# Reproducible Research: Peer Assessment 1

## Loading and preprocessing the data

1. Load the data (i.e. read.csv())
2. Process/transform the data (if necessary) into a format suitable for your analysis

```
activityData <- read.csv("activity.csv", colClasses = c("integer", "Date", "facto
r"))
activityData$month <- as.numeric(format(activityData$date, "%m"))
noNAs <- na.omit(activityData)
```

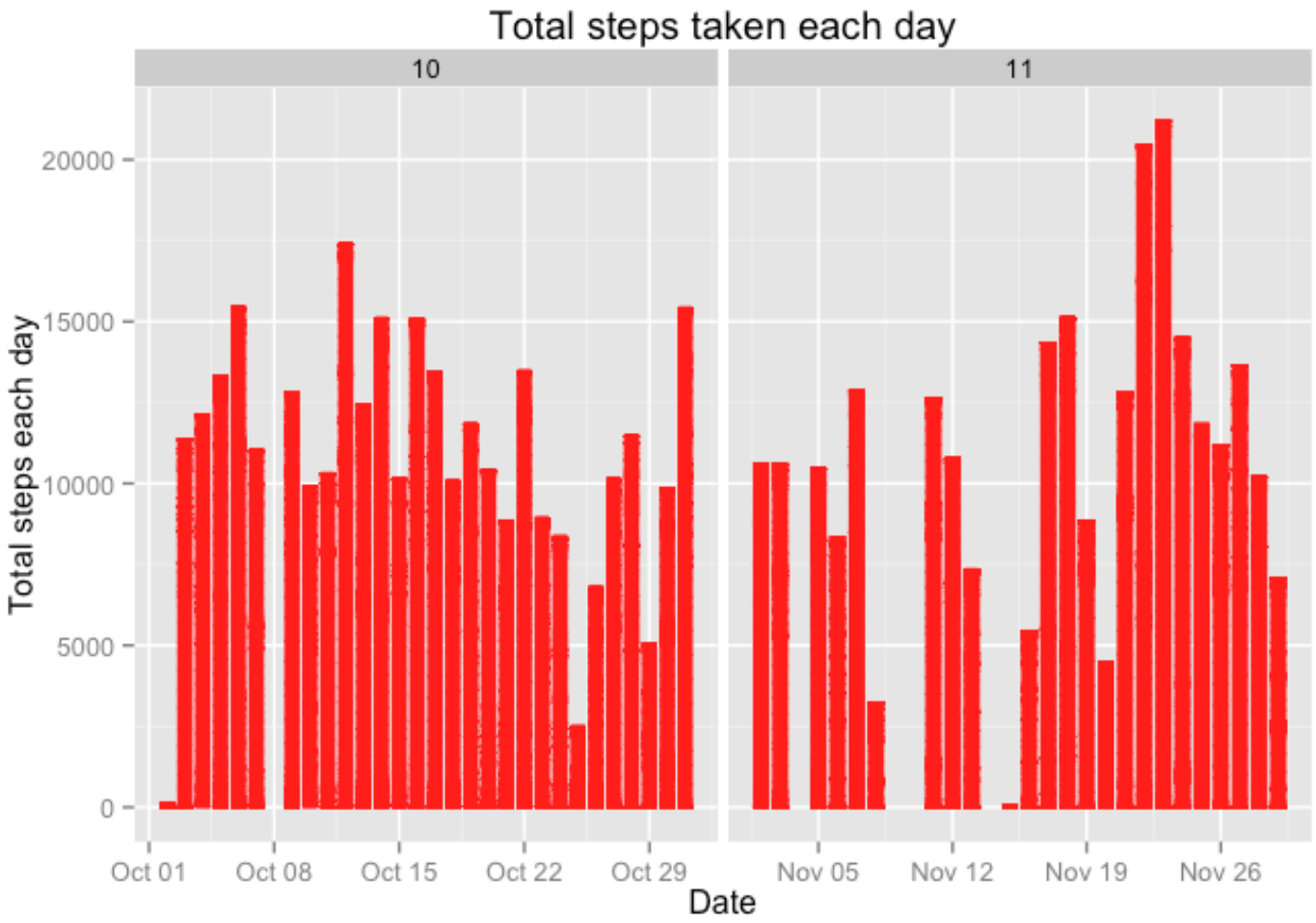
## What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day

```
stepsTotal <- aggregate(steps ~ date, data = activityData, sum, na.rm = TRUE)
```

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

```
library(ggplot2)
ggplot(noNAs, aes(date, steps)) + geom_bar(stat = "identity", colour = "red", fill
= "red", width = 0.7) + facet_grid(. ~ month, scales = "free") + labs(title = "Tot
al steps taken each day", x = "Date", y = "Total steps each day")
```




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

```
mean(stepsTotal$steps)
```

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

```
## Should be 10766.19
median(stepsTotal$steps)
```

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

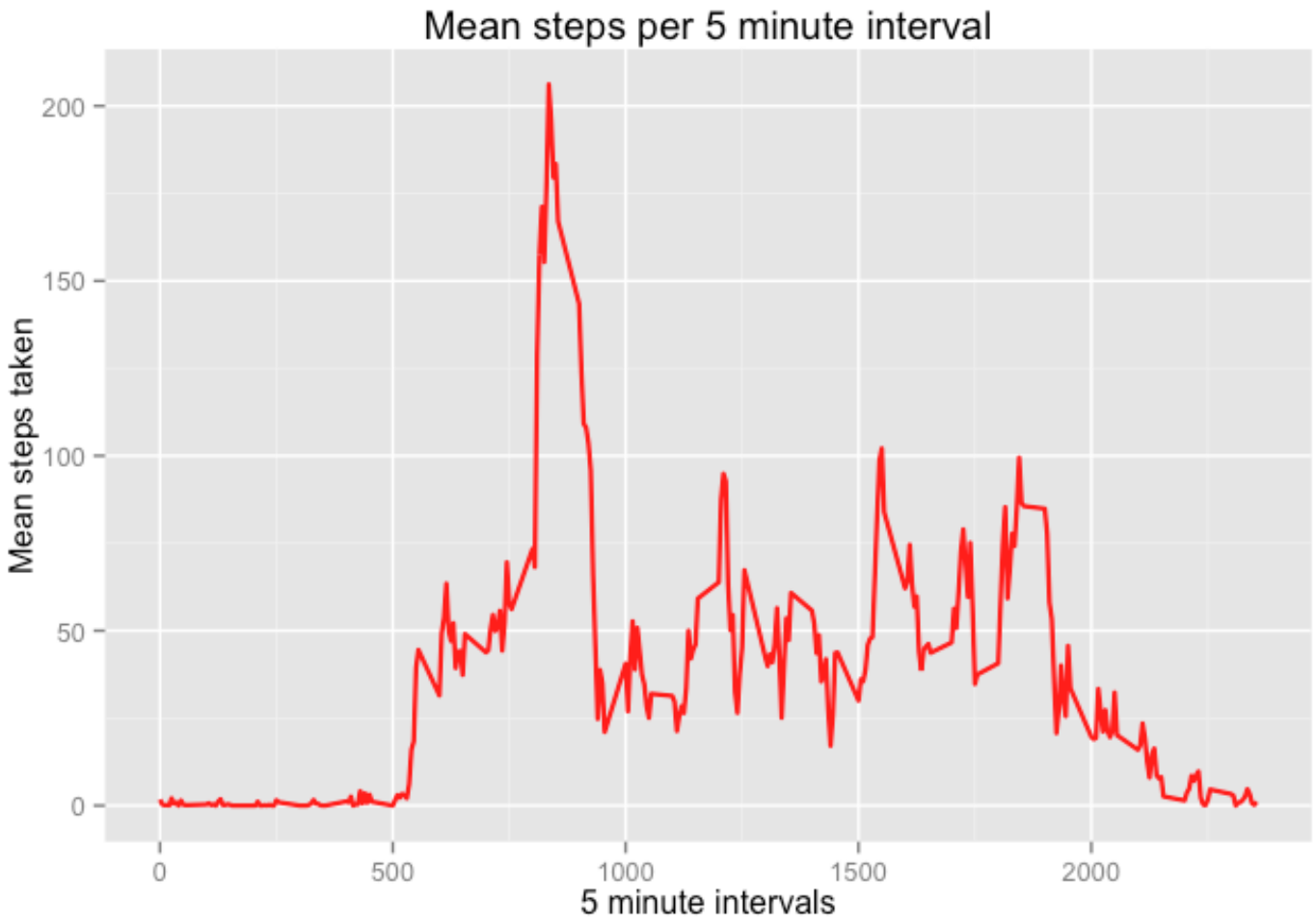
```
## Should be 10765
```

# 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)

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

ggplot(meanSteps, aes(interval, meanStepsTaken)) + geom_line(color = "red", size =
0.8) + labs(title = "Mean steps per 5 minute interval", x = "5 minute intervals",
y = "Mean steps taken")
```




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

```
timeSeries <- tapply(activityData$steps, activityData$interval, mean, na.rm = TRU
E)
maxInterval <- which.max(timeSeries)
names(maxInterval)
```

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

```
##Should be 835
```

# 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)

```
activityDataNAs <- sum(is.na(activityData))
activityDataNAs
```

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

```
##Should be 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.

```
stepsMean <- aggregate(steps ~ interval, data = activityData, FUN = mean)
fillNAs <- numeric()
for (i in 1:nrow(activityData)) {
    obs <- activityData[i, ]
    if (is.na(obs$steps)) {
        steps <- subset(stepsMean, interval == obs$interval)$steps
    } else {
        steps <- obs$steps
    }
    fillNAs <- c(fillNAs, steps)
}
```

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

```
newActivityData <- activityData
newActivityData$steps <- fillNAs
```

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?

ggplot(newActivityData, aes(date, steps)) + geom_bar(stat = “identity”, colour = “purple”, fill = “purple”, width = 0.7) + facet_grid(. ~ month, scales = “free”) + labs(title = “Total steps taken each day (missing values imputed)”, x = “Date”, y = “Total 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.

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

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

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

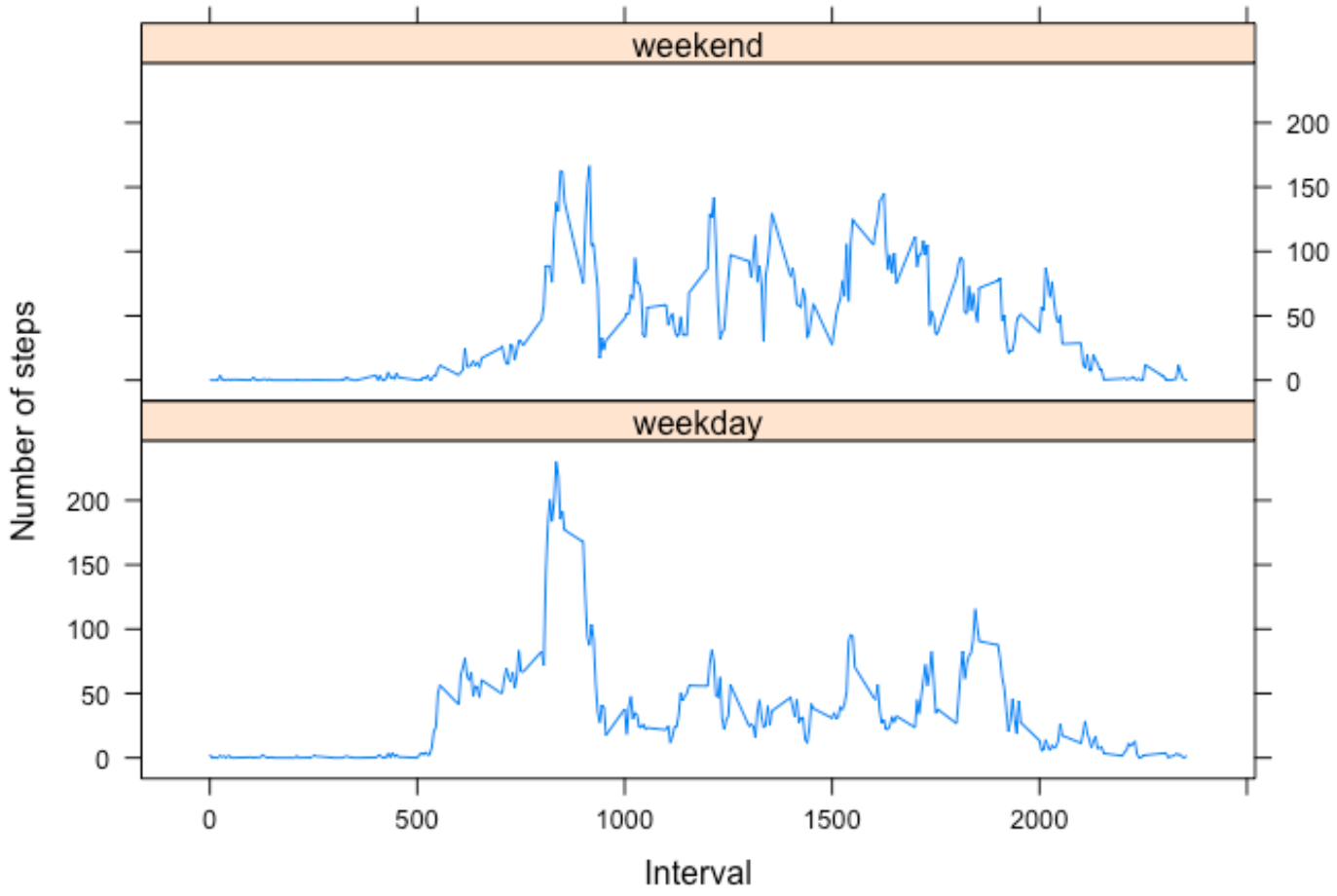
```
## [1] "weekday" "weekend"
```

```
table(newActivityData$weekdays)
```

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

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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
meanSteps <- aggregate(newActivityData$steps,
                       list(interval = as.numeric(as.character(newActivityData$inte
rval)),
                            weekdays = newActivityData$weekdays),
                       FUN = "mean")
names(meanSteps)[3] <- "meanStepsTaken3"
library(lattice)
xyplot(meanSteps$meanStepsTaken3 ~ meanSteps$interval | meanSteps$weekdays,
       layout = c(1, 2), type = "l",
       xlab = "Interval", ylab = "Number of steps")
```


weekend
weekday
Number of steps
Interval
0
50
100
150
200
0
500
1000
1500
2000