Modeling in Education, and Social Sciences Using R

William M. Murrah

2021-08-16

Contents

Preface

Chapter 1

Introduction to Simple Models

This chapter of the notebook corresponds to chapters 1 and 2 of Judd et al. (2017).

Chapter 2

Sampling Distributions and Statistical Inference

This chapter of the notebook corresponds to chapters 3 and 4 of Judd et al. (2017).

Chapter 3

Simple Regression

This chapter of the notebook corresponds to chapter 5 of Judd et al. (2017).

Chapter 4

Model Evaluation

This chapter of the notebook corresponds to Chapter 13 of Judd et al. (2017).

Chapter 5

Models with Multiple Categorical Predictors

This chapter of the notebook corresponds to chapter 6 of Judd et al. (2017).

Chapter 6

Moderation and Nonlinear Models

This chapter of the notebook corresponds to chapter 7 of Judd et al. (2017).

Chapter 7

Models with Single Categorical Predictor

This chapter of the notebook corresponds to chapter 8 of Judd et al. (2017).

Chapter 8

Models with Multiple Categorical Predictors

This chapter of the notebook corresponds to chapter 9 of Judd et al. (2017).

Chapter 9

Models with Continuous and Categorical Predictors

This chapter of the notebook corresponds to chapter 10 of Judd et al. (2017).

Chapter 10

Models with Nonindependent Errors

This chapter of the notebook corresponds to chapter 11 of Judd et al. (2017).

Chapter 11

Incorporating Predictors with Nonindependent Data

This chapter of the notebook corresponds to chapter 12 of Judd et al. (2017).

Chapter 12

Logistic Models for Categorical Dependent Variables

This chapter of the notebook corresponds to chapter 14 of Judd et al. (2017).

Appendix A

Appendix A: R as a Statistical Programming Language

A.1 Overview

```
library(knitr)
```

- Why use Statistical Programming Languages?

Computers and Programs

- Fixed program computer
  - programs are "hard-wired" into computer
  - calculator, stopwatch
- Stored program computer
  - machine stores and executes instructions
  - most modern computers, your phone, etc.

Statistical Programs versus Statistical Programming Language

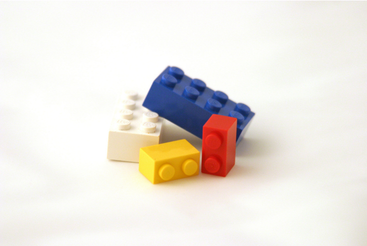
- Statistical Programs
  - fixed menus
  - limited procedures (at least in the menus)
  - leads to compartmentalizing models (e.g. ANOVA, regression, GLM)
- Statistical Programming Languages (SPLs)
  - Turing complete: if you can create an algorithm you can program it
  - Very flexible
  - Integration of models: One model to rule them all!

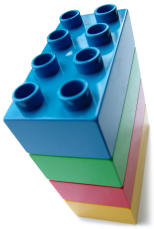
Everythin

A.2 Elements of Statistical Programming

A.2.1 Basic Elements of a Good SPL

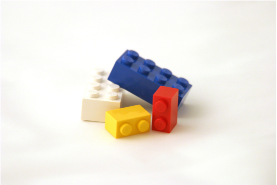
1. a rich set of **primitive expressions**
2. mechanisms for **combining expressions** into more complex expressions
3. means of **abstraction**, which allow for naming and manipulating compound objects





A.3 Expressions

A.4 Primitive Expressions



- Everything in R is an object
- Primitive objects are the simplest elements of a programming language, and include:
  - *primitive data*
  - *primitive functions*
- They can be thought of as the basic building blocks for everything else in the language.
- An **expression** is an input that the programming language can evaluate, and consists of function and data objects.

A.5 Primitive Data Types:

Data objects are the primary means of storing information in R. R has a few basic *data types*:

- **Numeric** -
  - `numeric`
    - `int` - integers (`1,2`)
    - `num` - real number (`1.2, -3.1, 200.0`)
- **character** or **string** -
  - `character`
    - `"Hello world!"`, `"Ten"`, `'Cat'`
    - `"This is a sentence, which is a string"`
    - `"10"` (in single or double quotes, as long as they match)
- **Boolean** or **Logical**
  - `logical`
    - `TRUE` or `FALSE` (use operators such as *or*, *and* and *not*).
    - They will evaluate to numbers where `FALSE` evaluates to zero, and `TRUE` evaluates to one.
    - For example. if you enter `TRUE + 1` you will get `2` in return.

```
mode(TRUE)
```

```
## [1] "logical"
```

```
TRUE + 1
```

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

A.6 Primitive Functions

R uses functions to do all computations.

A.6.1 Operators

- Arithmetic Operators
  - +, -, *, /, ^
- Comparison (also called Boolean, Logical or Predicate) Operators
  - `<, >, ==, <=, >=, !=`
  - less than, greater than, equal to, less than or equal to, greater than or equal to, not equal to
  - return `TRUE` or `FALSE`
- Logical Operator
  - `&`, `|` ,`!`
  - also return `TRUE` or `FALSE`
- Other functions
  - `mode()`
  - `length()`
  - `sum()`
  - `sqrt()`
  - `log()`
  - `exp()`
- Assignment operators (assignment will be discussed below)
  - **`<-` preferred assignment operator - always use this one**
  - `=` this will also work, but can be confusing (note different from `==`, the comparison operator)
  - `->` is also an assignment operator, but we will not use it.

A.7 Programming Languages are Not Forgiving

A.7.1 Syntactically valid expressions

Expressions must be syntactically valid.

- syntax (form)
  - English: “cat dog boy” - not syntactically valid
  - English: “cat hugs boy” - syntactically valid

- programming language:
    - “hi” 5 - not syntactically valid
    - 3.2*5 - syntactically valid

A.7.2 Semantically valid expressions

- semantics - (meaning)
    - English: “I are hungry” - syntactically valid but semantic error
    - programming language:
        - 3 + “hi” - semantic error (you can’t use addition on character strings)
- Chomsky: “colorless green ideas sleep furiously”

This statement is syntactically valid, but does not make sense, so makes a semantic error.

A.8 Assignment

We will often want to save data in a variable. We can do that with **assignment**, which utilizes an assignment operator.

```
x <- 2
```

```
x
```

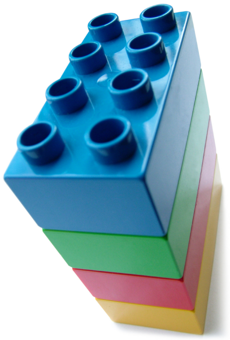
```
## [1] 2
```

```
pet <- "dog"
```

```
pet
```

```
## [1] "dog"
```

A.9 Combining Expressions



A.10 Complex Data Types

- Scalars, Vectors, Matrices, and Arrays
- Lists
- Dataframes

A.11 Grouping Homogeneous Data Types

- combining scalars

```
c()
```

- combining expressions

```
{}
```

- combining vectors

```
cbind()
rbind()
```

A.12 Complex Functions

- Vectorization
- Nested Functions
- Loops and Conditional execution

class: inverse, center, middle

A.13 Abstraction

A.14 Abstraction

- Assignment
-

A.15 Data Abstraction

A.16 Functional Abstraction

A.17 Anatomy of a Function

```
name <- function(arg_1, arg_2, ...) expression
```

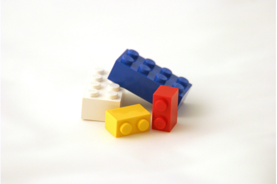
A.18 Teaching With A Statistical Programming Language

A.18.1 An Example

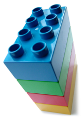
A.19 myMean

A.20 Basic Elements of a Good SPL

A rich set of **primitive expressions**



Mechanisms for **combining expressions** into more complex expressions



Means of **abstraction**, which allow for naming and manipulating compound objects

Bibliography

Judd, C. M., McClelland, G. H., and Ryan, C. S. (2017). *Data analysis: A model comparison approach to regression, ANOVA, and beyond*. Routledge.