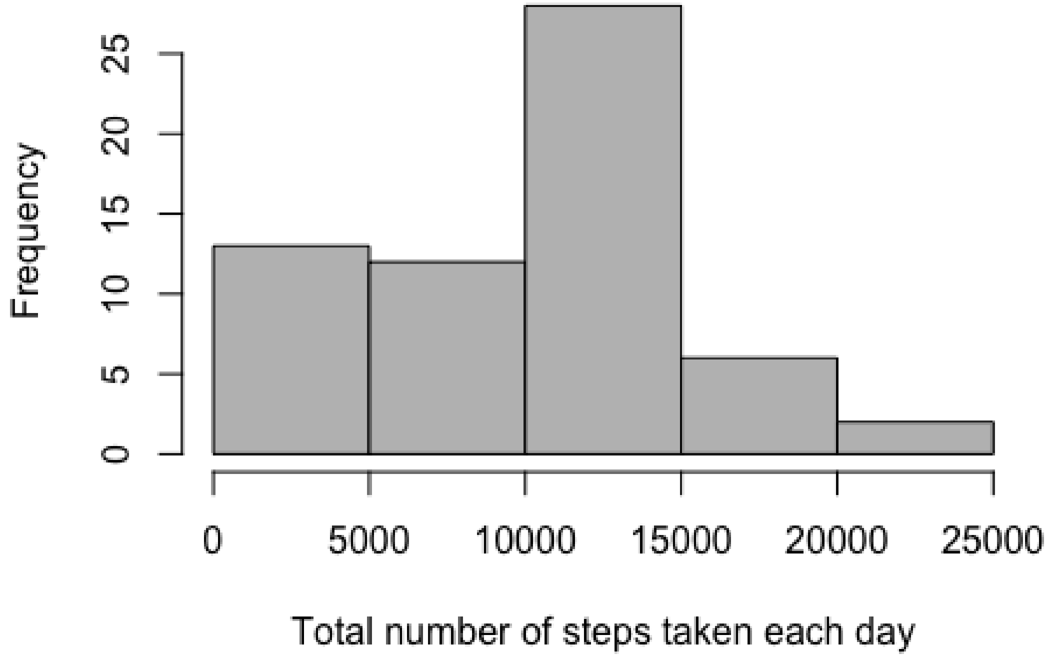
Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

```r
fileurl <-
"https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
if (!file.exists("activity.zip")) {
  download.file(fileurl, destfile = "activity.zip", method = "curl")
}

file <- "activity.csv"
if (!file.exists(file)) {
  unzip(zipfile = "activity.zip")
}
activity_dta <- read.csv(file, header = TRUE)
```

What is mean total number of steps taken per day?

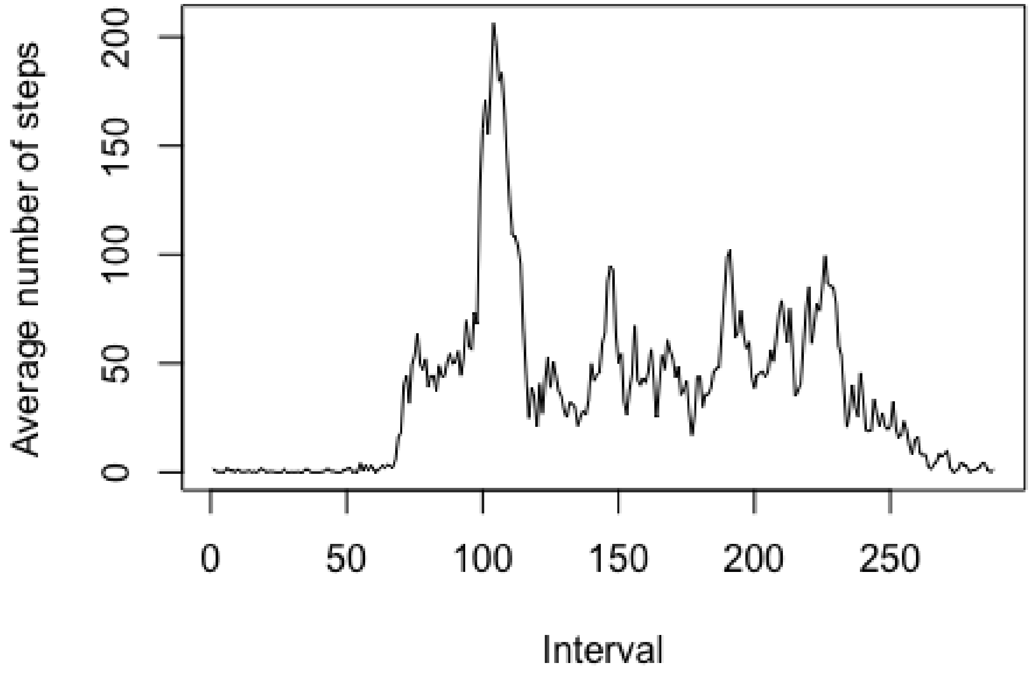
```r
# 1. Calculate the total number of steps taken per day
total_steps_per_day <- tapply(activity_dta$steps, activity_dta$date, FUN =
sum, na.rm = TRUE)
# 2. Make a histogram of the total number of steps taken each day
hist(total_steps_per_day,
    main = "Histogram of the total number of steps taken each day",
    xlab = "Total number of steps taken each day",
    col  = "gray")
```



```
# 3. Calculate and report the mean and median of the total number of steps
taken per day
paste("Mean total number of steps taken per day:", mean(total_steps_per_day,
na.rm = TRUE))

## [1] "Mean total number of steps taken per day: 9354.22950819672"

paste("Median total number of steps taken per day:",
median(total_steps_per_day, na.rm = TRUE))

## [1] "Median total number of steps taken per day: 10395"
```

What is the average daily activity pattern?

```
# 1. Make a time series plot of the 5-minute interval (x-axis) and the
average number of steps taken, averaged across all days (y-axis)

avg_steps_per_interval <- tapply(activity_dta$steps, activity_dta$interval,
FUN = mean, na.rm = TRUE)
plot(avg_steps_per_interval, type = "l",
     xlab = "Interval",
     ylab = "Average number of steps",
     main = "Time series plot of average number of steps per interval")
```

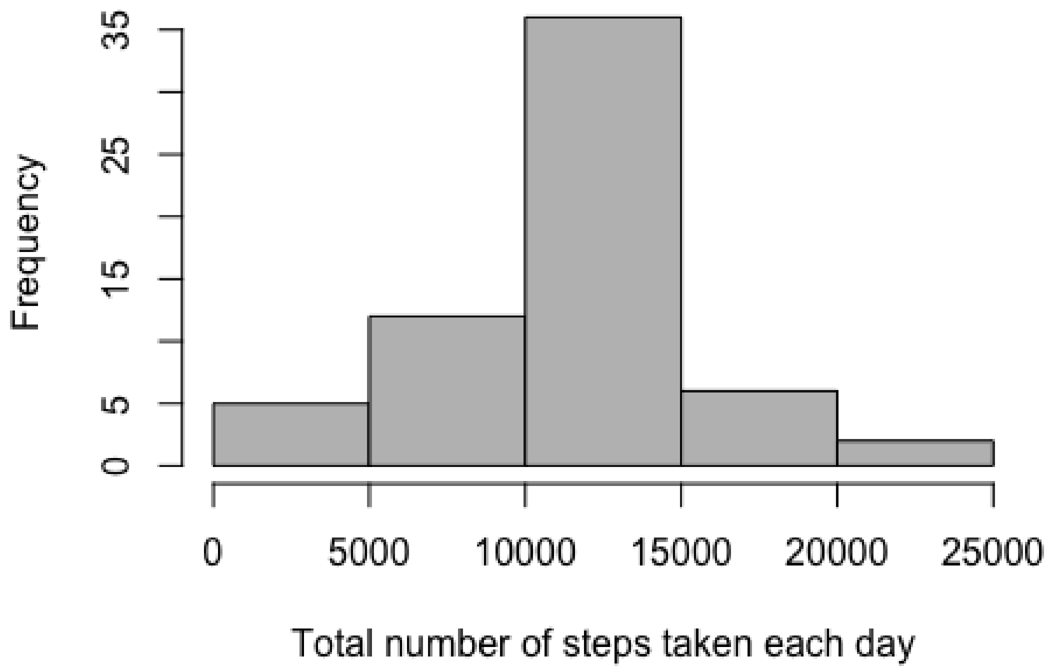


```
# 2. Which 5-minute interval, on average across all the days in the dataset,
contains the maximum number of steps?
interval_with_max_steps <- avg_steps_per_interval[avg_steps_per_interval ==
max(avg_steps_per_interval)]
paste("The 5-minute interval with maximum number of steps:",
names(interval_with_max_steps))

## [1] "The 5-minute interval with maximum number of steps: 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)
paste("The total number of missing values in the dataset:",
sum(is.na(activity_dta)))

## [1] "The total number of missing values in the dataset: 2304"

# 2. Devise a strategy for filling in all of the missing values in the
dataset. The strategy does not need to be sophisticated.
# Using the mean for the corresponding 5-minute interval to fill in all of
the missing values in the dataset.

# 3. Create a new dataset that is equal to the original dataset but with the
missing data filled in.
imputed_data <- activity_dta
for (i in 1:nrow(imputed_data)) {
  if (is.na(imputed_data$steps[i])) {
```

```r
    interval = as.character(imputed_data[i, "interval"])
    imputed_data$steps[i] <- avg_steps_per_interval[interval]
  }
}
paste("After imputation, the total number of missing values in the dataset:",
sum(is.na(imputed_data)))

## [1] "After imputation, the total number of missing values in the dataset:
0"

# 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_per_day_imp <- tapply(imputed_data$steps, imputed_data$date, FUN
= sum)
hist(total_steps_per_day_imp,
     main = "Histogram of the total number of steps taken each day: Imputed
Dataset",
     xlab = "Total number of steps taken each day",
     col  = "gray")
```



```r
paste("Mean total number of steps taken per day:",
mean(total_steps_per_day_imp))

## [1] "Mean total number of steps taken per day: 10766.1886792453"

paste("Median total number of steps taken per day:",
median(total_steps_per_day_imp))
```

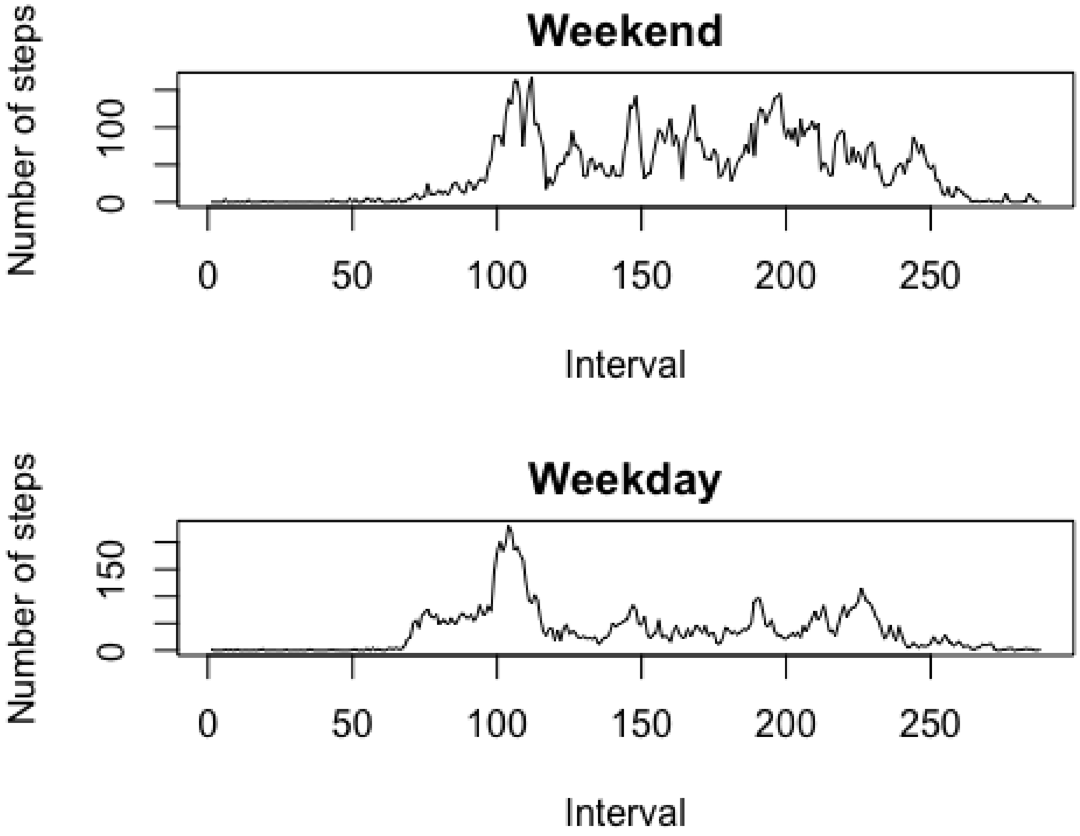
```
## [1] "Median total number of steps taken per day: 10766.1886792453"
```

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?

The mean of total number of steps per day does differ the from the estimate from the first part. The median of total number of steps has also changed. Imputing the missing values has increased the total daily number of steps, thereby affecting (increasing) the frequencies of some bins, as can be seen from the histogram.

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.
imputed_data$day <- weekdays(as.Date(imputed_data$date, format = "%Y-%m-%d"))
imputed_data$daytype <- factor(ifelse(imputed_data$day == "Saturday" |
imputed_data$day == "Sunday", "weekend", "weekday"), levels = c("weekday",
"weekend"))

# 2. Make a panel plot containing a time series plot of the 5-minute interval
(x-axis) and the average number of steps taken, averaged across all weekday
days or weekend days (y-axis).
imputed_data_weekday <- imputed_data[imputed_data$daytype == "weekday", ]
imputed_data_weekend <- imputed_data[imputed_data$daytype == "weekend", ]

avg_steps_per_day_imp_weekday = tapply(imputed_data_weekday$steps,
imputed_data_weekday$interval, FUN = mean)
avg_steps_per_day_imp_weekend = tapply(imputed_data_weekend$steps,
imputed_data_weekend$interval, FUN = mean)

par(mfrow = c(2, 1), mar = c(5, 4, 2, 1))

plot(avg_steps_per_day_imp_weekend, type = "l",
     xlab = "Interval",
     ylab = "Number of steps",
     main = "Weekend")

plot(avg_steps_per_day_imp_weekday, type = "l",
     xlab = "Interval",
     ylab = "Number of steps",
     main = "Weekday")
```



Weekend

Weekday