An Introduction to Data Analysis using Programming

Sumedh Girish

Contents

How to use this book

Introduction to the R Programming Paradigm

- R, while being powerful lacks abstraction
- To avoid having to reinvent the wheel, we use packages in R, which is basically code written by other people to make our life simpler
- The most popular and widely used R package(s) come under the umbrella of the "tidyverse"
- It streamlines the process of dealing with data, standardizing a way to write R code for data analysis

```
library("tidyverse")
```

Does working more hours make you smoke more?

1. Importing the data

- We use the readr library to read data in R

```
# Reading from a csv file ../raw_data/ESS11.csv
filename <- "../raw_data/ESS11.csv"
datatibble <- read_csv(filename, show_col_types = FALSE)
print(datatibble)
```

```
## # A tibble: 40,156 x 640
##    name   essround edition proddate  idno cntry dweight pspwght pweight anweight
##    <chr>     <dbl>   <dbl> <chr>    <dbl> <chr>   <dbl>   <dbl>   <dbl>    <dbl>
##  1 ESS11~       11       2 20.11.2~ 50014 AT      1.19    0.393   0.331   0.130
##  2 ESS11~       11       2 20.11.2~ 50030 AT      0.610   0.325   0.331   0.108
##  3 ESS11~       11       2 20.11.2~ 50057 AT      1.39    4.00    0.331   1.32
##  4 ESS11~       11       2 20.11.2~ 50106 AT      0.556   0.176   0.331   0.0583
##  5 ESS11~       11       2 20.11.2~ 50145 AT      0.723   1.06    0.331   0.351
##  6 ESS11~       11       2 20.11.2~ 50158 AT      0.993   1.39    0.331   0.461
##  7 ESS11~       11       2 20.11.2~ 50211 AT      0.540   0.577   0.331   0.191
##  8 ESS11~       11       2 20.11.2~ 50212 AT      0.815   0.619   0.331   0.205
##  9 ESS11~       11       2 20.11.2~ 50213 AT      1.36    0.694   0.331   0.230
## 10 ESS11~       11       2 20.11.2~ 50235 AT      0.873   0.492   0.331   0.163
## # i 40,146 more rows
## # i 630 more variables: nwspol <dbl>, netusoft <dbl>, netustm <dbl>,
## #   ppltrst <dbl>, pplfair <dbl>, pplhlp <dbl>, polintr <dbl>, psppsgva <dbl>,
## #   actrolga <dbl>, psppipla <dbl>, cptppola <dbl>, trstprl <dbl>,
## #   trstlgl <dbl>, trstplc <dbl>, trstplt <dbl>, trstprt <dbl>, trstep <dbl>,
## #   trstun <dbl>, vote <dbl>, prtvtdat <dbl>, prtvtebe <dbl>, prtvtchr <dbl>,
## #   prtvtccy <dbl>, prtvtffi <dbl>, prtvtffr <dbl>, prtvgde1 <dbl>, ...
```

- Notice how the information we want is just a subset of this data

2. Filtering the data we have

- We use the dplyr library to do this

```r
filtered_data <- datatibble |>
  select("cgtsmok", "wkhct", "wkhtot", "tporgwk") |>
  drop_na()
print(filtered_data)
```

```
## # A tibble: 40,156 x 4
##    cgtsmok wkhct wkhtot tporgwk
##      <dbl> <dbl>  <dbl>   <dbl>
##  1       4   666    666      66
##  2       5    32     32       4
##  3       1    25     25       6
##  4       6    39     39       2
##  5       1    40     35       4
##  6       5    40     40       4
##  7       6    30     30       4
##  8       4    40     50       1
##  9       5    38     40       2
## 10       4    40     40       4
## # i 40,146 more rows
```

- notice how data has a lot of invalid row values

```r
cleaned_data <- filtered_data |>
  filter(cgtsmok >= 1 & cgtsmok <= 6) |>
  filter(tporgwk >= 1 & tporgwk <= 6) |>
  filter(wkhct >= 0 & wkhct <= 168) |>
  filter(wkhtot >= 0 & wkhtot <= 168)
print(cleaned_data)
```

```
## # A tibble: 31,668 x 4
##    cgtsmok wkhct wkhtot tporgwk
##      <dbl> <dbl>  <dbl>   <dbl>
##  1       5    32     32       4
##  2       1    25     25       6
##  3       6    39     39       2
##  4       1    40     35       4
##  5       5    40     40       4
##  6       6    30     30       4
##  7       4    40     50       1
##  8       5    38     40       2
##  9       4    40     40       4
## 10       5    38     38       4
## # i 31,658 more rows
```