Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

```
#Import Libraries

library(data.table)
library(ggplot2)

#Get Data
fileUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(fileUrl, destfile = paste0(getwd(), '/repdata%2Fdata%2Factivity.zip'))
unzip("repdata%2Fdata%2Factivity.zip",exdir = "data")

activityDT <- data.table::fread(input = "data/activity.csv")
```

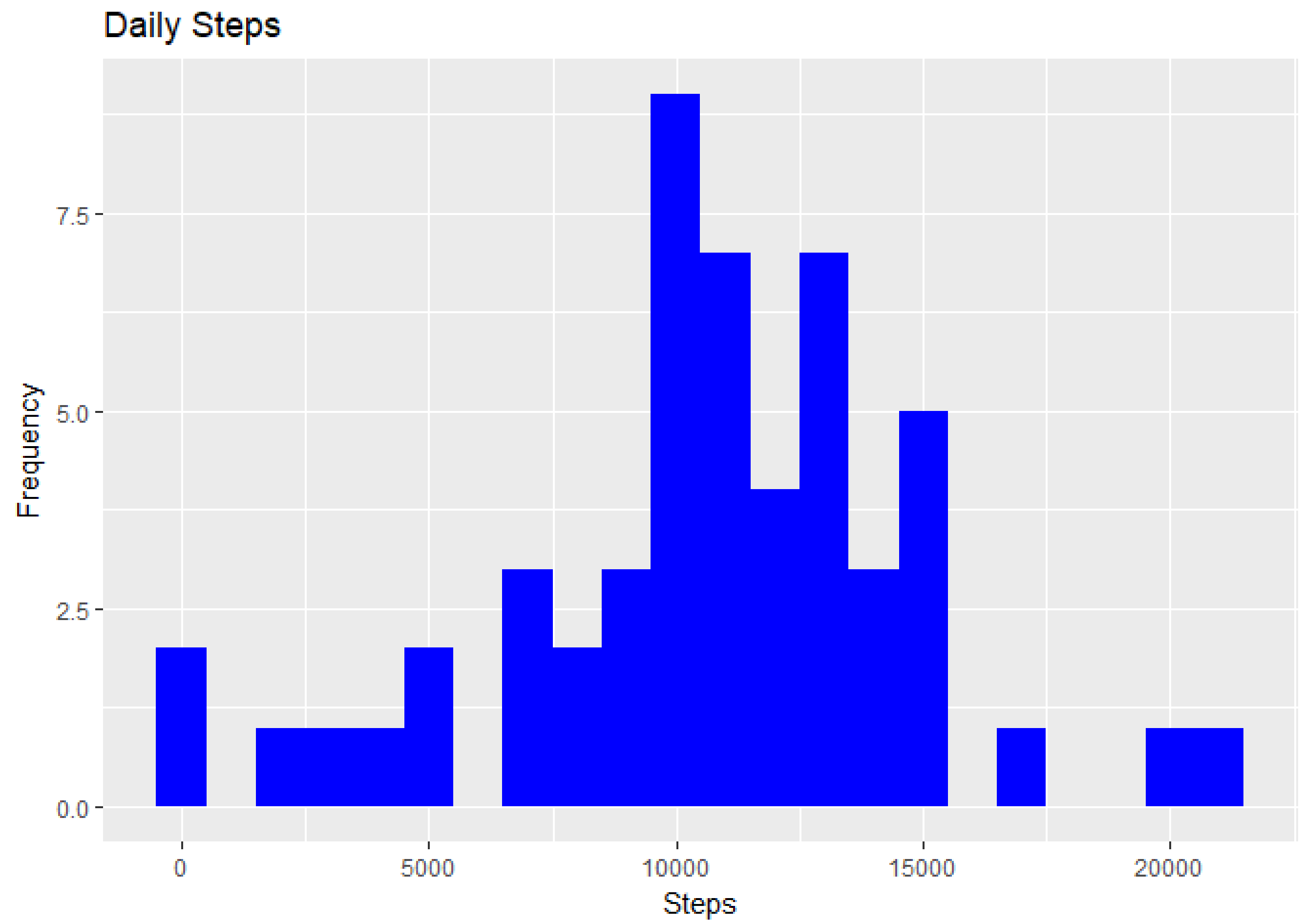
What is mean total number of steps taken per day?

```
#  Calculate the total number of steps taken per day
Total_Steps <- activityDT[, c(lapply(.SD, sum, na.rm = FALSE)), .SDcols = c("steps"), by = .
(date)]

head(Total_Steps, 10)
```

```
##           date steps
##  1: 2012-10-01    NA
##  2: 2012-10-02   126
##  3: 2012-10-03 11352
##  4: 2012-10-04 12116
##  5: 2012-10-05 13294
##  6: 2012-10-06 15420
##  7: 2012-10-07 11015
##  8: 2012-10-08    NA
##  9: 2012-10-09 12811
## 10: 2012-10-10  9900
```

```
ggplot(Total_Steps, aes(x = steps)) +
  geom_histogram(fill = "blue", binwidth = 1000) +
  labs(title = "Daily Steps", x = "Steps", y = "Frequency")
```



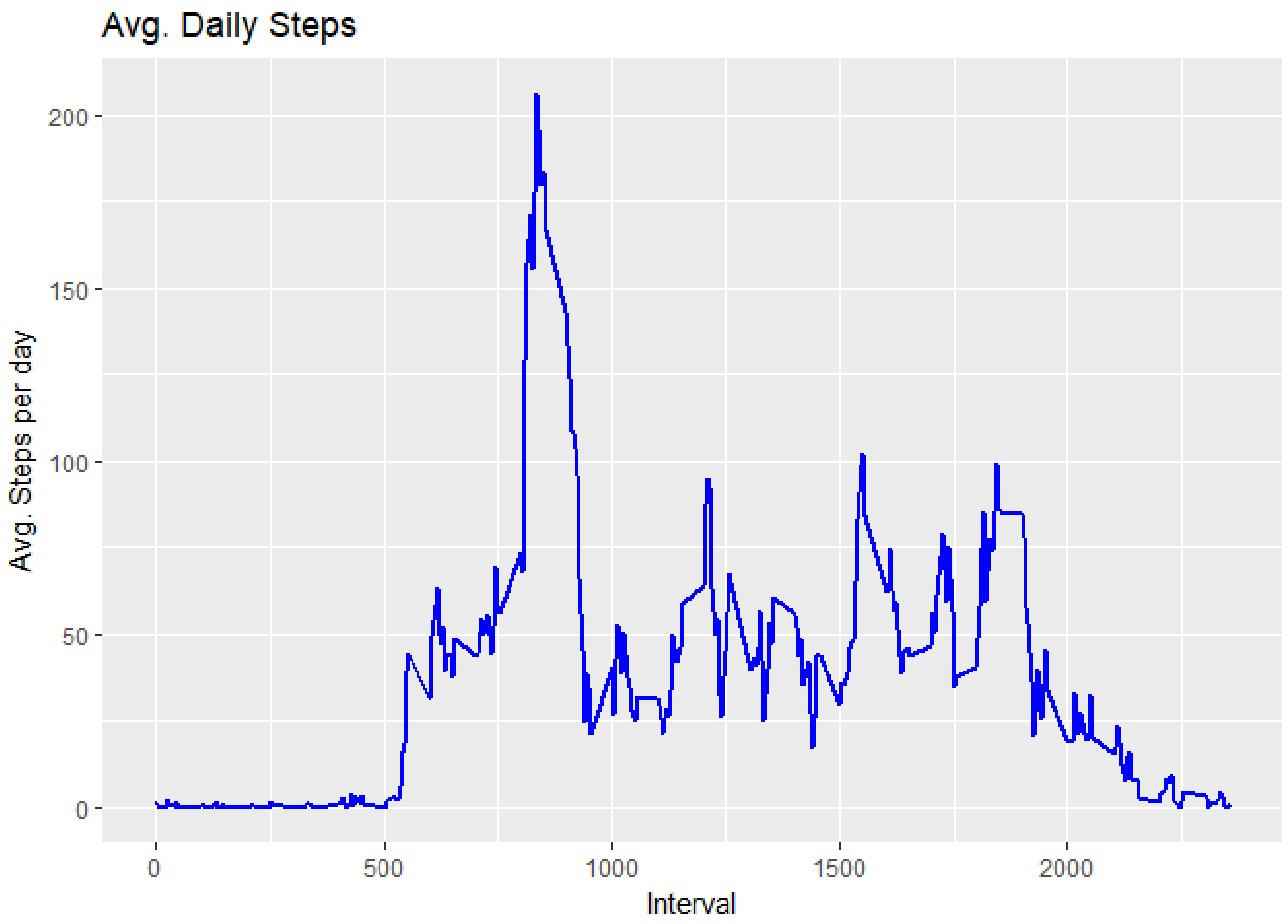
```
#Calculate and report the mean and median of the total number of steps taken per day
Total_Steps[, .(Mean_Steps = mean(steps, na.rm = TRUE), Median_Steps = median(steps, na.rm =
TRUE))]
```

```
##    Mean_Steps Median_Steps
## 1:   10766.19        10765
```

The mean total number number of steps taken per day was 10766.

What is the average daily activity pattern?

```
#Make a time series plot of the 5-minute interval (x-axis) and the average number of steps ta
ken, averaged across all days (y-axis)
IntervalDT <- activityDT[, c(lapply(.SD, mean, na.rm = TRUE)), .SDcols = c("steps"), by = .(i
nterval)]
ggplot(IntervalDT, aes(x = interval , y = steps)) + geom_line(color="blue", size=1) + labs(ti
tle = "Avg. Daily Steps", x = "Interval", y = "Avg. Steps per day")
```



```
#Which 5-minute interval, on average across all the days in the dataset, contains the maximum
 number of steps?
IntervalDT[steps == max(steps), .(max_interval = interval)]
```

```
##    max_interval
## 1:          835
```

The the average daily activity pattern was 835.

Imputing missing values

```
#Calculate and report the total number of missing values in the dataset
activityDT[is.na(steps), .N ]
```

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

```
# alternative solution
nrow(activityDT[is.na(steps),])
```

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

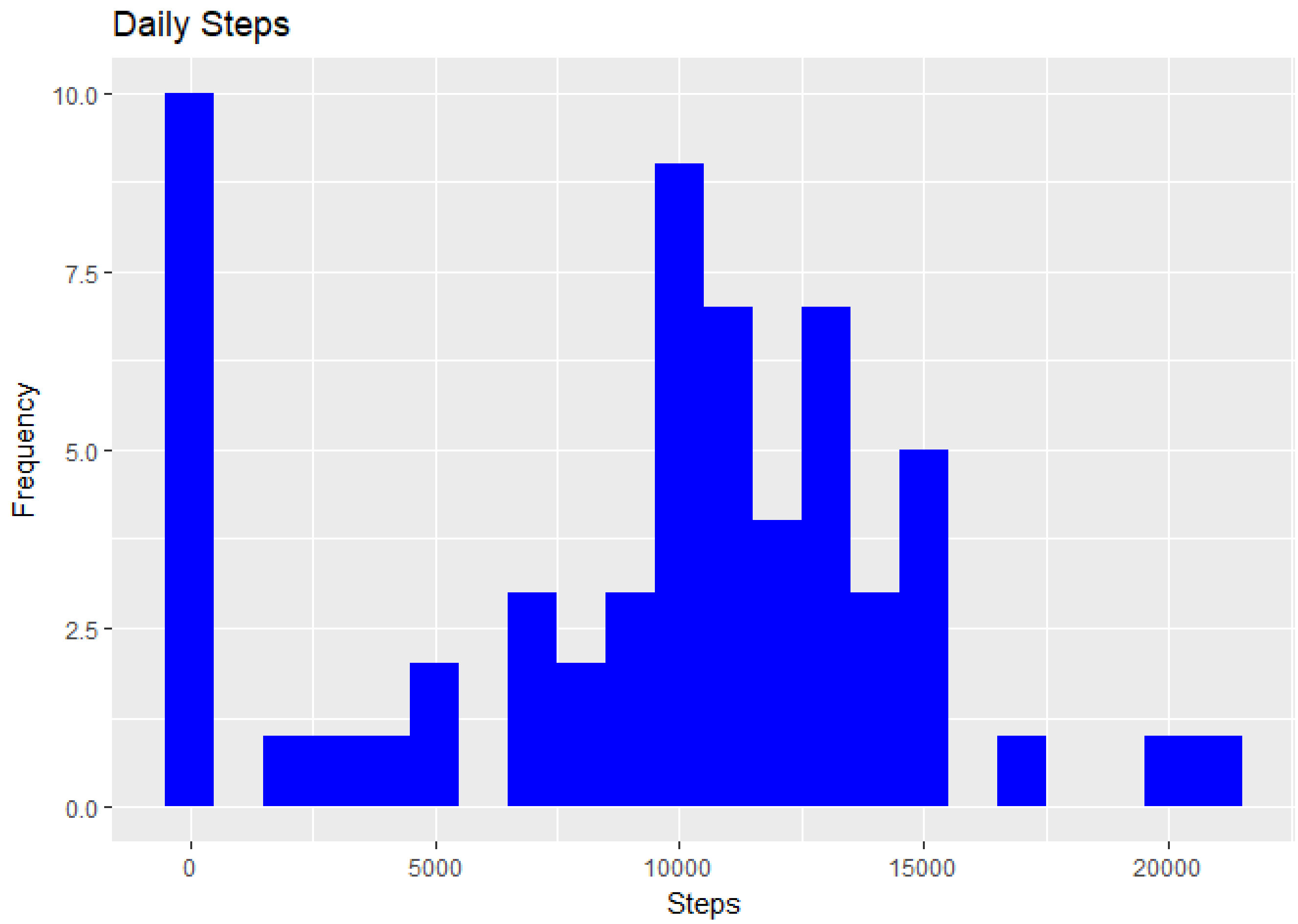
```
#Revise a strategy for filling in all of the missing values in the dataset. The strategy does
 not need to be sophisticated.
# For example, you could use the mean/median for that day, or the mean for that 5-minute inte
rval, etc.
# Filling in missing values with median of dataset.
activityDT[is.na(steps), "steps"] <- activityDT[, c(lapply(.SD, median, na.rm = TRUE)), .SDco
ls = c("steps")]

#Create a new dataset that is equal to the original dataset but with the missing data filled
 in.
data.table::fwrite(x = activityDT, file = "data/tidyData.csv", quote = FALSE)
#Make a histogram of the total number of steps taken each day and calculate and report the me
an 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 es
timates of the total daily number of steps?
  # total number of steps taken per day
  Total_Steps <- activityDT[, c(lapply(.SD, sum)), .SDcols = c("steps"), by = .(date)]

# mean and median total number of steps taken per day
Total_Steps[, .(Mean_Steps = mean(steps), Median_Steps = median(steps))]
```

```
##    Mean_Steps Median_Steps
## 1:    9354.23        10395
```

```
ggplot(Total_Steps, aes(x = steps)) + geom_histogram(fill = "blue", binwidth = 1000) + labs(t
itle = "Daily Steps", x = "Steps", y = "Frequency")
```



#Type of Estimate Mean_Steps Median_Steps *First Part (with na) 10765 10765 Second Part (fillin in na with median) 9354.23 10395*

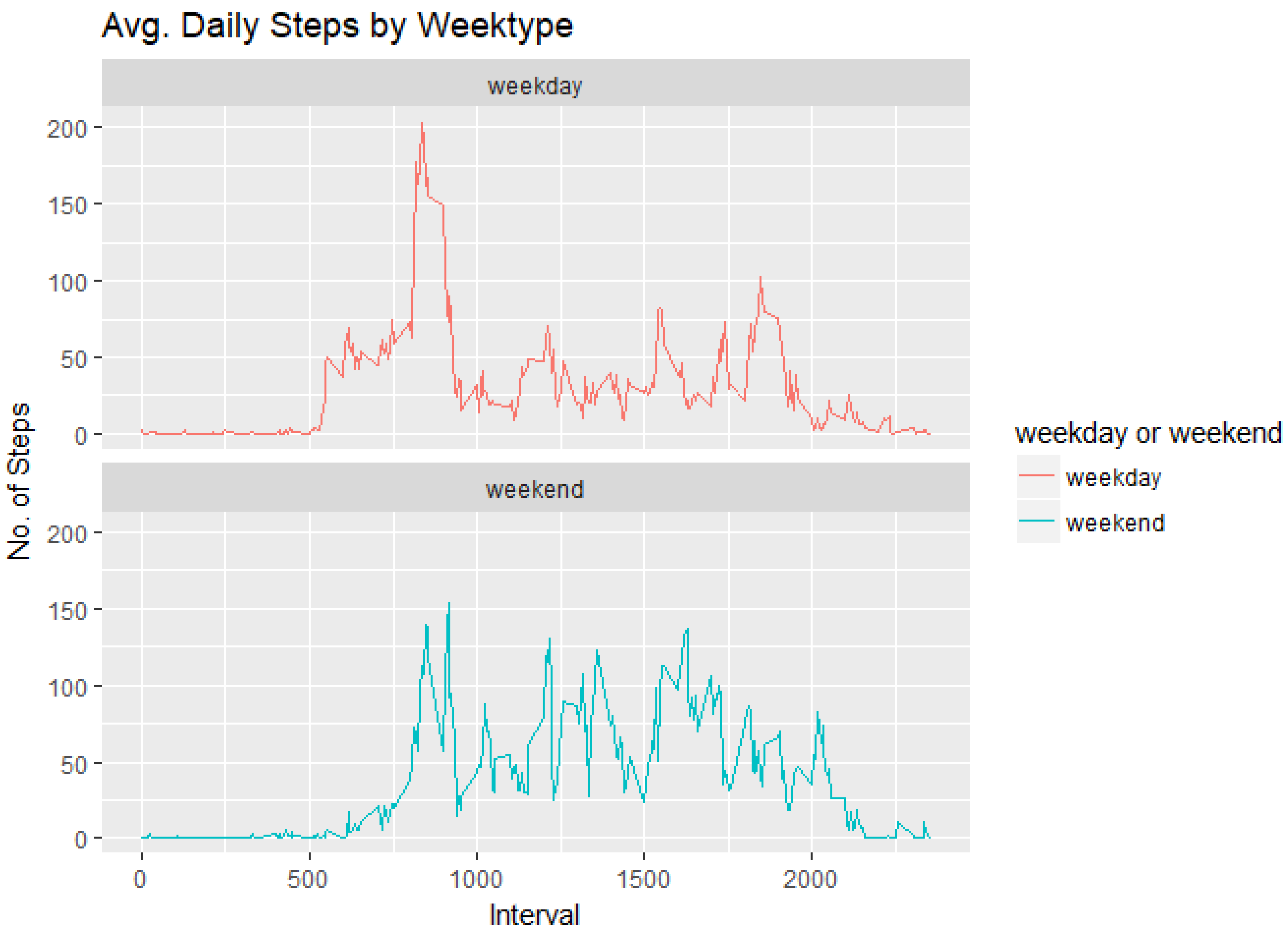
Are there differences in activity patterns between weekdays and weekends?

```
# Create a new factor variable in the dataset with two levels indicating whether a given date is a weekday or weekend day.
# Just recreating activityDT from scratch then making the new factor variable. (No need to, just want to be clear on what the entire process is.)
activityDT <- data.table::fread(input = "data/activity.csv")
activityDT[, date := as.POSIXct(date, format = "%Y-%m-%d")]
activityDT[, `Day of Week`:= weekdays(x = date)]
activityDT[grepl(pattern = "Monday|Tuesday|Wednesday|Thursday|Friday", x = `Day of Week`), "weekday or weekend"] <- "weekday"
activityDT[grepl(pattern = "Saturday|Sunday", x = `Day of Week`), "weekday or weekend"] <- "weekend"
activityDT[, `weekday or weekend` := as.factor(`weekday or weekend`)]
head(activityDT, 10)
```

```
##     steps       date interval Day of Week weekday or weekend
##  1:    NA 2012-10-01        0      Monday            weekday
##  2:    NA 2012-10-01        5      Monday            weekday
##  3:    NA 2012-10-01       10      Monday            weekday
##  4:    NA 2012-10-01       15      Monday            weekday
##  5:    NA 2012-10-01       20      Monday            weekday
##  6:    NA 2012-10-01       25      Monday            weekday
##  7:    NA 2012-10-01       30      Monday            weekday
##  8:    NA 2012-10-01       35      Monday            weekday
##  9:    NA 2012-10-01       40      Monday            weekday
## 10:    NA 2012-10-01       45      Monday            weekday
```

```
# 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). See the README file in the GitHub repository to see an example of what this
# plot should look like using simulated data.
activityDT[is.na(steps), "steps"] <- activityDT[, c(lapply(.SD, median, na.rm = TRUE)), .SDcols = c("steps")]
IntervalDT <- activityDT[, c(lapply(.SD, mean, na.rm = TRUE)), .SDcols = c("steps"), by = .(interval, `weekday or weekend`)]

ggplot(IntervalDT , aes(x = interval , y = steps, color=`weekday or weekend`)) + geom_line() + labs(title = "Avg. Daily Steps by Weektype", x = "Interval", y = "No. of Steps") + facet_wrap(~`weekday or weekend` , ncol = 1, nrow=2)
```



There are differences in activity patterns between weekdays and weekends.