Solutions: Dow Jones monthly performance

YSC2210 - DAVis with R

Michael T. Gastner

We need the following packages for these exercises.

```
library(lubridate)
library(readxl)
library(tidyverse)
```

(1)
```
dj <- read_excel("dja-performance-report-monthly.xls", skip = 4)
```

It is possible to hard-code the number of rows in `read_excel()` with the argument `n_max`. However, this number is going to change when we download the spreadsheet in the future. Instead, I take the presence of an 'Effective Date' as a sign that the corresponding row should be included.

Base-R option:

```
dj <- dj[!(is.na(dj[["Effective Date"]])), ]
```

Later in this course, we use **dplyr** to subset data frames. Here is the **dplyr**-based equivalent of the previous code chunk.

```
dj_dplyr <- read_excel("dja-performance-report-monthly.xls", skip = 4) |>
  filter(!is.na(`Effective Date`))
```

(2)
```
dj <- dj[c("Effective Date", "Close Value")]
```

(3) Here is the base-R option:

```
names(dj) <- c("date", "close_value")
```

And here is the **dplyr**-based alternative to steps (2) and (3).

```
dj_dplyr <- dj_dplyr |>
  select(date = "Effective Date", close_value = "Close Value")
```

(4) Here is a base-R option.

```
class(dj$date)
```

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

```
class(dj$close_value)
```

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

And, as a bonus, here are two more ways to accomplish the same task. Both of these options avoid hard-coding the column names.

(a) Use the base-R function `sapply()`.

```
sapply(dj, class)
```

```
##        date close_value
## "character"   "numeric"
```

(b) Use the function `map_chr()` from the **purrr** package, which is part of the tidyverse. We cover **purrr** later in this course.

```
map_chr(dj, class)
```

```
##        date close_value
## "character"   "numeric"
```

(5)
```
dj$month <-
  mdy(dj$date) |>
  month()
```

Here is a **dplyr**-based alternative.

```
dj_dplyr <- dj_dplyr |>
  mutate(month = mdy(date) |> month())
```

(6)
```
dj$year <-
  mdy(dj$date) |>
  year()
dj$next_month_in_dj <- lead(dj$month)
dj$year_of_next_month_in_dj <- lead(dj$year)
dj$next_month_in_calendar <- if_else(dj$month < 12, dj$month + 1, 1)
dj$year_of_next_month_in_calendar <- dj$year + (dj$month == 12)

# Are all months in `dj` in consecutive calendrical order?
all(dj$next_month_in_dj == dj$next_month_in_calendar &
      dj$year_of_next_month_in_dj == dj$year_of_next_month_in_calendar)
```

```
## [1] FALSE
```

`dj$next_month_in_dj[nrow(dj)]` equals `NA` because the last row does not have a next row. The same is true for `dj$year_of_next_month_in_dj`. For this reason, I remove the last row from `dj`.

```
dj_without_last_row <- dj[-nrow(dj), ]
dj_without_last_row$date[dj_without_last_row$next_month_in_dj !=
  dj_without_last_row$next_month_in_calendar]
```

```
## [1] "07/30/1914"
```

Here is a **dplyr**-based version for (6) and (7).

```
rows_before_gap <-
  dj_dplyr |>
  mutate(
    year = mdy(date) |> year(),
    next_month_in_dj = lead(month),
    year_of_next_month_in_dj = lead(year),
    next_month_in_calendar = if_else(month < 12, month + 1, 1),
    year_of_next_month_in_calendar = year + (month == 12),
  ) |>
  slice_head(n = nrow(dj_dplyr)) |>
  filter(
```

```
    next_month_in_dj != next_month_in_calendar |
      year_of_next_month_in_dj != year_of_next_month_in_calendar
  )

# Are all months in `dj` in consecutive calendrical order?
nrow(rows_before_gap) == 0
```

```
## [1] FALSE
```

```
rows_before_gap$date
```

```
## [1] "07/30/1914"
```

(7) On 28 June 1914, Archduke Franz Ferdinand of Austria was assassinated by a group of Bosnian nationalists. This event unleashed hostilities between Austria-Hungary and Serbia that eventually led to the start of World War 1 in July 1914. The New York Stock Exchange closed on 31 July 1914 to prevent a crash. Here is a quote from Noble (1915), the president of the New York Stock Exchange from 1914 to 1919:

> The fundamental reason for closing the Exchange was that America, when the war broke out, was in debt to Europe, and that Europe was sure to enforce the immediate payment of that debt in order to put herself in funds to prosecute this greatest of all wars. To use an illustration popular in Wall Street at the time, there was to be an unexpected run on Uncle Sam's Bank and the Stock Exchange was the paying teller's window through which the money was to be drawn out, so the window was closed to gain time.

(8) Here is a base-R version:

```
dj$close_previous <- lag(dj$close_value)
dj$rel_change_pct <-
  100 * (dj$close_value - dj$close_previous) / dj$close_previous

# Remove columns that are no longer needed
dj <- dj[c("date", "month", "close_value", "rel_change_pct")]
head(dj)
```

```
## # A tibble: 6 x 4
##   date       month close_value rel_change_pct
##   <chr>      <dbl>       <dbl>          <dbl>
## 1 05/29/1896     5        40.6         NA
## 2 06/30/1896     6        36.2        -11.0
## 3 07/31/1896     7        32.0        -11.4
## 4 08/31/1896     8        32.0         -0.156
## 5 09/30/1896     9        36.0         12.8
## 6 10/31/1896    10        39.5          9.65
```

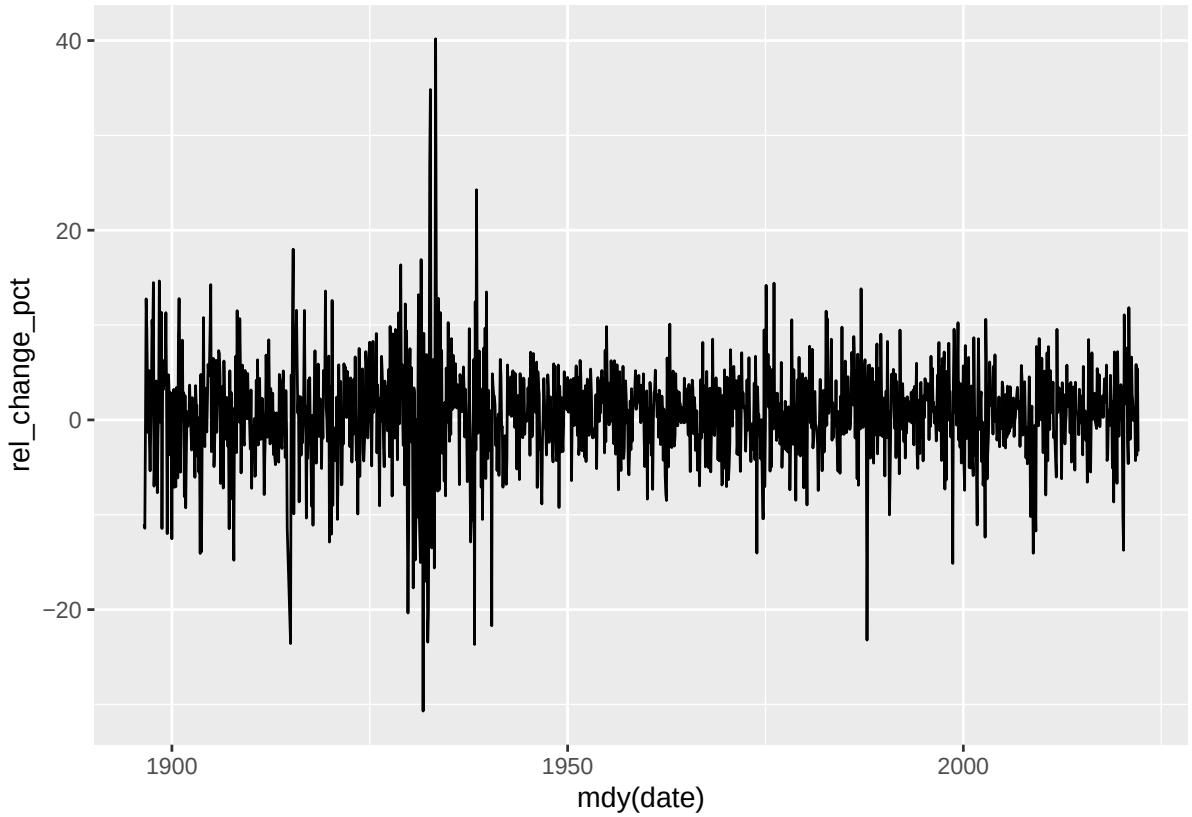
And here is a **dplyr**-based version:

```
dj_dplyr <- dj_dplyr |>
  mutate(
    close_previous = lag(close_value),
    rel_change_pct = 100 * (close_value - close_previous) / close_previous
  ) |>
  select(date, month, close_value, rel_change_pct)
head(dj_dplyr)
```

```
## # A tibble: 6 x 4
##   date       month close_value rel_change_pct
```

```
##   <chr>      <dbl>       <dbl>          <dbl>
## 1 05/29/1896     5        40.6         NA
## 2 06/30/1896     6        36.2        -11.0
## 3 07/31/1896     7        32.0        -11.4
## 4 08/31/1896     8        32.0         -0.156
## 5 09/30/1896     9        36.0         12.8
## 6 10/31/1896    10        39.5          9.65
```

(9)
```
ggplot(dj[-1, ], aes(mdy(date), rel_change_pct)) +
  geom_line()
```



(10) Here is a base-R option:

```
dj$date[which.max(dj$rel_change_pct)]
```

```
## [1] "04/29/1933"
```

And here is a **dplyr**-based option:

```
dj_dplyr |>
  slice_max(rel_change_pct, n = 1) |>
  pull(date)
```

```
## [1] "04/29/1933"
```

We conclude that the largest monthly increase in the history of the Dow Jones happened in April 1933. Franklin D. Roosevelt was inaugurated as US president in March 1933, when the Great Depression had reached its worst phase. Roosevelt immediately enacted policies aimed at economic reform. The abandonment of the gold standard in April 1933 caused investors to buy stocks, spurring a historical rally.

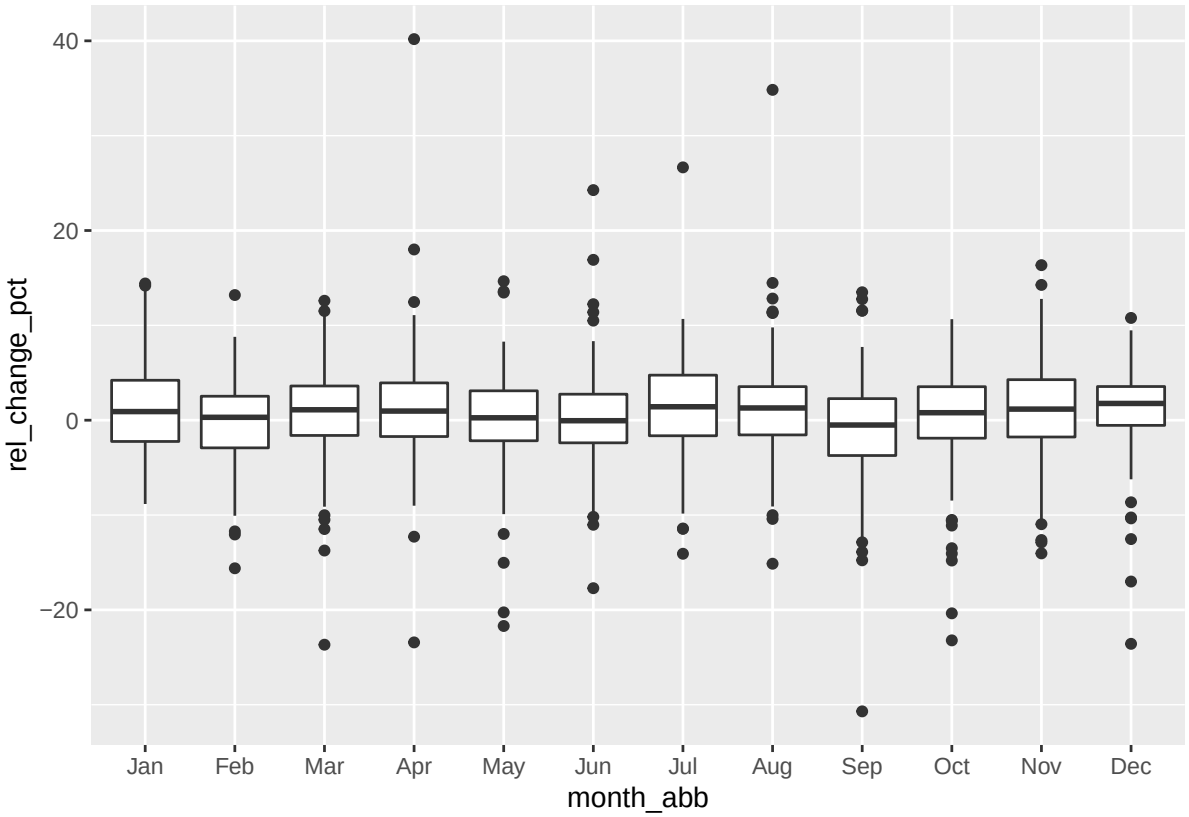
(11) Here is a base-R version:

```
dj$month_abb <-
  factor(month.abb[dj$month], levels = month.abb)
```

And here is a **dplyr**-based version:

```
dj_dplyr <- dj_dplyr |>
  mutate(month_abb = factor(month.abb[month], levels = month.abb))
```

(12)
```
ggplot(dj[-1, ], aes(month_abb, rel_change_pct)) +
  geom_boxplot()
```



Many months exhibit outliers; thus, we should not conduct a test that assumes that relative changes are normally distributed. Differences between the monthly medians are smaller than the typical monthly interquartile range. Therefore, an investment strategy based on buying and selling in specific months is unlikely to be practical. However, there are many data points for each month; hence, the effect may still be statistically significant. In (14), we conduct further tests to determine statistical significance.

(13)
```
agg <- aggregate(rel_change_pct ~ month_abb, data = dj, FUN = median)

# Month with the maximum median relative change
agg$month_abb[which.max(agg$rel_change_pct)] |> as.character()
```

```
## [1] "Dec"
```

```
# Month with the minimum median relative change
agg$month_abb[which.min(agg$rel_change_pct)] |> as.character()
```

```
## [1] "Sep"
```

We conclude that Decemeber has the best median performance, September the worst.

Here is an alternative **dplyr**-based approach:

```
agg_dplyr <-
  dj_dplyr |>
  group_by(month_abb) |>
  summarise(median = median(rel_change_pct, na.rm = TRUE))

# Month with the maximum median relative change
slice_max(agg_dplyr, median, n = 1) |> pull(month_abb) |> as.character()
```

```
## [1] "Dec"
```

```
# Month with the minimum median relative change
slice_min(agg_dplyr, median, n = 1) |> pull(month_abb) |> as.character()
```

```
## [1] "Sep"
```

(14)
```
kruskal.test(rel_change_pct ~ month, data = dj)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  rel_change_pct by month
## Kruskal-Wallis chi-squared = 31.371, df = 11, p-value = 0.0009616
```

The small p-value indicates that there are differences between the months. However, the best time to buy appears to be at the end of September and the best time to sell is at the end of December. Thus, our simple analysis casts doubts on whether the sell-in-May strategy is optimal.

References

Noble, H. G. S. (1915). *The New York Stock Exchange in the Crisis of 1914*. Country Life Press, Garden City.