Activity Mon Analysis

TJM

February 25, 2018

Title: TJM file on Reprod Rsch Week 2

Unzip and read in file to “Data”

```
setwd("C:/Users/ctmun/OneDrive/Desktop/ReproRsch")

zipF <- file.choose()    ## chose activity file in zip folder
outDir <- "C:/Users/ctmun/OneDrive/Desktop/ReproRsch"
unzip(zipF, exdir=outDir)
data <- read.csv("activity.csv", header=T)
```

Initial Explor analysis with some output shown head(data) ## steps date interval ## 1 NA 2012-10-01 0 ## 2 NA 2012-10-01 5

tail(data) ## steps date interval ## 17563 NA 2012-11-30 2330 ## 17564 NA 2012-11-30 2335

```
dim(data)
```

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

```
##   [1] 17568     3

summary(data)
```

```
##      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
```

```
##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

str(data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

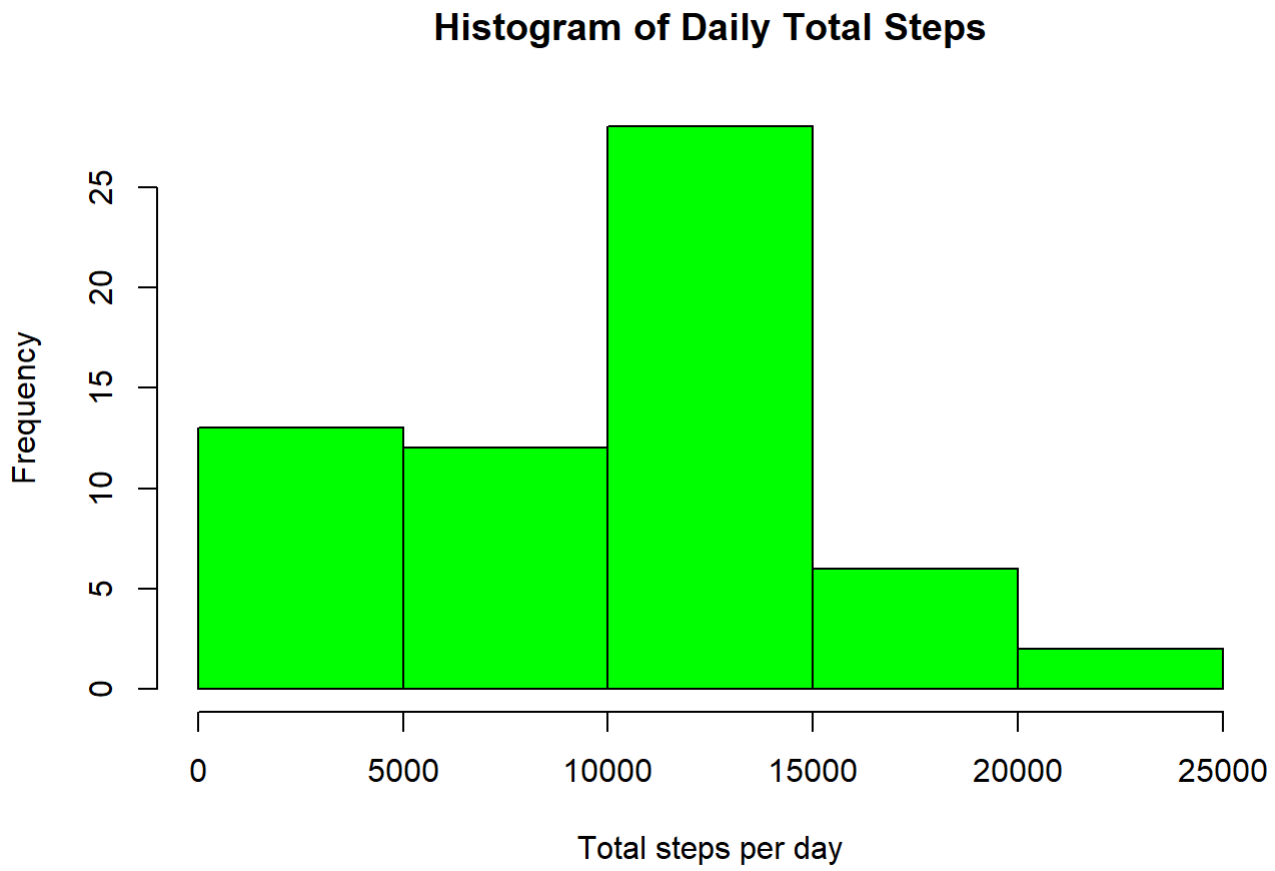
```
## 'data.frame':    17568 obs. of  3 variables:
## $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
## $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
## $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

Now some data manipulation with date and calculating total steps followed by histogram

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

              ## Calculate and plot steps per day ##
## Calculate steps per day.  Group by date then sum steps
steps_day <- tapply(data$steps, data$date, sum, na.rm=TRUE)
steps_day <- as.data.frame(steps_day)

## plot histogram
hist(steps_day$steps_day, col = "green", xlab = "Total steps per day", main = "Histogram of Dail
y Total Steps")
```



Eliminate na and 0 values and calc mean and median

```
ref_steps <- steps_day[!(steps_day$steps_day == 0),]
Meansteps <- mean(ref_steps)
Meansteps <- round(Meansteps, digits = 2)
## Meansteps
## [1] 10766.19
Mediansteps <- median(ref_steps)
## Mediansteps
## [1] 10765
print("The summary statistics for steps per day follow:")
```

```
## [1] "The summary statistics for steps per day follow:"
```

```
print(c("Average steps per day : " , Meansteps))
```

```
## [1] "Average steps per day : " "10766.19"
```

```
print(c("Median steps per day : ", Mediansteps))
```

```
## [1] "Median steps per day : " "10765"
```

Calculating Daily Activity Pattern

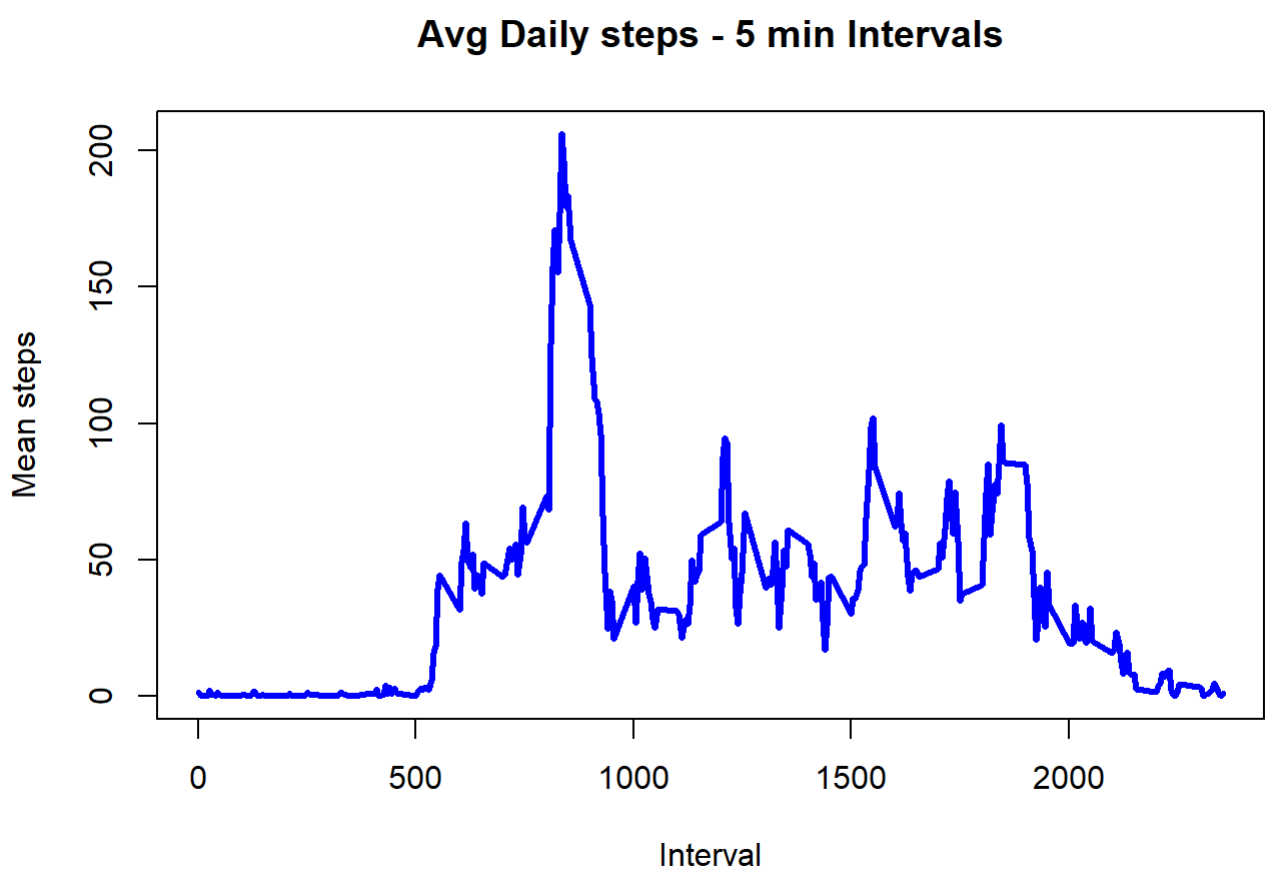
```
library(dplyr)
```

```
##
## 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
```

```
Intmean <- data %>% group_by(interval) %>% summarise(meanst=mean(steps, na.rm = TRUE))
plot(Intmean$interval, Intmean$meanst, type="l", lwd=3, main = "Avg Daily steps - 5 min Interval
s", xlab = "Interval", ylab = "Mean steps", col = "blue")
```




```
## Calculate max and print
a <- Intmean$meanst == max(Intmean$meanst)
b <- Intmean[a,]
b[1,2] <- round(b[1,2], digits = 2)
print(c("The maximum average step interval is:", b[1,]))
```

```
## [[1]]
## [1] "The maximum average step interval is:"
##
## $interval
## [1] 835
##
## $meanst
## [1] 206.17
```

Impute missing values

Tried mice package first with limited success ## switched to manual repl with interval mean

```
s <- sum(is.na(data))
## [1] 2304
print(c("The number of missing values is:", s))
```

```
## [1] "The number of missing values is:" "2304"
```

```
## imputing with mean by interval  ###
Updated_data <- data
ExtInterval <- rep(Intmean$interval, 61)
Extmeansteps <- rep(Intmean$meanst, 61)
Extmeandata <- as.data.frame(cbind(ExtInterval, Extmeansteps))
```

Checking data and calc's dim(Extmeandata) [1] 17568 2 head(Extmeandata) ExtInterval Extmeansteps 1 0 1.7169811 2 5 0.3396226 3 10 0.1320755 Extmeandata[286:290,] ExtInterval Extmeansteps 286 2345 0.6415094 287 2350 0.2264151 288 2355 1.0754717 289 0 1.7169811 290 5 0.3396226

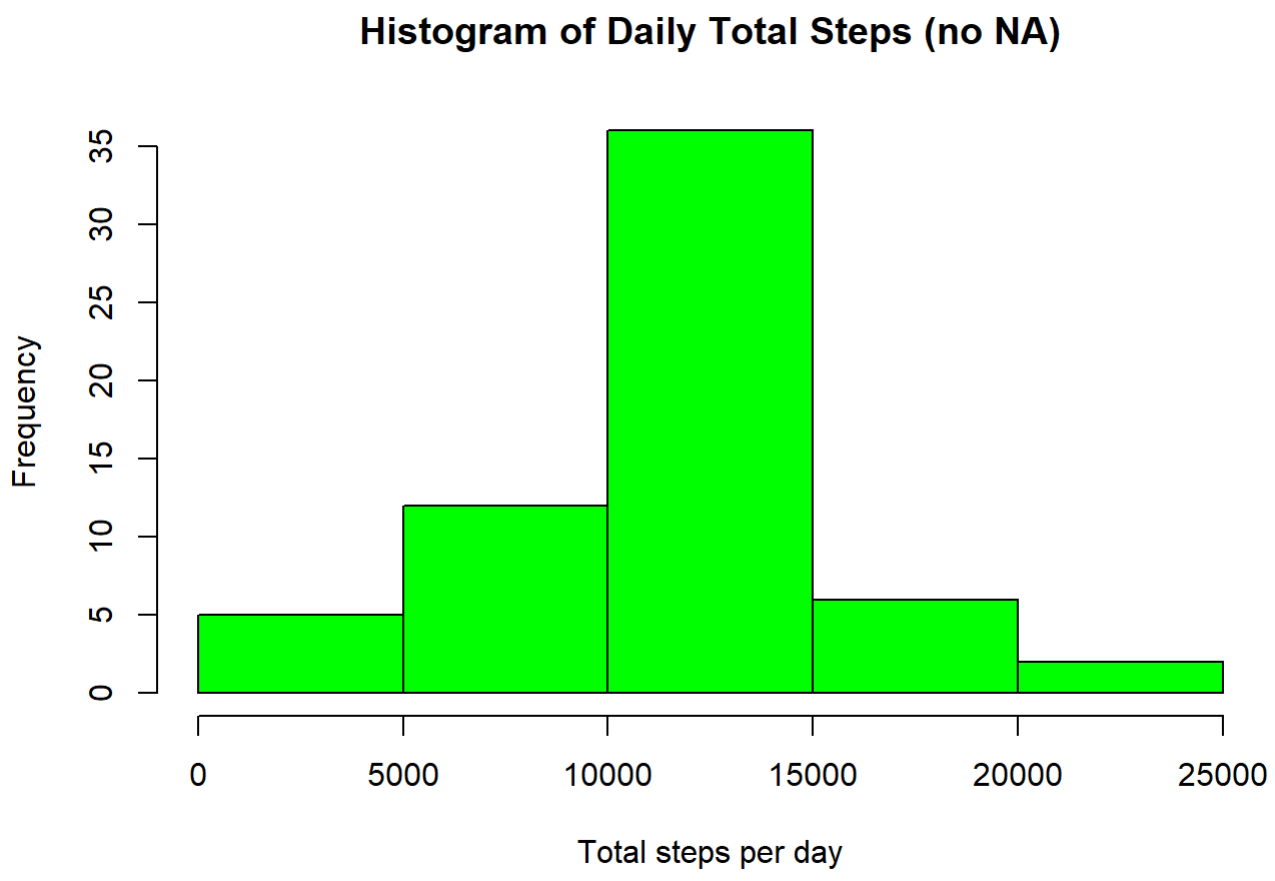
Now replace NA or missing values with mean for that interval

```
for (i in 1:17568) {  while (is.na(Updated_data[i,1]) | Updated_data[i,1] =="") { Updated_data
[i,1] <- Extmeandata[i,2]}}
## head(Updated_data)
##      steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
##  sum(is.na(Updated_data$steps))
##n[1] 0

## Summing data and making new histogram

newdailysteps <- tapply(Updated_data$steps, Updated_data$date, sum)
newdailysteps <- as.data.frame(newdailysteps)

### second histogram with innputed data
hist(newdailysteps$newdailysteps, col = "green", xlab = "Total steps per day", main = "Histogram
 of Daily Total Steps (no NA)")
```



Calculate new mean / median with Imputed data

```
newmeansteps <- round(mean(newdailysteps$newdailysteps), digits=2)
## newmeansteps
## [1] 10766.19
newmediansteps <- median(newdailysteps$newdailysteps)
##  newmediansteps
##  [1] 10766.19
print("The updaated summary statistics for steps per day follow:")
```

```
## [1] "The updaated summary statistics for steps per day follow:"
```

```
print(c("New Average steps per day : " , newmeansteps))
```

```
## [1] "New Average steps per day : " "10766.19"
```

```
print(c("New Median steps per day : ", newmediansteps))
```

```
## [1] "New Median steps per day : " "10766.1886792453"
```

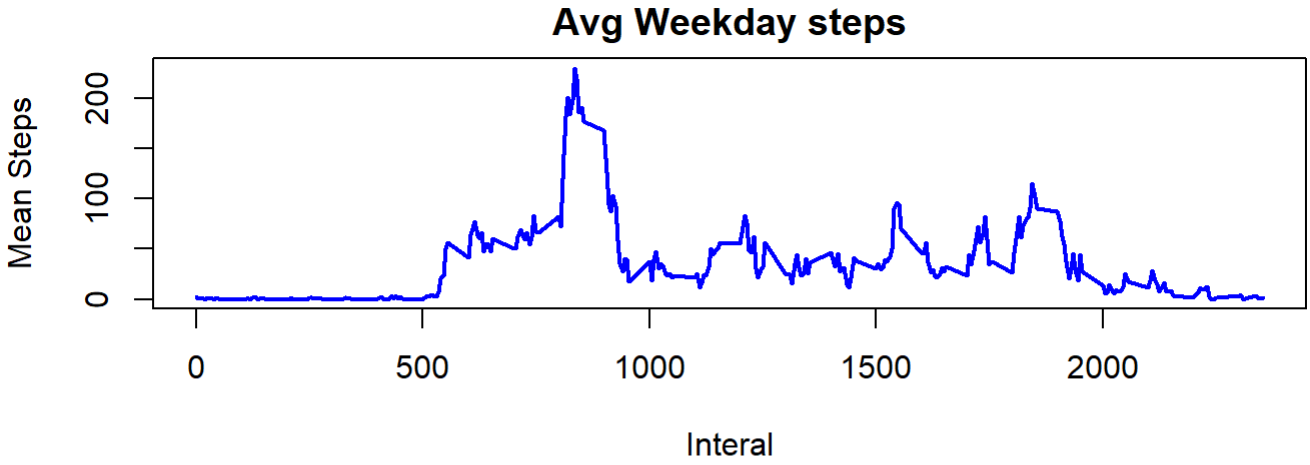
Patterns for weekdays and weekends with panel plot

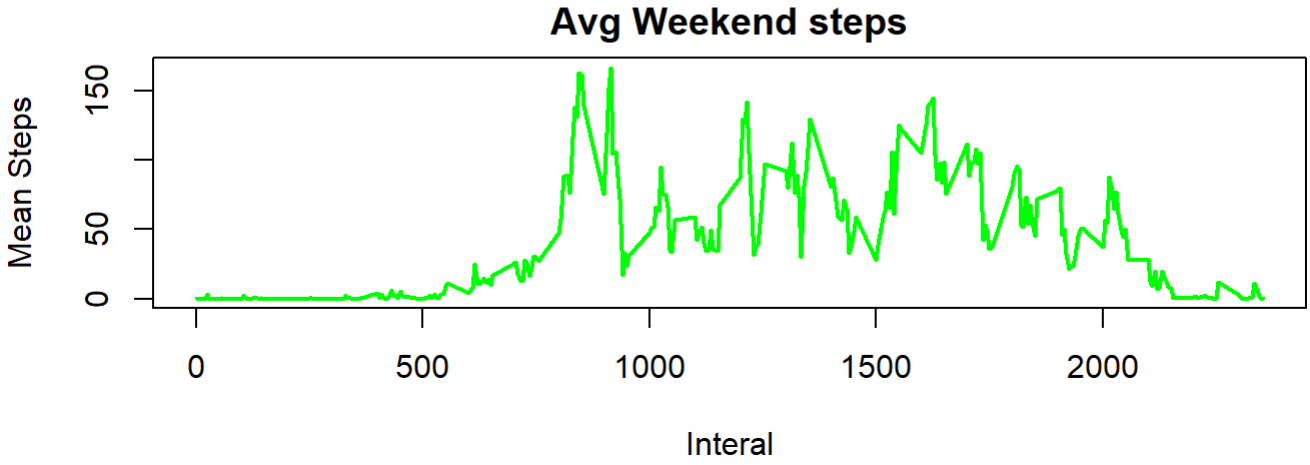
```
## calculate factor of weekday/weekend
Updated_data$date <- as.Date(Updated_data$date)
Updated_data$week <- ifelse(weekdays(Updated_data$date) %in% c("Saturday", "Sunday"), "weekend",
 "weekday")

## subset dataframe into 2 separate df's.
Weekdaydata <- Updated_data[Updated_data$week == "weekday",]
weekenddata <- Updated_data[Updated_data$week == "weekend",]
##  dim(Weekdaydata)
##  [1] 12960     4
##  dim(weekenddata)
##  [1] 4608    4

## now calculate mean steps by interval
wkdaymeans <- Weekdaydata %>% group_by(interval) %>% summarise(meanst=mean(steps, na.rm = TRUE))
wkendmeans <- weekenddata %>% group_by(interval) %>% summarise(meanst=mean(steps, na.rm = TRUE))

## panel plot with mean steps weekend and weekday  ##
par(mfrow =c(2,1), mar=c(4,4,2,1))
plot(wkdaymeans$interval, wkdaymeans$meanst, type="l", lwd=2, main="Avg Weekday steps", xlab="In
teral", ylab = "Mean Steps", col="blue")
plot(wkendmeans$interval, wkendmeans$meanst, type="l", lwd=2, main="Avg Weekend steps", xlab="In
teral", ylab = "Mean Steps", col="green")
```


Avg Weekday steps
Mean Steps
0
100
200
0
500
1000
1500
2000
Interal


Avg Weekend steps
Mean Steps
0
50
150
0
500
1000
1500
2000
Interal