Reproducible Research: Peer Assessment 1

Elena Fedorova

Loading and preprocessing the data

1. Loading data set into R using read.csv()

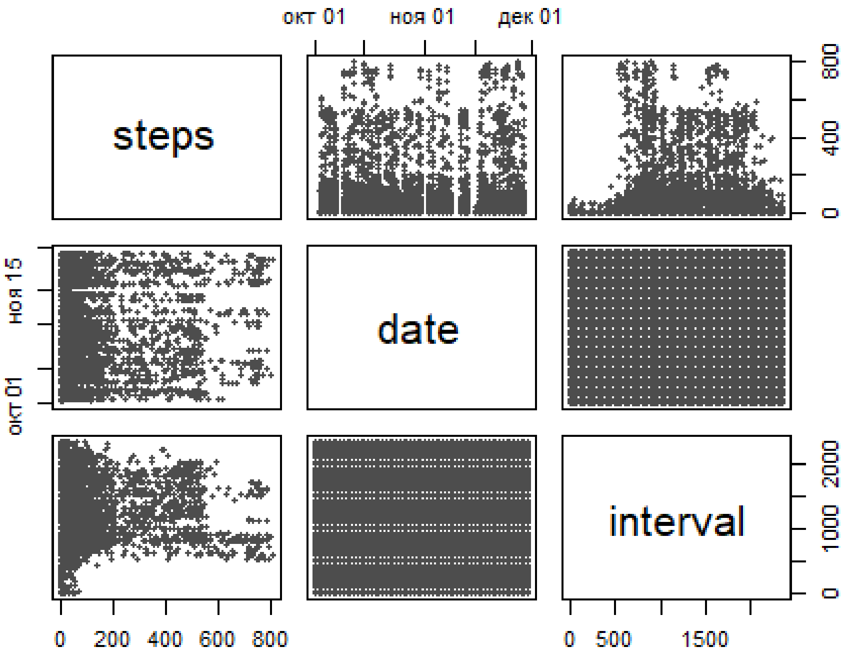
```
unzip("activity.zip")
activity <- read.csv("activity.csv", stringsAsFactors = FALSE)
```

2. Processing and transforming the data into a format suitable for the analysis

```
activity <- transform(activity, date = as.Date(date))
```

3. A quick glance at the structure of the data set

```
summary(activity)

##      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                                          

pairs(activity, col = "grey30", pch = 20)
```



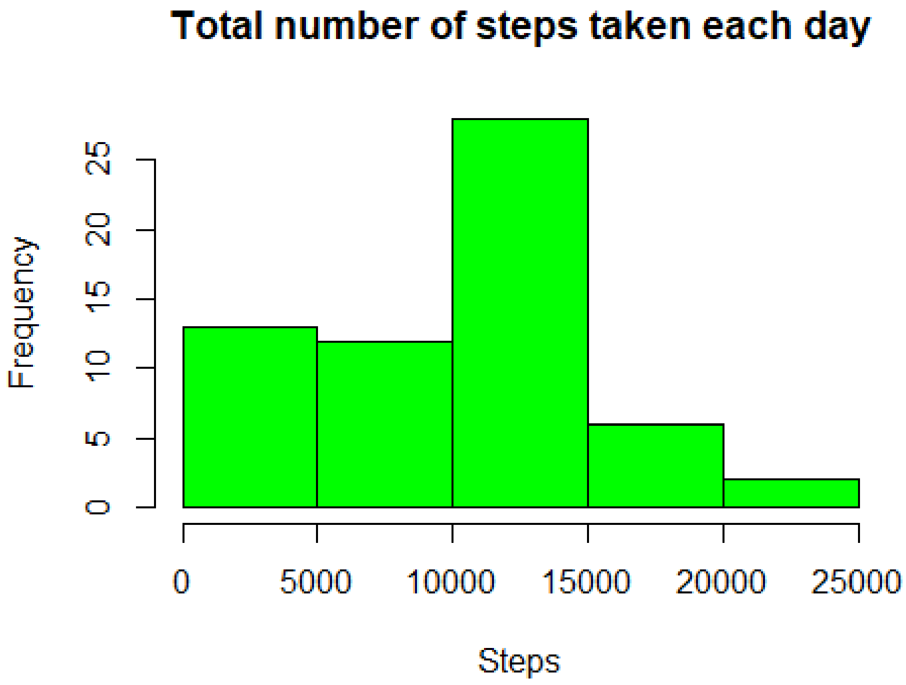
What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day

```
steps_per_day <- tapply(activity$steps, activity$date, sum, na.rm = TRUE)
```

2. Make a histogram of the total number of steps taken each day

```
hist(steps_per_day, xlab = "Steps", main = "Total number of steps taken each
day", col = "green")
```



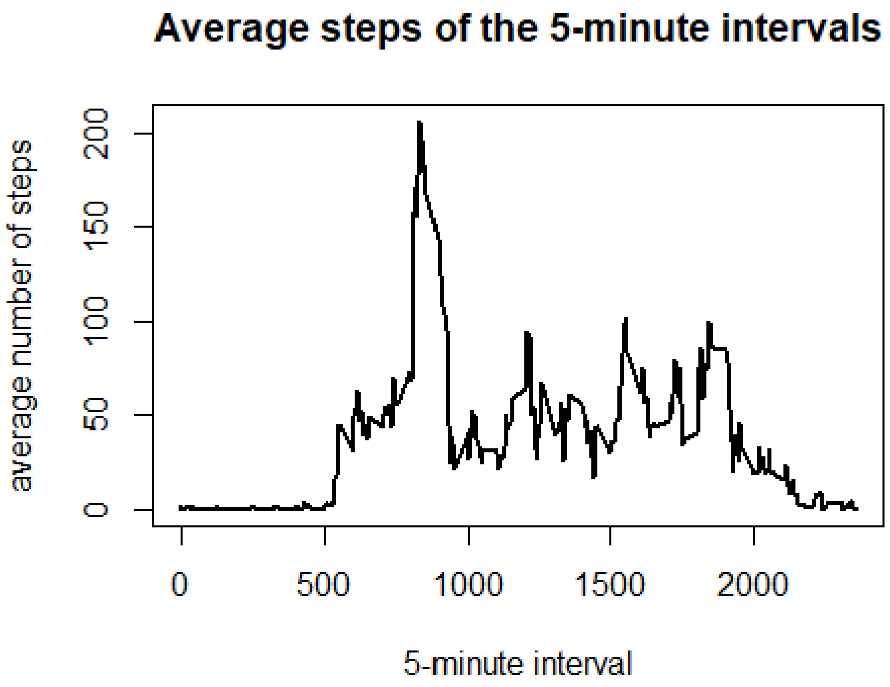
3. Calculate and report the mean and median of the total number of steps taken per day: “mean_steps” variable contains the mean of the total number of steps taken per day; “median_steps” variable contains the median of the total number of steps taken per day;

```
mean_steps <- mean(steps_per_day)
mean_steps

## [1] 9354.23

median_steps <- median(steps_per_day)
median_steps

## [1] 10395
```

What is the average daily activity pattern?

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)

```
mean_interval <- tapply(activity$steps, activity$interval, mean, na.rm = TRUE)

plot(x = names(mean_interval), y = mean_interval, type = "l",
     xlab = "5-minute interval", ylab = "average number of steps",
     main = "Average steps of the 5-minute intervals", lwd = 2)
```



2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps? The resulting interval is stored under variable called "max_interval"

```
max_interval <- names(which.max(mean_interval))
max_interval

## [1] "835"
```

Imputing missing values

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
total_NAs <- sum(is.na(activity$steps))
total_NAs

## [1] 2304
```

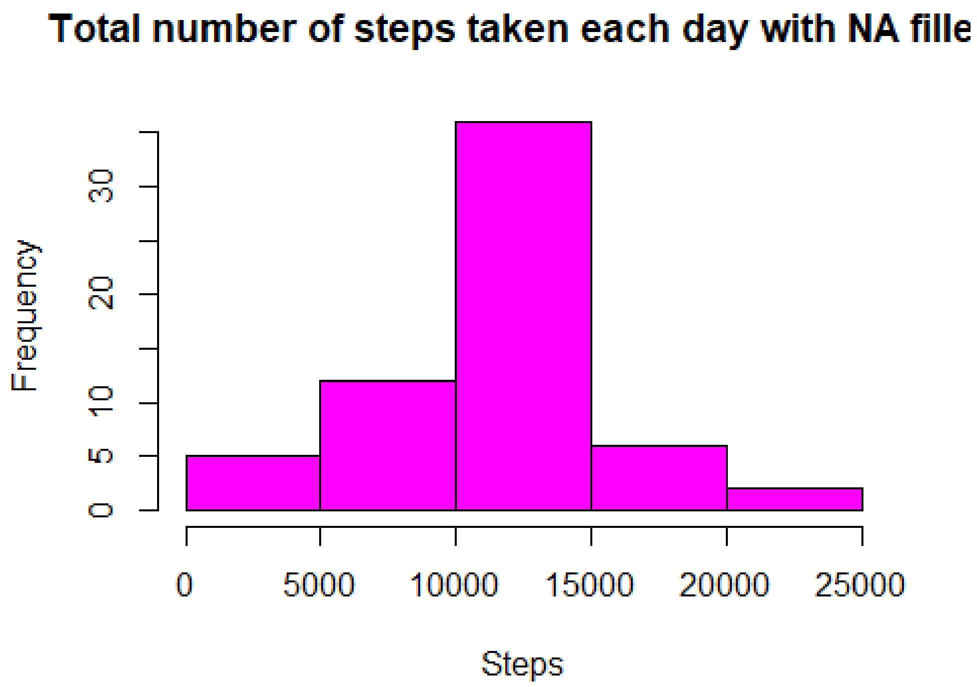
2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc. The missing values will be filled with using of the average per interval.

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

```
activity_filled <- activity
activity_filled$steps <- ifelse(is.na(activity_filled$steps),
round(mean_interval), activity_filled$steps)
```

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. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```
hist(tapply(activity_filled$steps, activity_filled$date, sum), xlab = "Steps",
     main = "Total number of steps taken each day with NA filled", col =
"magenta")
```



To undertand the impact of the NA imputation to the mean and median number of steps per day, let’s define “steps_per_day_new” by using the new data set:

```
steps_per_day_new <- tapply(activity_filled$steps, activity_filled$date, sum)
```

Definition of the “new_mean” and “new_median” after NA imputation:

```
new_mean <- mean(steps_per_day_new)
new_mean

## [1] 10765.64

new_median <- median(steps_per_day_new)
new_median

## [1] 10762
```

We can conclude that after filling the NAs number of steps taken each day increased so did the mean and median.

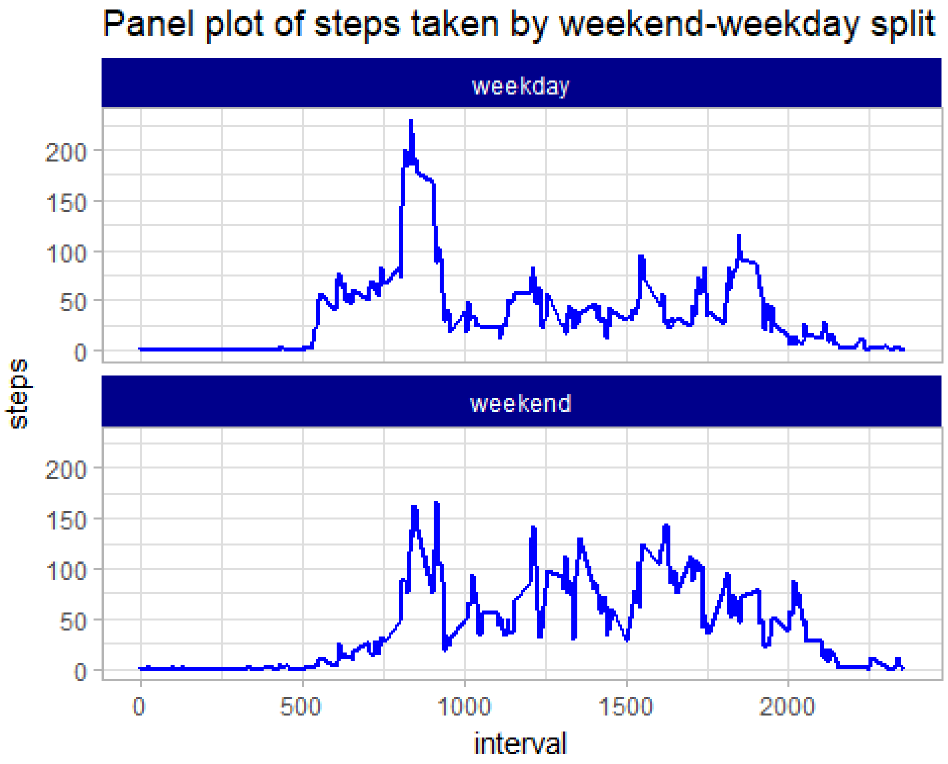
Are there differences in activity patterns between weekdays and weekends?

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_day <- format(activity_filled$date, "%u")
factor_week_day <- factor(ifelse(week_day > 5, "weekend", "weekday"))
activity_filled$week <- factor_week_day
```

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.

```
weekly_split <- split(activity_filled, activity_filled$week)
weekly_mean <- sapply(weekly_split, function(x) with(x, tapply(steps, interval,
mean)))
require(data.table)

## Loading required package: data.table

weekly_mean_melt <- data.table::melt(weekly_mean)
names(weekly_mean_melt) <- c("interval", "week", "steps")
require(ggplot2)

## Loading required package: ggplot2

ggplot(weekly_mean_melt, aes(x = interval, y = steps)) +
        geom_line(pch = 20, col = "blue", lwd = 1) +
        facet_wrap(~ week, nrow = 2) +
        theme_light() +
        theme(strip.background = element_rect(fill = "darkblue")) +
        ggtitle("Panel plot of steps taken by weekend-weekday split")

## Warning: Ignoring unknown parameters: shape
```


Panel plot of steps taken by weekend-weekday split
weekday
weekend
steps
200
150
100
50
0
200
150
100
50
0
0
500
1000
1500
2000
interval