Title

Author 1*
Department of YYY, University of XXX
and
Author 2
Department of ZZZ, University of WWW

March 5, 2024

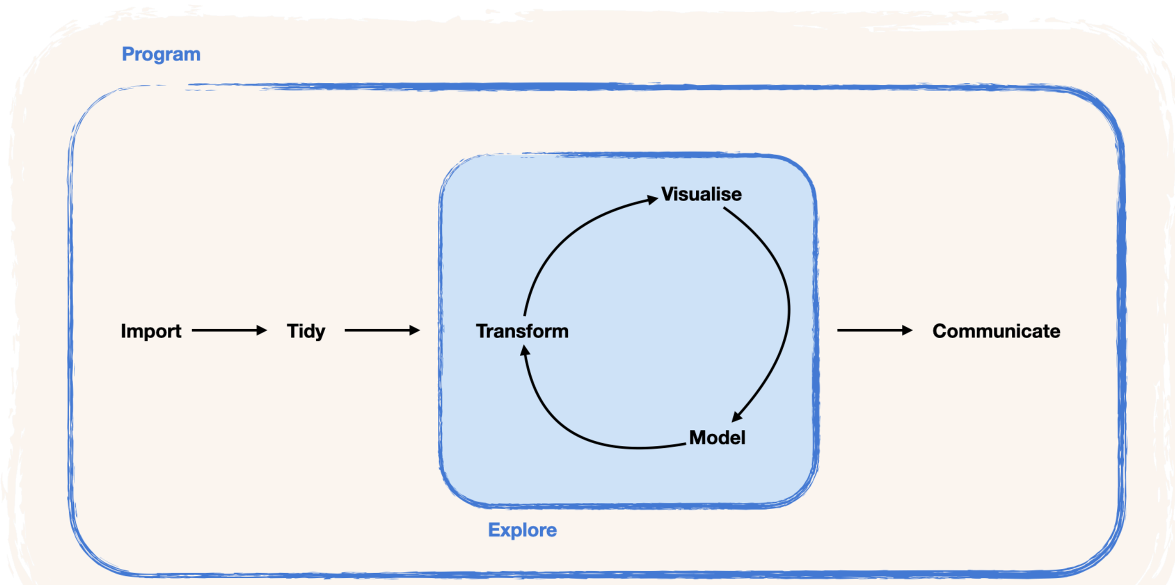
Abstract

The text of your abstract. 200 or fewer words.

Keywords: 3 to 6 keywords, that do not appear in the title

*The authors gratefully acknowledge *please remember to list all relevant funding sources in the non-anonymized (unblinded) version.*

In this lesson we will explore how to carry the steps of the Data Science pipeline to produce an analysis



The Data Pipeline (from Wickham & Grolemund, 2017)

To illustrate, we will study the stocks of **five tech companies**

- First we'll focus on a single company
- Then we'll see how all these companies have fared over time.

Importing the data

- We've imported the data using the package `{tidyquant}`.

```
get_what <- "stock.prices"

companies <- c("AAPL",

               "MSFT",

               "GOOG",

               "AMZN",

               "TSLA")
```

```r
stocks_data <-
  companies |>
  map(tq_get, get = get_what) |>
  bind_rows()
```

- The dataset contains the information on stocks for 6 companies: Apple Inc. **(AAPL)**, Microsoft Corporation **(MSFT)** , Alphabet Inc. **(GOOG)**, Amazon Inc. **(AMZN)** and Tesla Inc. **(TSLA).**

```r
stocks_data |>
  slice_head(n=6) |>
  kable()
```

Table 1: The first six lines of the whole dataset

symbol	date	open	high	low	close	volume	adjusted
AAPL	2014-01-02	19.84571	19.89393	19.71500	19.75464	234684800	17.29665
AAPL	2014-01-03	19.74500	19.77500	19.30107	19.32071	392467600	16.91672
AAPL	2014-01-06	19.19464	19.52857	19.05714	19.42607	412610800	17.00897
AAPL	2014-01-07	19.44000	19.49857	19.21143	19.28714	317209200	16.88733
AAPL	2014-01-08	19.24321	19.48429	19.23893	19.40929	258529600	16.99427
AAPL	2014-01-09	19.52857	19.53071	19.11964	19.16143	279148800	16.77726

Tidying the data

1 Choosing a Company

- A ticker can be stored in the variable `company_ticker`.
- Later, this will help us *parametrise* our report

```
company_ticker <- "AAPL"
```

2 Choosing a Timeframe

- To narrow our focus, we restrain our analysis to a given timeframe.
- We will focus on the performance of these stocks since the beginning of the COVID-19 pandemic (March 11 2020) until now.
- The `start_date` variable can store this information.

```
start_date <-  ymd("2020-03-11")
```

3 Filtering the data

- We're interested in the company APPLE INC, we can use the ticker AAPL
- We'll use the `filter()` to subset the data for the company in the given timeframe.

```
company_data <-
  stocks_data |>
  filter(symbol == company_ticker,
         date > start_date)
```

- We can explore the data by printing its first 6 lines:

```r
head(company_data) |>
  kable()
```

Table 2: The first six lines of the apple dataset

symbol	date	open	high	low	close	volume	adjusted
AAPL	2020-03-12	63.9850	67.5000	62.0000	62.0575	418474000	60.52466
AAPL	2020-03-13	66.2225	69.9800	63.2375	69.4925	370732000	67.77602
AAPL	2020-03-16	60.4875	64.7700	60.0000	60.5525	322423600	59.05684
AAPL	2020-03-17	61.8775	64.4025	59.6000	63.2150	324056000	61.65356
AAPL	2020-03-18	59.9425	62.5000	59.2800	61.6675	300233600	60.14429
AAPL	2020-03-19	61.8475	63.2100	60.6525	61.1950	271857200	59.68346

Understanding the data

4 Visualise

- With this data and the functions in `ggplot()`, we can create a first visualisation of the closing stock price (`close`).
- We set the dates on the `x` axis and the `close` price in the `y` axis.

```r
company_data |>
  ggplot(aes(x = date))+
  geom_line(aes(y = close))
```

5 Transform

- The visualisation offers a first glance. We can transform again to ask the questions on the returns.
- Remember that the difference in log-price are approximations of the returns, i.e. the **percentage gain after selling the stock**.

Definition 5.1. Let p_t denote the closing price of the stock, the log-return r_t can be defined as:

$$r_t = \log(p_t) - \log(p_{t-1}) \approx \frac{p_t - p_{t-1}}{p_{t-1}} \tag{1}$$

- We use the function `mutate()`, alongside `lag()` to create a column with the daily (log) returns and the definition in equation Definition 5.1

```
company_data <- company_data |>
  mutate(daily_log_returns = log(close)-log(lag(close)))
```

6 Visualize again: log-returns

We can construct a visualisation with this. Additionally, we can add layers to our visualisation to decorate it at will.

```r
company_data |>
  ggplot(aes(x = date))+
  geom_line(aes(y = daily_log_returns), alpha = 0.5, color = "#555555") +
  geom_hline(yintercept = 0, lty = 3)+
  labs(
    title = str_glue("Daily Returns of the stock for {company_ticker}"),
    subtitle = str_glue("Close stock prices since {start_date |> format('%d %B, %Y')}")
    x = "Date",
    y = "Returns"
    ) +
  theme_minimal()
```

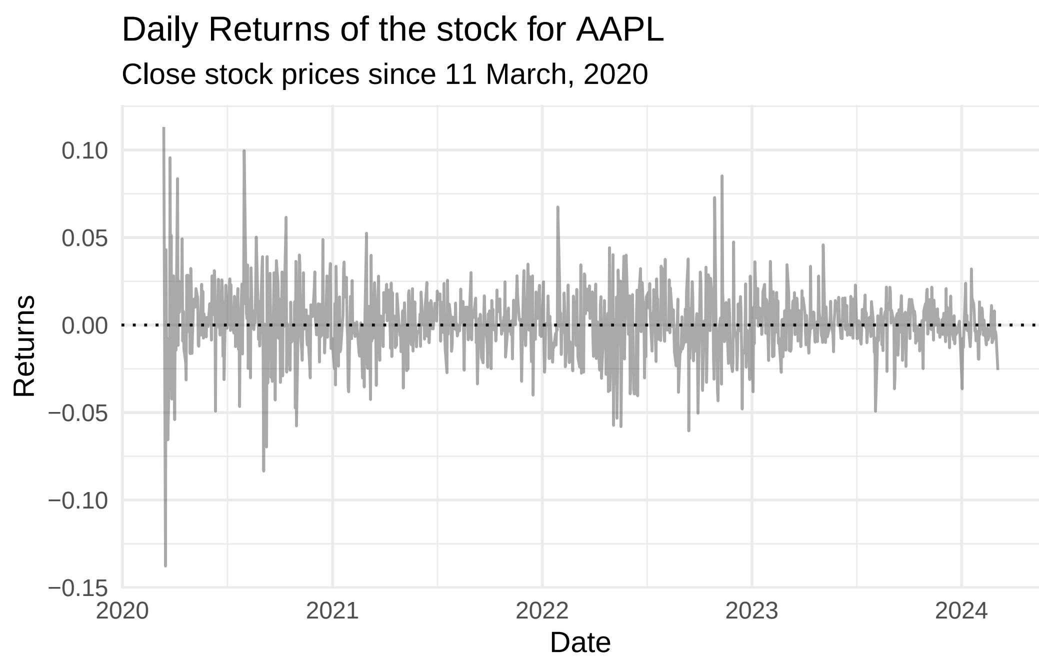


Figure 1: Log-returns for AAPL

! Look at the years!

What can you say about this plot?

- It appears that there is a high **variability** of the log-returns in the year 2020
- This increase in **variance** seems to stabilise in 2021 and **reappear** in 2022
- The year 2023 is also quite stable

All these point to signs of increased **variability** in times of **global crises**, which have added elements of uncertainty to the global supply chain.

7 Model: Create summary statistics

- We can obtain a summary table for some summary statistics with `summarise()`.
- We will compute the average return $\bar{r}$ and estimate the standard deviation (SD) of the log-returns $\hat{\sigma}$

Definition 7.1. These statistics are defined as follows:

$$\bar{r} = \frac{1}{T}\sum_{t=1}^{T} r_t \tag{2}$$

$$\hat{\sigma} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T} (r_t - \bar{r})^2} \tag{3}$$

> A word of warning
>
> While the average return might not mean much in theoretical terms, the standard deviation might give some idea of the *risk* or *volatility*.

- We can show the values in the following table:

```
company_data |>
  summarise(`Average Return` = mean(daily_log_returns, na.rm = TRUE),
            `Average Risk (SD)` = sd(daily_log_returns, na.rm = TRUE)) |>
  gt() |>
  tab_header(
    title= "Summary statistics of Tech companies stocks",
    subtitle  = str_glue("From {start_date |> format('%d %b, %Y')} to {Sys.Date() |> fo
  fmt_number(decimals  = 4)
```

Table 3: Summary statistics of the log-returns for AAPL

Summary statistics of Tech companies stocks

From 11 Mar, 2020 to 05 Mar, 2024

Average Return	Average Risk (SD)

0.0010	0.0200

7.1 Summary statistics by year

And, seeing that the visualisation shows periods of high volatility in the year 2020, we can compute yearly measures of risk and volatility:

```
company_data |>
  mutate(year = year(date)) |>
  group_by(year) |>
  summarise(`Average Return` = mean(daily_log_returns, na.rm = TRUE),
            `Risk (SD)` = sd(daily_log_returns, na.rm = TRUE)) |>
  gt() |>
  tab_header(
    title= "Summary statistics of Tech companies stocks",
    subtitle  = str_glue("From {start_date |> format('%d %b, %Y')} to {Sys.Date() |> fo
  fmt_number(columns = -year, decimals  = 4) |>
  tab_style(
    style = list(
      cell_fill(color = "lightgreen"),
      cell_text(color = "white")),
    locations =
      cells_body(columns = `Average Return`,
                 rows= `Average Return` >0)
  ) |>
```

```
tab_style(
  style = list(
    cell_fill(color = "red"),
    cell_text(color = "white")),
  locations =
    cells_body(columns = `Risk (SD)`,
               rows= `Risk (SD)` >0.020)
)
```

Summary statistics of Tech companies stocks

From 11 Mar, 2020 to 05 Mar, 2024

year	Average Return	Risk (SD)
2020	0.0037	0.0283
2021	0.0012	0.0158
2022	−0.0012	0.0224
2023	0.0016	0.0128
2024	−0.0022	0.0123

! Look at the years (again)!

We're highlighting the years where the risk is higher. These numbers reflect the remarks we've made in the previous points, namely that global uncertainties have affected the risk of this stock.

Analyzing multiple companies

- Now we'll carry the same analysis for multiple companies

- We import the data and prepare it for analysis

```r
returns_data <-

  stocks_data |>

  filter(date > start_date) |>

  select(symbol,date, close) |>

  group_by(symbol) |>

  mutate(daily_log_returns = log(close) - log(lag(close))) |>

  ungroup()
```

8 Plotting the stock price

- Let's plot the whole

```r
returns_data |>

  ggplot(aes(x = date, y = close,  group = symbol, color = symbol))+

  geom_line(alpha=0.5)+

  labs(

    title = str_glue("Daily price for {glue::glue_collapse(company_tickers, sep = ', ')

    subtitle = str_glue("Close stock prices since {start_date |> format('%d %B, %Y')}")

    x = "Date",

    y = "Close Price"

    ) +

  theme_minimal()+

  scale_color_manual(values = company_colors)
```

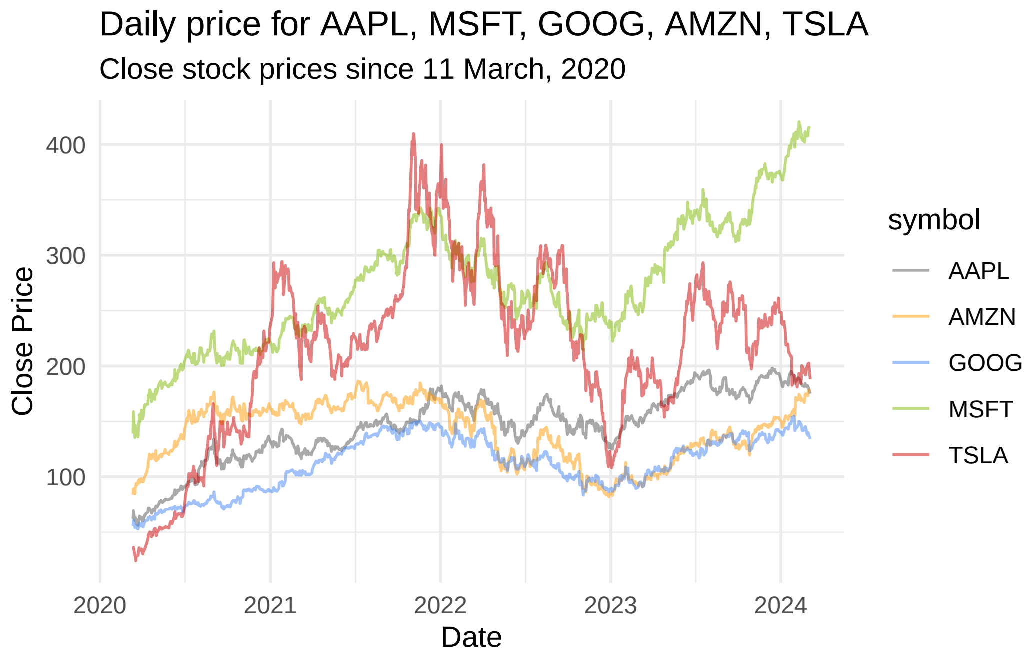


Figure 2: Stocks prices for all companies in the selected period

- A single chart is not very satisfactory, even if we try to make differentiate companies with colors.

9 Visualising the log-returns

- As the interest is in the `log`-returns, let's plot the returns in a single frame

```
returns_data |>
  ggplot(aes(x = date, y = daily_log_returns,  group = symbol, color = symbol))+
  geom_line(alpha = 0.5) +
  labs(
    title = str_glue("Daily Returns of the stock for all companies"),
    subtitle = str_glue("Close stock prices since {start_date |> format('%d %B, %Y')}")
    x = "Date",
```

```
  y = "Returns"
  ) +
theme_minimal()+
theme(legend.position = "none")+
scale_color_manual(values = company_colors)
```

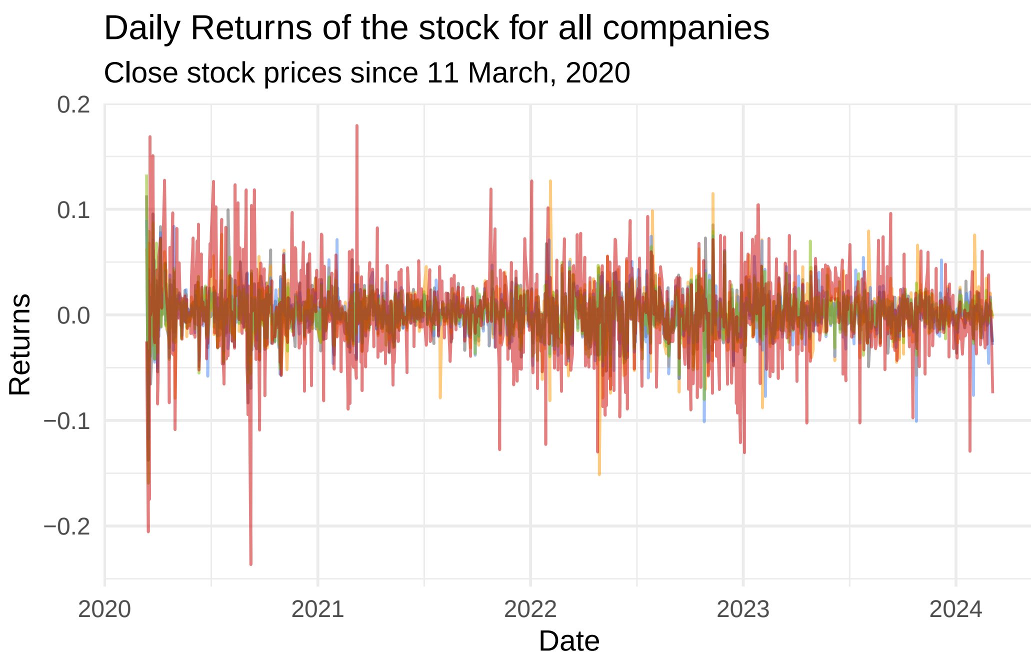


Figure 3: Log returns of all companies

> **!** Create a faceted chart
>
> - There's a problem here, as you can see. The log-returns are confounded and become difficult to distinguish in spite of the use of relevant colors!

10 Creating a faceted chart

- To distinguish better, we can use the function `facet_wrap()` .
- This creates a _mini_ plot according to a variable

```r
returns_data |>

  ggplot(aes(x = date, y = daily_log_returns,  group = symbol, color = symbol))+

  geom_line(alpha = 0.5) +

  labs(

    title = str_glue("Daily Returns of the stock for all companies"),

    subtitle = str_glue("Close stock prices since {start_date |> format('%d %B, %Y')}")

    x = "Date",

    y = "Returns"

    ) +

  theme_minimal()+

  facet_grid(rows = vars(symbol)) +                                          ①

  theme(legend.position = "none")+

  scale_color_manual(values = company_colors)
```

① On this line we added the faceted chart

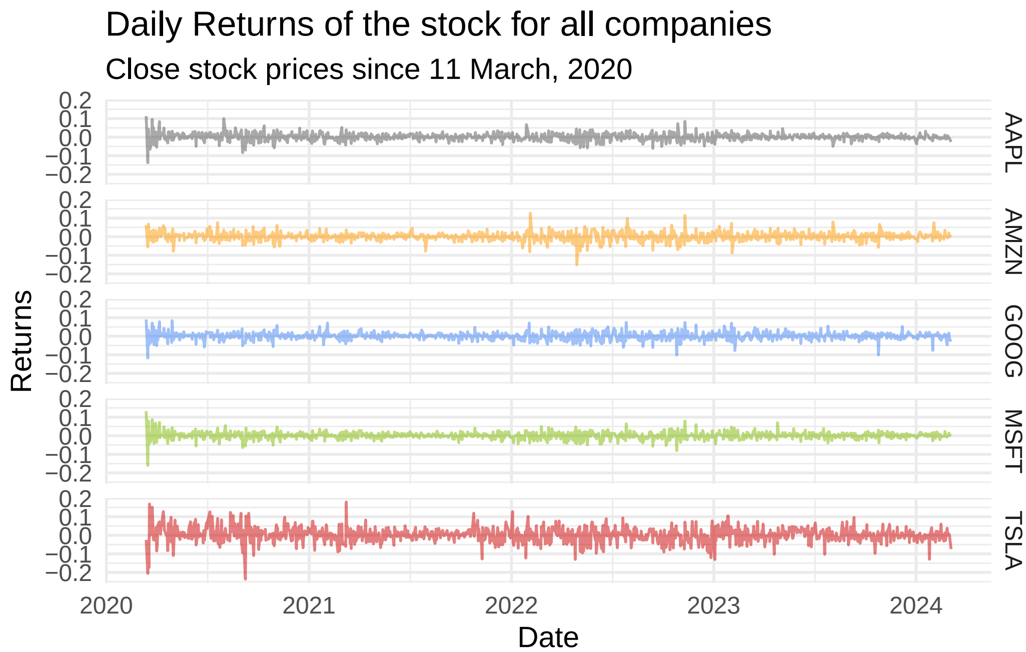


Figure 4: Log returns of all companies - faceted

> ❗ What can we see?
>
> - Uncertainties in the supply chains produced by the international crises seem to have affected companies in the same way
> - However the magnitude of these impacts seems to have been different across companies, e.g.
>   - TSLA shows significantly more risk

11 Average Returns for all companies

```
mean_and_sd <-
  returns_data |>
  mutate(year = year(date)) |>
  group_by(symbol, year) |>
  summarise(
```

```r
    average_return = mean(daily_log_returns, na.rm = TRUE),
    sd_return = sd(daily_log_returns, na.rm = TRUE)
    ) |>
  ungroup()


mean_and_sd |>
  select(-sd_return) |>
  pivot_wider(names_from = year,values_from = average_return) |>
  gt() |>
  tab_header(
    title= "Average Yearly returns of Tech companies",
    subtitle  = str_glue("From {start_date |> format('%d %b, %Y')} to {Sys.Date() |> fo
  fmt_number(columns = -symbol, decimals  = 4) |>
  tab_style(
    style = list(
      cell_fill(color = "lightgreen"),
      cell_text(color = "white")),
    locations = list(
      cells_body(columns =`2020`,
                 rows= `2020`>0 ),
      cells_body(columns =`2021`,
                 rows= `2021`>0 ),
       cells_body(columns =`2022`,
                 rows= `2022`>0 ),
```

```
    cells_body(columns =`2022`,
               rows= `2022`>0 ),
    cells_body(columns =`2023`,
               rows= `2023`>0 ),
   cells_body(columns =`2024`,
              rows= `2024`>0 )
  )
) |>
tab_style(
  style = list(
    cell_fill(color = "red"),
    cell_text(color = "white")),
      locations = list(
    cells_body(columns =`2020`,
               rows= `2020`<=0 ),
    cells_body(columns =`2021`,
               rows= `2021`<=0 ),
     cells_body(columns =`2022`,
               rows= `2022`<=0 ),
     cells_body(columns =`2022`,
               rows= `2022`<= 0 ),
     cells_body(columns =`2023`,
               rows= `2023`<=0 ),
    cells_body(columns =`2024`,
```

```
                 rows= `2024`<=0 )
    )
  )
```

Table 4: Average Returns for all companies

Average Yearly returns of Tech companies

From 11 Mar, 2020 to 05 Mar, 2024

symbol	2020	2021	2022	2023	2024
AAPL	0.0037	0.0012	−0.0012	0.0016	−0.0022
AMZN	0.0033	0.0001	−0.0027	0.0024	0.0036
GOOG	0.0022	0.0020	−0.0019	0.0019	−0.0011
MSFT	0.0023	0.0016	−0.0013	0.0018	0.0023
TSLA	0.0090	0.0016	−0.0042	0.0028	−0.0065

12 Risk (SD) all companies

```
mean_and_sd |>
  select(-average_return) |>
  pivot_wider(names_from = year,values_from = sd_return) |>
  gt() |>
  tab_header(
    title= "Average Yearly SDs of Tech companies",
    subtitle  = str_glue("From {start_date |> format('%d %b, %Y')} to {Sys.Date() |> fo
  fmt_number(columns = -symbol, decimals  = 4) |>
```

```
tab_style(

  style = list(

    cell_fill(color = "red"),

    cell_text(color = "white")),

      locations = list(

    cells_body(columns =`2020`,

                rows= `2020`>=0.020),

    cells_body(columns =`2021`,

                rows= `2021`>=0.020 ),

     cells_body(columns =`2022`,

                rows= `2022`>=0.020  ),

     cells_body(columns =`2022`,

                rows= `2022`>=0.020  ),

     cells_body(columns =`2023`,

                rows= `2023`>=0.020  ),

    cells_body(columns =`2024`,

                rows= `2024`>=0.020)

  )

)
```

Table 5: Risk (SD) all companies in the period

Average Yearly SDs of Tech companies

From 11 Mar, 2020 to 05 Mar, 2024

symbol	2020	2021	2022	2023	2024

AAPL	0.0283	0.0158	0.0224	0.0128	0.0123
AMZN	0.0237	0.0152	0.0316	0.0207	0.0180
GOOG	0.0234	0.0149	0.0244	0.0193	0.0185
MSFT	0.0268	0.0132	0.0223	0.0157	0.0113
TSLA	0.0538	0.0342	0.0423	0.0340	0.0309

References