High-Tech Companies on NASDAQ

Ken Law

2023-02-27

Kaggle Profile (https://www.kaggle.com/maebel)

1. Ask
2. Prepare
3. Process
4. Analyze
5. Share
6. Act

ASK & PREPARE

Description: This dataset contains information on more than 8,000 high-tech companies in the electronics, computers, telecommunications, and biotechnology sectors listed on the NASDAQ exchange in the United States.

Variables:

1. Symbol: A unique identifier for each company listed on the NASDAQ exchange. (String)
2. Name: The name of the company. (String)
3. Price: The current price of the company's stock. (Float)
4. Pricing changes: The change in price from day-to-day. (Float)
5. Pricing percentage changes: The percentage change in price from day-to-day. (Float)
6. Sector: The sector the company belongs to. (String)
7. Industry: The industry the company belongs to. (String)
8. Market Capitalization: The market capitalization of the company. (Float)
9. Share Volume: The amount of shares traded over 24 hours. (Integer)
10. Earnings per share: A measure of profitability.(Float)
11. Annualized Dividend: Total amount of money a company pays out to its shareholders in dividends over the course of a year.(Float)
12. Dividend pay date (Date)
13. Symbol yield: The dividend yield paid by companies over time. ()
14. Beta: The volatility index. (Float)
15. Errors: Any errors associated with data imputed related to these columns. (String)

Business Tasks

Identify top-performing companies:
Sort the data based on the highest percentage change in stock price or market capitalization to identify the top-performing companies. This could be useful for identifying potential investment opportunities or for analyzing market trends.

Analyze pricing trends:
Track the daily changes in stock prices and pricing percentage changes for each company listed on the NASDAQ exchange. By analyzing pricing trends, you can identify which companies are performing well and which ones are not, and make informed investment decisions.

Analyze company size:
Use the data on market capitalization to compare the size of different companies. This could be useful for identifying potential acquisition targets or for understanding market trends.

Conduct sector and industry analysis:
Use the data on the sector and industry for each company to gain insights into which areas of the market are performing well and which ones are struggling. This information could be useful for investment decision-making or for identifying potential business opportunities.

Compare sector and industry performance:
Use the data on sector and industry to compare the performance of companies within the same sector or industry. This could help you identify which sectors or industries are outperforming others.

Evaluate market capitalization:
The data on market capitalization could be used to compare the size of different companies and identify potential investment opportunities.

Monitor share volume:
Use share volume data to understand the liquidity of each company's stock and identify potential investment opportunities.

Track earnings per share and dividend payouts:
Use this information to identify companies that are profitable and have a history of paying dividends to shareholders. This data could be useful for making investment decisions or for identifying potential acquisition targets.

Track dividend growth:
Track the annualized dividend data over time to see if the dividend payout has been increasing, decreasing, or staying the same. This could be useful for identifying companies with a strong track record of increasing their dividend payout.

Calculate dividend yield:
Use the data on annualized dividends and share price to calculate the dividend yield for each company. This could be useful for identifying potential investment opportunities for income-seeking investors.

Evaluate volatility:
The data on beta could be used to evaluate the volatility of each company's stock and identify potential investment opportunities or risks.

Compare volatility:
Compare the beta values of different companies to understand which ones are more volatile than others. This could be useful for identifying potential investment risks.

Monitor for errors:
Finally, monitor the errors column to identify any data imputation issues and take corrective action to ensure the accuracy of your analysis.

PROCESS

DATA-CLEANING

Load readr package to import data-set. Remove leading/trailing *white spaces* using the check option in readr.

```
library(here)
```

```
## here() starts at C:/Users/lawke/Desktop/portfolio_2023
```

```
library(readr)
```

Creating data-frame `nasdaq_2022` from `2022_03_17_02_06_nasdaq.csv` using the read_csv() function.

```
nasdaq_2022 <- read_csv("2022_03_17_02_06_nasdaq.csv")
```

```
## New names:
## Rows: 8339 Columns: 16
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: "," chr
## (13): symbol, name, pricing_changes, pricing_percentage_changes, sector,... dbl
## (2): ...1, price lgl (1): errors
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## • `` -> `...1`
```

Load R packages for cleaning.

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.0     ✔ purrr     1.0.1
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.1.8
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors
```

```
library(dplyr)
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(lubridate)
library(stringr)
```

Open the dataset in a new tab.

```
view(nasdaq_2022)
```

Show summaries of the overall dataset and data types.

```
skim_without_charts(nasdaq_2022)
```

Data summary

Name	nasdaq_2022
Number of rows	8339
Number of columns	16
________________________	
Column type frequency:	
character	13
logical	1
numeric	2
________________________	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
symbol	0	1.00	1	6	0	8339	0
name	0	1.00	3	229	0	8296	0
pricing_changes	0	1.00	5	9	0	1808	0
pricing_percentage_changes	2349	0.72	5	10	0	1613	0
sector	0	1.00	3	21	0	13	0
industry	0	1.00	3	62	0	142	0
market_cap	0	1.00	3	17	0	6757	0
share_volume	0	1.00	2	11	0	7600	0
earnings_per_share	0	1.00	3	12	0	1042	0
annualized_dividend	0	1.00	3	12	0	672	0
dividend_pay_date	0	1.00	1	12	0	484	0
symbol_yield	0	1.00	1	12	0	1367	0
beta	2241	0.73	1	12	0	793	0

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
errors	0	1	0.27	FAL: 6098, TRU: 2241

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
…1	0	1	4200.20	2419.35	0.00	2106.50	4202.0	6295.50	8386
price	0	1	96.29	5520.50	0.02	4.78	11.5	27.51	504036

```
glimpse(nasdaq_2022)
```

```
## Rows: 8,339
## Columns: 16
## $ ...1                        <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1…
## $ symbol                      <chr> "AAPL", "MSFT", "GOOG", "GOOGL", "AMZN", "T…
## $ name                        <chr> "Apple Inc. Common Stock", "Microsoft Corpo…
## $ price                       <dbl> 157.5100, 289.5600, 2639.7550, 2629.0100, 3…
## $ pricing_changes             <chr> "+2.42", "+2.41", "+46.545", "+45.05", "+61…
## $ pricing_percentage_changes <chr> "(+1.56%)", "(+0.84%)", "(+1.79%)", "(+1.74…
## $ sector                      <chr> "Technology", "Technology", "Technology", "…
## $ industry                    <chr> "Computer Manufacturing", "Computer Softwar…
## $ market_cap                  <chr> "2,699,423,838,000", "2,143,429,080,429", "…
## $ share_volume                <chr> "63,429,579", "22,790,662", "900,760", "1,0…
## $ earnings_per_share          <chr> "$6.04", "$9.39", "$112.23", "$112.23", "$6…
## $ annualized_dividend         <chr> "$0.88", "$2.48", "N/A", "N/A", "N/A", "N/A…
## $ dividend_pay_date           <chr> "Feb 10, 2022", "Jun 9, 2022", "N/A", "N/A"…
## $ symbol_yield                <chr> "0.58%", "0.9%", "N/A", "N/A", "N/A", "N/A"…
## $ beta                        <chr> "1.18", "0.91", "1.06", "1.06", "1.11", "2.…
## $ errors                      <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, F…
```

```
head(nasdaq_2022)
```

```
## # A tibble: 6 × 16
##    ...1 symbol name         price prici…¹ prici…² sector indus…³ marke…⁴ share…⁵
##   <dbl> <chr>  <chr>        <dbl> <chr>   <chr>   <chr>  <chr>   <chr>   <chr>
## 1     0 AAPL   Apple Inc. …  158. +2.42   (+1.56… Techn… Comput… 2,699,… 63,429…
## 2     1 MSFT   Microsoft C…  290. +2.41   (+0.84… Techn… Comput… 2,143,… 22,790…
## 3     2 GOOG   Alphabet In… 2640. +46.545 (+1.79… Techn… Intern… 1,724,… 900,760
## 4     3 GOOGL  Alphabet In… 2629. +45.05  (+1.74… Techn… Intern… 1,718,… 1,008,…
## 5     4 AMZN   Amazon.com,… 3009. +61.74  (+2.09… Consu… Catalo… 1,511,… 2,623,…
## 6     5 TSLA   Tesla, Inc.…  833. +30.70  (+3.83… Capit… Auto M… 845,70… 20,220…
## # … with 6 more variables: earnings_per_share <chr>, annualized_dividend <chr>,
## #   dividend_pay_date <chr>, symbol_yield <chr>, beta <chr>, errors <lgl>, and
## #   abbreviated variable names ¹pricing_changes, ²pricing_percentage_changes,
## #   ³industry, ⁴market_cap, ⁵share_volume
```

```
str(nasdaq_2022)
```

```
## spc_tbl_ [8,339 × 16] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ...1                       : num [1:8339] 0 1 2 3 4 5 6 7 8 9 ...
##  $ symbol                     : chr [1:8339] "AAPL" "MSFT" "GOOG" "GOOGL" ...
##  $ name                       : chr [1:8339] "Apple Inc. Common Stock" "Microsoft Corporation Common Stock" "Al
phabet Inc. Class C Capital Stock" "Alphabet Inc. Class A Common Stock" ...
##  $ price                      : num [1:8339] 158 290 2640 2629 3009 ...
##  $ pricing_changes            : chr [1:8339] "+2.42" "+2.41" "+46.545" "+45.05" ...
##  $ pricing_percentage_changes: chr [1:8339] "(+1.56%)" "(+0.84%)" "(+1.79%)" "(+1.74%)" ...
##  $ sector                     : chr [1:8339] "Technology" "Technology" "Technology" "Technology" ...
##  $ industry                   : chr [1:8339] "Computer Manufacturing" "Computer Software: Prepackaged Software"
"Internet and Information Services" "Internet and Information Services" ...
##  $ market_cap                 : chr [1:8339] "2,699,423,838,000" "2,143,429,080,429" "1,724,718,735,878" "1,718
,961,675,672" ...
##  $ share_volume               : chr [1:8339] "63,429,579" "22,790,662" "900,760" "1,008,687" ...
##  $ earnings_per_share         : chr [1:8339] "$6.04" "$9.39" "$112.23" "$112.23" ...
##  $ annualized_dividend        : chr [1:8339] "$0.88" "$2.48" "N/A" "N/A" ...
##  $ dividend_pay_date          : chr [1:8339] "Feb 10, 2022" "Jun 9, 2022" "N/A" "N/A" ...
##  $ symbol_yield               : chr [1:8339] "0.58%" "0.9%" "N/A" "N/A" ...
##  $ beta                       : chr [1:8339] "1.18" "0.91" "1.06" "1.06" ...
##  $ errors                     : logi [1:8339] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_double(),
##   ..   symbol = col_character(),
##   ..   name = col_character(),
##   ..   price = col_double(),
##   ..   pricing_changes = col_character(),
##   ..   pricing_percentage_changes = col_character(),
##   ..   sector = col_character(),
##   ..   industry = col_character(),
##   ..   market_cap = col_character(),
##   ..   share_volume = col_character(),
##   ..   earnings_per_share = col_character(),
##   ..   annualized_dividend = col_character(),
##   ..   dividend_pay_date = col_character(),
##   ..   symbol_yield = col_character(),
##   ..   beta = col_character(),
##   ..   errors = col_logical()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Checking and renaming columns.

```
colnames(nasdaq_2022)
```

```
##  [1] "...1"                       "symbol"
##  [3] "name"                       "price"
##  [5] "pricing_changes"            "pricing_percentage_changes"
##  [7] "sector"                     "industry"
##  [9] "market_cap"                 "share_volume"
## [11] "earnings_per_share"         "annualized_dividend"
## [13] "dividend_pay_date"          "symbol_yield"
## [15] "beta"                       "errors"
```

Rename *[1] …1* to *index_count* - Run previous code chunk to check column names.

```
colnames(nasdaq_2022)[colnames(nasdaq_2022) == "...1"] <- "index_count"
```

Data Validation & Transformation

1. Data Type
2. Data Range
3. Data Constraint
4. Data Consistency
5. Data Structure
6. Code Validation

Note: Remove all instances of the ($) symbol from a character string using the str_remove_all() function. Because $ is a special character in regex that matches the end of a line.
Regular expressions (regex), backslashes () are used to escape special characters that have a specific meaning in the regex syntax. This means that if you want to match a literal backslash in a regex pattern, you need to escape it with another backslash, like (\).

Data type check

```
class(nasdaq_2022$price)
```

```
## [1] "numeric"
```

```
class(nasdaq_2022$pricing_changes)
```

```
## [1] "character"
```

```
class(nasdaq_2022$sector)
```

```
## [1] "character"
```

```
class(nasdaq_2022$industry)
```

```
## [1] "character"
```

```
class(nasdaq_2022$market_cap)
```

```
## [1] "character"
```

```
class(nasdaq_2022$share_volume)
```

```
## [1] "character"
```

```
class(nasdaq_2022$earnings_per_share)
```

```
## [1] "character"
```

```
class(nasdaq_2022$annualized_dividend)
```

```
## [1] "character"
```

```
class(nasdaq_2022$dividend_pay_date)
```

```
## [1] "character"
```

```
class(nasdaq_2022$symbol_yield)
```

```
## [1] "character"
```

```
#Creating a clean dataframe from the original dataset.
clean_nasdaq <- nasdaq_2022
```

Convert Pricing Changes

```
#Convert to numeric Remove + symbol with REGEX "\\+"
clean_nasdaq <- clean_nasdaq %>%
  mutate(pricing_changes = as.numeric(str_remove_all(pricing_changes, "\\+")))

class(clean_nasdaq$pricing_changes)
```

```
## [1] "numeric"
```

Convert Pricing Percentage Changes

```
#Replacing [%+()] symbols with "" void using str_replace_all.
clean_nasdaq <- clean_nasdaq %>%
  mutate(pricing_percentage_changes = as.numeric(str_replace_all(pricing_percentage_changes, "[%+()]", "")))

class(clean_nasdaq$pricing_percentage_changes)
```

```
## [1] "numeric"
```

Convert Market Capitalization

```
#Remove commas and convert to numeric.
clean_nasdaq <- clean_nasdaq %>%
  mutate(market_cap = as.numeric(str_remove_all(market_cap, ",")))

class(clean_nasdaq$market_cap)
```

```
## [1] "numeric"
```

Convert Share Volume

```
#Remove commas and convert to integer.
clean_nasdaq <- clean_nasdaq %>%
  mutate(share_volume = as.integer(str_remove_all(share_volume, ",")))

class(clean_nasdaq$share_volume)
```

```
## [1] "integer"
```

Convert Earnings per Share

```
# Remove $ symbol with REGEX \\ and convert to numeric.
clean_nasdaq <- clean_nasdaq %>%
  mutate(earnings_per_share = as.numeric(str_remove_all(earnings_per_share, "\\$")))

class(clean_nasdaq$earnings_per_share)
```

```
## [1] "numeric"
```

Convert Annualized Dividend

```
# Remove $ symbol with REGEX \\ and convert to numeric.
clean_nasdaq <- clean_nasdaq %>%
  mutate(annualized_dividend = as.numeric(str_remove_all(annualized_dividend, "\\$")))

class(clean_nasdaq$annualized_dividend)
```

```
## [1] "numeric"
```

Dividend pay date

```
# Format date format to Month-Day-Year and convert to date.

clean_nasdaq <- clean_nasdaq %>%
  mutate(dividend_pay_date = mdy(dividend_pay_date))

class(clean_nasdaq$dividend_pay_date)
```

```
## [1] "Date"
```

Check and clean data into respective data variable types

```
class(clean_nasdaq$price)
```

```
## [1] "numeric"
```

```
class(clean_nasdaq$pricing_changes)
```

```
## [1] "numeric"
```

```
class(clean_nasdaq$market_cap)
```

```
## [1] "numeric"
```

```
class(clean_nasdaq$share_volume)
```

```
## [1] "integer"
```

```
class(clean_nasdaq$earnings_per_share)
```

```
## [1] "numeric"
```

```
class(clean_nasdaq$annualized_dividend)
```

```
## [1] "numeric"
```

```
class(clean_nasdaq$dividend_pay_date)
```

```
## [1] "Date"
```

ANALYZE

Price

```
#Sort by price in descending order and create stock_prices df.

stock_prices <- clean_nasdaq %>%
   arrange(desc(price))
```

Earnings per share

Earnings Per Share (EPS) is a measure of a company's profitability that represents the portion of the company's profit allocated to each outstanding share of its common stock. A company's EPS can be a good indicator of its profitability and future growth potential.

```
# Sort EPS in descending order
earnings_per_share <- clean_nasdaq %>%
  arrange(desc(earnings_per_share))
```

Price / Earnings per share (P/E ratio)

Price-to-Earnings (P/E) Ratio is a popular valuation metric that compares a company's current stock price to its earnings per share. A high P/E ratio may indicate that investors expect the company to experience strong growth in the future, while a low P/E ratio may indicate that investors have lower expectations for the company's growth potential.

```
# Calculating P/E ratio and adding a new column.
clean_nasdaq <- clean_nasdaq %>%
   mutate(PE_ratio = clean_nasdaq$price/clean_nasdaq$earnings_per_share)
```

```
# Create new PE_ratio df

PE_ratio_df <- data.frame(name = clean_nasdaq$name,
              price = clean_nasdaq$price,
              earnings_per_share = clean_nasdaq$earnings_per_share,
              PE_ratio = clean_nasdaq$PE_ratio,
              industry = clean_nasdaq$industry, sector = clean_nasdaq$sector)
```

```
# Sort P/E ratio in descending order
PE_ratio_df <- arrange(PE_ratio_df, desc(PE_ratio))
```

Dividend yield

Dividend yield reflects the percentage return on investment that investors can expect to receive from dividend payments alone. It is calculated by dividing the annual dividend yield indicates a higher return on investment in the form of dividend income, but a high dividend yield may also indicate that the market views the company as risky. Dividend yield analysis should be considered in conjunction with other fundamental analysis techniques to gain a comprehensive understanding of the investment potential of a stock.

```
clean_nasdaq <- clean_nasdaq %>%
  mutate(dividend_yield = clean_nasdaq$annualized_dividend / clean_nasdaq$price)
```

```
# Creating a new dividend_yield df
dividend_yield_df <- data.frame(name = clean_nasdaq$name,
             price = clean_nasdaq$price,
             annualized_dividend = clean_nasdaq$annualized_dividend,
             dividend_yield = clean_nasdaq$dividend_yield,
             industry = clean_nasdaq$industry,
             sector = clean_nasdaq$sector)
```

```
# Sort dividend yield in descending order
dividend_yield_df <- arrange(dividend_yield_df,desc(dividend_yield))
```

SHARE

Data visualization

```
# Loading data viz package
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
```

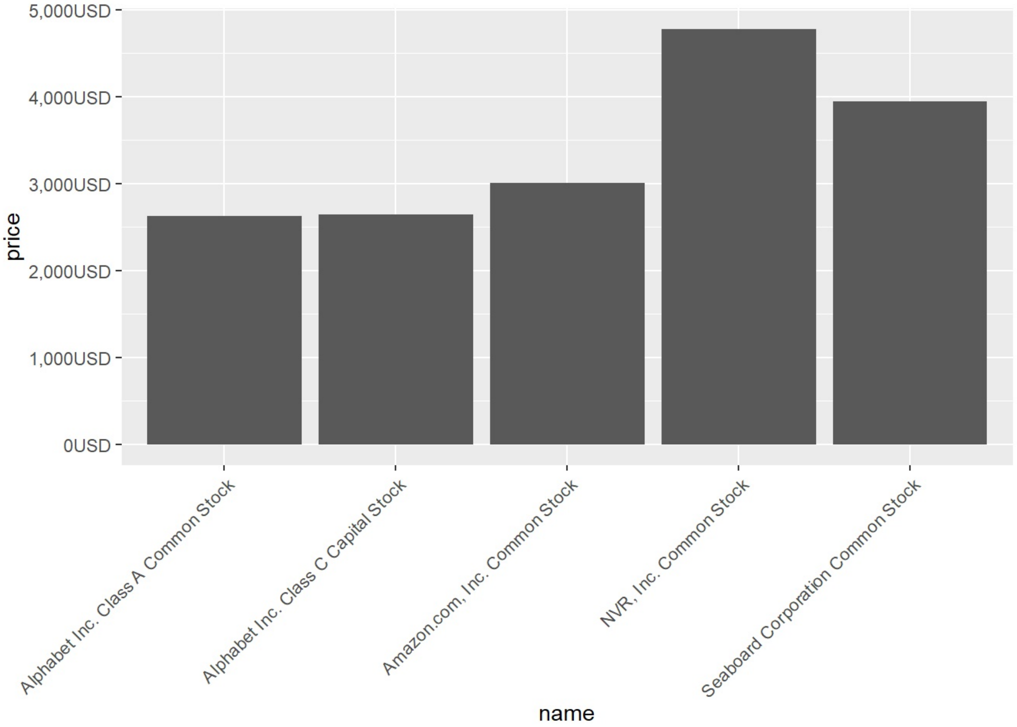
```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

Price

```
# PLot Share/Stock Prices
clean_nasdaq_filtered <- subset(clean_nasdaq, price >= 2600 & price < 10000)

ggplot(data = clean_nasdaq_filtered,aes(x = name, y = price)) + geom_bar(stat = "identity") + theme(axis.text.x =
element_text(angle = 45, hjust = 1)) + scale_y_continuous(labels = dollar_format(suffix = "USD", prefix = "", big
.mark = ",", decimal.mark = "."))
```



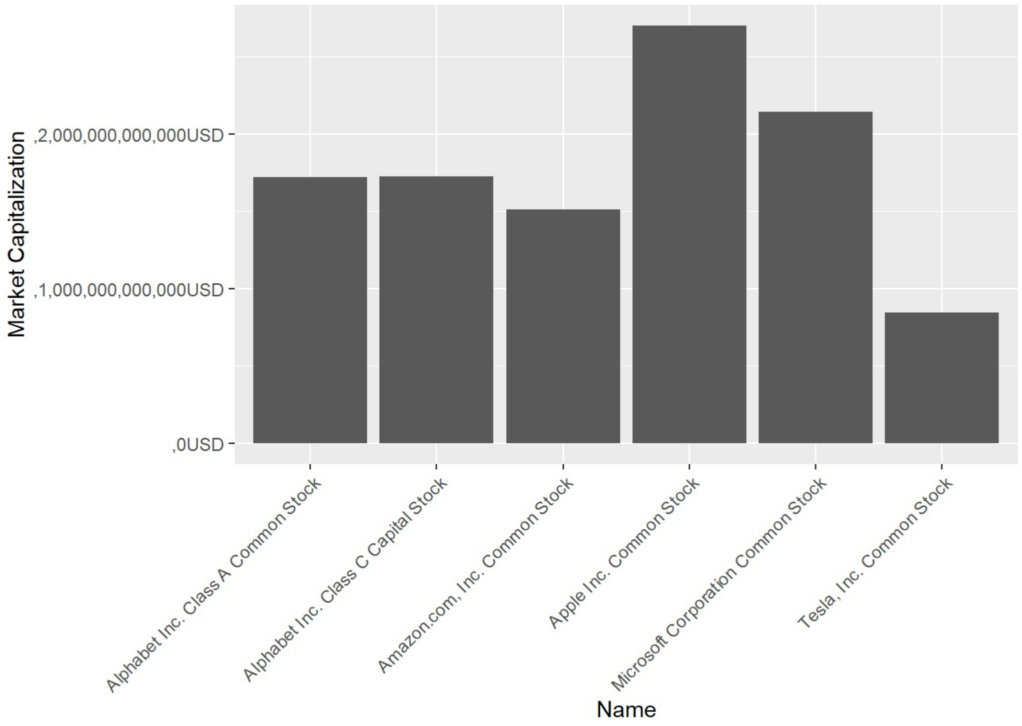
Top 5 Expensive stock prices ($ per share)

1. Berkshire Hathaway = $504,036
2. NVR, Inc = $4,774
3. Seaboard Corp = $3,940
4. Amazon.com = $3,009
5. Alphabet, Inc = $2,640

Market Capitalization

```
clean_nasdaq_filtered <- subset(clean_nasdaq, market_cap > 845708900000)

ggplot(data = clean_nasdaq_filtered, aes(x = name, y = market_cap)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = dollar_format(suffix = "USD", prefix = ",", big.mark = ",", decimal.mark = ".")) +
  labs(x = "Name", y = "Market Capitalization")
```



Top 5 Market Capitalization

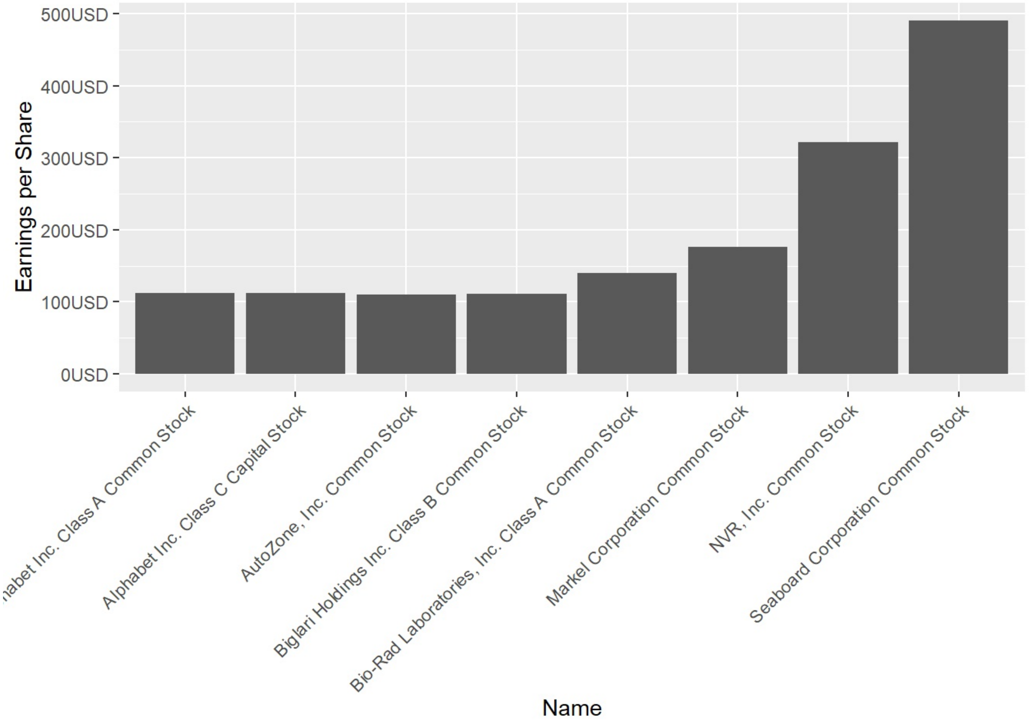
1. Apple, Inc = $2,699,424,000,000
2. Microsoft Corp = $2,143,429,000,000
3. Alphabet, Inc Class C = $1,724,719,000,000

4. Alphabet, Inc Class A = $1,718,962,000,000
5. Amazon.com, Inc = $1,511,268,000,000

Earnings per Share

```
clean_nasdaq_filtered <- subset(clean_nasdaq, earnings_per_share > 100)

ggplot(data = clean_nasdaq_filtered,aes(x = name, y = earnings_per_share)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = dollar_format(suffix = "USD", prefix = "", big.mark = ",", decimal.mark = ".")) +
  labs(x = "Name", y = "Earnings per Share")
```



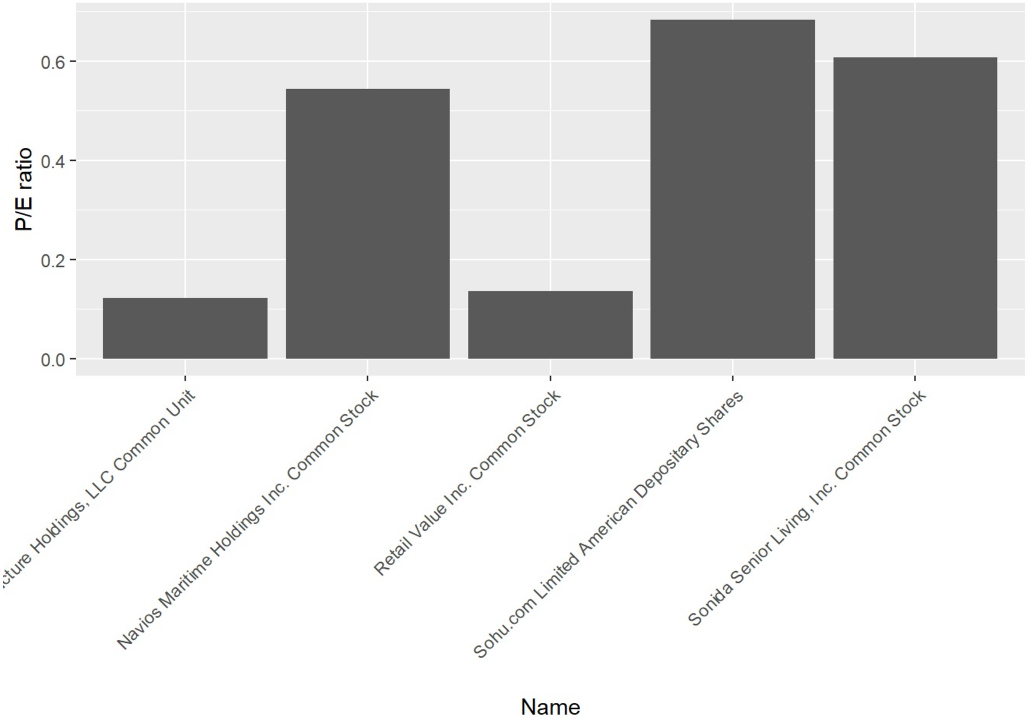
Top 5 Earnings per Share (EPS)

1. Seaboard Corp = $490
2. NVR, Inc = $321
3. Markel Corp = $176
4. Bio-Rad Labs = $140
5. Alphabet, Inc = $112

Price to Earnings (P/E) ratio

```
clean_nasdaq_filtered <- subset(PE_ratio_df, PE_ratio > 0 & PE_ratio < 0.7)

ggplot(data = clean_nasdaq_filtered, aes(x= name, y = PE_ratio)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Name", y = "P/E ratio")
```



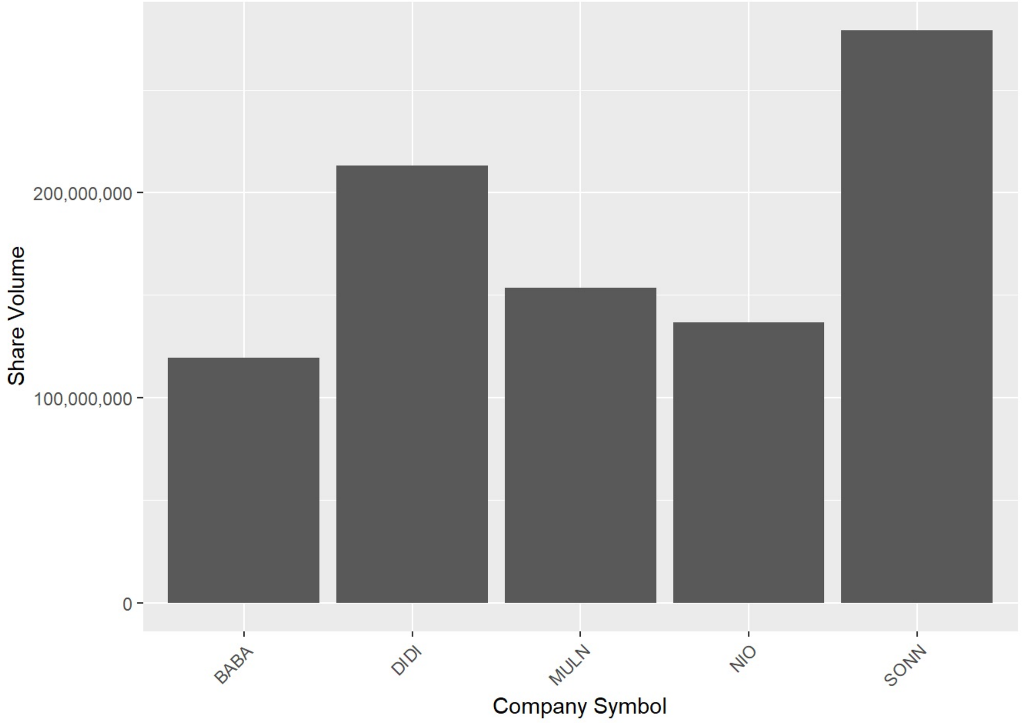
Lowest P/E ratio

1. Macquarie Infrastructure = 0.12
2. Retail Value Inc. = 0.14
3. Navios Maritime = 0.54
4. Sonida Senior Living = 0.61
5. Sohu.com Limited = 0.68

Share Volume

```
share_volume_df <- subset(clean_nasdaq, share_volume > 110000000)

ggplot(data = share_volume_df, aes(x = symbol, y = share_volume)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Company Symbol", y = "Share Volume") +
  scale_y_continuous(labels = comma)
```



Top 5 Share Volume

1. Sonnet BioTherapeutics = 279,021,147
2. DiDi Global, Inc = 213,252,041
3. Mullen Automotive, Inc = 153,618,147

4. NIO, Inc* = 136,551,330
5. Alibaba Group* = 119,603,599
   **NIO Inc. & Alibaba Group - American depository shares (Each representing one Class A ordinary share)*

ACT

Disclaimer
Analyzing stock Stock markets are often considered a low validity environment when it comes to forecasting short-term stock price changes over the long-term due to several factors.

Firstly, stock markets are highly unpredictable and subject to sudden changes in sentiment, which can lead to short-term volatility in stock prices. This volatility makes it difficult to accurately predict short-term price changes, as even small fluctuations in market sentiment can have a significant impact on stock prices.

Secondly, there are a wide range of factors that can impact stock prices, many of which are difficult to predict. These factors can include economic indicators, political events, company-specific news, and even rumors and speculation. Trying to predict how all of these factors will interact in the short-term is a difficult task, making it challenging to accurately forecast short-term price changes.

Thirdly, stock markets are often influenced by irrational behavior, such as herd mentality and emotional reactions, which can cause prices to deviate significantly from their intrinsic value. This can make it difficult to predict short-term price changes based solely on fundamental analysis, as market sentiment can often trump fundamental factors.

Finally, it’s worth noting that while short-term forecasting is difficult, longer-term forecasting is generally more reliable. Over the long-term, stock prices tend to follow earnings growth, making it easier to predict how a company’s earnings will impact its stock price over the long-term. However, even long-term forecasting can be subject to unexpected events, such as major changes in the competitive landscape or shifts in consumer behavior, which can impact a company’s earnings and stock price in unforeseen ways.