Reproducible Research of "Data Science Specialization": Peer Assessment 1

Loading and preprocessing the data

Show any code that is needed to:

1. Load the data (i.e. read.csv())

```
if(!file.exists('activity.csv')){
    unzip('activity.zip')
}
data = read.csv ('activity.csv', sep = ',')
```

2. Process/transform the data (if necessary) into a format suitable for your analysis

- Create a new column 'time_interval'

```
time_interval = formatC (data$interval/100, 2, format = 'f')
data$time_interval = as.POSIXct (paste (data$date, time_interval), format = '%Y-%m-%d %H.%M')
data$time_interval = format (data$time_interval, format = '%H:%M')
data$time_interval = as.POSIXct (data$time_interval, format='%H:%M')
```

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.

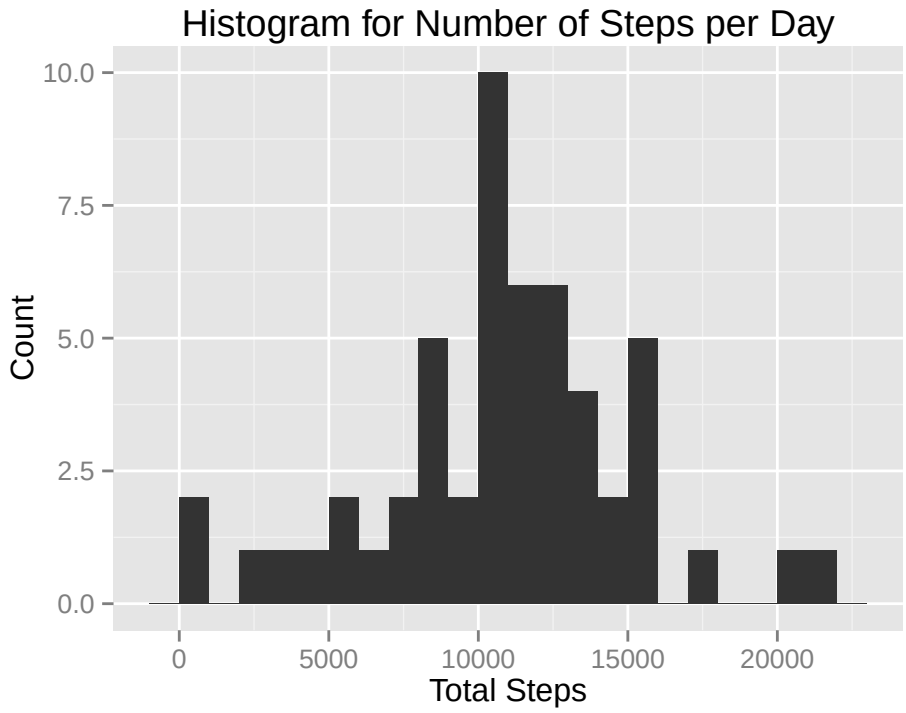
1. Make a histogram of the total number of steps taken each day

- Calculate the total number of steps taken each day (ignore missing values)

```
na_omit_data = na.omit (data)
total_steps = tapply(na_omit_data$steps, na_omit_data$date, sum, na.rm = TRUE)
```

- Make a histogram using ggplot2

```
library (ggplot2)
hist_steps = qplot (total_steps, xlab = 'Total Steps', ylab = 'Count', binwidth = 1000) + ggtitle ("His
hist_steps
```



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

```
mean (total_steps, na.rm = TRUE)
```

```
## [1] 10766.19
```

```
median (total_steps, na.rm = TRUE)
```

```
## [1] 10765
```

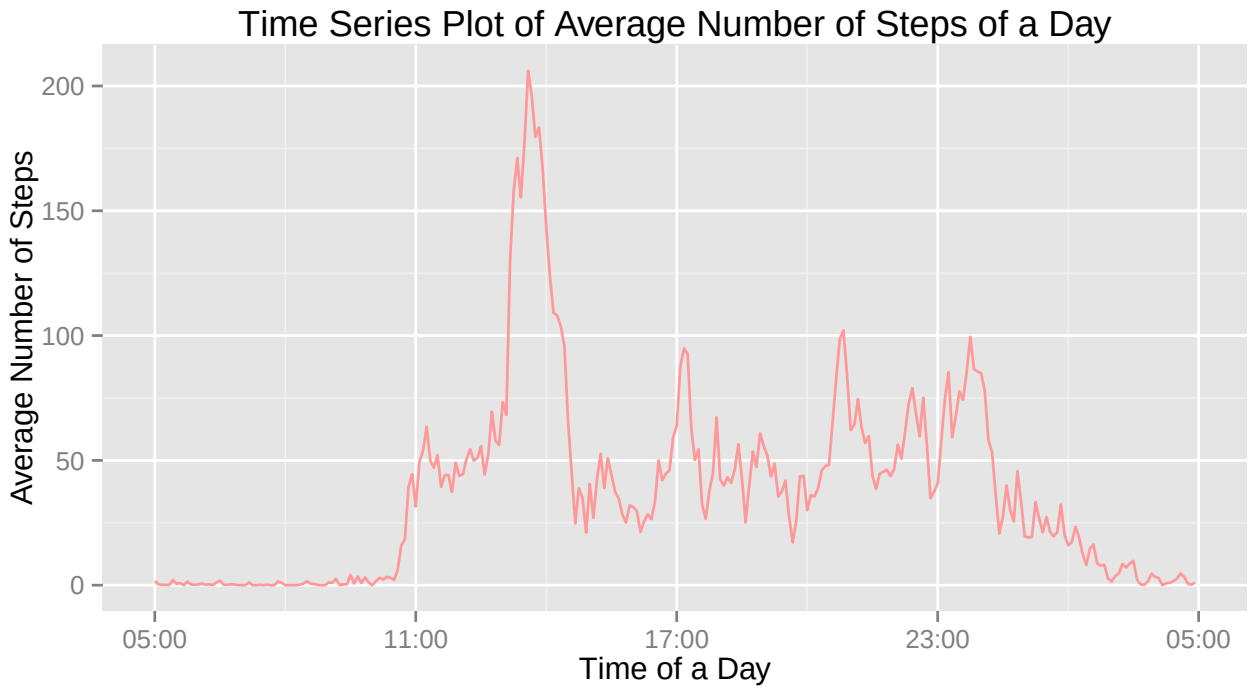
What is the average daily activity pattern?

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)

```
avg_steps_vs_interval = aggregate (
    data = na_omit_data,
    steps ~ time_interval,
    FUN = mean,
    na.action = na.omit
)

library (scales)
plot_step_interval = ggplot (aes (x = time_interval, y = steps), data = avg_steps_vs_interval) + geom_li

plot_step_interval
```



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

```
data$interval [which.max (avg_steps_vs_interval$steps)]
```

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

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)

```
sum (is.na (data$steps))
```

```
## [1] 2304
```

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.

Use the mean/median for that day to replace the missing values in the dataset.

```
imputed_data = data
imputed_data[is.na (imputed_data$steps), 'steps'] = ceiling (tapply (X = data$steps, INDEX = data$inter
```

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

- original dataset

```
head (data [colnames (data) [1 : 3]], 5)
```

```
##   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
```

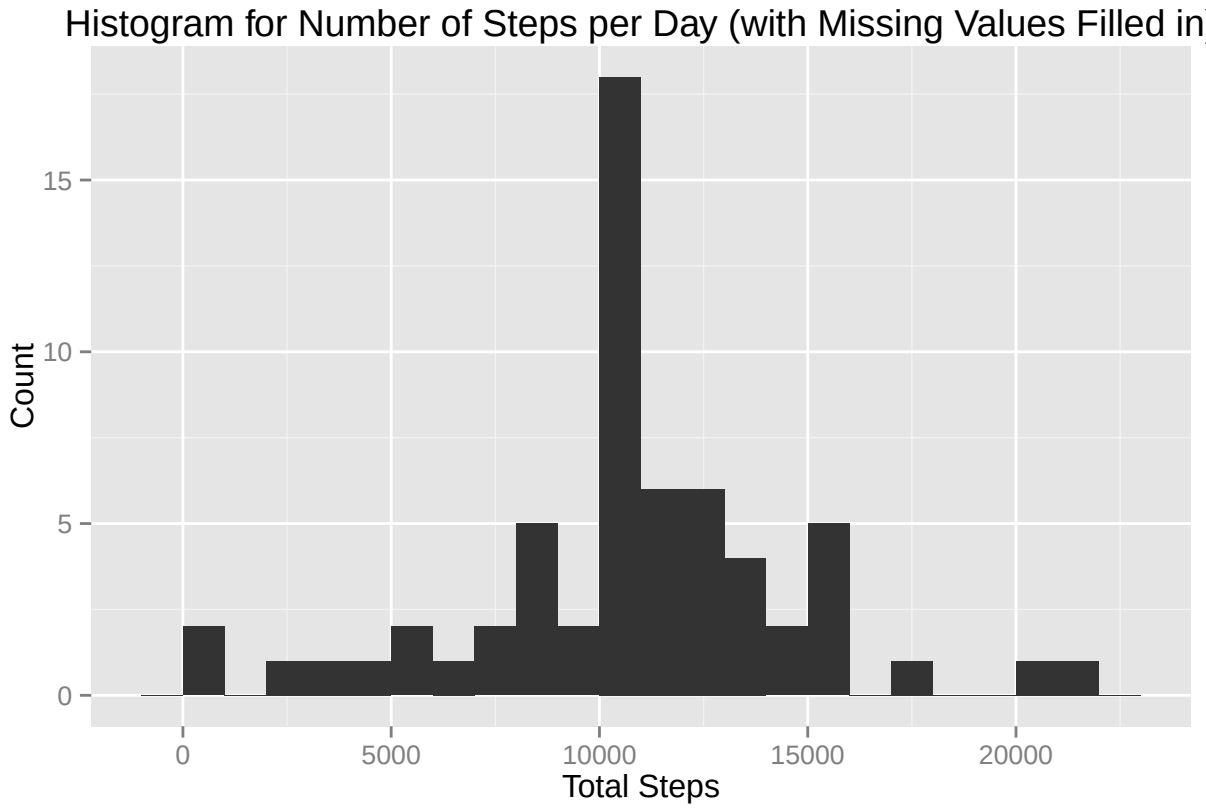
- new dataset with missing values filled in

```
head (imputed_data [colnames (imputed_data) [1 : 3]], 5)
```

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

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.

```
total_steps_imputed = tapply(imputed_data$steps, imputed_data$date, sum)
hist_steps_imputed = qplot (total_steps_imputed, xlab = 'Total Steps', ylab = 'Count', binwidth = 1000)
hist_steps_imputed
```




```
mean (total_steps_imputed)
```

```
## [1] 10784.92
```

```
median (total_steps_imputed)
```

```
## [1] 10909
```

- Do these values differ from the estimates from the first part of the assignment?

Yes.

- What is the impact of imputing missing data on the estimates of the total daily number of steps?

It seems that imputing missing data causes both the mean and median values to increase.

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.

```
weekday_type = function(date)
        {
        if (weekdays(as.Date(date)) %in% c('Saturday', 'Sunday'))
                {
                return('weekend')
                }
        else
                {
                return('weekday')
                }
        }

imputed_data$weekday_type = sapply (imputed_data$date, weekday_type)

head (imputed_data [colnames (imputed_data) [c(1 : 3, 5)]], 5)
```

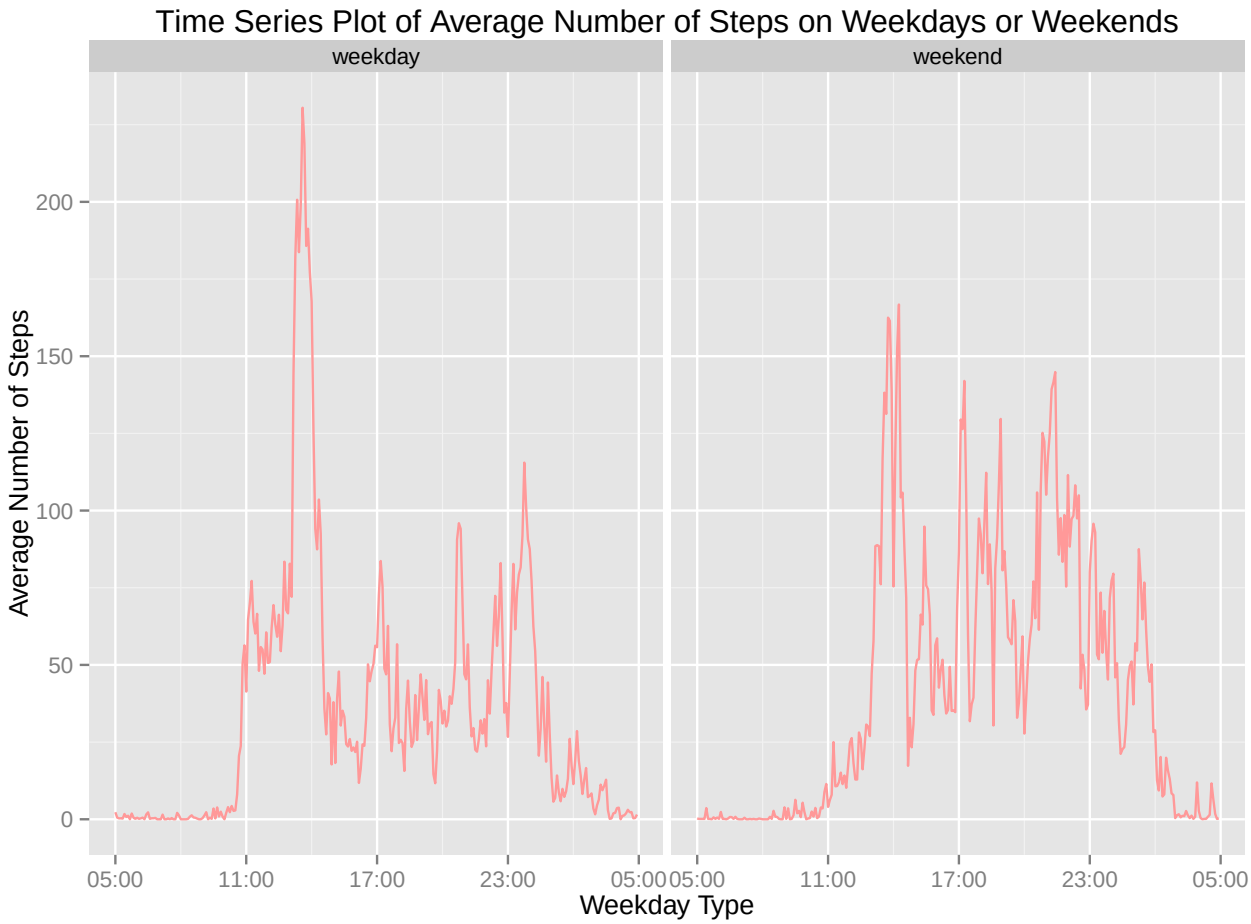
```
##   steps       date interval weekday_type
## 1     2 2012-10-01        0      weekday
## 2     1 2012-10-01        5      weekday
## 3     1 2012-10-01       10      weekday
## 4     1 2012-10-01       15      weekday
## 5     1 2012-10-01       20      weekday
```

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).

```
avg_steps_vs_weekday = aggregate (
    data = imputed_data,
    steps ~ weekday_type + time_interval,
    FUN = mean,
    na.action = na.omit
)

plot_step_weekday = ggplot (aes (x = time_interval, y = steps), data = avg_steps_vs_weekday) + geom_lin

plot_step_weekday
```


Time Series Plot of Average Number of Steps on Weekdays or Weekends
weekday
weekend
Average Number of Steps
200
150
100
50
0
05:00
11:00
17:00
23:00
05:00
05:00
11:00
17:00
23:00
05:00
Weekday Type