Course5 Project1

Loading and preprocessing the data.

1. Load the data (i.e. read.csv())
2. Process/transform the data (if necessary) into a format suitable for your analysis

```
df <- read.csv("activity.csv")
df$date <- as.Date(df$date)
```

What is mean total number of steps taken per day?

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

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

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

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

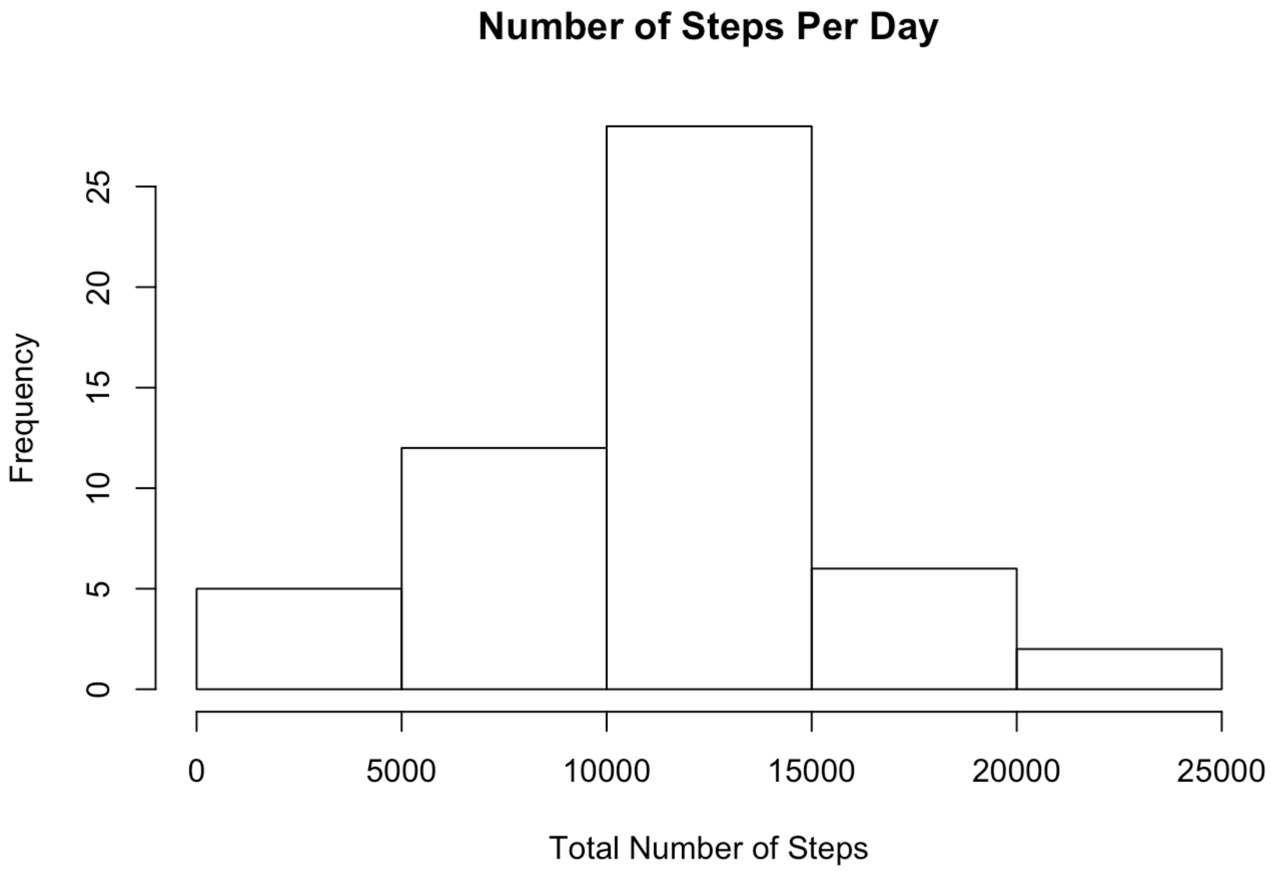
```
df2 <- na.omit(df)
head(df2)
```

```
##     steps       date interval
## 289     0 2012-10-02        0
## 290     0 2012-10-02        5
## 291     0 2012-10-02       10
## 292     0 2012-10-02       15
## 293     0 2012-10-02       20
## 294     0 2012-10-02       25
```

```
# Calculate the total number of steps taken per day
stepsPrDay <- aggregate(steps ~ date, df2, FUN = sum)

# If you do not understand the difference between a histogram and a barplot, research th
e difference between them. Make a histogram of the total number of steps taken each day

#png("Total_Number_of_Steps.png")
hist(stepsPrDay$steps, xlab = "Total Number of Steps", main = "Number of Steps Per Day")
```



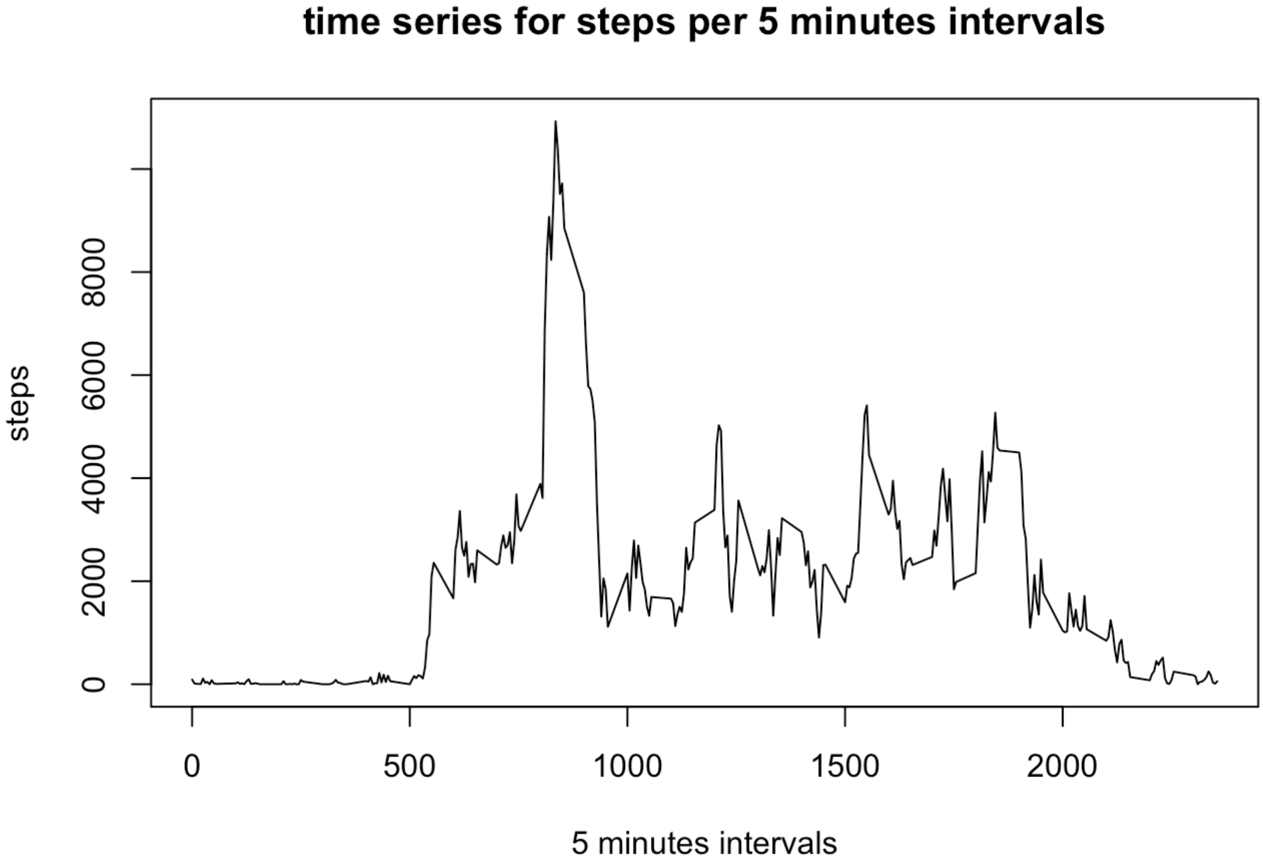
```
#dev.off()

#Calculate and report the mean and median of the total number of steps taken per day
meanSteps <- mean(stepsPrDay$steps)
medianSteps <- median(stepsPrDay$steps)
```

mean and median of the total steps per day are 1.076618910^{4} and 10765 respectly.

What is the average daily activity pattern?

```
#Make a time series plot (i.e. \color{red}{\verb|type = "l"|}type="l") of the 5-minute i
nterval (x-axis) and the average number of steps taken, averaged across all days (y-axi
s)
stepsPerIntervals <- aggregate(steps ~ interval, df2, FUN = sum)
#png("Time_Series_for_5Minutes_intervals.png")
plot(stepsPerIntervals$interval, stepsPerIntervals$steps, type = "l", xlab = "5 minutes
 intervals",
     ylab = "steps", main = "time series for steps per 5 minutes intervals")
```



```
#dev.off()

#Which 5-minute interval, on average across all the days in the dataset, contains the ma
ximum number of steps?
maxInterval <- stepsPerIntervals$interval[which(stepsPerIntervals$steps == max(stepsPerI
ntervals$steps))]
```

the maximum steps is happening in the 835 interval

Imputing missing values

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
table(is.na(df))
```

```
##
## FALSE  TRUE
## 50400  2304
```

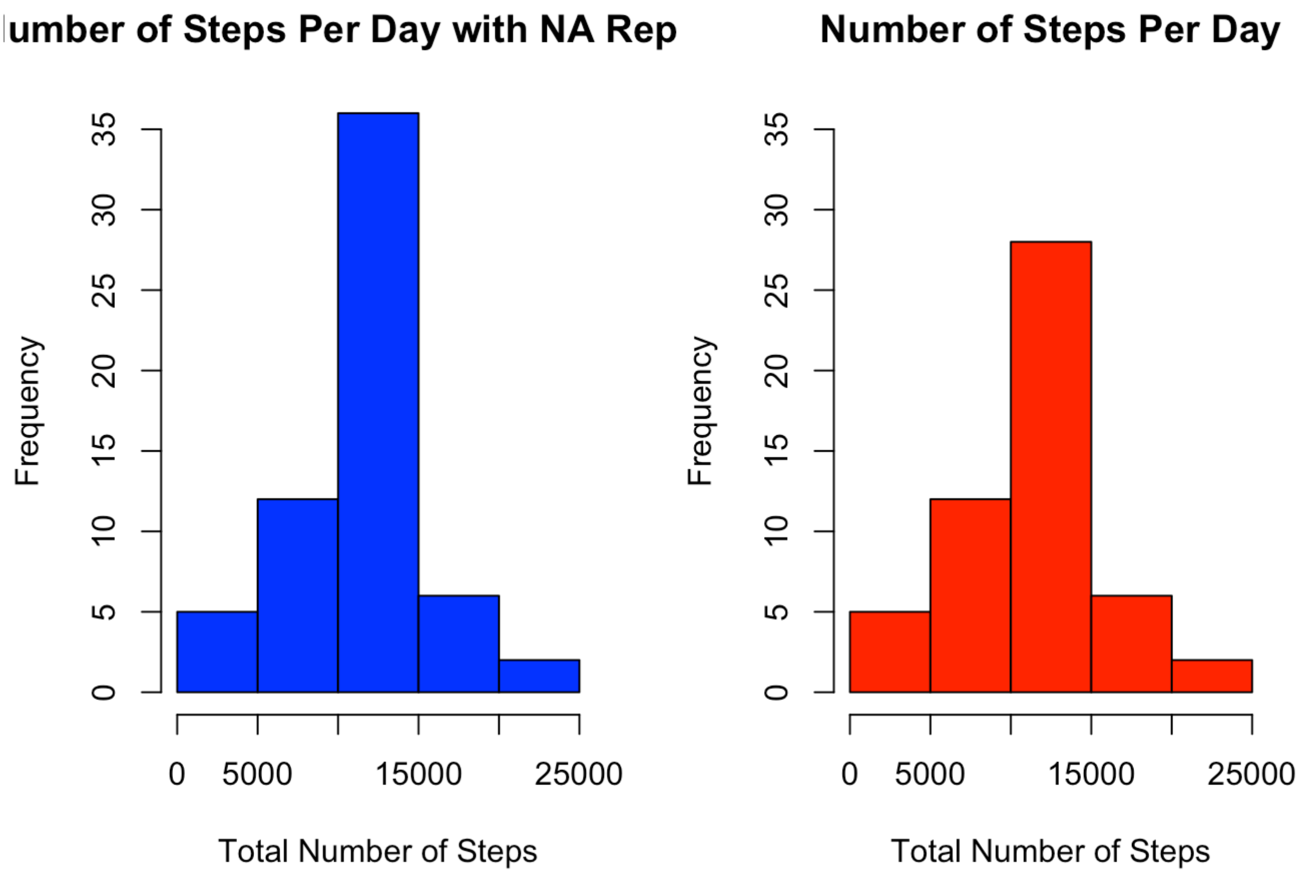
total number of missing data is 2304 in the data.

2. Devise 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 interval, etc.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
stepsMeanDayIntr <- aggregate(steps ~ interval, df2, FUN = mean)
mergedData <- merge(x = df, y = stepsMeanDayIntr, by = "interval")
mergedData$steps.x <- ifelse(is.na(mergedData$steps.x), mergedData$steps.y, mergedData$s
teps.x)
dfNew <- mergedData %>% select(steps.x, date, interval)
names(dfNew) <- c("steps", "date", "interval")
```

4. Make a histogram of the total number of steps taken each day and Calculate and report the mean 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 estimates of the total daily number of steps?

```
stepsPrDay_new <- aggregate(steps ~ date, dfNew, FUN = sum)
#png("Total_Number_of_Steps_with and without NA.png")
par(mfrow=c(1,2))
hist(stepsPrDay_new$steps, xlab = "Total Number of Steps", main = "Number of Steps Per D
ay with NA Replaced", col = "blue", ylim = c(0, 35))
hist(stepsPrDay$steps, xlab = "Total Number of Steps", main = "Number of Steps Per Day",
 col = "red", ylim = c(0, 35))
```



```
#dev.off()
```

now we calculate the difference of means and meadians:

```
meanSteps_new <- mean(stepsPrDay_new$steps)
medianSteps_new <- median(stepsPrDay_new$steps)

print(paste("Original mean is: ", meanSteps, " and the new mean is: ", meanSteps_new, "
 and the difference is: ", meanSteps - meanSteps_new))
```

```
## [1] "Original mean is:  10766.1886792453  and the new mean is:  10766.1886792453  and
the difference is:  0"
```

```
print(paste("Original median is: ", medianSteps, " and the new median is: ", medianSteps
_new, " and the difference is: ", -medianSteps + medianSteps_new))
```

```
## [1] "Original median is:  10765  and the new median is:  10766.1886792453  and the di
fference is:  1.1886792452824"
```

the means are the same but the medians have a difference of 1.88 unit.

Are there differences in activity patterns between weekdays and weekends?

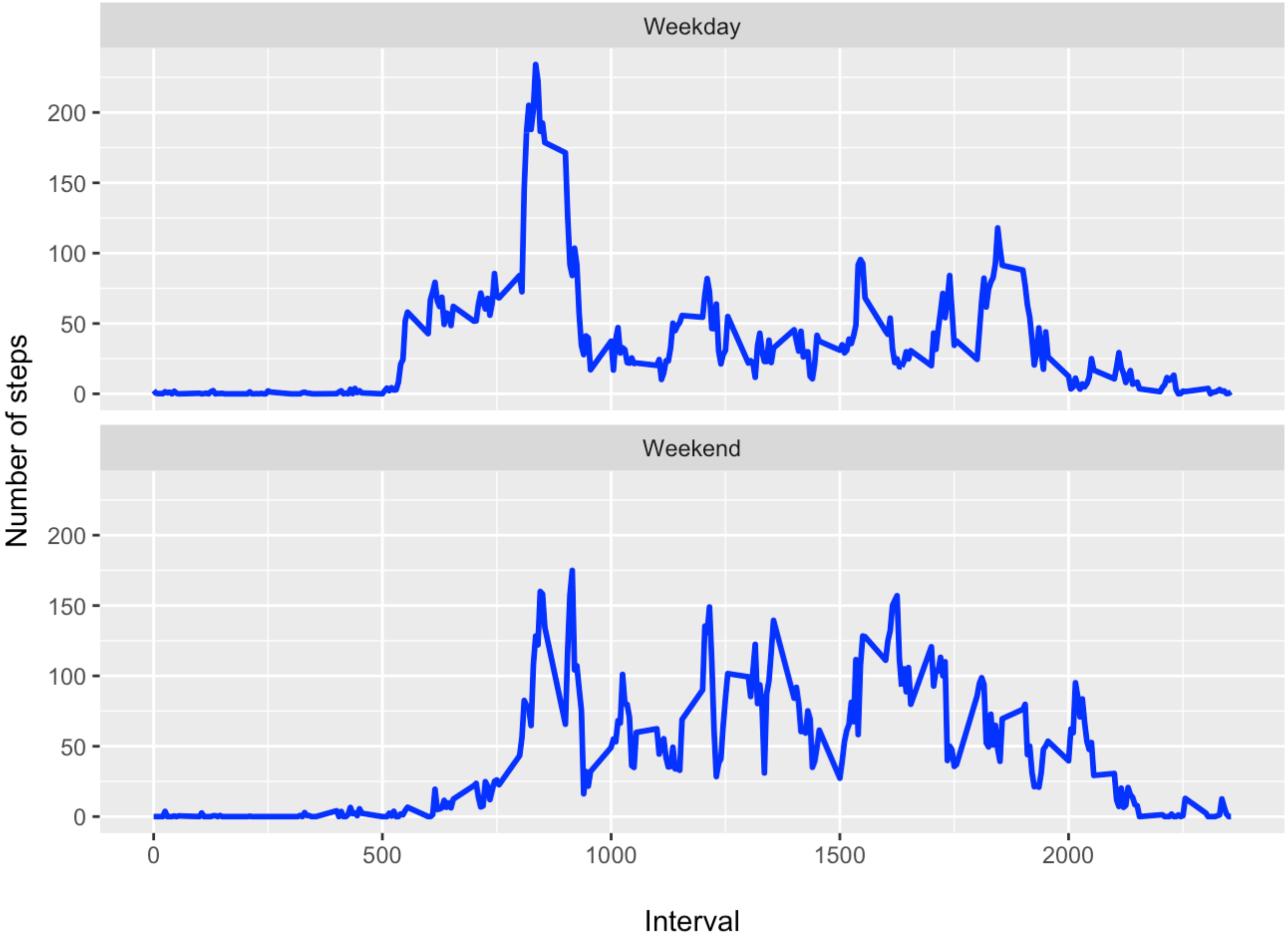
1. Are there differences in activity patterns between weekdays and weekends?

```
dfWdy <- df
dfWdy$WeekDay <- ifelse(weekdays(dfWdy$date, abbr = TRUE) %in% c("Sat", "Sun"), "Weeken
d", "Weekday")
dfWdy$WeekDay <- as.factor(dfWdy$WeekDay)
table(dfWdy$WeekDay)
```

```
##
## Weekday Weekend
##   12960    4608
```

2. Make a panel plot containing 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 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.

```
library(ggplot2)
meanAggregate <- aggregate(steps ~ interval + WeekDay, dfWdy, FUN =  mean)
#png("Weekday_and_Weekend_Steps.png")
ggplot(meanAggregate, aes(x=interval, y=steps)) +
  geom_line(color="blue", size=1) +
  facet_wrap(~WeekDay, nrow=2) +
  labs(x="\nInterval", y="\nNumber of steps")
```


Weekday
Weekend
Number of steps
200
150
100
50
0
200
150
100
50
0
0
500
1000
1500
2000
Interval

```
#dev.off()
```