Financial Analyst

Robert Manni

8/24/2021

# Financial Data Review using TidyVerse

These data are from the second course on Udemy tuaght by Kiril Eremenko on R programming with the focus on data science methods. This document will demonstrate how I used dplyr to accomplish the original content but with cleaner code and methodology to explore the dataset and provide a final report that answers the following:

### 1. Finding missing or NA values in the data.

### 2. subsetting missing values using methods described later in this document.

### 3. Analysis and graphical representation.

# Loading The Tidyverse:

# install.packages("tidyverse") # Run if not already installed.  
  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.2 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(skimr)

# Load the Data:

wd = getwd()  
setwd(wd)

data = read.csv(choose.files(), na.strings = c("")) # Choose file and makes missing values NA's so they're easy to work with.  
  
# Inspect the data  
  
# head(data) # Checks top few rows  
# tail(data) # Checks the bottom few rows  
# str(data) # checks the class of each variable (column)  
  
#Or use skimr:  
skim(data)

Data summary

|  |  |
| --- | --- |
| Name | data |
| Number of rows | 500 |
| Number of columns | 11 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 7 |
| numeric | 4 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| Name | 0 | 1.00 | 5 | 20 | 0 | 500 | 0 |
| Industry | 2 | 1.00 | 6 | 19 | 0 | 7 | 0 |
| State | 4 | 0.99 | 2 | 2 | 0 | 42 | 0 |
| City | 0 | 1.00 | 4 | 18 | 0 | 297 | 0 |
| Revenue | 2 | 1.00 | 10 | 11 | 0 | 498 | 0 |
| Expenses | 3 | 0.99 | 14 | 17 | 0 | 497 | 0 |
| Growth | 1 | 1.00 | 2 | 3 | 0 | 32 | 0 |

**Variable type: numeric**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| ID | 0 | 1 | 250.50 | 144.48 | 1 | 125.75 | 250.5 | 375.25 | 500 | ▇▇▇▇▇ |
| Inception | 1 | 1 | 2010.17 | 3.23 | 1999 | 2009.00 | 2011.0 | 2012.00 | 2014 | ▁▁▁▇▆ |
| Employees | 2 | 1 | 148.61 | 397.35 | 1 | 27.25 | 56.0 | 126.00 | 7125 | ▇▁▁▁▁ |
| Profit | 2 | 1 | 6539474.01 | 3869933.65 | 12434 | 3272073.75 | 6513366.0 | 9303950.75 | 19624534 | ▇▇▇▂▁ |

Inspecting the data allows us to immediately see the format of the data. - Is it in Tidy format? - Do you see missing observations?

## Important Findings: 1. 1 observation has a missing Inception date 2. 2 observations have missing Employees data 3. 2 observations have missing Revenue, Expenses, and Profit data 4. 1 observation has missing Growth.

We will need to address each of these parts will also need to employ the right action for dealing with missing data. It is important to mention that some of these might require consulting a supervisor or the owner of the dataset before performing some of the actions.

Using the str() function we also can see that we have other things we’ll likely need to work with. - Inception is an integer, but we won’t be doing any mathematical operations with the year. - Notice revenue, expenses and growth are all characters. This means we’ll need to convert these in order to use mathematical operations with these columns.

Let’s deal with all those using dplyr. Below I have also changed the missing values. Using dplyr I tested each line separately with the data. dplyr works really fast so I can play with it and find out what works and what doesn’t.

# Finding Missing Values:

I type the code below after trying each line of code above to determine what still needs to be fixed. If it fixes it I keep and move onto the next line.

data %>% # not assigning the data yet. You can play around and see what you like before you assign a vector.   
 filter(!complete.cases(data)) # Returns all rows with incomplete data fields.

## ID Name Industry Inception Employees State  
## 1 3 Greenfax Retail 2012 NA SC  
## 2 8 Rednimdox Construction 2013 73 NY  
## 3 11 Canecorporation Health 2012 6 <NA>  
## 4 14 Techline <NA> 2006 65 CA  
## 5 15 Cityace <NA> 2010 25 CO  
## 6 17 Ganzlax IT Services 2011 75 NJ  
## 7 22 Lathotline Health NA 103 VA  
## 8 44 Ganzgreen Construction 2010 224 TN  
## 9 84 Drilldrill Software 2010 30 <NA>  
## 10 267 Circlechop Software 2010 14 <NA>  
## 11 332 Westminster Financial Services 2010 NA MI  
## 12 379 Stovepuck Retail 2013 73 <NA>  
## City Revenue Expenses Profit Growth  
## 1 Greenville $9,746,272 1,044,375 Dollars 8701897 16%  
## 2 Woodside <NA> <NA> NA <NA>  
## 3 New York $10,597,009 7,591,189 Dollars 3005820 7%  
## 4 San Ramon $13,898,119 5,470,303 Dollars 8427816 23%  
## 5 Louisville $9,254,614 6,249,498 Dollars 3005116 6%  
## 6 Iselin $14,001,180 <NA> 11901180 18%  
## 7 McLean $9,418,303 7,567,233 Dollars 1851070 2%  
## 8 Franklin <NA> <NA> NA 9%  
## 9 San Francisco $7,800,620 2,785,799 Dollars 5014821 17%  
## 10 San Francisco $9,067,070 5,929,828 Dollars 3137242 20%  
## 11 Troy $11,861,652 5,245,126 Dollars 6616526 15%  
## 12 New York $13,814,975 5,904,502 Dollars 7910473 10%

Now, I’ll work through each NA value and each action is shown:

data2.0 = data %>%   
 mutate(Expenses = gsub("Dollars", "", Expenses), # Removed "Dollars".  
 Expenses = gsub(",", "", Expenses), # Removed commas and the space.  
 Expenses = str\_squish(Expenses),  
 Expenses = as.numeric(Expenses), # Changed to integer (hopefully. Will need to check later.).  
 Revenue = gsub("\\$", "", Revenue), # used esscape sequence '\\' to remove '$'.   
 Revenue = gsub(",", "", Revenue),# Removed commas and the space.  
 Revenue = as.numeric(Revenue), # Makes Revenue MATHEMATICAL MAN!  
 Profit = as.numeric(Profit), # Makes Profits MATHEMAICAL MAN!  
 Growth = gsub("%", "", Growth), # Removed % symbol from values.   
 Growth = as.numeric(Growth)) %>% # Makes Growth MATHEMATICAL MAN!  
 mutate(State = ifelse(City == 'San Francisco', 'CA', State), # NA value in State is CA  
 State = ifelse(City == 'New York', 'NY', State), # NA value in State is NY  
 Industry = ifelse(Name == 'Techline', 'IT Services', Industry), #NA value in Industry is IT Services  
 Industry = ifelse(Name == 'Cityace', 'Retail', Industry), #NA value in Industry is Retail  
 Expenses = ifelse(is.na(Expenses), Revenue-Profit, Expenses), # Calculated what Expenses are from existing columns.   
 Inception = ifelse(Name == 'Lathotline', '2011', Inception)) %>% # Made the inception date 2011 for the NA value.  
 drop\_na() #drops NA's for the two construction columns that I cannot infer data for.  
   
   
  
# # Recheck the data:  
# head(data2.0, 25)  
# tail(data2.0, 25)  
# str(data2.0)  
  
# Or use skimr  
skim(data2.0)

Data summary

|  |  |
| --- | --- |
| Name | data2.0 |
| Number of rows | 496 |
| Number of columns | 11 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 5 |
| numeric | 6 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| Name | 0 | 1 | 5 | 20 | 0 | 496 | 0 |
| Industry | 0 | 1 | 6 | 19 | 0 | 7 | 0 |
| Inception | 0 | 1 | 4 | 4 | 0 | 16 | 0 |
| State | 0 | 1 | 2 | 2 | 0 | 42 | 0 |
| City | 0 | 1 | 4 | 18 | 0 | 295 | 0 |

**Variable type: numeric**

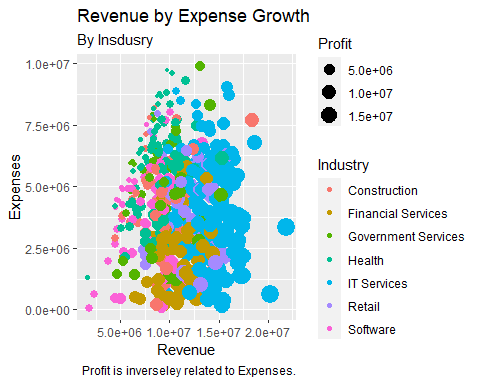
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| ID | 0 | 1 | 251.74 | 143.88 | 1 | 127.75 | 251.5 | 376.25 | 500 | ▇▇▇▇▇ |
| Employees | 0 | 1 | 148.61 | 398.13 | 1 | 27.00 | 55.5 | 126.00 | 7125 | ▇▁▁▁▁ |
| Revenue | 0 | 1 | 10845336.56 | 3199361.46 | 1614585 | 8684289.00 | 10647231.0 | 13112126.75 | 21810051 | ▁▇▇▅▁ |
| Expenses | 0 | 1 | 4310377.61 | 2114916.16 | 71219 | 2758422.75 | 4341072.0 | 5833223.00 | 9860686 | ▅▇▇▅▁ |
| Profit | 0 | 1 | 6534958.94 | 3876521.39 | 12434 | 3259191.25 | 6503795.0 | 9331246.25 | 19624534 | ▇▇▇▂▁ |
| Growth | 0 | 1 | 14.38 | 6.91 | -3 | 8.00 | 15.5 | 20.00 | 30 | ▁▆▅▇▂ |

# Visualization:

The the dataset is not ready to be used for visual analysis. We have been asked to make three graphs:

##1. A scatter plot by industry showing revenue, expenses and profit.

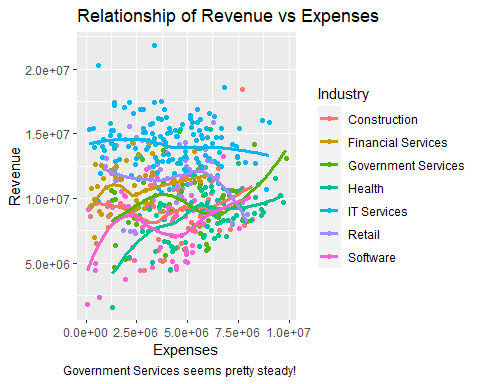
plot = ggplot(data = data2.0) # Assign data to plot to a veriable. I just like that.   
  
plot + geom\_point(aes(x = Revenue, y = Expenses,  
 size = Profit,  
 color = Industry)) +  
 labs(title = "Revenue by Expense Growth",  
 subtitle = "By Insdusry",  
 caption = "Profit is inverseley related to Expenses.")



##2. A scatter plot by industry trends for expenses ~ revenue relationship.

plot + geom\_point(aes(x = Expenses, y = Revenue,  
 color = Industry)) +  
 geom\_smooth(aes(x = Expenses, y = Revenue,  
 color = Industry),  
 fill = NA, size = 1.2) +   
 labs(title = "Relationship of Revenue vs Expenses",  
 caption = "Government Services seems pretty steady!")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

 Although I could have saved some lines here by making the aes() saved into the plot, I chose not to for the sake of being able to clearly see what I am graphing. So, it is redundant for learning purposes.

##3. A box plot showing growht by industry.

plot + geom\_boxplot(aes(x = Industry, y = Growth,   
 color = Industry))

