Course Project 1

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.

The variables included in this dataset are:

- **steps:** Number of steps taking in a 5-minute interval (missing values are 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.

Commit

1. Code for reading in the dataset

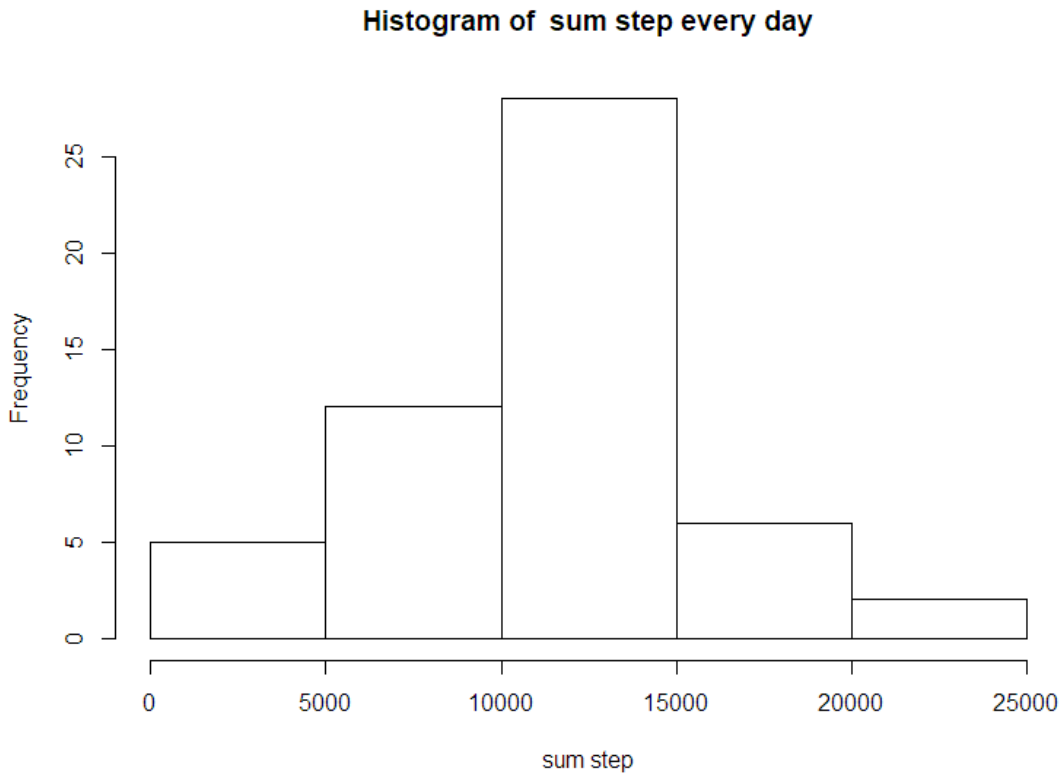
```
df = read.csv('D:\\OneDrive - zju.edu.cn\\Coursera\\R-5\\repdata_data_activity\\activity.csv')
head(df)
```

```
##   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
```

2. Histogram of the total number of steps taken each day

```
sum_df = tapply(df[, 'steps'], df[, 'date'], sum)
hist(
     sum_df,
```

```
     main = paste("Histogram of ", 'sum step every day'),
     xlab = 'sum step'
     )
```



3. Mean and median number of steps taken each day

- number of steps taken each day

  beacause of there are too many words to print, so I choose some to print

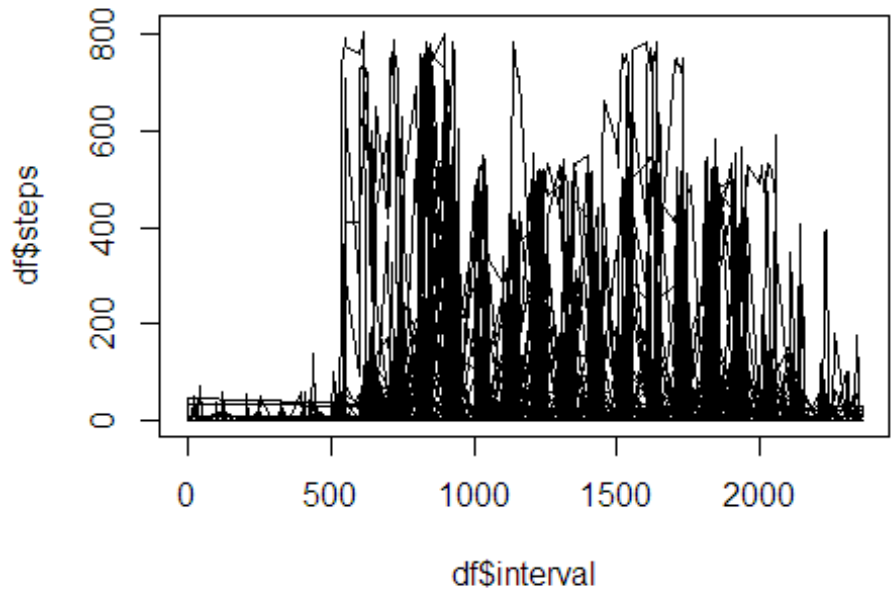
```
mean_df = tapply(df[, 'steps'], df[, 'date'], mean)
mean_eachday = mean(mean_df)
median_eachday = median(mean_df)
mean_df[1]

## 2012-10-01
##         NA

mean_df[2]

## 2012-10-02
##     0.4375

mean_df[3]

## 2012-10-03
##   39.41667
```

```
mean_df[4]

## 2012-10-04
##   42.06944

mean_df[5]

## 2012-10-05
##   46.15972
```

so the mean of the total number of steps taken per day is NA, the meadian of the total number of steps taken per day is NA.

4. Time series plot of the average number of steps taken

```
plot(
     df$interval, df$steps,
     type = 'l'
     )
```



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

```
df_s = subset(df, steps == max(df$steps, na.rm = TRUE))
inter = df_s$interval
```

the 615 interval that, on average, contains the 806 number of steps

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

```
library(mice)
imp = mice(df)
```

```
##
##  iter imp variable
##   1   1  steps
##   1   2  steps
##   1   3  steps
##   1   4  steps
##   1   5  steps
##   2   1  steps
##   2   2  steps
##   2   3  steps
##   2   4  steps
##   2   5  steps
##   3   1  steps
##   3   2  steps
##   3   3  steps
##   3   4  steps
##   3   5  steps
##   4   1  steps
##   4   2  steps
##   4   3  steps
##   4   4  steps
##   4   5  steps
##   5   1  steps
##   5   2  steps
##   5   3  steps
##   5   4  steps
##   5   5  steps
```

```
fit = with(imp, lm(steps ~ date, data = df))
pooled = pool(fit)
result = complete(imp)
df = as.data.frame(result)
head(df)
```

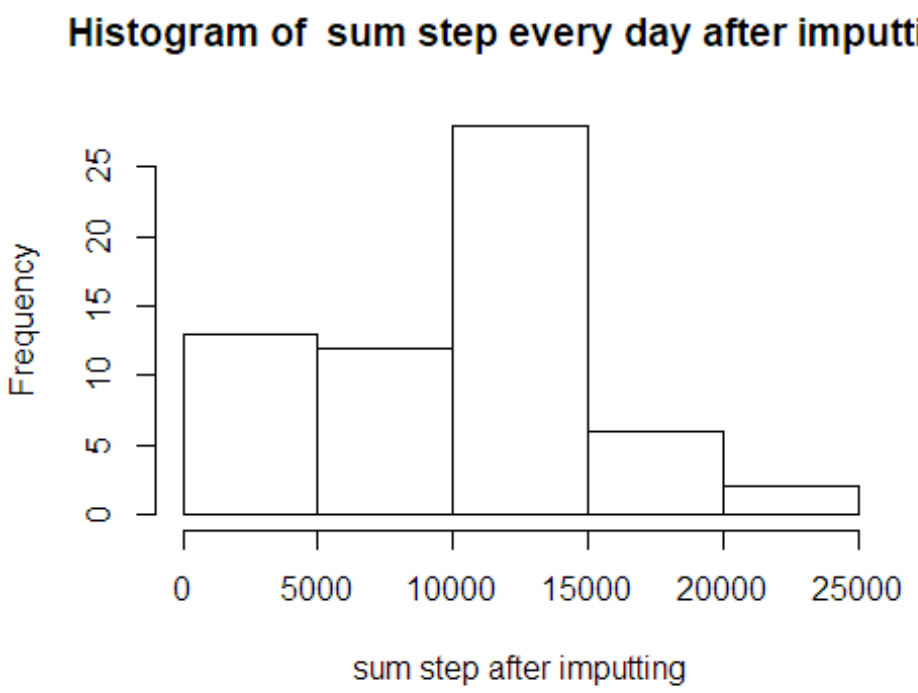
```
##   steps       date interval
## 1     0 2012-10-01        0
## 2     0 2012-10-01        5
## 3     0 2012-10-01       10
## 4     0 2012-10-01       15
## 5     0 2012-10-01       20
## 6     0 2012-10-01       25
```

I use the library `mice` to imput the data. I suppose that there is a liner relation ship between steps and date

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

```
sum_df = tapply(df[, 'steps'], df[, 'date'], sum)
hist(
     sum_df,
```

```
     main = paste("Histogram of ", 'sum step every day after imputting
'),
     xlab = 'sum step after imputting'
     )
```



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

```
library(lubridate)
df_weekend = subset(df, wday(as.Date(df$date, '%Y - %m - %d')) >= 6)
sum_df_weekend = tapply(df_weekend[, 'steps'], df_weekend[, 'date'], su
m)
df_weekdays = subset(df, wday(as.Date(df$date, '%Y - %m - %d')) < 6)
sum_df_weekdays = tapply(df_weekdays[, 'steps'], df_weekdays[, 'date'],
 sum)
```

I sperate the dateset by weekdays

weekdays

```
head(df_weekdays)

##   steps       date interval
## 1     0 2012-10-01        0
## 2     0 2012-10-01        5
## 3     0 2012-10-01       10
## 4     0 2012-10-01       15
```

```
## 5     0 2012-10-01       20
## 6     0 2012-10-01       25
```

weekends

```
head(df_weekend)
```

```
##      steps       date interval
## 1153     0 2012-10-05        0
## 1154     0 2012-10-05        5
## 1155     0 2012-10-05       10
## 1156     0 2012-10-05       15
## 1157     0 2012-10-05       20
## 1158     0 2012-10-05       25
```

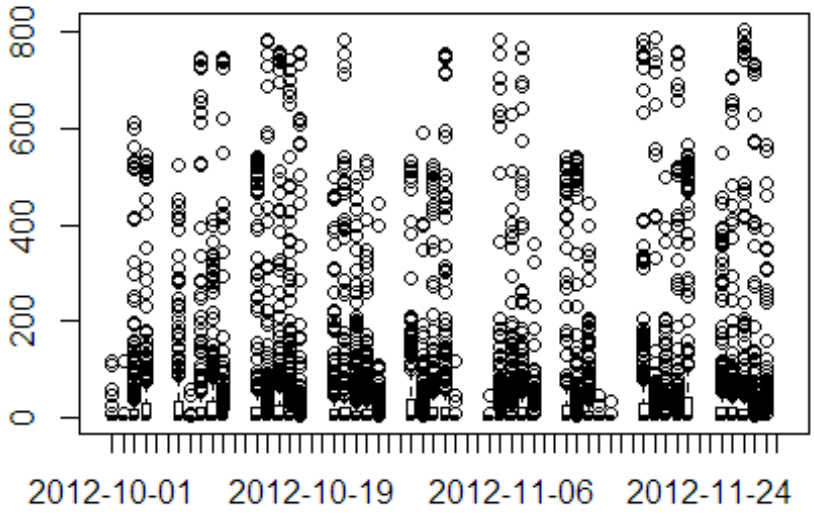
The result of the t-test

```
t.test(sum_df_weekend, sum_df_weekdays)
```

```
##
##  Welch Two Sample t-test
##
## data:  sum_df_weekend and sum_df_weekdays
## t = 0.7565, df = 25.241, p-value = 0.4564
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2157.750  4665.114
## sample estimates:
## mean of x mean of y
## 10287.000  9033.318
```

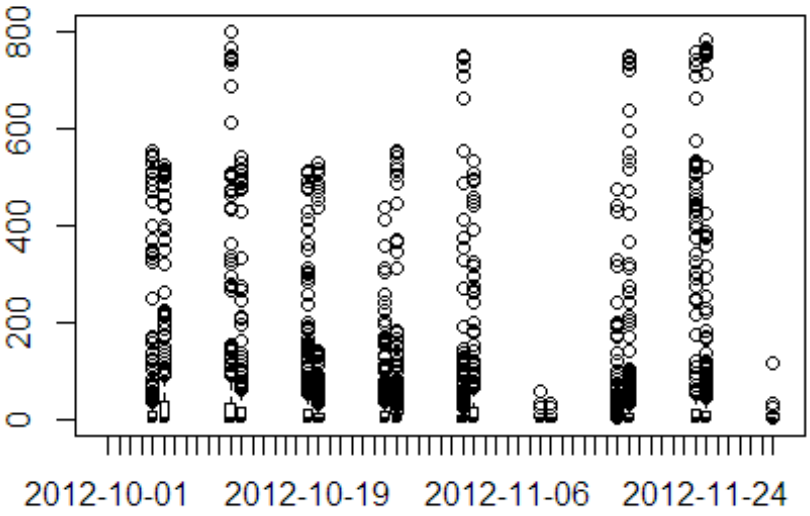
we can find that the steps in weekends *significantly* greater than steps in weekdays
weekdays plot

```
plot(
     df_weekdays$date,
     df_weekdays$steps,
     type = 'l'
     )
```



weekends plot

```
plot(
     df_weekend$date,
     df_weekend$steps,
     type = 'l'
     )
```


0
200
400
600
800
2012-10-01
2012-10-19
2012-11-06
2012-11-24