Exercises: Dow Jones monthly performance

YSC2210 - DAVis with R

Michael T. Gastner

Introduction

'Sell in May and go away' is an investment strategy based on the hypothesis that stock market prices perform weaker between May and October than during all other months. In this exercise, we want to test whether there are monthly differences, using the Dow Jones Industrial Average as an indicator of stock market trends since 1896 . Our approach only uses very basic statistical tools, and the main learning outcome lies in the data wrangling rather than the statistics. For a more sophisticated statistical analysis, see Witte (2010).

Data

We work with data from S&P Global, a company specialising in financial information. Please go to: https://www.spglobal.com/spdji/en/indices/equity/dow-jones-industrial-average/?go=industrial-index-data#overview

In the 'Documents' section, activate the drop-down menu 'Additional Info' (figure 1). Click on 'DJIA Monthly Performance History' to download an Excel spreadsheet named `dja-performance-report-monthly.xls`. Move this spreadsheet into your project folder.

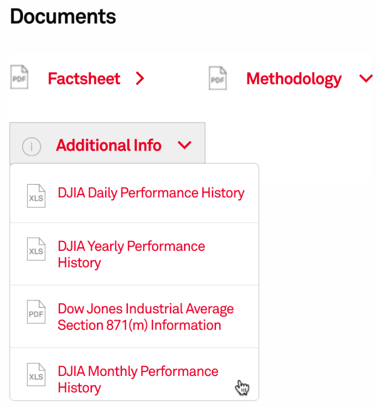
Documents
Factsheet
Methodology
Additional Info
DJIA Daily Performance History
DJIA Yearly Performance History
Dow Jones Industrial Average Section 871(m) Information
DJIA Monthly Performance History

Figure 1: Screenshot of the S&P Global website.

Tasks

(1) Import the Excel spreadsheet as a tibble named `dj`, which stands for 'Dow Jones'. Make sure you correctly cut off non-data rows at the top and bottom.

(2) Remove all columns except 'Effective Date' and 'Close Value'.

(3) Column names with spaces are awkward to work with in R. Change the names of the columns so that they are short and do not contain spaces: `date` and `close_value`.

(4) What are the R classes of the columns?

(5) Append a column month with the month that is implicit in the date column. One option is to use the function mdy(), which stands for *m*onth-*d*ay-*y*ear, followed by month() to extract the month as a number. Both functions are in the **lubridate** package. Here is an example:

```r
library(lubridate)  # Put this command at the top of your RMD file
mdy("08/31/1956") |>
  month()
```

```
## [1] 8
```

(6) Answer the question: are all months in dj in consecutive calendrical order? The result should be a logical value (i.e. TRUE or FALSE). Do not use a for-loop. You may find the functions lead() and lag() in **dplyr** useful for this task.

Be careful! Suppose we had the dates "6/30/1867" and "7/31/1869" in consecutive rows of dj. The months appear to be consecutive (6 and 7), but they are in different years; thus, they are not consecutive calendrical months. Ensure that your code handles such cases correctly.

(7) In the previous problem, you should have found that not all months are in consecutive calendrical order. Use R to find the date just before the gap(s). Perform some research about the reason for the gap(s). Summarise your findings in maximally 10 sentences.

(8) Append a column rel_change_pct with the relative change of the Dow Jones (in percent) compared to its value at the start of the month. In the first row, enter NA as a sign that there is no closing value in the previous month; Charles Dow calculated the eponymous index for the first time in May 1896. Again, do not use a for-loop.

(9) Make a quick-and-dirty plot of the Dow Jones's relative change as a function of time using the following code:

```r
# Remove first row because it contains NA
ggplot(dj[-1, ], aes(mdy(date), rel_change_pct)) +
  geom_line()
```

(10) Which month saw the largest relative increase in the history of the Dow Jones? Perform some research about the reason for the stock market rally. Summarise your findings in maximally 10 sentences.

(11) For the purpose of plotting, append a column month_abb that contains the abbreviated name of the month as a factor (e.g. "Jan", "Feb"). You can find the abbreviations in R's built-in vector month.abb. Sort levels in chronological order from "Jan" to "Dec".

(12) Make a quick-and-dirty box plot of relative change as a function of month.

```r
ggplot(dj[-1, ], aes(month_abb, rel_change_pct)) +
  geom_boxplot()
```

Which preliminary conclusion can you draw from the box plot?

(13) Which month has the highest median relative change? Which month has the lowest (i.e. the most strongly negative) change? No for-loop! In Quantitative Reasoning, you learned how to get the answer with aggregate().[1]

(14) Now we know the differences in the median between different months, but are these differences statistically significant?

The Kruskal-Wallis test is a nonparametric statistical method to infer whether samples come from the same distribution. To test the hypothesis that there are monthly differences, the null hypothesis is

[1] We learn another dplyr-based method later in this course.

that the relative changes come from the same distribution in all months. Reactivate your Quantitative Reasoning knowledge to interpret the result of:

```r
kruskal.test(rel_change_pct ~ month, data = dj)
```

Comment on the result.

References

Witte, H. D. (2010). Outliers and the Halloween effect: comment on Maberly and Pierce. *Econ Journal Watch*, **7**(1), 91–98.