Loading and preprocessing the data

```
#install.packages("RCurl")

#install.packages("RCurl")

library(RCurl)
```

```
## Warning: package 'RCurl' was built under R version 3.3.3
```

```
## Loading required package: bitops
```

```
## Warning: package 'bitops' was built under R version 3.3.2
```

```
#### Package Installation Completed. File extraction begins ###

URL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(URL, destfile = "./activity.zip")
unzip(zipfile="activity.zip")
activitydata <- read.csv("activity.csv")
summary(activitydata)
```

```
##      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
```

Mean of total number of steps taken per day

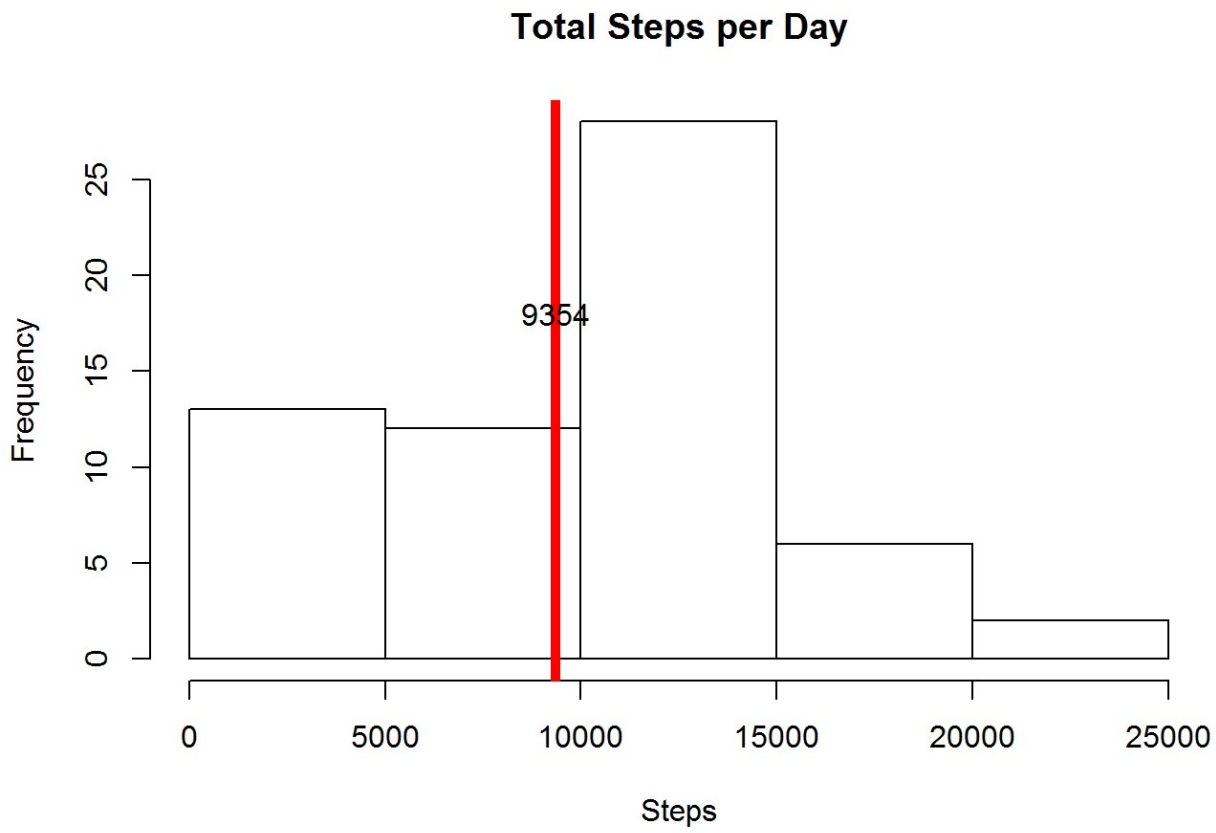
```
#install.packages("ggplot2")

library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
total.no.steps <- tapply(activitydata$steps, activitydata$date, FUN=sum, na.rm=
TRUE)
hist(total.no.steps, breaks=5, xlab="Steps", main = "Total Steps per Day")
mx <- round(mean(total.no.steps))
mx
```

```
## [1] 9354
```

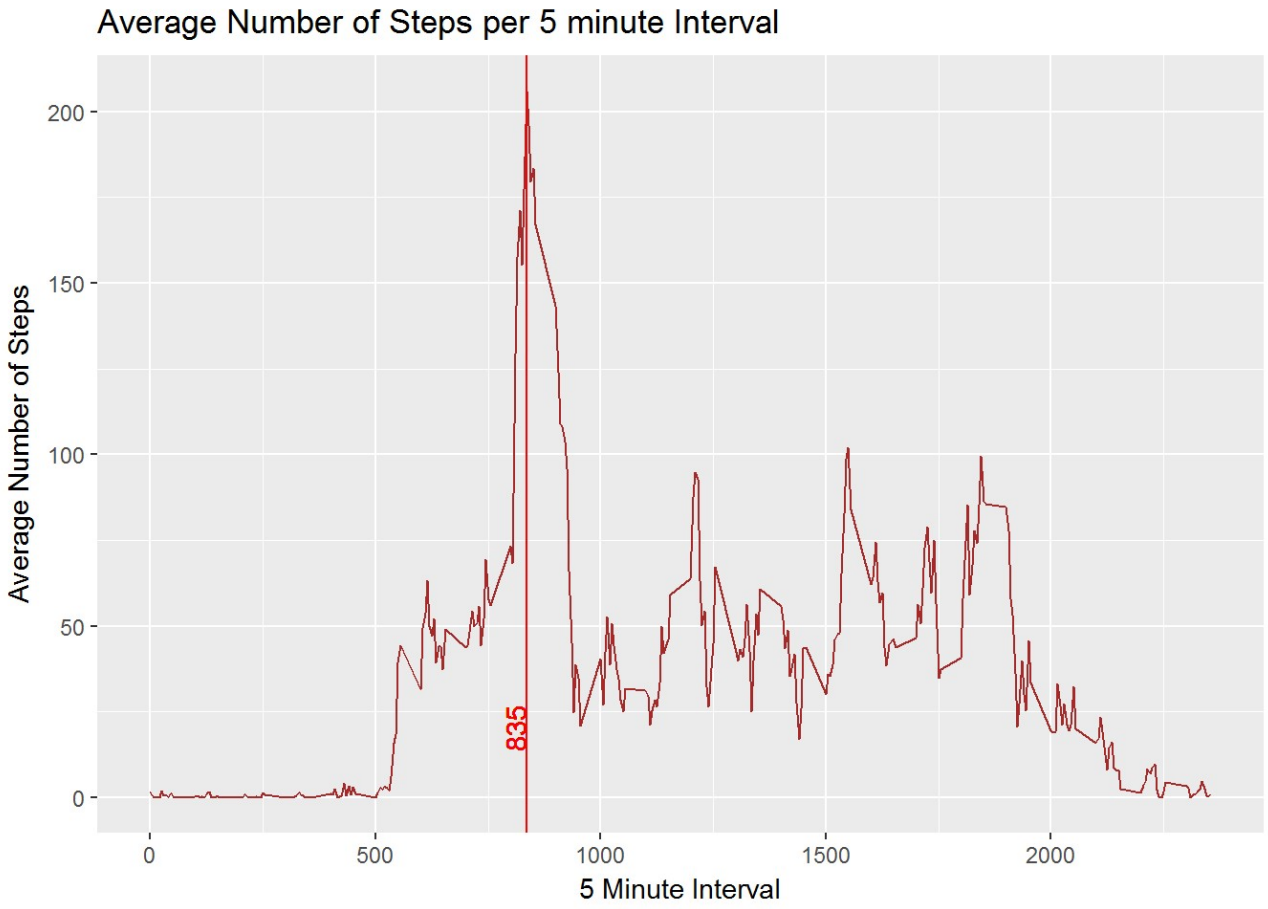
```
abline(v = mx, col = "red", lwd = 5)
text(mx, 18 , round(mx, 2))
```



average daily activity pattern

```
averagesteps <-aggregate(steps ~ interval ,data=activitydata,FUN=mean)
maxSteps <- max(averagesteps$steps)
ggplot(data=averagesteps, aes(x=interval, y=steps)) + geom_line(col="brown") +x
lab("5 Minute Interval")+ylab("Average Number of Steps")+ggtitle("Average Numbe
r of Steps per 5 minute Interval") + geom_vline(xintercept = averagesteps[avera
gesteps$steps==maxSteps,1],col="red") + geom_text(aes(x=averagesteps[averageste
ps$steps==maxSteps,1]-25, label=averagesteps[averagesteps$steps==maxSteps,1], y
=20), colour="red", angle=90, text=element_text(size=9))
```

```
## Warning: Ignoring unknown parameters: text
```

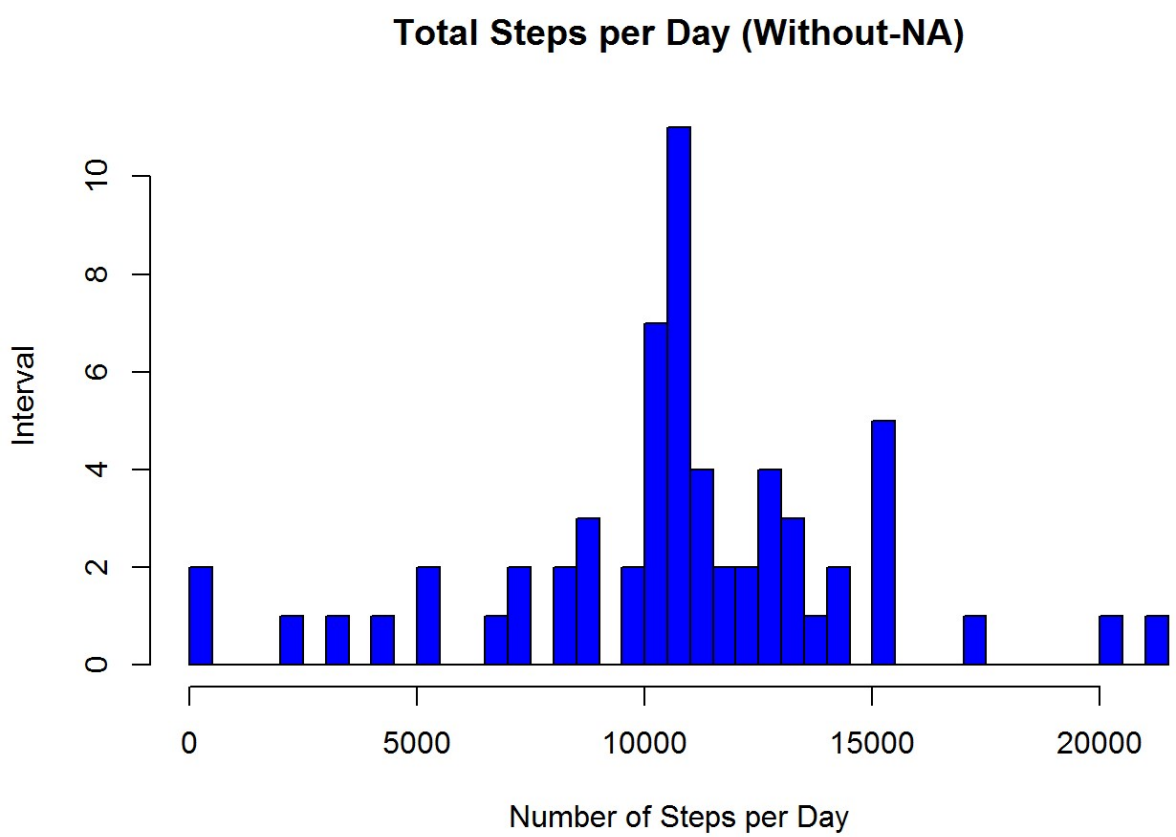


Imputing Null values

```
activity.na.count <- sum(is.na(activitydata$steps))

activity.na.count
```

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

```
activity.na<-activitydata

nas <- is.na(activity.na$steps)

avg_interval <- tapply(activity.na$steps,activity.na$interval, mean, na.rm=TRU
E, simplify = TRUE)

activity.na$steps[nas] <- avg_interval[as.character(activity.na$interval[nas])]

totalstepsperday.nas <- aggregate(steps ~ date, data = activity.na, FUN = sum,
na.rm = TRUE)


hist(totalstepsperday.nas$steps,main = "Total Steps per Day (Without-NA)", xla
b = "Number of Steps per Day",
    ylab = "Interval",
    col="blue",
    breaks=50)
```



```
summary(totalstepsperday.nas)
```

```
##          date        steps
##  2012-10-01: 1   Min.   :   41
##  2012-10-02: 1   1st Qu.: 9819
##  2012-10-03: 1   Median :10766
##  2012-10-04: 1   Mean   :10766
##  2012-10-05: 1   3rd Qu.:12811
##  2012-10-06: 1   Max.   :21194
##  (Other)   :55
```

WeekDay vs Weekend

```
## Create WeekDay & WeekendDay identifier
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.3.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lattice)


activity.na$day <- weekdays(as.Date(activity.na$date))
activity.na$DayIdentifier <- ifelse(activity.na$day %in% c("Saturday", "Sunda
y"), "Weekend", "Weekday")
weekdayplot <- ddply(activity.na, .(interval, DayIdentifier), summarize, Avg =
mean(steps))
xyplot(Avg~interval|DayIdentifier, data=weekdayplot,type="l",layout = c(1,2),
       main="Average Steps per Interval Based on Type of Day",
       ylab="Average Number of Steps", xlab="Interval")
```