PA1\_template.Rmd

seh

November 15, 2015

# This is the markdown file for Programming Assignment One in Coursera's Reproducile Research Class.

### Show any code that is needed to load the data:

activity<-read.csv("/Users/humbertsanders/Downloads/activity.csv", colClasses = c("numeric", "character", "numeric"))

### Show any code that is needed to process and transform the data. Here, that means formatting the date and interval columns into a format that R can understand more easily.

activity$interval<-sprintf("%04d", activity$interval)  
activity$interval<-format(strptime(activity$interval, "%H%M"), format = "%H:%M")  
activity$date<-as.Date(activity$date, "%Y-%m-%d")

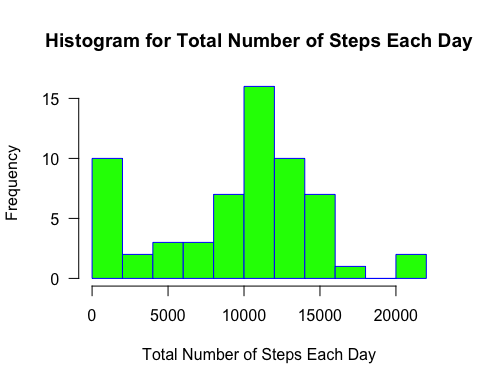
### Calculate the total number of steps taken per day.

totalperday<-tapply(activity$steps, activity$date, sum, na.rm=TRUE)  
totalperday<-as.data.frame(totalperday)  
print(totalperday)

## totalperday  
## 2012-10-01 0  
## 2012-10-02 126  
## 2012-10-03 11352  
## 2012-10-04 12116  
## 2012-10-05 13294  
## 2012-10-06 15420  
## 2012-10-07 11015  
## 2012-10-08 0  
## 2012-10-09 12811  
## 2012-10-10 9900  
## 2012-10-11 10304  
## 2012-10-12 17382  
## 2012-10-13 12426  
## 2012-10-14 15098  
## 2012-10-15 10139  
## 2012-10-16 15084  
## 2012-10-17 13452  
## 2012-10-18 10056  
## 2012-10-19 11829  
## 2012-10-20 10395  
## 2012-10-21 8821  
## 2012-10-22 13460  
## 2012-10-23 8918  
## 2012-10-24 8355  
## 2012-10-25 2492  
## 2012-10-26 6778  
## 2012-10-27 10119  
## 2012-10-28 11458  
## 2012-10-29 5018  
## 2012-10-30 9819  
## 2012-10-31 15414  
## 2012-11-01 0  
## 2012-11-02 10600  
## 2012-11-03 10571  
## 2012-11-04 0  
## 2012-11-05 10439  
## 2012-11-06 8334  
## 2012-11-07 12883  
## 2012-11-08 3219  
## 2012-11-09 0  
## 2012-11-10 0  
## 2012-11-11 12608  
## 2012-11-12 10765  
## 2012-11-13 7336  
## 2012-11-14 0  
## 2012-11-15 41  
## 2012-11-16 5441  
## 2012-11-17 14339  
## 2012-11-18 15110  
## 2012-11-19 8841  
## 2012-11-20 4472  
## 2012-11-21 12787  
## 2012-11-22 20427  
## 2012-11-23 21194  
## 2012-11-24 14478  
## 2012-11-25 11834  
## 2012-11-26 11162  
## 2012-11-27 13646  
## 2012-11-28 10183  
## 2012-11-29 7047  
## 2012-11-30 0

Make a histogram of the total number of steps taken per day.

totalperday$totalperday<-as.vector(totalperday$totalperday)  
hist(totalperday$totalperday, main = "Histogram for Total Number of Steps Each Day", xlab = "Total Number of Steps Each Day", border = "blue", col = "green", xlim = c(0,22000), las = 1, breaks = 8)



### Calculate and report the mean and median of the total number of steps taken each day.

m<-mean(totalperday$totalperday)  
print(m)

## [1] 9354.23

m2<-median(totalperday$totalperday)  
print(m2)

## [1] 10395

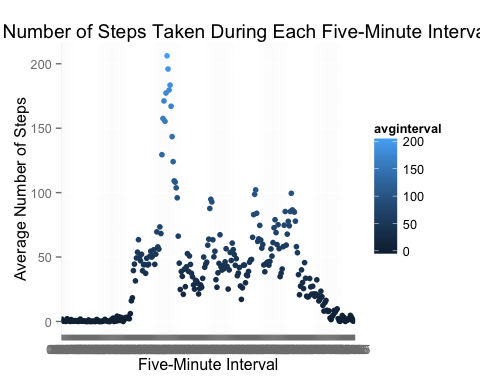
### Make a time series plot of the five-minute interval (x-axis) and the average number of steps taken, averaged across all days.

#### I will use ggplot2() instead of plot() because ggplot2() works more easily with tapply().

install.packages("ggplot2", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/rn/\_gkwqm210252xdk1pfsqfjlr0000gn/T//Rtmpn90J76/downloaded\_packages

library(ggplot2)  
avginterval<-tapply(activity$steps, activity$interval, mean, na.rm=TRUE)  
avginterval<-as.data.frame(avginterval)  
g<-ggplot(avginterval, aes(row.names(avginterval), avginterval))  
p<-g + geom\_point(aes(color=avginterval)) + labs(x= "Five-Minute Interval", y = "Average Number of Steps", title = "Average Number of Steps Taken During Each Five-Minute Interval")  
par(mar = c(5,4,1,1), las=1)  
print(p)



### Which five-minute interval, on average across all days in the dataset, contains the maximum number of steps? The answer will be the first interval listed in the below dataframe.

head(sort(avginterval$avginterval, decreasing = TRUE))

## 08:35 08:40 08:50 08:45 08:30 08:20   
## 206.1698 195.9245 183.3962 179.5660 177.3019 171.1509

### Calculate and report the total number of missing values in the dataset (i.e., the total number of rows with NAs). (I already checked the date and interval columns and verified that they do not also have NAs)

na<-sum(is.na(activity$steps))  
print(na)

## [1] 2304

### 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. Then create a new dataset that is equal to the original dataset but with the missing data filled in.

#### I will calculate the total number of rows in the activity dataframe in order to apply a for loop to replace each of the missing NAs. I will also use the aggregate() function to come up with the median for each five-minute interval,as tapply() has limitations for this purpose.

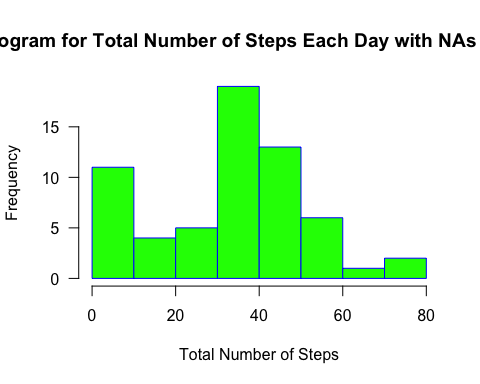
nrow(activity)

## [1] 17568

mediansteps<-aggregate(steps ~ interval, data = activity, FUN=median, na.rm=TRUE)  
replaceNA<-vector()  
for (i in 1:17568) {  
 nobs<-activity[i,]  
 if (is.na(nobs$steps)) {  
 steps<-subset(mediansteps, mediansteps$interval == nobs$interval)$steps  
 } else {  
 steps<-nobs$steps  
 }  
 replaceNA<-c(replaceNA, steps)  
 }  
new\_activity<-activity  
new\_activity$steps<-replaceNA

### 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.

totalperday\_new<-aggregate(steps ~ date, data = new\_activity, FUN=mean)  
hist(totalperday\_new$steps, main = "Histogram for Total Number of Steps Each Day with NAs Imputed", xlab = "Total Number of Steps", border = "blue", col = "green", xlim = c(0, 80), las=1, breaks = 8)



### Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

answer<-c("Yes -- there are fewer steps clustered in the middle of the distribution and more on the edges of the distribution")  
print(answer)

## [1] "Yes -- there are fewer steps clustered in the middle of the distribution and more on the edges of the distribution"

### Are there differences in activity patterns between weekdays and weekends? Use the dataset with the filled-in missing values for this part. 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.

new\_activity$dayofweek<-weekdays(as.Date(new\_activity$date))  
day<-vector()  
for (i in 1:17568) {  
 if (new\_activity$dayofweek[i] == "Saturday") {  
 day[i]<-"Weekend"  
 }else if (new\_activity$dayofweek[i] == "Sunday") {  
 day[i]<-"Weekend"  
 }else {  
 day[i]<-"Weekday"  
 }  
 }  
 new\_activity$day<-day

### Make a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

total<-aggregate(steps ~ interval + day, data=new\_activity, mean)  
g<-ggplot(total, aes(interval, steps)) + geom\_point(color= "blue") + facet\_grid(day~.) + theme\_bw() + labs(title = "Average Number of Steps Taken Per Interval, Weekends Versus Weekdays", x = "Five-Minute Interval", y = "Average Number of Steps Taken")  
par(mar = c(5,4,1,1), las=1)  
print(g)

