Reproducible Research Project - Daily Steps Analysis

Heather Meylemans

6/12/2019

## Setup the dataset and R for the analysis

First we want to load the appropriate packages to clean the data and generate the plots

library(ggplot2)  
library(plyr)  
library(lattice)

Then we want to read the data in from the working directory

activity <- read.csv("activity.csv")

Next we want to do some processing of the dataset to get a consistent table

activity$day <- weekdays(as.Date(activity$date))  
activity$DateTime<- as.POSIXct(activity$date, format="%Y-%m-%d")

Finally, before we start the analysis, we want to remove null (NA) values from the data set

clean <- activity[!is.na(activity$steps),]

## Task #1

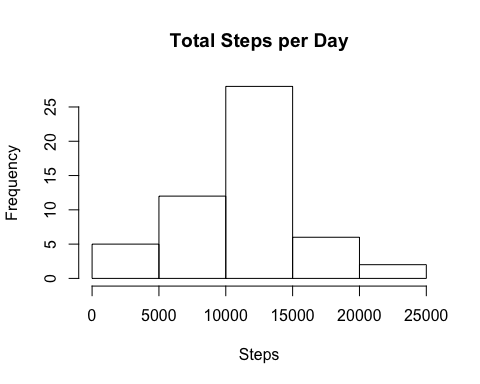
Calculate the number of steps walked each day (mean & median) & create a histogram

First we want to create a table where we sum the values daily:

sumTable <- aggregate(activity$steps ~ activity$date, FUN=sum, )  
colnames(sumTable)<- c("Date", "Steps")

We can use this table to create a histogram of the data

hist(sumTable$Steps, breaks=5, xlab="Steps", main = "Total Steps per Day")



We can also use this table to calculate the mean steps per day

as.integer(mean(sumTable$Steps))

## [1] 10766

And, to calculate the median steps per day

as.integer(median(sumTable$Steps))

## [1] 10765

## Task #2

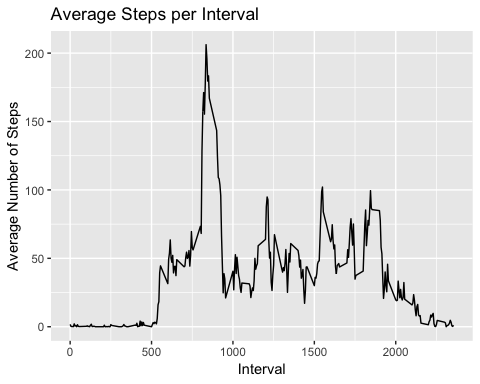
Now we want to look at the daily activity pattern for the entire dataset.

We begin by creating a table of averages for 5 minutes intervals throughout the day.

intervalTable <- ddply(clean, .(interval), summarize, Avg = mean(steps))

Then we can plot this data in a line plot to get an idea of the daily pattern.

p <- ggplot(intervalTable, aes(x=interval, y=Avg), xlab = "Interval", ylab="Average Number of Steps")  
p + geom\_line()+xlab("Interval")+ylab("Average Number of Steps")+ggtitle("Average Steps per Interval")



We can also determine which interval has the maximum number of steps on average.

maxSteps <- max(intervalTable$Avg)  
intervalTable[intervalTable$Avg==maxSteps,1]

## [1] 835

## Task #3

This task looks at imputing the missing values with a value of our choice. For this I chose to use the average steps for the day at the missing given interval

First we look at the number of null (NA) values in the original dataset:

nrow(activity[is.na(activity$steps),])

## [1] 2304

Next we create a dataset that includes just the NA values from the original data.

NAdata<- activity[is.na(activity$steps),]

Next we want to determine the average number of steps on a day during an interval

avgTable <- ddply(clean, .(interval, day), summarize, Avg = mean(steps))

Then we merge the data from above and the NA table we created

newdata<-merge(NAdata, avgTable, by=c("interval", "day"))

We need to Reorder the new data in the same format as clean data set

newdata2<- newdata[,c(6,4,1,2,5)]  
colnames(newdata2)<- c("steps", "date", "interval", "day", "DateTime")

Then, we merge the imputed data into the clean dataset that we were just using in tasks 1 & 2:

AllData <- rbind(clean, newdata2)

Now we can analyze this new dataset and compare it to the answers in task 1

We start by creating the sum of steps per date to compare with step 1

sumTable2 <- aggregate(AllData$steps ~ AllData$date, FUN=sum, )  
colnames(sumTable2)<- c("Date", "Steps")

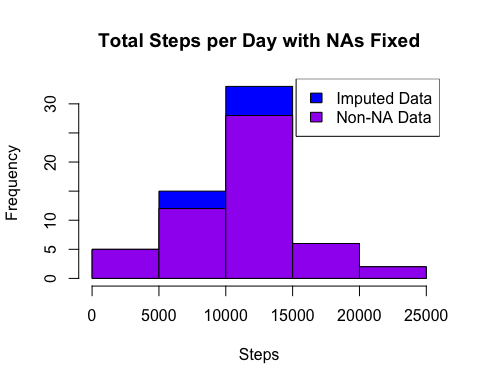
Next we can calculate the mean of the steps in the new data table:

as.integer(mean(sumTable2$Steps))

## [1] 10821

Lastly we can create a histogram to look at the difference between the steps in the table without NA values and this new table with the NA values added in.

hist(sumTable2$Steps, breaks=5, xlab="Steps", main = "Total Steps per Day with NAs Fixed", col="Blue")  
hist(sumTable$Steps, breaks=5, xlab="Steps", main = "Total Steps per Day with NAs Fixed", col="Purple", add=T)  
legend("topright", c("Imputed Data", "Non-NA Data"), fill=c("blue", "purple") )



## Task #4

Lastly, we want to compare the number of steps taken on the weekend days vs. week days

We need to create new category based on the days of the week

AllData$DayCategory <- ifelse(AllData$day %in% c("Saturday", "Sunday"), "Weekend", "Weekday")

Next we want to look at the average steps taken on the weekend days versus the weekdays. To do this we need to separate the table into weekdays and weekends.

intervalTable2 <- ddply(AllData, .(interval, DayCategory), summarize, Avg = mean(steps))

Then we want to plot the data in a panel plot with the intervals on the x axis and then number of steps on the y axis. The two panels are for weekends and weekday averages.

xyplot(Avg~interval|DayCategory, data=intervalTable2, type="l", layout = c(1,2),  
 main="Average Steps per Interval",   
 ylab="Average Number of Steps", xlab="Interval")

