Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

```
library(readr)
unzip("activity.zip")
activity_data <- read_csv("activity.csv")
```

```
## Parsed with column specification:
## cols(
##   steps = col_double(),
##   date = col_date(format = ""),
##   interval = col_double()
## )
```

What is mean total number of steps taken per day?

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

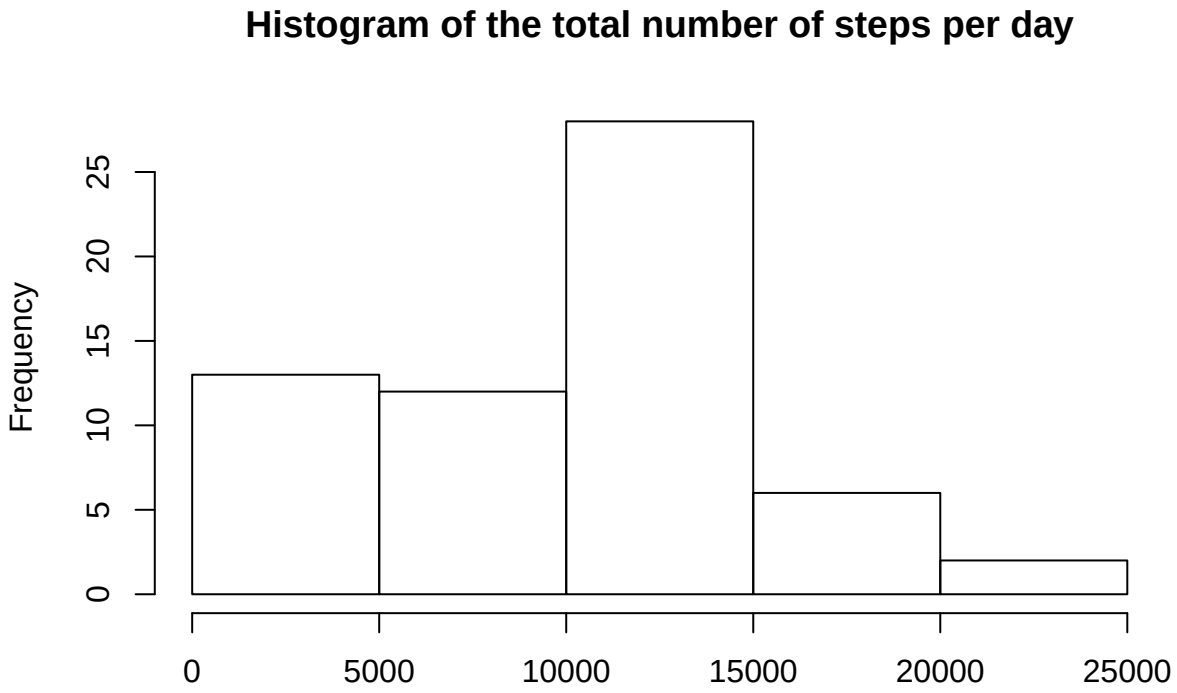
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
dt <- activity_data %>%
        group_by(date) %>%
        summarise(freq = sum(steps, na.rm = TRUE))
hist(dt$freq, main = "Histogram of the total number of steps per day", xlab = "")
```



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

```
mean(dt$freq)
```

```
## [1] 9354.23
```

```
median(dt$freq)
```

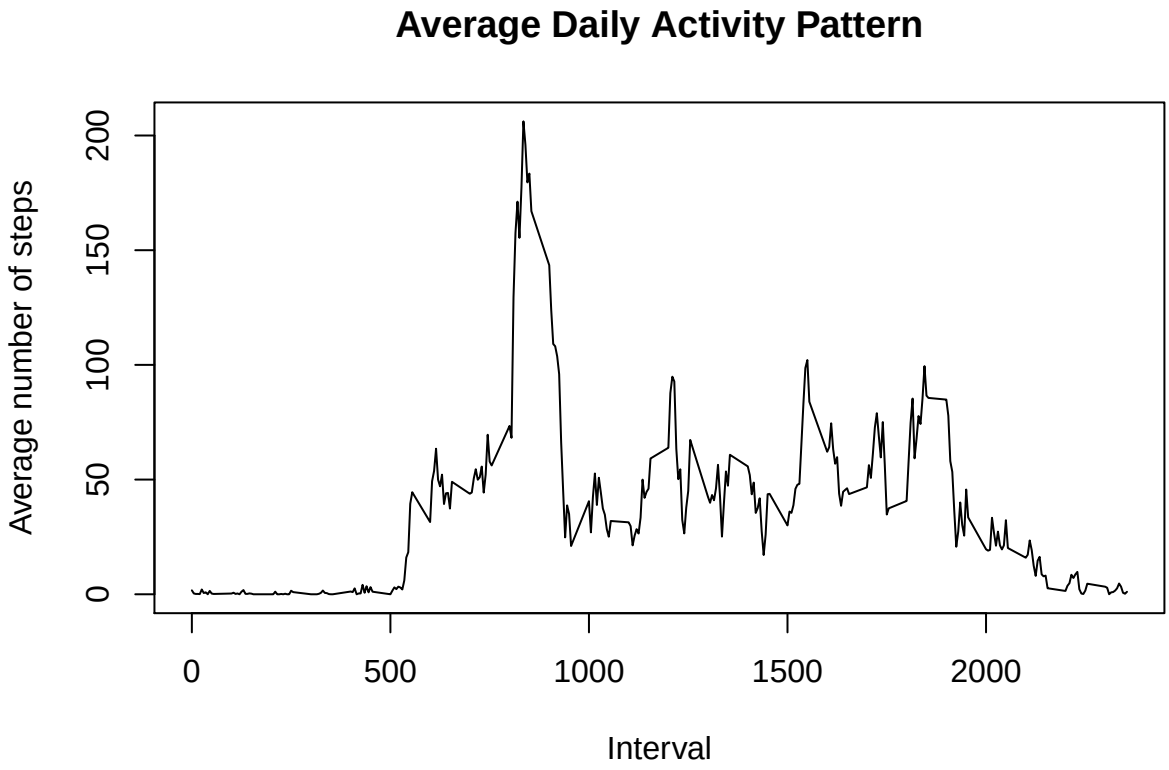
```
## [1] 10395
```

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)

```
dt2 <- activity_data %>%
        group_by(interval) %>%
        summarise(avg_step = mean(steps, na.rm = TRUE))

plot(
        dt2$interval,
        dt2$avg_step,
        type = "l",
        xlab = "",
        ylab = ""
        ) +
        title(main = "Average Daily Activity Pattern", xlab = "Interval", ylab = "Average number of step
```




```
## integer(0)
```

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

```
subset(dt2,avg_step == max(dt2$avg_step))
```

```
## # A tibble: 1 x 2
##   interval avg_step
##      <dbl>    <dbl>
## 1      835     206.
```

Imputing missing values

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(activity_data))
```

```
## [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.

```
fill <- activity_data %>%
                group_by(interval) %>%
                summarise(mean_step_of_the_interval = median(steps, na.rm = TRUE))
```

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

```
merged_dt <- merge(activity_data,fill,by.x = "interval", by.y = "interval")
new_dt <- merged_dt %>%
                transmute(steps = ifelse(is.na(steps), mean_step_of_the_interval,steps),
                     date,
                     interval)
```

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. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```
dt_to_plot <- new_dt %>%
                group_by(date) %>%
                summarise(total_step_per_day = sum(steps))
```

```
mean(dt_to_plot$total_step_per_day)
```

```
## [1] 9503.869
```

```
median(dt_to_plot$total_step_per_day)
```

```
## [1] 10395
```

As we can see, comparing to previous results, the mean increases but the median remain the same. Imputing missing data will increaese the mean of estimates of the total daily number of steps.

Are there differences in activity patterns between weekdays and weekends?

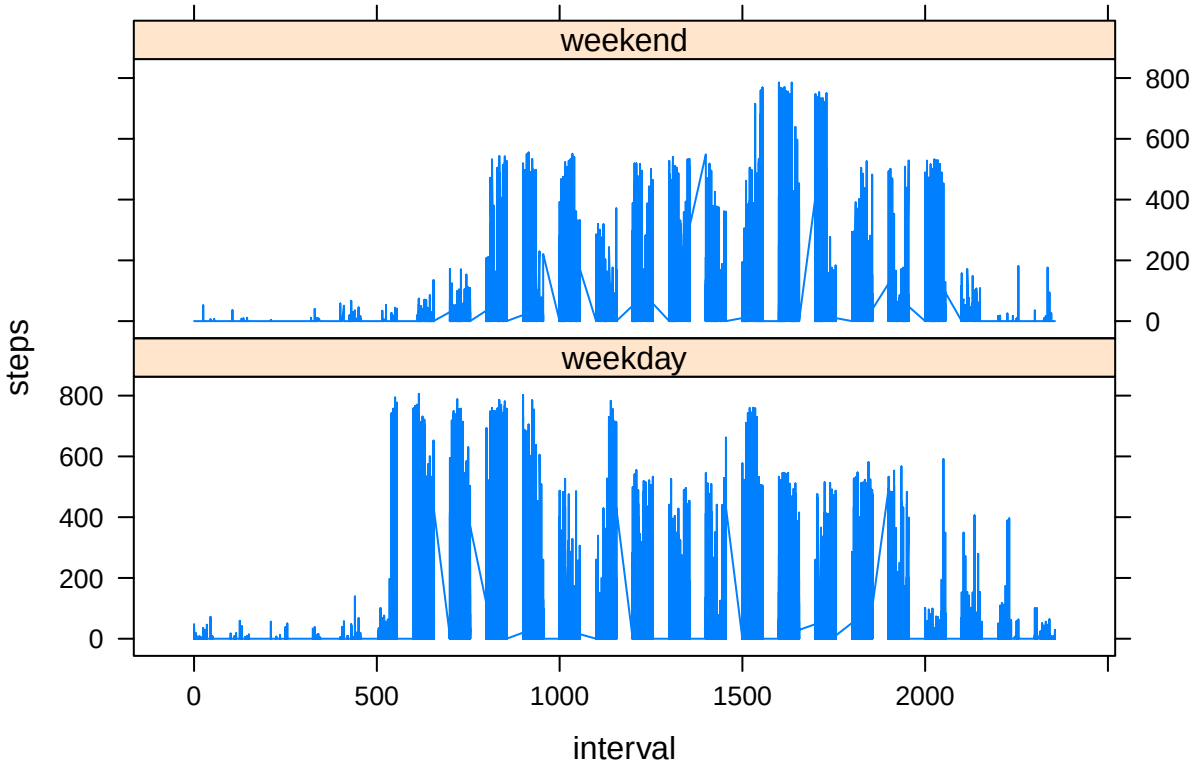
1. Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
weekday <- c("Mon", "Tue", "Wed", "Thu", "Fri")
new_dt$weekday <- ifelse(weekdays(new_dt$date, abbreviate = T) %in% weekday, "weekday", "weekend")
new_dt$weekday <- as.factor(new_dt$weekday)
levels(new_dt$weekday)
```

```
## [1] "weekday" "weekend"
```

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).

```
library(lattice)
xyplot(steps~interval|weekday, data = new_dt, type = "l", layout=c(1,2))
```


weekend
weekday
steps
interval
0
200
400
600
800
0
500
1000
1500
2000