Lecture 6 Notes

Damian Thomas

2017-10-06

Contents

Setup . . . 2
Introduction . . . 2
Objectives . . . 2
Financial Literacy Topics . . . 2
Programming Topics . . . 2
Reproducibile Research . . . 2
    Reproduciblity . . . 2
    Literate Programming . . . 2
    Replicability . . . 3
Introduce R Markdown . . . 3
Live coding . . . 3
Create an R Markdown Document in RStudio . . . 3
Main parts of an R Markdown document . . . 3
Rendering a document . . . 3
Editing a Document . . . 4
    Header . . . 4
    Markdown text . . . 4
    LaTeX (Math) . . . 4
Code Blocks . . . 4
    Edit the code block with the summary . . . 4
    Edit the code block with the plot . . . 4
    Run code blocks interactively . . . 4
    Code block options . . . 5
Inline R Code . . . 5
Start new document . . . 5
Add an introduction . . . 5
Take the quiz . . . 5
Take notes . . . 5
Question 1 . . . 6
Question 1: R code . . . 6
Question 2 . . . 8
Question 2: R code . . . 8
Question 2: Plots . . . 9
Survey Data . . . 10
Create a new section heading . . . 10
Create directories . . . 10
Download the raw data, unzip . . . 11
Import the raw data . . . 11
Clean the data, filter an reformat variables . . . 11
References . . . 19

Setup

- Etherpad page https://etherpad.net/ecog314_2017-10-06
- Create a directory for today's session
- Download lecture files https://github.com/wampeh1/Ecog314_Fall2017

Introduction

- About the instructor
- Announcements

Objectives

- Write a program and present results using R Markdown
- Define reproduciblity and explain the benefits
- Review R programming topics
- Illustrate financial literacy concepts
- List investment strategies for ordinary investors

Financial Literacy Topics

- Interest rates and inflation
- Risk and diversification
- The relationship between bond prices and interest rates
- The impact that a shorter term can have on total interest payments

Programming Topics

- Reproducible Research
- R Markdown

Reproducibile Research

Reproduciblity

The ability to repeat an analysis and get the same result with the same data

- *What did you do? What did you use?*
- Repeatable

Literate Programming

A programming paradigm where programs are written with a mix of natural language explanations of the steps taken and computer code to carry it out.

- *What were you thinking? Did you explain your decisions?*
- Communicate

Replicability

The ability to duplicate an analysis and get the same results with new data

- Were your results a fluke? Do your findings hold using different statistical samples?
- Corroborate

Introduce R Markdown

R Markdown package:

- Combine R code, output, and formatted text
- Encourages Reproducible Research through Literate Programming.
- Creates nicely formatted documents that include output from R
- Useful for communicating results, collaborating with others, and documenting your work

References

- http://rstudio.com/cheatsheets
- http://r4ds.had.co.nz/r-markdown.html

Live coding

- Create an R Markdown Document in RStudio
- Demonstrate R Markdown features

Create an R Markdown Document in RStudio

1. From the menu: File >> New >> R Markdown
2. Take the defaults
3. Click OK

This opens a new file in the source pane, with a template created by RStudio.

Main parts of an R Markdown document

1. YAML header
2. R code chunks
3. Markdown text

Rendering a document

1. Save the file with `.Rmd` file extension
2. Knit the document
    - Click on the "Knit" button,
    - Type `Ctrl + Shift + K`
    - Use the menu: File >> Knit Document
    - console command

Editing a Document

Refer to http://rstudio.com/cheatsheets

Header

- title
- date
- author
- change the output format
  - `html_document`
  - `pdf_document`
  - `beamer_presentation`

Markdown text

Add text to the body

- Add / delete text
- bold, italics, underline
- headings
- lists
- links
- tables

LaTeX (Math)

Reference: https://en.wikibooks.org/wiki/LaTeX/Mathematics

- Add an inline equation: `$ E = mc^{2} $`
- Add a centered Equation: `$$ y = \beta_{0} + \beta_{1}x_{1} + \beta_{2}x_{2} + \epsilon $$`
- Complex math: $\forall x \in \mathbb{R}^1, \sqrt{x} = i \implies x = -1$

Code Blocks

Edit the code block with the summary

- change the data frame to `mtcars`
- run the `str` function
- print the data frame

Edit the code block with the plot

- load ggplot2 library
- create a scatterplot of speed vs stopping distance

Run code blocks interactively

- line-by-line
- entire code block

Code block options

- specify programming language
- code chunk name
- `echo = TRUE` – Show the code
- `results = "hide"` – run the code, show the code, but do not show the results
- `include = FALSE` – run the code, but do not show either the code or the results
- `eval = FALSE` – show the code, but do not run it
- `cache = TRUE` – save the output
- `results = "asis"`

Inline R Code

Enclose the expression in backticks `‘r` with the name of the language

`Pick a random number between 1 and 10: ‘r sample(1:10, size = 1) ‘`

Renders as:

Pick a random number between 1 and 10: 9

Start new document

- Create a new R Markdown file
- Take the defaults
- Clear the template

Add an introduction

What do we intend to do?

The FINRA Investor Education Foundation conducted a survey to study how people manage their resources and make financial decisions. The survey includes several benchmark questions to assess understanding of key financial concepts. In today's class we will try a few of these benchmark questions, illustrate the concepts using R code, and explore the nationwide survey results.

- Insert a link to the FINRA homepage: http://www.usfinancialcapability.org/
- Save the file, knit it, and click on the link in the rendered document

Take the quiz

http://www.usfinancialcapability.org/quiz.php

Take notes

Describe what we are doing: taking the quiz, getting familiar with the survey.

- Add two sections with headings to the document:
    1. “Taking the online survey”, “About the survey”, etc.

    In order to familiarize ourselves with the survey, we took the online financial literacy quiz. The first question deals with interest earning savings…

```
2. "References"
```

"Financial Literacy Quiz," Take the Financial Literacy Quiz, FINRA Investor Education Foundation, Accessed October 6, 2017, http://www.usfinancialcapability.org/quiz.php.

Question 1

Suppose you have $100 in a savings account earning 2 percent interest a year. After five years, how much would you have?

a. More than $102
b. Exactly $102
c. Less than $102
d. Don't Know

- Key concept: interest earning assets gain value as time passes

To show that we have answered correctly, we will model the accumulation of interest using R

Given:

- r is the interest rate where $r \in [0, 1]$
- b is a vector of account balances
- $b_t = b_{t-1} + (rb_{t-1})$

$$b = \begin{bmatrix} b_t \\ b_{t+1} \\ \vdots \\ b_{t+n} \end{bmatrix}$$

Goal: create a vector where each element is the account balance at a different point in time

Question 1: R code

To implement this mental model in R, start with the fixed parameters, then use a for-loop to compute the interest in each period

```r
interest_rate <- 0.02
balance <- 100 # starting balance
n <- 5

for ( i in 1:n ) {
    t <- i + 1
    balance[t] <- balance[t-1]  + (interest_rate * balance[t-1])
}
```

Based on this model we can confirm that: after 5 periods of compounding the account balance will be 110.4080803. A complete table of values is presented below.

```r
# take the vector and create a data frame

account <- data.frame(
    t = seq(0, n),
    balance = balance
)
```

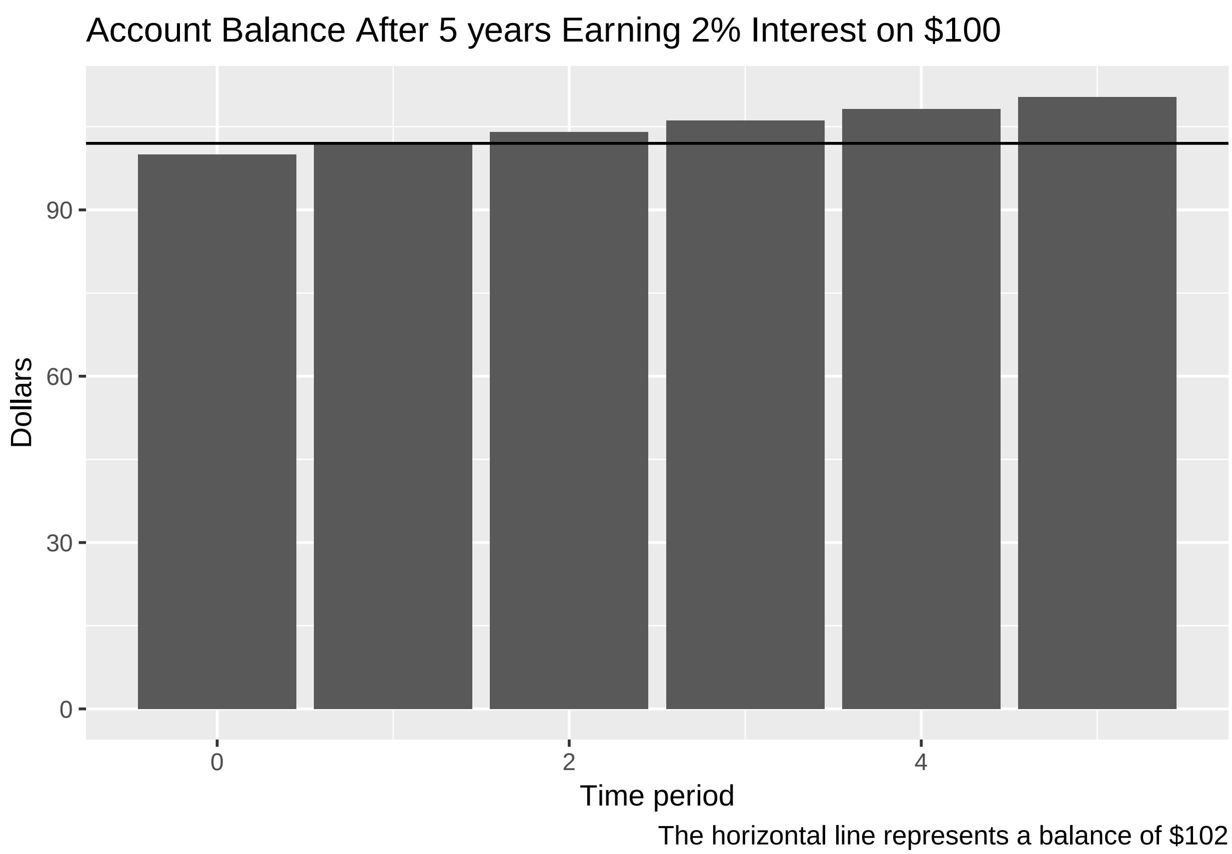
```r
knitr::kable(account, caption = "Table of Account Balances")
```

Table 1: Table of Account Balances

t	balance
0	100.0000
1	102.0000
2	104.0400
3	106.1208
4	108.2432
5	110.4081

Let's visualize these results.
* Add a code block * load dplyr and ggplot2 * Suppress the messages and hide the code

```r
account %>%
    ggplot(mapping = aes(x = t, y = balance)) +
    geom_bar(stat = "identity") +
    geom_hline(yintercept = 102) +
    labs(title = "Account Balance After 5 years Earning 2% Interest on $100",
         caption = "The horizontal line represents a balance of $102",
         x = "Time period",
         y = "Dollars")
```



Question 2

Imagine that the interest rate on your savings account is 1 percent a year and inflation is 2 percent a year. After one year, would the money in the account buy more than it does today, exactly the same or less than today?

- Inflation reduces the buying power of the savings account. The nominal value is not the same thing as real value.
- Let's compute the Inflation adjusted returns. Using the investopedia rule.
- Add a reference to the source at the end of the document

  "Inflation-Adjusted Returns", Inflation-Adjusted Returns, accessed 2017-10-07, http://www.investopedia.com/terms/i/inflation_adjusted_return.asp

Question 2: R code

```
years <- 2017:2030
starting_balance <- 100
interest_rate  <- 0.01
inflation_rate <- 0.02
adjusted_rate <- ((1 + interest_rate) / (1 + inflation_rate)) - 1
adjusted_rate
```

```
## [1] -0.009803922
```

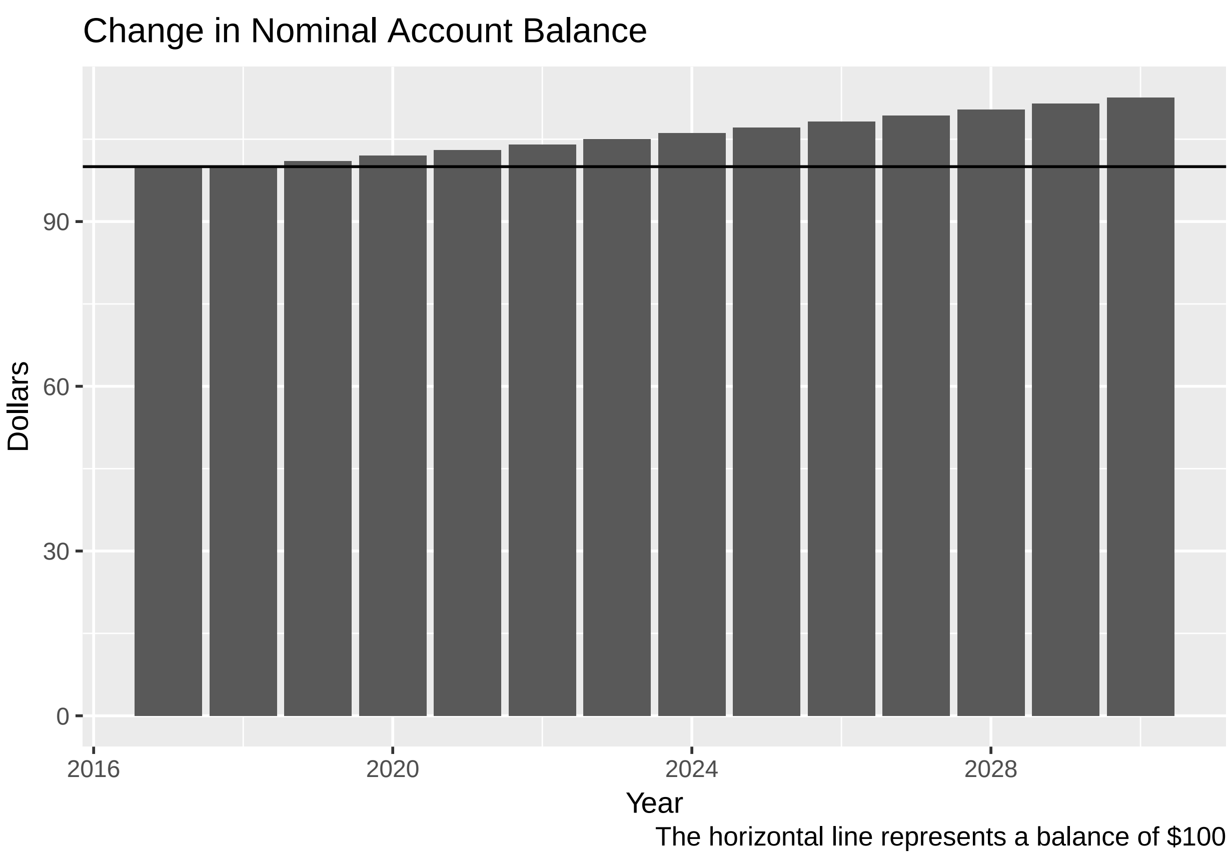
```
nominal_balance  <- c()
nominal_gains    <- c()
adjusted_gains   <- c()
adjusted_balance <- c()

for ( t in seq_along(years) ) {
    if ( t == 1 ) {
        nominal_gains[t] <- 0
        adjusted_gains[t] <- 0
        nominal_balance[t] <- starting_balance
        adjusted_balance[t] <- nominal_balance[t]
    } else {
        nominal_gains[t] <- interest_rate * nominal_balance[t-1]
        adjusted_gains[t] <- adjusted_rate * adjusted_balance[t-1]
        nominal_balance[t] <- nominal_balance[t-1] + nominal_gains[t-1]
        adjusted_balance[t] <- adjusted_balance[t-1] + adjusted_gains[t-1]
    }
}

account <- data.frame(year = years,
                      nominal_gains,
                      adjusted_gains,
                      nominal_balance,
                      adjusted_balance)

account
```

```
##    year nominal_gains adjusted_gains nominal_balance adjusted_balance
## 1  2017      0.000000      0.0000000        100.0000        100.00000
```

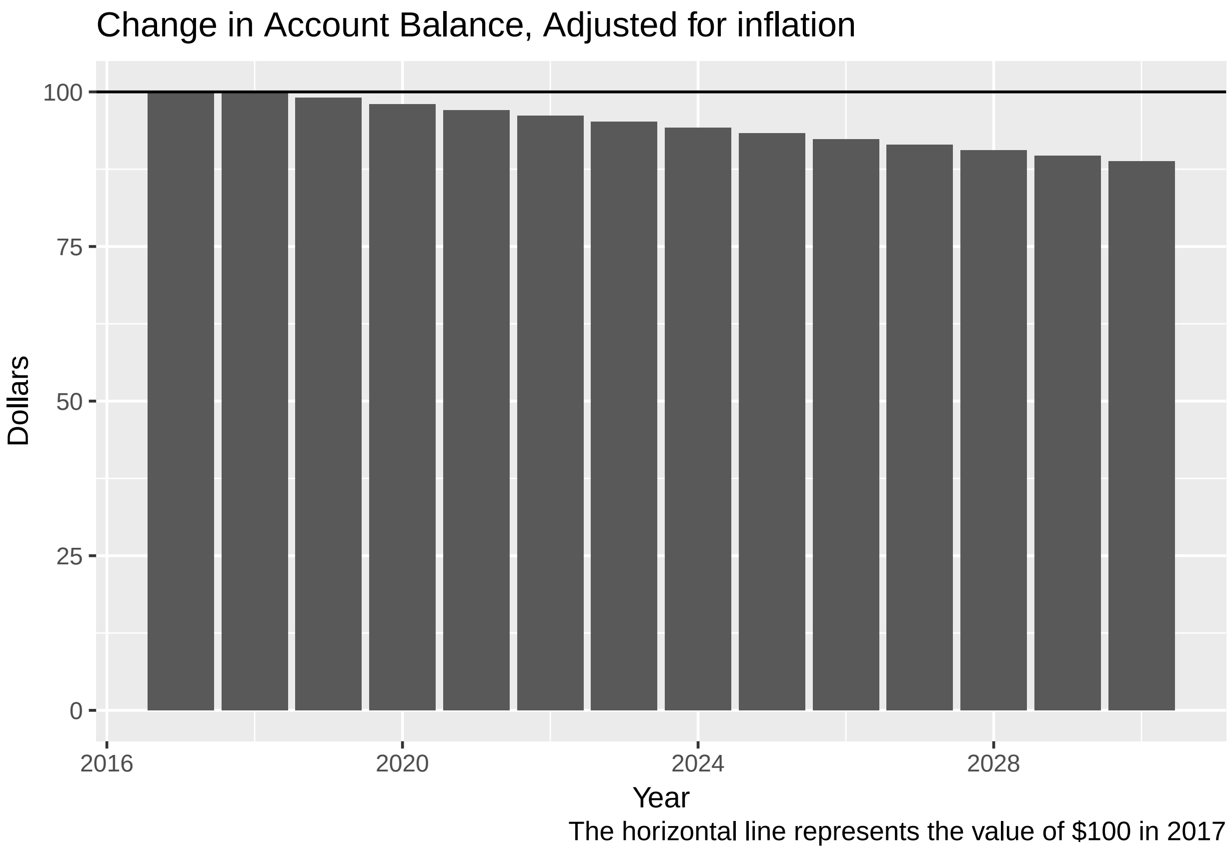
```
## 2  2018       1.000000     -0.9803922        100.0000         100.00000
## 3  2019       1.000000     -0.9803922        101.0000          99.01961
## 4  2020       1.010000     -0.9707805        102.0000          98.03922
## 5  2021       1.020000     -0.9611688        103.0100          97.06844
## 6  2022       1.030100     -0.9516513        104.0300          96.10727
## 7  2023       1.040300     -0.9422281        105.0601          95.15562
## 8  2024       1.050601     -0.9328982        106.1004          94.21339
## 9  2025       1.061004     -0.9236607        107.1510          93.28049
## 10 2026       1.071510     -0.9145146        108.2120          92.35683
## 11 2027       1.082120     -0.9054591        109.2835          91.44231
## 12 2028       1.092835     -0.8964933        110.3656          90.53685
## 13 2029       1.103656     -0.8876162        111.4585          89.64036
## 14 2030       1.114585     -0.8788271        112.5621          88.75274
```

Question 2: Plots

```r
account %>%
    ggplot(mapping = aes(x = year, y = nominal_balance)) +
    geom_bar(stat = "identity") +
    geom_hline(yintercept = 100) +
    labs(title = paste("Change in Nominal Account Balance"),
         caption = "The horizontal line represents a balance of $100",
         x = "Year",
         y = "Dollars")
```



```r
account %>%
    ggplot(mapping = aes(x = year, y = adjusted_balance)) +
    geom_bar(stat = "identity") +
    geom_hline(yintercept = 100) +
    labs(title = paste("Change in Account Balance, Adjusted for inflation"),
         caption = paste("The horizontal line represents the value of $100 in", min(account$year)),
         x = "Year",
         y = "Dollars")
```



Survey Data

Let's look at the data and see how people responded to these questions in the survey.

Create a new section heading

Create a section entitled "Data". Add text to describe where we are getting our data

Data for the 2015 National Financial Capability Study (NFCS) are available for download from the foundation's website. To keep track of what we are doing we will use R to download the 2015 State-by-State Respondent-Level Data.

Create directories

Store raw data separately. To maintain reproducibility, do not alter the raw input data.

```r
# location for raw source data downloaded from the internet
raw_dir <- file.path("data", "raw")

if ( !dir.exists(raw_dir) ) {
    dir.create(raw_dir)
}
```

Download the raw data, unzip

```r
# Source data
link <- "http://www.usfinancialcapability.org/downloads/NFCS_2015_State_by_State_Data_Excel.zip"

# Path for a local copy of the file
save_as <- file.path(raw_dir, basename(link))


# Conditionally downlad and unzip the file
if ( !file.exists(save_as) ) {

    # Download the raw data file if we don't already have it
    download.file(link, save_as)
    unzip(save_as, exdir = raw_dir)

} else if ( !file.exists(file.path(raw_dir, "NFCS 2015 State Data 160619.csv")) ) {

    # Extract all raw data files from an existing zip file
    unzip(zip_file, exdir = raw_dir)

} else {

    list.files(raw_dir)

}
```

Import the raw data

```r
raw_data <- read.csv(file = file.path(raw_dir, "NFCS 2015 State Data 160619.csv"))
```

Clean the data, filter an reformat variables

The raw data set has 126 variables and we aren't going to use every variable, so we will restrict the data to a few columns and use more descriptive names. The columns we plan to keep are:

```r
raw_variables <- c(
    "NFCSID" = "Respondent ID",
    "STATEQ" = "State ID",
    "A3"     = "Gender",
    "A3Ar_w" = "Age Group",
    "M6"     = "Question 1: Interest rate",
    "M7"     = "Question 2: Interest rates and inflation",
```

```
    "M8"     = "Question 3: Interest rates and bond prices",
    "M31"    = "Question 4: Compounding interest",
    "M9"     = "Question 5: Compounding and loan term",
    "M10"    = "Question 6: Diversification",
    "wgt_n2" = "National-level weight",
    "wgt_s3" = "State-level weight"
    )

data.frame(variable = names(raw_variables),
           description = unname(raw_variables)) %>%
    knitr::kable()
```

variable	description
NFCSID	Respondent ID
STATEQ	State ID
A3	Gender
A3Ar_w	Age Group
M6	Question 1: Interest rate
M7	Question 2: Interest rates and inflation
M8	Question 3: Interest rates and bond prices
M31	Question 4: Compounding interest
M9	Question 5: Compounding and loan term
M10	Question 6: Diversification
wgt_n2	National-level weight
wgt_s3	State-level weight

Also, we will need to tidy the data by applying labels to the coded values.

```
clean_data <- raw_data %>%
    #select_(.dots = names(raw_variables)) %>% # using a character vector to select columns
    mutate(id = NFCSID,
           state = stateq_label[STATEQ],
           #question_1 = factor(M6, levels = names(m6_label), labels = unname(m6_label)),
           question_1 = m6_label[M6],
           question_2 = m7_label[M7],
           question_1_correct = ifelse(question_1 == "More than $102", TRUE, FALSE),
           question_2_correct = ifelse(question_2 == "Less than today", TRUE, FALSE),
           state_weight = wgt_s3,
           national_weight = wgt_n2) %>%
    select(id, state, question_1, question_2, question_1_correct, question_2_correct, state_weight, nat

str(clean_data)
```

```
## 'data.frame':    27564 obs. of  8 variables:
##  $ id                : int  2015010001 2015010002 2015010003 2015010004 2015010005 2015010006 201501
##  $ state             : Named chr  "Arizona" "Ohio" "New York" "Florida" ...
##   ..- attr(*, "names")= chr  "3" "36" "33" "10" ...
##  $ question_1        : Named chr  "Less than $102" "More than $102" "More than $102" "Less than $102
##   ..- attr(*, "names")= chr  "3" "1" "1" "3" ...
##  $ question_2        : Named chr  "Less than today" "Less than today" NA NA ...
##   ..- attr(*, "names")= chr  "3" "3" NA NA ...
##  $ question_1_correct: Named logi  FALSE TRUE TRUE FALSE TRUE TRUE ...
##   ..- attr(*, "names")= chr  "3" "1" "1" "3" ...
```

```
##  $ question_2_correct: Named logi  TRUE TRUE NA NA NA TRUE ...
##   ..- attr(*, "names")= chr  "3" "3" NA NA ...
##  $ state_weight      : num  1.288 1.022 0.823 1.02 1.145 ...
##  $ national_weight   : num  0.537 1.664 0.709 2.032 2.521 ...
```

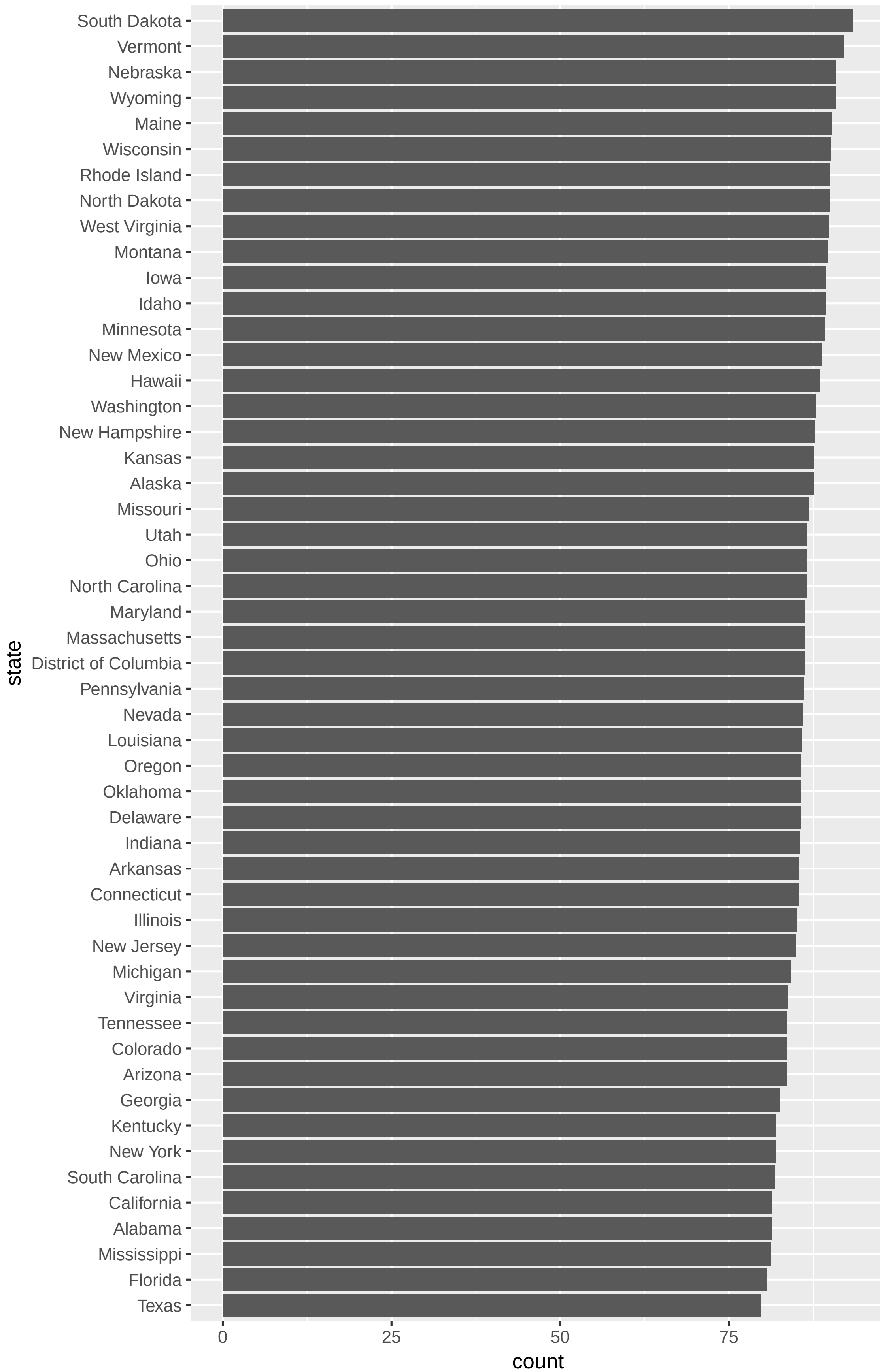
Explore data

```
clean_data %>%
    filter(!is.na(question_1)) %>%
    group_by(state) %>%
    summarize(q1_correct_pct = 100 * sum(state_weight * question_1_correct) / sum(state_weight)) %>%
    arrange(desc(q1_correct_pct)) %>%
    knitr::kable()
```

state	q1_correct_pct
South Dakota	93.42885
Vermont	92.03837
Nebraska	90.87675
Wyoming	90.82055
Maine	90.26080
Wisconsin	90.12008
Rhode Island	89.99030
North Dakota	89.92055
West Virginia	89.85283
Montana	89.71388
Iowa	89.42744
Idaho	89.34921
Minnesota	89.29376
New Mexico	88.82149
Hawaii	88.43982
Washington	87.92472
New Hampshire	87.79586
Kansas	87.65354
Alaska	87.61183
Missouri	86.90420
Utah	86.60036
Ohio	86.56556
North Carolina	86.52179
Maryland	86.33032
Massachusetts	86.28379
District of Columbia	86.27268
Pennsylvania	86.15649
Nevada	86.04625
Louisiana	85.87886
Oregon	85.70074
Oklahoma	85.63632
Delaware	85.58835
Indiana	85.57814
Arkansas	85.41661
Connecticut	85.36274
Illinois	85.16158
New Jersey	84.88538
Michigan	84.16438
Virginia	83.82402
Tennessee	83.65231

state	q1_correct_pct
Colorado	83.63104
Arizona	83.57275
Georgia	82.64554
Kentucky	81.94844
New York	81.93528
South Carolina	81.83426
California	81.44903
Alabama	81.35037
Mississippi	81.22223
Florida	80.66018
Texas	79.76124

```
clean_data %>%
    filter(!is.na(question_1)) %>%
    group_by(state) %>%
    summarize(q1_correct_pct = 100 * sum(state_weight * question_1_correct) / sum(state_weight)) %>%
    arrange(q1_correct_pct) %>%
    mutate(state = factor(state, state)) %>%
    ggplot(mapping = aes(x = state, weight = q1_correct_pct)) +
    geom_bar(stat = "count") +
    coord_flip()
```

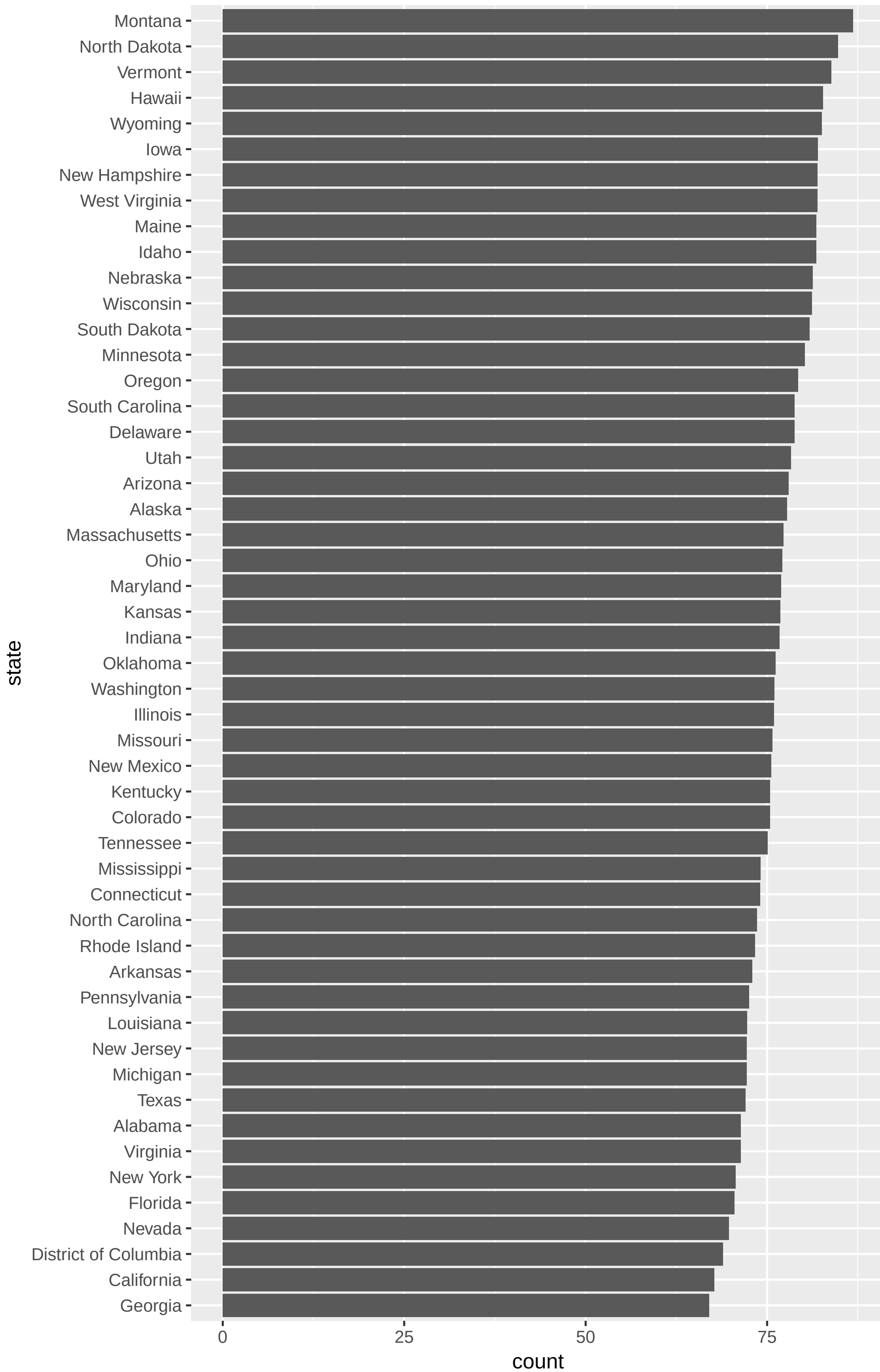

South Dakota
Vermont
Nebraska
Wyoming
Maine
Wisconsin
Rhode Island
North Dakota
West Virginia
Montana
Iowa
Idaho
Minnesota
New Mexico
Hawaii
Washington
New Hampshire
Kansas
Alaska
Missouri
Utah
Ohio
North Carolina
Maryland
Massachusetts
District of Columbia
Pennsylvania
Nevada
Louisiana
Oregon
Oklahoma
Delaware
Indiana
Arkansas
Connecticut
Illinois
New Jersey
Michigan
Virginia
Tennessee
Colorado
Arizona
Georgia
Kentucky
New York
South Carolina
California
Alabama
Mississippi
Florida
Texas
state
0
25
50
75
count

```r
clean_data %>%
    filter(!is.na(question_2)) %>%
    group_by(state) %>%
    summarize(q2_correct_pct = 100 * sum(state_weight * question_2_correct) / sum(state_weight)) %>%
    arrange(desc(q2_correct_pct)) %>%
    knitr::kable()
```

state	q2_correct_pct
Montana	86.87319
North Dakota	84.78064
Vermont	83.84618
Hawaii	82.70407
Wyoming	82.57399
Iowa	82.02434
New Hampshire	81.95886
West Virginia	81.93912
Maine	81.80877
Idaho	81.76657
Nebraska	81.30206
Wisconsin	81.20892
South Dakota	80.85972
Minnesota	80.19868
Oregon	79.29114
South Carolina	78.78244
Delaware	78.77940
Utah	78.31037
Arizona	77.96150
Alaska	77.75200
Massachusetts	77.25472
Ohio	77.08211
Maryland	76.96365
Kansas	76.82806
Indiana	76.71693
Oklahoma	76.17492
Washington	76.00086
Illinois	75.95689
Missouri	75.73211
New Mexico	75.56935
Kentucky	75.40363
Colorado	75.39506
Tennessee	75.07704
Mississippi	74.08778
Connecticut	74.06846
North Carolina	73.64251
Rhode Island	73.33409
Arkansas	72.94043
Pennsylvania	72.54430
Louisiana	72.24531
New Jersey	72.21675
Michigan	72.20661
Texas	72.01013
Alabama	71.40786
Virginia	71.35715

state	q2_correct_pct
New York	70.66398
Florida	70.49277
Nevada	69.76585
District of Columbia	68.95779
California	67.74350
Georgia	67.01169

```
clean_data %>%
    filter(!is.na(question_2)) %>%
    group_by(state) %>%
    summarize(q2_correct_pct = 100 * sum(state_weight * question_2_correct) / sum(state_weight)) %>%
    arrange(q2_correct_pct) %>%
    mutate(state = factor(state, state)) %>%
    ggplot(mapping = aes(x = state, weight = q2_correct_pct)) +
    geom_bar(stat = "count") +
    coord_flip()
```


Montana
North Dakota
Vermont
Hawaii
Wyoming
Iowa
New Hampshire
West Virginia
Maine
Idaho
Nebraska
Wisconsin
South Dakota
Minnesota
Oregon
South Carolina
Delaware
Utah
Arizona
Alaska
Massachusetts
Ohio
Maryland
Kansas
Indiana
Oklahoma
Washington
Illinois
Missouri
New Mexico
Kentucky
Colorado
Tennessee
Mississippi
Connecticut
North Carolina
Rhode Island
Arkansas
Pennsylvania
Louisiana
New Jersey
Michigan
Texas
Alabama
Virginia
New York
Florida
Nevada
District of Columbia
California
Georgia
state
0
25
50
75
count

References

"R for Data Science", http://r4ds.had.co.nz/r-markdown.html

"R Markdown Cheatsheet", http://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf

"LaTeX mathematics", https://en.wikibooks.org/wiki/LaTeX/Mathematics

"Literate Programming", https://en.wikipedia.org/wiki/Literate_programming

"Inflation-Adjusted Returns", Inflation-Adjusted Returns, accessed 2017-10-07, http://www.investopedia.com/terms/i/inflation_adjusted_return.asp

Freakonomics Radio Podcast. "Everything You Always Wanted to Know About Money (But Were Afraid to Ask)". Posted August 2, 2017 at 11:00pm. http://freakonomics.com/podcast/everything-always-wanted-know-money-afraid-ask/

Chicago Manual of Style Online. "Notes and Bibliography: Sample Citations." 2017. http://www.chicagomanualofstyle.org/tools
guide-1.html#cg-website

American Economic Association. "Sample References." 2017. https://www.aeaweb.org/journals/policies/sample-references