Reproducible Research: Peer Assessment 1

```
echo = TRUE  # Always make code visible
```

Loading and preprocessing the data

```
unzip("activity.zip")
data <- read.csv("activity.csv", colClasses = c("integer", "Date", "factor"))
data$month <- as.numeric(format(data$date, "%m"))
noNA <- na.omit(data)
rownames(noNA) <- 1:nrow(noNA)
head(noNA)
```

```
##   steps       date interval month
## 1     0 2012-10-02        0    10
## 2     0 2012-10-02        5    10
## 3     0 2012-10-02       10    10
## 4     0 2012-10-02       15    10
## 5     0 2012-10-02       20    10
## 6     0 2012-10-02       25    10
```

```
## [1] 15264     4
```

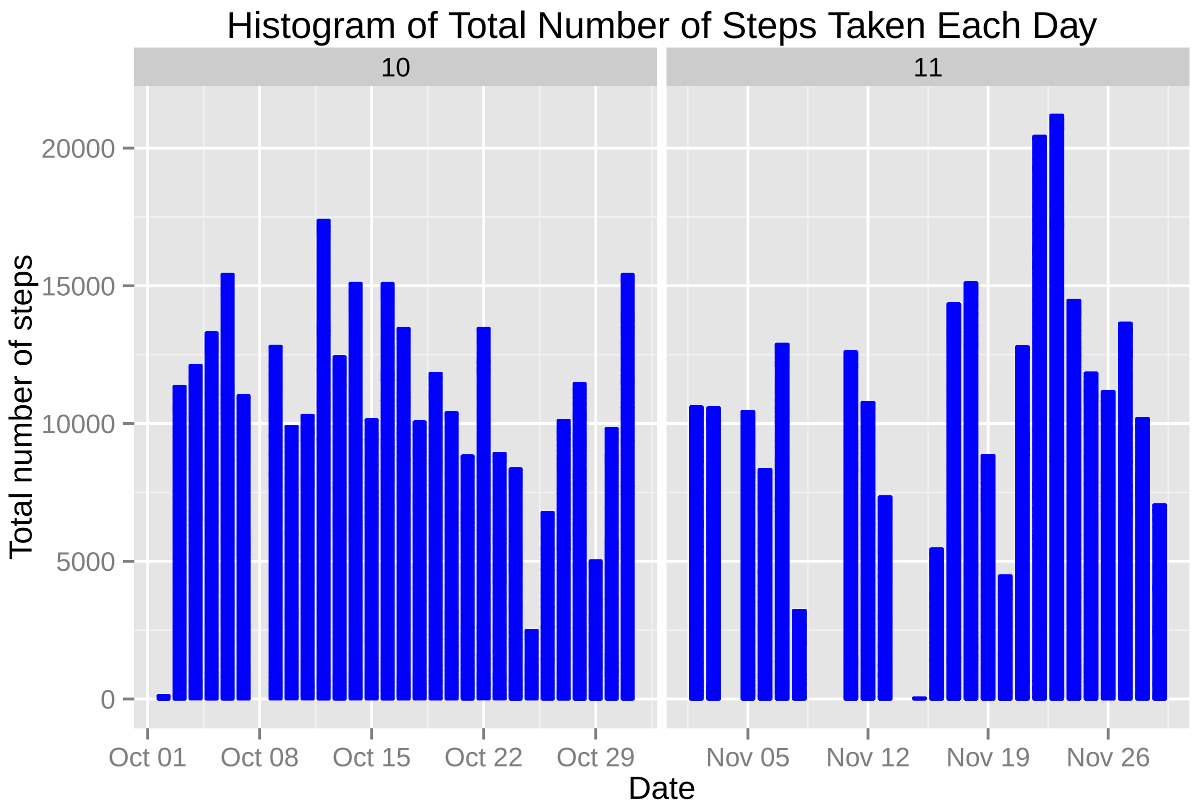
What is mean total number of steps taken per day?

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.

If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day

```
ggplot(noNA, aes(date, steps)) + geom_bar(stat = "identity", colour = "blue", fill = "blue", width = 0.7
```



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

```
totalSteps <- aggregate(noNA$steps, list(Date = noNA$date), FUN = "sum")$x
mean(totalSteps)
```

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

Median total number of steps taken per day:

```
median(totalSteps)
```

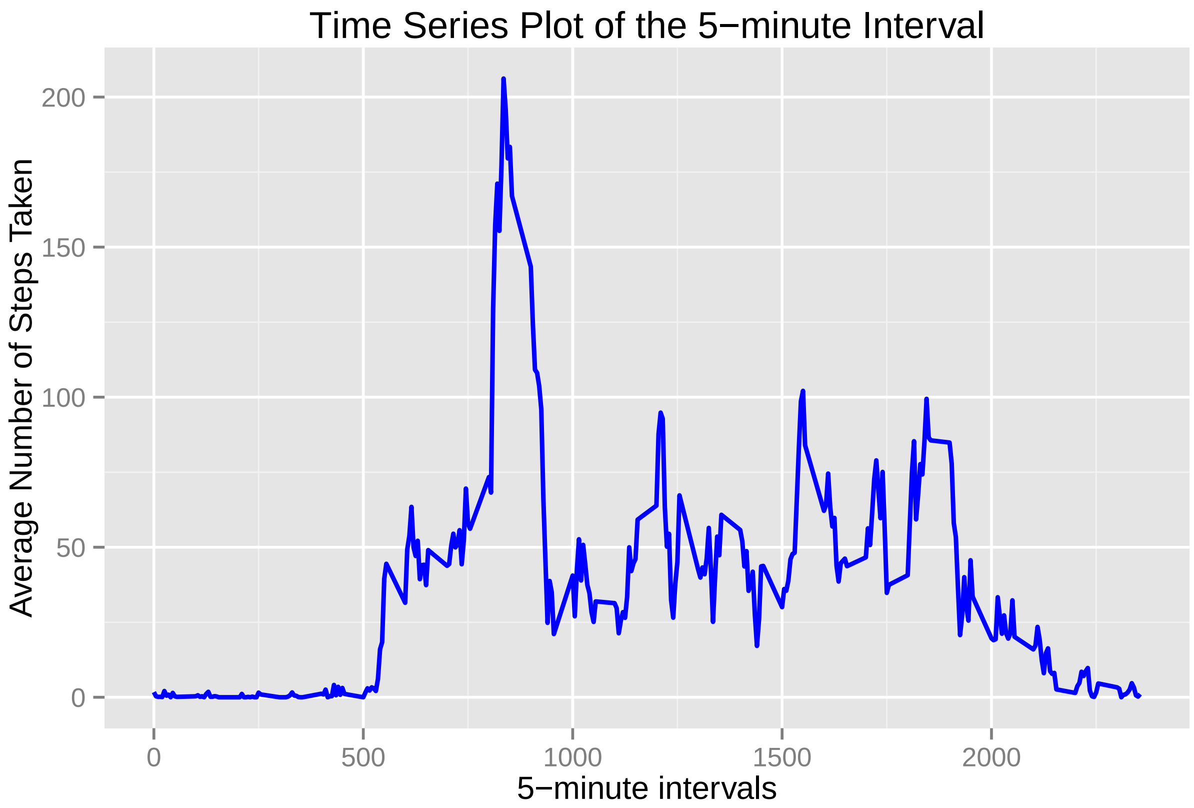
```
## [1] 10765
```

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)

```
avgSteps <- aggregate(noNA$steps, list(interval = as.numeric(as.character(noNA$interval))), FUN = "mean
names(avgSteps)[2] <- "meanOfSteps"

ggplot(avgSteps, aes(interval, meanOfSteps)) + geom_line(color = "blue", size = 0.8) + labs(title = "Ti
```



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

```
avgSteps[avgSteps$meanOfSteps == max(avgSteps$meanOfSteps), ]
```

```
##     interval meanOfSteps
## 104      835    206.1698
```

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))
```

```
## [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. My strategy is to use the mean for that 5-minute interval to fill each NA value in the steps column.

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

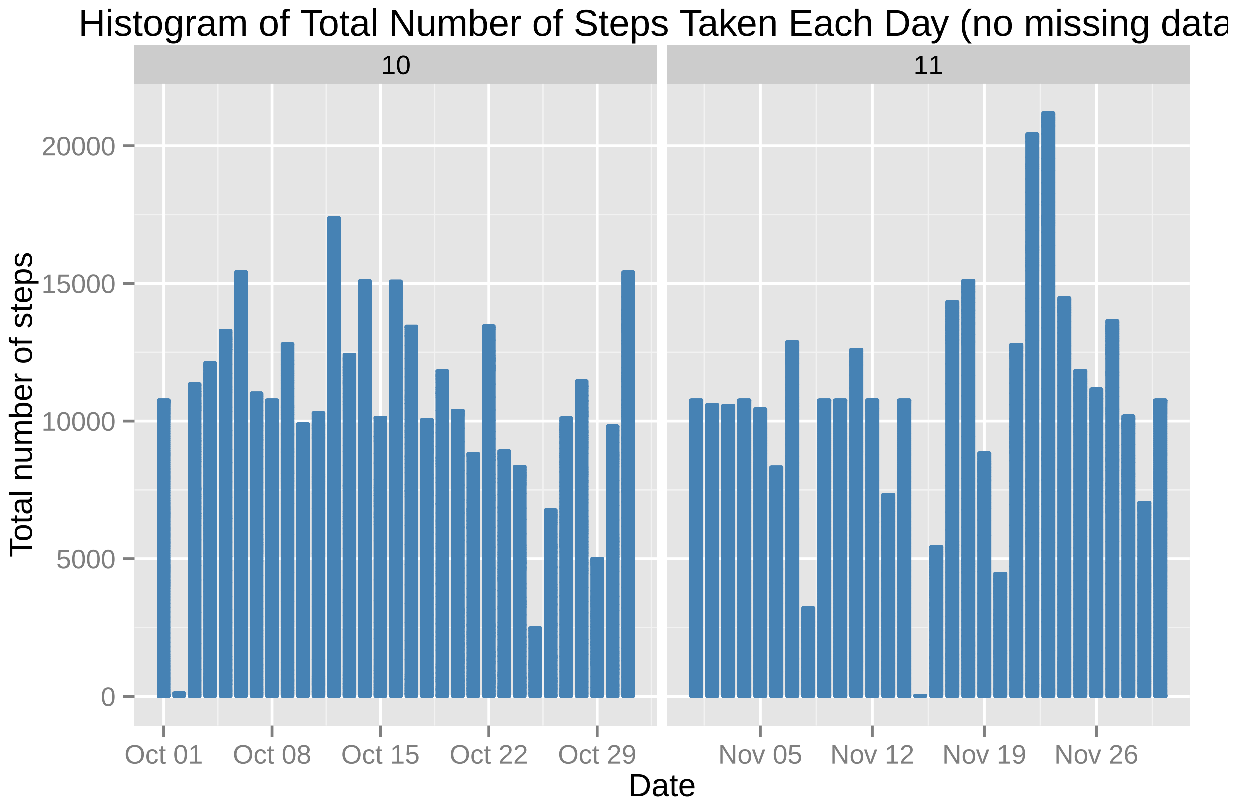
```
newData <- data
for (i in 1:nrow(newData)) {
    if (is.na(newData$steps[i])) {
        newData$steps[i] <- avgSteps[which(newData$interval[i] == avgSteps$interval), ]$meanOfSteps
    }
}

head(newData)
```

```
##       steps       date interval month
## 1 1.7169811 2012-10-01        0    10
## 2 0.3396226 2012-10-01        5    10
## 3 0.1320755 2012-10-01       10    10
## 4 0.1509434 2012-10-01       15    10
## 5 0.0754717 2012-10-01       20    10
## 6 2.0943396 2012-10-01       25    10
```

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

```
r ggplot(newData, aes(date, steps)) + geom_bar(stat = "identity", colour = "steelblue",fill
= "steelblue", width = 0.7) + facet_grid(.  ~ month, scales = "free") + labs(title =
"Histogram of Total Number of Steps Taken Each Day (no missing data)", x = "Date", y =
"Total number of steps")
```



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?

Mean total number of steps taken per day:

```
newTotalSteps <- aggregate(newData$steps,
                           list(Date = newData$date),
                           FUN = "sum")$x
newMean <- mean(newTotalSteps)
newMean
```

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

Median total number of steps taken per day:

```
newMedian <- median(newTotalSteps)
newMedian
```

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

Compare them with the two before imputing missing data:

```
oldMean <- mean(totalSteps)
oldMedian <- median(totalSteps)
newMean - oldMean
```

```
## [1] 0
```

```
newMedian - oldMedian
```

```
## [1] 1.188679
```

So, after imputing the missing data, the new mean of total steps taken per day is the same as that of the old mean; the new median of total steps taken per day is greater than that of the old median.

Are there differences in activity patterns between weekdays and weekends?

```
head(newData)
```

```
##       steps       date interval month
## 1 1.7169811 2012-10-01        0    10
## 2 0.3396226 2012-10-01        5    10
## 3 0.1320755 2012-10-01       10    10
## 4 0.1509434 2012-10-01       15    10
## 5 0.0754717 2012-10-01       20    10
## 6 2.0943396 2012-10-01       25    10
```

```
newData$weekdays <- factor(format(newData$date, "%A"))
levels(newData$weekdays)
```

```
## [1] "Friday"    "Monday"    "Saturday"  "Sunday"    "Thursday"  "Tuesday"
## [7] "Wednesday"
```

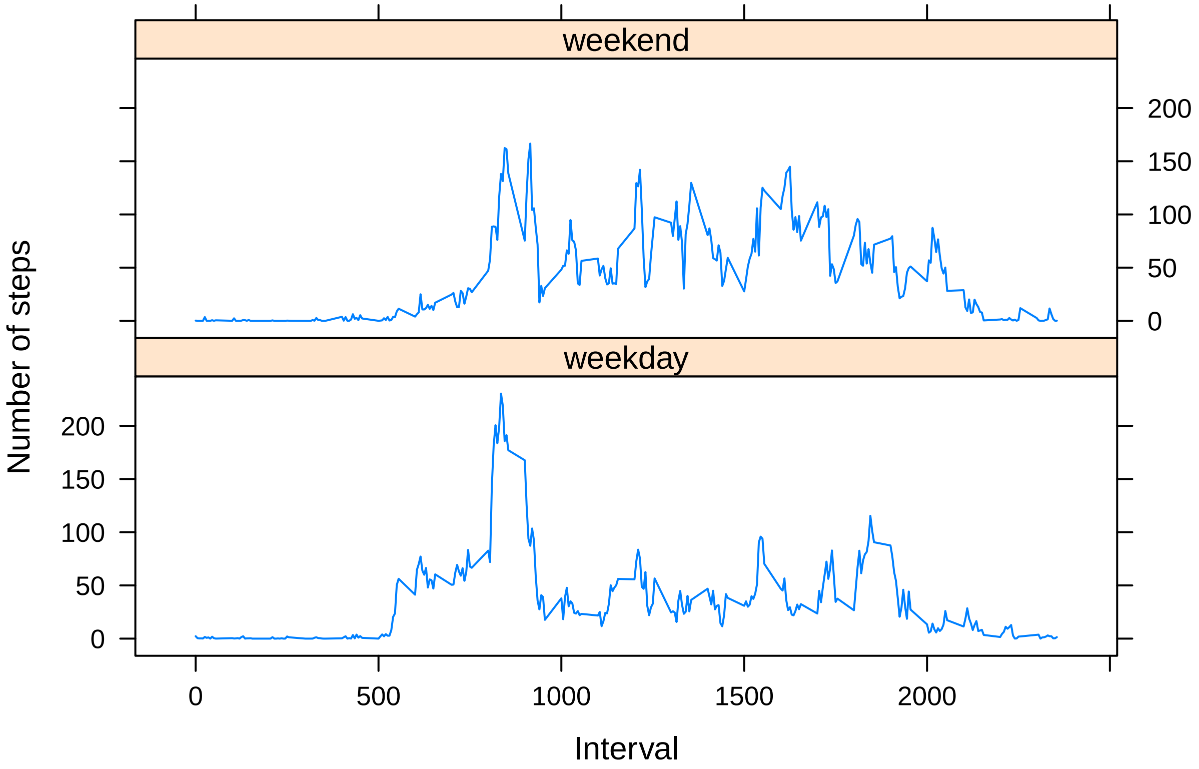
```
levels(newData$weekdays) <- list(weekday = c("Monday", "Tuesday",
                                             "Wednesday",
                                             "Thursday", "Friday"),
                                 weekend = c("Saturday", "Sunday"))
levels(newData$weekdays)
```

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

```
table(newData$weekdays)
```

```
##
## weekday weekend
##   12960    4608
```

```
avgSteps <- aggregate(newData$steps,
                      list(interval = as.numeric(as.character(newData$interval)),
                           weekdays = newData$weekdays),
                      FUN = "mean")
names(avgSteps)[3] <- "meanOfSteps"
library(lattice)
xyplot(avgSteps$meanOfSteps ~ avgSteps$interval | avgSteps$weekdays,
       layout = c(1, 2), type = "l",
       xlab = "Interval", ylab = "Number of steps")
```


weekend
weekday
Number of steps
Interval