Project1

# Reproducible Research Project1

#Set the working directory and load necessary libararies  
setwd("/Volumes/DiskStation1/jhufton/MyDownload/Coursera/DataScientistsToolbox/ReproducableResearch/CourseProject1/RepData\_PeerAssessment1")  
library(plyr)  
library(ggplot2)  
library(lubridate)

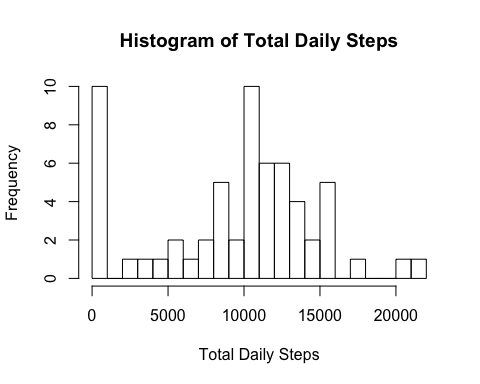
##   
## Attaching package: 'lubridate'  
##   
## The following object is masked from 'package:plyr':  
##   
## here

library(xtable)  
  
#Read in the data set, set NA values to 0  
activity.raw <- read.csv("./activity.csv", header=TRUE)  
activity <- activity.raw  
activity[is.na(activity)] <- 0

## Histogram of Steps Taken by Day

Below is a histogram based on the total number of steps taken for each day in the dataset.

#Aggregate the steps for each interval in a day to a daily number of total   
#steps for each day  
daily.steps <- ddply(activity, .(date), summarize, daily\_steps =   
 sum(steps, na.rm = TRUE))  
   
max\_daily\_steps <- as.integer(max(daily.steps$daily\_steps))  
  
#Plot a histogram of daily steps  
hist(daily.steps$daily\_steps, breaks=25, main="Histogram of Total Daily Steps", xlab="Total Daily Steps")



Note that there are eight days in the dataset for which there is no step data, and two days with near zero number of steps. This results in a large number of days for which there is a 0 or near 0 number of steps. In this data set the maximum sum of daily steps is 21194 (note this value is computed and displayed in-line), which agrees with the frequency counts at 20,000 steps or more in the histogram.

## Calculation of Mean and Median Daily Steps

### Mean Daily Steps

mean\_daily\_steps <- as.integer(mean(daily.steps$daily\_steps))  
#NOTE: This value appears in-line in the text below using `r mean\_daily\_steps`  
# see footnote

Using the sum of daily steps data calculated in the previous section, we find the mean number of daily steps is 9354 [[1]](#footnote-26).

### Median Daily Steps

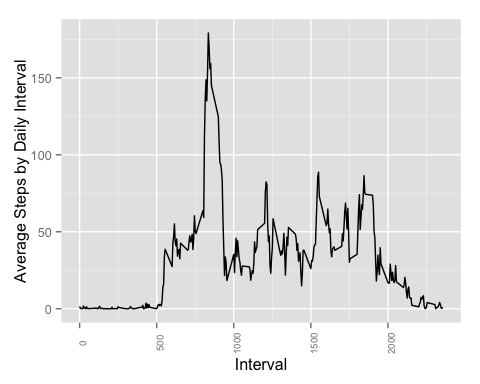
median\_daily\_steps <- as.integer(median(daily.steps$daily\_steps))  
#NOTE: This value appears in-line in the text below using `r median\_daily\_steps`  
# see footnote

Using the sum of daily steps data calculated in the previous section, we find the median number of daily steps is 10395 [[2]](#footnote-28).

### Average Daily Activity Patterm

Below is a graph showing the average number of steps for each interval of the day (computed as an average number of steps across corresponding intervals for all days in the data set).

average.interval.steps <- ddply(activity, .(interval), summarize,   
 avg\_steps = mean(steps, na.rm = TRUE))  
ggplot(average.interval.steps, aes(x=interval, y=avg\_steps)) +   
 geom\_line(stat="identity") +   
 theme(axis.text.x = element\_text(angle = 90, size = rel(0.75))) +   
 ylab("Average Steps by Daily Interval") + xlab("Interval")



The maximum average number of steps per interval and the daily interval where the maximum occurs is:

average.interval.steps[average.interval.steps$avg\_steps ==   
 max(average.interval.steps$avg\_steps),]

## interval avg\_steps  
## 104 835 179.1311

Since the interval column counts in 5 minute increments, and the value 100 corresponds to 60 minutes, the interval 835 corresponds to 8 hours and 35 minutes into the day, or 8:35AM.

### Imputing Missing Values

#### Number of Missing Values in the Raw Dataset

missing\_values <- nrow(activity.raw[is.na(activity.raw$steps), ])  
#NOTE: This value appears in-line in the text below using `r missing\_value`  
# see footnote

The number of intervals in the orginal data set with missing values is 2304 [[3]](#footnote-33).

From observation of the numbers of steps per day in the first graph (Histogram of Steps Taken by Day), we can see that these missing values correspond to the 8 full days of missing step values in the dataset.

Inspection of the data for the mean number of daily steps shows these days as having 0 average steps. That the missing values are for 8 full days of data can also be confirmed by inspecting the raw data file as well.

Days with NA values for step data are:

\* 2012-10-01  
\* 2012-10-08  
\* 2012-11-01  
\* 2012-11-04  
\* 2012-11-09  
\* 2012-11-10  
\* 2012-11-14  
\* 2012-11-30

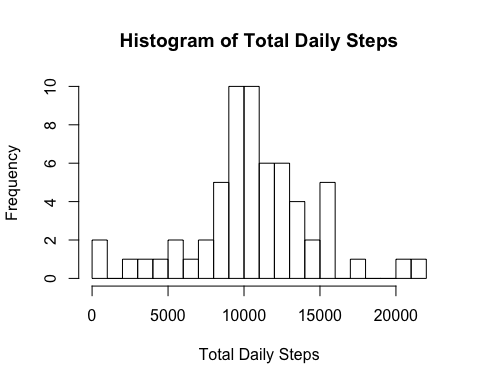
Since we have calculated an average number of steps by interval using all the available daily data in the dataset, we can impute (i.e. replace) the missing daily data with that calculated daily average data and recompute the mean and median figures for each day by interval.

#Replace the missing step data for the days that have no data in the original  
#dataset  
dates <- c("2012-10-01", "2012-10-08", "2012-11-01", "2012-11-04", "2012-11-09",   
 "2012-11-10", "2012-11-14", "2012-11-30")  
for (d in dates) {   
 activity[activity$date == d, 1] <- average.interval.steps[ ,2]  
}

### Histogram of Steps Taken by Day Using Imputed Data

Below is a histogram based on the total number of steps taken for each day in the dataset, now using the imputed data added in the preceding step.

#Aggregate the steps for each interval in a day to a daily number of total   
#steps for each day  
daily.steps <- ddply(activity, .(date), summarize, daily\_steps =   
 sum(steps, na.rm = TRUE))  
max\_imputed\_daily\_steps <- as.integer(max(daily.steps$daily\_steps))  
  
#Plot the total   
hist(daily.steps$daily\_steps, breaks=25, main="Histogram of Total Daily Steps", xlab="Total Daily Steps")



We now re-calculate the mean and median steps for each day in the dataset, this time including the imputed data for the eight days missing data in the original dataset.

Note that there remain two days with near zero number of steps. This results in two days for which there is a near 0 number of steps. In this data set the maximum sum of daily steps is 21194 (note this value is computed and displayed in-line), which agrees with the frequency counts at 20,000 steps or more in the histogram.

### Mean Daily Steps with Imputed Missing Values

mean\_imputed\_daily\_steps <- as.integer(mean(daily.steps$daily\_steps))  
#NOTE: This value appears in-line in the text below using `r mean\_imputed\_daily\_steps`  
# see footnote

Using the sum of daily steps data calculated in the previous section which now includes the imputed data substitued for the days originally without data, we find the mean number of daily steps is 10581[[4]](#footnote-37).

This compares to a mean number of daily steps of 9354[[5]](#footnote-38) before the imputed data was used to replace the missing data in the original data set.

### Median Daily Steps with Imputed Missing Values

#Create a data frame with the median daily steps from the activity data  
#with the imputed data replacing missing data in the oringal dataset  
median\_imputed\_daily\_steps <- as.integer(median(daily.steps$daily\_steps))  
#NOTE: This value appears in-line in the text below using `r median\_imputed\_daily\_steps`  
# see footnote

Using the sum of daily steps data calculated in the previous section which now includes the imputed data substitued for the days originally without data, we find the median number of daily steps is 10395[[6]](#footnote-40).

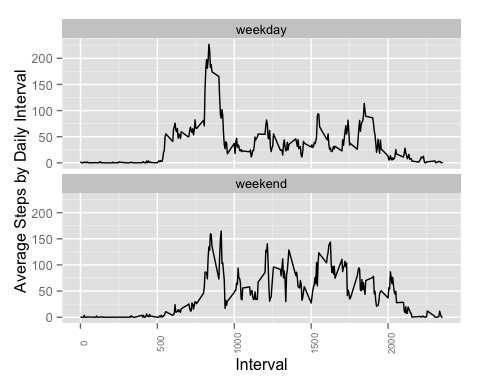
This compares to a median number of daily steps of 10395 [[7]](#footnote-41) before the imputed data was used to replace the missing data in the original data set.

### Weekday vs. Weekend Comparison

The next set of graphs shows a comparison of the average number of steps by daily interval between weekdays and weekend days.

The weekday figure closely resembles the first graph (Histogram of Steps Taken by Day) for the full daily average which encompassed both weekend days and weekdays, while the weekend figure shows activity spread more evenly through out the day.

#Add a new column showing whether the date is a weekday or weekend day  
activity$DoW <-   
 with(activity, ifelse(weekdays(as.POSIXct(date)) == "Monday", "weekday",  
 ifelse(weekdays(as.POSIXct(date)) == "Tuesday", "weekday",  
 ifelse(weekdays(as.POSIXct(date)) == "Wednesday", "weekday",  
 ifelse(weekdays(as.POSIXct(date)) == "Thursday", "weekday",  
 ifelse(weekdays(as.POSIXct(date)) == "Friday", "weekday",  
 ifelse(weekdays(as.POSIXct(date)) == "Saturday", "weekend",   
 "weekend")))))) )  
  
#Convert the DoW column to a factor  
activity$DoW <- as.factor(activity$DoW)  
  
#Recalculate the average daily interval steps data frame using the modified   
#activity information  
average.interval.steps <- ddply(activity, .(interval, DoW), summarize,   
 avg\_steps = mean(steps, na.rm = TRUE))  
  
#Create a two panel plot showing average steps per daily interval for   
#weekdays and weekend days  
ggplot(average.interval.steps, aes(x=interval, y=avg\_steps)) +   
 geom\_line(stat="identity") +   
 theme(axis.text.x = element\_text(angle = 90, size = rel(0.75))) +   
 ylab("Average Steps by Daily Interval") + xlab("Interval") +   
 facet\_wrap(~DoW, ncol=1)



#### Footnotes:

1. Note this value is computed and displayed in-line using the 'r mean\_daily\_steps' syntax. [↑](#footnote-ref-26)
2. Note this value is computed and displayed in-line using the 'r median\_daily\_steps' syntax. [↑](#footnote-ref-28)
3. Note this value is computed and displayed in-line using the 'r missing\_values' syntax. [↑](#footnote-ref-33)
4. Note this value is computed and displayed in-line using the 'r mean\_imputed\_daily\_steps' syntax. [↑](#footnote-ref-37)
5. Note this value is computed and displayed in-line using the 'r mean\_daily\_steps' syntax. [↑](#footnote-ref-38)
6. Note this value is computed and displayed in-line using the 'r median\_imputed\_daily\_steps' syntax. [↑](#footnote-ref-40)
7. Note this value is computed and displayed in-line using the 'r median\_daily\_steps' syntax. [↑](#footnote-ref-41)