PA1\_Template

Ryan Jolicoeur

January 23, 2016

First load the activity dataset into your working directory and load the packages you will be working with:

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.2.3

library(plyr)

## Warning: package 'plyr' was built under R version 3.2.3

library(knitr)  
opts\_chunk$set(echo = TRUE)  
  
  
library(sqldf)

## Warning: package 'sqldf' was built under R version 3.2.3

## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite

## Loading required package: DBI

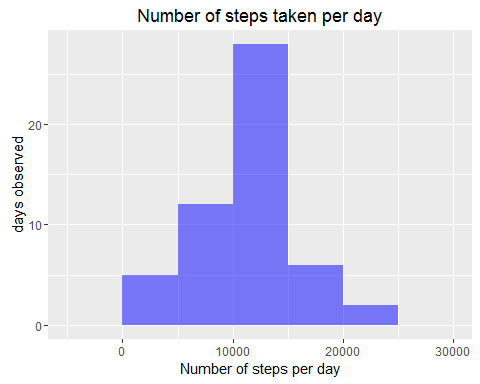
library(tcltk)  
  
Activity<- read.csv('activity.csv', header = TRUE, sep = ",",  
 colClasses=c("numeric", "character", "numeric"))  
  
  
#Lets go ahead and transform the data frame columns "date" into a date format and interval into a character format  
  
Activity$datetime <- as.POSIXct(Activity$date, format = "%Y-%m-%d")  
Activity$weekday <- weekdays(as.Date(Activity$date))

From here lets utilize the sqldf package to find the sum of steps by days. I utilize sqldf package to aggregate the data together as well as to remove the missing values here:

Remove\_NAS <- Activity[!is.na(Activity$steps),]  
daily\_steps <- sqldf("select date, sum(steps) as steps from Remove\_NAS group by date")  
colnames(daily\_steps) <- c("date", "steps")

Now we will construct the histogram of average steps by day:

steps <- ggplot(daily\_steps, aes(steps))  
steps <- steps+geom\_histogram(binwidth=5000,fill="blue", alpha = 0.5)+labs(y="days observed", x="Number of steps per day",   
 title = "Number of steps taken per day")  
   
plot(steps)



Now that we have aggregated the steps by days and created daily\_steps we can calculate the mean and the median:

average\_steps <- mean(daily\_steps$steps)  
round(average\_steps)

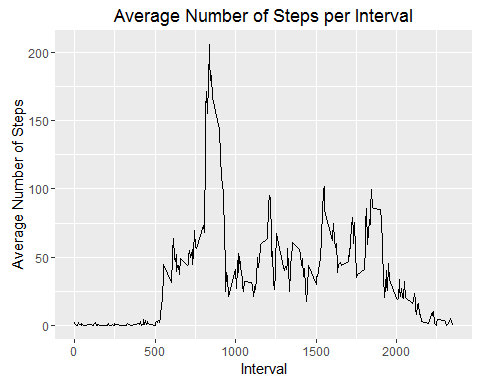
## [1] 10766

median\_steps <- median(daily\_steps$steps)  
median\_steps

## [1] 10765

Next lets create a function to find the number of steps taken per five minute interval early in the morning:

intervals <- ddply(Activity, .(interval), summarize, Avg = mean(steps, na.rm=T))  
 avg\_steps <- ggplot(intervals, aes(x=interval, y=Avg), xlab="Interval", ylab="Average number of steps")  
 avg\_steps + geom\_line()+xlab("Interval")+ylab("Average Number of Steps")+ggtitle("Average Number of Steps per Interval")



Next we want to determine the maximum number of steps per interval:

Max <- max(intervals$Avg)  
intervals[intervals$Avg==Max,1]

## [1] 835

as.integer(Max) # This tells us which interval contains the max steps and what that average max is in

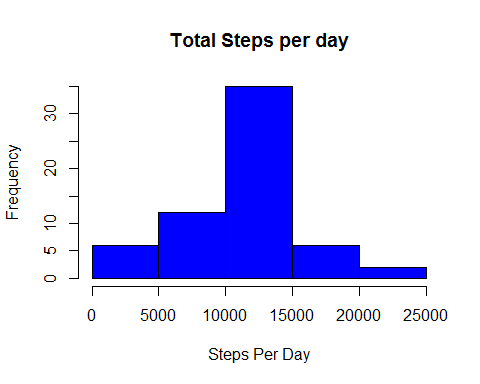
## [1] 206

Now we need to impute for our missing step to remove all the NAS by imputing our data.

NAS <- sum(!complete.cases(Activity))  
  
#Similar to the interval calculations, we need to take the average steps per weekday and intervals  
imput <- transform(Activity, steps=ifelse(is.na(Activity$steps), intervals$Avg[match(Activity$interval,intervals$interval)],Activity$steps))  
  
  
imput[as.character(imput$date)=="2012-10-01",1]<- 0  
  
NAS2 <- sum(!complete.cases(imput))  
NAS2

## [1] 0

#Again here i will utilize the sqldf function to aggreate the data  
steps\_per\_day<- sqldf("select date, sum(steps) as steps from imput where steps >= 0 group by date")  
  
hist(steps\_per\_day$steps, col="blue", breaks=5, main= "Total Steps per day", xlab="Steps Per Day")



From here we need to be able to calculate the mean and median difference of steps taken from when the missings were removed:

mean\_difference <- mean(steps\_per\_day$steps) - mean(daily\_steps$steps)  
median\_difference <- median(steps\_per\_day$steps) - median(daily\_steps$steps)  
  
mean\_difference

## [1] -176.4949

median\_difference

## [1] 1.188679

summary(steps\_per\_day$steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0 8918 10770 10590 12810 21190

This result was a little suprising the variation between the mean with and without the NAS included. After running through the result a few times I didn't see any changes that I would make to the aggregation function once the NAS were removed from our "steps per day calculation".

Next is where I utilize the sqldf to reflect the case statement here. If the days are coded as a Saturday or a Sunday then they are a weekend. If not then it is a weekday.

Day\_indicator <- sqldf("select \*, case  
 when weekday like '%Saturday%' then 'Weekend'  
 when weekday like '%Sunday%' then 'Weekend'  
 else 'Weekday' end Indicator from imput")  
  
table(Day\_indicator$Indicator)

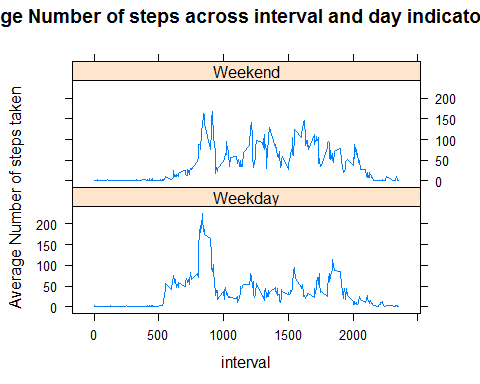
##   
## Weekday Weekend   
## 12960 4608

Here I will load the lattice package to be able to graph the average steps by weekday and weekend by panel graphing:

library(lattice)  
  
Day\_type\_comparison <- ddply(Day\_indicator,.(interval,Indicator), summarize, Avg=mean(steps))

Now we will be able to plot these values to see the differences in weekends vs weekdays:

xyplot(Avg~interval|Indicator, data=Day\_type\_comparison,   
 type="l", layout=c(1,2), main="Average Number of steps across interval and day indicator type",   
 ylab = "Average Number of steps taken", xlab="interval")



From the data we are able to see that yes there is a difference. Typically during the week most people with sedentary office jobs don't move around much. During the weekend they tend to move around more frequently throughout the day.