Using R markdown to document R programs.

author: A. A. El Haddi date: Jan, 13th, 2019

Note: how to process the Rmd file.

Use the following command line to process the Rmd file. Best to put the following line in a bash script or Makefile and pass in the file names as arguments

R -e "rmarkdown::render('$1')"

Include the packages that we will need

```
library(lubridate)
library(ggplot2)
```

Loading and preprocessing the data

Load the data

1. Code for reading in the dataset and/or processing the data

```
zipactivity <- "activity.zip"
unzipactivity <- "activity.csv"
activity <- read.csv2(unz(zipactivity, unzipactivity), header = TRUE, sep = ",",  dec = ".", fill = TRU
```

Quick check the data

```
dim(activity)
```

```
## [1] 17568     3
```

```
head(activity)
```

```
##   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
```

```r
names(activity)
```

```
## [1] "steps"    "date"     "interval"
```

Transform dates, add weekends, etc

```r
activity$date<-as.Date(activity$date, format="%Y-%m-%d")
activity$weekday <- weekdays(activity$date)
activity$dayofweek <- wday(activity$date)
activity$isweekend <- wday(activity$date) %in% c(1,7)

ssize <- dim(activity)[1]
 activity$wdaytype <- "weekday"
 activity$wcolor <- "red"

 for (i in 1:ssize) {
    if (activity[i, "isweekend"])  {
        activity[i, "wdaytype"] <- "weekend"
        activity$wcolor <- "green"
    }
 }

head(activity)
```

```
##   steps       date interval weekday dayofweek isweekend wdaytype wcolor
## 1    NA 2012-10-01        0  Monday         2     FALSE  weekday  green
## 2    NA 2012-10-01        5  Monday         2     FALSE  weekday  green
## 3    NA 2012-10-01       10  Monday         2     FALSE  weekday  green
## 4    NA 2012-10-01       15  Monday         2     FALSE  weekday  green
## 5    NA 2012-10-01       20  Monday         2     FALSE  weekday  green
## 6    NA 2012-10-01       25  Monday         2     FALSE  weekday  green
```

```r
summary(activity)
```

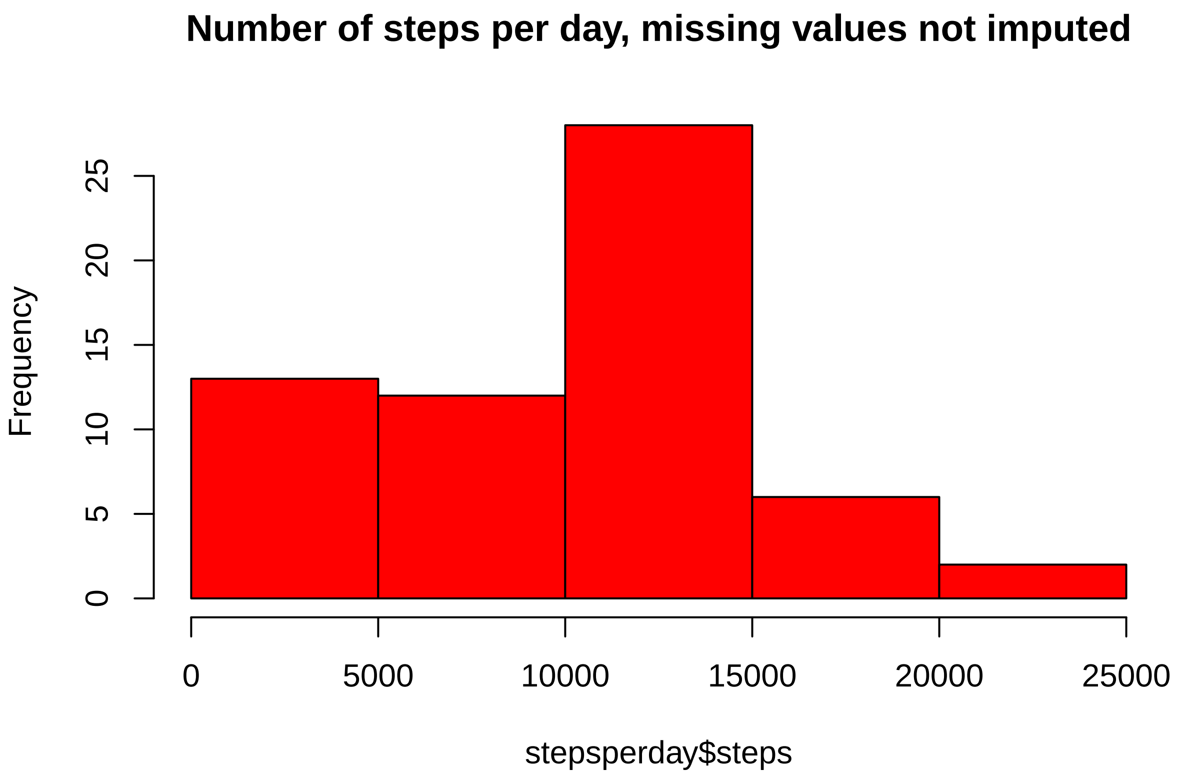
```
##      steps             date               interval        weekday
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0   Length:17568
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8   Class :character
##  Median :  0.00   Median :2012-10-31   Median :1177.5   Mode  :character
##  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
##    dayofweek isweekend         wdaytype            wcolor
##  Min.   :1   Mode :logical   Length:17568       Length:17568
##  1st Qu.:2   FALSE:12960     Class :character   Class :character
##  Median :4   TRUE :4608      Mode  :character   Mode  :character
##  Mean   :4
##  3rd Qu.:6
##  Max.   :7
##
```

Crude analysis including missing values

2. Histogram of the total number of steps taken each day

Ignore missing values

```r
stepsperday <- aggregate(activity[, "steps"], by=list(activity$date), FUN=sum, na.rm=TRUE)
colnames(stepsperday) <- c("date", "steps")
hist(stepsperday$steps, col="red", main="Number of steps per day, missing values not imputed")
```



3. Mean and median number of steps taken each day

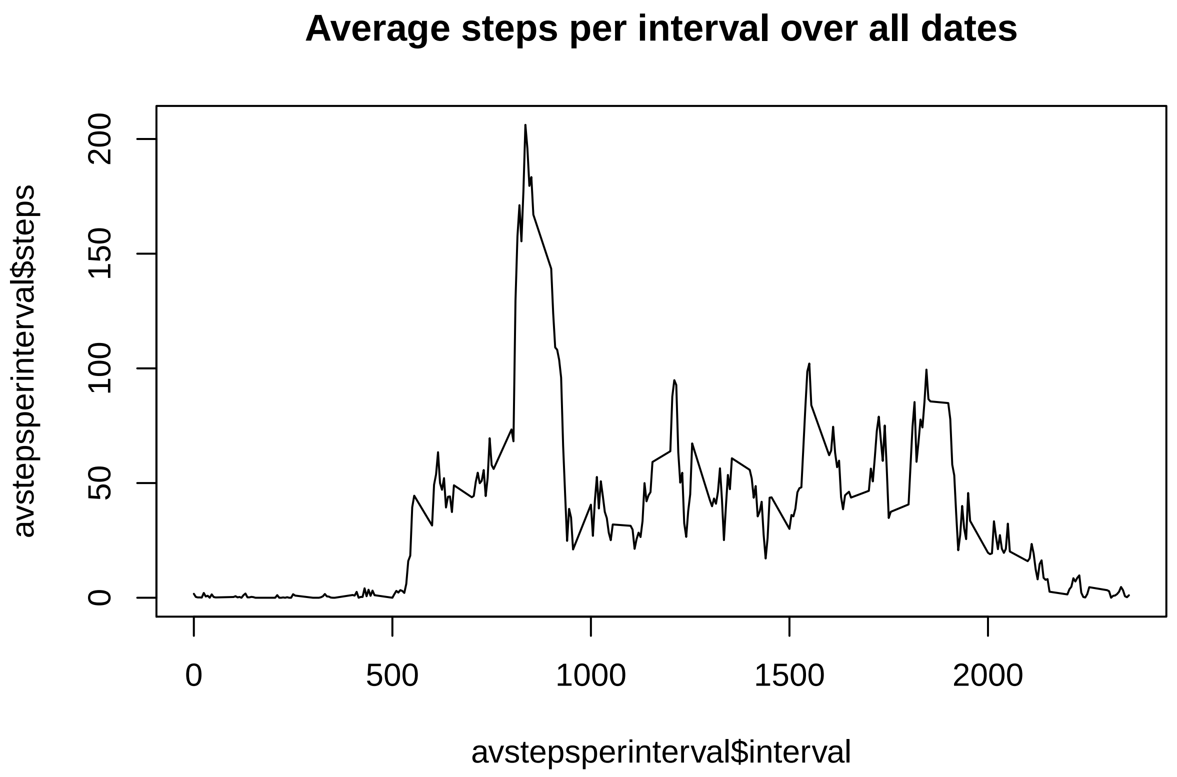
Get total steps per day

Here we did not exclude the missing values yet

- The mean total number of steps taken is 10395
- The median total number of steps taken is 10395

4. Time series plot of the average number of steps taken

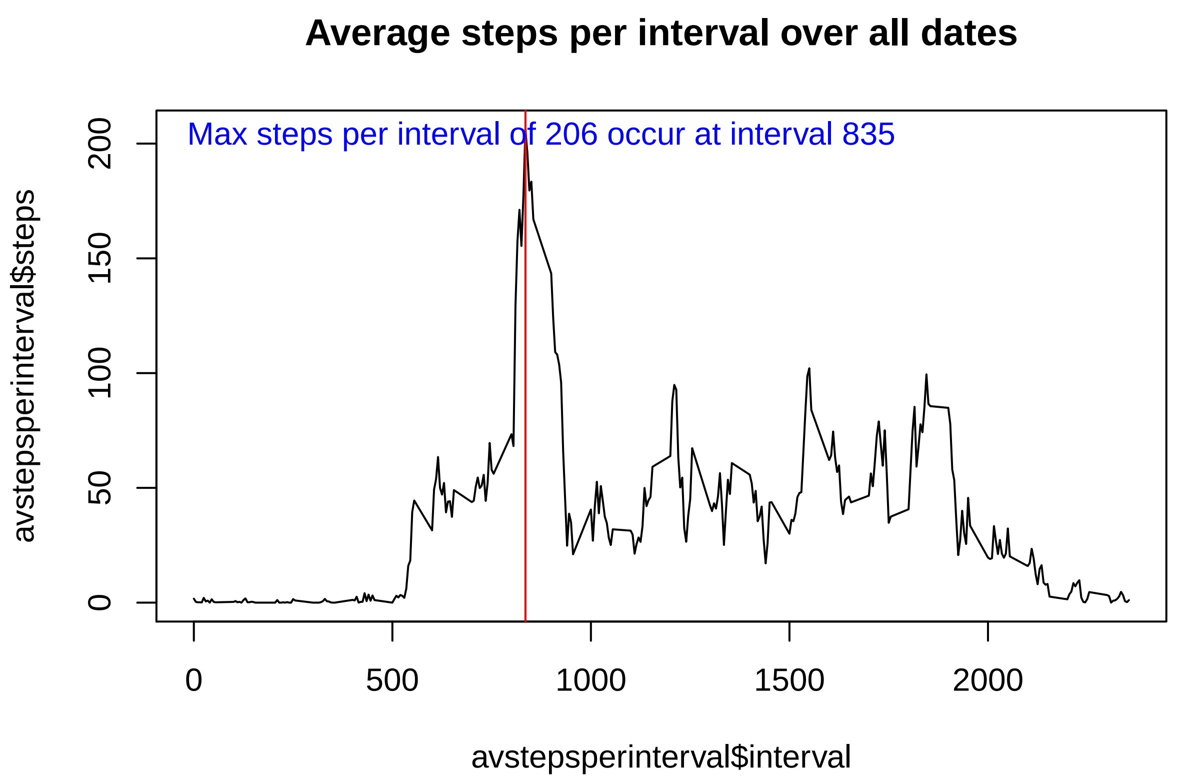
```r
avstepsperinterval <- aggregate(activity[, "steps"], by=list(activity$interval), FUN=mean, na.rm=TRUE)
colnames(avstepsperinterval) <- c("interval", "steps")
plot(avstepsperinterval$interval, avstepsperinterval$steps, type='l', main="Average steps per interval
```



```
maxsteps<-max(avstepsperinterval$steps, na.rm=TRUE)
```

5. Maximum number of steps per interval

```
s1<-subset(avstepsperinterval, steps==maxsteps, na.rm=TRUE)
maxi<-s1$interval
plot(avstepsperinterval$interval, avstepsperinterval$steps, type='l', main="Average steps per interval
abline(v=maxi, col="red")
slabel<-sprintf("Max steps per interval of %s occur at interval %s\n", round(maxsteps), maxi)
text(maxi+40, maxsteps-10,slabel, col="blue")
```



The red vertical line shows that the maximum number of steps per interval occurs at 835

6. Code to describe and show a strategy for imputing missing data

Find and treat missing values

We create another subset without missing values. We will use later to estimate missing values means.

imputing missing values

```r
# find missing values
mv <- is.na(activity$steps)

length(mv)
```

```
## [1] 17568
```

```r
# set without missing values
nomvactivity<-activity[!mv,]
dim(nomvactivity)
```

```
## [1] 15264     8
```

```r
# set with missing values
mvactivity<-activity[mv,]

print("Dimension of missing value data set")
```

```
## [1] "Dimension of missing value data set"
```

```
dim(mvactivity)
```

```
## [1] 2304    8
```

Total number of missing values : 2304

```
# We impute
# For each missing value we estimate the value using non missing values for that step

 # create another instance of the activity data set but track which steps were estimated

 imactivity <- activity
 imactivity$hadmv <- FALSE
 imactivity$orgsteps <- imactivity$steps

 for (i in 1:dim(imactivity)[1]) {

    ## redundant test but leave here in case we want to replace within same data set
    if (is.na(imactivity[i, "steps"]))  {

        # current interval
         cinterval<-imactivity[i, "interval"]

         # mean of all intervals equal to current interval without missing values
         umv <- mean(subset(imactivity, interval == cinterval)$steps, na.rm=TRUE)

         # Here we round up the number of steps
         imactivity[i, "steps"] <- round(umv)
         imactivity[i, "hadmv"] <- TRUE
    }
 }

summary(imactivity)
```

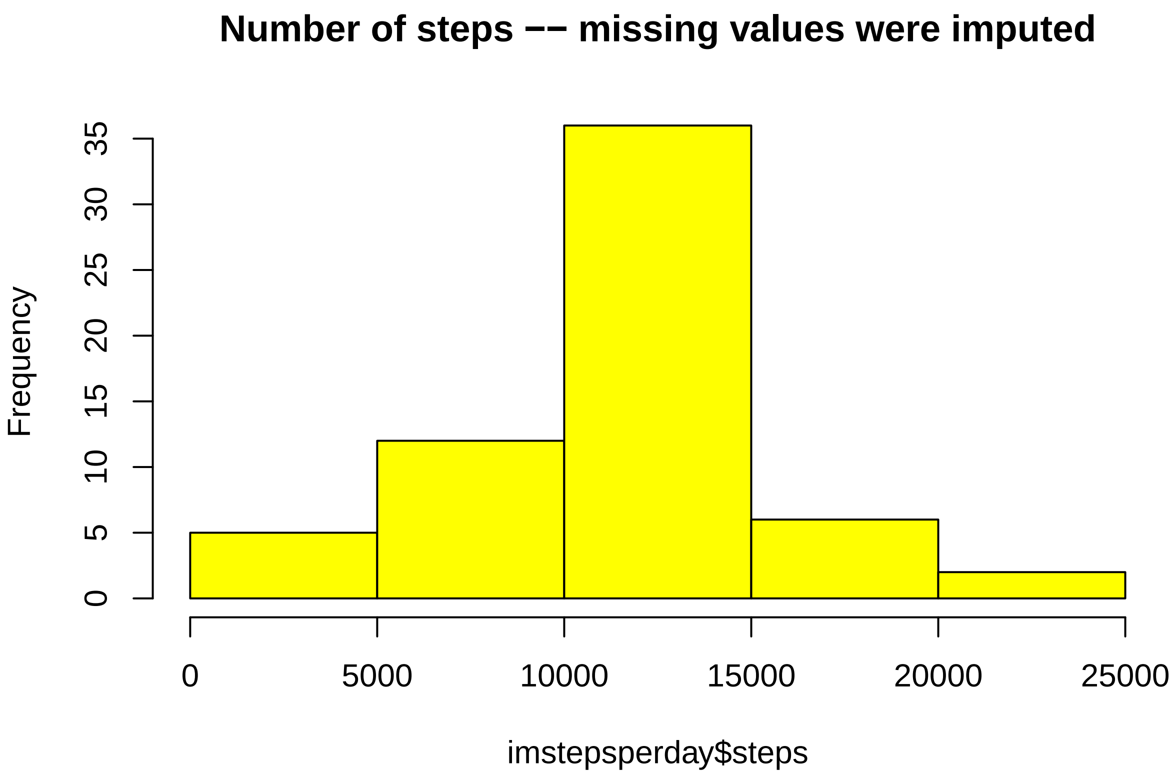
```
##      steps             date               interval        weekday
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0   Length:17568
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8   Class :character
##  Median :  0.00   Median :2012-10-31   Median :1177.5   Mode  :character
##  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
##
##    dayofweek isweekend         wdaytype            wcolor
##  Min.   :1   Mode :logical   Length:17568       Length:17568
##  1st Qu.:2   FALSE:12960     Class :character   Class :character
##  Median :4   TRUE :4608      Mode  :character   Mode  :character
##  Mean   :4
##  3rd Qu.:6
##  Max.   :7
##
##    hadmv             orgsteps
##  Mode :logical   Min.   :  0.00
##  FALSE:15264     1st Qu.:  0.00
##  TRUE :2304      Median :  0.00
##                  Mean   : 37.38
##                  3rd Qu.: 12.00
##                  Max.   :806.00
##                  NA's   :2304
```

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

```
# Total number of steps per day using imputed values.

imstepsperday <- aggregate(imactivity[, "steps"], by=list(imactivity$date), FUN=sum, na.rm=FALSE)
colnames(imstepsperday) <- c("date", "steps")

hist(imstepsperday$steps, col="yellow", main="Number of steps -- missing values were imputed")
```

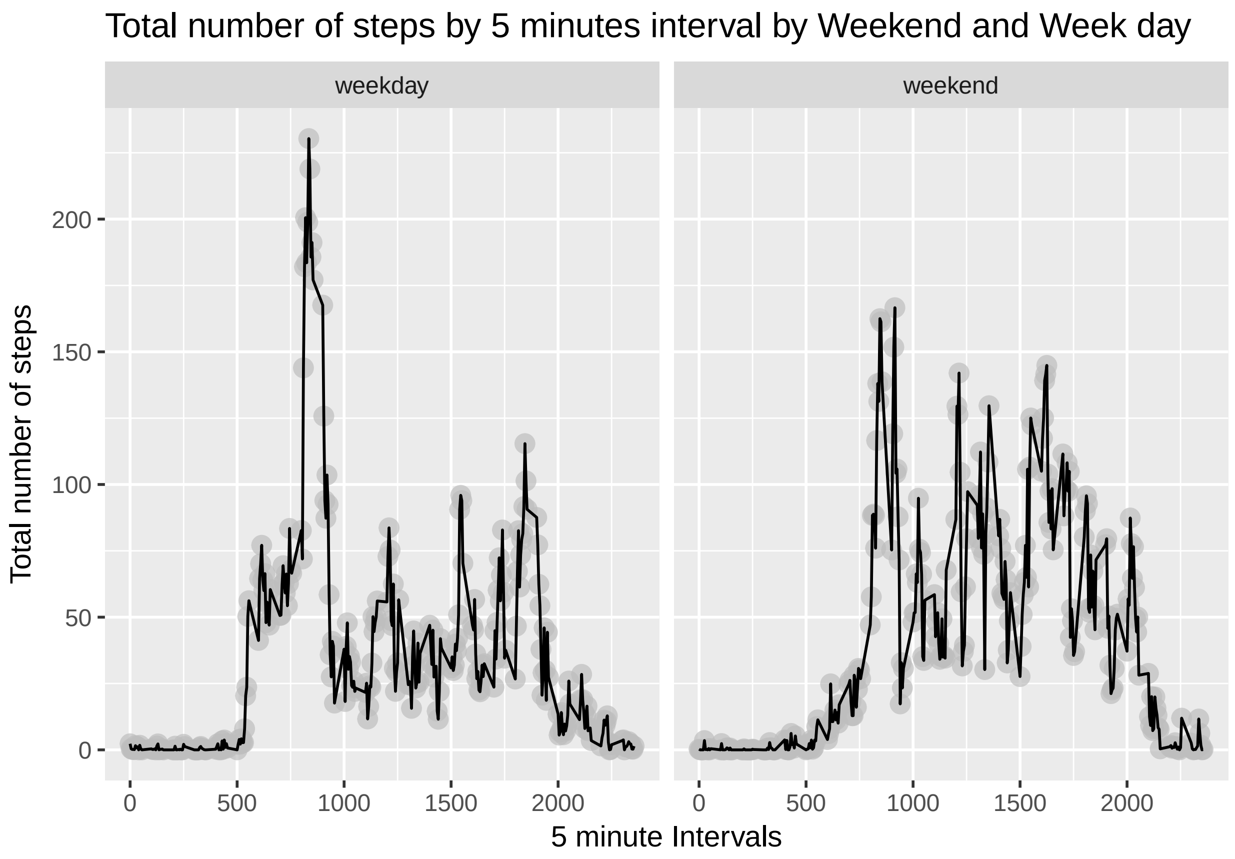


8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

(see transformsdates chunck where we already added weekdays, weekends etc)

```
avstepsperintervalwkday <- aggregate(imactivity[, "steps"], by=list(imactivity$interval, imactivity$wda
colnames(avstepsperintervalwkday) <- c("interval", "wdaytype", "steps")
```

```
ggplot(avstepsperintervalwkday, aes(x=interval, y=steps)) +
   geom_point( color="gray", size=3, alpha=.7) +
   facet_grid(. ~ wdaytype) +
   labs(x="5 minute Intervals", y="Total number of steps") +
   ggtitle("Total number of steps by 5 minutes interval by Weekend and Week day") +
   geom_line()
```


Total number of steps by 5 minutes interval by Weekend and Week day
weekday
weekend
Total number of steps
5 minute Intervals