Daily\_Steps\_Taken - Data Analysis

Michelsone Presendieu

July 28, 2018

1.Loading and preprocessing the data We set the working directory followed by downloading the dataset from its url and unzipping the file to “step\_data.csv”. The data comes from Roger Pengs github account.

setwd("~/Coursera/Reproducibile Research/Week\_2/RepData\_PeerAssessment1")  
url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"  
destfile <- "step\_data.zip"  
download.file(url, destfile)  
unzip(destfile)  
act\_data <- read.csv("activity.csv", sep = ",", stringsAsFactors = FALSE)

Explorarory Analysis w/ variable names and the structure of the file are given by

act\_data$day <- weekdays(as.Date(act\_data$date))  
act\_data$date\_time <- as.POSIXct(act\_data$date, format= "%Y-%m-%d")  
names(act\_data)

## [1] "steps" "date" "interval" "day" "date\_time"

str(act\_data)

## 'data.frame': 17568 obs. of 5 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : chr "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...  
## $ interval : int 0 5 10 15 20 25 30 35 40 45 ...  
## $ day : chr "Monday" "Monday" "Monday" "Monday" ...  
## $ date\_time: POSIXct, format: "2012-10-01" "2012-10-01" ...

##removing NAS from the data  
act\_data1 <- act\_data[!is.na(act\_data$steps),]

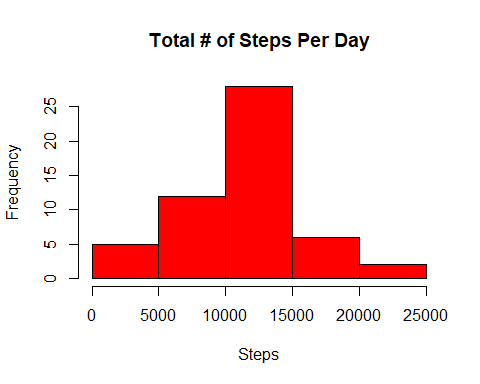
## What is mean total number of steps taken per day?

Calculate the total number of steps taken per day

##summarizing total steps per date  
act\_table\_sum <- aggregate(act\_data1$steps ~ act\_data1$date, FUN = sum)  
colnames(act\_table\_sum) <- c("Date", "Steps")

Make a histogram of the total number of steps taken each day #make a histogram of the total # of steps taken each day

hist(act\_table\_sum$Steps, main ="Total # of Steps Per Day", xlab = "Steps", col = "red", breaks = 5)



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

#calculate the mean of the total # of steps taken per day  
as.integer(mean(act\_table\_sum$Steps))

## [1] 10766

#calculate the median of the total # of steps taken per day  
as.integer(median(act\_table\_sum$Steps))

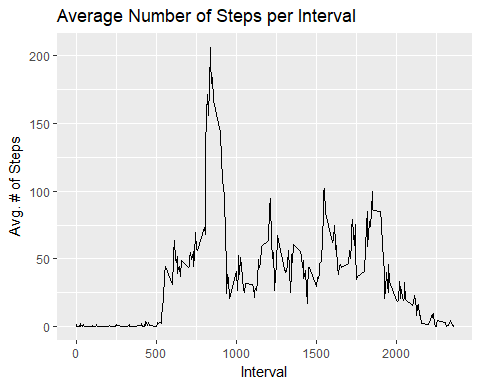
## [1] 10765

The average # of steps taken each day was 10766. The median # of steps taken each day was 10765.

## What is the average daily activity pattern?

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)

##load libraries plyr & ggplot  
library(plyr)  
library(ggplot2)  
  
##removing NAS from the data  
act\_data1 <- act\_data[!is.na(act\_data$steps),]  
inv\_table <- ddply(act\_data1, .(interval), summarize, Avg = mean(steps))  
  
##create line plot og avg. # of step per interval  
time\_plot <- ggplot(inv\_table, aes(x = interval, y= Avg), xlab = "Interval", ylab= "Avg. # of Steps")  
time\_plot + geom\_line() + xlab("Interval") + ylab ("Avg. # of Steps") + ggtitle("Average Number of Steps per Interval")

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

##maximum steps by interval  
max\_steps <- max(inv\_table$Avg)  
round(max\_steps, digits = 0)

## [1] 206

##which interval contains the maximum avg. # of steps  
inv\_table[inv\_table$Avg == max(max\_steps),1]

## [1] 835

The maximum number of steps for a 5-minute interval was 206 steps.

The 5-minute interval which had the maximum number of steps was the 835 interval.

## Imputing missing values

Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

#calculate and report the total number of missing values  
sum(is.na(act\_data))

## [1] 2304

So the original data set has 2304 rows with missing data.

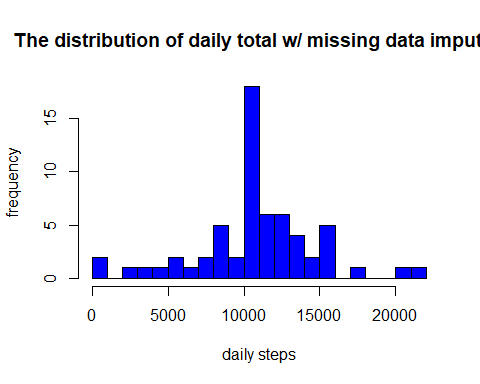
We use a simple strategy for filling in all of the missing values in the dataset. If a 5-minute interval has missing value, we use the mean for that 5-minute interval.

We create a new data frame df\_impute that is equal to the original dataset but with the missing data filled in (using mean for that interval for imputation):

df\_act <- act\_data  
miss\_df <- is.na(df\_act$steps)  
int\_avg <- tapply(df\_act$steps, df\_act$interval, mean, na.rm = TRUE, simplify = T)  
df\_act$steps[miss\_df] <- int\_avg[as.character(df\_act$interval[miss\_df])]

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

new\_daily <- tapply(df\_act$steps, df\_act$date, sum, na.rm = TRUE, simplify = T)  
hist(x = new\_daily, col = "blue", breaks =20, xlab = "daily steps", ylab ="frequency", main = "The distribution of daily total w/ missing data imputed")



#mean of new\_daily  
round(mean(new\_daily), digits = 0)

## [1] 10766

#median of new\_daily  
round(median(new\_daily), digits = 0)

## [1] 10766

Based on the imputed data set, the new mean is 10766 and the new median is 10766 . Compare with the original mean 10766 and median 10765 , the mean doesn’t change, and the median has a small change. Thus, the new median becomes identical to the mean. One possible explanation is that when we fill the missing data for the intervals, we use means for intervals, so we have more data close or identical to the means, and median is shifted and becomes identical to the mean. The overall shape of the distribution has not changed.

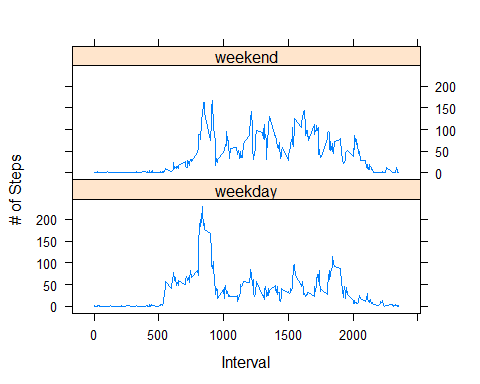
## Are there differences in activity patterns between weekdays and weekends?

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.

#create a helper functions to decide if a day is a week day or not  
is\_weekday <- function(d) {  
 wd <- weekdays(d, abbreviate= FALSE)  
 ifelse(wd == "Saturday" | wd == "Sunday", "weekend", "weekday")  
}  
wm <- sapply(df\_act$date\_time, is\_weekday)  
df\_act$wk <- as.factor(wm)  
  
##make a panel plot containing a time series plot  
wk\_df <- aggregate(steps ~ wk + interval, data = df\_act, FUN = mean)  
  
library(lattice)

## Warning: package 'lattice' was built under R version 3.4.4

xyplot(steps ~ interval | factor(wk), layout = c(1,2), xlab="Interval", ylab= "# of Steps", type ="l", data= wk\_df)



Yes, the step activity trends are different based on whether the day occurs on a weekend or not. This may be due to people having an increased opportunity for activity beyond normal work hours for those who work during the week.