Workflow For Efficient Data Analysis in R

Clifton Lee

Last modified: 19 June, 2021

Introduction

It is exceedingly uncommon to get a dataset that does not require any processing before analysis, therefore we must always be prepared to clean it to our specifications.

In this project, I will be working as a data analyst for a firm that offers programming books. The business has published a number of books, all of which have gotten favorable reviews. The firm wants me to look at the sales data and see if there is any valuable information I can get from it. As I go through the analysis, I'll guide you through this procedure. You may get additional information about the dataset by downloading it here.

Business Problem

With an increase in the amount of good reviews for the company's published books. They are unable to make intuitive judgments about the next steps to take, therefore they require a more data-driven approach to determine which book is the most lucrative and popular, as well as any other valuable insights I can glean.

Data Understanding

When we're talking about data analysis in general, it is easy to lose track of the context. Before I begin any analysis, I would first familiarize myself with my dataset. I verify a plethora of things with the data. How much information is there? What type of information do I truly have? Is there anything "strange" that might interfere with any analysis I need to conduct? Is there any missing data? Responding to these questions now will save me time and work later.

If I don't double-check the data first, it's possible for me to make incorrect assumptions about the data, which might stymie my progress later. Perhaps I misread one column as a number, but it was really read in as a string. It's possible that some words were misspelled. In any event, familiarizing myself with the data is the first stage in my data analysis methodology.

```
dataset <- read_csv("book_reviews.csv") # Import the dataset
```

How big is the dataset?

```
dim(dataset)
```

```
## [1] 2000    4
```

As shown above the dataset has 2000 rows and 4 coloumns.

What are the column names?

```
colnames(dataset)
```

```
## [1] "book"   "review" "state"  "price"
```

Each column reflects a feature of the books that have been published.

What are the types of each of the columns?

```
for (name in colnames(dataset)){
  print(paste(name, " column is of type ", typeof(dataset[[name]])))
}
```

```
## [1] "book  column is of type  character"
## [1] "review  column is of type  character"
## [1] "state  column is of type  character"
## [1] "price  column is of type  double"
```

There are several techniques to determine the type of each column, however this would be an excellent application of a for loop, therefore my selection.

What are the unique values in each column?

```
list_of_columns <- list()
for (name in colnames(dataset)){
  list_of_columns[[name]] <- unique(dataset[[name]])
}
list_of_columns
```

```
## $book
## [1] "R Made Easy"
## [2] "R For Dummies"
## [3] "Secrets Of R For Advanced Students"
## [4] "Top 10 Mistakes R Beginners Make"
## [5] "Fundamentals of R For Beginners"
##
## $review
## [1] "Excellent" "Fair"      "Poor"      "Great"     NA          "Good"
##
## $state
## [1] "TX"         "NY"         "FL"         "Texas"      "California"
## [6] "Florida"    "CA"         "New York"
##
## $price
## [1] 19.99 15.99 50.00 29.99 39.99
```

When dealing with numbers, it's useful to get a feel of how high and low the values may go. When dealing with strings, it's useful to examine all of the many potential values.

Data Preprocessing

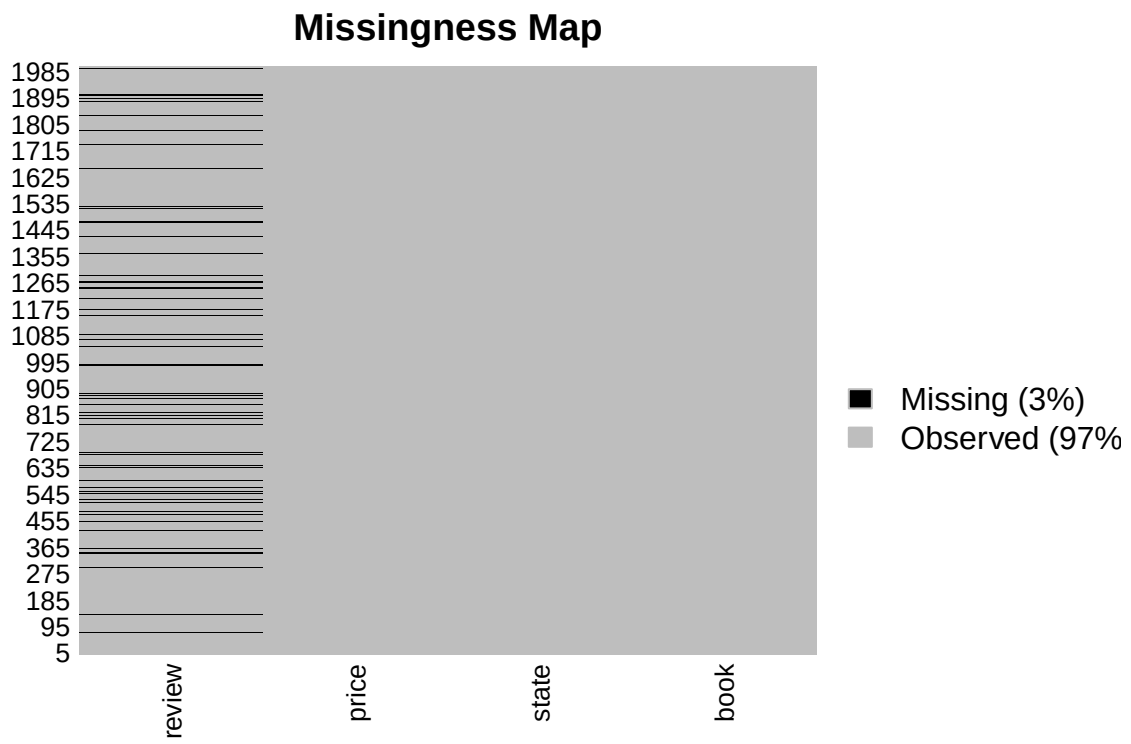
Now that I am more comfortable with the data, I can go into the finer points of data analysis. A major portion of my job is to convert a raw dataset into a format that can be analyzed. Many times, I will be unable to just grab a dataset and begin studying it. It's excellent practice to check over the data ahead of time and make a note of any modifications I'll need to make for it.

Missing Data

The first issue I'll have to deal with is the issue of missing data. I can deal with missing data in two ways: 1) Remove any rows or columns with missing data (usually, rows) or 2) Fill in the missing data in an educated, disciplined manner. The second method is called imputation. For the time being, I will take the first method with this dataset.

A `missing plot` is my preferred way for quickly determining the quantity of missing data in a dataset.

```
missmap(dataset, col = c("black", "grey"))
```



The x-axis shows attributes and the y-axis shows instances. Horizontal lines indicate missing data for an instance, vertical blocks represent missing data for an attribute.

Some instances contain missing data in the `review` field, as I can see. I will now solve this issue by making a duplicate of the dataset and removing all rows with missing data.

```
new_dataset <- dataset %>%
               filter(!is.na(review))

dim(new_dataset)
```

```
## [1] 1794    4
```

My new dataset now has 1794 rows and 4 columns.

The basic rule of thumb for determining the best solution for this situation is to consider the percentage of data that is missing. In general, if the missing data points account for fewer than 5% of our entire dataset, we can consider deleting rows. If it is greater than 5%, then the results of any calculation may be influenced.

Given that the missing data points account for 3% of the total, I am comfortable with deleting the rows with missing data.

Noise Handling

I now have a complete dataset after removing all of the missing data from it. This is the perfect situation in which I would like to begin any data analysis, therefore I'm already working on a better dataset.

The next thing I need to focus on is dealing with noise in the data, namely in the `state` column. You might have noticed that the labeling for each state varies. California, for example, is written as both "California" and "CA." Both "California" and "CA" relate to the same location in the United States, so I'll try to straighten

this out. I must select one of the conventions for referring to the state and adhere to it. Making labels/strings more consistent in the data will make things easier to analyze later, so I'll take care of this now.

What are all the states that are present in the dataset?

```
list_of_columns$state
```

```
## [1] "TX"         "NY"         "FL"         "Texas"      "California"
## [6] "Florida"    "CA"         "New York"
```

Given that not all of my readers will be familiar with the abbreviations for the various states, I will make my dataset more visible by using the full names of the states.

```
new_dataset <- new_dataset %>%
               mutate(state_name = case_when(state == "TX" ~ "Texas",
                                             state == "NY" ~ "New York",
                                             state == "FL" ~ "Florida",
                                             state == "CA" ~ "California",
                                             TRUE ~ state)) %>%
               select(-state)

head(new_dataset, 10) # Examine the modifications to confirm they were made appropriately.
```

```
## # A tibble: 10 x 4
##    book                                review    price state_name
##    <chr>                               <chr>     <dbl> <chr>
##  1 R Made Easy                         Excellent  20.0 Texas
##  2 R For Dummies                       Fair       16.0 New York
##  3 R Made Easy                         Excellent  20.0 New York
##  4 R Made Easy                         Poor       20.0 Florida
##  5 Secrets Of R For Advanced Students  Great      50   Texas
##  6 R Made Easy                         Great      20.0 Florida
##  7 R Made Easy                         Poor       20.0 California
##  8 Top 10 Mistakes R Beginners Make    Fair       30.0 California
##  9 Secrets Of R For Advanced Students  Fair       50   Texas
## 10 R Made Easy                         Great      20.0 New York
```

Feature Modification

The next thing I'll do with the dataset is deal with the reviews themselves. In my data exploration, you may have observed that the reviews took the form of strings ranging from "Poor" to "Excellent." My objective is to assess the ratings of each textbook, but I can't do much with text versions of the review scores. It would be preferable if I converted the reviews into numbers.

```
new_dataset <- new_dataset %>%
               mutate(review_num = case_when(review == "Poor" ~ 1,
                                             review == "Fair" ~ 2,
                                             review == "Good" ~ 3,
                                             review == "Great" ~ 4,
                                             review == "Excellent" ~ 5),
                      is_high_review = if_else(review_num >= 4, TRUE, FALSE)) %>%
              select(-review)

new_dataset
```

```
## # A tibble: 1,794 x 5
##    book                          price state_name review_num is_high_review
##    <chr>                         <dbl> <chr>           <dbl> <lgl>
```

```
##  1 R Made Easy                    20.0 Texas              5 TRUE
##  2 R For Dummies                  16.0 New York           2 FALSE
##  3 R Made Easy                    20.0 New York           5 TRUE
##  4 R Made Easy                    20.0 Florida            1 FALSE
##  5 Secrets Of R For Advanced St~  50   Texas              4 TRUE
##  6 R Made Easy                    20.0 Florida            4 TRUE
##  7 R Made Easy                    20.0 California         1 FALSE
##  8 Top 10 Mistakes R Beginners ~  30.0 California         2 FALSE
##  9 Secrets Of R For Advanced St~  50   Texas              2 FALSE
## 10 R Made Easy                    20.0 New York           4 TRUE
## # ... with 1,784 more rows
```

Another column that helps me judge if a score is "high" or not would be useful. As a result, I added a new column that shows TRUE if the observed row has a review number of 4 or above.

Data Analysis

It's critical that I keep the main aim in mind as I handle all of the cleaning specifics. I'm playing the role of an analyst, attempting to determine which publications are the most profitable for the firm. Because the initial data was not in a usable format for analysis, I needed to clean it up.

Now that I've completed all of my data cleaning, I'm ready to conduct some data analysis. My primary objective is to determine which book is the most profitable. But how will I choose which book is the "most profitable"? My dataset is made up of consumer purchases. One approach to determine "most lucrative" is to simply select the book that has been purchased the most times. Another approach to describe it is to look at how much money each book produces in total.

```
most_profitable <- new_dataset %>%
                   group_by(book) %>%
                   summarize(amount_sold = n(),
                             total_sales_value = sum(price),
                             typical_rating = median(review_num)) %>%
                   arrange(desc(total_sales_value))

most_profitable
```

```
## # A tibble: 5 x 4
##   book                           amount_sold total_sales_val~ typical_rating
##   <chr>                                <int>            <dbl>          <dbl>
## 1 Secrets Of R For Advanced St~          360           18000              3
## 2 Fundamentals of R For Beginn~          366           14636.             3
## 3 Top 10 Mistakes R Beginners ~          355           10646.             3
## 4 R Made Easy                            352            7036.             3
## 5 R For Dummies                          361            5772.             3
```

In this situation, I'll use the latter metric to calculate profitablity. As seen in the code cell above with 360 books sold, **Secrets Of R For Advanced Students** is the most lucrative, with a total sales value of $18000.

Conclusion

To summarize, throughout my data analysis and exploration, I discovered **Secrets Of R For Advanced Students** to be the most lucrative book. Despite not being the most popular, it is in the top three (3) most popular, with Fundamentals of R For Beginners being the most popular. Secrets Of R For Advanced Students had a typical rating of 3, which was the same as the other books. As a result, I would advise raising the Secrets Of R For Advanced Students inventory budget to assure a profit.