RR Peer Project 1

Sooraj Shetty

September 10, 2017

## Introduction

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

This document presents the results from Project Assignment 1 in the Coursera course Reproducible Research, written in a single R markdown document that can be processed by knitr and transformed into an HTML file.

## Data For the Analysis

The data can be downloaded from the course web site:

dataset: Activity Monitoring data

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

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

## Loading and preprocessing the data

Loading the data

#Loading necessary library files  
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.3.3

##   
## 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(lubridate)

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.3

#Reading the activity  
activity <- read.csv("activity.csv")  
  
#Tidying the data (using lubridate)  
activity$date <- ymd(activity$date)  
str(activity)

## '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 ...

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

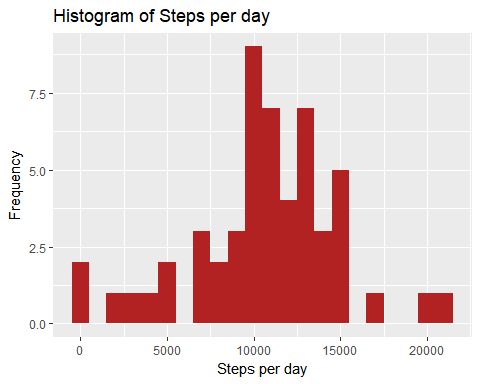
## What is mean total number of steps taken per day?

For this part of the assignment the missing values can be ignored.

#Calculate the total number of steps per day using dplyr and group by date:  
steps <- activity %>%  
 filter(!is.na(steps)) %>%  
 group\_by(date) %>%  
 summarize(steps = sum(steps)) %>%  
 print

## # A tibble: 53 × 2  
## date steps  
## <date> <int>  
## 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  
## # ... with 43 more rows

#Use ggplot for making the histogram:  
stepsplot <- ggplot(steps, aes(x = steps)) +  
 geom\_histogram(fill = "firebrick", binwidth = 1000) +  
 labs(title = "Histogram of Steps per day", x = "Steps per day", y = "Frequency")  
stepsplot



#Calculate 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

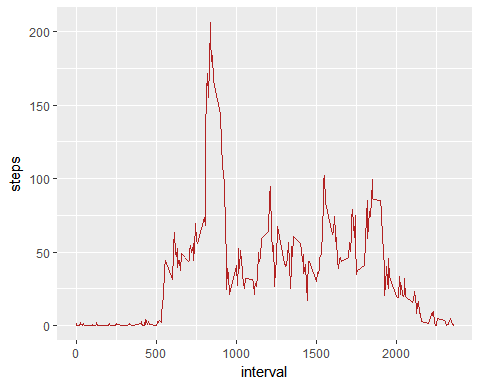
## [1] 10765

## What is the average daily activity pattern?

#Calculate the average number of steps taken in each 5-minute interval per day using dplyr and group by interval:  
interval <- activity %>%  
 filter(!is.na(steps)) %>%  
 group\_by(interval) %>%  
 summarize(steps = mean(steps))  
interval

## # A tibble: 288 × 2  
## interval steps  
## <int> <dbl>  
## 1 0 1.7169811  
## 2 5 0.3396226  
## 3 10 0.1320755  
## 4 15 0.1509434  
## 5 20 0.0754717  
## 6 25 2.0943396  
## 7 30 0.5283019  
## 8 35 0.8679245  
## 9 40 0.0000000  
## 10 45 1.4716981  
## # ... with 278 more rows

#Use ggplot for making the time series of the 5-minute interval and average steps taken:  
intervalplot <- ggplot(interval, aes(x=interval, y=steps)) +  
 geom\_line(color = "firebrick")  
intervalplot



#Use which.max() to find out the maximum steps, on average, across all the days:  
interval[which.max(interval$steps),]

## # A tibble: 1 × 2  
## interval steps  
## <int> <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 activity.

#Summarize all the missing values:  
sum(is.na(activity$steps))

## [1] 2304

Let’s take the approach to fill in a missing NA with the average number of steps in the same 5-min interval.

#Create a new dataset as the original and use tapply for filling in the missing values with the average number of steps per 5-minute interval:  
activity\_full <- activity  
nas <- is.na(activity\_full$steps)  
avg\_interval <- tapply(activity\_full$steps, activity\_full$interval, mean, na.rm=TRUE, simplify=TRUE)  
activity\_full$steps[nas] <- avg\_interval[as.character(activity\_full$interval[nas])]  
  
#Check that there are no missing values:  
sum(is.na(activity\_full$steps))

## [1] 0

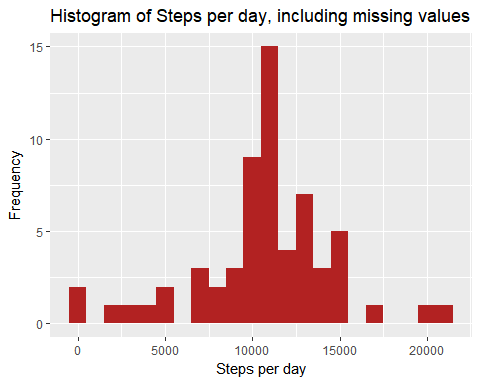
Calculate the number of steps taken in each 5-minute interval per day using dplyr and group by interval. Use ggplot for making the histogram:

steps\_full <- activity\_full %>%  
 filter(!is.na(steps)) %>%  
 group\_by(date) %>%  
 summarize(steps = sum(steps)) %>%  
 print

## # A tibble: 61 × 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  
## # ... with 51 more rows

Plotting the above result

intervalplotna <- ggplot(steps\_full, aes(x = steps)) +  
 geom\_histogram(fill = "firebrick", binwidth = 1000) +  
 labs(title = "Histogram of Steps per day, including missing values", x = "Steps per day", y = "Frequency")  
intervalplotna



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

mean\_steps\_full <- mean(steps\_full$steps, na.rm = TRUE)  
median\_steps\_full <- median(steps\_full$steps, na.rm = TRUE)  
mean\_steps\_full

## [1] 10766.19

median\_steps\_full

## [1] 10766.19

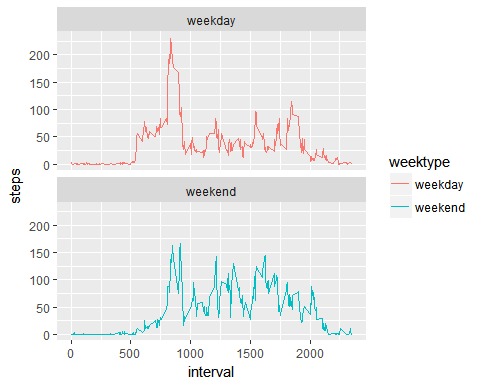
## Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() will come handy. Use the dataset with the filled-in missing values for this part.

#Use dplyr and mutate to create a new column, weektype, and apply whether the day is weekend or weekday:  
activity\_full <- mutate(activity\_full, weektype = ifelse(weekdays(activity\_full$date) == "Saturday" | weekdays(activity\_full$date) == "Sunday", "weekend", "weekday"))  
activity\_full$weektype <- as.factor(activity\_full$weektype)  
head(activity\_full)

## 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

#Calculate the average steps in the 5-minute interval and use ggplot for making the time series of the 5-minute interval for weekday and weekend, and compare the average steps:  
interval\_full <- activity\_full %>%  
 group\_by(interval, weektype) %>%  
 summarise(steps = mean(steps))  
s <- ggplot(interval\_full, aes(x=interval, y=steps, color = weektype)) +  
 geom\_line() +  
 facet\_wrap(~weektype, ncol = 1, nrow=2)  
print(s)



From the two plots it seems that the test object is more active earlier in the day during weekdays compared to weekends, but more active throughout the weekends compared with weekdays (probably because the oject is working during the weekdays, hence moving less during the day).