Project Assignment 1 - Reproducible Research

Ciro Barone

28 febbraio 2016

Introduction

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.

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

Loading and preprocessing the data

Show any code that is needed to:

- Load the data (i.e. read.csv())
- Process/transform the data (if necessary) into a format suitable for your analysis

Load the data

```
activity <- read.csv("D:/Users/cibarone/Progetti_Capgemini/reproducible_research_ass_1/activity.csv",hea
```

```
library(dplyr)
```

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

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

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

```
library(ggplot2)
```

Process/transform the data

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

```
##   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
```

Treat of Missing Value

```
table(is.na(activity$steps))
```

```
##
## FALSE  TRUE
## 15264  2304
```

```
table(is.na(activity$date))
```

```
##
## FALSE
## 17568
```

```
table(is.na(activity$interval))
```

```
##
## FALSE
## 17568
```

Subsetting Data set

With the following code I exclude drastically all NA value into the Steps Column.

```
activity_clean <- filter(activity,is.na(activity$steps)==FALSE)
activity_clean <-  droplevels.data.frame(activity_clean)
```

Then the dataset looks like that:

```
head(activity_clean)
```

```
##   steps       date interval
## 1     0 2012-10-02        0
## 2     0 2012-10-02        5
## 3     0 2012-10-02       10
## 4     0 2012-10-02       15
## 5     0 2012-10-02       20
## 6     0 2012-10-02       25
```

Formatting other columns

```
str(activity_clean)
```

```
## 'data.frame':    15264 obs. of  3 variables:
##  $ steps   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ date    : chr  "2012-10-02" "2012-10-02" "2012-10-02" "2012-10-02" ...
##  $ interval: num  0 5 10 15 20 25 30 35 40 45 ...
```

```
lct <- Sys.getlocale("LC_TIME"); Sys.setlocale("LC_TIME", "C")
```

```
## [1] "C"
```

```
activity$date <-as.Date(activity$date)
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 ...
```

AS is Tidy Data

```
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
```

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
- If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day
- Calculate and report the mean and median of the total number of steps taken per day

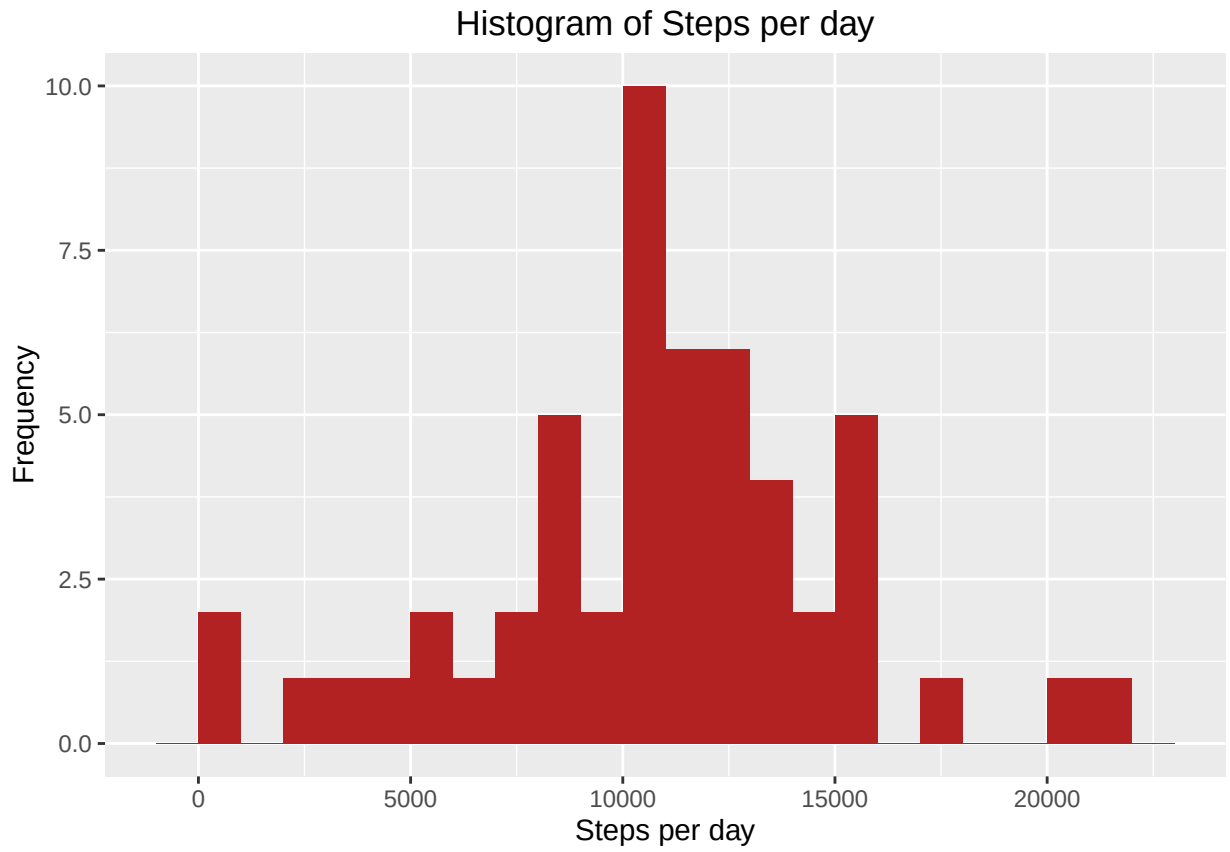
1 Calculate the total number of steps taken per day

```
steps <- activity %>%
  filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarize(steps = sum(steps)) %>%
print()
```

```
## Source: local data frame [53 x 2]
##
##          date steps
##        (date) (dbl)
## 1  2012-10-02   126
## 2  2012-10-03 11352
## 3  2012-10-04 12116
## 4  2012-10-05 13294
## 5  2012-10-06 15420
## 6  2012-10-07 11015
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
## ..        ...   ...
```

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

```
ggplot(steps, aes(x = steps)) +
  geom_histogram(fill = "firebrick", binwidth = 1000) +
  labs(title = "Histogram of Steps per day", x = "Steps per day", y = "Frequency")
```



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

```
mean_steps <- mean(steps$steps, na.rm = TRUE)
median_steps <- median(steps$steps, na.rm = TRUE)
mean_steps
```

```
## [1] 10766.19
```

```
median_steps
```

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

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)

1.1 First of all, Calculate average for interval with dplyr:

```
interval <- activity %>%
  filter(!is.na(steps)) %>%
  group_by(interval) %>%
  summarize(steps = mean(steps))
print
```

```
## function (x, ...)
## UseMethod("print")
## <bytecode: 0x000000000469d0e8>
## <environment: namespace:base>
```

1.2 Then, plot it :

```
p <- ggplot(interval, aes(x=interval, y=steps)) +
  geom_line(color = "firebrick")
print
```

```
## function (x, ...)
## UseMethod("print")
## <bytecode: 0x000000000469d0e8>
## <environment: namespace:base>
```

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

```
interval[which.max(interval$steps),]
```

```
## Source: local data frame [1 x 2]
##
##   interval    steps
##      (dbl)    (dbl)
## 1      835 206.1698
```

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

1 Summarize all the missing values:

```
table(is.na(activity$steps))
```

```
##
## FALSE  TRUE
## 15264  2304
```

Missing values are 2304.

2

```
activity_clean <- activity
nas <- is.na(activity_clean$steps)
avg <- tapply(activity_clean$steps, activity_clean$interval, mean, na.rm=TRUE, simplify=TRUE)
activity_clean$steps[nas] <- avg[as.character(activity_clean$interval[nas])]
```

```
table(is.na(activity_clean$steps))
```

```
##
## FALSE
## 17568
```

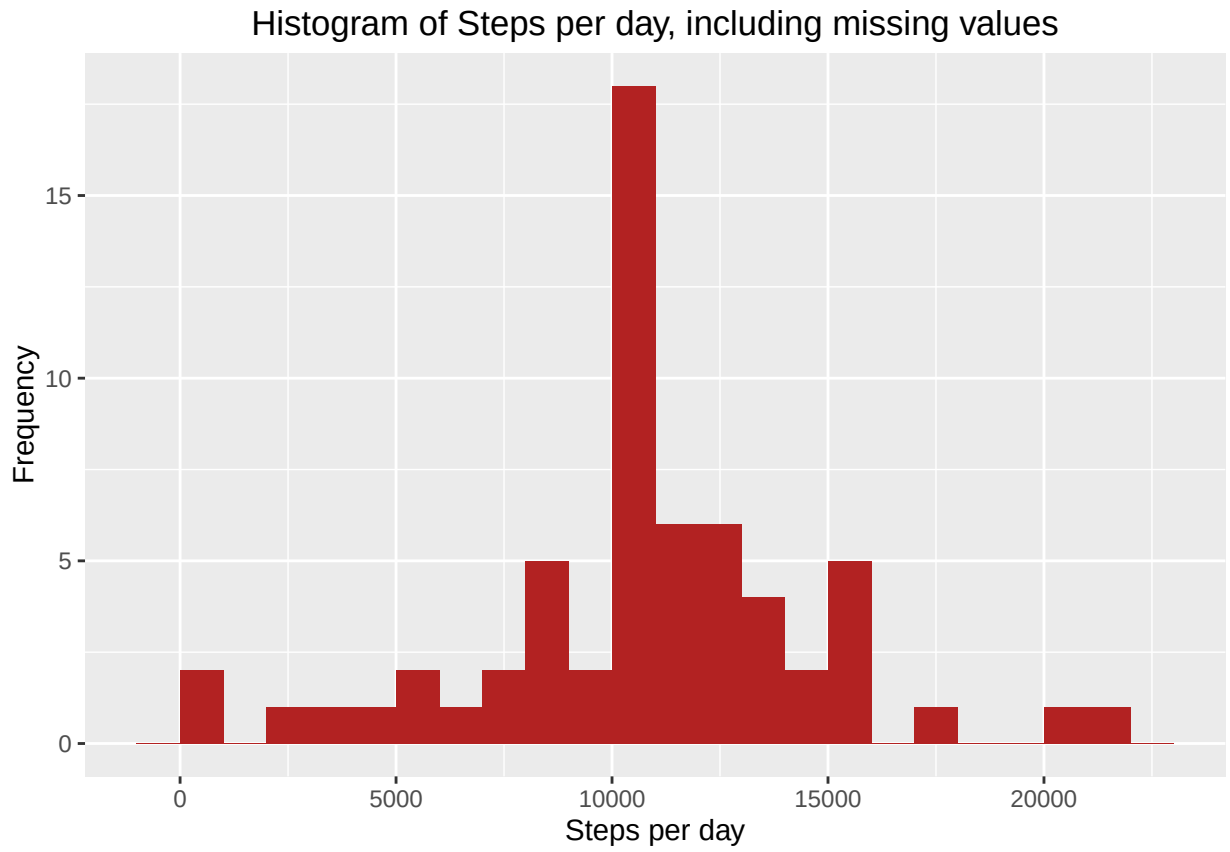
3 Calculate the number of steps taken in each 5-minute interval per day :

```
steps_clean <- activity_clean %>%
  filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarize(steps = sum(steps)) %>%
  print
```

```
## Source: local data frame [61 x 2]
##
##          date    steps
##        (date)    (dbl)
## 1  2012-10-01 10766.19
## 2  2012-10-02   126.00
## 3  2012-10-03 11352.00
## 4  2012-10-04 12116.00
## 5  2012-10-05 13294.00
## 6  2012-10-06 15420.00
## 7  2012-10-07 11015.00
## 8  2012-10-08 10766.19
## 9  2012-10-09 12811.00
## 10 2012-10-10  9900.00
## ..        ...      ...
```

3.1 Histogram:

```
ggplot(steps_clean, aes(x = steps)) +
  geom_histogram(fill = "firebrick", binwidth = 1000) +
  labs(title = "Histogram of Steps per day, including missing values", x = "Steps per day", y = "Freque
```



4 Calculate the mean and median steps with the filled in values:

```
mean_steps_clean<- mean(steps_clean$steps, na.rm = TRUE)
median_steps_clean <- median(steps_clean$steps, na.rm = TRUE)
mean_steps_clean
```

```
## [1] 10766.19
```

```
median_steps_clean
```

```
## [1] 10766.19
```

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 – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

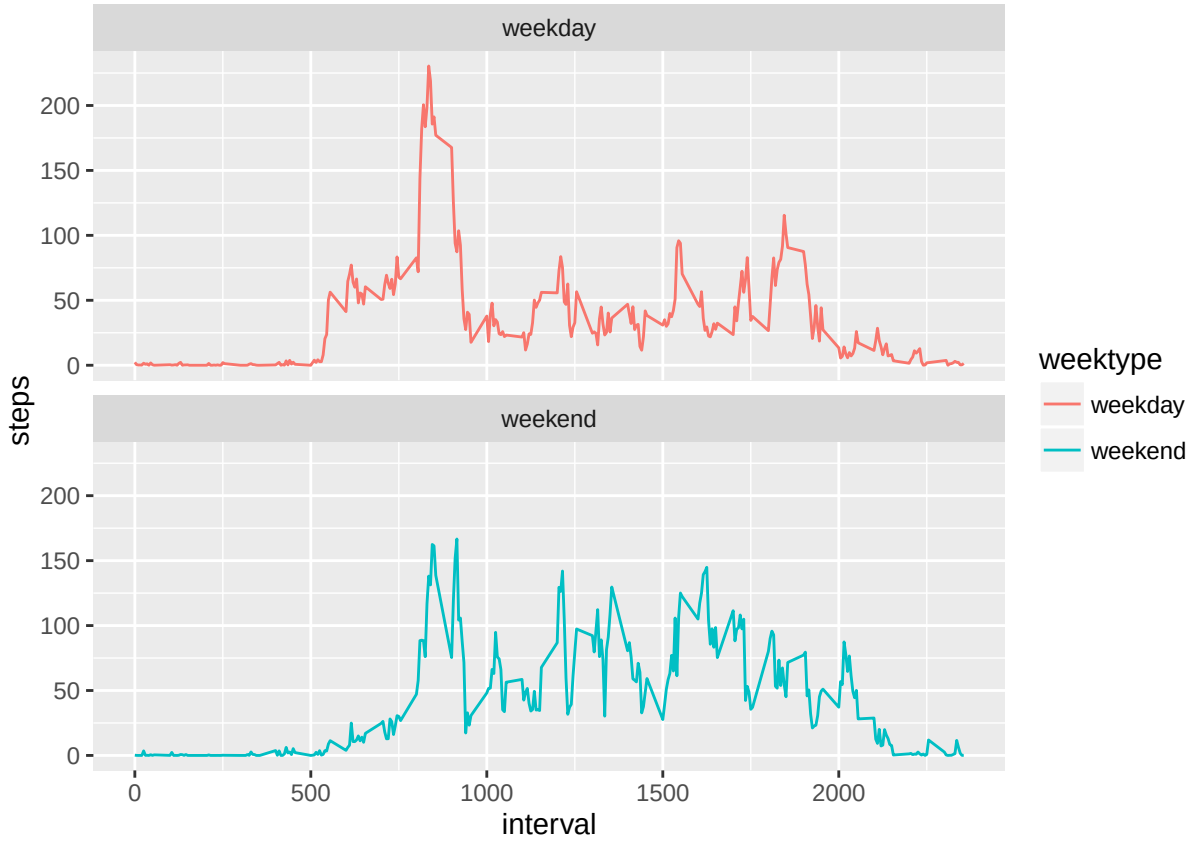
- Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using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

```
activity_clean <- mutate(activity_clean, weektype = ifelse(weekdays(activity_clean$date) == "Saturday"
activity_clean$weektype <- as.factor(activity_clean$weektype)
head(activity_clean)
```

```
##       steps       date interval weektype
## 1 1.7169811 2012-10-01        0  weekday
## 2 0.3396226 2012-10-01        5  weekday
## 3 0.1320755 2012-10-01       10  weekday
## 4 0.1509434 2012-10-01       15  weekday
## 5 0.0754717 2012-10-01       20  weekday
## 6 2.0943396 2012-10-01       25  weekday
```

2 finally we have :

```
interval_clean<- activity_clean %>%
  group_by(interval, weektype) %>%
  summarise(steps = mean(steps))
s <- ggplot(interval_clean, aes(x=interval, y=steps, color = weektype)) +
  geom_line() +
  facet_wrap(~weektype, ncol = 1, nrow=2)
print(s)
```



Finish