ActivityDataProcessing

Alexander Connelly

Thursday, July 16, 2015

Introduction:

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the "quantified self" movement -- a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

Data:

The data for this assignment can be downloaded from the course web site:

Dataset: Activity monitoring data [52K] The variables included in this dataset are:

steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)

date: The date on which the measurement was taken in YYYY-MM-DD format

interval: Identifier for the 5-minute interval in which measurement was taken

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

Assignment:

We are going to process this activity data collected by completing the following, analysing, and answering the following questions:

- **Loading and preprocessing the data**
- **What is mean total number of steps taken per day?**
- **What is the average daily activity pattern?**
- **Imputing Missing Values**

- **Are there differences in activity patterns between weekdays and wekend?**

These questions will be elaborated on in a "Question and Answer" format where te details or requirments of the questions will be elaborated on in text then answered via "code chunks" in R, finally conclusions answering these questions in text will follow.

Loading and Preprocessing the Data:

```
knitr::opts_chunk$set(echo=TRUE)
library(lubridate)
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##Read Data
maindata<-read.csv("activity.csv",header=TRUE,na.strings='NA')
##Alter Date and Time Columns to Be in Date and Time Format from Factor
maindata$date<-ymd(maindata$date)
```

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

1. Make a histogram of the total number of steps taken each day

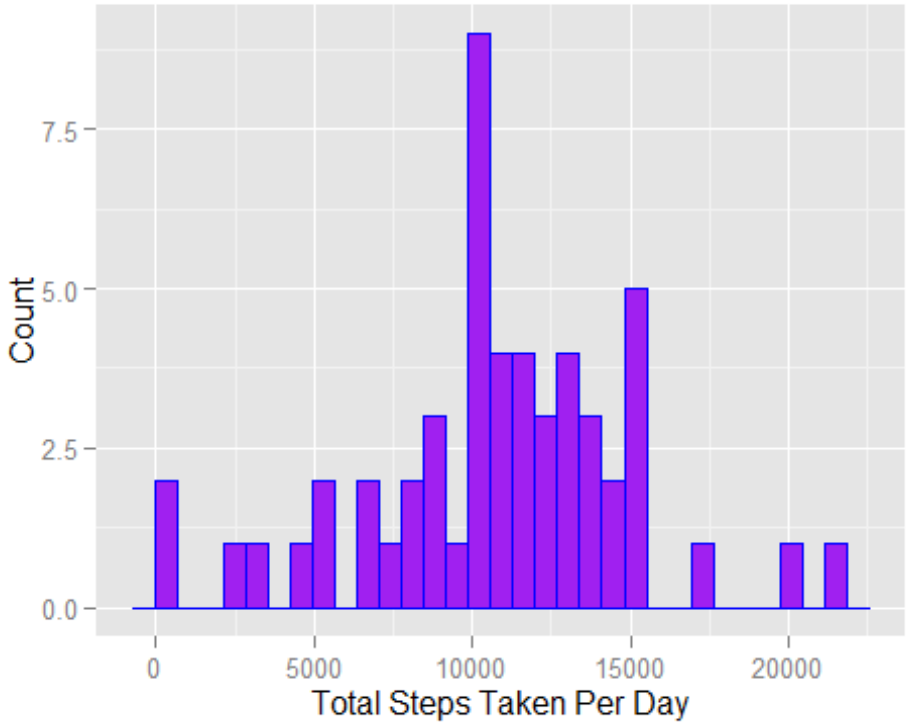
2. Calculate and report the **mean** and **median** total number of steps taken per day

```
library(ggplot2)
library(dplyr)
#Format Data
maindata<-tbl_df(maindata)
maindata$date<-as.character(maindata$date)
#Organize
maindata<-group_by(maindata,date)
steps.day<-summarise(maindata,StepsPerDay=sum(steps))
#find mean and median steps per day, ignoring NA
mean.steps<-mean(steps.day$StepsPerDay,na.rm=TRUE)
```

```
median.steps<-median(steps.day$StepsPerDay,na.rm = TRUE)

qplot(steps.day$StepsPerDay,geom="histogram", xlab="Total Steps Taken Per Day
", ylab="Count",fill=I("purple"),col=I("blue"))

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust thi
s.
```

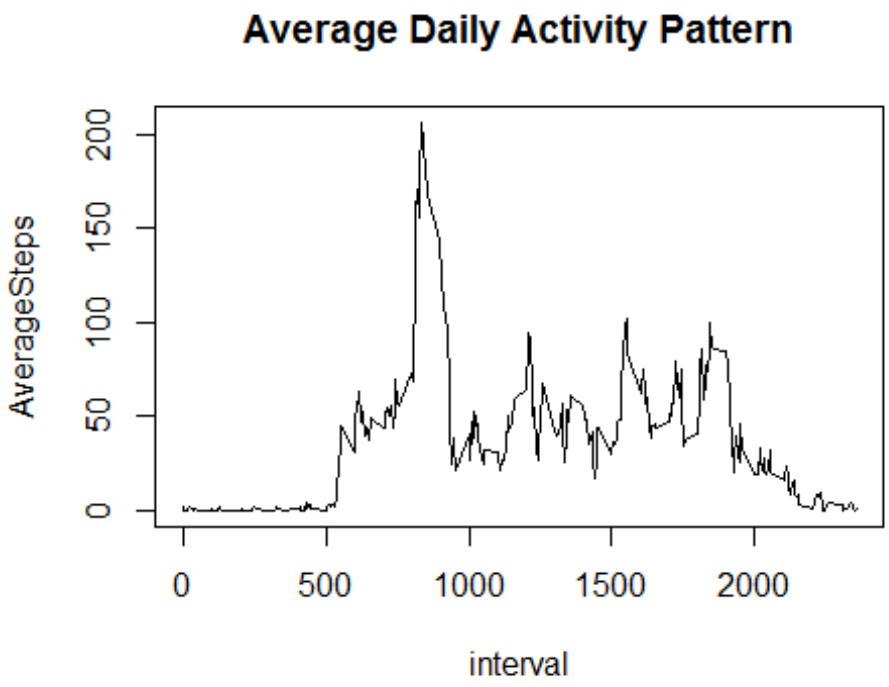


Here we can see the histogram or distribution of the total number of steps taken per day during the study period. We calculated the average steps per day to have a **mean** of **1.076618910^{4}** steps per day and a **median** of **10765** steps per day. We can see evidence of this being the case in the large spike of total steps taken per day just above 10,000 steps.

What is the average daily activity pattern?

1. 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)
2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
#Organize
maindata<-group_by(maindata,interval)
mean.interval<-summarise(maindata,AverageSteps=mean(steps,na.rm=TRUE))
#Plot with graphics base commands:
plot(mean.interval,type="l",main='Average Daily Activity Pattern')
```




```
#Find Max
max.interval<-max(mean.interval$AverageSteps,na.rm=TRUE)
#Found interval by looking at data in csv.
interval<-835
```

As we can see from the plot the average steps taken over the time interval of a day taken over the duration of the study. This has a fairly typical distribution of a person's walking patterns because the beginning of the day is when the device is not being used. Upon waking up, and then some mid day activity that consistently occurs around the interval of **835** where the max average number of steps taken per day is **206.1698113.**

My guess? The user got up to eat lunch and walk around doing that every day at the same time. While other activites using the most steps were staggered depending on the days demands. We see average steps drop down at the end of the day when in activity due to sitting around at night is common.

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

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

2. 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.

3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

4. 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. 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?

```
#Count Number of NA's
count.na<-sum(is.na(maindata$steps))
ungroup(maindata)
```

```
## Source: local data frame [17,568 x 3]
##
##    steps       date interval
## 1     NA 2012-10-01        0
## 2     NA 2012-10-01        5
## 3     NA 2012-10-01       10
## 4     NA 2012-10-01       15
## 5     NA 2012-10-01       20
## 6     NA 2012-10-01       25
## 7     NA 2012-10-01       30
## 8     NA 2012-10-01       35
## 9     NA 2012-10-01       40
## 10    NA 2012-10-01       45
## ..   ...        ...      ...
```

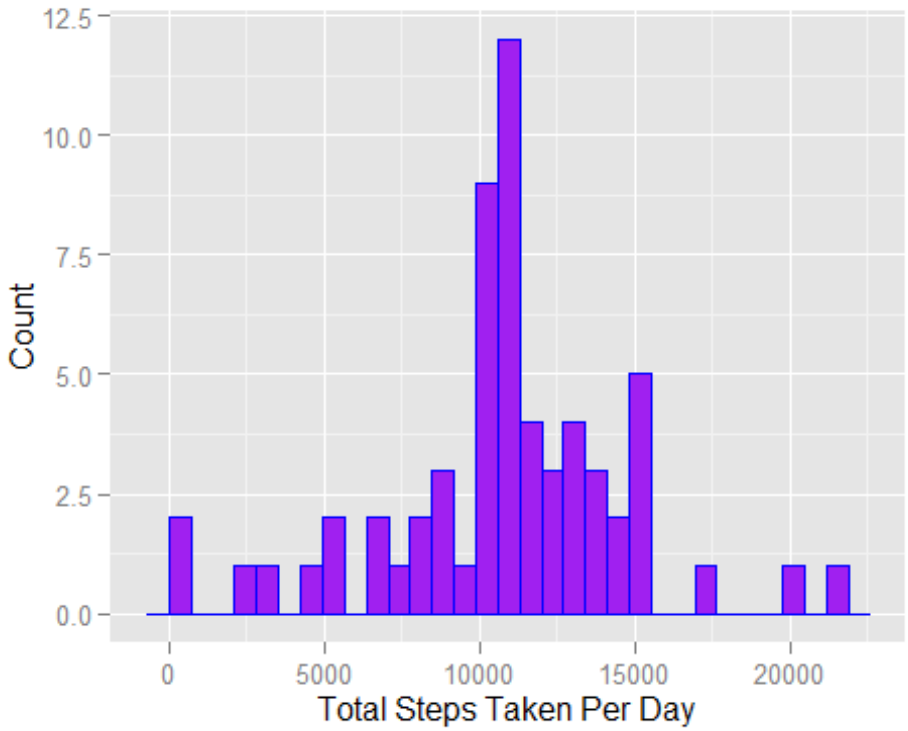
```
new.data<-maindata
#Use Match to insert the average steps for that day based on the average for
the over all interval # it corresponds to.
index<-is.na(maindata$steps)
new.data$steps[index==TRUE]<-mean.interval$AverageSteps[match(maindata$interv
al[index==TRUE],mean.interval$interval)]
head(new.data)
```

```
## Source: local data frame [6 x 3]
## Groups: interval
##
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

```
#Organize
new.data<-group_by(new.data,date)
new.steps.day<-summarise(new.data,StepsPerDay=sum(steps))
new.mean.steps<-mean(new.steps.day$StepsPerDay)
new.median.steps<-median(new.steps.day$StepsPerDay)

qplot(new.steps.day$StepsPerDay,geom="histogram", xlab="Total Steps Taken Per
Day", ylab="Count",fill=I("purple"),col=I("blue"))

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust thi
s.
```



Here we can see the histogram or distribution of the total number of steps taken per day during the study period. BUT NOW the calculated values based on the overall average steps taken per interval is inserted in where NA was before:

We calculated the average **NEW** steps per day to have a **mean** of **1.076618910^{4}** steps per day and a **median** of **1.076618910^{4}** steps per day.

The **intital** calculations based on the data dropping NA's has a **mean** of **1.076618910^{4}** and a **median** of **10765** steps per day.

The impact of adding in the average is very small when comparing the difference it made to the mean and median of the data. Since we added in **averages** it didn't upset the mean and only altered the median by a nudge upward. This is because the whole distribtion got some added values that only matched the calculated mean in the first place.

Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

1. 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.

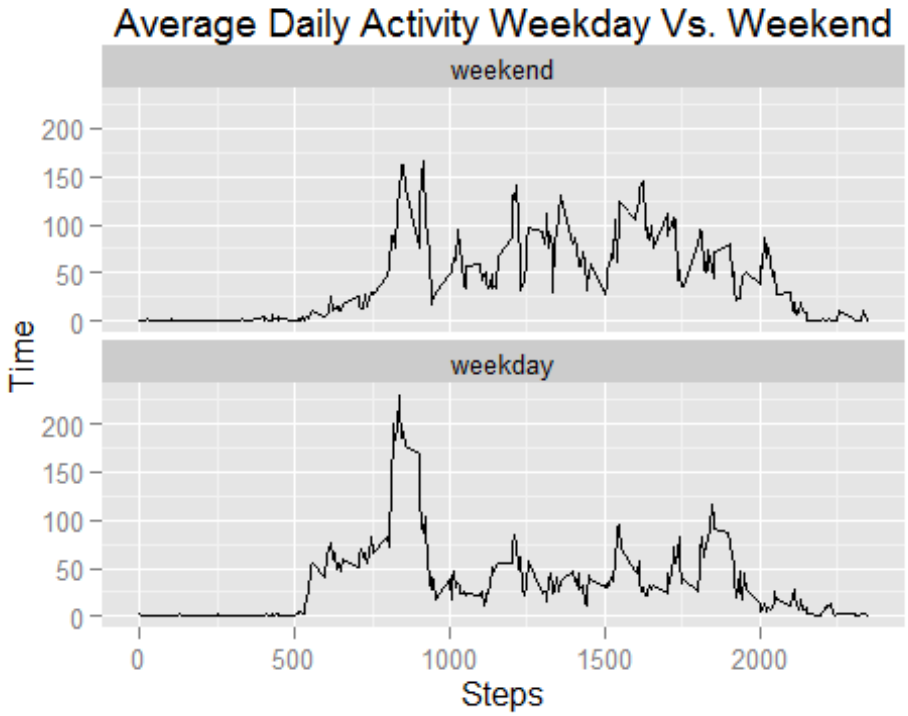
2. Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
new.data$date<-ymd(new.data$date)
new.data$weekday<-weekdays(new.data$date)

weekdays1 <- c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')
new.data$weekday <- factor((weekdays(new.data$date) %in% weekdays1), levels=c
(FALSE, TRUE), labels=c('weekend', 'weekday'))

weekday.activity<-select(new.data,interval,steps,weekday)%>%
        group_by(interval, weekday) %>%
        summarise(steps = mean(steps))

ggplot(weekday.activity, aes(x=interval, y=steps)) +
        geom_line() +
        facet_wrap(~weekday, ncol = 1) +
        ggtitle("Average Daily Activity Weekday Vs. Weekend") +
        xlab("Steps") +
        ylab("Time")
```



We can see form the plots that there is considerably more activity thruought the dat on **weekends** than on **weekdays** this is typically because we aren't at our desks working!