Activity Data

Simran

25/11/2020

Loading and prepocessing Data

```
library(ggplot2)
activity <- read.csv("./activity.csv")
summary(activity)
```

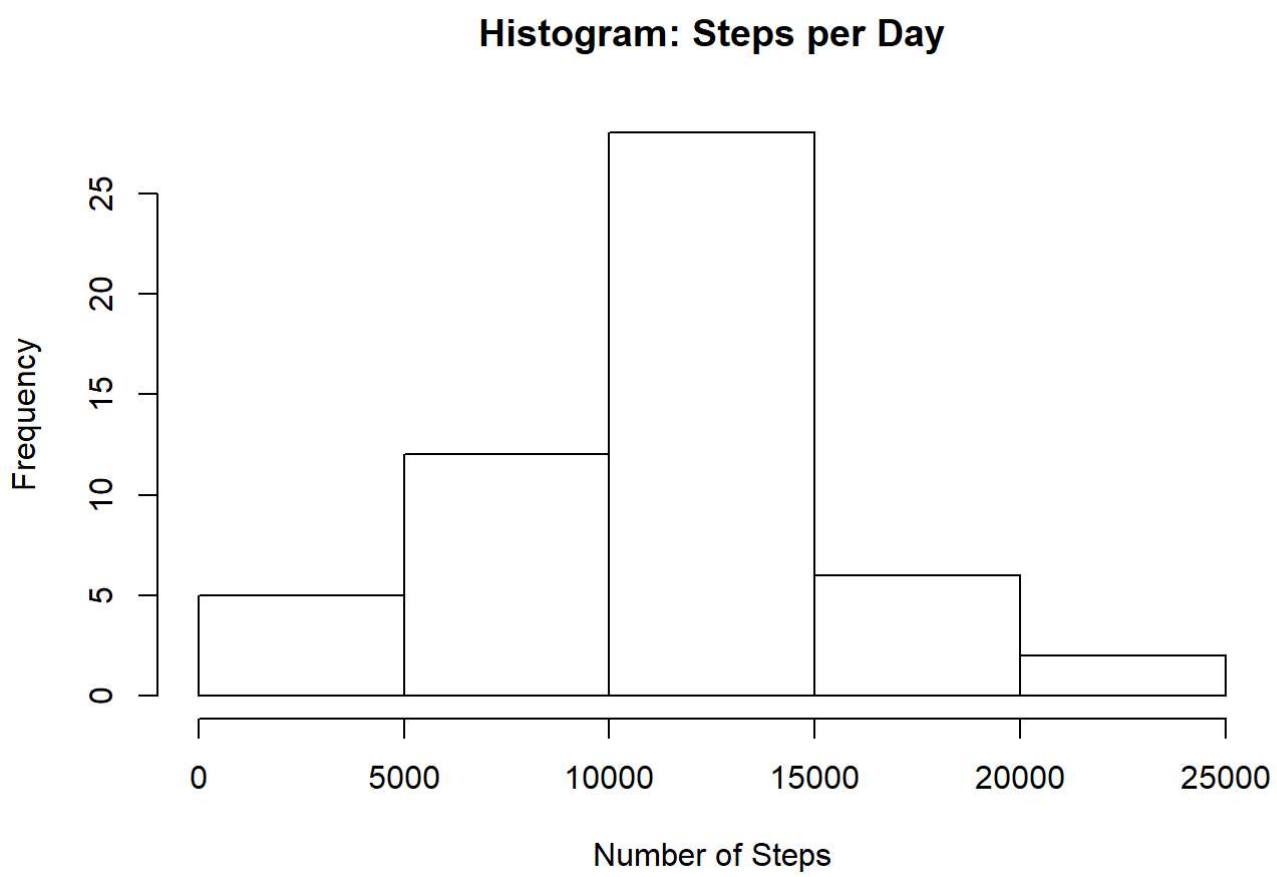
```
##      steps               date          interval     
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0  
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8  
##  Median :  0.00   2012-10-03:  288   Median :1177.5  
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5  
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2  
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0  
##  NA's   :2304     (Other)   :15840
```

What is mean total number of steps taken per day? Calculate the total number of steps taken per day

```
StepsPerDay <- tapply(activity$steps, activity$date, sum)
```

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

```
hist(StepsPerDay, xlab = "Number of Steps", main = "Histogram: Steps per Day")
```



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

```
MeanPerDay <- mean(StepsPerDay, na.rm = TRUE)
MeanPerDay
```

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

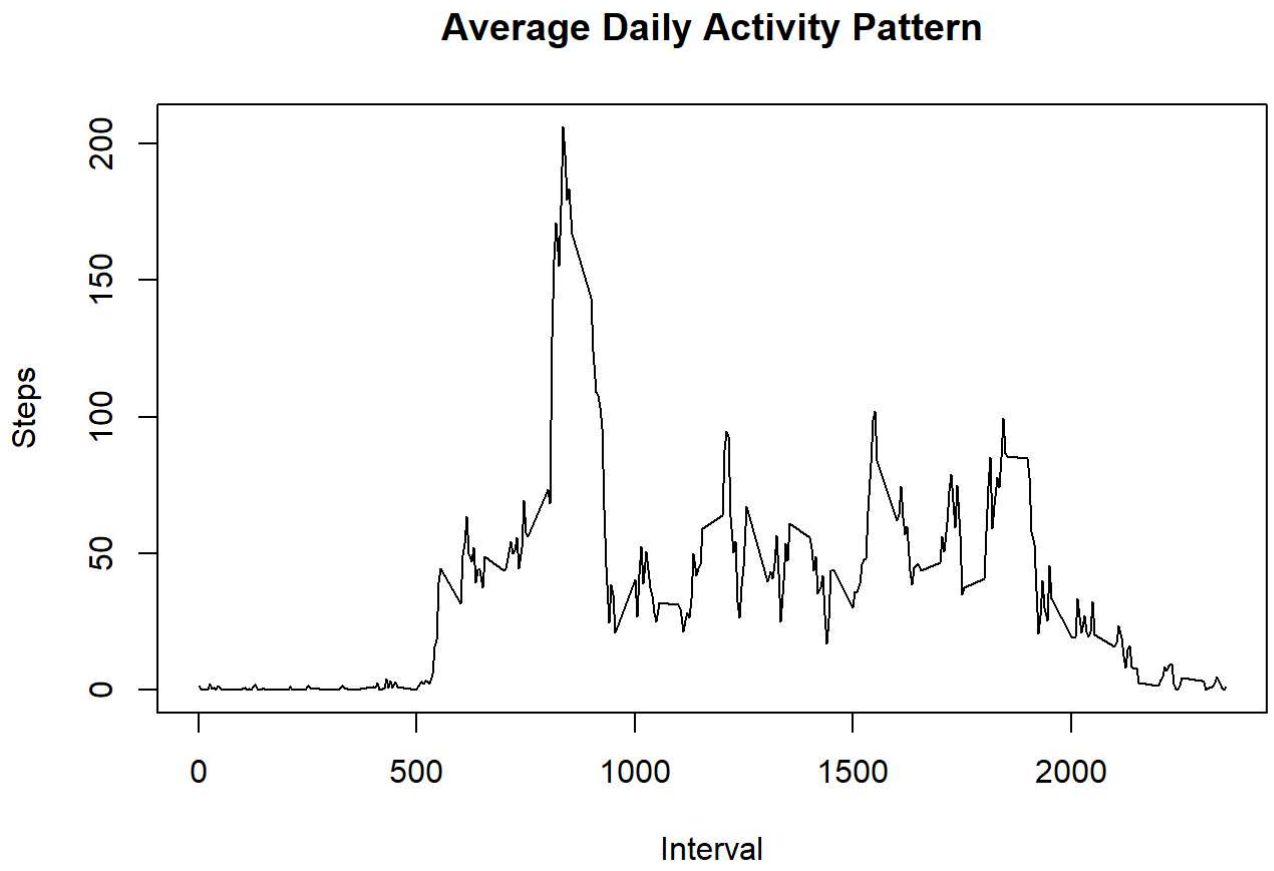
Median Per day

```
MedianPerDay <- median(StepsPerDay, na.rm = TRUE)
MedianPerDay
```

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

Make a time series plot of the 5-minute interval and the average number of steps taken, averaged across all days

```
StepsPerInterval <- tapply(activity$steps, activity$interval, mean, na.rm = TRUE)
plot(as.numeric(names(StepsPerInterval)),
     StepsPerInterval,
     xlab = "Interval",
     ylab = "Steps",
     main = "Average Daily Activity Pattern",
     type = "l")
```



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

```
maxInterval <- names(sort(StepsPerInterval, decreasing = TRUE)[1])
maxInterval
```

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

```
maxSteps <- sort(StepsPerInterval, decreasing = TRUE)[1]
maxSteps
```

```
##      835
## 206.1698
```

Calculate and report the total number of missing values in the dataset

```
NA.vals <- sum(is.na(activity$steps))
NA.vals
```

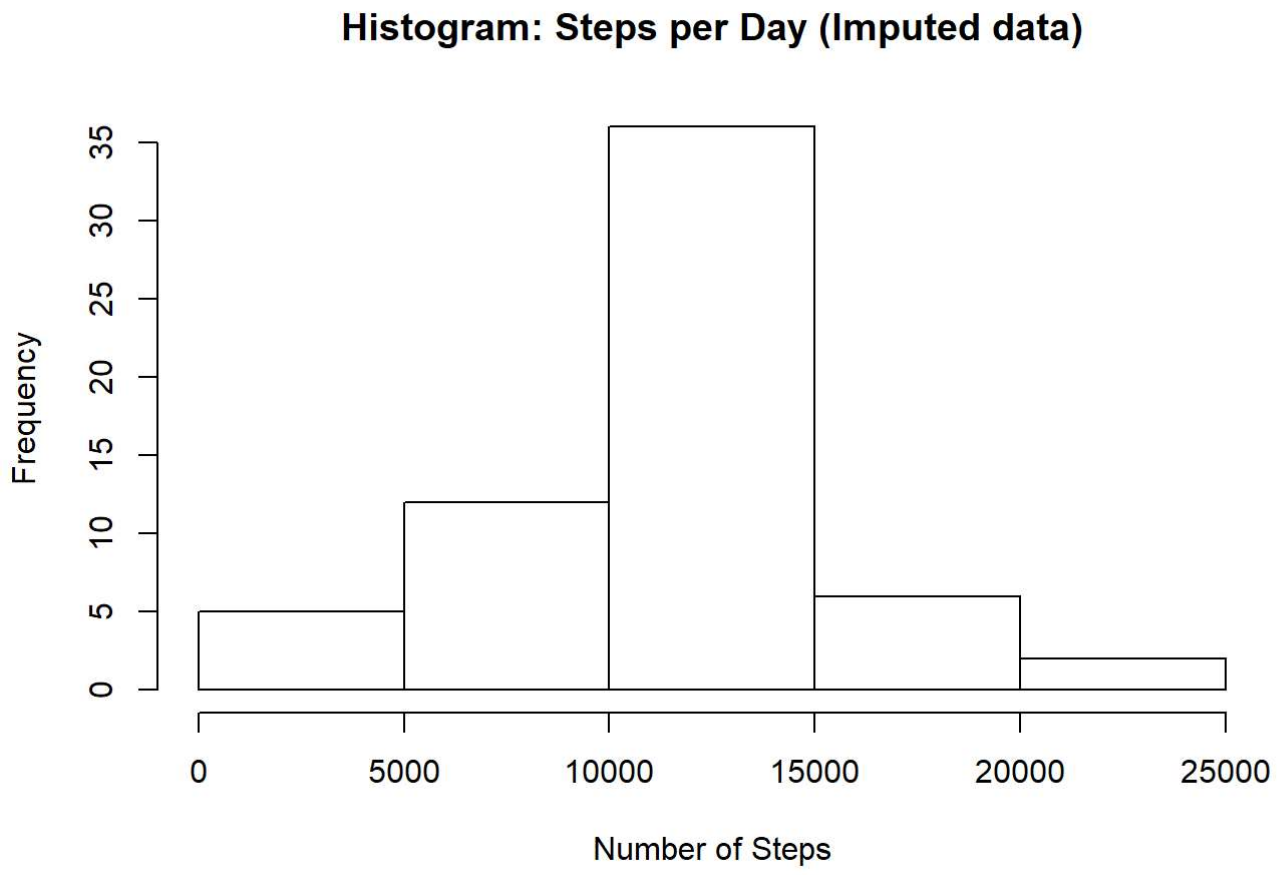
```
## [1] 2304
```

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

```
StepsPerInterval <- tapply(activity$steps, activity$interval, mean, na.rm = TRUE)
# split activity data by interval
activity.split <- split(activity, activity$interval)
# fill in missing data for each interval
for(i in 1:length(activity.split)){
    activity.split[[i]]$steps[is.na(activity.split[[i]]$steps)] <- StepsPerInterval[i]
}
activity.imputed <- do.call("rbind", activity.split)
activity.imputed <- activity.imputed[order(activity.imputed$date) ,]
```

Make a histogram of the total number of steps taken each day and

```
StepsPerDay.imputed <- tapply(activity.imputed$steps, activity.imputed$date, sum)
hist(StepsPerDay.imputed, xlab = "Number of Steps", main = "Histogram: Steps per Day (Imputed da
ta)")
```



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

```
MeanPerDay.imputed <- mean(StepsPerDay.imputed, na.rm = TRUE)
MeanPerDay.imputed
```

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

```
MedianPerDay.imputed <- median(StepsPerDay.imputed, na.rm = TRUE)
MedianPerDay.imputed
```

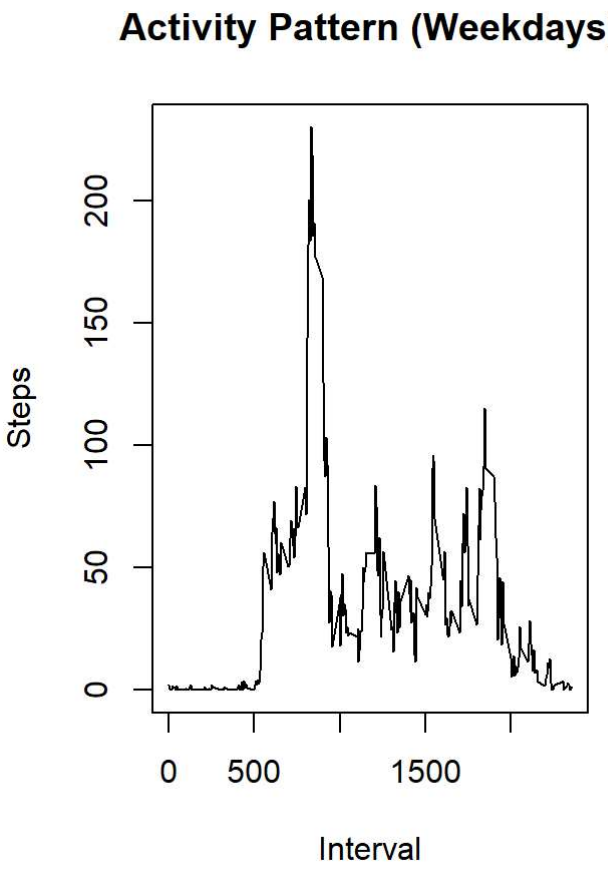
```
## [1] 10766.19
```

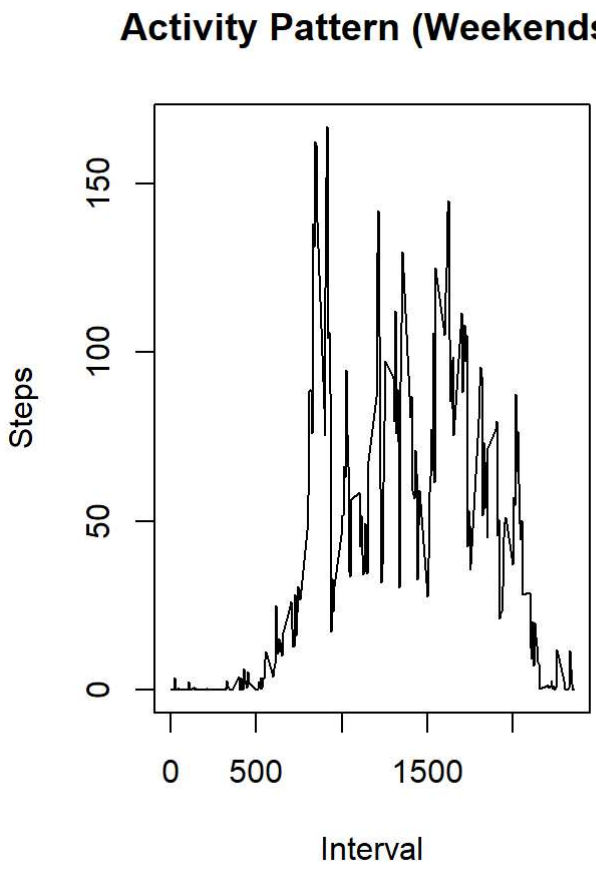
Create a new factor variable in the dataset with two levels “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
activity.imputed$day <- ifelse(weekdays(as.Date(activity.imputed$date)) == "Saturday" | weekdays
(as.Date(activity.imputed$date)) == "Sunday", "weekend", "weekday")
```

Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
# Calculating average steps per interval for weekends
StepsPerInterval.weekend <- tapply(activity.imputed[activity.imputed$day == "weekend" ,]$steps,
 activity.imputed[activity.imputed$day == "weekend" ,]$interval, mean, na.rm = TRUE)

# Calculating average steps per interval for weekdays
StepsPerInterval.weekday <- tapply(activity.imputed[activity.imputed$day == "weekday" ,]$steps,
 activity.imputed[activity.imputed$day == "weekday" ,]$interval, mean, na.rm = TRUE)

# Set a 2 panel plot
par(mfrow=c(1,2))

# Plot weekday activity
plot(as.numeric(names(StepsPerInterval.weekday)),
     StepsPerInterval.weekday,
     xlab = "Interval",
     ylab = "Steps",
     main = "Activity Pattern (Weekdays)",
     type = "l")

# Plot weekend activity
plot(as.numeric(names(StepsPerInterval.weekend)),
     StepsPerInterval.weekend,
     xlab = "Interval",
     ylab = "Steps",
     main = "Activity Pattern (Weekends)",
     type = "l")
```


Activity Pattern (Weekdays)
Steps
0
50
100
150
200
0
500
1500
Interval


Activity Pattern (Weekends)
Steps
0
50
100
150
0
500
1500
Interval