Reproducible Research Project 1

Alana Escoto

2022-05-13

Data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. The measurements were regularly taken to improve people´s health and to find patterns in their behavior.

Data from the personal activity monitoring device collects data at 5 minute intervals through out the day. It consists of 2 months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

Pre-procession of data

1. Loading the data

Show any code that is needed to

1.1- Setting the working directory

```
wd<-"D:/Alana/Coursera - Edx/21-25-Data Science Foundations using R Specialization"
course<- "/25. Reproducible Research/Course Projects"
setwd(paste0(wd, course)); rm(wd, course)
```

1.2- Loading all the packages needed...

```
library(filesstrings); library(dplyr); library(lattice);
```

```
## Warning: package 'filesstrings' was built under R version 4.1.3
```

```
## Loading required package: stringr
```

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

```
## The following object is masked from 'package:filesstrings':
##
##     all_equal
```

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

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

```
library(ggplot2); library("ggpmisc"); library(grid); library(gridExtra)
```

```
## Warning: package 'ggpmisc' was built under R version 4.1.3

## Loading required package: ggpp

## Warning: package 'ggpp' was built under R version 4.1.3

##
## Attaching package: 'ggpp'

## The following object is masked from 'package:ggplot2':
##
##     annotate

## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
## "packedMatrix" of class "replValueSp"; definition not updated

## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
## "packedMatrix" of class "mMatrix"; definition not updated

## Warning: package 'gridExtra' was built under R version 4.1.3

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

1.3- Creating a folder for the project, downloading and unziping data

```
if(!file.exists("RR_project1")) {
    dir.create("RR_project1")
}

#file.move("./Rep_Res_Project_1.Rmd", "./RR_project1", overwrite = TRUE)
setwd("./RR_project1")
#file.edit("Rep_Res_Project_1.Rmd")

if(!file.exists("RR1_data.zip")){
    data<- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
    download.file(data, destfile = "./RR1.zip", mode = "wb")
    unzip("./RR1.zip")
}
```

1.4- Load the data (i.e. read.csv())

```
RR_data <- read.csv("activity.csv", sep = ",", header = TRUE)
```

2. Preprocessing the data

2.1- Process/transform the data (if necessary) into a format suitable for your analysis

```
RR_data$date <- as.Date(RR_data$date, "%Y-%m-%d")
```

2.2- Overview of the data

```
head(RR_data)
```

```
##   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
```

```
str(RR_data)
```

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

```
summary(RR_data)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

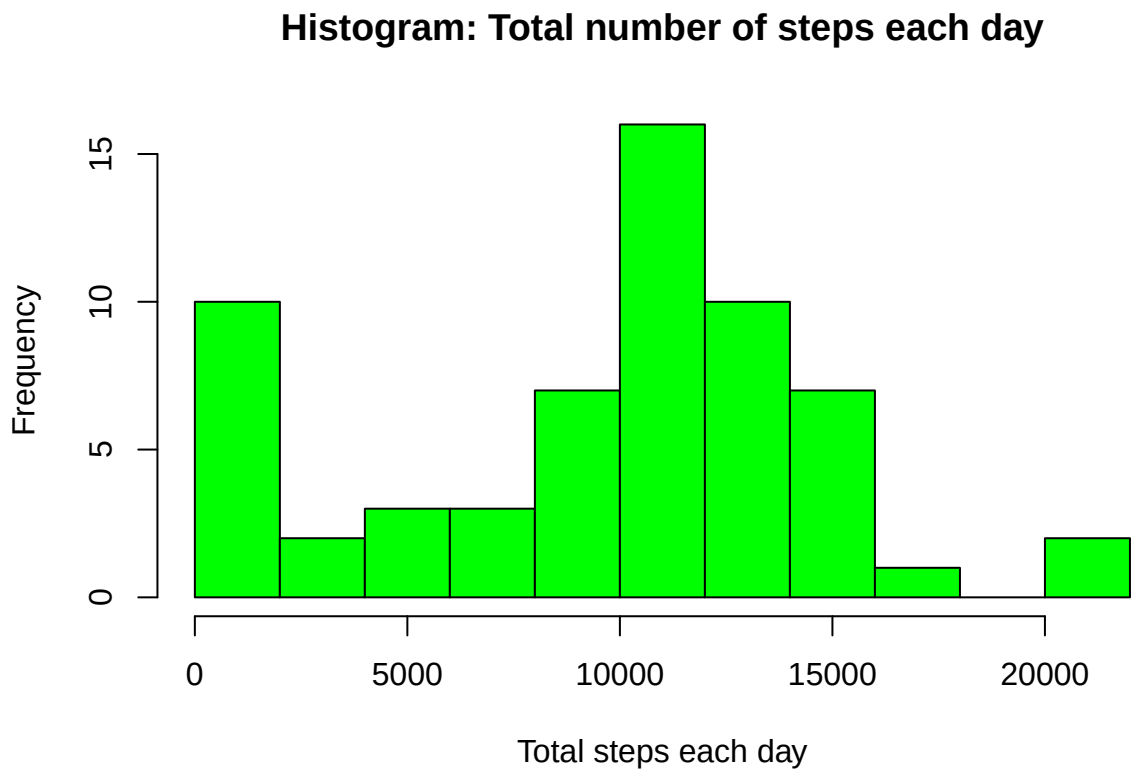
Data analysis

1. What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

1.1- Calculate the total number of steps taken per day & Make a histogram of the total number of steps taken each day

```
Tot_Steps<- RR_data %>%
            group_by(date) %>%
            summarise(Total.Steps = sum(steps, na.rm = TRUE))
hist(Tot_Steps$Total.Steps, col = "green", breaks = 8,
     main = "Histogram: Total number of steps each day",
     xlab = "Total steps each day")
```



1.2- Calculate and report the mean and median of the total number of steps taken per day

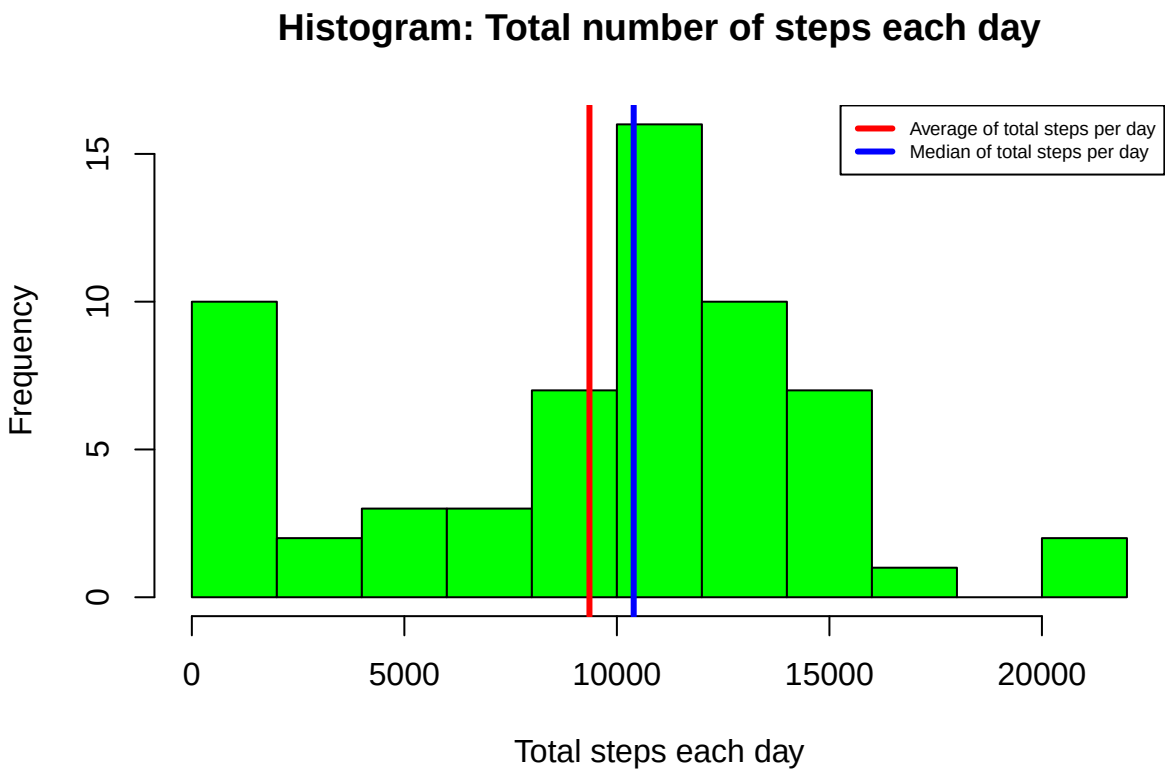
```
Tot_mean <- mean(Tot_Steps$Total.Steps)
Tot_median <- median(Tot_Steps$Total.Steps)
```

The average of the total number of steps taken per day is: **9354.2295082**.

The median of the total number of steps taken per day is: **10395**.

The average and median can be seen in the following histogram

```
hist(Tot_Steps$Total.Steps, col = "green", breaks = 8,
     main = "Histogram: Total number of steps each day",
     xlab = "Total steps each day")
abline(v = mean(Tot_Steps$Total.Steps), col = "red", lwd = 3)
abline(v = median(Tot_Steps$Total.Steps), col = "blue", lwd = 3)
legend("topright", cex = 0.6, lty = 1, lwd = 3,
       col = c("red", "blue"),
       legend = c("Average of total steps per day", "Median of total steps per day"))
```

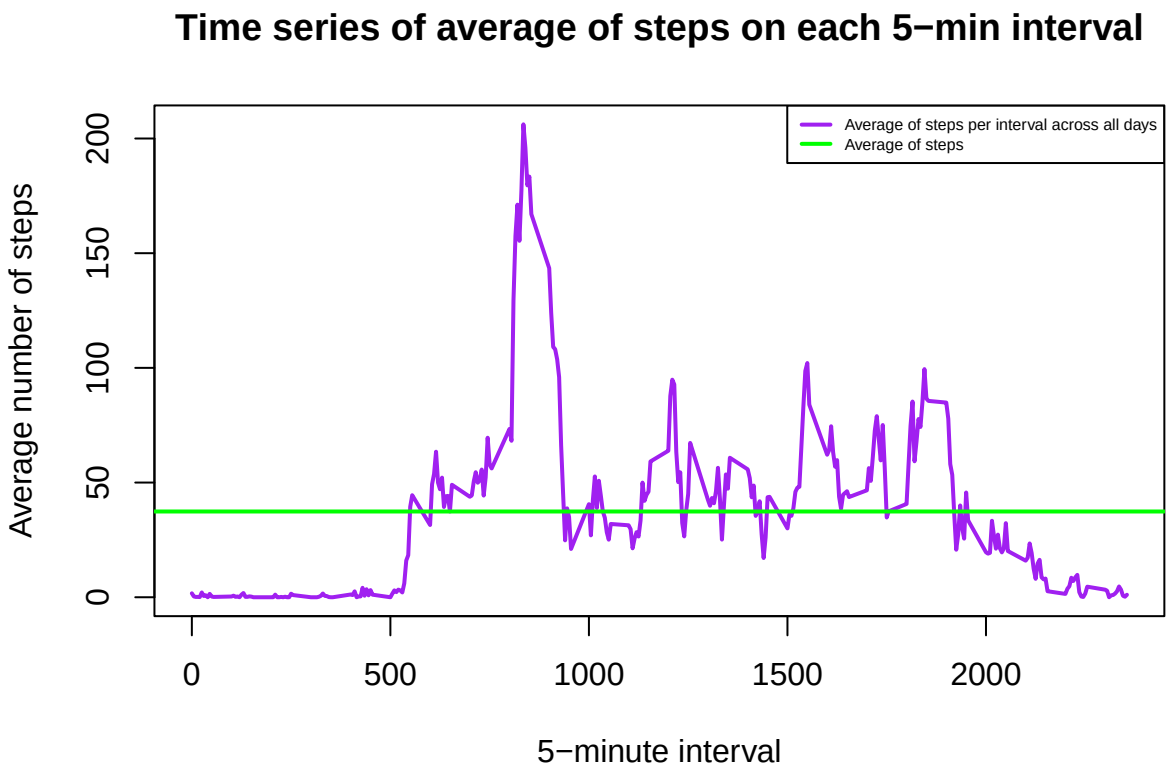


2. What is the average daily activity pattern?

2.1- Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
interval_avg<- RR_data %>%
                group_by(interval) %>%
                summarise(Avg.Steps = mean(steps, na.rm = TRUE))

plot(interval_avg$interval, interval_avg$Avg.Steps,
     col = "purple", type = "l", lwd = 2,
     main = "Time series of average of steps on each 5-min interval",
     xlab = "5-minute interval",
     ylab = "Average number of steps")
abline(h=mean(interval_avg$Avg.Steps, na.rm = T), col = "green", lwd = 2)
legend("topright", cex = 1/2, lty = 1, lwd = 2, col = c("purple", "green"),
       legend = c("Average of steps per interval across all days", "Average of steps"))
```

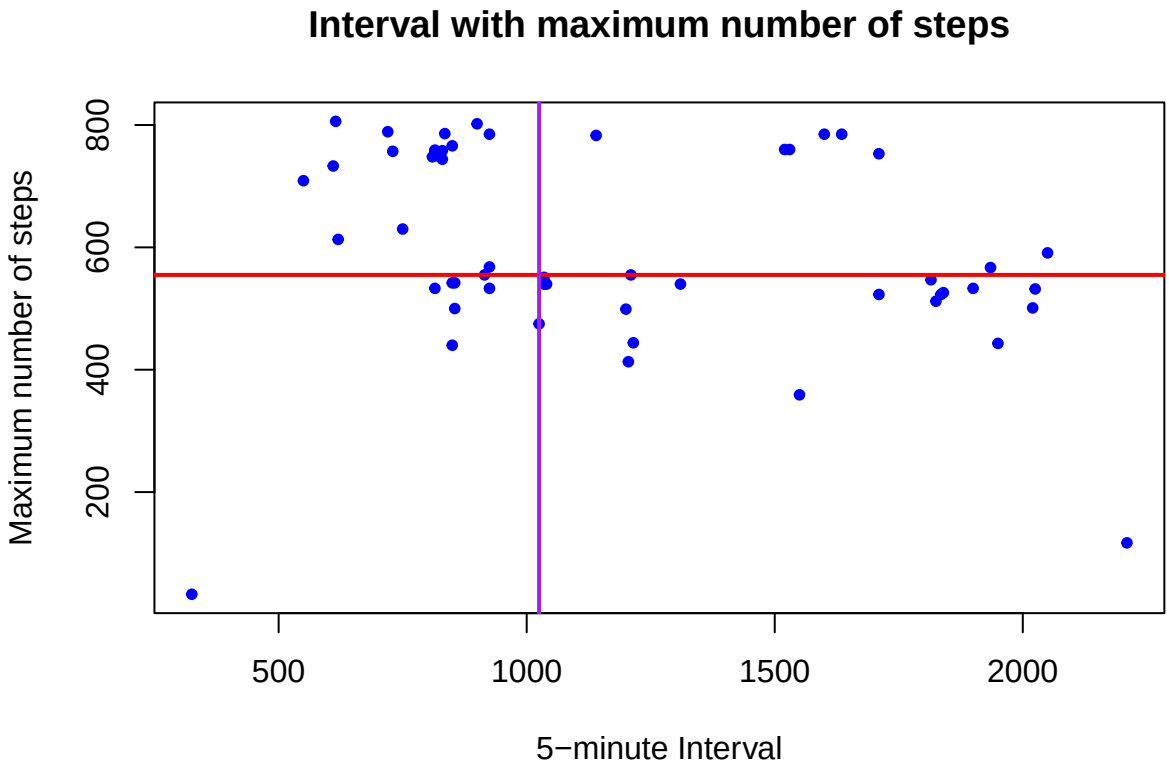


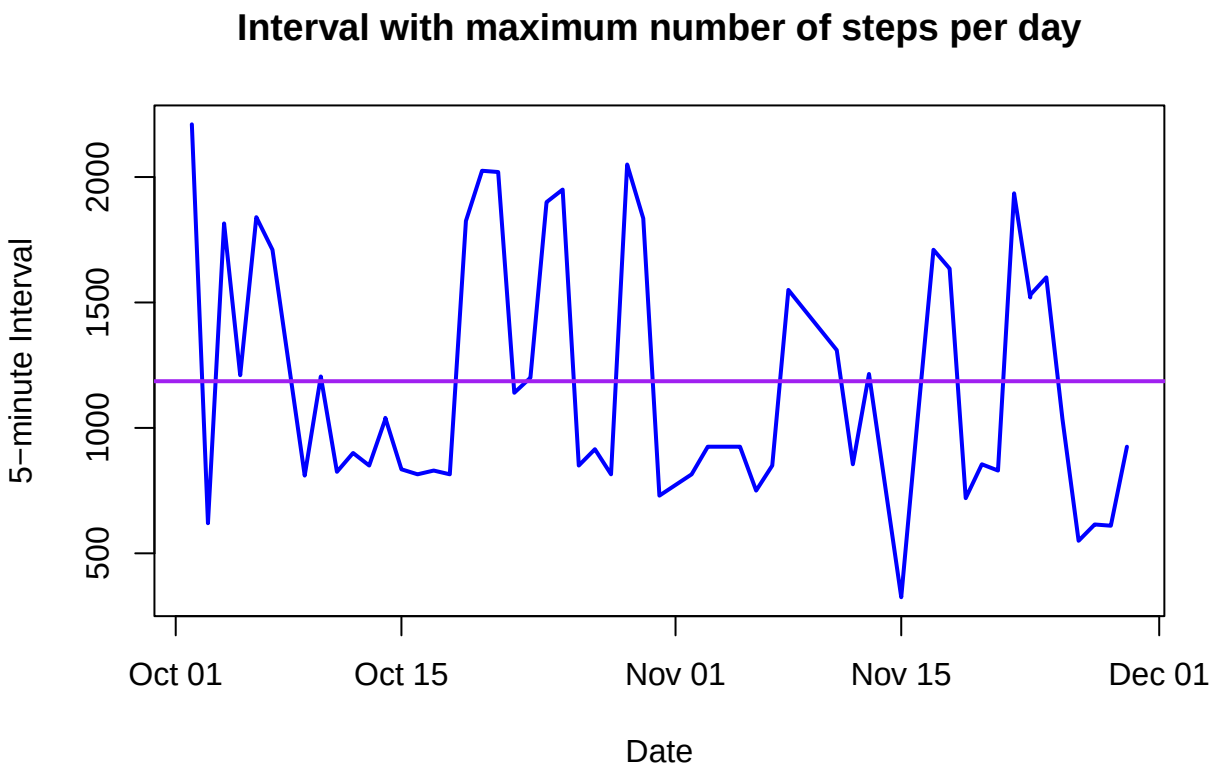
2.2- Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
Max_Steps<- RR_data %>%
            group_by(date) %>%
            filter(steps == max(steps))

par(mfrow = c(2,1))
plot(Max_Steps$interval, Max_Steps$steps, col = "blue", pch = 20,
     main = "Interval with maximum number of steps",
     xlab = "5-minute Interval",
     ylab = "Maximum number of steps")
abline(h = median(Max_Steps$steps), col = "red", lwd = 2)
abline(v = median(Max_Steps$interval), col = "purple", lwd = 2)

plot(Max_Steps$date, Max_Steps$interval, col = "blue", lwd = 2, type = "l",
     main = "Interval with maximum number of steps per day",
     xlab = "Date",
     ylab = "5-minute Interval")
abline(h = mean(Max_Steps$interval), col = "purple", lwd = 2)
```


Interval with maximum number of steps
Maximum number of steps
200
400
600
800
500
1000
1500
2000
5−minute Interval


Interval with maximum number of steps per day
5−minute Interval
500
1000
1500
2000
Oct 01
Oct 15
Nov 01
Nov 15
Dec 01
Date

```
int_avg<- mean(Max_Steps$interval)
```

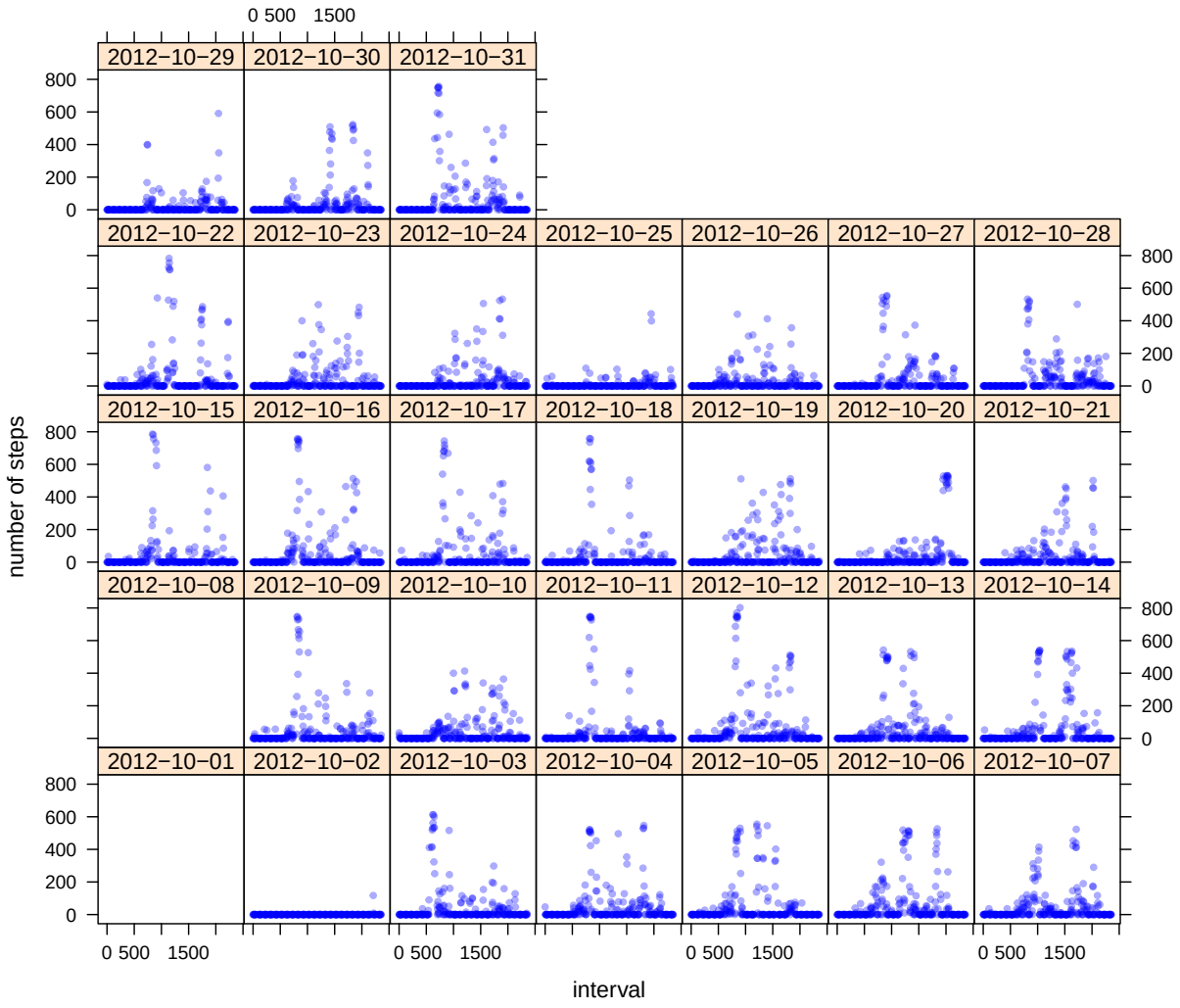
The 5-minute interval that, on average, contains the maximum number of steps: **1186.0909091**

3. Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as `NA`). The presence of missing days may introduce bias into some calculations or summaries of the data.

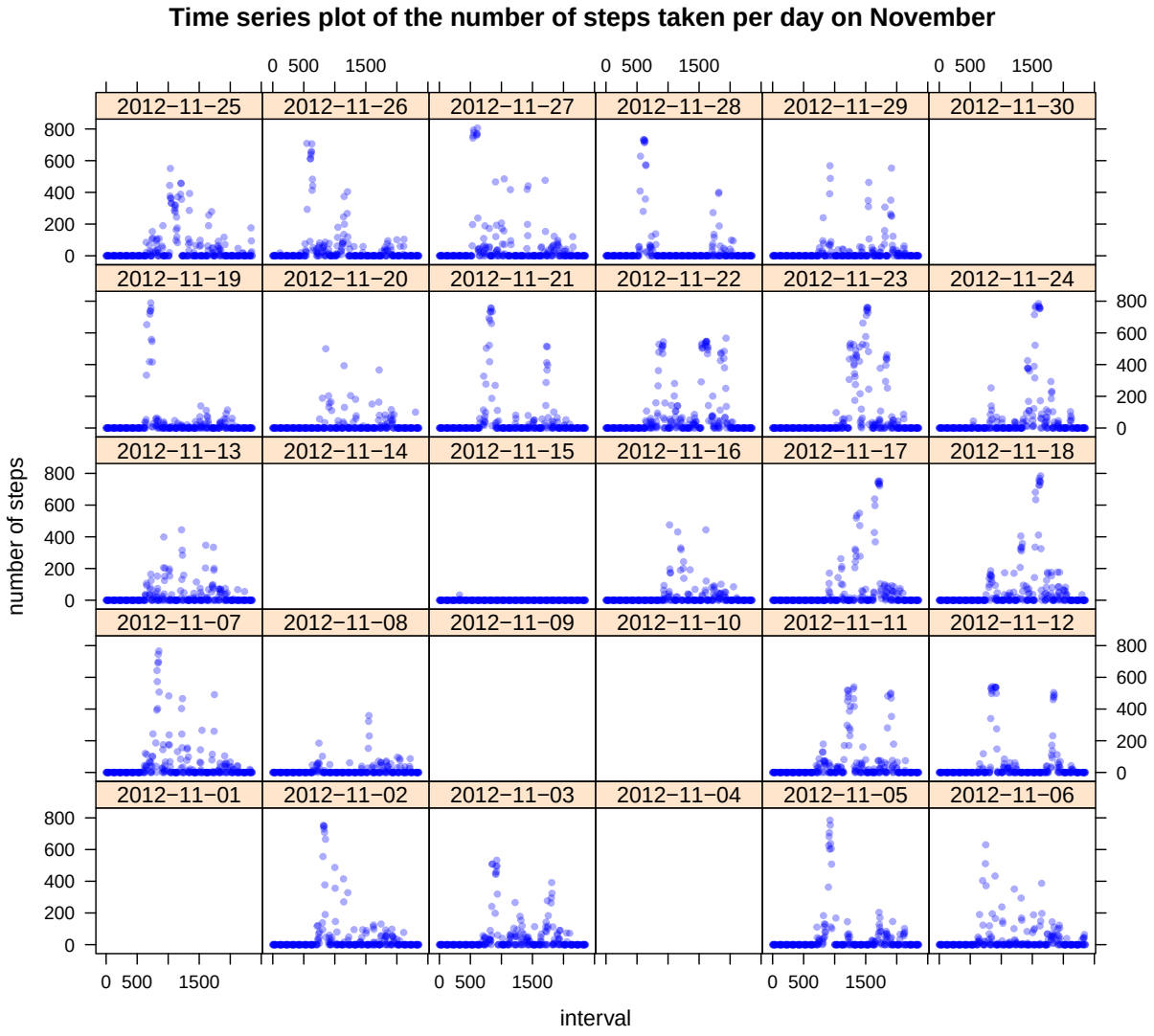
First identify visually the dates where there are missing values

```
date_group10<- RR_data %>%
    mutate(Month = format(RR_data$date, "%m"))  %>%
    filter(format(RR_data$date, "%m") == "10") %>%
    group_by(date)
xyplot(steps ~  interval | date, data = date_group10,
       strip = T, pch = 20, col = "blue" , alpha = 1/3,
       xlab = "interval", ylab = "number of steps",
       main="Time series plot of the number of steps taken per day on October")
```




```
date_group11<- RR_data %>%
    mutate(Month = format(RR_data$date, "%m"))  %>%
    filter(format(RR_data$date, "%m") == "11") %>%
    group_by(date)

xyplot(steps ~  interval | date, data = date_group11,
       strip = T, pch = 20, col = "blue" , alpha = 1/3,
       xlab = "interval", ylab = "number of steps",
       main="Time series plot of the number of steps taken per day on November")
```



3.1- Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with **NA**)

```
Tot_NA<- RR_data %>%
    filter(is.na(steps) == TRUE)
num_NA<- nrow(Tot_NA)
```

The total number of **NAs**in the data set is: **2304**

Specifically:

```
date_group10NA<- date_group10 %>%
    filter(is.na(steps) == TRUE)
num_10NA<- nrow(date_group10NA)
days10NA <-unique(date_group10NA$date)
```

- In October, there are a total of **576** **NA** corresponding to days:
  2012-10-01, 2012-10-08

```
date_group11NA<- date_group11 %>%
    filter(is.na(steps) == TRUE)
num_11NA<- nrow(date_group11NA)
days11NA <-unique(date_group11NA$date)
```

- In November, there are a total of **1728** `NA` corresponding to days:
  2012-11-01, 2012-11-04, 2012-11-09, 2012-11-10, 2012-11-14, 2012-11-30

3.2- Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

The missing values are replaced with the mean for that 5 min interval on that weekday

```
Clean.data<- RR_data %>%
            filter(is.na(steps) == FALSE)
Rep.NA <- Tot_NA

for (i in 1:nrow(Rep.NA)) {
    a<- weekdays(Rep.NA$date[i])
    b<- Rep.NA$interval[i]
    dat.sel<- Clean.data %>%
            filter(weekdays(Clean.data$date) == a &
                       Clean.data$interval == b)
    dat.mean<- mean(dat.sel$steps)
    Rep.NA$steps[i] <- dat.mean
}
```

3.3- Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
NA_free_RR <- rbind(Clean.data, Rep.NA)
NA_free_RR$steps <- as.numeric(NA_free_RR$steps)
NA_free_RR <- arrange(NA_free_RR, date, interval)
head(NA_free_RR)
```

```
##      steps       date interval
## 1 1.428571 2012-10-01        0
## 2 0.000000 2012-10-01        5
## 3 0.000000 2012-10-01       10
## 4 0.000000 2012-10-01       15
## 5 0.000000 2012-10-01       20
## 6 5.000000 2012-10-01       25
```

3.4- Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day.

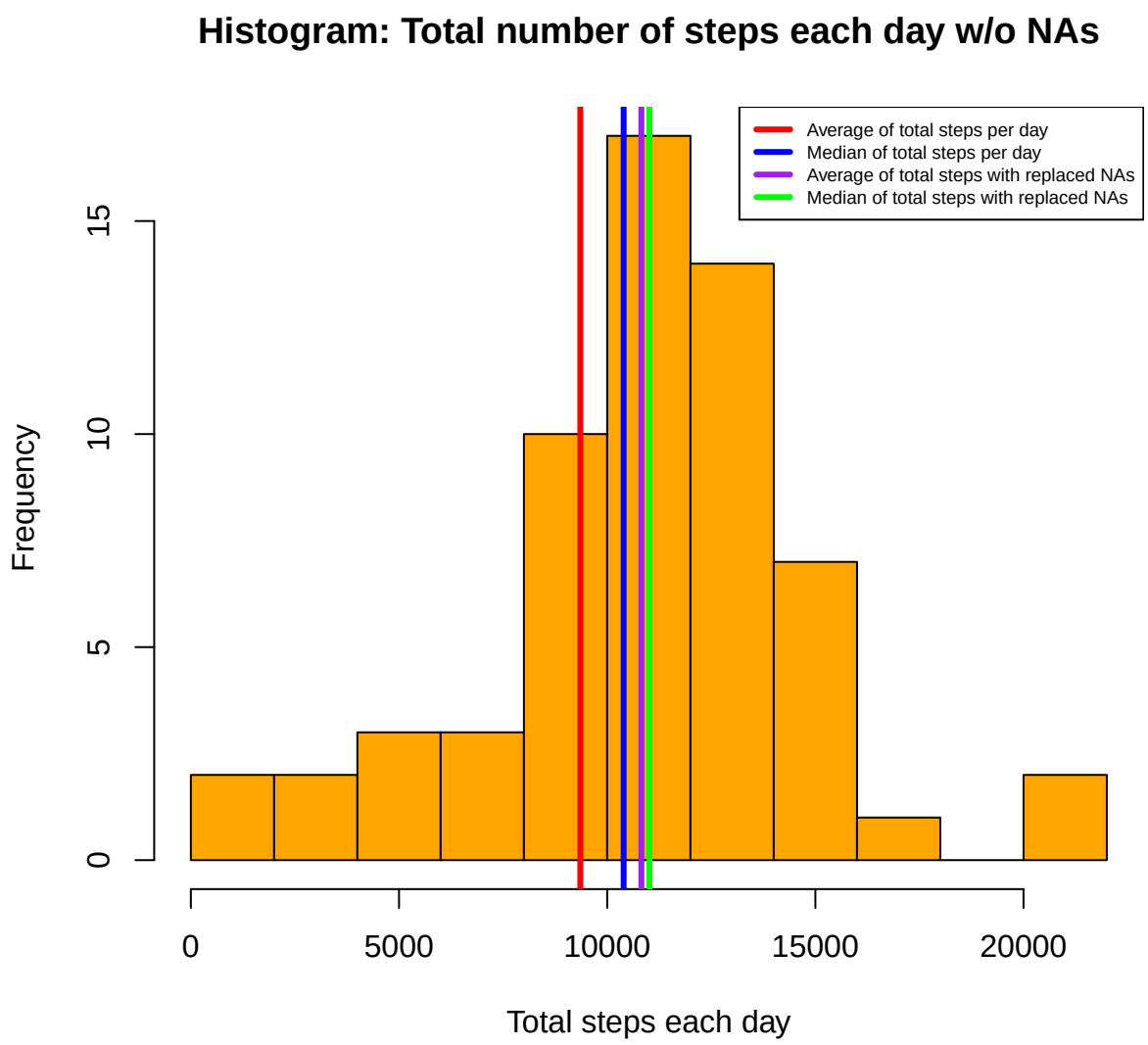
```
Tot_Steps2<- NA_free_RR %>%
            group_by(date) %>%
            summarise(Total.Steps = sum(steps))

A1<- round(mean(Tot_Steps$Total.Steps), digits = 3)
```

```r
M1<- round(median(Tot_Steps$Total.Steps), digits = 3)

A2<- round(mean(Tot_Steps2$Total.Steps), digits = 3)
M2<- round(median(Tot_Steps2$Total.Steps), digits = 3)

hist(Tot_Steps2$Total.Steps, col = "orange", breaks = 8,
     main = "Histogram: Total number of steps each day w/o NAs",
     xlab = "Total steps each day")
abline(v = mean(Tot_Steps$Total.Steps), col = "red", lwd = 3)
abline(v = median(Tot_Steps$Total.Steps), col = "blue", lwd = 3)
abline(v = mean(Tot_Steps2$Total.Steps), col = "purple", lwd = 3)
abline(v = median(Tot_Steps2$Total.Steps), col = "green", lwd = 3)
legend("topright", cex = 0.6, lty = 1, lwd = 3,
       col = c("red", "blue", "purple", "green"),
       legend = c("Average of total steps per day", "Median of total steps per day", "Average of total s
```



The total steps *mean* with **NAs** is: **9354.23**.

The total steps *mean* **without NAs** is: 1.0395×10^4.

The total steps *mean* with **NAs** is: 1.082121×10^4.

The total steps *median* **without NAs** is: 1.1015×10^4.

- Do these values differ from the estimates from the first part of the assignment?
  - **Yes**
- What is the impact of imputing missing data on the estimates of the total daily number of steps?
  - **Both estimates are higher when imputing missing data**

4. Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

4.1- Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
week_days <- weekdays(NA_free_RR$date)
for (i in 1:length(week_days)) {
    if (week_days[i] == "Saturday" | week_days[i] == "Sunday"){
        week_days[i] <- "weekend"
    } else {
        week_days[i] <- "weekday"
    }
}

NA_free_RR <- cbind(NA_free_RR, "Week" = week_days)
head(NA_free_RR)
```

```
##      steps       date interval    Week
## 1 1.428571 2012-10-01        0 weekday
## 2 0.000000 2012-10-01        5 weekday
## 3 0.000000 2012-10-01       10 weekday
## 4 0.000000 2012-10-01       15 weekday
## 5 0.000000 2012-10-01       20 weekday
## 6 5.000000 2012-10-01       25 weekday
```

4.2- Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
int_avg_wd<- NA_free_RR %>%
                group_by(Week, interval) %>%
                summarise(Avg.Steps = mean(steps))
```

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

```
theme_update(plot.title = element_text(hjust = 0.5))
g<- g<- qplot(interval, Avg.Steps, data = int_avg_wd, facets = Week ~.)
g + geom_point(aes(color = Week)) + geom_line(aes(color = Week)) +
    labs(x = "5-minute Interval", y = "Number of Steps",
         title = "Panel plot weekdays vs weekend")
```