Reproducible Research: Peer Assessment 1

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

```
packagesToLoad <-c("rstudioapi","stringr","knitr","lattice")
install.packages(setdiff(packagesToLoad, rownames(installed.packages())))

rm(list=ls())

library(rstudioapi)
library(stringr)
library(knitr)
library(lattice)

opts_chunk$set(echo=TRUE, results="hide")

setwd(paste0(head(str_split(getSourceEditorContext()$path,"/")[[1]],-1),collapse="/"))
getwd()
```

```
## [1] "E:/Proyectos/practices/RepData_PeerAssessment1"
```

```
downloadUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
fileZip <- "repdata_data_activity.zip"
file <- "activity.csv"
if (!file.exists(fileZip)) {
  print(paste("Download ZIP",downloadUrl))
  download.file(downloadUrl, fileZip, method = "curl")
}

if (!file.exists(file)) {
  unzip(fileZip)
  print(paste("UNZIP ZIP",fileZip))
}

DF <- read.csv(file)
DF$date <- as.Date(DF$date,"%Y-%m-%d")
str(DF)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

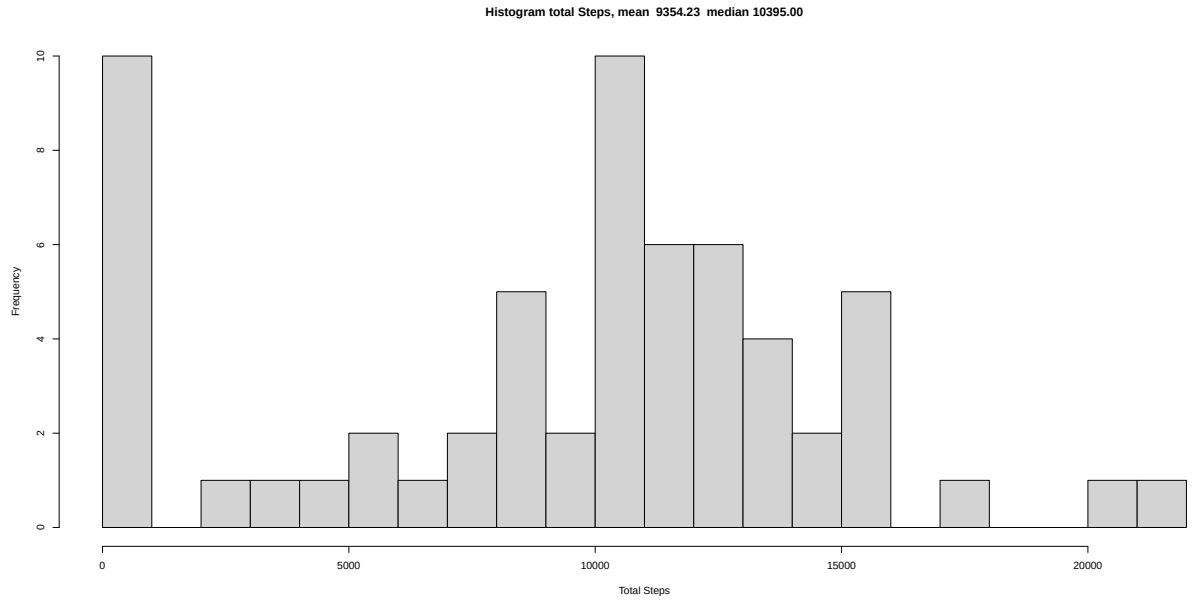
```
summary(DF)
```

```
##      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.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2  
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0  
##  NA's   :2304
```

What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day
2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day
3. Calculate and report the mean and median of the total number of steps taken per day

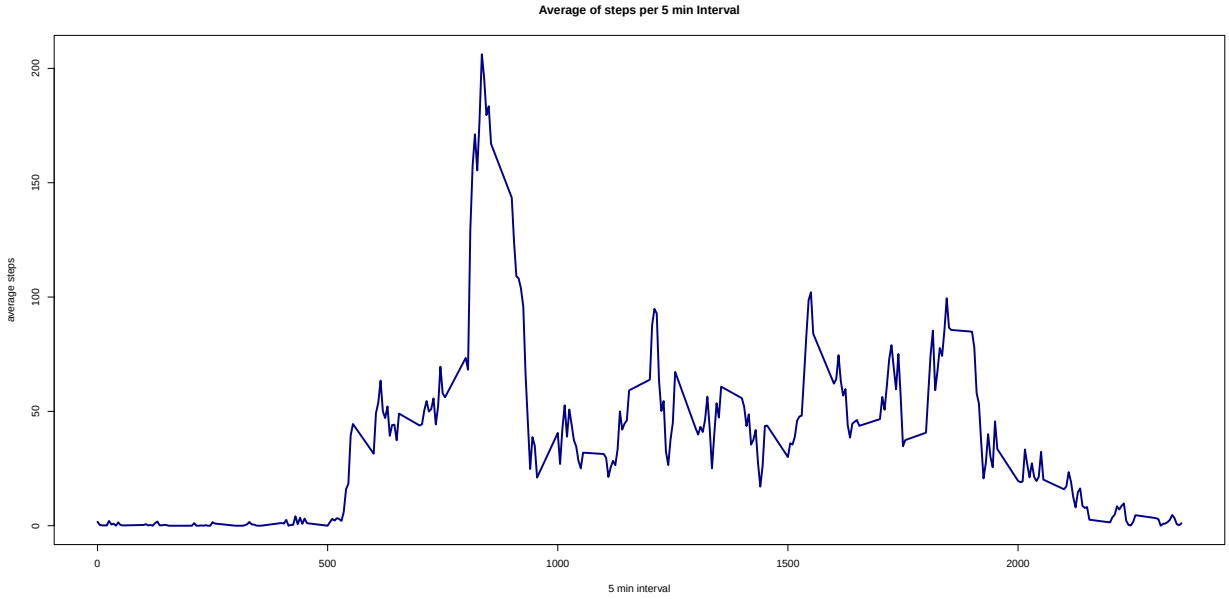
```
DFsum <- aggregate(DF$steps,by=list(DF$date),FUN=sum, na.rm=TRUE)
names(DFsum) <- c("date","totalSteps")

# I added the mean and median to title of histogram
hist(DFsum$totalSteps, main=paste("Histogram total Steps, mean ",
    sprintf("%.2f",mean(DFsum$totalSteps)), " median",
    sprintf("%.2f",median(DFsum$totalSteps))), breaks = 25,
    xlab = "Total Steps")
```



What is the average daily activity pattern?

1. 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)
2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
DF5min <- aggregate(DF$steps,by=list(DF$interval),FUN=mean, na.rm=TRUE)
names(DF5min) <- c("interval","average")

plot(DF5min$interval, DF5min$average, type="l", col="dark blue", lwd=3,
     main = "Average of steps per 5 min Interval",
     xlab="5 min interval", ylab="average steps")
```



Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

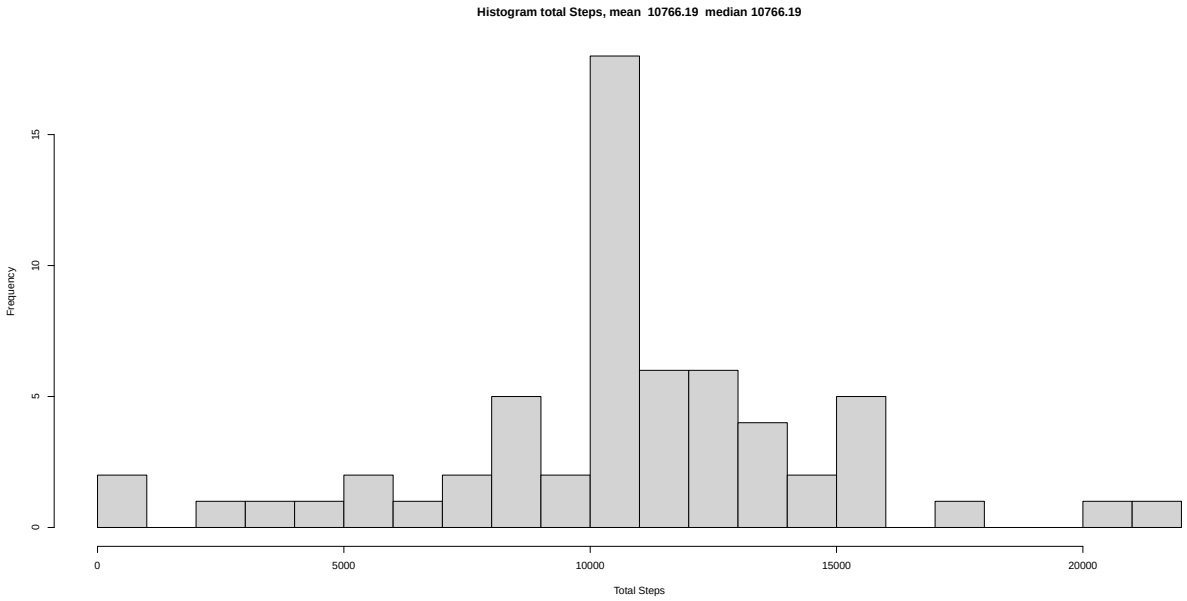
1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

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.
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?

```
#Show information of nas
naCount <- apply(DF,2,function(x) sum(is.na(x)))
naCount[naCount != 0]

# only has nas in steps.
# strategy use the average per 5 min interval DF5min dataset

DFNoNa <- DF
DFNoNa$steps[is.na(DFNoNa$steps)] <-
             sapply(DFNoNa$interval[is.na(DFNoNa$steps)],
                    function(x) DF5min$average[DF5min$interval == x])

#Show information of nas
naCount <- apply(DFNoNa,2,function(x) sum(is.na(x)))
naCount[naCount != 0]
str(DFNoNa)
summary(DFNoNa)
```

```
DFNoNaSum <- aggregate(DFNoNa$steps,by=list(DFNoNa$date),FUN=sum, na.rm=TRUE)
names(DFNoNaSum) <- c("date","totalSteps")

# I added the mean and median to title of histogram
hist(DFNoNaSum$totalSteps,
     main=paste("Histogram total Steps, mean ",
                sprintf("%.2f",mean(DFNoNaSum$totalSteps)), " median",
                sprintf("%.2f",median(DFNoNaSum$totalSteps))),
     breaks = 25, xlab = "Total Steps")
```


Histogram total Steps, mean 10766.19 median 10766.19
Frequency
0
5
10
15
0
5000
10000
15000
20000
Total Steps

Are there differences in activity patterns between weekdays and weekends?

1. 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.
2. Make a panel plot containing a time series plot **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.

```
DF$typeDay <- factor(as.character(format(DF$date,"%w") == 0 |
                                  format(DF$date,"%w") == 6 )
                     , levels =c("FALSE","TRUE")
                     , labels=c("weekday","weekend"))

DF5minTypeDay <- aggregate(DF$steps,by=list(DF$interval, DF$typeDay),
                           FUN=mean, na.rm=TRUE)
names(DF5minTypeDay) <- c("interval","typeDay","average")
summary(DF5minTypeDay)
```

```
xyplot(average ~ interval | typeDay, data=DF5minTypeDay, type="l",layout=c(1,2)
       , ylab="Number of steps")
```