Activity Analysis

Yashu Seth

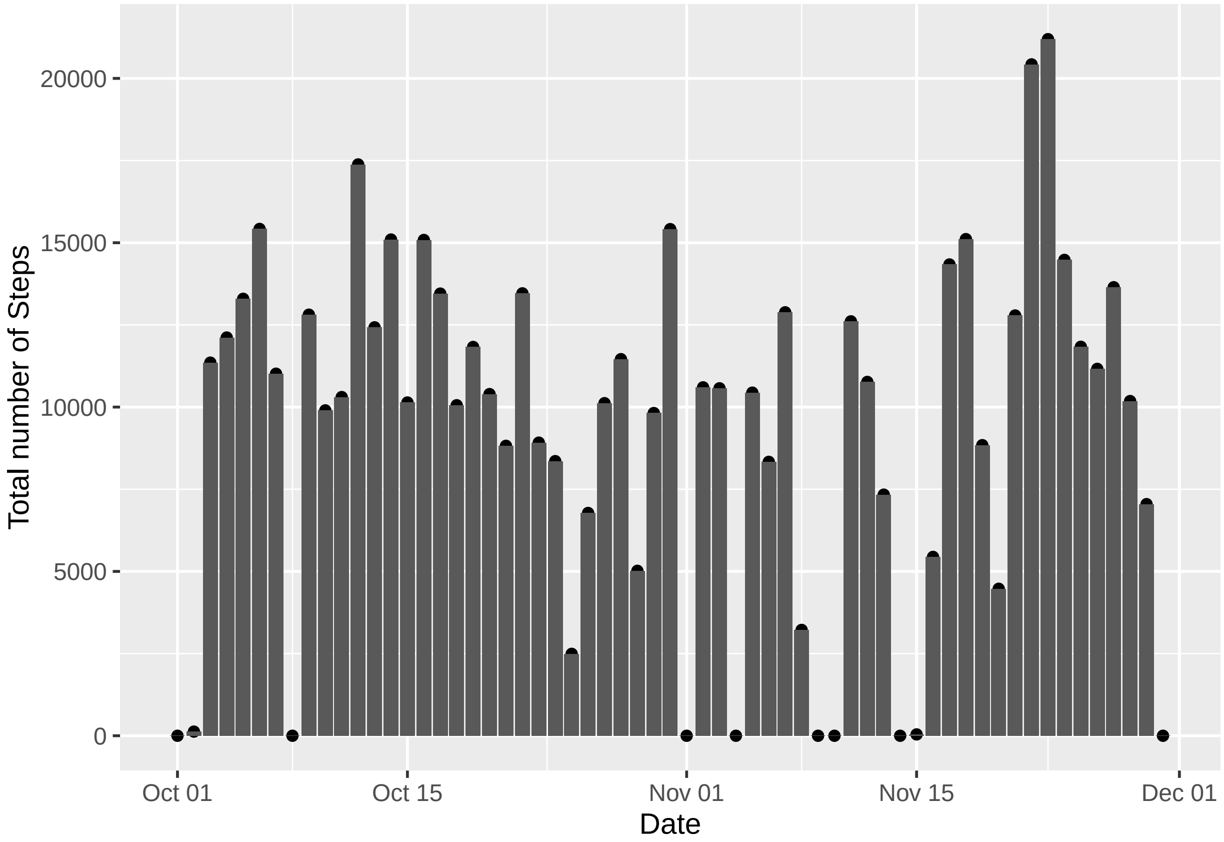
May 11, 2016

- **Code for reading in the dataset and/or processing the data**

```
setwd("C:/work/R")
data <- read.csv("activity.csv", colClasses = 'character')
```

- **Histogram of the total number of steps taken each day**

```
tot_steps <- tapply(as.integer(data$steps), data$date, sum, na.rm=T)
x <- strptime(rownames(tot_steps), "%Y-%m-%d")
y <- as.vector(tot_steps)
qplot(x,y) + geom_bar(stat = 'identity') + xlab("Date") +
    ylab("Total number of Steps")
```



- **Mean and median number of steps taken each day**

```
mean_steps <- tapply(as.integer(data$steps), data$date, mean,
                     na.rm=T)
```

The mean number of steps taken in each day -

```
mean_steps
```

```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06
##        NaN  0.4375000 39.4166667 42.0694444 46.1597222 53.5416667
## 2012-10-07 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12
## 38.2465278        NaN 44.4826389 34.3750000 35.7777778 60.3541667
## 2012-10-13 2012-10-14 2012-10-15 2012-10-16 2012-10-17 2012-10-18
## 43.1458333 52.4236111 35.2048611 52.3750000 46.7083333 34.9166667
## 2012-10-19 2012-10-20 2012-10-21 2012-10-22 2012-10-23 2012-10-24
## 41.0729167 36.0937500 30.6284722 46.7361111 30.9652778 29.0104167
## 2012-10-25 2012-10-26 2012-10-27 2012-10-28 2012-10-29 2012-10-30
##  8.6527778 23.5347222 35.1354167 39.7847222 17.4236111 34.0937500
## 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04 2012-11-05
## 53.5208333        NaN 36.8055556 36.7048611        NaN 36.2465278
## 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11
## 28.9375000 44.7326389 11.1770833        NaN        NaN 43.7777778
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17
## 37.3784722 25.4722222        NaN  0.1423611 18.8923611 49.7881944
## 2012-11-18 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23
## 52.4652778 30.6979167 15.5277778 44.3993056 70.9270833 73.5902778
## 2012-11-24 2012-11-25 2012-11-26 2012-11-27 2012-11-28 2012-11-29
## 50.2708333 41.0902778 38.7569444 47.3819444 35.3576389 24.4687500
## 2012-11-30
##        NaN
```

```
median_steps <- tapply(as.integer(data$steps), data$date, median,
                     na.rm=T)
```

The median number of steps taken in each day -

```
median_steps
```

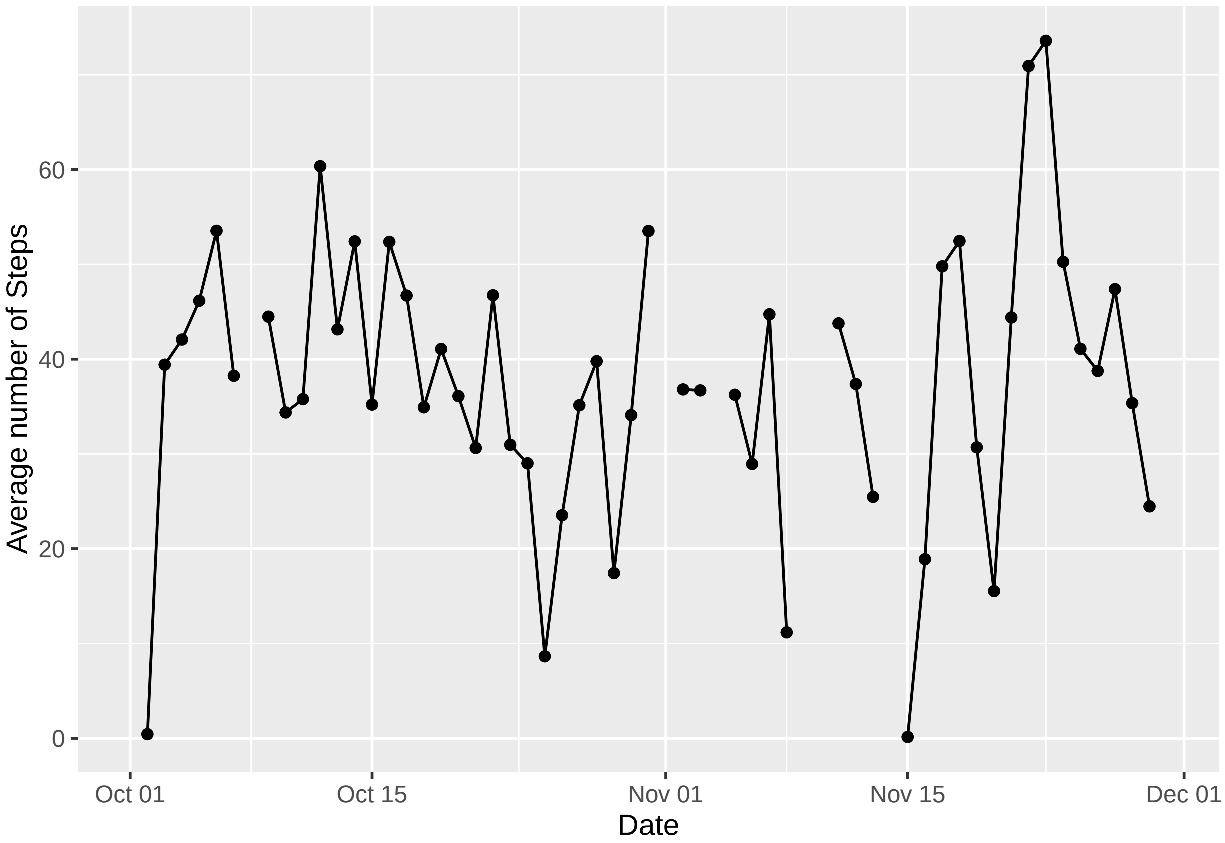
```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06
##         NA          0          0          0          0          0
## 2012-10-07 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12
##          0         NA          0          0          0          0
## 2012-10-13 2012-10-14 2012-10-15 2012-10-16 2012-10-17 2012-10-18
##          0          0          0          0          0          0
## 2012-10-19 2012-10-20 2012-10-21 2012-10-22 2012-10-23 2012-10-24
##          0          0          0          0          0          0
## 2012-10-25 2012-10-26 2012-10-27 2012-10-28 2012-10-29 2012-10-30
##          0          0          0          0          0          0
## 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04 2012-11-05
##          0         NA          0          0         NA          0
## 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11
##          0          0          0         NA         NA          0
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17
##          0          0         NA          0          0          0
## 2012-11-18 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23
##          0          0          0          0          0          0
## 2012-11-24 2012-11-25 2012-11-26 2012-11-27 2012-11-28 2012-11-29
##          0          0          0          0          0          0
## 2012-11-30
##         NA
```

- **Time series plot of the average number of steps taken**

```
y <- as.vector(mean_steps)
qplot(x,y) + geom_line() + xlab("Date") + ylab("Average number of Steps")
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```



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

```
int_avg <- tapply(as.integer(data$steps),
                  data$interval, mean, na.rm = T)
```

Therefore the 5-minute interval that, on average, contains the maximum number of steps is -

```
rownames(int_avg)[int_avg==max(int_avg)]
```

```
## [1] "835"
```

- **Code to describe and show a strategy for imputing missing data** I will replace the missing values in the steps column with the mean steps of that particular day. But there are certain missing values in the mean steps data as well. I will replace this with the total mean.

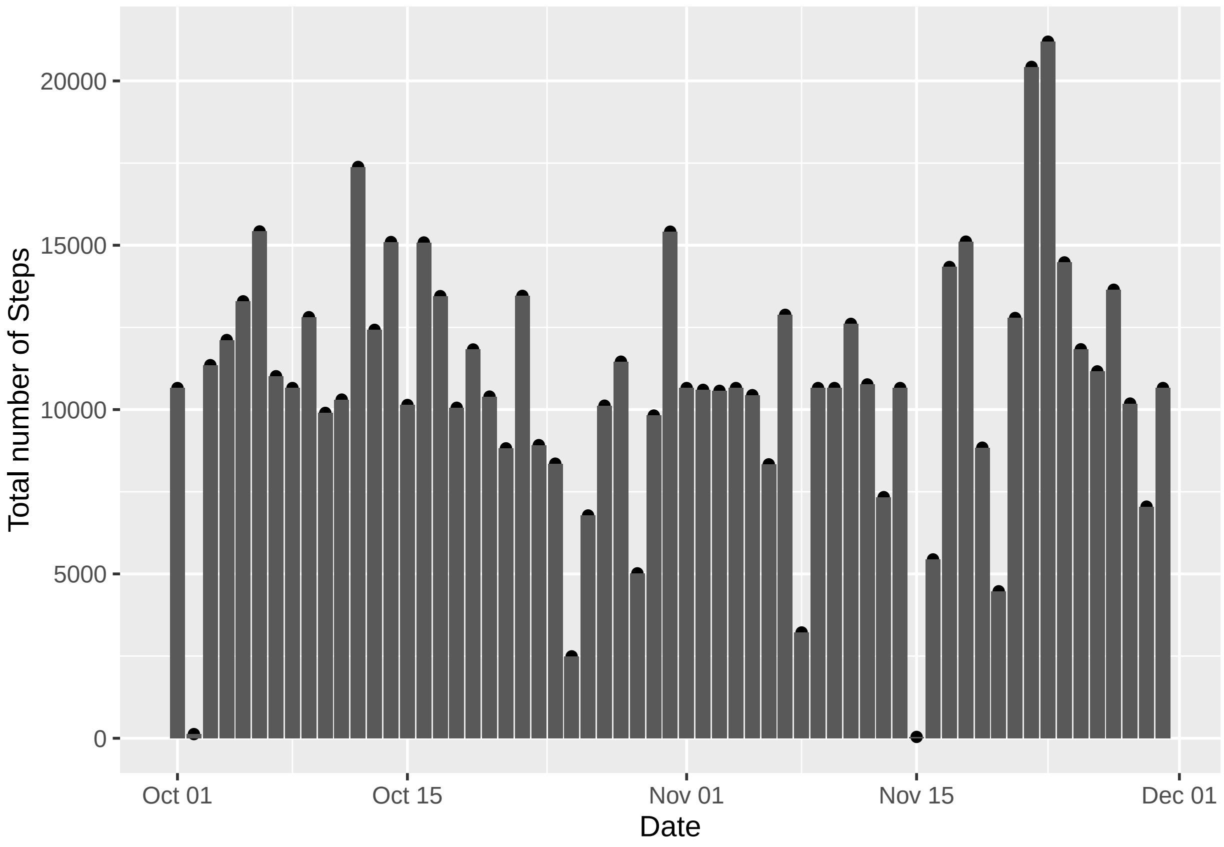
```
total_mean <- mean(mean_steps, na.rm = T)
mean_steps[is.nan(mean_steps)] <- total_mean
```

Now I will replace the missing values in the steps columns with the average number of steps taken that day.

```
for(i in 1:nrow(data)){
    row <- data[i,]
    if(is.na(row$steps)){
        row$steps <- mean_steps[row$date]
    }
    data[i,] <- row
}
```

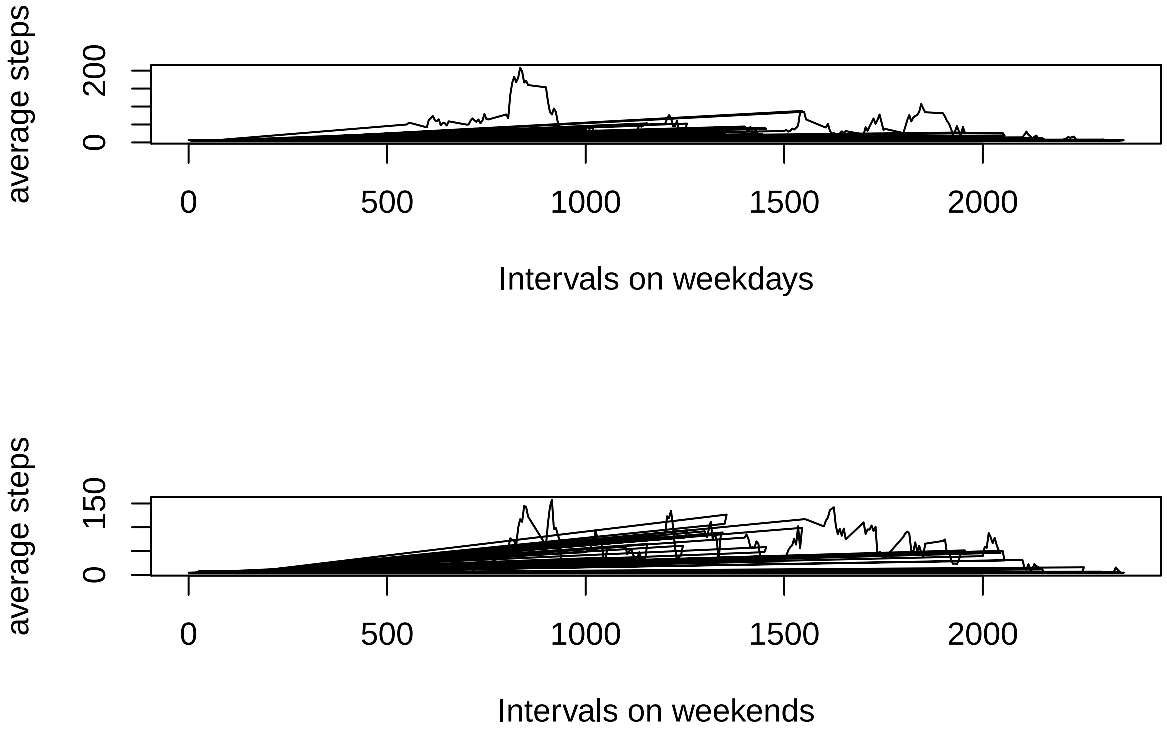
- **Histogram of the total number of steps taken each day after missing values are imputed**

```
tot_steps <- tapply(as.integer(data$steps), data$date, sum, na.rm=T)
x <- strptime(rownames(tot_steps), "%Y-%m-%d")
y <- as.vector(tot_steps)
qplot(x,y) + geom_bar(stat = 'identity') + xlab("Date") +
    ylab("Total number of Steps")
```



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

```
day <- weekdays(strptime(data$date, "%Y-%m-%d"))
r <- day=="Saturday" | day=="Sunday"
day[r] <- "weekend"
day[!r] <- "weekday"
data <- cbind(data, day)
day_data <- split(data, data$day)
int_avg_wd <- tapply(as.integer(day_data$weekday$steps),
                  day_data$weekday$interval, mean, na.rm = T)
int_avg_we <- tapply(as.integer(day_data$weekend$steps),
                  day_data$weekend$interval, mean, na.rm = T)
par(mfrow=c(2,1))
plot(rownames(int_avg_wd), as.vector(int_avg_wd), pch=20,
     xlab='Intervals on weekdays', ylab = 'average steps', type = 'l')
plot(rownames(int_avg_we), as.vector(int_avg_we), pch=20, type = 'l',
     xlab='Intervals on weekends', ylab = 'average steps')
```



- **All of the R code needed to reproduce the results (numbers, plots, etc.) in the report**

```
library('ggplot2')

setwd("C:/work/R")
data <- read.csv("activity.csv", colClasses = 'character')
```

```r
tot_steps <- tapply(as.integer(data$steps), data$date, sum, na.rm=T)
x <- strptime(rownames(tot_steps), "%Y-%m-%d")
y <- as.vector(tot_steps)
qplot(x,y) + geom_bar(stat = 'identity') + xlab("Date") +
    ylab("Total number of Steps")

mean_steps <- tapply(as.integer(data$steps), data$date, mean,
                     na.rm=T)

mean_steps

median_steps <- tapply(as.integer(data$steps), data$date, median,
                     na.rm=T)

median_steps

y <- as.vector(mean_steps)
qplot(x,y) + geom_line() + xlab("Date") + ylab("Average number of Steps")

int_avg <- tapply(as.integer(data$steps),
                  data$interval, mean, na.rm = T)

rownames(int_avg)[int_avg==max(int_avg)]

total_mean <- mean(mean_steps, na.rm = T)
mean_steps[is.nan(mean_steps)] <- total_mean

for(i in 1:nrow(data)){
    row <- data[i,]
    if(is.na(row$steps)){
        row$steps <- mean_steps[row$date]
    }
    data[i,] <- row

tot_steps <- tapply(as.integer(data$steps), data$date, sum, na.rm=T)
x <- strptime(rownames(tot_steps), "%Y-%m-%d")
y <- as.vector(tot_steps)
qplot(x,y) + geom_bar(stat = 'identity') + xlab("Date") +
    ylab("Total number of Steps")

day <- weekdays(strptime(data$date, "%Y-%m-%d"))
r <- day=="Saturday" | day=="Sunday"
day[r] <- "weekend"
day[!r] <- "weekday"
data <- cbind(data, day)
day_data <- split(data, data$day)
int_avg_wd <- tapply(as.integer(day_data$weekday$steps),
                  day_data$weekday$interval, mean, na.rm = T)
int_avg_we <- tapply(as.integer(day_data$weekend$steps),
                  day_data$weekend$interval, mean, na.rm = T)
par(mfrow=c(2,1))
plot(rownames(int_avg_wd), as.vector(int_avg_wd), pch=20,
     xlab='Intervals on weekdays', ylab = 'average steps', type = 'l')
```

```
plot(rownames(int_avg_we), as.vector(int_avg_we), pch=20, type = 'l',
     xlab='Intervals on weekends', ylab = 'average steps')
```