

RepData_PeerAssessment1

Loading the data

```
data <- read.csv2("activity.csv", header= TRUE, sep = ",")
summary(data)
```

```
##      steps      date      interval
## Min.   : 0.00   Length:17568   Min.    : 0.0
## 1st Qu.: 0.00   Class :character 1st Qu.: 588.8
## Median : 0.00   Mode  :character Median :1177.5
## Mean   : 37.38                Mean   :1177.5
## 3rd Qu.: 12.00                3rd Qu.:1766.2
## Max.   :806.00                Max.    :2355.0
## NA's   :2304
```

```
head(data)
```

```
##  steps      date interval
## 1    NA 2012-10-01      0
## 2    NA 2012-10-01      5
## 3    NA 2012-10-01     10
## 4    NA 2012-10-01     15
## 5    NA 2012-10-01     20
## 6    NA 2012-10-01     25
```

```
data2 = as.data.frame(data[complete.cases(data), ])
summary(data)
```

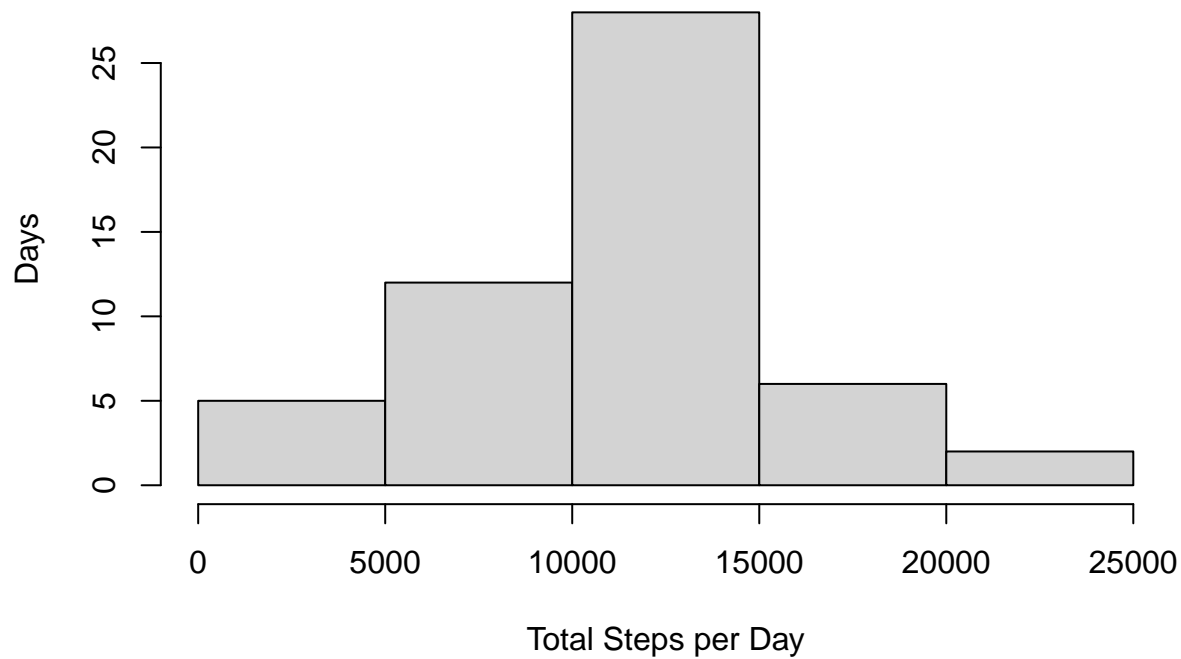
```
##      steps      date      interval
## Min.   : 0.00   Length:17568   Min.    : 0.0
## 1st Qu.: 0.00   Class :character 1st Qu.: 588.8
## Median : 0.00   Mode  :character Median :1177.5
## Mean   : 37.38                Mean   :1177.5
## 3rd Qu.: 12.00                3rd Qu.:1766.2
## Max.   :806.00                Max.    :2355.0
## NA's   :2304
```

Histogram of the total number of steps taken each day

```
steps <- tapply(data2$steps, data2$date, sum)
```

```
hist(steps, main = "Number of steps per Day", xlab = "Total Steps per Day", ylab = "Days")
```

Number of steps per Day



Mean and median number of steps taken each day

```
meansteps = mean(steps, na.rm = TRUE)
meansteps
```

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

```
mediansteps = median(steps, na.rm = TRUE)
mediansteps
```

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

Time series plot of the average number of steps taken

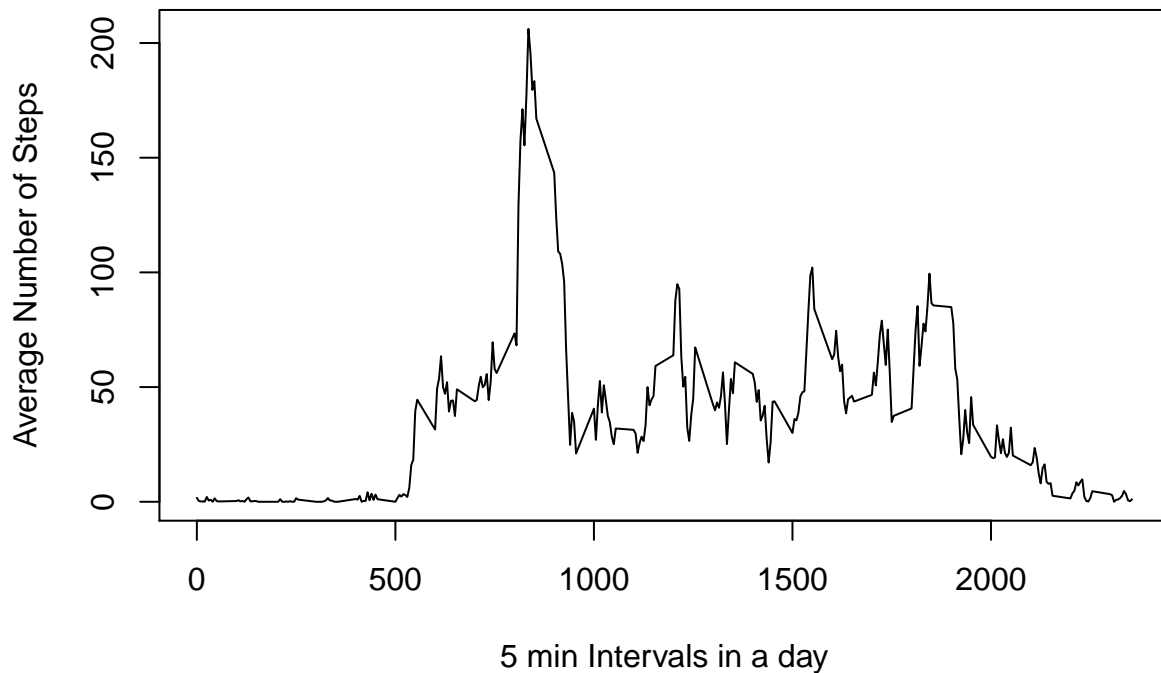
```
avgerage_number = aggregate(data2$steps, by = list(data2$interval), FUN = mean)
summary(avgerage_number)
```

```
##      Group.1      x
## Min.   : 0.0    Min.   : 0.000
## 1st Qu.: 588.8  1st Qu.:  2.486
## Median :1177.5  Median : 34.113
## Mean   :1177.5  Mean   : 37.383
## 3rd Qu.:1766.2  3rd Qu.: 52.835
## Max.   :2355.0  Max.   :206.170
```

```
plot(avgerage_number[, 1], avgerage_number[, 2], type = "l",
     xlab = "5 min Intervals in a day",
     ylab = "Average Number of Steps",
```

```
main = "The Average Daily Activity Pattern")
```

The Average Daily Activity Pattern



The 5-minute interval that, on average, contains the maximum number of steps

```
maxsteps = avgerage_number[which.max(avgerage_number[, 2]), 1]
maxsteps
```

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

Code to describe and show a strategy for imputing missing data

```
sum(!complete.cases(data))
```

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

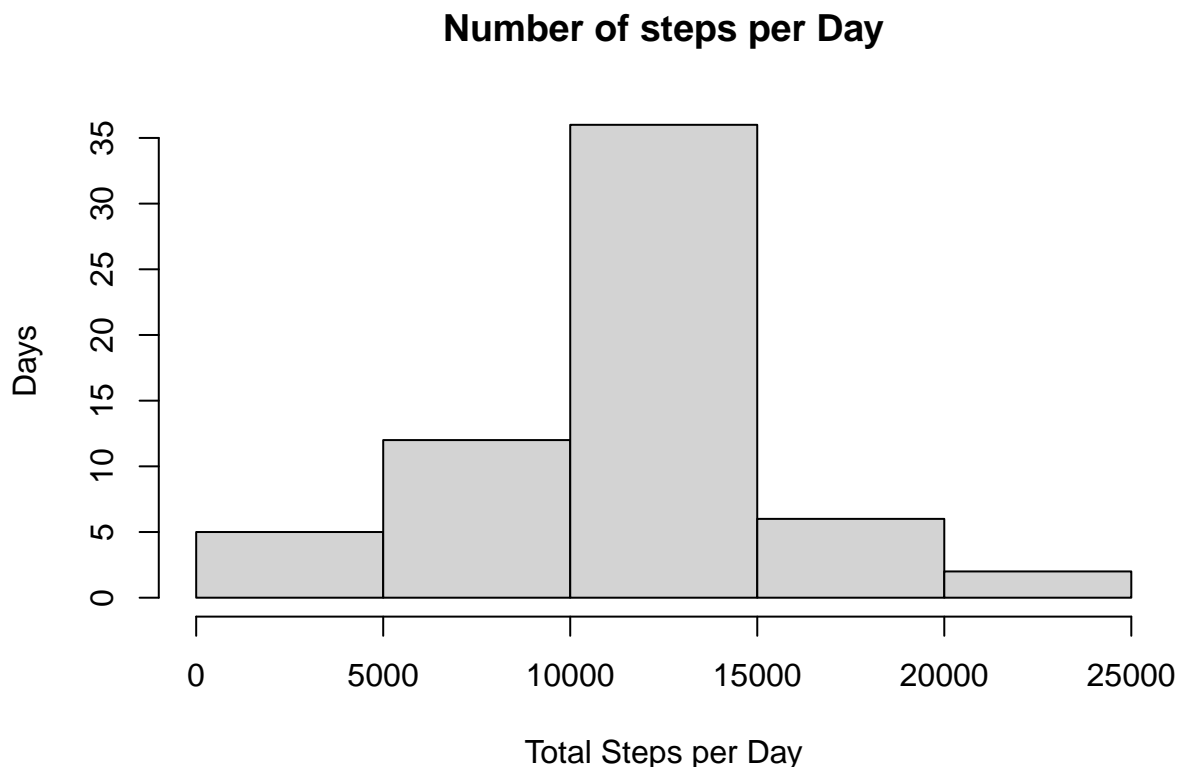
```
data3 = data
len1 = nrow(data3)
len2 = nrow(avgerage_number)
for (i in 1:len1) {
  if (is.na(data3$steps[i])) {
    for (j in 1:len2) {
      if (data3$interval[i] == avgerage_number[j, 1]) {
        data3$steps[i] = avgerage_number[j, 2]
      }
    }
  }
}
```

```
}
summary(data3)
```

```
##      steps      date      interval
## Min.   : 0.00   Length:17568   Min.    : 0.0
## 1st Qu.: 0.00   Class :character 1st Qu.: 588.8
## Median : 0.00   Mode  :character Median :1177.5
## Mean   : 37.38                      Mean   :1177.5
## 3rd Qu.: 27.00                      3rd Qu.:1766.2
## Max.   :806.00                      Max.    :2355.0
```

Histogram of the total number of steps taken each day after missing values are imputed

```
totalsteps2 = tapply(data3$steps, data3$date, sum)
hist(totalsteps2, main = "Number of steps per Day", xlab = "Total Steps per Day", ylab = "Days" )
```



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

```
data2$weekday = TRUE
weekday = weekdays(as.POSIXct(data2$date, format = "%Y-%m-%d" ))
for (i in 1:length(weekday)) {
  if (weekday[i] == "Saturday" | weekday[i] == "Sunday") {
    data2$weekday[i] = FALSE
  }
}
```

```

}
data_weekday = data2[which(data2$weekday == TRUE), ]
data_weekend = data2[which(data2$weekday == FALSE), ]

average_weekday = aggregate(data_weekday$steps,
                             by = list(data_weekday$interval),
                             FUN = mean)
names(average_weekday) = c("interval", "steps")
average_weekday$dayTag = "weekday"

average_weekend = aggregate(data_weekend$steps,
                             by = list(data_weekend$interval),
                             FUN = mean)
names(average_weekend) = c("interval", "steps")
average_weekend$dayTag = "weekend"

average_pattern = rbind(average_weekday, average_weekend)

xyplot(steps ~ interval | dayTag, data = average_pattern,
        type = "l", layout = c(1, 2))

```

