Activity Monitoring Data : Peer Assesment 1

Manu Srivastav

Friday, April 17, 2015

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Loading and preprocessing the data

Part A : What is mean total number of steps taken per day?

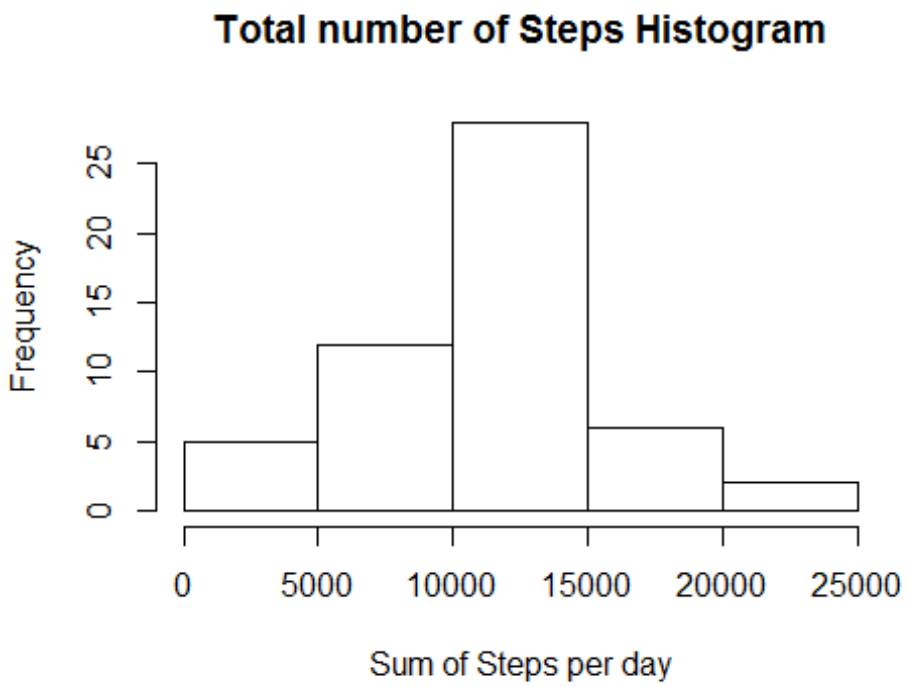
1. Calculate the total number of steps taken per day

2. Plot a histogram of the total steps by day

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Read the file
setwd("D:/Personal/msworksOthers/201210_DocsAndResearchAndKnowledge/Coursera/data/repdata_data_activity")
ourdata <- read.csv("activity.csv")
## Ignore the NA values
ourdata1 <-na.omit(ourdata)
##use dplyr to group the steps by date
by_date <- group_by(ourdata1,date)
## use dplyr to sum the steps per day
plot_by_date <-summarize(by_date, sum(steps))
names(plot_by_date)[2] <- c("SumSteps")

## Plot a histogram of the total steps by day
hist(plot_by_date$SumSteps, xlab = "Sum of Steps per day", main = "Total number of Steps Histogram")
```




```
## take a quick look at the summary
summary(plot_by_date)

##          date       SumSteps
##  2012-10-02: 1   Min.   :   41
##  2012-10-03: 1   1st Qu.: 8841
##  2012-10-04: 1   Median :10765
##  2012-10-05: 1   Mean   :10766
##  2012-10-06: 1   3rd Qu.:13294
##  2012-10-07: 1   Max.   :21194
##  (Other)   :47

## Find the mean of the Steps
mymean <- round(mean(plot_by_date$SumSteps))

## Find the medean of the Steps
mymedian <- round(median(plot_by_date$SumSteps))
```

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

The Mean of the total Steps is 10766

The Medean of the total Steps is 10765

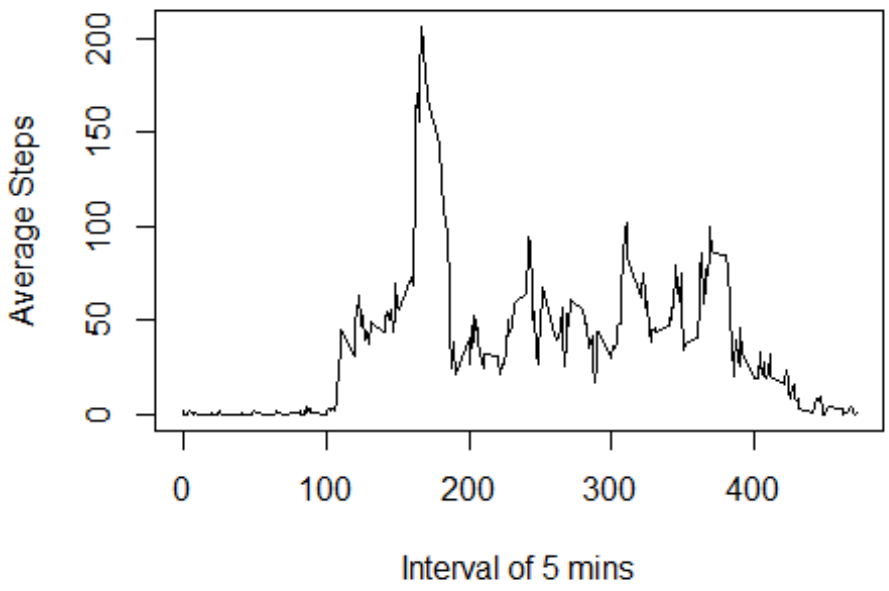
End of Part A

Part B : 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)

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

```
library(dplyr)
## Read the file
setwd("D:/Personal/msworksOthers/201210_DocsAndResearchAndKnowledge/Coursera/data/repdata_data_activity")
ourdata <- read.csv("activity.csv")
## Ignore the NA values
ourdata1 <-na.omit(ourdata)
##use dplyr to group the steps by interval
by_interval <- group_by(ourdata1,interval)
## use dplyr to mean the steps by interval

plot_by_interval <-summarize(by_interval, mean(steps))
names(plot_by_interval)[2] <- c("AvgSteps")

## TS plot of interval and mean steps during that interval of all days
plot.ts(plot_by_interval$interval/5, plot_by_interval$AvgSteps, type="l", xlab = "Interval of 5 mins", ylab = "Average Steps")
```




```
## take a quick look at the summary
summary(plot_by_interval)

##     interval         AvgSteps
##  Min.   :   0.0   Min.   :  0.000
##  1st Qu.: 588.8   1st Qu.:  2.486
##  Median :1177.5   Median : 34.113
##  Mean   :1177.5   Mean   : 37.383
##  3rd Qu.:1766.2   3rd Qu.: 52.835
##  Max.   :2355.0   Max.   :206.170

## Find the max steps
mymaxAvgSteps <- max(plot_by_interval$AvgSteps)

## arrange in desc order to get the interval
##head(arrange( plot_by_interval, desc(AvgSteps)))

## Divide the interval by 5 to get the which 5 min interval had the max
average steps
```

The max Avg Steps was 206.1698 and occured in interval 167

End of Part B

Part C : 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)

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

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

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.

```
library(dplyr)
## Read the file
setwd("D:/Personal/msworksOthers/201210_DocsAndResearchAndKnowledge/Coursera/data/repdata_data_activity")
ourdata <- read.csv("activity.csv")

## total number of rows with missing values in steps
length(which(is.na(ourdata$steps)))

## [1] 2304

totalNumberOfmissingRows <- length(which(is.na(ourdata$steps)))
ourdata1 <- ourdata
## filling the NA values in steps with  a value
## I could have used mean of the interval or mean of the day, but I preferred to used the mean of all the available steps for all the days to keep it simple and unbiased

ourdata1$steps[is.na(ourdata1$steps)] <- mean(ourdata1$steps, na.rm=TRUE)

##use dplyr to group the steps by date
by_date <- group_by(ourdata1,date)
## use dplyr to sum the steps per day
plot_by_date <-summarize(by_date, sum(steps))
names(plot_by_date)[2] <- c("SumSteps")

## Plot a histogram of the total steps by day
```

```r
hist(plot_by_date$SumSteps, xlab = "Sum of Steps per day", main = "Total
number of Steps Histogram")
```


```r
## take a quick look at the summary
summary(plot_by_date)
```

```
##          date       SumSteps
##  2012-10-01: 1   Min.   :   41
##  2012-10-02: 1   1st Qu.: 9819
##  2012-10-03: 1   Median :10766
##  2012-10-04: 1   Mean   :10766
##  2012-10-05: 1   3rd Qu.:12811
##  2012-10-06: 1   Max.   :21194
##  (Other)   :55
```

```r
## Find the mean of the Steps
mymean <- round(mean(plot_by_date$SumSteps))

## Find the medean of the Steps
mymedian <- round(median(plot_by_date$SumSteps))
```

The Mean of the total Steps with NA values replaced is 10766

The Medean of the total Steps with NA values replaced is 10766

As we can see the Mean and Median values dont defer with filling the NA values with a fixed values of the mean of the total steps.

End of Part C

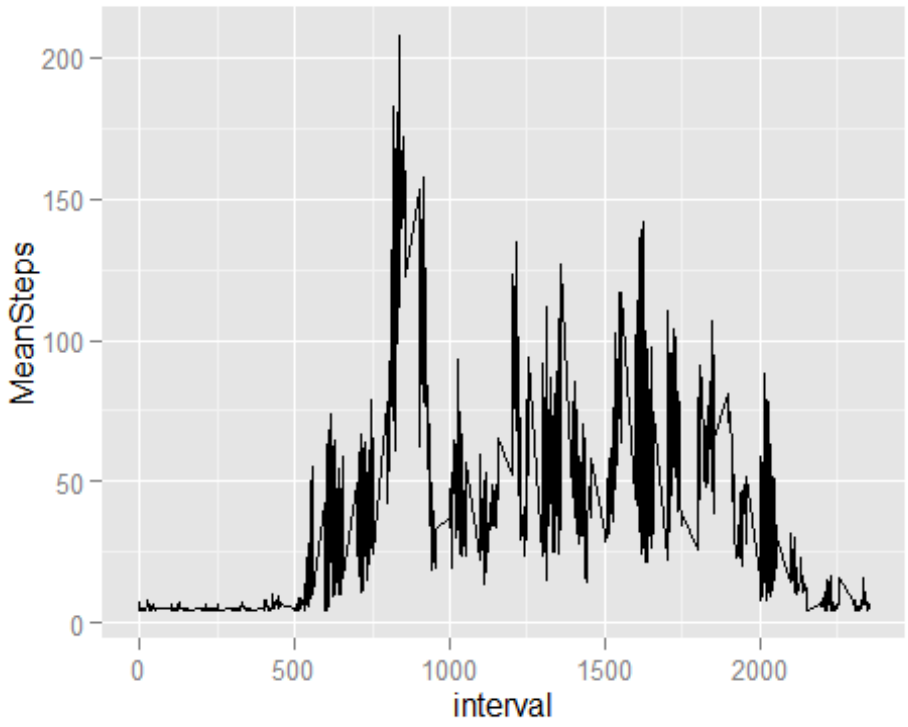
Part D : Are there differences in activity patterns between weekdays and weekends?

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.

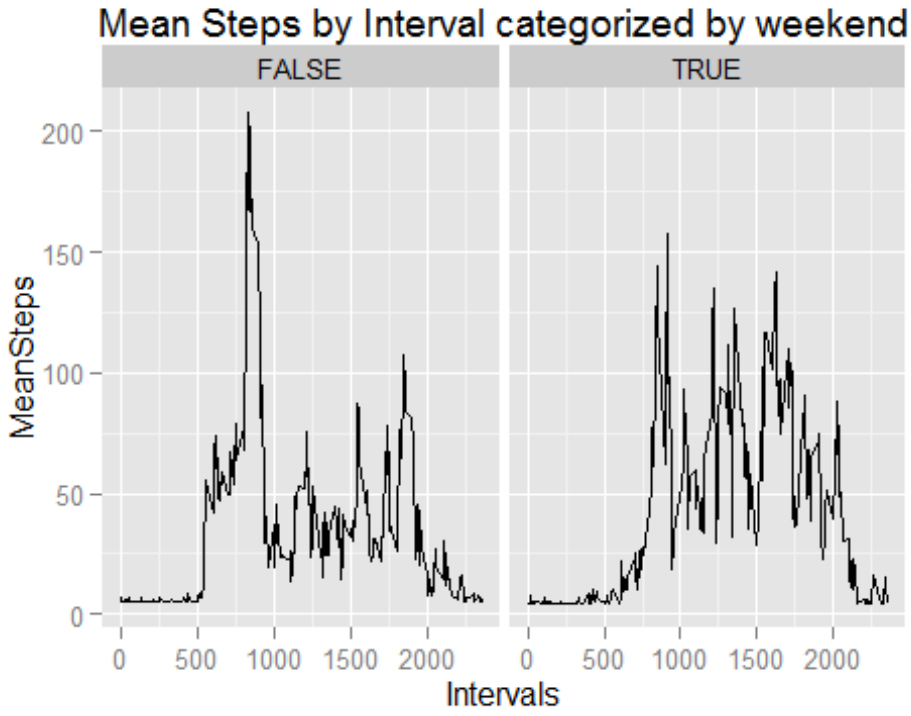
2.Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using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

```
library(dplyr)
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.1.3

## Read the file
setwd("D:/Personal/msworksOthers/201210_DocsAndResearchAndKnowledge/Coursera/data/repdata_data_activity")
ourdata <- read.csv("activity.csv")



ourdata1 <- ourdata
## filling the NA values in steps with  a value
## I could have used mean of the interval or mean of the day, but I preferred to used the mean of all the available steps for all the days to keep it simple and unbiased

ourdata1$steps[is.na(ourdata1$steps)] <- mean(ourdata1$steps, na.rm=TRUE)

## create another df so that we dont mess up the original dataframe
ourdata3 <-ourdata1
## Add another factor column for weekday or weekend
ourdata3$weekend <- factor(weekdays(as.Date(as.character(ourdata3$date),"%Y-%m-%d")) %in% c('Sunday','Saturday'))
levels(ourdata3$weekend)[levels(ourdata3$weekend)=="1"] <- "1"
levels(ourdata3$weekend)[levels(ourdata3$weekend)=="0"] <- "0"
```

```
##use dplyr to group the steps by interval and weekend
by_interval_weekend <- group_by(ourdata3,interval, weekend)

## use dplyr to average the steps per interval
plot_by_interval_weekend<-summarize(by_interval_weekend, mean(steps))
names(plot_by_interval_weekend)[3] <- c("MeanSteps")

## Plot a Mean Steps by Interval categorized by weekend
g <- ggplot(plot_by_interval_weekend, aes(interval, MeanSteps))
p <- g + geom_line()
print(p)
```




```
g + geom_line() + facet_grid(.~weekend) + labs(x = "Intervals") + labs(y =
"MeanSteps") + labs ( title = "Mean Steps by Interval categorized by
weekend")
```


Mean Steps by Interval categorized by weekend
FALSE
TRUE
MeanSteps
200
150
100
50
0
0
500
1000
1500
2000
Intervals

End of Part D