ASSIGNEMNT 1

Imporing the assignment data

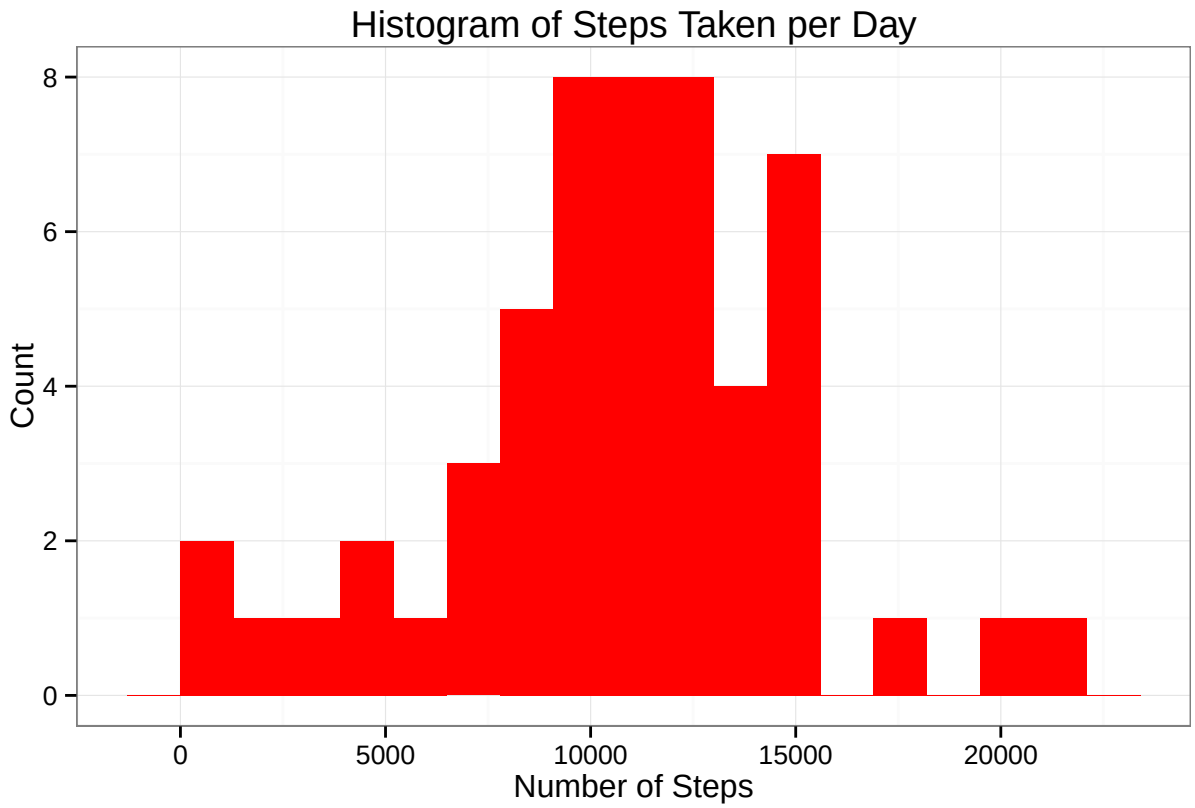
```
data<-read.csv("activity.csv",header=T)
```

Transform column date into date format

```
data$date<-as.Date(data$date)
```

Calculating the mean and median total number of steps taken per day Make a histogram of the total number of steps taken each day

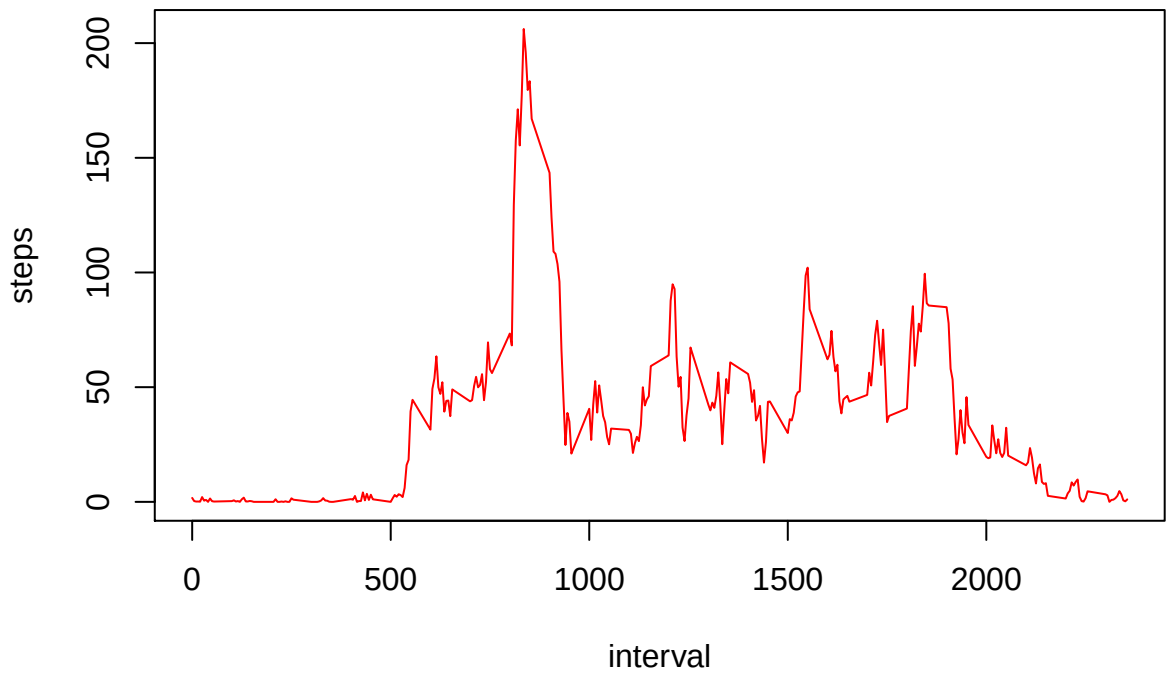
```
stepsPerDay <- aggregate(steps ~ date, data, sum)
colnames(stepsPerDay) <- c("date", "steps")
meanSteps <- round(mean(stepsPerDay$steps), 2)
medianSteps <- round(median(stepsPerDay$steps), 2)

library(ggplot2)
ggplot(stepsPerDay, aes(x = steps)) + geom_histogram(fill = "red",, binwidth = 1300) +
labs(title = "Histogram of Steps Taken per Day", x = "Number of Steps",
        y = "Count") + theme_bw() + theme(legend.position = "bottom")
```



Median of total total number of steps taken per day is 10765. Mean of total number of steps taken per day 10766.19.

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

```
stepsPer_5min_Interval <- aggregate(steps ~ interval, data,  mean, na.rm = TRUE)
colnames(stepsPer_5min_Interval) <- c("interval", "steps")
stepsPer_5min_Interval$interval <- as.integer(stepsPer_5min_Interval$interval)
plot(stepsPer_5min_Interval, type = "l", col = "red")
```




```
maxStepInterval <- stepsPer_5min_Interval[which.max(stepsPer_5min_Interval$steps), ]$interval
maxStepInterval
```

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

Interval 835 contains the maximum step number.

Calculating the number of NA’s

```
sum(is.na(data))
```

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

The total number of missing data points is 2304.

Impute missing data using mean

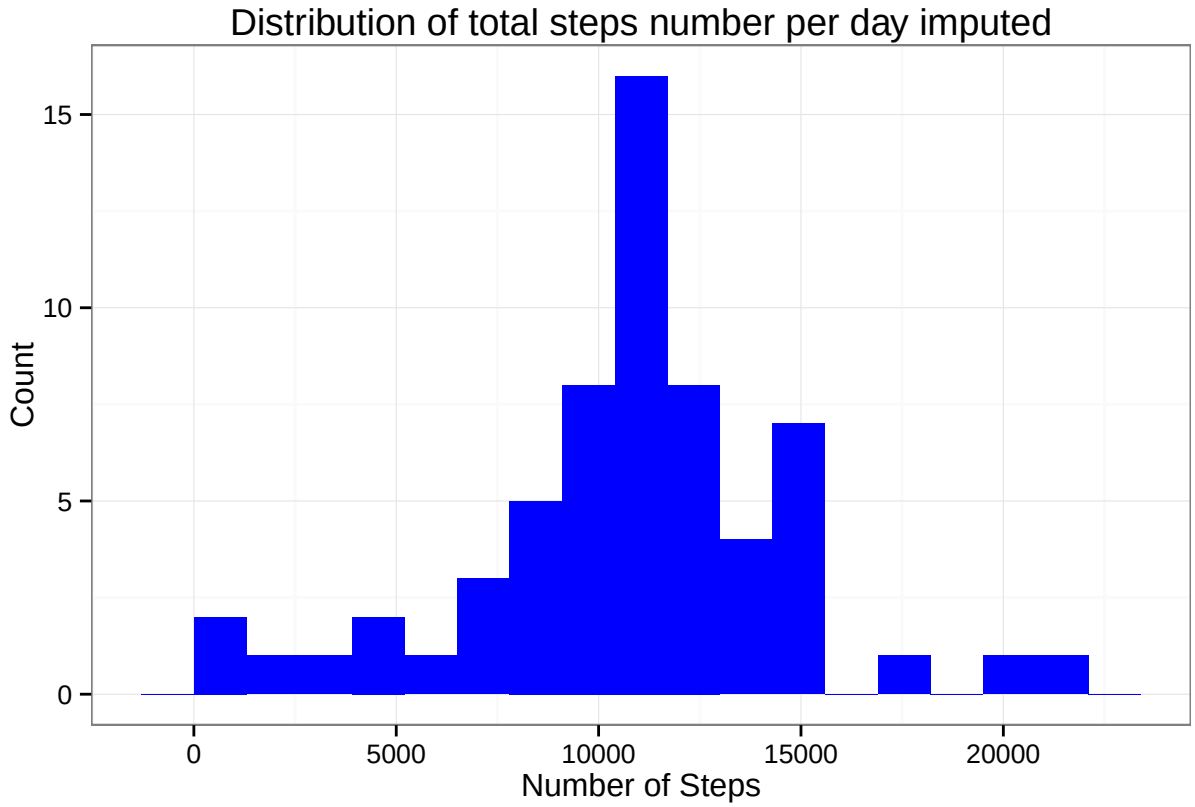
```
na_indices <- which(is.na(data$steps))
nMissing <- nrow(na_indices)
dataMeans <- stepsPer_5min_Interval
na_replacements <- unlist(lapply(na_indices, FUN = function(idx) {
    interval = data[idx, ]$interval
```

```
    dataMeans[dataMeans$interval == interval, ]$steps
}))
imp_steps <- data$steps
imp_steps[na_indices] <- na_replacements
imp_data <- data.frame(steps = imp_steps, date = data$date, interval = data$interval)
summary(imp_data)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 27.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
```

Make a histogram of the total number of steps taken each day for imputed data

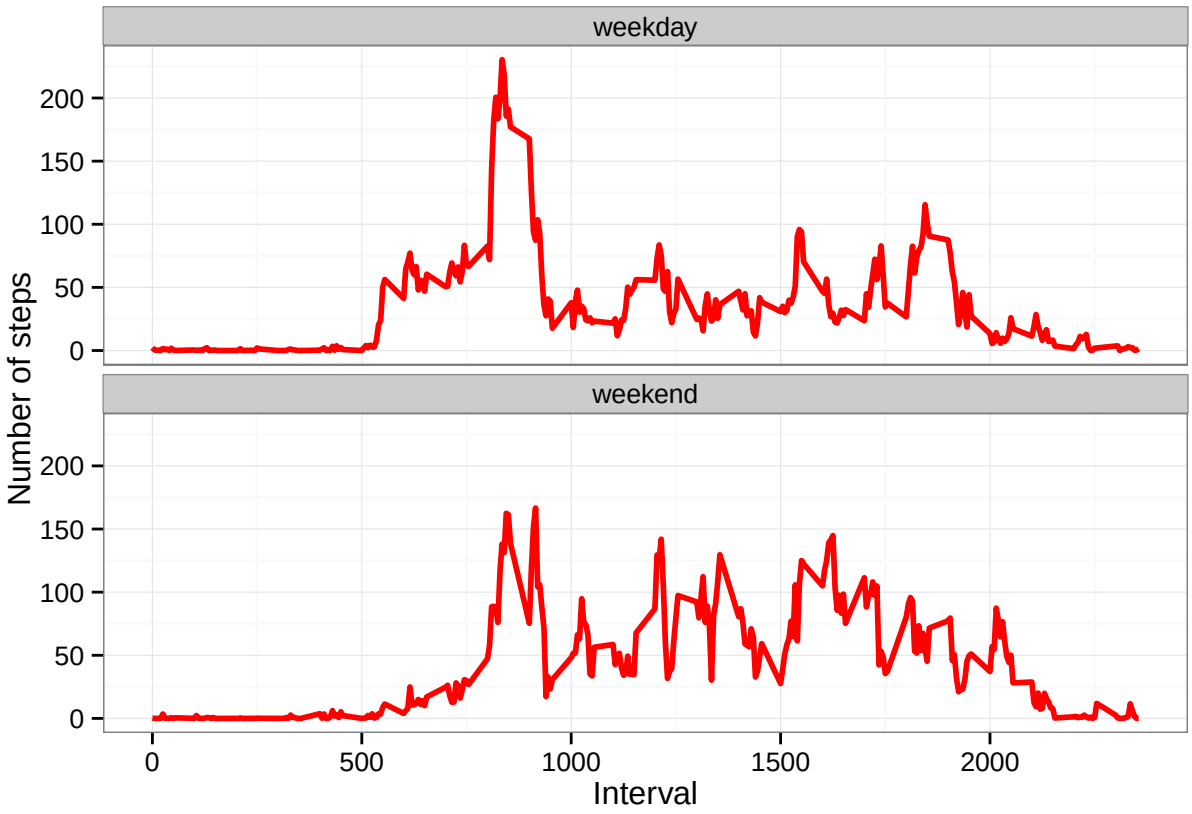
```
imp_stepsPerDay <- aggregate(steps ~ date, imp_data, sum)
colnames(imp_stepsPerDay) <- c("date", "steps")
meanSteps <- round(mean(imp_stepsPerDay$steps), 2)
medianSteps <- round(median(imp_stepsPerDay$steps), 2)

library(ggplot2)
ggplot(imp_stepsPerDay, aes(x = steps)) + geom_histogram(fill = "blue", binwidth = 1300) +
labs(title = "Distribution of total steps number per day imputed", x = "Number of Steps", y = "Count") +
```



Median of total total number of steps taken per day in imputed data is 10766.19. Mean of total number of steps taken per day in imputed data is 10766.19. Mean and meadian are the same, and they just slightly differnet from data with missing values.

Creating a new factor variable in the dataset with two levels weekday and weekend

```
imp_data$dayOfWeek <- (weekdays(as.Date(imp_data$date, "%Y-%m-%d")) %in% c("Saturday",
    "Sunday"))
for (i in 1:nrow(imp_data)) {
    if (imp_data$dayOfWeek[i]) {
        imp_data$dayOfWeek[i] <- "weekend"
    } else {
        imp_data$dayOfWeek[i] <- "weekday"
    }
}
imp_data$dayOfWeek <- factor(imp_data$dayOfWeek)

spi_dayOfWeek <- aggregate(steps ~ interval + dayOfWeek, imp_data, mean, na.rm = TRUE)

ggplot(spi_dayOfWeek, aes(x = interval, y = steps)) + geom_line(color = "red",
    size = 1) + facet_wrap(~dayOfWeek, nrow = 2, ncol = 1) + labs(x = "Interval",
    y = "Number of steps") + theme_bw()
```



Activities during the weekday show more intense activities during the morning hours, and activities are less during the other time of the day. During weekend movement is similar during the day.