Reproducible Research: Peer Assessment 1

sekargaluh

2024-12-30

Loading and preprocessing the data

```
library(readr)
activity <- read_csv("Z:/Onderzoekers/Galuh/Academic/Courses/Coursera/activity.csv")
```

```
## Rows: 17568 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (2): steps, interval
## date (1): date
##
## 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.
```

```
head(activity)
```

```
## # A tibble: 6 x 3
##   steps date       interval
##   <dbl> <date>        <dbl>
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

What is mean total number of steps taken per day?

Here we calculate the total number, mean, and median for each day. The file is sorted based on 'date' column 2.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

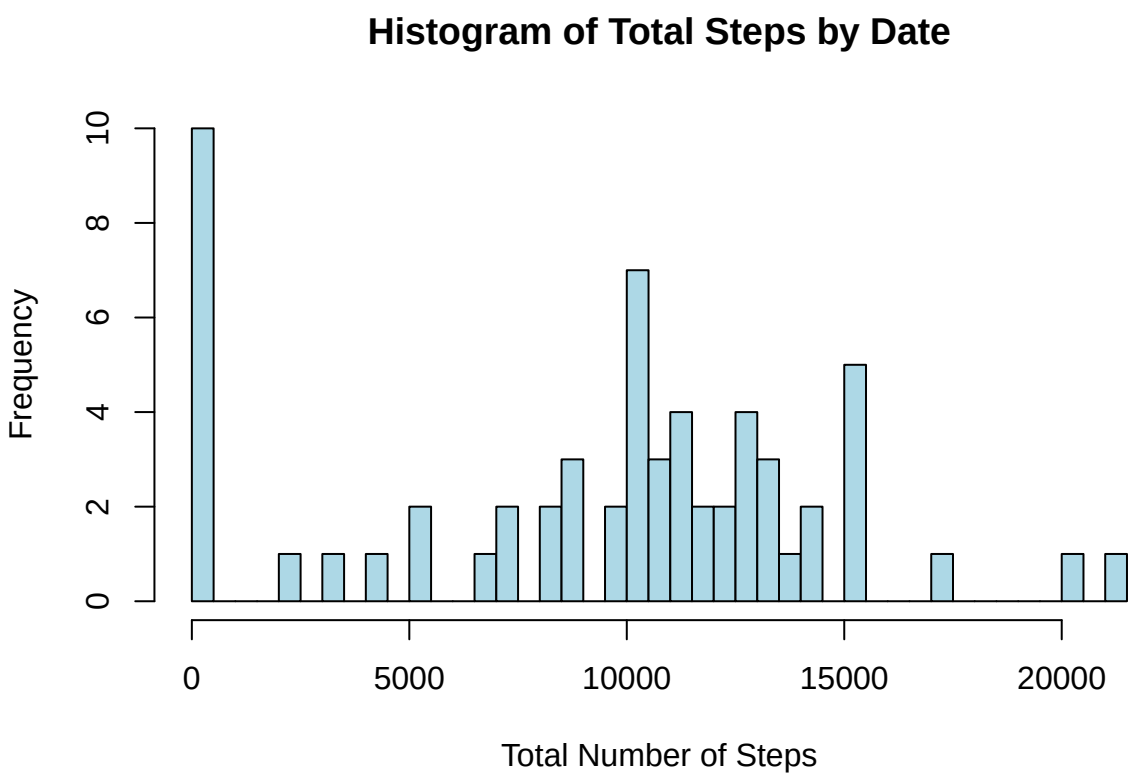
```
library(readr)

# Group by Date and calculate Total, Mean, and Median for column 2
daily_stats <- activity %>%
               group_by(date) %>%
               summarize(Total = sum(steps, na.rm = TRUE)) %>% # Total for each day
               ungroup() # Ungroup after summarizing

# View the result
print(daily_stats)
```

```
## # A tibble: 61 x 2
##    date       Total
##    <date>     <dbl>
##  1 2012-10-01     0
##  2 2012-10-02   126
##  3 2012-10-03 11352
##  4 2012-10-04 12116
##  5 2012-10-05 13294
##  6 2012-10-06 15420
##  7 2012-10-07 11015
##  8 2012-10-08     0
##  9 2012-10-09 12811
## 10 2012-10-10  9900
## # i 51 more rows
```

```
#plot the histogram
hist(daily_stats$Total,
     main = "Histogram of Total Steps by Date",
     xlab = "Total Number of Steps",
     ylab = "Frequency",
     col = "lightblue",
     border = "black",
     breaks = 50
)
```




```
# Calculate and print the mean and median of the total steps
mean_steps <- mean(daily_stats$Total, na.rm = TRUE)
median_steps <- median(daily_stats$Total, na.rm = TRUE)

# Print the results
cat("Mean of Total Steps: ", round(mean_steps, 2), "\n")
```

```
## Mean of Total Steps:  9354.23
```

```
cat("Median of Total Steps: ", round(median_steps, 2), "\n")
```

```
## Median of Total Steps:  10395
```

What is the average daily activity pattern?

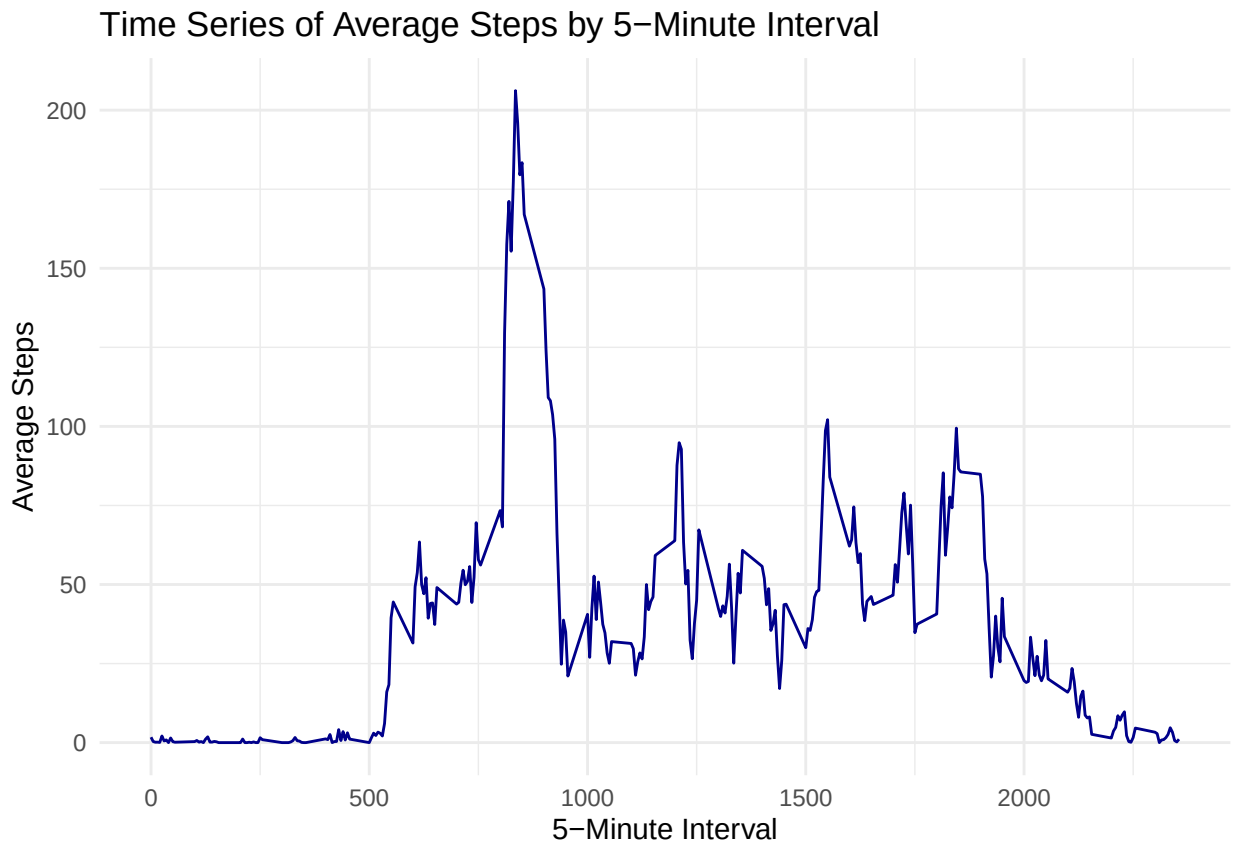
Here we calculate the daily steps by grouping based on interval

```
library(dplyr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```r
# Group by interval and calculate the mean steps for each 5-minute interval
average_steps <- activity %>%
                 group_by(interval) %>%
                 summarize(average_steps = mean(steps, na.rm = TRUE))

# Plotting the time series
ggplot(average_steps, aes(x = interval, y = average_steps)) +
      geom_line(color = "blue4") + # Line plot
      labs(
          title = "Time Series of Average Steps by 5-Minute Interval",
          x = "5-Minute Interval",
          y = "Average Steps"
          ) +
      theme_minimal()
```




```r
# Find the interval with the maximum average steps
max_interval <- average_steps %>%
  slice_max(order_by = average_steps, n = 1)

# Display the result
max_interval
```

```
## # A tibble: 1 x 2
##   interval average_steps
##      <dbl>         <dbl>
```

```
## 1      835          206.
```

Imputing missing values

There are missing values in the dataset. The missing values are filled with average of 5-minute interval that day. Here, we compare the first analysis which contains missing values and after the missing values were replaced, or so called 'filled'. In this case, filling the missing value change the distribution of the dataset, the dataset becomes distributed normally.

```
# Calculate the total number of missing values (NA)
total_missing <- sum(is.na(activity$steps))

# Report the total number of missing values
cat("Total number of missing values in the dataset: ", total_missing, "\n")
```

```
## Total number of missing values in the dataset:  2304
```

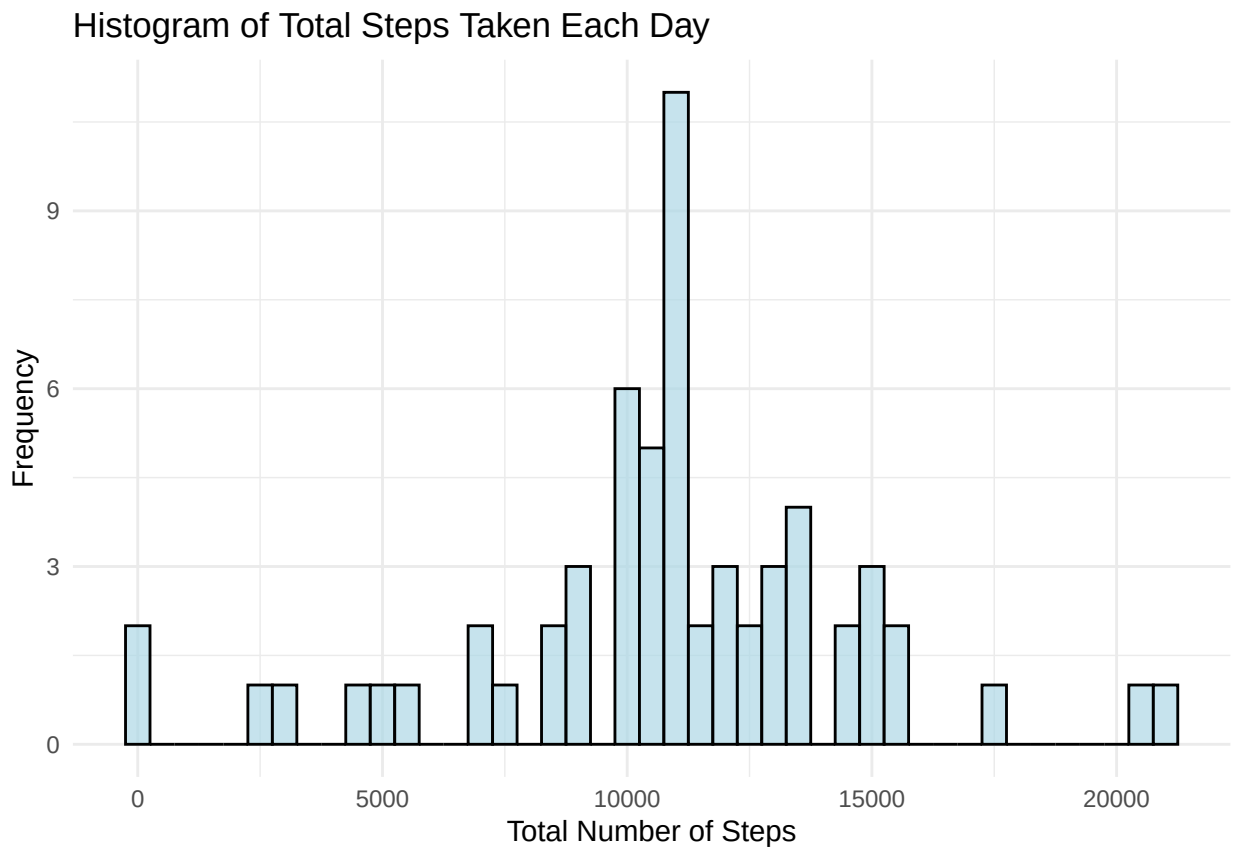
```
# We have already calculated the mean number of steps for each 5-minute interval across all days
# Join the mean values back to the original dataset
activity_filled <- activity %>%
  left_join(average_steps, by = "interval") %>%
  mutate(steps = ifelse(is.na(steps), average_steps, steps)) %>%
  select(-average_steps)  # Remove the 'average_steps' column after filling

# Display the first few rows of the updated dataset
head(activity_filled)
```

```
## # A tibble: 6 x 3
##    steps date       interval
##    <dbl> <date>        <dbl>
## 1 1.72   2012-10-01        0
## 2 0.340  2012-10-01        5
## 3 0.132  2012-10-01       10
## 4 0.151  2012-10-01       15
## 5 0.0755 2012-10-01       20
## 6 2.09   2012-10-01       25
```

```
# Calculate the total number of steps taken per day (sum of steps for each date)
total_daily_steps <- activity_filled %>%
  group_by(date) %>%
  summarize(total_steps = sum(steps, na.rm = TRUE))

# Create a histogram of total number of steps per day
ggplot(total_daily_steps, aes(x = total_steps)) +
  geom_histogram(binwidth = 500, fill = "lightblue", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Total Steps Taken Each Day",
       x = "Total Number of Steps",
       y = "Frequency") +
  theme_minimal()
```




```r
# Calculate and report the mean and median total steps per day
mean_steps <- mean(total_daily_steps$total_steps)
median_steps <- median(total_daily_steps$total_steps)

cat("Mean total steps per day (after imputing missing data):", mean_steps, "\n")
```

```
## Mean total steps per day (after imputing missing data): 10766.19
```

```r
cat("Median total steps per day (after imputing missing data):", median_steps, "\n")
```

```
## Median total steps per day (after imputing missing data): 10766.19
```

```r
# Step 5: Compare with the estimates from the first part of the assignment
# In the first part, we calculated total steps per day (without imputing missing data)
# Assuming we already have the `daily_stats` dataframe calculated earlier
mean_steps_ori <- mean(daily_stats$Total)
median_steps_ori <- median(daily_stats$Total)

cat("Mean total steps per day (before imputing missing data):", mean_steps_ori, "\n")
```

```
## Mean total steps per day (before imputing missing data): 9354.23
```

```
cat("Median total steps per day (before imputing missing data):", median_steps_ori, "\n")
```

```
## Median total steps per day (before imputing missing data): 10395
```

Are there differences in activity patterns between weekdays and weekends?

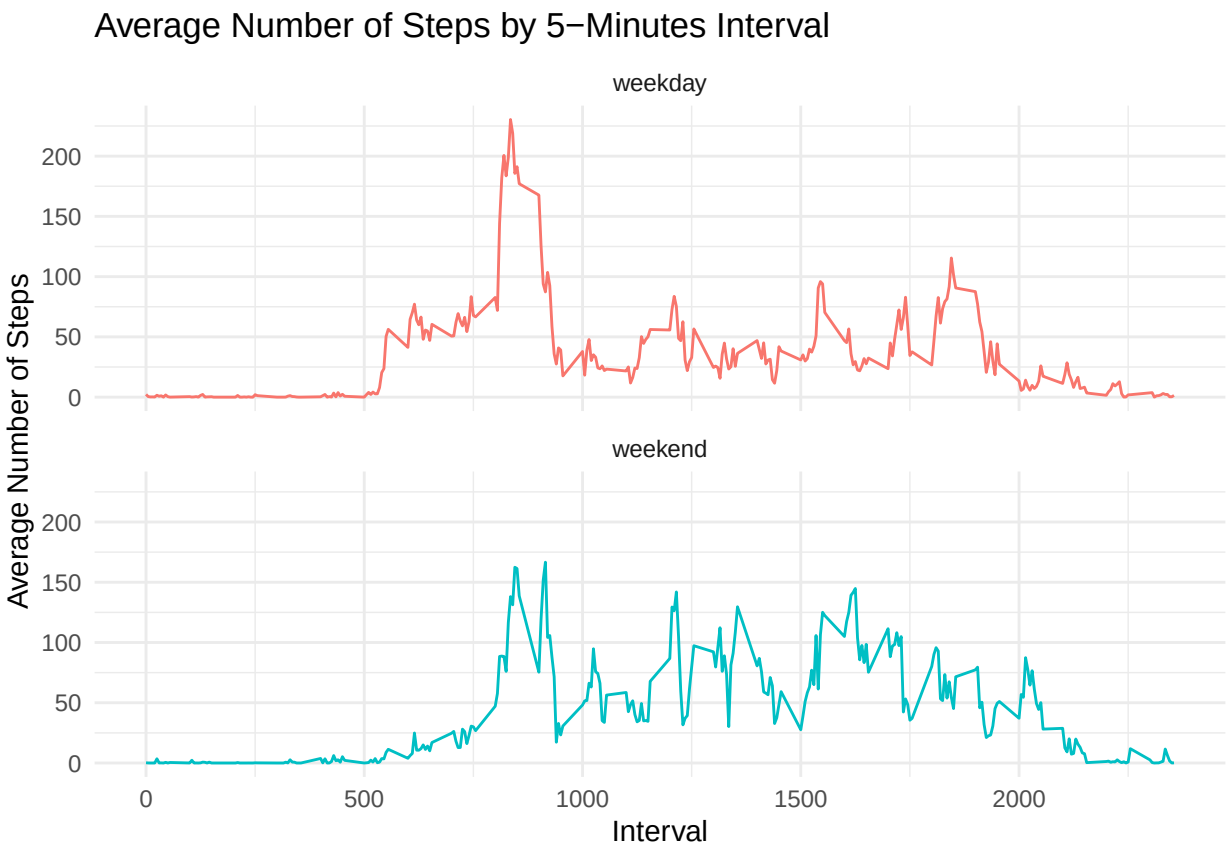
```
library(dplyr)
library(ggplot2)

#Create a new factor variable for "weekday" and "weekend"
activity_filled_byday <- activity_filled %>%
  mutate(day_of_week = weekdays(as.Date(date))) %>%
  mutate(type_of_day = ifelse(day_of_week %in% c("zaterdag", "zondag"), "weekend", "weekday"))

# Calculate the average number of steps for each 5-minute interval and type of day
avg_steps_byday <- activity_filled_byday %>%
  group_by(interval, type_of_day) %>%
  summarize(avg_steps = mean(steps, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'interval'. You can override using the
## `.groups` argument.
```

```
# Create the panel plot (time series plot) of average steps
ggplot(avg_steps_byday, aes(x = interval, y = avg_steps, color = type_of_day)) +
  geom_line() +  # Add a line for each group
  facet_wrap(~type_of_day, ncol = 1) +  # Create separate panels for "weekday" and "weekend"
  labs(title = "Average Number of Steps by 5-Minutes Interval",
       x = "Interval",
       y = "Average Number of Steps",
       color = "Day Type") +
  theme_minimal() +  # Use a minimal theme for better clarity
  theme(legend.position = "none")  # Remove legend, since we already have labels
```


Average Number of Steps by 5−Minutes Interval
weekday
weekend
Average Number of Steps
0
50
100
150
200
0
500
1000
1500
2000
Interval