Project_1

ZZ

12/2/2020

R Markdown

What is mean total number of steps taken per day?

1. Calculation of total number of step taken per day.

```
library(ggplot2)
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
```

```
setwd("~/Documents/jhdatascience")
activity <- read.csv("activity.csv")

activity$date <- as.Date(activity$date)

stepsPerDay <- activity %>%
        group_by(date) %>%
        summarize(sumsteps = sum(steps, na.rm = TRUE))
```

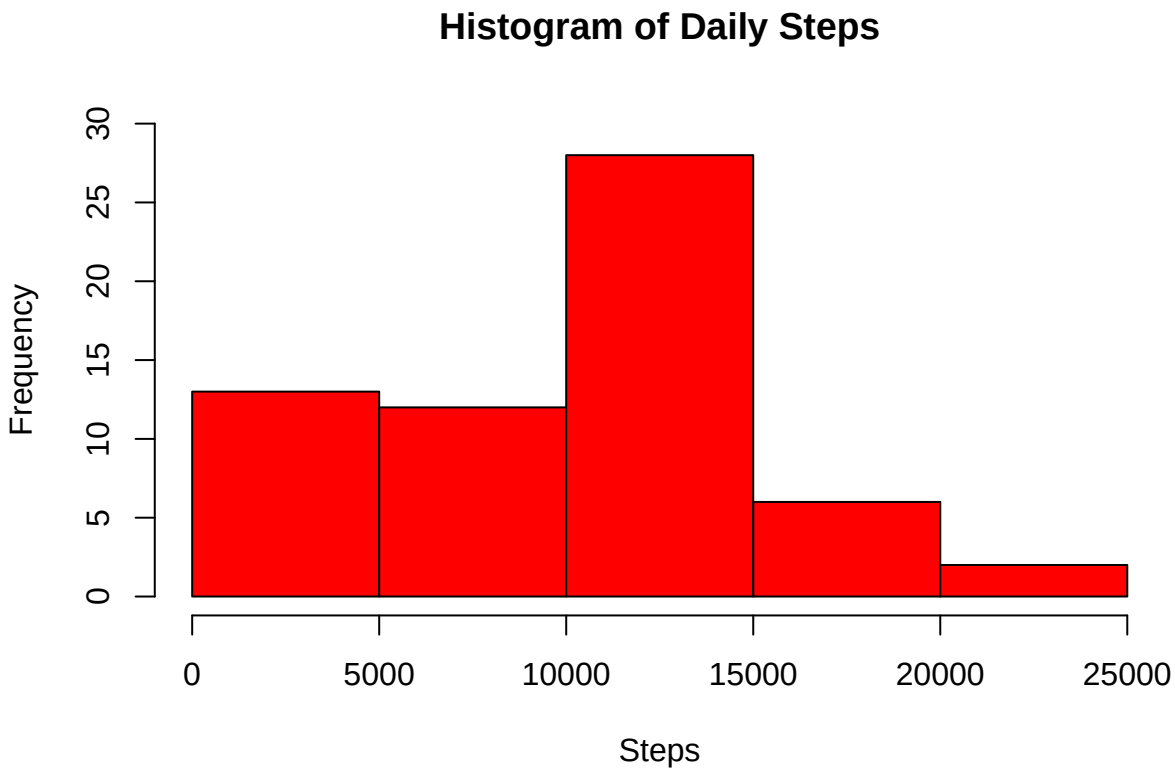
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(stepsPerDay,10)
```

```
## # A tibble: 10 x 2
##    date       sumsteps
##    <date>        <int>
##  1 2012-10-01        0
##  2 2012-10-02      126
##  3 2012-10-03    11352
##  4 2012-10-04    12116
##  5 2012-10-05    13294
##  6 2012-10-06    15420
##  7 2012-10-07    11015
##  8 2012-10-08        0
##  9 2012-10-09    12811
## 10 2012-10-10     9900
```

2. a histogram of the total number of steps taken each day.

```
hist(stepsPerDay$sumsteps, main = "Histogram of Daily Steps",
     col="red", xlab="Steps", ylim = c(0,30))
```



3. Calculate and report the mean and median of the total number of steps taken per day.

```
meanPreNA <- round(mean(stepsPerDay$sumsteps),digits = 2)
medianPreNA <- round(median(stepsPerDay$sumsteps),digits = 2)

print(paste("The mean is: ", meanPreNA))
```

```
## [1] "The mean is:  9354.23"
```

```
print(paste("The median is: ", medianPreNA))
```

```
## [1] "The median is:  10395"
```

What is the average daily activity pattern? 1. step by time interval

```
stepsPerInterval <- activity %>%
        group_by(interval) %>%
        summarize(meansteps = mean(steps, na.rm = TRUE))
```

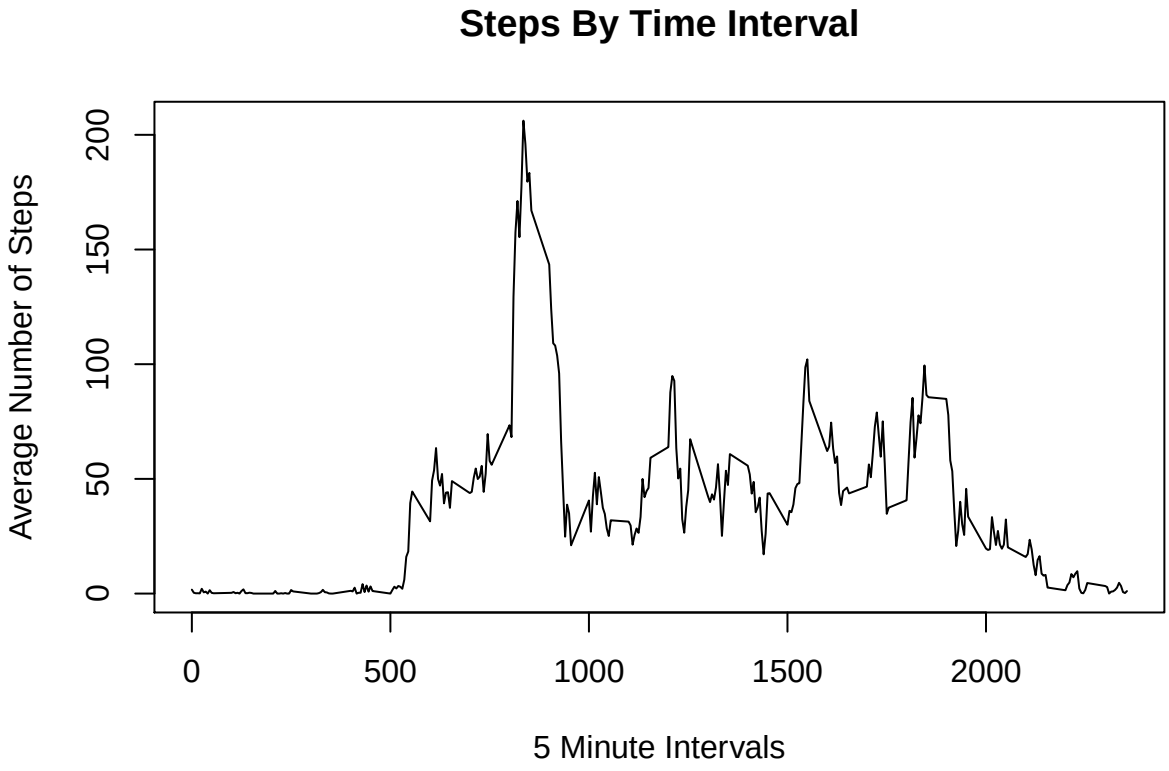
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
#Display first 10 rows of data
head(stepsPerInterval,10)
```

```
## # A tibble: 10 x 2
##    interval meansteps
##       <int>     <dbl>
##  1        0    1.72
##  2        5    0.340
```

```
##  3        10     0.132
##  4        15     0.151
##  5        20     0.0755
##  6        25     2.09
##  7        30     0.528
##  8        35     0.868
##  9        40     0
## 10        45     1.47
```

```
plot(stepsPerInterval$meansteps ~ stepsPerInterval$interval,
     col="black", type="l", xlab = "5 Minute Intervals", ylab = "Average Number of Steps",
     main = "Steps By Time Interval")
```



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

```
print(paste("Interval containing the most steps on average: ",stepsPerInterval$interval[which.max(steps
```

```
## [1] "Interval containing the most steps on average:  835"
```

```
print(paste("Average steps for that interval: ",round(max(stepsPerInterval$meansteps),digits=2)))
```

```
## [1] "Average steps for that interval:  206.17"
```

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

```
print(paste("The total number of rows with NA is: ", sum(is.na(activity$steps))))
```

```
## [1] "The total number of rows with NA is:  2304"
```

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. Answer: There are several strategy to do this, one way that is listed in the example is the

midrange values, which take the mean and median numuric values, it is easy to compare, the disvantage of this method would bethe posibility of overfitting. The other way would be imputation, specifically perturbation.

3.Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
head(activity,15)
```

```
##    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
## 7     NA 2012-10-01       30
## 8     NA 2012-10-01       35
## 9     NA 2012-10-01       40
## 10    NA 2012-10-01       45
## 11    NA 2012-10-01       50
## 12    NA 2012-10-01       55
## 13    NA 2012-10-01      100
## 14    NA 2012-10-01      105
## 15    NA 2012-10-01      110
```

```
activityNoNA <- activity
for (i in 1:nrow(activity)){
        if(is.na(activity$steps[i])){
                activityNoNA$steps[i]<- stepsPerInterval$meansteps[activityNoNA$interval[i] == stepsPer
        }
}
head(activityNoNA,15)
```

```
##        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
## 7  0.5283019 2012-10-01       30
## 8  0.8679245 2012-10-01       35
## 9  0.0000000 2012-10-01       40
## 10 1.4716981 2012-10-01       45
## 11 0.3018868 2012-10-01       50
## 12 0.1320755 2012-10-01       55
## 13 0.3207547 2012-10-01      100
## 14 0.6792453 2012-10-01      105
## 15 0.1509434 2012-10-01      110
```

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?

```
stepsPerDay <- activityNoNA %>%
        group_by(date) %>%
```

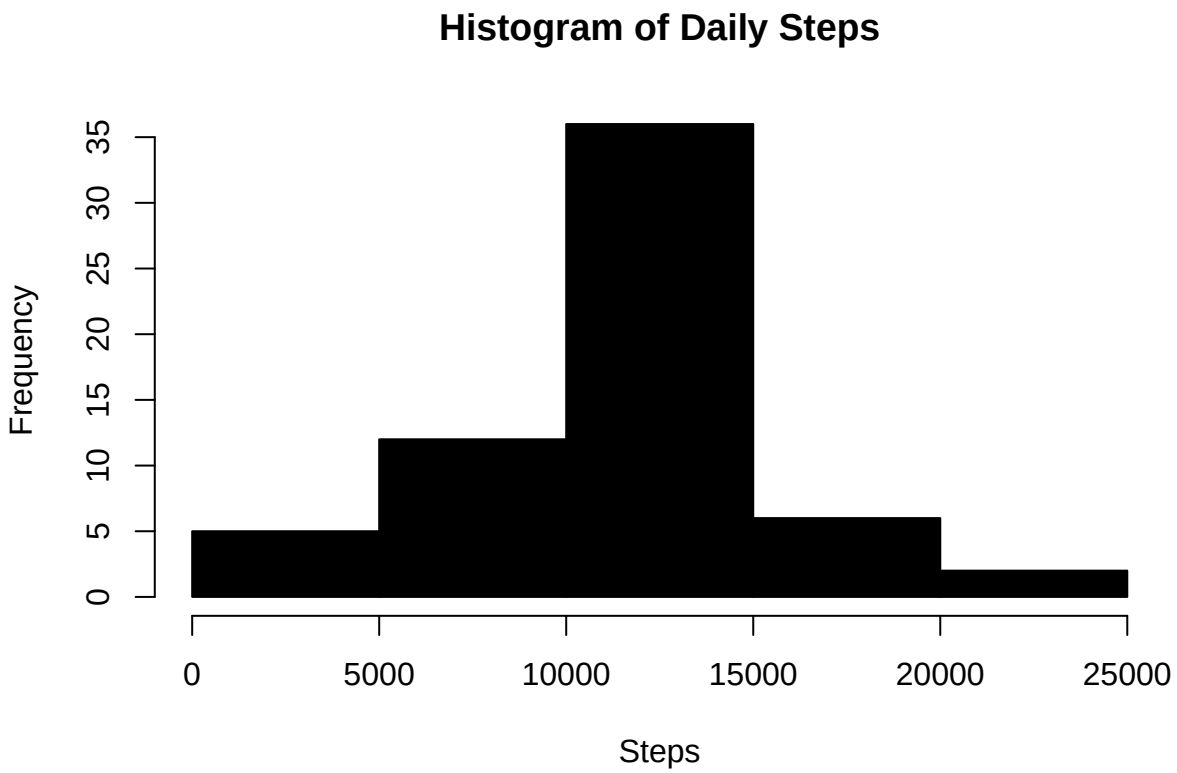
```
        summarize(sumsteps = sum(steps, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(stepsPerDay,5)
```

```
## # A tibble: 5 x 2
##   date       sumsteps
##   <date>        <dbl>
## 1 2012-10-01   10766.
## 2 2012-10-02     126
## 3 2012-10-03   11352
## 4 2012-10-04   12116
## 5 2012-10-05   13294
```

```
hist(stepsPerDay$sumsteps, main = "Histogram of Daily Steps",
     col="black", xlab="Steps")
```




```
meanPostNA <- round(mean(stepsPerDay$sumsteps), digits = 2)
medianPostNA <- round(median(stepsPerDay$sumsteps), digits = 2)

print(paste("mean = ", mean(meanPostNA)))
```

```
## [1] "mean =  10766.19"
```

```
print(paste("median = ", median(medianPostNA)))
```

```
## [1] "median =  10766.19"
```

```
NACompare <- data.frame(mean = c(meanPreNA,meanPostNA),median = c(medianPreNA,medianPostNA))
rownames(NACompare) <- c("Pre NA Transformation", "Post NA Transformation")
print(NACompare)
```

```
##                              mean   median
## Pre NA Transformation   9354.23 10395.00
## Post NA Transformation 10766.19 10766.19
```

Are there differences in activity patterns between weekdays and weekends?

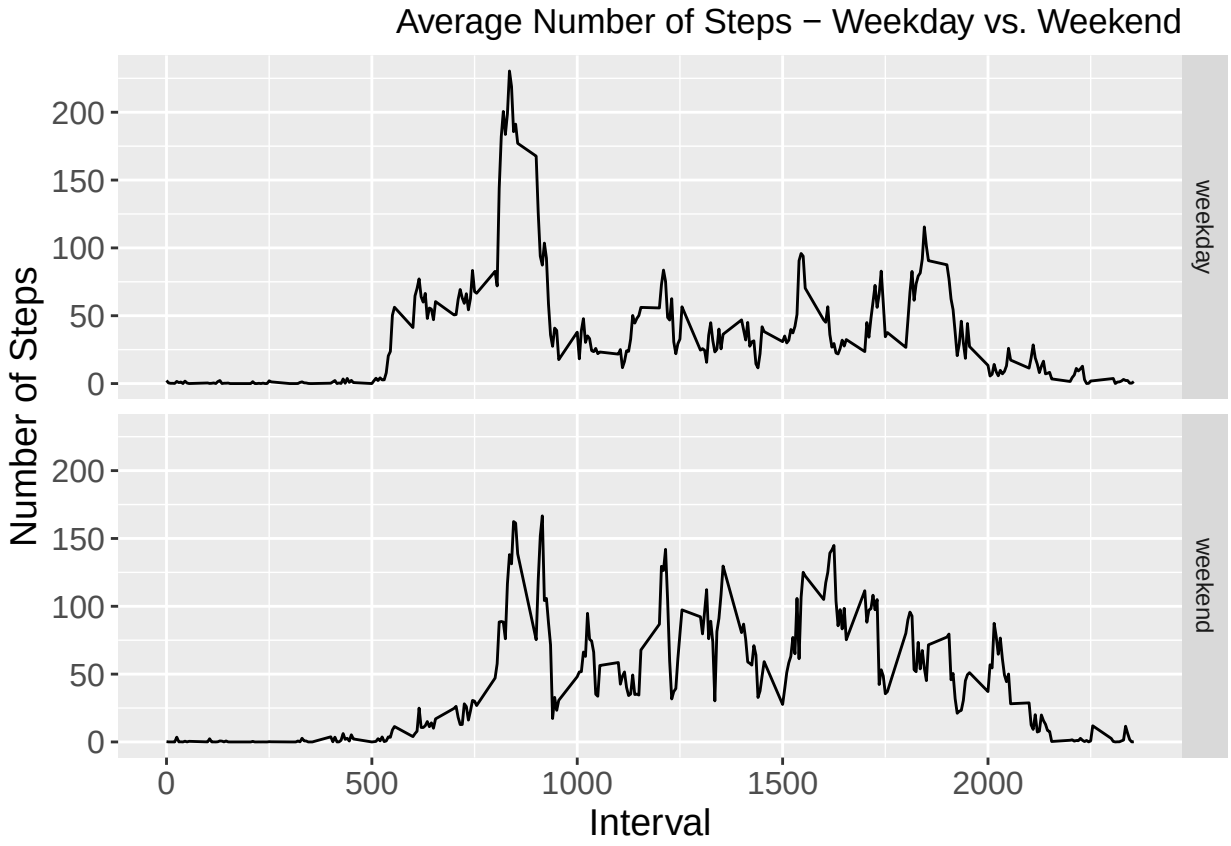
```
activityDoW <- activityNoNA
activityDoW$date <- as.Date(activityDoW$date)
activityDoW$day <- ifelse(weekdays(activityDoW$date) %in% c("Saturday", "Sunday"), "weekend", "weekday")
activityDoW$day <- as.factor(activityDoW$day)

activityWeekday <- filter(activityDoW, activityDoW$day == "weekday")
activityWeekend <- filter(activityDoW, activityDoW$day == "weekend")

activityWeekday <- activityWeekday %>%
        group_by(interval) %>%
        summarize(steps = mean(steps))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
activityWeekday$day <- "weekday"

activityWeekend <- activityWeekend %>%
        group_by(interval) %>%
        summarize(steps = mean(steps))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
activityWeekend$day <- "weekend"

wkdayWkend <- rbind(activityWeekday, activityWeekend)
wkdayWkend$day <- as.factor(wkdayWkend$day)


g <- ggplot (wkdayWkend, aes (interval, steps))
g + geom_line() + facet_grid (day~.) +
        theme(axis.text = element_text(size = 12),axis.title = element_text(size = 14)) +
        labs(y = "Number of Steps") + labs(x = "Interval") +
        ggtitle("Average Number of Steps - Weekday vs. Weekend") +
        theme(plot.title = element_text(hjust = 1.0))
```



The graph demonstrate differences in the patterns on the average daily intervals. For weekdays - there are larger spike in the morning intervals which could indicate their activites of morning work out or commuting to work. Fod the weekend- the acitivites are more balances as the individual may doing all kind of activites instead of working or sitting in an set hours time of time.