Activity processing device\_assessment 1

7 November 2016

## 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 "quantified self" movement - 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

The data for this assignment can be downloaded from the course web site. 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

Read Data

activityData <- read.csv ("activity.csv", header = T, sep = ",", stringsAsFactors = F)

Convert the date column to the appropriate format:

activityData$date <- as.Date(activityData$date, "%Y-%m-%d")  
str(activityData)

## '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 ...

Check the dimensions and a few rows of our newly created data frame

dim(activityData)

## [1] 17568 3

head(activityData)

## 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

The above output shows we have indeed the number of observations and variables mentioned in the assignment description, and we can see that during the first day of data collection we have several intervals with missing values that we will need to deal later with.

## Analysis

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

We can use dplyr to group and summarize the data and store it in the variable AvgDay, the following lines calculate the total number of steps per day and the mean number of daily steps:

library (dplyr)

## Warning: package 'dplyr' was built under R version 3.3.2

##   
## 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

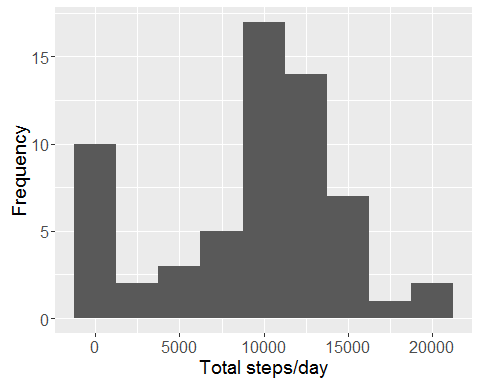
AvgDay <- activityData %>% group\_by(date) %>%  
 summarize(total.steps = sum(steps, na.rm = T),   
 mean.steps = mean(steps, na.rm = T))

Now we can construct the histogram of the total steps:

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.2

g <- ggplot(AvgDay, aes(x=total.steps))  
g + geom\_histogram(binwidth = 2500) + theme(axis.text = element\_text(size = 12),   
 axis.title = element\_text(size = 14)) + labs(y = "Frequency") + labs(x = "Total steps/day")



The histogram shows the largest count around the 10000-12500 step class thus we can infer that the median will be in this interval, the data is symmetrically distributed around the center of the distribution, except for one class at the extreme left. Let's get a summary of the data, which will include the mean and the median, to get a more quantitative insight of the data:

summary(AvgDay$total.steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0 6778 10400 9354 12810 21190

summary (AvgDay$mean.steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.1424 30.7000 37.3800 37.3800 46.1600 73.5900 8

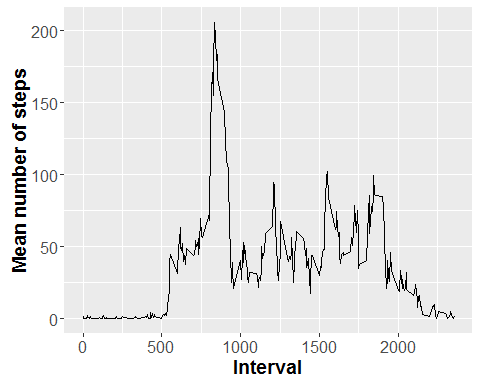
It looks like the mean and the median of the total steps are close in value. There are also 8 missing values.

1. What is the daily activity pattern?

In this section we will average the number of steps across each 5 min interval, this will give us an idea of the periods where the person might be the most and the least active (a screen shot of a "typical/average" day).

We group the data by interval this time and then calculate the mean of each interval goup:

AvgInterval <- activityData %>% group\_by(interval) %>%  
 summarize(mean.steps = mean(steps, na.rm = T))  
g <- ggplot(AvgInterval, aes(x = interval, y = mean.steps))  
g + geom\_line() + theme(axis.text = element\_text(size = 12),   
 axis.title = element\_text(size = 14, face = "bold")) +   
 labs(y = "Mean number of steps") + labs(x = "Interval")



We can observe the largest amount of steps occurs between time intervals 500 and 1000. The maximum average number of steps is: 206 and occurs in time interval #835

1. Imputing missing values

We noticed that there are missing values when we printed the first few rows of the activityData variable, but so far we have not determined how many values are missing. The following lines will calculate the percentage of missing data as well as the number of rows that contain an NA.

mean(is.na(activityData$steps))

## [1] 0.1311475

sum(is.na(activityData$steps))

## [1] 2304

About 13% of the data is missing. In order to evaluate the effect of filling in NAs with estimated values we will create a new dataset and then perform a comparison. There are several alternatives we can use to fill the NAs, for example:

Using the average steps during the day to fill in NAs within the same day. The drawbacks of this method are that we have seen there is a large variation thoughout the day (see timeseries plot) and more importantly we observed in the summary of the AvgDay that there are 8 days when no data was recorded so in those cases we would not have an estimator. Using the average steps per interval. We will use this metric as our first attempt to fill in the NAs. First, we will check for missing values in the interval column within AvgInterval, where we stored the mean number of steps for each 5 min interval

sum(is.na(AvgInterval$mean.steps))

## [1] 0

Since there are no missing values in this variable we will use it to fill in for NAs. Next we create a duplicate of the original data named newData and we will draw the appropriate values AvgInterval:

newData <- activityData

We check at each row if the column interval is NA, when the condition is true we look for the corresponding interval (index), we search for this particular interval in the AvgInterval data and extract it to a temporary variable values. Last we choose only the column of interest from values, which is the mean.steps and assign this number to the corresponding position in the newData set. We use a for loop to run through all the rows.

for (i in 1:nrow(newData)) {  
 if (is.na(newData$steps[i])) {  
 index <- newData$interval[i]  
 value <- subset(AvgInterval, interval==index)  
 newData$steps[i] <- value$mean.steps  
 }  
}  
head(newData)

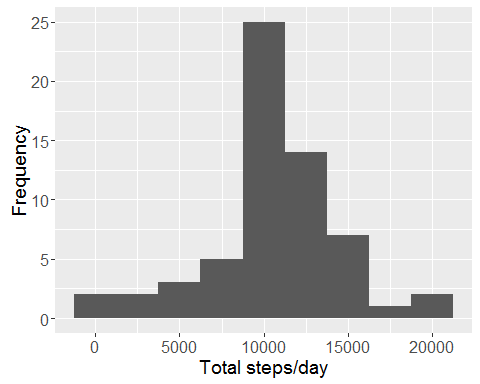
## steps date interval  
## 1 1.7169811 2012-10-01 0  
## 2 0.3396226 2012-10-01 5  
## 3 0.1320755 2012-10-01 10  
## 4 0.1509434 2012-10-01 15  
## 5 0.0754717 2012-10-01 20  
## 6 2.0943396 2012-10-01 25

We can observe from the previous output that now there are numeric values in the first rows of the dataset. We use a similar method as before to group the data by date and calculate daily totals:

newAvg <- newData %>% group\_by(date) %>%  
 summarize(total.steps = sum(steps, na.rm = T))

And we can construct the histogram:

g <- ggplot(newAvg, aes(x=total.steps))  
g + geom\_histogram(binwidth = 2500) + theme(axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 14)) + labs(y = "Frequency") + labs(x = "Total steps/day")



This figure shows, similarly to the first histogram, symmetrically distributed data around the maximum without the column in the extreme left (which contained the days with missing data). One must notice that filling values with the interval means increases the frequencies in the 10000-12500 class, which contains the median. For a more quantitative comparison lets review the 5 number summaries and standard deviations of the original data AvgDay vs the data with the imputed values newData

summary (AvgDay$total.steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0 6778 10400 9354 12810 21190

sd(AvgDay$total.steps, na.rm=T)

## [1] 5405.895

summary (newAvg$total.steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 41 9819 10770 10770 12810 21190

sd(newAvg$total.steps, na.rm=T)

## [1] 3974.391

The mean and the median stay the same, however the 1st quantile of the new data slides closer to the mean. When we look at the standard deviation values, we can also observe that the new data has a smaller standard deviation, thus the effect of imputing NAs with the mean values for the time intervals is a decrease in the spread, we obtained a distribution that is more concentrated around the center of gravity.

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

Different weekend vs weekday patterns are expected as people, in general, have a different set of activities on weekends. In order to find the specific patterns for each set of days, we will identify the weekdays from the weekend data. First, we create a new column in newData containing the values weekend or weekday:

newData$day <- ifelse(weekdays(newData$date) %in% c("Saturday", "Sunday"), "weekend", "weekday")

Next we create two subsets, one containing the weekend and one containing the weekday data:

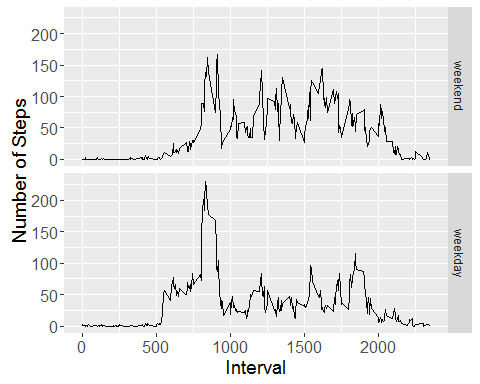
wkend <- filter(newData, day == "weekend")  
wkday <- filter(newData, day == "weekday")

Then, similarly to section 2, we group by the intervals and calculate the mean number of steps for each time interval. Since the day column is lots during the grouping, we add it again to the wkend and wday dataframes. Lastly, we merge both data sets into one named newInterval

wkend <- wkend %>%  
 group\_by(interval) %>%  
 summarize(mean.steps = mean(steps))   
wkend$day <- "weekend"  
  
wkday <- wkday %>%  
 group\_by(interval) %>%  
 summarize(mean.steps = mean(steps))   
wkday$day <- "weekday"  
  
newInterval <- rbind(wkend, wkday)  
newInterval$day <- as.factor(newInterval$day)  
newInterval$day <- relevel(newInterval$day, "weekend")

The two panel plot is now created, using the day column as a factor to spearate the weekday from the weekend timeseries.

g <- ggplot (newInterval, aes (interval, mean.steps))  
g + geom\_line() + facet\_grid (day~.) + theme(axis.text = element\_text(size = 12),   
 axis.title = element\_text(size = 14)) + labs(y = "Number of Steps") + labs(x = "Interval")



As expected, the activity profiles between weekdays and weekends greatly differ. During the weekdays, activity peaks in the morning between 7 and 9 and then the activity remains below ~100 steps. In contrast, the weekend data does not show a period with particularly high level of activity, but the activity remains higher than the weekday activity at most times and in several instances it surpases the 100 steps mark and it is overall more evenly distributed throughout the day.