Reproducible Research: Peer Assessment 1

Walking dataset

Santiago Botero Sierra

2019/07/07

```
knitr::opts_chunk$set(message = FALSE, warning = FALSE)
```

Introduction

This report uses data obtained from an anonymous subject whose daily activity has been tracked in Otober and November, 2012. The dataset includes the steps taken by this subject in 5 minutes time-spans.

We are interested in analyzing the total steps taken by the subject, visualizing if there was any activity pattern in those months, and looking for differences on activity between week-days and weekend-days.

Software

We used the statistical program R version 3.5.2 (2018-12-20) (R Core Team 2018). We used the following packages loaded in `R`:

- `knitr` (Xie 2018; Xie 2015; Xie 2014).
- `dplyr` (Wickham et al. 2019).
- `tidyr` (Wickham and Henry 2018).
- `ggplot2` (Wickham 2016).
- `lubridate` (Grolemund and Wickham 2011).
- `here` (Müller 2017).

Data

We used the walking dataset included with the Reproducible Research Course, available here. In the cleaning phase, we noted that each reported day has 288 observations, which correponds exactly with the 288 five minutes time-spans in a given day, so we transformed the interval identificator in a sequence in each day, we generated a variable called `time` which identifies the beggining time of each interval, and we unified the information available in the variables `date`and `time` in a variable called `daytime`, reflecting the initial time of each time-span.

```
# Unziping the dataset
if (!dir.exists(here(".", "data"))) {dir.create(here(".", "data"))}
unzip(here(".", "activity.zip"), exdir = here(".", "data"))

# Loading and tidying the dataset
walking <- read.csv(here(".", "data", "activity.csv")) %>%
  tbl_df %>%
  group_by(date) %>%
```

```
    mutate(interval = row_number(interval) - 1) %>%
    ungroup() %>%
    mutate(date = ymd(date),
           time = seconds_to_period(interval * 60 * 5),
           daytime = ymd_hms(date + time)) %>%
    group_by(date)
```

Analysis

What is the mean total number of steps taken per day?

```
total_steps <- walking %>%
  summarize(total = sum(steps, na.rm = TRUE))
```

In the following plot is depicted the number of days in which the individual walked each number of steps. It's interesting to note that the mean value (vertical black line) is in a bin just observed in one day, and corresponds to 9354.23 steps. Each day, this subject walked between 0 and 21194 steps.

```
# Create figure/ directory
if (!dir.exists(here(".", "figure"))) {dir.create(here(".", "figure"))}

# Export the graph
png(here(".", "figure", "mean_steps_histo.png"))
ggplot(total_steps, aes(total)) +
  geom_histogram() +
  geom_vline(aes(xintercept = mean(total, na.rm = TRUE))) +
  labs(title = "Total number of steps taken per day",
       x = "Number of steps", y = "Number of days") +
  theme_bw()
dev.off()
```

```
## pdf
##   2
```

What is the average daily activity pattern?

```
statistics <- total_steps %>%
  group_by(date) %>%
  summarize(mean = mean(total, na.rm = TRUE),
            median = median(total, na.rm = TRUE))
```

We take into account different summary statistics on the total daily walking steps, shown in the next table. We noted that the *median* and the *mean* steps were the same in each day of the months observed.[1]

[1] However, when taking the analyzing the *mean* and *median between* days, as opossed to *within* days, we obtained different values.

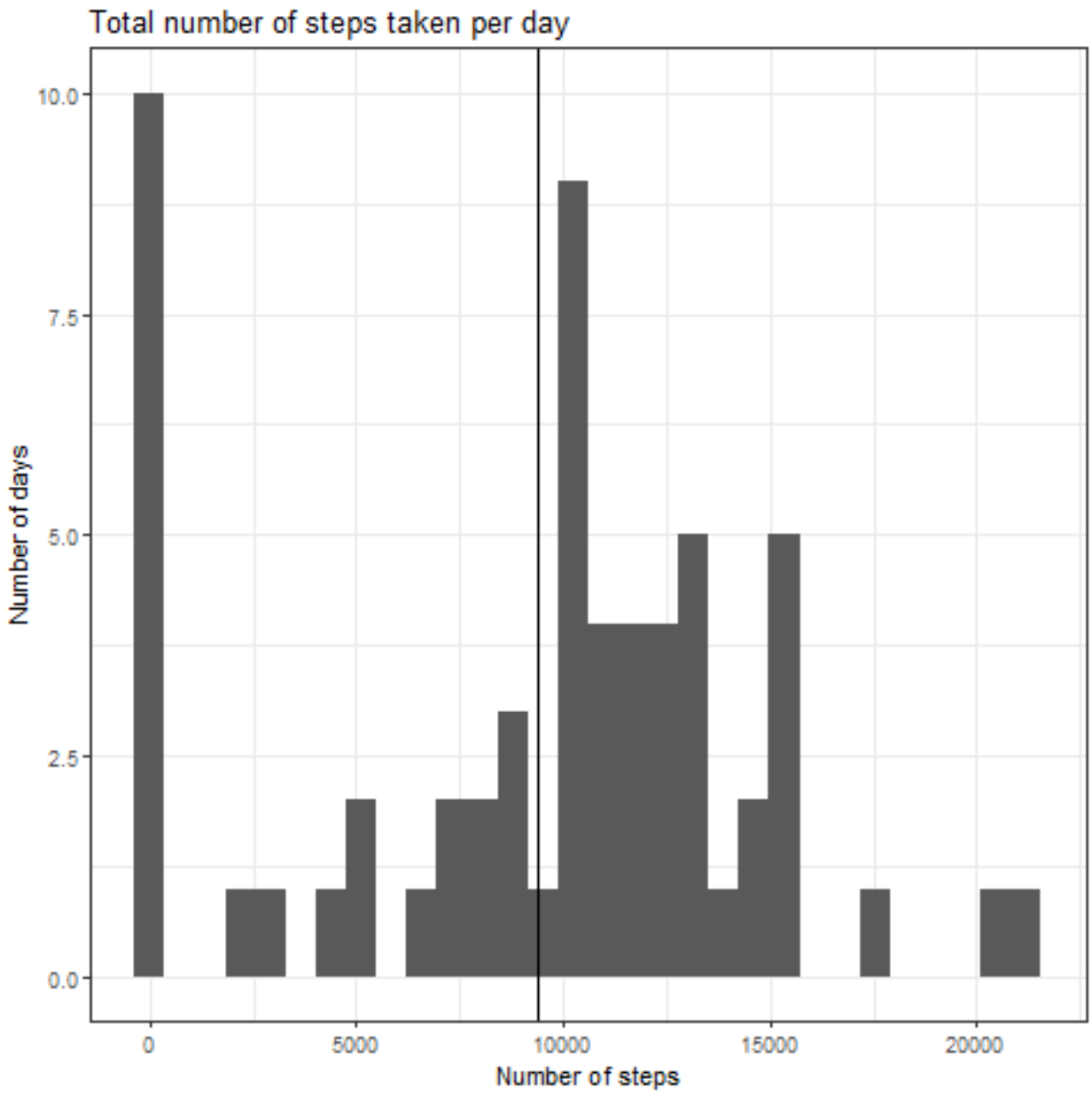

Figure 1: Mean steps histogram

```
  print(paste("The mean and the median differed in",
              sum(with(statistics, mean != median)), "days."))
```

```
## [1] "The mean and the median differed in 0 days."
```

```
  summary(total_steps$total) %>%
    unclass %>% as.data.frame %>%
    tibble::rownames_to_column() %>%
    knitr::kable(col.names = c("Statistic", "Value"),
                 caption = "Summary statistics.")
```

Table 1: Summary statistics.

Statistic	Value
Min.	0.00
1st Qu.	6778.00
Median	10395.00
Mean	9354.23
3rd Qu.	12811.00
Max.	21194.00

In the following graph we ploted the daily mean of total steps during the period.

```
  # Export graph
  png(here(".", "figure", "mean_steps_line.png"))
  ggplot(statistics, aes(x = date, y = mean)) +
    geom_line() +
    labs(title = "Mean steps taken each day",
         x = "Date", y = "Mean steps",
         caption = "Note: Mean and median steps taken each day are the same.") +
    theme_bw()
  dev.off()
```

```
## pdf
##   2
```

In the following graph we observe the average number of steps taken in each 5-minute time-span by the subject. Here, we can observe that, on average, from 00:00 to 05:00 this individual do not use to walk, and that the most intensive walking activity take place aproximately at 08:00.

```
  # Compute maximum steps
  max_steps <- walking %>%
    ungroup() %>%
    group_by(interval) %>%
    summarize(mean = mean(steps, na.rm = TRUE)) %>%
    mutate(time = seconds_to_period(interval * 5 * 60))
  # Export graph
  png(here(".", "figure", "max_steps_line.png"))
  ggplot(max_steps, aes(x = time, y = mean)) +
    geom_line() +
```

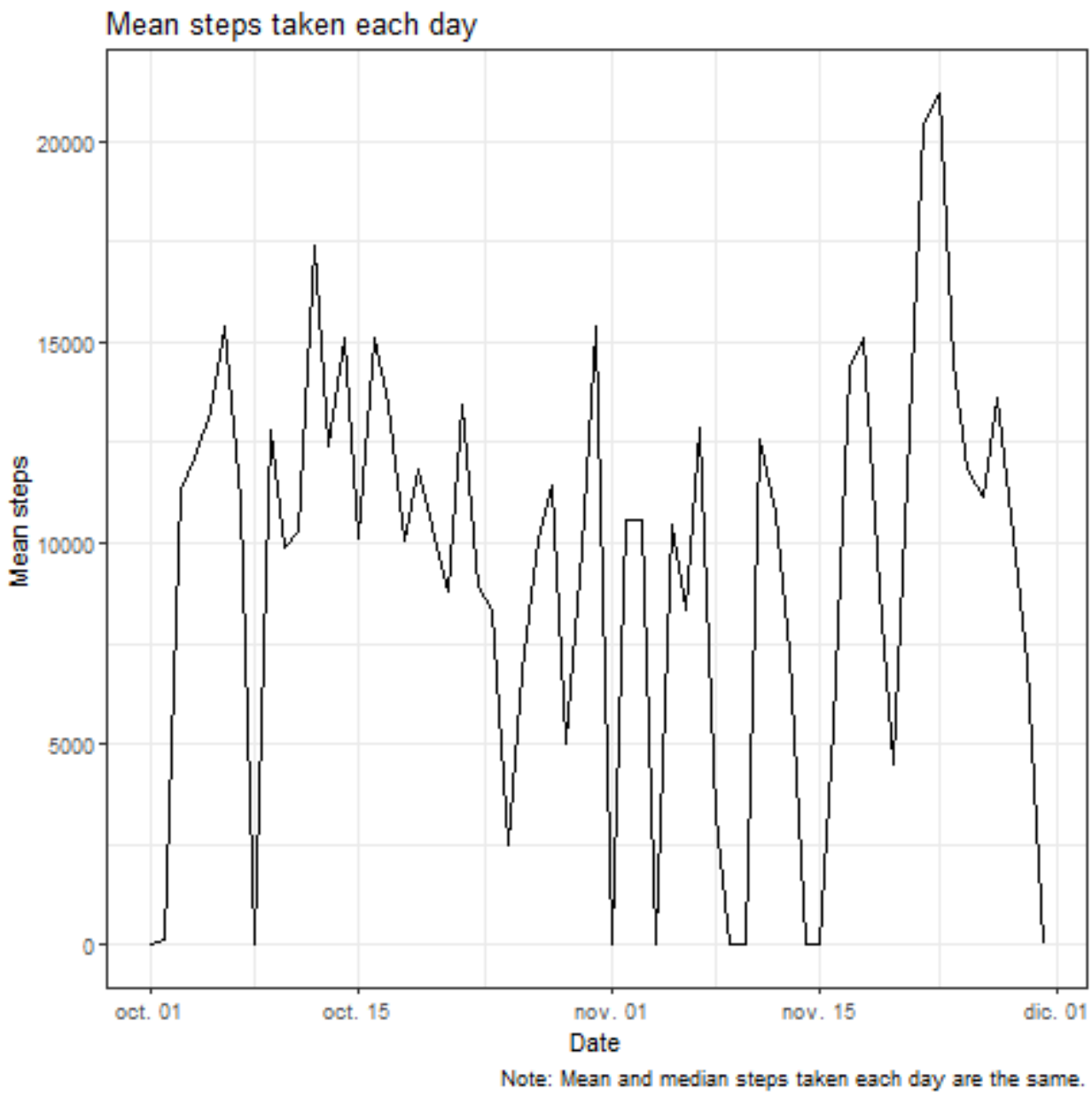

Figure 2: Daily mean steps

```
    theme_bw() +
    scale_x_time(breaks = seconds_to_period(seq(1, 24, by = 4) * 60^2)) +
    labs(title = "Average steps in a 5-minutes time-span",
         x = "Time of the day", y = "Average steps")
  dev.off()
```

```
## pdf
##   2
```

As a matter of fact, the most active 5-minute time-span, on average, is the one shown in the followig table.

```
  max_steps[which.max(max_steps$mean), 2:3] %>%
    knitr::kable(col.names = c("Steps", "Time"),
                 caption = "Maximum average number of steps.")
```

Table 2: Maximum average number of steps.

Steps	Time
206.1698	8H 35M 0S

Imputing missing values

In the following plot, we observed that the missing values are not related with the time of meassurement, since the missing values represent 13%, regardless of the time or the level of activity.

```
  # Compute missings
  missing <- walking %>%
    ungroup() %>%
    group_by(interval) %>%
    summarize(mean_steps = mean(steps, na.rm = TRUE),
              prop_missing = mean(is.na(steps)) * 100) %>%
    gather(key = "measure", value = "value", mean_steps, prop_missing) %>%
    mutate(time = seconds_to_period(interval * 5 * 60))
  # Export graph
  png(here(".", "figure", "missing_line.png"))
  ggplot(missing, aes(x = time, y = value)) +
    geom_line() +
    facet_grid(measure ~ ., scales = "free") +
    theme_bw() +
    scale_x_time(breaks = seconds_to_period(seq(1, 24, by = 4) * 60^2)) +
    labs(title = "Average steps in a 5-minutes time-span and missing proportion",
         x = "Time of the day")
  dev.off()
```

```
## pdf
##   2
```

We continued analyzing why the proportion of missing values were the same across all time-spans in the period, and noted that all missing values were in specific days, and that all the registries of these days were

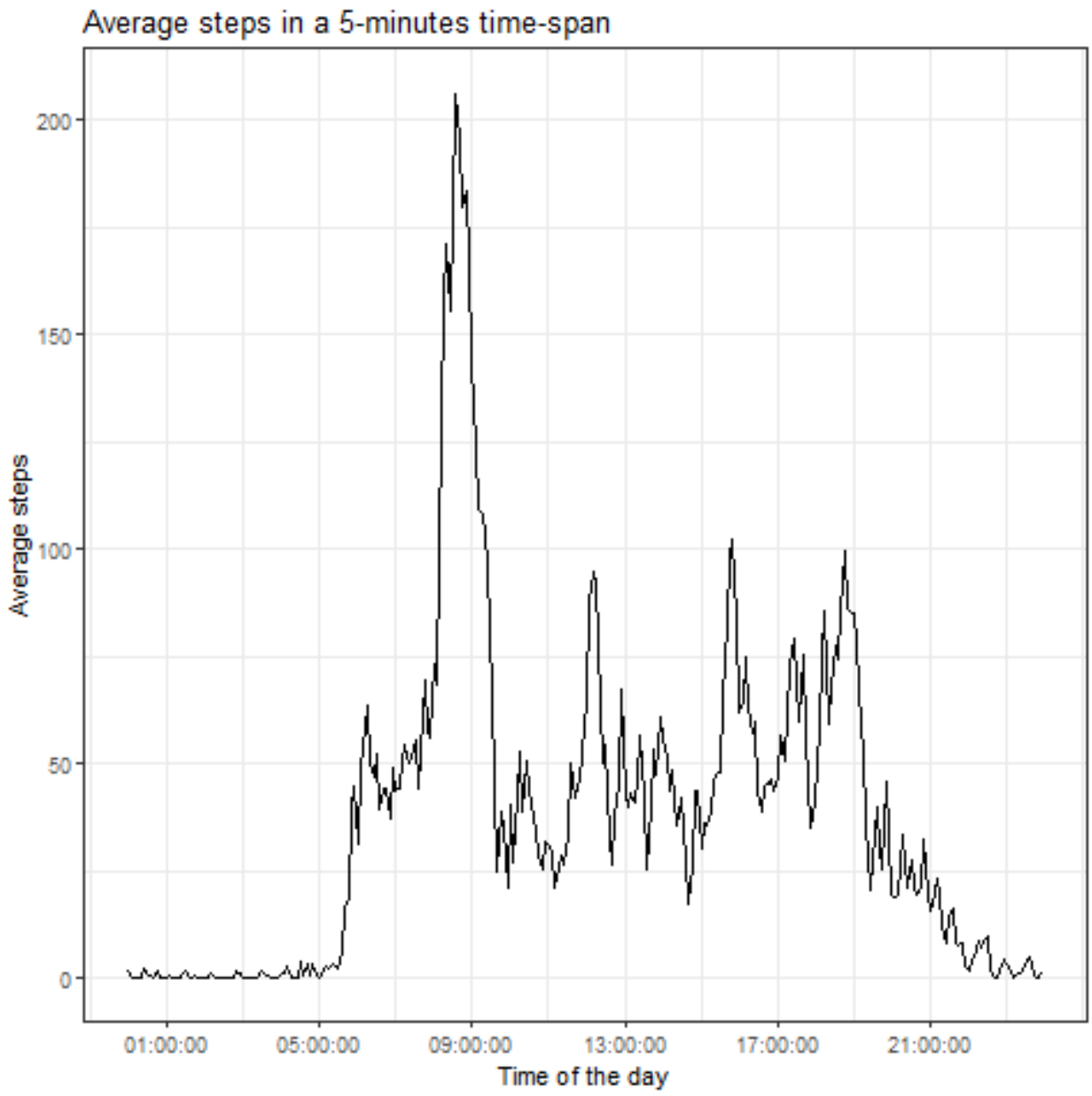

Figure 3: Time-span average steps

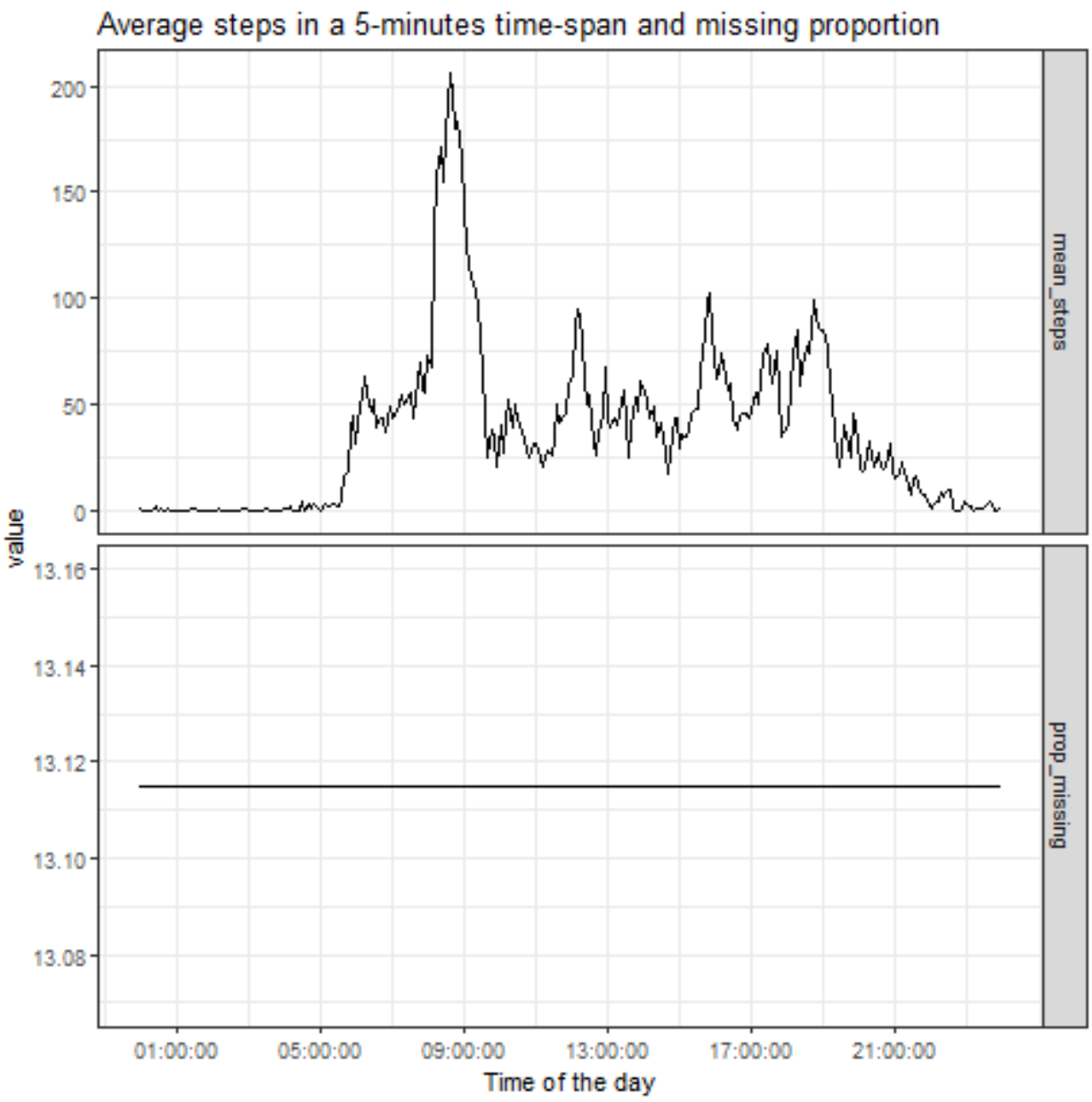

Figure 4: Missing values

missing. This information is shown in the following table. Note that there was 2304 missing values in the database.

```
walking %>% filter(is.na(steps)) %>% select(date) %>% table %>% tbl_df %>%
  knitr::kable(col.names = c("Date", "Missing values"),
               caption = "Missing values, by date")
```

Table 3: Missing values, by date

Date	Missing values
2012-10-01	288
2012-10-08	288
2012-11-01	288
2012-11-04	288
2012-11-09	288
2012-11-10	288
2012-11-14	288
2012-11-30	288

Therefore, our imputing strategy is replacing the missing values with the time-span average.

```
replace <- walking %>%
  ungroup %>%
  group_by(interval) %>%
  summarize(steps2 = mean(steps, na.rm = TRUE))
walking2 <- merge(walking, replace) %>%
  tbl_df %>%
  mutate(steps = ifelse(is.na(steps), steps2, steps)) %>%
  select(-steps2) %>%
  arrange(daytime)
```

Bellow, is shown a histogram with the steps after imputing missing values.

```
# Compute total_steps
total_steps2 <- walking2 %>%
  group_by(date) %>%
  summarize(total = sum(steps))
# Export graph
png(here(".", "figure", "mean_steps_histo2.png"))
ggplot(total_steps2, aes(total)) +
  geom_histogram() +
  geom_vline(aes(xintercept = mean(total, na.rm = TRUE))) +
  labs(title = "Total number of steps taken per day",
       x = "Number of steps", y = "Number of days",
       caption = "Note: Missing values imputed as time-spans averages") +
  theme_bw()
dev.off()
```

```
## pdf
##   2
```

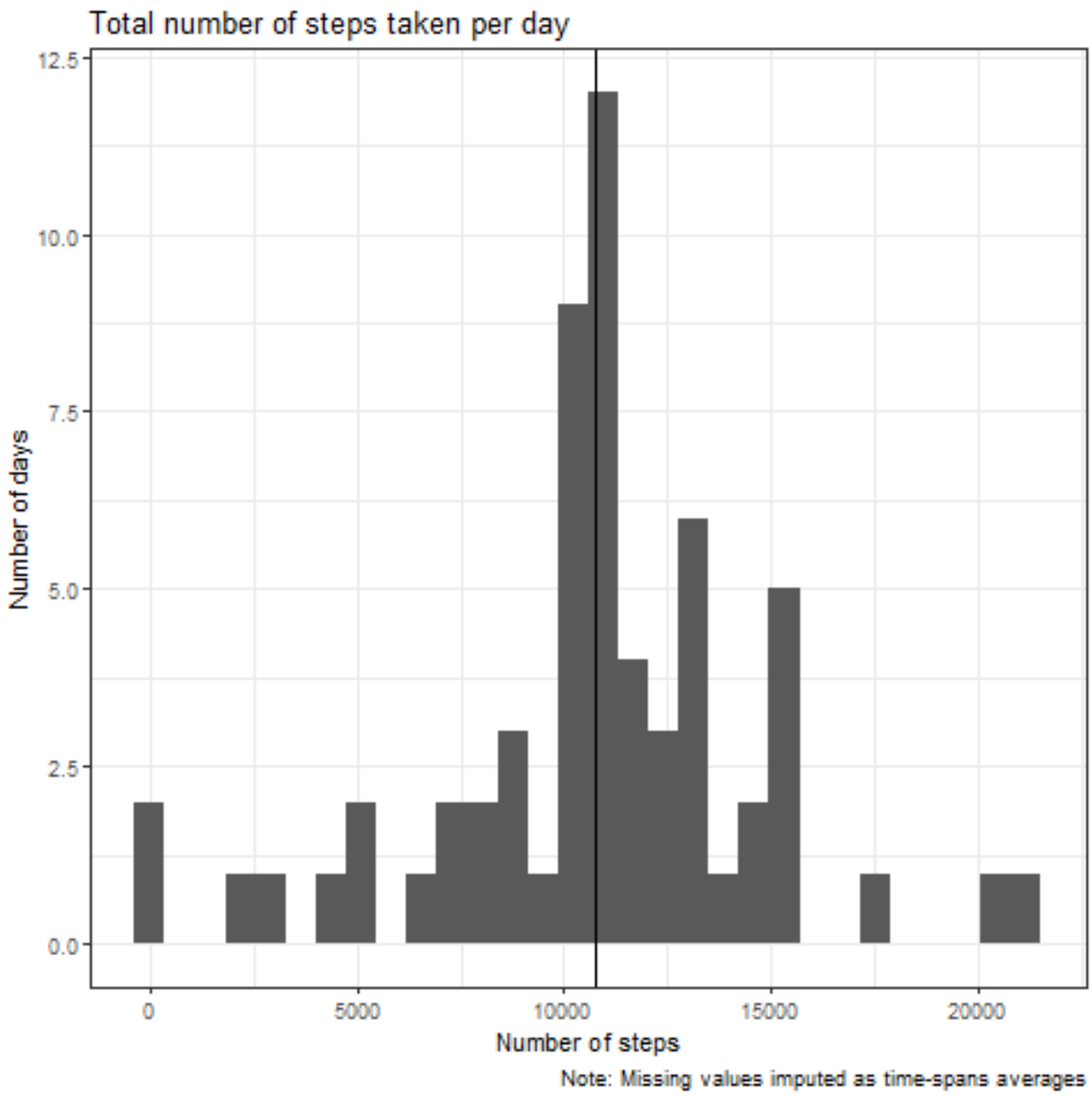

Figure 5: Mean steps histogram (imputed)

Note that the strategy of imputting average time-span values to replace missing values does not imply major changes neither in the histogram shown previously, nor in the discriptive statistics, shown bellow. The imputting strategy changed mainly the minimum and first quartile, moderately the mean and median, and did not changed neither the third quartile nor the maximum value.

```
rbind(summary(total_steps2$total), summary(total_steps$total)) %>%
  tbl_df %>%
  mutate(Database = row_number(),
         Database = ifelse(Database == 1, "Imputed", "Original")) %>%
  select(7, 1:6) %>%
  knitr::kable(caption = "Comparisson of summary statistics.")
```

Table 4: Comparisson of summary statistics.

Database	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Imputed	41	9819	10766.19	10766.19	12811	21194
Original	0	6778	10395.00	9354.23	12811	21194

Are there differences in activity patterns between weekdays and weekends?

In the following plot we noted that, on average, weekday and weekend walking intensity is different. We observed that the range of steps is lower in weekends than in weekdays, but the weekend activity has more spikes than the weekday one. One different approach for imputing data, would have been imputing by average time-spans, controlling for differences in the days of the week.

```
# Set locale to a common language
Sys.setlocale(locale = "English")
```

```
## [1] "LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MONETARY=English_U
```

```
# Identifying weekdays and weekends
weekly <- walking2 %>%
  mutate(weekdays = weekdays(date),
         weekdays = ifelse(weekdays %in% c("Saturday", "Sunday"),
                           "Weekend", "Weekday")) %>%
  group_by(weekdays, interval) %>%
  summarize(mean = mean(steps)) %>%
  ungroup %>%
  mutate(time = seconds_to_period(interval * 60 * 5))

# Export graph
png(here(".", "figure", "weekdays.png"))
ggplot(weekly, aes(x = time, y = mean)) +
  geom_line() +
  facet_grid(. ~ weekdays) +
  theme_bw() +
  scale_x_time(breaks = seconds_to_period(seq(1, 24, by = 8) * 60^2)) +
  labs(title = "Average steps in a 5-minutes time-span, by weekends and others",
       x = "Time of the day")
dev.off()
```

```
## pdf
##   2
```

References

Grolemund, Garrett, and Hadley Wickham. 2011. "Dates and Times Made Easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. http://www.jstatsoft.org/v40/i03/.

Müller, Kirill. 2017. *Here: A Simpler Way to Find Your Files*. https://CRAN.R-project.org/package=here.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. http://ggplot2.org.

Wickham, Hadley, and Lionel Henry. 2018. *Tidyr: Easily Tidy Data with 'Spread()' and 'Gather()' Functions*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2019. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

———. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.name/knitr/.

———. 2018. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.name/knitr/.

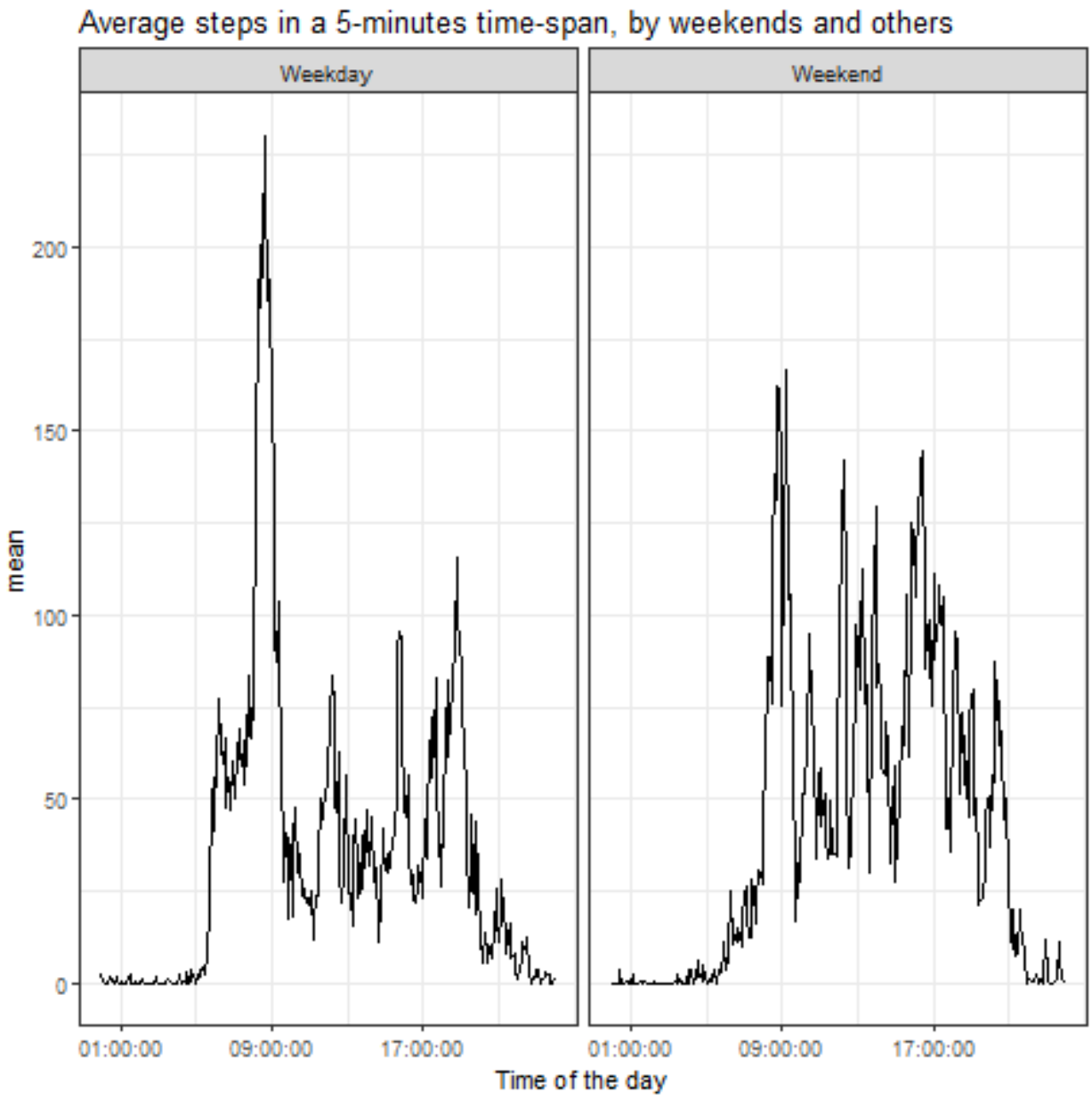

Figure 6: Weekday and weekend activity