

Week 2 Course Project Activity Data

Part 1: Code for reading in the dataset and/or processing the data

Data file found on <https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip>
(<https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip>)

Data file has been downloaded to working directory and unzipped into csv file.

Read File

```
setwd("D:/CourseRA/Reproducible Research/Week 2")
```

```
ActivityData<-read.csv("activity.csv")  
summary(ActivityData)
```

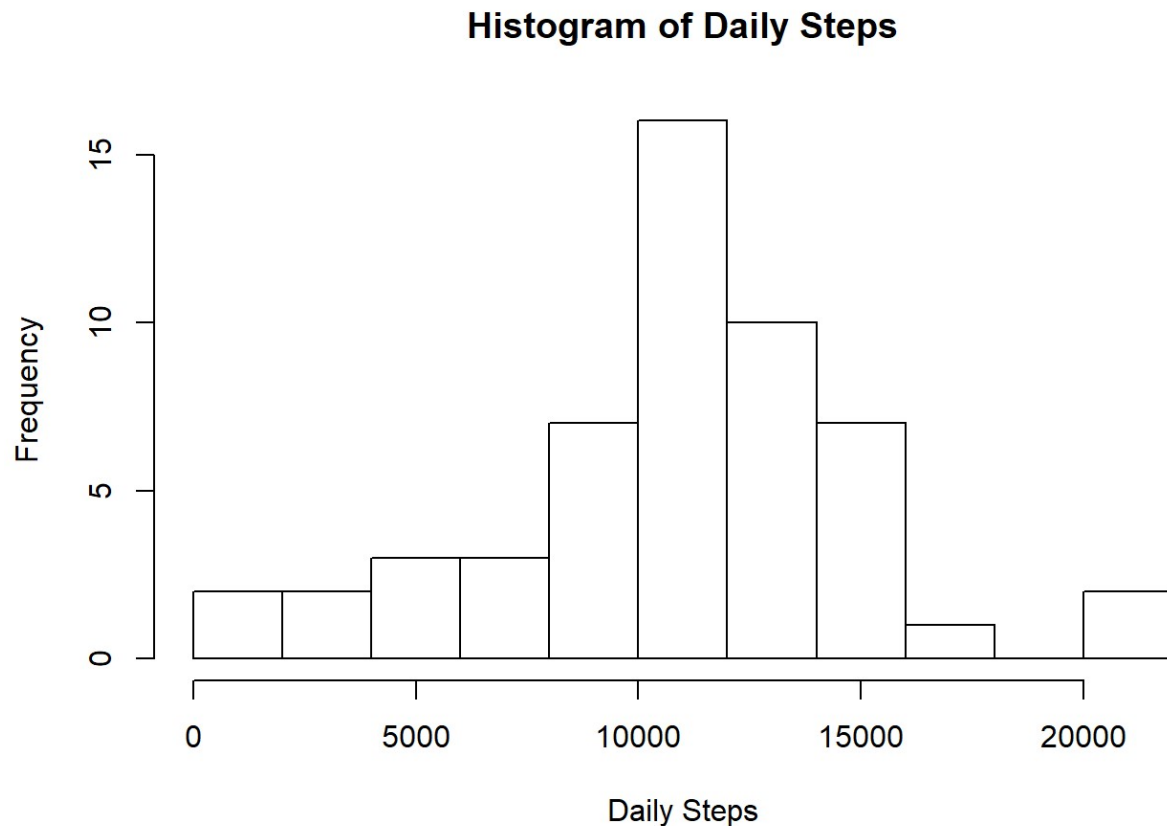
```
##      steps              date      interval  
## Min.   : 0.00  2012-10-01: 288  Min.   : 0.0  
## 1st Qu.: 0.00  2012-10-02: 288  1st Qu.: 588.8  
## Median : 0.00  2012-10-03: 288  Median :1177.5  
## Mean   : 37.38  2012-10-04: 288  Mean    :1177.5  
## 3rd Qu.: 12.00  2012-10-05: 288  3rd Qu.:1766.2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

Summary of data is shown to look at data format and identify any missing or NA fields.

Part 2: Histogram of the total number of steps taken each day

Examine steps on a daily basis

```
DailyData<-aggregate(ActivityData$steps, list(ActivityData$date), FUN=sum)  
names(DailyData)[1]<-"Date"  
names(DailyData)[2]<-"DailySteps"  
DailyData$Date<-as.Date(DailyData$Date, format="%Y-%m-%d")  
hist(DailyData$DailySteps,breaks=10,xlab="Daily Steps",main="Histogram of Daily Steps")
```



Part 3: Mean and median number of steps taken each day

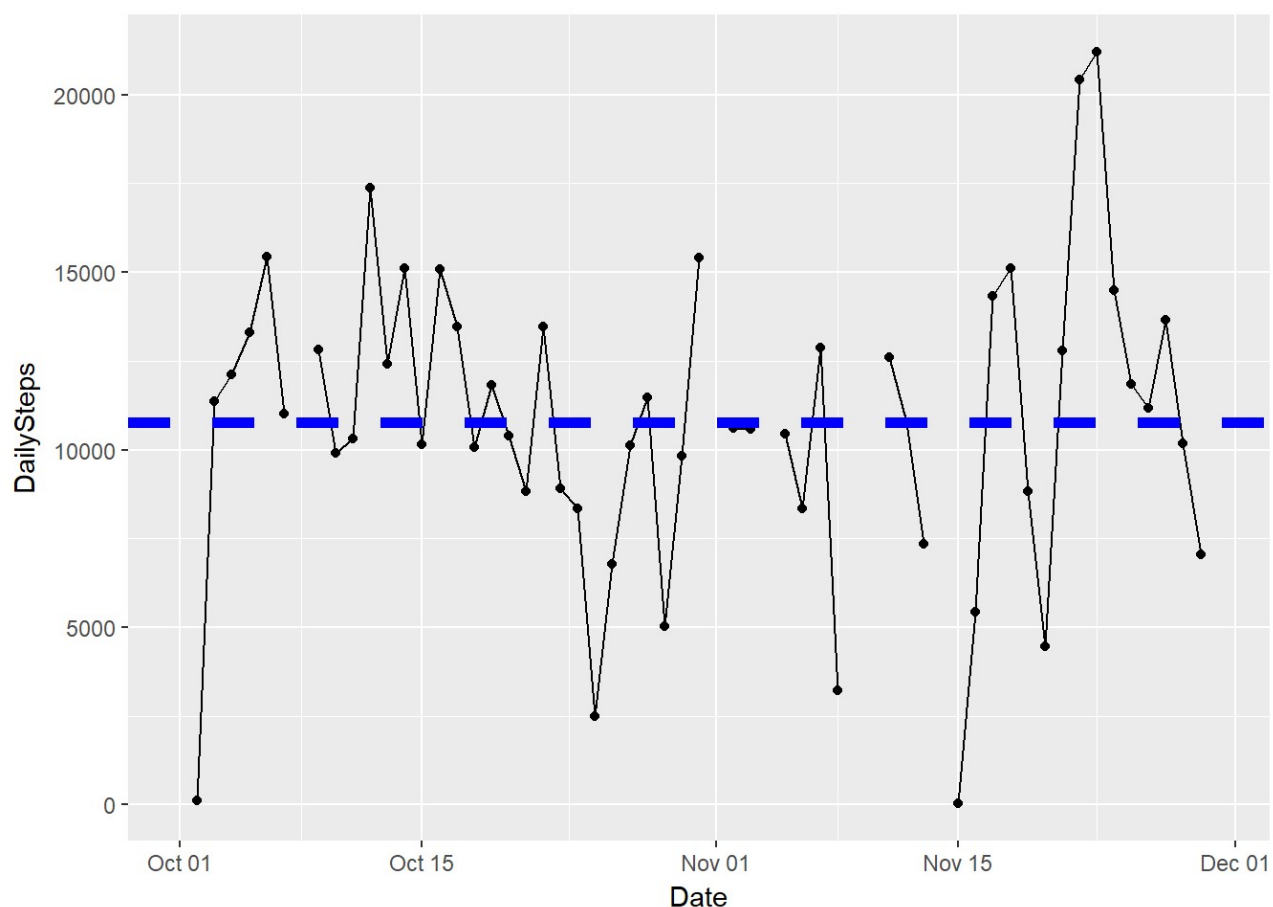
```
DailyMeanSteps<-mean(DailyData$DailySteps,na.rm=TRUE)
DailyMedianSteps<-median(DailyData$DailySteps,na.rm=TRUE)
```

The median number of steps per day is **10765** and the Mean number of steps per day is **10766.19** .

Part 4: Time series plot of the average number of steps taken

Plot time series plot showing average number steps taken per day.

```
library(ggplot2)
ggplot(DailyData,aes(Date,DailySteps))+geom_point(na.rm=TRUE)+geom_line(na.rm=TRUE)+geom_hline(aes(yintercept=DailyMeanSteps),color='blue',size=2,linetype=2)
```



overall average of steps is shown as dashed blue line. Daily data is plotted for each day.

Part 5: The 5-minute interval that, on average, contains the maximum number of steps

Need to look at the original 5 minute data and find the maximum value.

```
MaxSteps<-subset(ActivityData,steps==max(ActivityData$steps,na.rm=TRUE))
```

806 steps was maximum observed in a 5 minute interval and that occurred on 2012-11-27 interval number 615

Part 6: Code to describe and show a strategy for imputing missing data

Replace missing data with the average (mean) into Activity Data then reproduce DailyData data.frame

```

ActivityData$steps[is.na(ActivityData$steps)]<-DailyMeanSteps/288

DailyData<-aggregate(ActivityData$steps, list(ActivityData$date),FUN=sum)
names(DailyData)[1]<-"Date"
names(DailyData)[2]<-"DailySteps"
DailyData$Date<-as.Date(DailyData$Date, format="%Y-%m-%d")

summary(ActivityData)

```

```

##      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.: 37.38 2012-10-05: 288   3rd Qu.:1766.2
## Max.   :806.00 2012-10-06: 288   Max.    :2355.0
##                (Other)  :15840

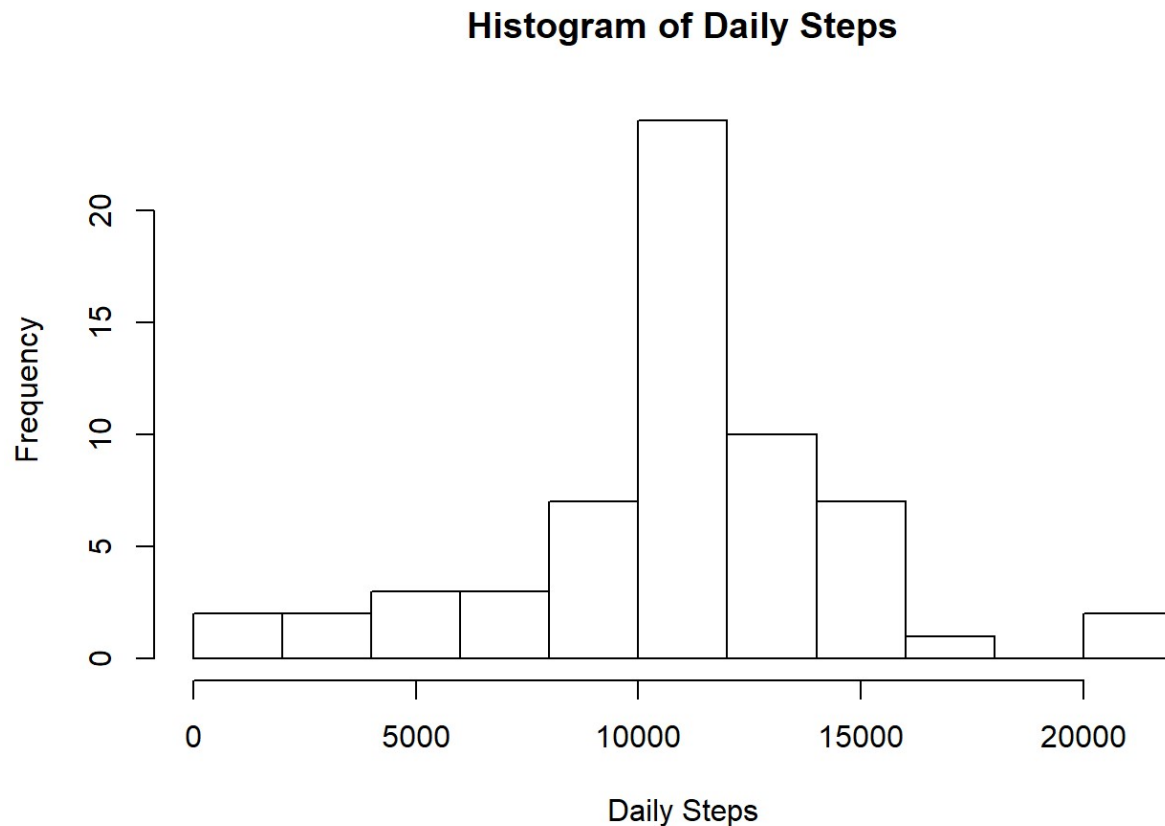
```

Part 7: Histogram of the total number of steps taken each day after missing values are imputed

```

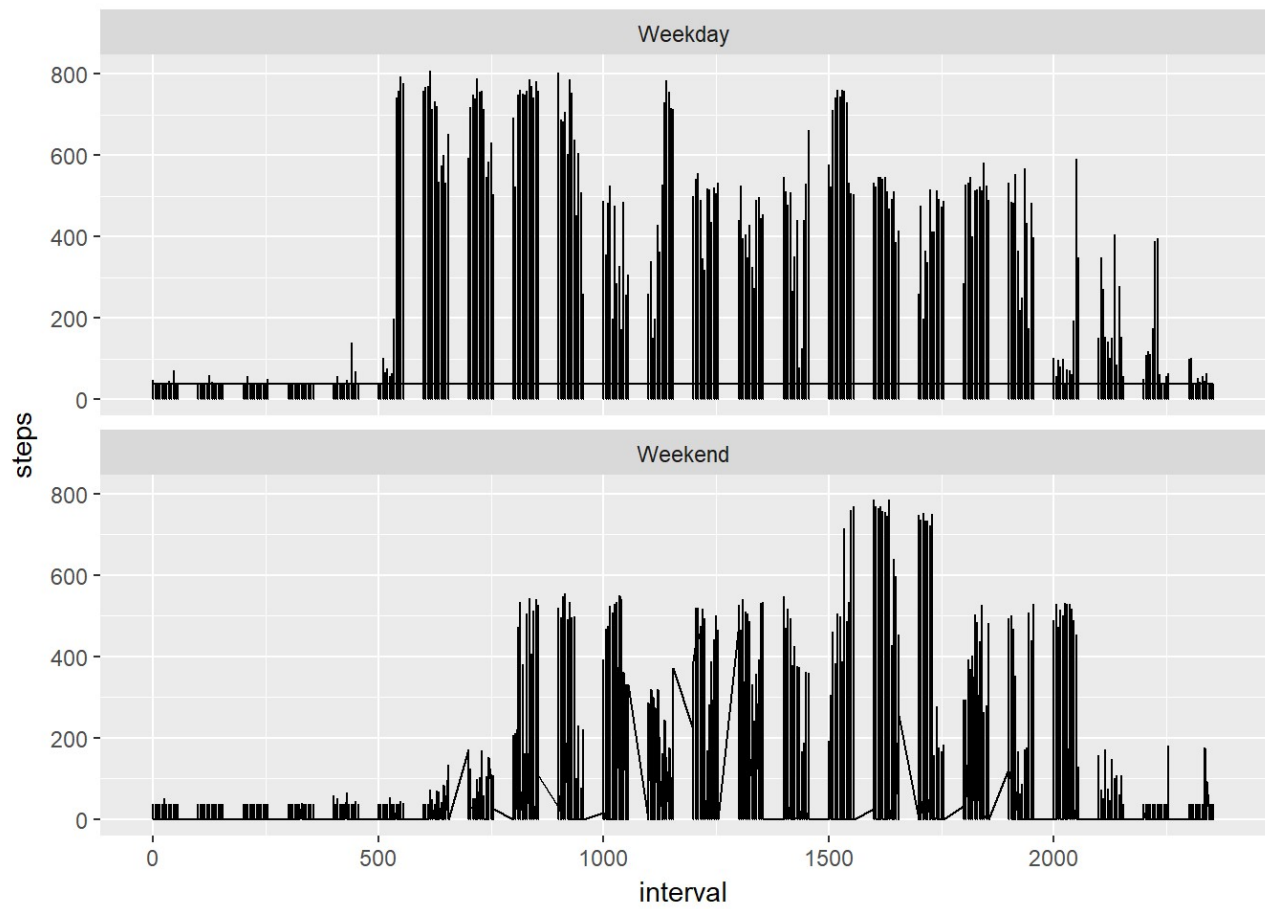
hist(DailyData$DailySteps,breaks=10,xlab="Daily Steps",main="Histogram of Daily Steps")

```



Part 8: Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
library(timeDate)
ActivityData$Weekend<-isWeekend(ActivityData$date)
ActivityData$Weekend<-as.factor(ActivityData$Weekend)
levels(ActivityData$Weekend)<-c("Weekday","Weekend")
ggplot(data=ActivityData, aes(x=interval,y=steps,group=Weekend))+geom_line()+facet_wrap(~Weekend,ncol=1)
```



```
means<-aggregate(steps~Weekend,ActivityData,mean)
means$steps<-round(means$steps,1)
ggplot(data=ActivityData, aes(x=Weekend,y=steps,fill=Weekend))+geom_boxplot()+stat_summary(fun.y=mean,geom="point",shape=18,size=3)+geom_text(data=means,aes(label=steps,y=steps+10))
```

