Reproducible Research: Peer Assessment 1

```
#Setting global options
```

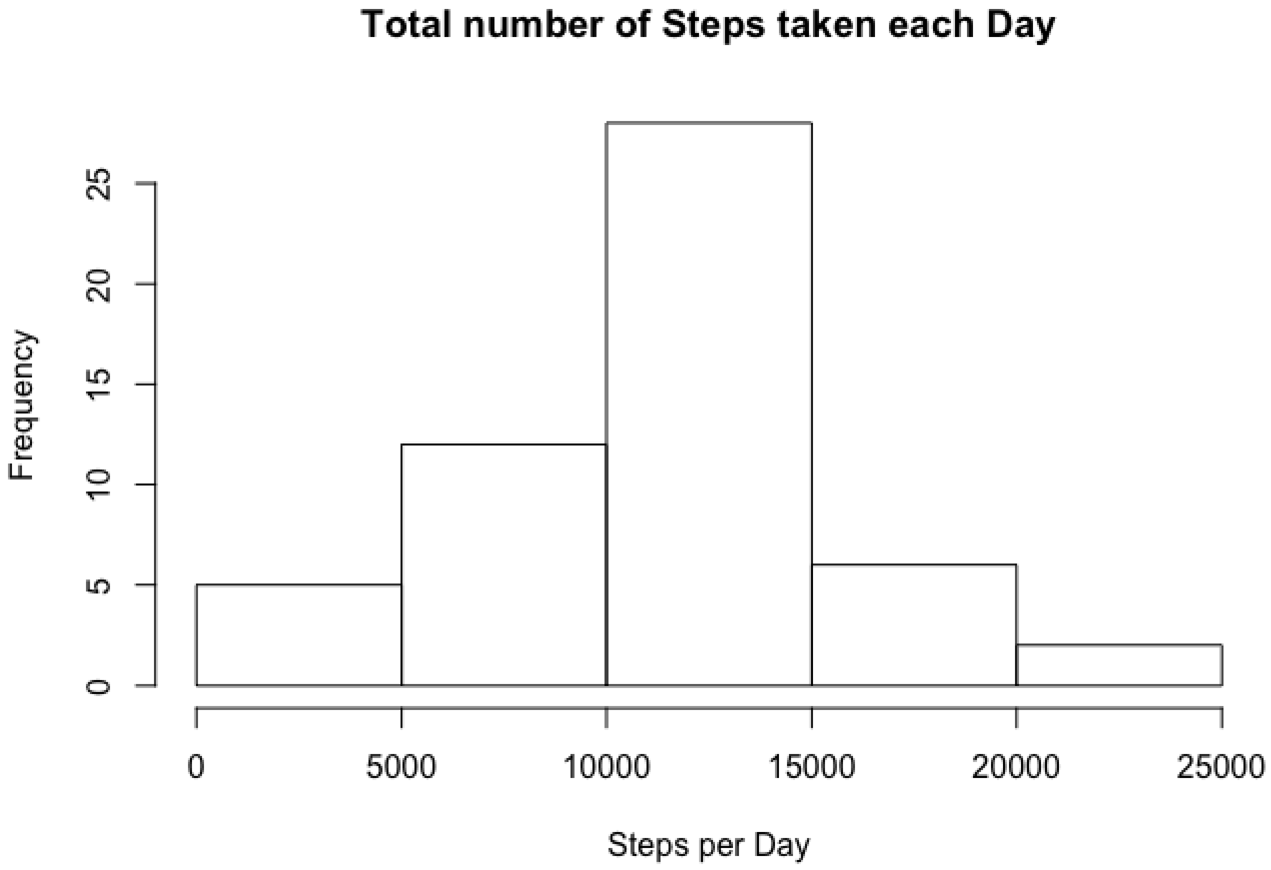
Loading and preprocessing the data

```
downURL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(downURL, destfile = "./activity.zip")
unzip("./activity.zip")
actdata <- read.csv(file = "./activity.csv", header = TRUE, colClasses = c("integer", "D
ate", "integer"))
```

What is mean total number of steps taken per day?

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

```
steps_pday <- tapply(actdata$steps, actdata$date, sum)
hist(steps_pday, main = "Total number of Steps taken each Day", xlab = "Steps per Day")
```



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

```
# mean
mean(steps_pday, na.rm = TRUE)
```

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

```
# median
median(steps_pday, 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)

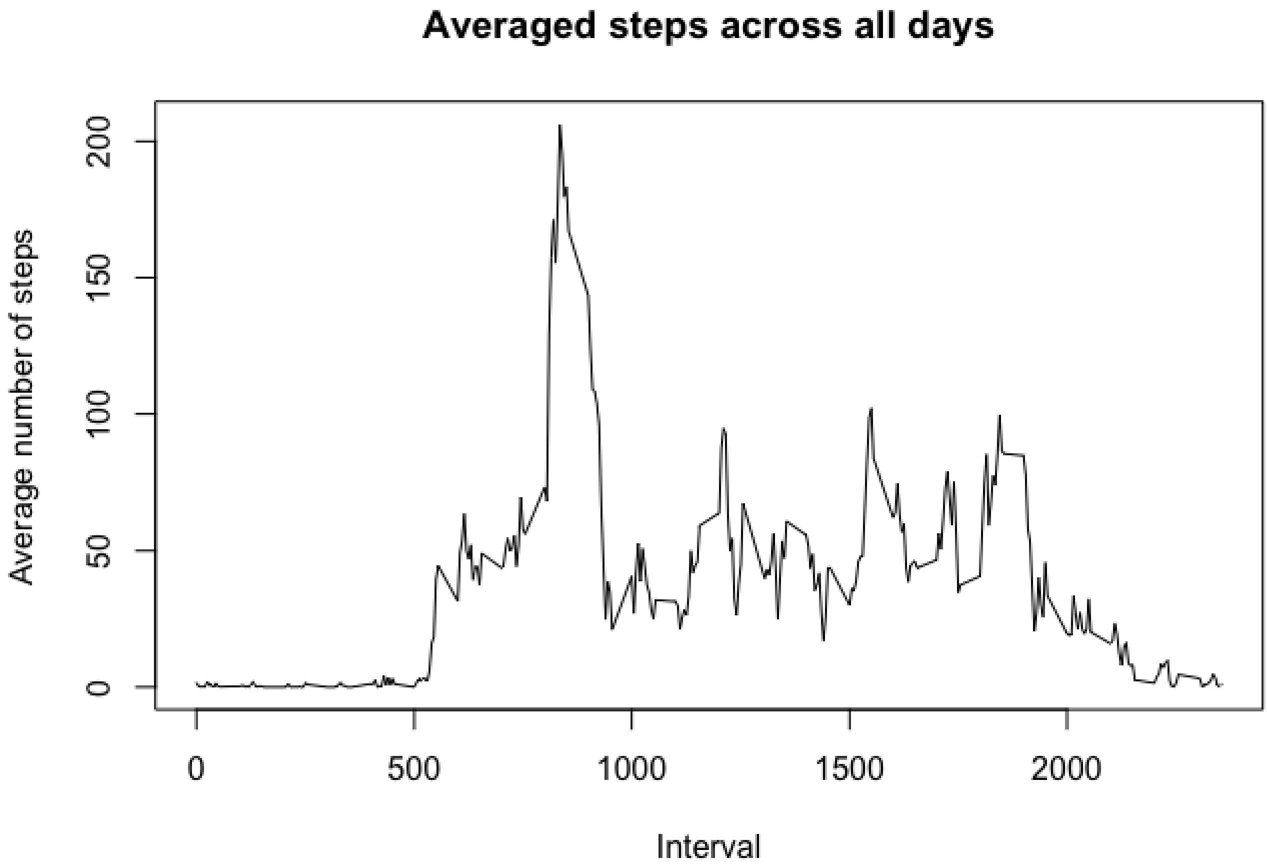
```
library(dplyr)
```

```
##
## 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
```

```
ndt <- actdata %>%
    group_by(interval) %>%
    summarise(averaged_steps = mean(steps, na.rm = TRUE))

plot(ndt$interval, ndt$averaged_steps, type = "l", main = "Averaged steps across all day
s", xlab = "Interval", ylab = "Average number of steps")
```



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

```
arrange(ndt, desc(averaged_steps))[1,]
```

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

```
## Interval 835-840 contains the maximum number of steps.
```

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)

```
library(dplyr)

#Split the original data into normal and NA data.
na_data <-
    actdata %>%
    mutate(completeCol = complete.cases(actdata)) %>%
    filter(completeCol == FALSE)  %>%
    select(-completeCol)
normal_data <-
    actdata %>%
    mutate(completeCol = complete.cases(actdata)) %>%
    filter(completeCol == TRUE) %>%
    select(-completeCol)

nrow(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. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

```
# We will use the mean of that 5-minute interval instead of the missing values,
# which is stored in a variable "ndt$averaged_steps".
```

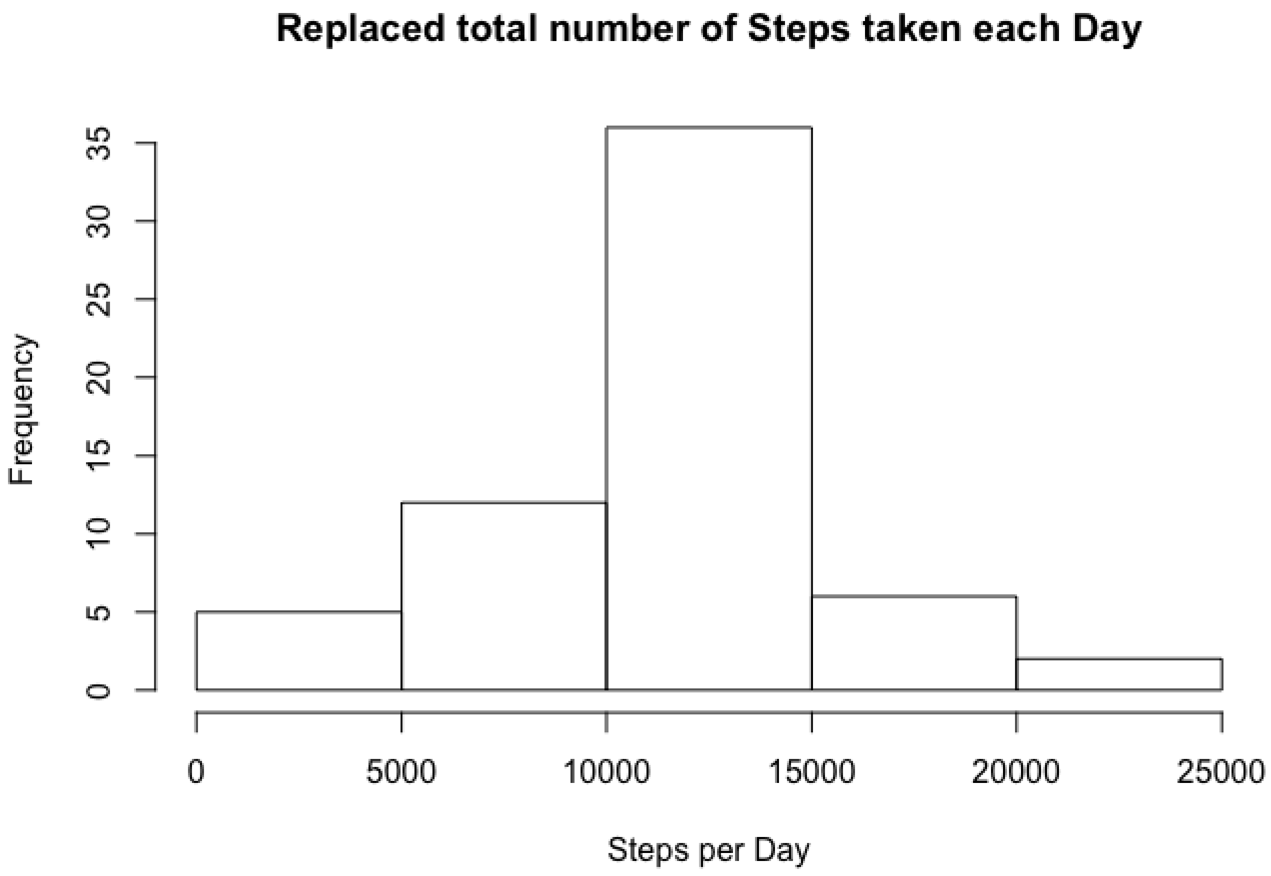
3.Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
replaced_na_data <-
    merge(na_data, ndt, by = "interval") %>%
    select(steps = averaged_steps, date, interval)

replaced_data <-
    rbind(normal_data, replaced_na_data) %>%
    arrange(date, interval)

head(replaced_data)
```

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

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?

```
replaced_steps_pday <-
    tapply(replaced_data$steps, replaced_data$date, sum)
hist(replaced_steps_pday, main = "Replaced total number of Steps taken each Day", xlab =
 "Steps per Day")
```



```
# mean
mean(replaced_steps_pday)
```

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

```
# median
median(replaced_steps_pday)
```

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

```
# mean of the imputed data differs by :
x <- abs(mean(replaced_steps_pday) - mean(steps_pday, na.rm = TRUE))/mean(replaced_steps
_pday)
sprintf("%.0f%%", x * 100)
```

```
## [1] "0%"
```

```
# median of the imputed data differs by :
y <- abs(median(replaced_steps_pday) - median(steps_pday, na.rm = TRUE))/median(replaced
_steps_pday)
sprintf("%.0f%%", y * 100)
```

```
## [1] "0%"
```

Are there differences in activity patterns between weekdays and weekends?

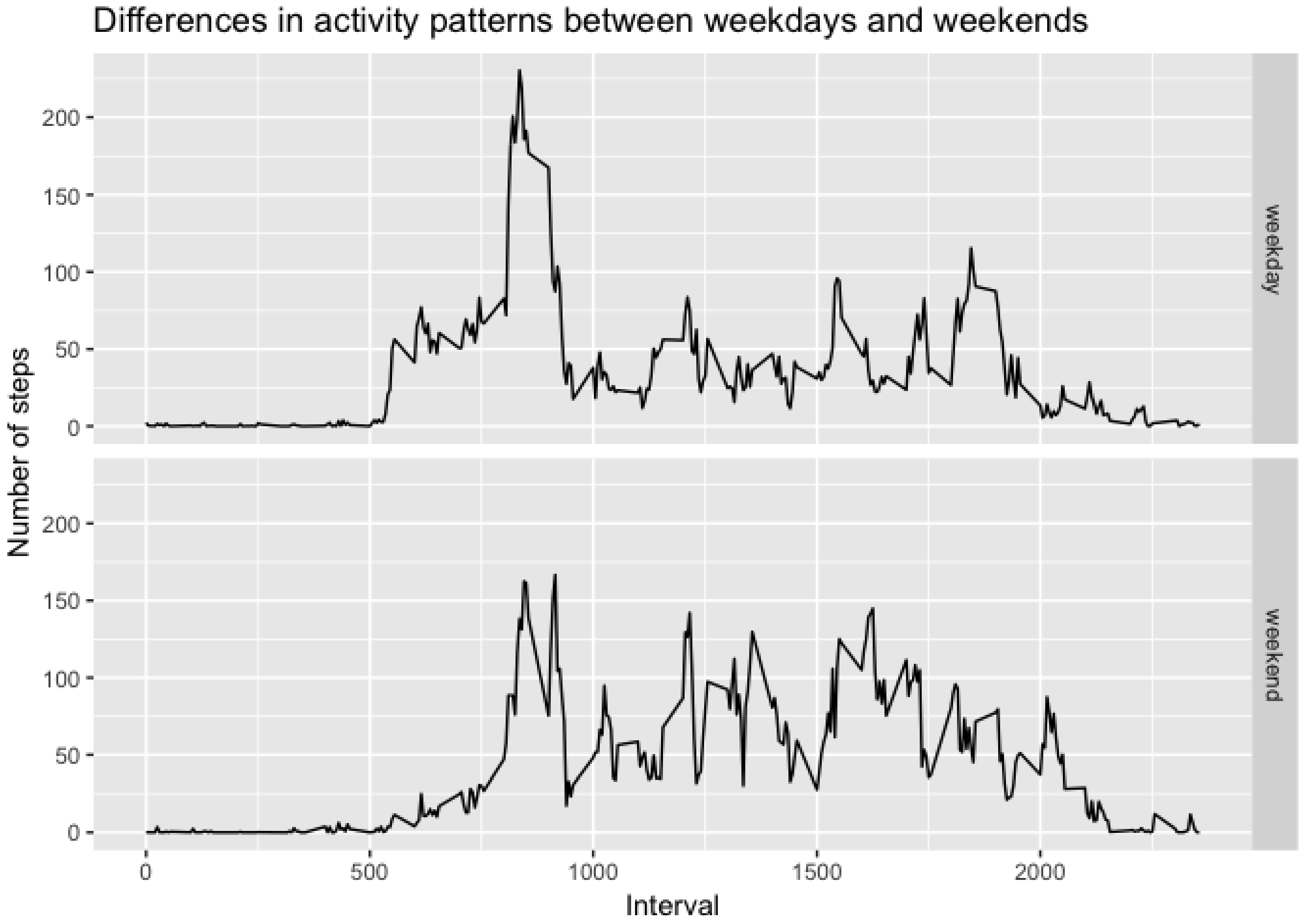
1. Create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
day_data <- replaced_data %>%
    mutate(day = weekdays(date))

factor_data <-
    replaced_data %>%
    mutate(day = weekdays(date)) %>%
    mutate(day = ifelse(day %in% c("Saturday", "Sunday"), "weekend", "weekday"))
factor_data$day <- as.factor(factor_data$day)

# Take a look at the new factor dataset with two day levels
str(factor_data)
```

```
## 'data.frame':    17568 obs. of  4 variables:
##  $ steps   : num  1.717 0.3396 0.1321 0.1509 0.0755 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ day     : Factor w/ 2 levels "weekday","weekend": 1 1 1 1 1 1 1 1 1 1 ...
```

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). The plot should look something like the following, which was created using simulated data:

```
library(ggplot2)

week_data <- factor_data %>%
    group_by(day, interval) %>%
    summarise(averaged_steps = mean(steps))

qplot(interval, averaged_steps, data = week_data, geom = "path", facets = day~., ylab =
"Number of steps", xlab = "Interval", main = "Differences in activity patterns between w
eekdays and weekends")
```


Differences in activity patterns between weekdays and weekends
Number of steps
200
150
100
50
0
weekday
weekend
0
500
1000
1500
2000
Interval