Reproducible Research - project 1

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the â€śquantified selfâ€ť movement â€“ a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two 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.

Data description

The variables included in this dataset are:

- steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)
- date: The date on which the measurement was taken in YYYY-MM-DD format
- interval: Identifier for the 5-minute interval in which measurement was taken (ex.0 is midnight,5 is 0:05am, 100 is 1am, 1300 is 1pm, 2355 is 11:35 pm“)

Loading and preprocessing the data

```
library(dplyr)
setwd("C:/Users/Aleksander/Downloads/")
data <- read.csv("activity.csv", stringsAsFactors = F)
data$date <- as.Date(data$date, format = "%Y-%m-%d")
```

Assignment

What is mean total number of steps taken per day?

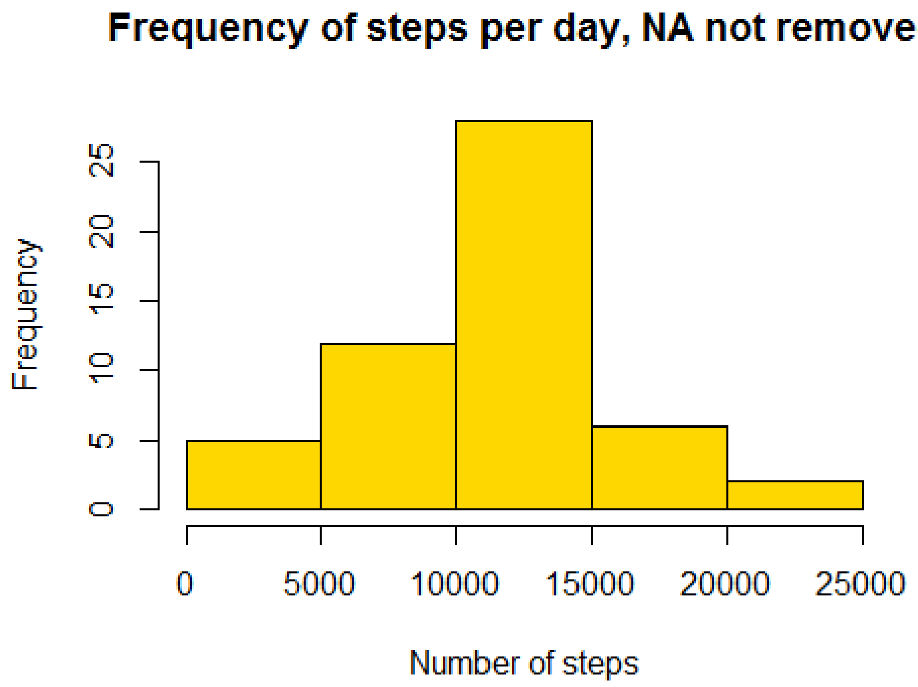
For this part of the assignment, you can ignore the missing values in the dataset.

- Calculate the total number of steps taken per day

```
steps_per_day <- aggregate(data$steps, list(data$date), FUN = sum)
colnames(steps_per_day) <- c("Date", "Steps")
head(steps_per_day, 10)

##          Date Steps
## 1  2012-10-01    NA
## 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    NA
## 9  2012-10-09 12811
## 10 2012-10-10  9900
```

- Make a histogram of the total number of steps taken each day

```
hist(steps_per_day$Steps, xlab = "Number of steps", main = "Frequency of
steps per day, NA not removed", col = "gold")
```



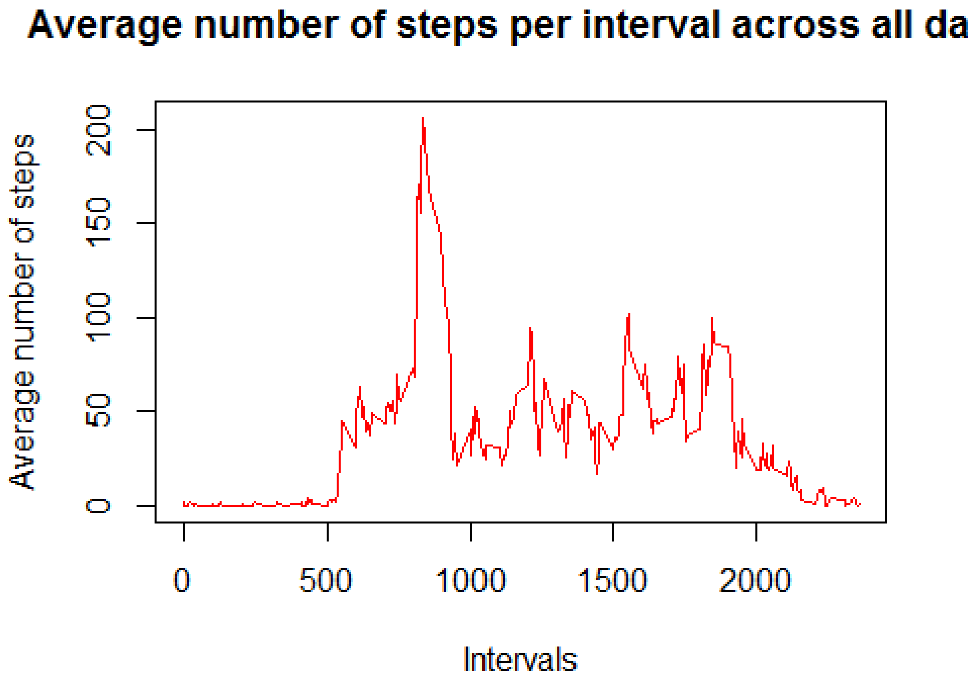
- Calculate and report the mean and median of the total number of steps taken per day

```
# Mean of the total number of steps taken per day
mean(steps_per_day$Steps, na.rm=TRUE)

## [1] 10766.19

# Median of the total number of steps taken per day
median(steps_per_day$Steps, na.rm=TRUE)

## [1] 10765
```

What is the average daily activity pattern?

- 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)

```
average_steps<-with(data,tapply(steps,interval,mean,na.rm=TRUE))
intervals<-unique(data$interval)
new_data<-data.frame(cbind(average_steps,intervals))
plot(new_data$intervals,new_data$average_steps,type = "l",xlab = "Intervals",
     ylab = "Average number of steps",main = "Average number of steps per
interval across all days", col='red')
```



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

```
new_data_ordered <- new_data[order(-average_steps),]
head(new_data_ordered, 1)

##     average_steps intervals
## 835      206.1698       835
```

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.

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

```
sum((is.na(data$steps)))

## [1] 2304
```

- 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.

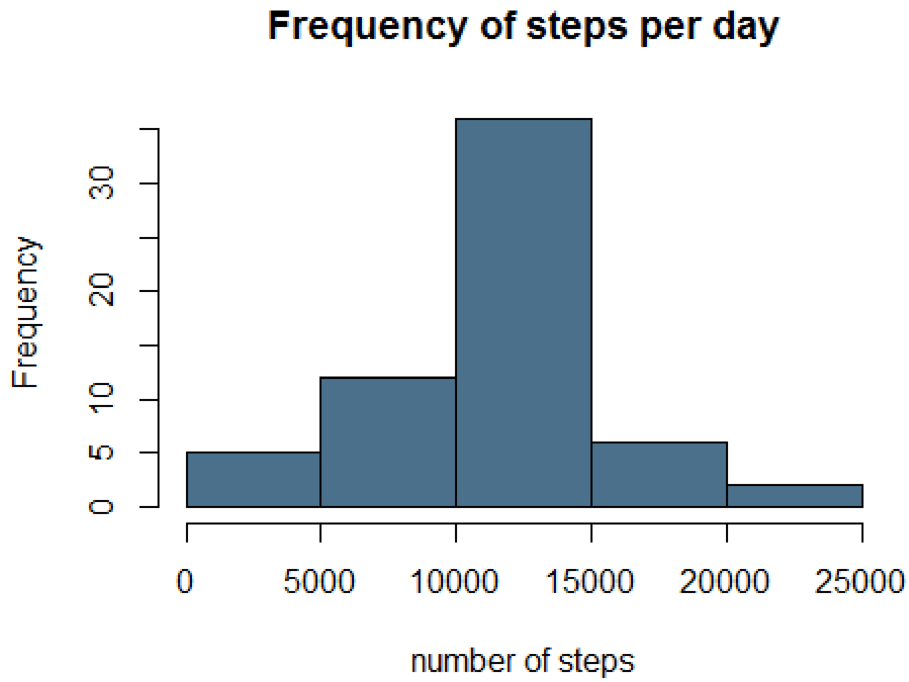
```r
data2 <- data %>%
  group_by(interval) %>%
  mutate(average = mean(steps,na.rm=TRUE))

data$complete_steps <- round(ifelse(is.na(data$steps), data2$average,
data$steps),2)
```

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

```r
data_no_NA <- data.frame(steps=data$complete_steps, interval = data$interval,
data = data$date)
```

- 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?

```r
steps_per_day_no_NA <-  aggregate(data_no_NA$steps, list(data_no_NA$data),
FUN = sum)
colnames(steps_per_day_no_NA) <- c("Date", "Steps")
hist(steps_per_day_no_NA$Steps, xlab = "number of steps", main = "Frequency
of steps per day", col = "skyblue4")
```



```
# Mean of the total number of steps taken per day
mean(steps_per_day_no_NA$Steps, na.rm=TRUE)

## [1] 10766.18

# Median of the total number of steps taken per day
median(steps_per_day_no_NA$Steps, na.rm=TRUE)

## [1] 10766.13
```

Here, there is almost no change between the values of the mean and the median with NA's and without NA's, althought different methods for filling in all of the missing values in the dataset can produce different results.

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.

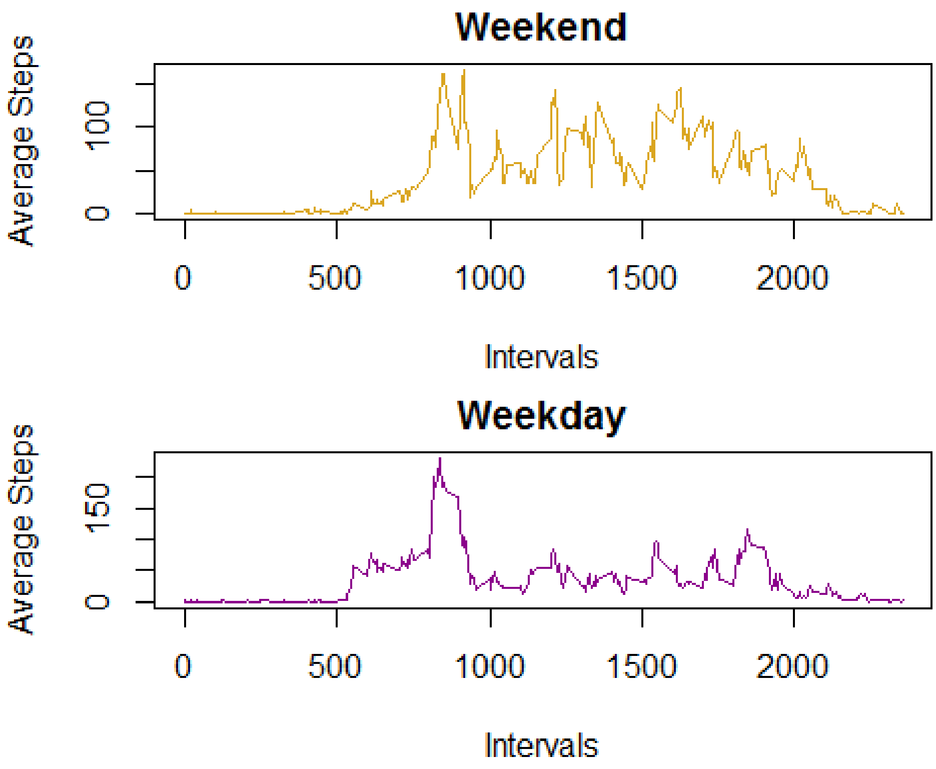
- Create a new factor variable in the dataset with two levels â€“ â€śweekdayâ€ť and â€śweekendâ€ť indicating whether a given date is a weekday or weekend day.

```
data_week <- data_no_NA
data_week$weekday <- weekdays(data_week$data)

# create a new variable indicating weekday or weekend
data_week$day_type <- ifelse(data_week$weekday=='sobota' |
data_week$weekday=='niedziela', 'weekend','weekday') # sobota is saturday in
polish and niedziela is sunday

head(data_week, n=10) # checking the method; poniedzialek is monday in polish

##    steps interval       data     weekday day_type
## 1   1.72        0 2012-10-01 poniedziałek  weekday
## 2   0.34        5 2012-10-01 poniedziałek  weekday
## 3   0.13       10 2012-10-01 poniedziałek  weekday
## 4   0.15       15 2012-10-01 poniedziałek  weekday
## 5   0.08       20 2012-10-01 poniedziałek  weekday
## 6   2.09       25 2012-10-01 poniedziałek  weekday
## 7   0.53       30 2012-10-01 poniedziałek  weekday
## 8   0.87       35 2012-10-01 poniedziałek  weekday
## 9   0.00       40 2012-10-01 poniedziałek  weekday
## 10  1.47       45 2012-10-01 poniedziałek  weekday
```

- 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

```
# making subsets for the 2 categories
activity_weekday<-subset(data_week,
as.character(data_week$day_type)=="weekday")
```

```
activity_weekend<-subset(data_week,
as.character(data_week$day_type)=="weekend")

#calculating the average numer of steps for each subset
steps_weekend<-with(activity_weekend,tapply(steps,interval,mean,na.rm=TRUE))
interval_weekend<-unique(activity_weekend$interval)
new_data_weekend<-data.frame(cbind(steps_weekend,interval_weekend))

steps_weekday<-with(activity_weekday,tapply(steps,interval,mean,na.rm=TRUE))
interval_weekday<-unique(activity_weekday$interval)
new_data_weekday<-data.frame(cbind(steps_weekday,interval_weekday))

# making the 2 plots for weekdays and weekends
par(mfrow = c(2,1), mar = c(4, 4, 2, 1))

plot(new_data_weekend$interval_weekend,new_data_weekend$steps_weekend,type =
"l",xlab = "Intervals",
     ylab = "Average Steps",main = "Weekend", col='goldenrod')

plot(new_data_weekday$interval_weekday,new_data_weekday$steps_weekday,type =
"l",xlab = "Intervals",
     ylab = "Average Steps",main = "Weekday", col='darkmagenta')
```



We can see that:

- the average steps during the weekends are lower that those during the weekdays before 10am.
- the average steps during the weekends are higher that those during the weekdays after 10am.