Assignment 2: Grammar of Graphics

Szymon Bujowski - 148050, Preetam Sharma - 150504

Contents

Foreword

The following document is an assignment for Poznan University of Technology's Data Visualization course. The course is conducted by Dariusz Brzeziński during the 4th semester of Artificial Intelligence Bachelor degree.

The assignment is an implementation of the grammar of graphics, intended to create rich visualizations from the data we were provided with. The data consists of two data sets, for both of which we've chosen the upcoming visualizations. As it is stated in the assignment description:

- The data in the Sectors folder present the percentage changes of stock prices and trading volume in selected sectors
  - Some of the data sets also contain information about the media sentiment about the companies
- The Correlations.csv data set contains correlations between the stock prices of pairs of companies identified by stock symbols (tickers)

For both the following visualizations, we will provide brief descriptions and reasoning behind them.

On top of that, we add an anticlimactic digression regarding one of the ideas for Sectors data visualization.

Credits

We have to credit the lecturer, **Dariusz Brzeziński**, for the interactive tables we have used. They were built using the `DT` package.

The interactive tables should work as intended in html format of the document, however they will not be visible in pdf format. For that reason, a standard head of the data frames are displayed.

TODO insert screenshot of interactive table

Resources

You can access the project's repository on GitHub - https://github.com/bujowskis/put-DV/tree/main/ass-2

Stocks by sectors

The data

For simplicity reasons (and keeping this document relatively short), we are going to show two out of 8 data sets. One of them will be a representative of the sets with sentiments included, and the other one with sentiments missing.

Sentiment included

```
##   X Symbol                         Name   Volume   Open   High  Close Sentiment
## 1 0   AAPL                   Apple Inc. 86580100 172.89 174.14 172.17      0.31
## 2 1   ADBE                   Adobe Inc.  3605200 513.66 520.42 510.70      0.22
## 3 3   AEHR            Aehr Test Systems  4290800  20.09  20.09  16.09      0.26
## 4 5     AI                  C3.ai, Inc.  2299200  29.71  30.97  29.94      0.51
## 5 7   AMAT      Applied Materials, Inc.  6334000 155.10 157.38 150.81      0.40
## 6 8    AMD Advanced Micro Devices, Inc. 58398000 136.28 137.44 132.00      0.26
```

Sentiment missing

```
##    X Symbol                              Name    Volume X1dC.  X1dV.   Open
## 1  6    CEI              Camber Energy, Inc. 639858500  3.12 -12.73   0.90
## 2 32    OXY Occidental Petroleum Corporation  81250100  1.41   0.86  57.71
## 3 51    XOM          Exxon Mobil Corporation  55140300 -0.76 -29.36  84.92
## 4 42    SWN      Southwestern Energy Company  37153500 -1.08 -55.73   5.51
## 5 28    MRO         Marathon Oil Corporation  35893000  0.99 -16.29  24.00
## 6 12    CVX              Chevron Corporation  34810600 -5.24 -63.04 159.90
##     High  Close Sentiment
## 1   1.47   1.28        NA
## 2  58.77  55.38        NA
## 3  87.23  87.12        NA
## 4   5.84   5.54        NA
## 5  25.39  24.33        NA
## 6 162.10 162.04        NA
```

Sketch

The underlying idea is to show the change in closing price of all the stocks of a particular sector on a single plot. We decided to use a **tree map**, alongside an **interactive table**.

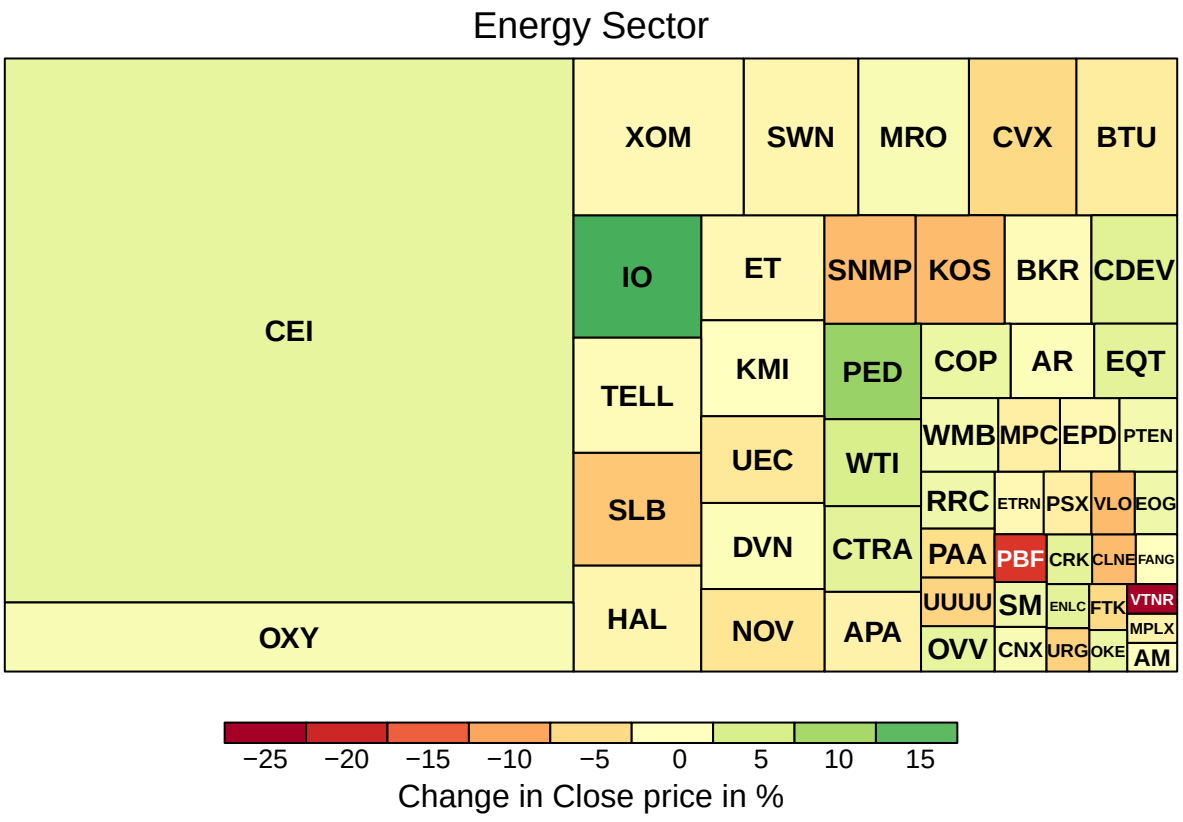
The tree map was chosen due to its effectiveness in capturing the most relevant and interesting stocks. It is very easy to spot some of the good choices right away. There are two different types of tree maps we have chosen. For both of them, **Size** of the tiles shows the **volume**. The difference lies in the **color** - it shows the **sentiment score** if sentiment is not missing, and **change in close price** otherwise.

The biggest drawback of the tree map is the clutter in case of the stocks with relatively low volume. For that reason, the interactive table is used alongside it. It both makes it possible to look up data of the stock with big volume of our interest, as well as sorting the stocks by volume (or any other attribute) to get a better insight into what the tree map does not capture very well.

The visualization

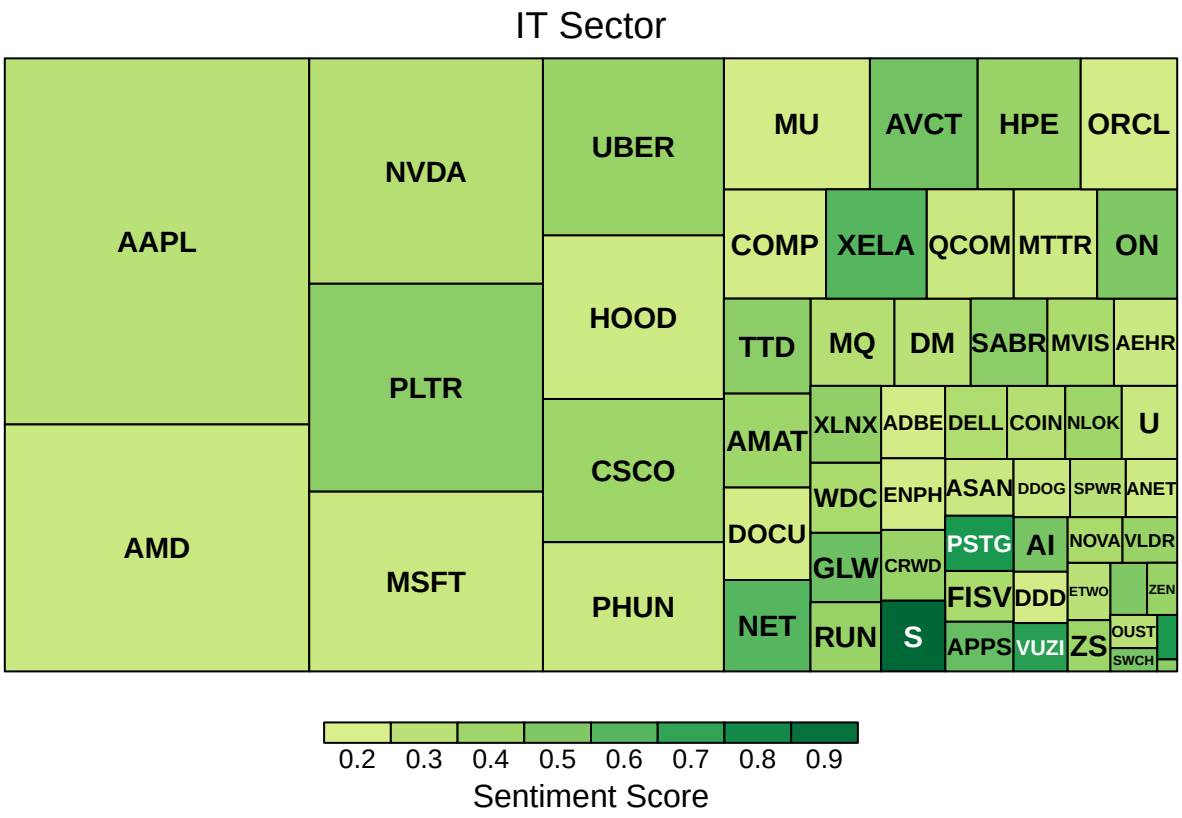
NOTE - the interactive table will not show in the pdf. Please use the html version instead.

```
library(treemap)

# no sentiment
energy = read.csv("Dataset/Sectors/energy.csv")
treemap(energy,index=c("Symbol"),vSize = "Volume", vColor = "X1dC.",type="value",border.col = "black",
        border.lwds = 1,title = "Energy Sector",title.legend = "Change in Close price in %")
```



```
prettyTable(energy)
```

```
# sentiment
# TODO - make sentiment score label 0-1
IT = read.csv("Dataset/Sectors/it.csv")
treemap(IT,index=c("Symbol"),vSize = "Volume", vColor = "Sentiment",type="value",border.col = "black",
        border.lwds = 1,title = "IT Sector",title.legend = "Sentiment Score")
```




```
prettyTable(IT)
```

Correlations

The data

```
##   Ticker.1 Ticker.2 Correlation.Value
## 1       GS      JPM         0.7955952
## 2     AAPL     MSFT         0.7069591
## 3      AXP      JPM         0.6833357
## 4       KO       PG         0.6553540
## 5      CRM     MSFT         0.6464821
## 6      HON      MMM         0.6289362
```

Sketch

Static visualization choice for the correlations was pretty obvious from the beginning - a heat map correlation matrix. For that reason, there was really no sketch here.

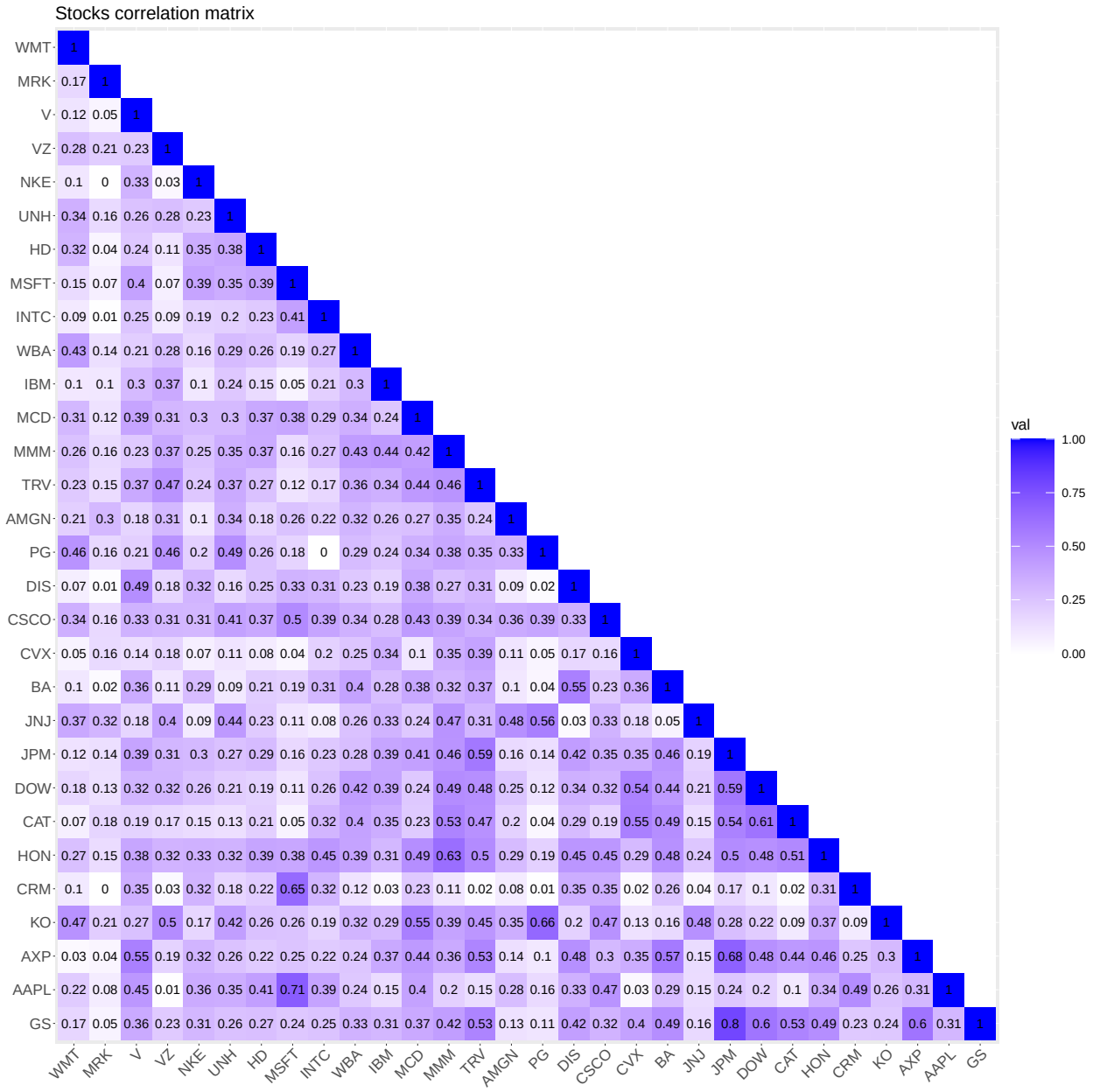
Regarding handling situations in which there is some correlation value missing, it sufficed to use `NA` value, which would result in a missing tile in the visualization.

However, there was no such situations in this case, and thus this feature cannot be seen.

The visualization

```
library(ggplot2)
library(plotly)

# get all unique tickers
ut <- data.frame(tickers=union(cor_data$Ticker.1, cor_data$Ticker.2))
rut <- data.frame(tickers=rev(ut$tickers))  # save a reversed copy for later

# create dataframe of all combinations
df <- expand.grid(ticker1=rut$tickers, ticker2=ut$tickers)

# read the correlation values
df$val <- NA # correlation not specified, cell will be colored black
for (i in 1:nrow(cor_data)) {
  # read from the dataset
  df$val[length(ut$tickers)*(match(cor_data$Ticker.1[i], ut$tickers) - 1) +
              match(cor_data$Ticker.2[i], rut$tickers)] = cor_data$Correlation.Value[i]
  # it's bidirectional
  df$val[length(ut$tickers)*(match(cor_data$Ticker.2[i], ut$tickers) - 1) +
              match(cor_data$Ticker.1[i], rut$tickers)] = cor_data$Correlation.Value[i]
}
j = length(ut$tickers)
for (i in 0:(length(ut$tickers) - 1)) {
  # remove upper triangle
  for (k in 0:i) {
    df$val[j - k] = NA
  }
  j = j + length(ut$tickers)
}
for (i in 0:(length(ut$tickers) - 1)) {
  # correlation = 1 between the same stock
  df$val[length(ut$tickers) + i*(length(ut$tickers) - 1)] = 1
}

# text for tooltip
df <- df %>%
  mutate(text = paste0(df$ticker1, "\n", df$ticker2, "\n", "Val: ", df$val))

# Heatmap
p = ggplot(df, aes(ticker1, ticker2, fill=val)) +
  geom_tile() +
  geom_text(aes(label=round(val, 2)),
```

```
            size=6
            ) +
  #scale_x_discrete(guide=guide_axis(n.dodge=2)) +
  theme(axis.title.x=element_blank(),  # remove x axis title
        axis.title.y=element_blank(),  # remove y axis title,
        text=element_text(size=20),
        axis.text=element_text(size=20),
        legend.key.size = unit(2, 'cm'),
        legend.key.height = unit(2, 'cm'),
        legend.key.width = unit(2, 'cm'),
        axis.text.x=element_text(angle=45, hjust=1)
        ) +
  scale_fill_gradient2(low="white", high="blue",
                       limits=c(c(0, 1)),
                       na.value="white"
                       ) +
  ggtitle("Stocks correlation matrix")
p
```



Digression - reinventing the candle stick chart

TODO