Fitness Activity Analysis

Jeffrey Uslan

September 14, 2015

Exectuive Summary

Exploratory Data Analysis

Load in relevant libraries and view data

```
library(ggplot2)
library(tidyr)
library(zoo)
library(dplyr)
```

Loading and preprocessing the data

```
activity<-tbl_df(read.csv(unzip("./repdata-data-activity.zip")))
```

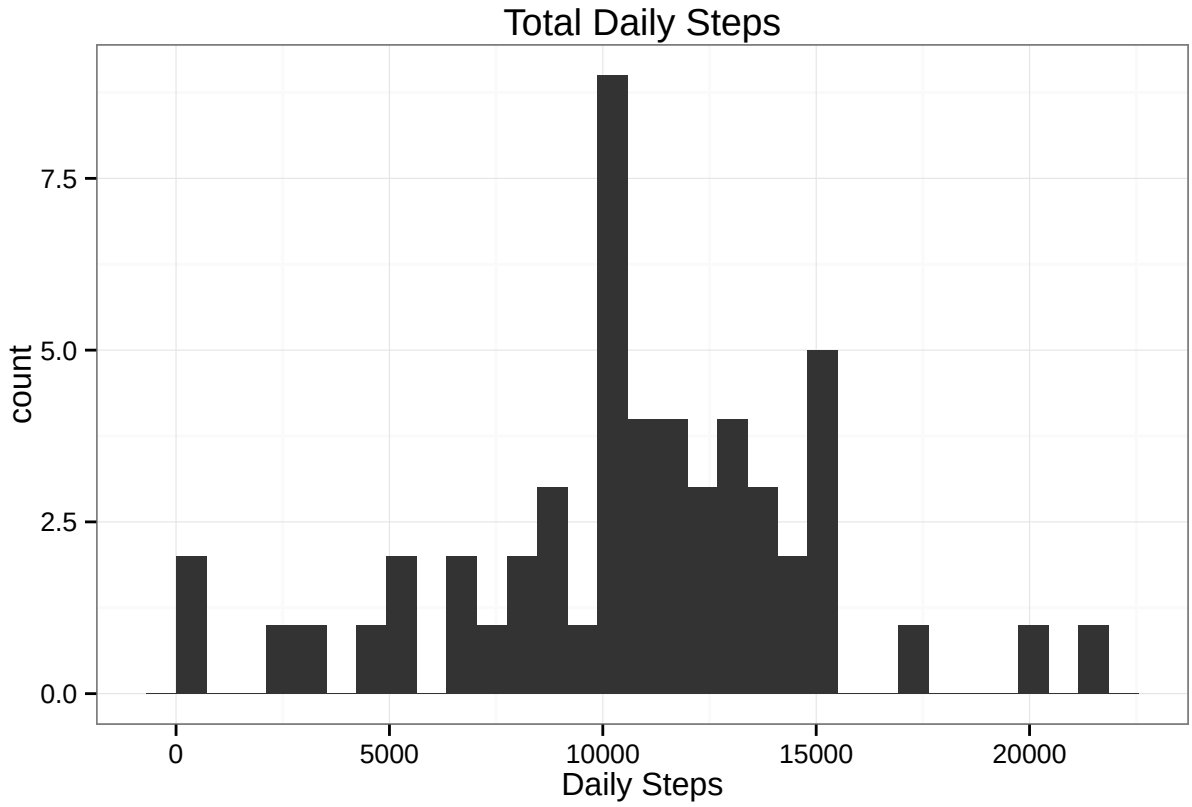
Checking basic summary.

```
##      steps               date          interval     
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0  
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8  
##  Median :  0.00   2012-10-03:  288   Median :1177.5  
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5  
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2  
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0  
##  NA's   :2304     (Other)   :15840
```

Calculating the total number of steps taken per day.

Making a histogram of the total number of steps taken each day.

```
ggplot(sum_steps,aes(x=daily_steps))+geom_histogram()+theme_bw()+
  xlab("Daily Steps")+ggtitle("Total Daily Steps")
```

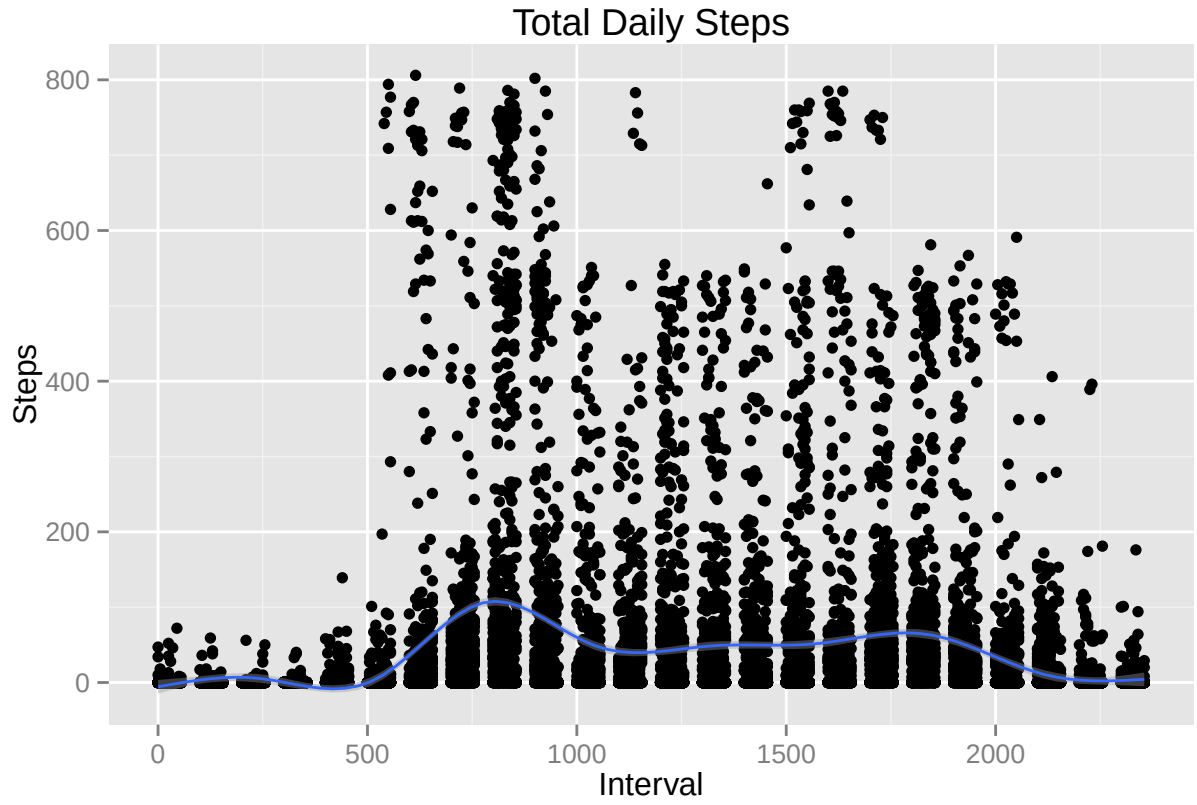


Calculating the mean and median of the total number of steps taken per day

```
## Source: local data frame [1 x 2]
##
##   Mean_Daily_Steps Median_Daily_Steps
## 1         10766.19              10765
```

Average daily activity pattern.

```
ggplot(activity,aes(x=interval,y=steps))+geom_point()+geom_smooth(aes(group=1))+
  xlab("Interval")+ylab("Steps")+ggtitle("Total Daily Steps")
```



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

```
activity$interval=as.factor(activity$interval)
mean_steps <- activity %>% na.omit() %>% group_by(interval) %>% summarise(interval_steps=mean(steps))
print(mean_steps[which(mean_steps$interval_steps==max(mean_steps$interval_steps)),])
```

```
## Source: local data frame [1 x 2]
##
##   interval interval_steps
## 1      835       206.1698
```

Imputing missing values

Calculate and report the total number of missing values in the dataset

```
print(apply(activity,2,function(x){
  sum(is.na(x))
})[1])
```

```
## steps
##  2304
```

Create a new dataset that is equal to the original dataset but with the missing data filled in. This method imputes using the averagr for that step.

```
imputed_activity<-apply(activity,1,function(x){
  if (is.na(x[1])){
    ind=which(x[3]==mean_steps$interval)
    x[1]=round(mean_steps$interval_steps[ind])

  }
  return(x)
}
)

imputed_activity=as.data.frame(t(imputed_activity))
imputed_activity$steps=as.numeric(as.character(imputed_activity$steps))
```

Checking the summaries of pre and post-imputed datasets as a sanity check.

```
summary(imputed_activity$steps)
```

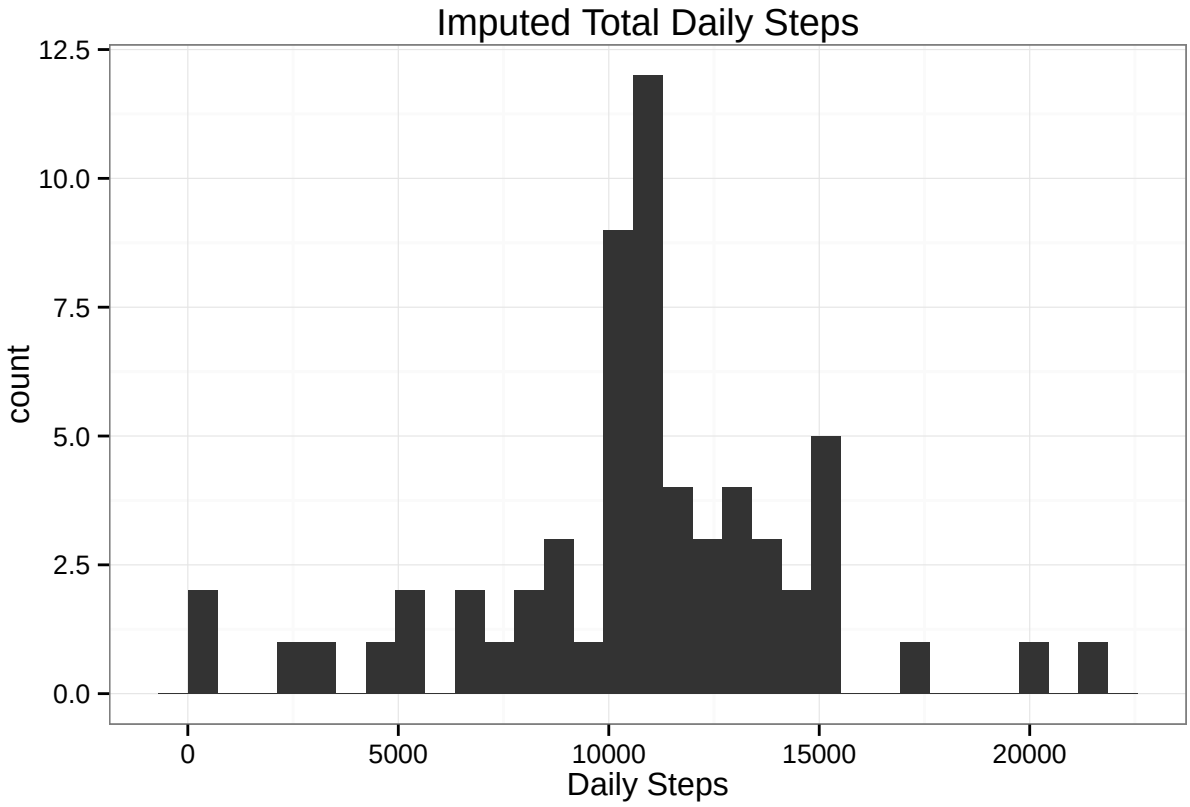
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   37.38   27.00  806.00
```

```
summary(activity$steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00   37.38   12.00  806.00    2304
```

Make a histogram of the total number of steps taken each day and

```
imputed_sum_steps <- imputed_activity %>% group_by(date) %>% summarise(daily_steps=sum(steps))
ggplot(imputed_sum_steps,aes(x=daily_steps))+geom_histogram()+theme_bw()+
  xlab("Daily Steps")+ggtitle("Imputed Total Daily Steps")
```



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?

```
## Source: local data frame [1 x 2]
##
##   Mean_Daily_Steps_Imputed Median_Daily_Steps_Imputed
## 1                 10765.64                       10762
```

```
## Source: local data frame [1 x 2]
##
##   Mean_Daily_Steps Median_Daily_Steps
## 1         10766.19              10765
```

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.

```
imputed_activity$date=as.POSIXct(activity$date)
week_break <- function(date) {
  day <- weekdays(date)
  if (day %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")){
    return("Weekday")
  }
  else { (day %in% c("Saturday", "Sunday"))
    return("Weekend")
  }
```

```
}
imputed_activity$week_break<-sapply(imputed_activity$date,FUN=week_break)
```

A panel plot of the average number of steps taken, averaged across all weekday days or weekend days.

```
imputed_sum_steps_weekday <- imputed_activity %>% group_by(date,week_break) %>% summarise(daily_steps=su
ggplot(imputed_activity,aes(x=interval,y=steps))+geom_point()+geom_smooth(aes(group=1))+
  xlab("Interval")+ylab("Steps")+ggtitle("Total Daily Steps")+facet_grid(week_break~.)
```