance: 240.15  on 198  degrees of freedom
AIC: 244.15

Number of Fisher Scoring iterations: 4
```

6.9 Running the Logistic Regression in Stata

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

6.10 Interpreting the Output

6.10.1 In R

The summary of the logistic regression model provides the following key pieces of information: - **Coefficients**: Estimates of the regression coefficients. - **Std. Error**: Standard errors of the coefficients. - **z value**: The test statistic for each coefficient. - **Pr(>|z|)**: The p-value associated with each coefficient, indicating whether it is statistically significant.

6.10.2 In Stata

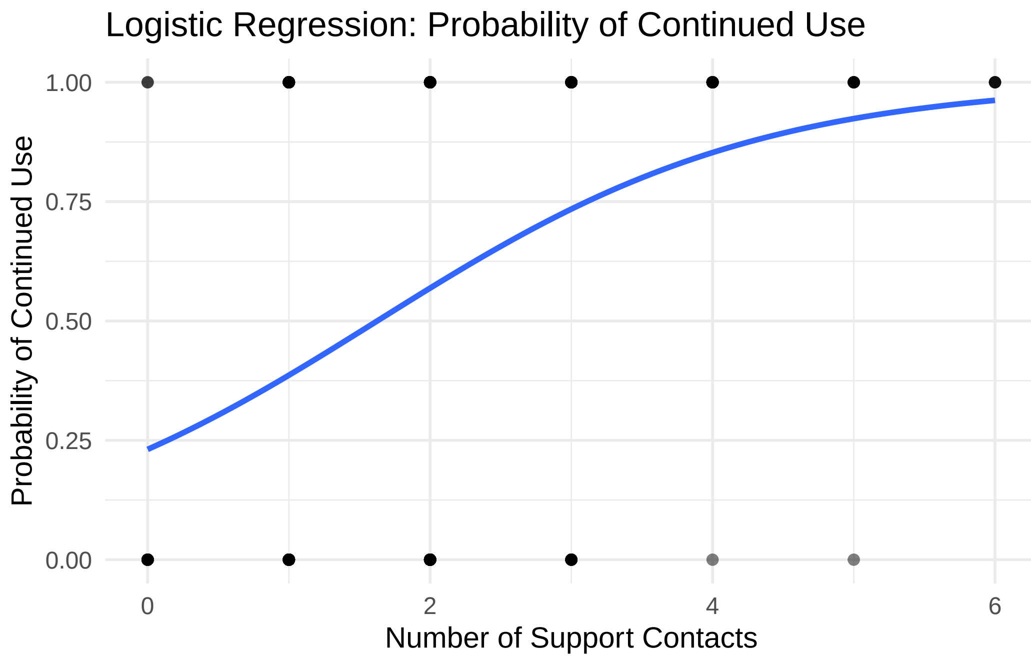
The output of the logistic regression in Stata provides similar information: - **Coef.**: Estimates of the regression coefficients. - **Std. Err.**: Standard errors of the coefficients. - **z**: The test statistic for each coefficient. - **P>|z|**: The p-value associated with each coefficient, indicating whether it is statistically significant.

If the p-value is less than the significance level (typically 0.05), we reject the null hypothesis that the coefficient is equal to zero.

6.11 Plotting the Results in R

```r
# Plotting the logistic regression curve
ggplot(data, aes(x = support_contact, y = continued_use)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "glm", method.args = list(family = "binomial"), se = FALSE) +
  labs(title = "Logistic Regression: Probability of Continued Use",
       x = "Number of Support Contacts",
       y = "Probability of Continued Use") +
  theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



6.12 Plotting the Results in Stata

```
* Create logistic regression plot (approximation)
twoway (scatter continued_use support_contact) (lfit continued_use support_contact, ci)
```

6.13 Assumptions

6.13.1 In R and Stata

- **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.

These assumptions should be checked to ensure the validity of the logistic regression results.

6.14 Syntax Comparison: R vs Stata

This table summarizes the main differences between R and Stata in terms of syntax for performing Logistic Regression analysis.

Task	R Command	Stata Command
Simulating Data	`rpois()`, `rbinom()`	`rpoisson()`, `rbinomial()`
Setting Seed for Reproducibility	`set.seed(123)`	`set seed 123`
Visualizing Descriptives	`ggplot()` with `geom_boxplot()`	`graph box`
Running Logistic Regression	`glm()` with `family = "binomial"`	`logit`
Plotting the Results	`ggplot()` with `geom_smooth(method = "glm", ...)`	`twoway scatter` and `lfit`

7 Summary

In summary, this book has no content whatsoever.

```r
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.