Reproducible Research: Peer Assessment 1

Tim Quivooij

30-10-2023

These are the packages you will need

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.4     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflic
ts to become errors
```

Loading and preprocessing the data

```
activity <- read.csv("activity.csv", colClasses =c("numeric","Date", "numeric"))
```

Then look at the data

```
names(activity)
```

```
## [1] "steps"    "date"     "interval"
```

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: num  0 5 10 15 20 25 30 35 40 45 ...
```

```
head(activity)
```

```
##   steps       date interval
## 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 the mean total number of steps taken per day?

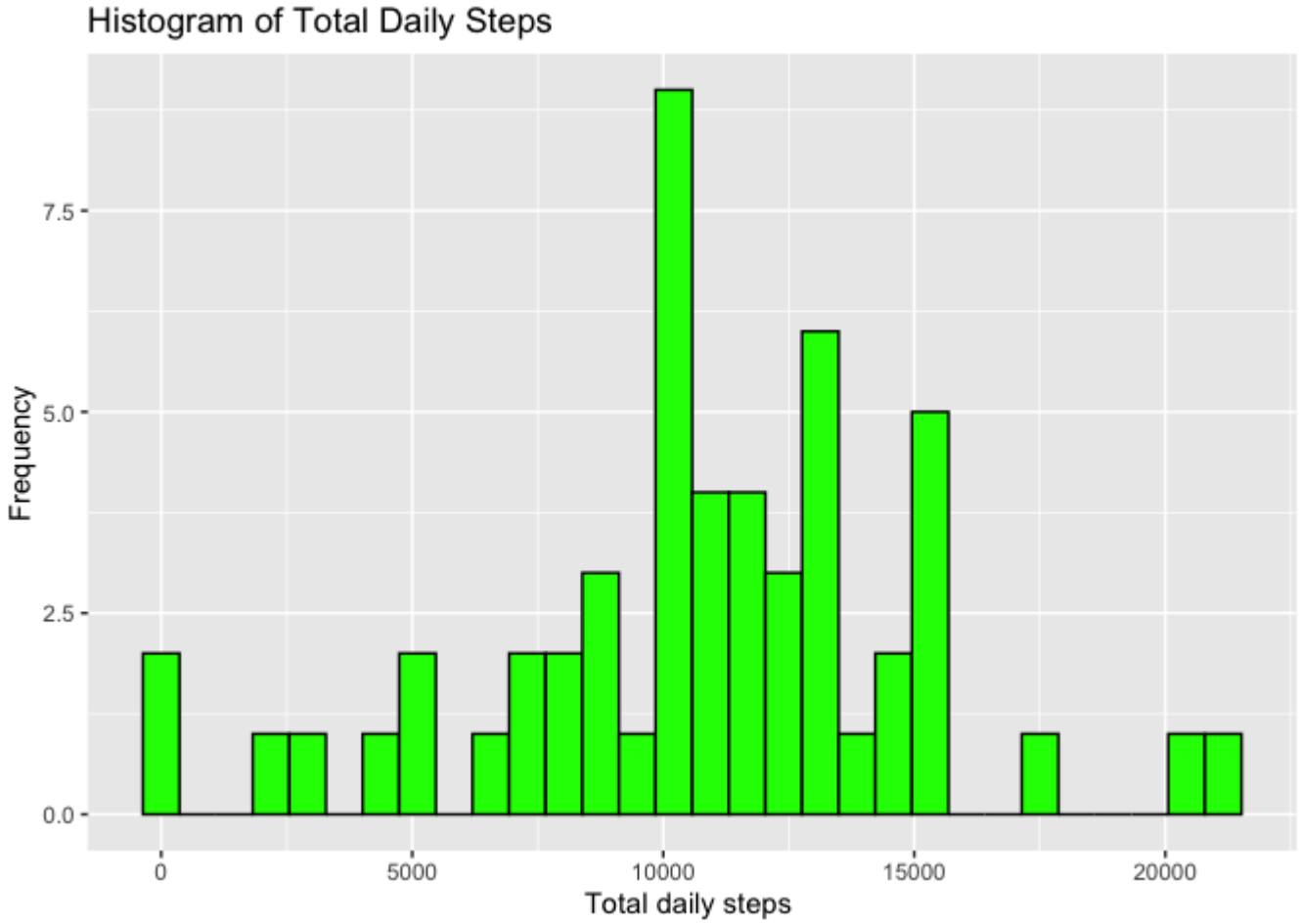
First calculate the total daily steps, omitting NA values

```
daily <- na.omit(activity) %>%
    group_by(date) %>%
    summarise(tot.steps = sum(steps))
```

Plot a histogram

```
ggplot(daily, aes(x=tot.steps), breaks=30) +
    geom_histogram(color="black",fill="green") +
    labs(x= "Total daily steps ") +  # Add labels
    labs(y= "Frequency") +
    labs(title = "Histogram of Total Daily Steps")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Next, calculate the mean and median of the daily steps:

```
dailyMean <- round(mean(daily$tot.steps), 0)
print(dailyMean)
```

```
## [1] 10766
```

```
dailyMedian <- round(median(daily$tot.steps), 0)
print(dailyMedian)
```

```
## [1] 10765
```

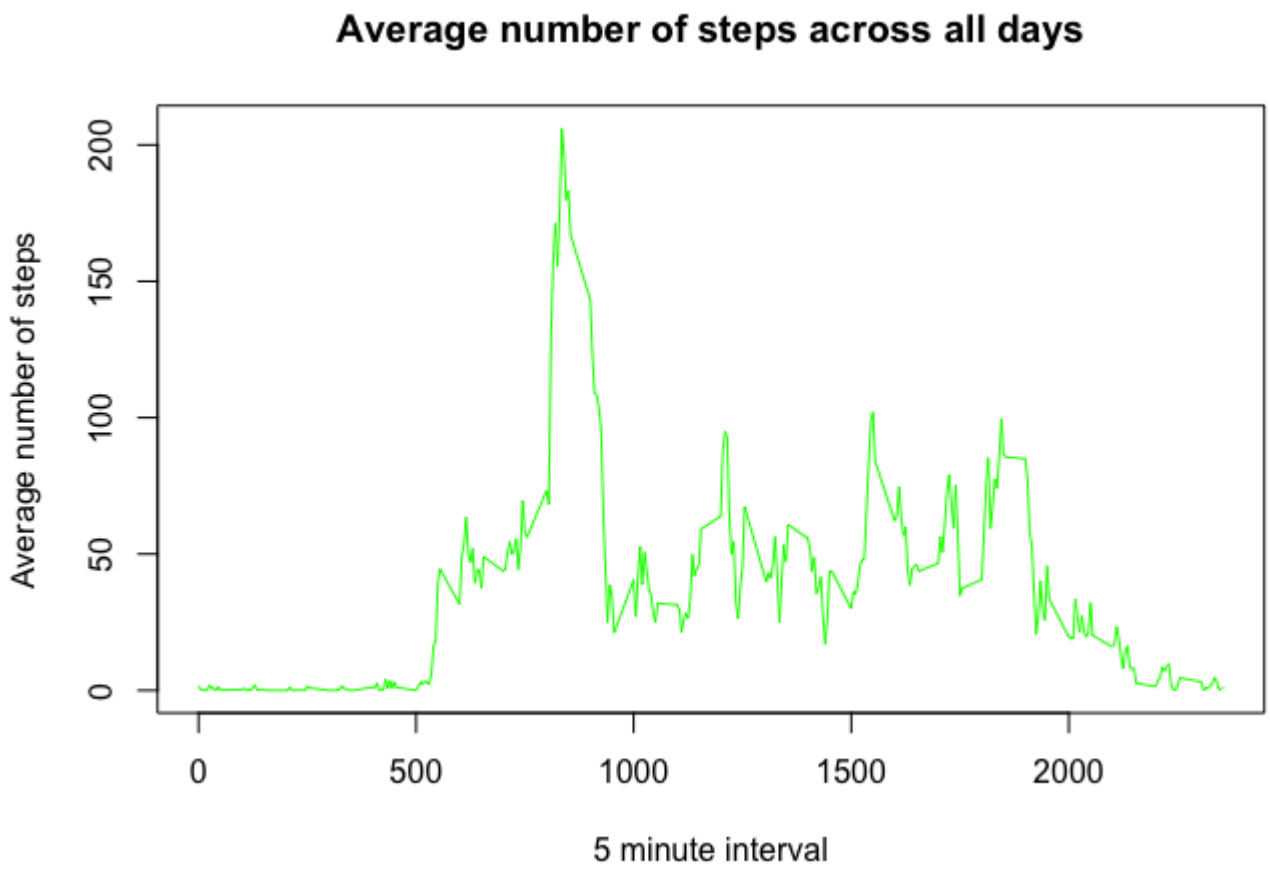
What is the average daily activity pattern?

The average number of steps per 5 minute interval

```
fivemin <- na.omit(activity) %>%
             group_by(interval) %>%
            summarise(avg.steps = mean(steps))
```

A time series plot with type = “l”:

```
plot(fivemin$interval, fivemin$avg.steps, type="l",
    xlab="5 minute interval", ylab = "Average number of steps",
    main = "Average number of steps across all days", col= "green")
```



Daily 5 minute interval with max. average steps:

```
maxfivemin <- fivemin[which.max(fivemin$avg.steps),]
print( maxfivemin)
```

```
## # A tibble: 1 × 2
##   interval avg.steps
##      <dbl>     <dbl>
## 1      835      206.
```

Imputing missing values

The total number of missing values in the dataset

```
nas <- nrow(activity[is.na(activity$steps), ])
print(nas)
```

```
## [1] 2304
```

Fill the missing values with the average of the 5 minute interval.

```
activityNoNA <- merge(x = activity, y = fivemin, by="interval", all.x = TRUE)
activityNoNA$steps <-
    ifelse(is.na(activityNoNA$steps), activityNoNA$avg.steps, activityNoNA$steps)
```

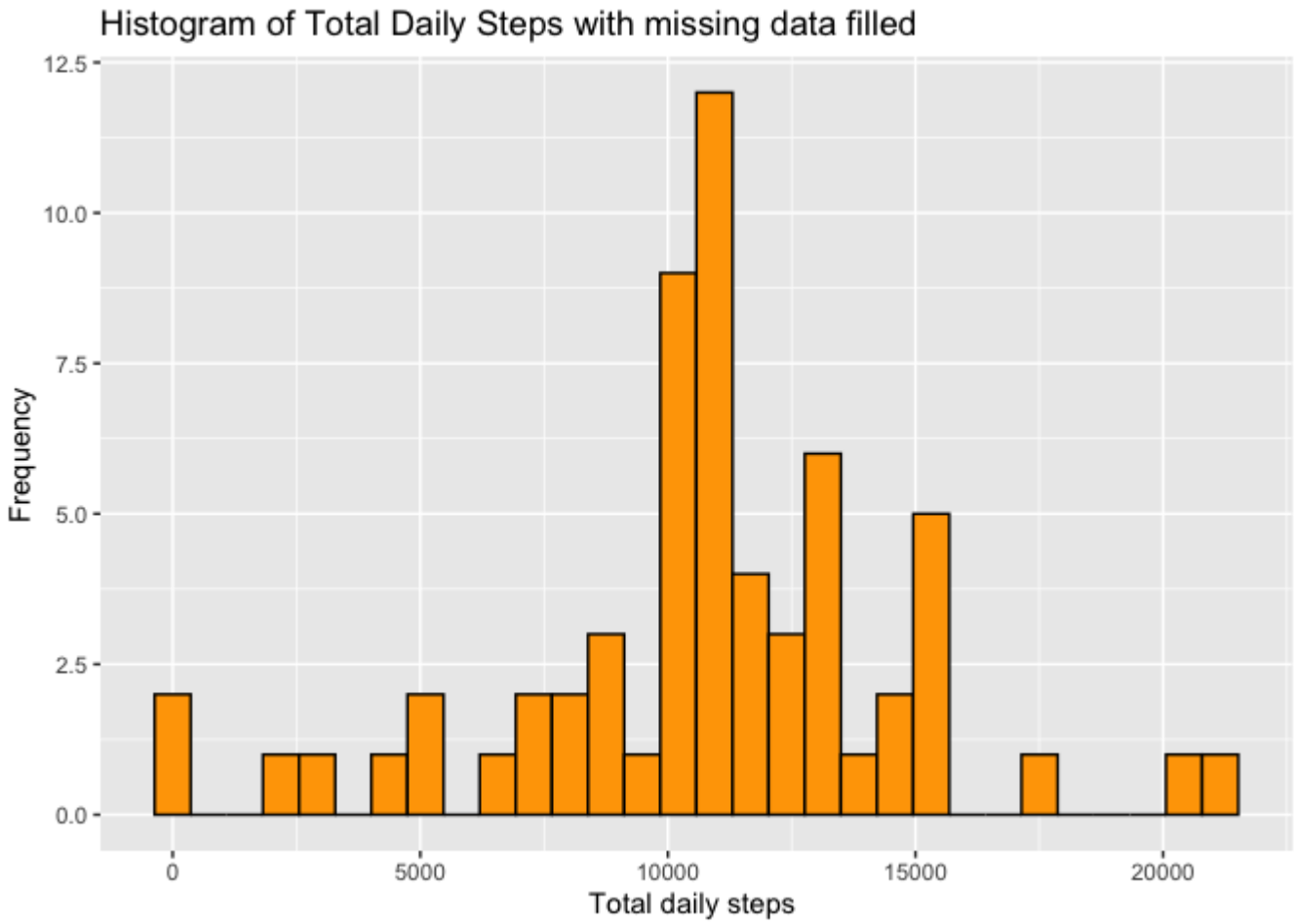
Make the dataset format equal to the original daily format

```
dailyNoNA <- activityNoNA[, 1:3] %>%
            group_by(date) %>%
            summarise(tot.steps = sum(steps))
```

Make a histogram of the total numbers of steps each day

```
ggplot(dailyNoNA, aes(x=tot.steps), breaks=30) +
    geom_histogram(color="black",fill="orange") +
    labs(x= "Total daily steps ") +  # Add labels
    labs(y= "Frequency") +
    labs(title = "Histogram of Total Daily Steps with missing data filled")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Next, calculate the mean and median of the daily steps:

```
dailyMeanNoNA <- round(mean(dailyNoNA$tot.steps), 0)
print(dailyMeanNoNA)
```

```
## [1] 10766
```

```
dailyMedianNoNA <- round(median(dailyNoNA$tot.steps), 0)
print(dailyMedianNoNA)
```

```
## [1] 10766
```

The mean remains equal to the mean of the first part of the assignment, because the NA values are replaced by the mean for that interval. The daily median has changed from 1.0765^{4} to 1.0766^{4} by updating the NA values with the mean for that interval.

Are there differences in activity patterns between weekdays and weekends?

First add a column that determines weekday or weekend using the date column

```
activityNoNA$day <- as.POSIXlt(activityNoNA$date)$wday
activityNoNA$dayType <- as.factor(ifelse(activityNoNA$day == 0 | activityNoNA$day ==
6 , "weekend", "weekday"))
head(activityNoNA)
```

```
##   interval    steps       date avg.steps day dayType
## 1        0 1.716981 2012-10-01  1.716981   1 weekday
## 2        0 0.000000 2012-11-23  1.716981   5 weekday
## 3        0 0.000000 2012-10-28  1.716981   0 weekend
## 4        0 0.000000 2012-11-06  1.716981   2 weekday
## 5        0 0.000000 2012-11-24  1.716981   6 weekend
## 6        0 0.000000 2012-11-15  1.716981   4 weekday
```

Now make the panel plot with a timeseries of the 5-minute interval (x-axes) and the average number of steps taken, averaged across all weekday days or weekend days (y-axes)

```
weekend <- activityNoNA[activityNoNA$dayType=="weekend", !(names(activityNoNA)%in% c
("steps", "date", "day"))]
weekday <- activityNoNA[activityNoNA$dayType=="weekday", !(names(activityNoNA)%in% c
("steps", "date", "day"))]


fiveminWeekend <- aggregate (avg.steps ~ interval, weekend, mean)
fiveminWeekday <- aggregate (avg.steps ~ interval, weekday, mean)


par(mfrow = c(2,1)) # 2 rows, 1 column

plot(fiveminWeekday$interval, fiveminWeekday$avg.steps, type="l", col="black", main
="Weekdays", xlab = "5-minute interval", ylab="Number of steps" )

plot(fiveminWeekend$interval, fiveminWeekend$avg.steps, type="l", col="black", main
="Weekend" , xlab = "5-minute interval", ylab="Number of steps")
```