Live Case: S&P500 (3 of 3)

July 30, 2023 -=- This chapter is being heavily edited; It is very much Work in Progress

S&P 500.

We will continue our analysis of the S&P 500,

S&P 500 Data - Preliminary Analysis

Recall that we are analyzing a real-world, recent dataset containing information about the S&P500 stocks. The dataset is located in a Google Sheet

1. The complete URL is https://docs.google.com/spreadsheets/d/11ahk9uWxBkDqrhNm7qYmiTwrlSC53N1zvXYfv7ttOCM/
2. The Google Sheet ID is: `11ahk9uWxBkDqrhNm7qYmiTwrlSC53N1zvXYfv7ttOCM`.

```r
# Read S&P500 stock data present in a Google Sheet.
library(gsheet)
prefix <- "https://docs.google.com/spreadsheets/d/"
sheetID <- "11ahk9uWxBkDqrhNm7qYmiTwrlSC53N1zvXYfv7ttOCM"
url500 <- paste(prefix,sheetID) # Form the URL to connect to
sp500 <- gsheet2tbl(url500) # Read it into a tibble called sp500
```

```
No encoding supplied: defaulting to UTF-8.
```

2. We will rename the data columns to make it easier to work with the data, using the `rename_with()` function.

```r
# Define a mapping of new column names
new_names <- c(
  "Date", "Stock", "StockName", "Sector", "Industry",
  "MarketCap", "Price", "Low52Wk", "High52Wk",
  "ROE", "ROA", "ROIC", "GrossMargin",
  "OperatingMargin", "NetMargin", "PE",
  "PB", "EVEBITDA", "EBITDA", "EPS",
  "EBITDA_YOY", "EBITDA_QYOY", "EPS_YOY",
  "EPS_QYOY", "PFCF", "FCF",
  "FCF_QYOY", "DebtToEquity", "CurrentRatio",
  "QuickRatio", "DividendYield",
  "DividendsPerShare_YOY", "PS",
  "Revenue_YOY", "Revenue_QYOY", "Rating"
)
# Rename the columns using the new_names vector
sp500 <- sp500 %>%
  rename_with(~ new_names, everything())
```

Review the data again after renaming columns

1. We review the column names again after renaming them, using the `colnames()` function can help.

```
colnames(sp500)
```

```
 [1] "Date"                  "Stock"                 "StockName"
 [4] "Sector"                "Industry"              "MarketCap"
 [7] "Price"                 "Low52Wk"               "High52Wk"
[10] "ROE"                   "ROA"                   "ROIC"
[13] "GrossMargin"           "OperatingMargin"       "NetMargin"
[16] "PE"                    "PB"                    "EVEBITDA"
[19] "EBITDA"                "EPS"                   "EBITDA_YOY"
[22] "EBITDA_QYOY"           "EPS_YOY"               "EPS_QYOY"
[25] "PFCF"                  "FCF"                   "FCF_QYOY"
[28] "DebtToEquity"          "CurrentRatio"          "QuickRatio"
[31] "DividendYield"         "DividendsPerShare_YOY" "PS"
[34] "Revenue_YOY"           "Revenue_QYOY"          "Rating"
```

Understand the Data Columns

1. The complete data has 36 columns. Our goal is to gain a deeper understanding of what the data columns mean.

2. We reorganize the column names into eight tables, labeled Table 1a, 1b.. 1h.

Remove Rows containing no data or Null values

1. The following code checks if the “Stock” column in the sp500 dataframe contains any null or blank values. If there are null or blank values present, it removes the corresponding rows from the sp500 dataframe, resulting in a filtered dataframe without null or blank values in the “Stock” column.

```
# Check for blank or null values in the "Stock" column
hasNull <- any(sp500$Stock == "" | is.null(sp500$Stock))
if (hasNull) {
    # Remove rows with null or blank values from the dataframe tibble
    sp500 <- sp500[!(is.null(sp500$Stock) | sp500$Stock == ""), ]
}
```

```
# View the filtered dataframe
nrow(sp500)
```

```
[1] 503
```

Thus, we have nrow(sp500) stocks of the S&P500 in our dataset.

5. The S&P500 shares are divided into multiple Sectors. Each stock belongs to a unique sector. Thus, it makes sense to model Sector as a factor() variable.

```
sp500$Sector <- as.factor(sp500$Sector)
```

6. We can use the levels() function to review the different levels it can take.

```
levels(sp500$Sector)
```

```
 [1] "Commercial Services"    "Communications"          "Consumer Durables"
 [4] "Consumer Non-Durables"  "Consumer Services"       "Distribution Services"
 [7] "Electronic Technology"  "Energy Minerals"         "Finance"
[10] "Health Services"        "Health Technology"       "Industrial Services"
[13] "Non-Energy Minerals"    "Process Industries"      "Producer Manufacturing"
[16] "Retail Trade"           "Technology Services"     "Transportation"
[19] "Utilities"
```

The table() function allows us to count how many stocks are part of each sector.

```r
table(sp500$Sector)
```

```
   Commercial Services         Communications      Consumer Durables 
                    13                      3                     12 
 Consumer Non-Durables      Consumer Services  Distribution Services 
                    31                     29                      9 
 Electronic Technology        Energy Minerals                Finance 
                    49                     16                     92 
       Health Services      Health Technology    Industrial Services 
                    12                     47                      9 
   Non-Energy Minerals     Process Industries Producer Manufacturing 
                     7                     24                     31 
          Retail Trade    Technology Services         Transportation 
                    23                     50                     15 
             Utilities 
                    31 
```

Thus, we can see how many stocks are part of each one of the 19 sectors.

We can sum them to confirm that they add up to 502.

```r
sum(table(sp500$Sector))
```

```
[1] 503
```

7. Stock Ratings: In the data, the S&P500 shares have Technical Ratings such as {Buy, Sell, ..}. Since each Stock has a unique Technical Rating, it makes sense to model the data column Rating as a factor() variable.

```r
sp500$Rating <- as.factor(sp500$Rating)
```

We can use the levels() function to review the different levels it can take.

```r
levels(sp500$Rating)
```

```
[1] "Buy"         "Neutral"     "Sell"        "Strong Buy"  "Strong Sell"
```

The table() function allows us to count how many stocks have each Rating.

```
table(sp500$Rating)
```

```

        Buy     Neutral        Sell  Strong Buy Strong Sell 
        130          85         203          13          72 
```

Thus, we can see how many stocks have ratings ranging from “Strong Sell” to “Strong Buy”. This completes our review of Technical Rating.

Filter the data by sector Technology Services, and display the number of stocks in the sector

```
ts <- sp500 %>%
        filter(Sector=='Technology Services')

nrow(ts)
```

```
[1] 50
```

There are 50 number of of stocks in the sector Technology Services

Select the Specific Coulumns from the filtered dataframe `ts` (Technology Services)

```
ts2 <- ts %>%
        select(Date, Stock, StockName,Sector,  Industry, MarketCap, Price,Low52Wk, High52W
               ROE, ROA,ROIC,GrossMargin, GrossMargin,
               NetMargin, Rating)

colnames(ts2)
```

```
 [1] "Date"         "Stock"        "StockName"    "Sector"       "Industry"    
 [6] "MarketCap"    "Price"        "Low52Wk"      "High52Wk"     "ROE"         
[11] "ROA"          "ROIC"         "GrossMargin" "NetMargin"    "Rating"      
```

Arrange the Dataframe by ROE

```
ts3 <- ts2 %>% arrange(desc(ROE))
```

Top 10 Shares in Sector Technology Services Based on ROE

```
head(ts3,10)
```

```
# A tibble: 10 x 15
   Date   Stock StockName Sector Industry MarketCap Price Low52Wk High52Wk   ROE
   <chr>  <chr> <chr>     <fct>  <chr>        <dbl> <dbl>   <dbl>    <dbl> <dbl>
 1 8/10/~ FTNT  Fortinet~ Techn~ Informa~   5.85e10  74.5    42.6     81.2 844
 2 8/10/~ IT    Gartner,~ Techn~ Interne~   2.67e10 339.    273.      378    417
 3 8/10/~ GEN   Gen Digi~ Techn~ Package~   1.19e10  18.6    15.4     26.7 128
 4 8/10/~ ADSK  Autodesk~ Techn~ Package~   4.36e10 204.    180.      235    106
 5 8/10/~ ADP   Automati~ Techn~ Data Pr~   1.02e11 247.    201.      275    101
 6 8/10/~ CDW   CDW Corp~ Techn~ Informa~   2.67e10 198.    148.      215     79.3
 7 8/10/~ PAYX  Paychex,~ Techn~ Data Pr~   4.48e10 124.    104.      139     47.3
 8 8/10/~ MSFT  Microsof~ Techn~ Package~   2.43e12 327.    213.      367     38.8
 9 8/10/~ FDS   FactSet ~ Techn~ Data Pr~   1.65e10 434.    378.      474     34.6
10 8/10/~ ADBE  Adobe In~ Techn~ Package~   2.39e11 524.    275.      553     33.7
# i 5 more variables: ROA <dbl>, ROIC <dbl>, GrossMargin <dbl>,
#   NetMargin <dbl>, Rating <fct>
```

Mutate a data column called (Low52WkPerc), then show top 10 ROE stocks

```
ts4 <- ts3 %>% mutate(Low52WkPerc = round((Price - Low52Wk)*100 / Low52Wk,2))
head(ts4[,c(1:3,10,16)],10)
```

```
# A tibble: 10 x 5
   Date      Stock StockName                         ROE Low52WkPerc
   <chr>     <chr> <chr>                           <dbl>       <dbl>
 1 8/10/2023 FTNT  Fortinet, Inc.                  844          74.9
 2 8/10/2023 IT    Gartner, Inc.                   417          24.3
 3 8/10/2023 GEN   Gen Digital Inc.                128          20.5
 4 8/10/2023 ADSK  Autodesk, Inc.                  106          13.6
 5 8/10/2023 ADP   Automatic Data Processing, Inc. 101          22.7
 6 8/10/2023 CDW   CDW Corporation                   79.3         33.7
 7 8/10/2023 PAYX  Paychex, Inc.                     47.3         19.3
 8 8/10/2023 MSFT  Microsoft Corporation             38.8         53.2
 9 8/10/2023 FDS   FactSet Research Systems Inc.    34.6         14.7
10 8/10/2023 ADBE  Adobe Inc.                        33.7         90.9
```