Reproducible Research: Assignment 1

Di Y.

9/1/2020

Loading and preprocessing the data

```
# Read the data
url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(url, destfile = paste0(getwd(), '/repdata%2Fdata%2Factivity.zip'), method = "curl")
unzip("repdata%2Fdata%2Factivity.zip",exdir = "data")
raw <- read.csv("./downloads/activity.csv", header=T, sep=",")
dim(raw) # 17568x3
```

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

```
str(raw)
```

```
## '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 ...
```

```
# Change the types of "date" and "interval" to date format
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
raw$date <- ymd(raw$date)

temp<- sprintf("%04d",raw$interval)
raw$interval <- format(strptime(temp,format="%H%M"),format="%H:%M") # data type:string
```

What is mean total number of steps taken per day?

```
# calculate the total number of steps taken per day
library(dplyr)
```

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

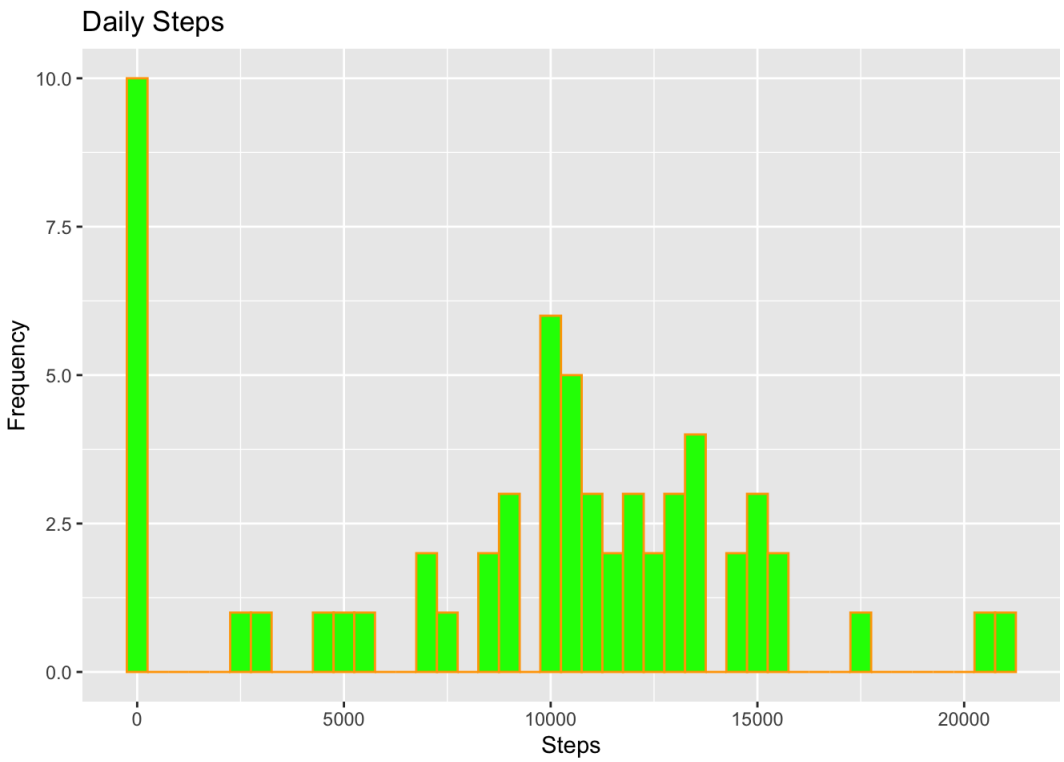
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
total_steps <- raw %>% group_by(date) %>% summarise(total_step=sum(steps, na.rm=T))
```

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

```
# Make a histogram of the total number of steps taken each day
library(ggplot2)
```




```
# Calculate and report the mean and median of the total number of steps taken per day
mean_median <- total_steps %>% summarise(mean=mean(total_step,na.rm=T),median=median(total_step,na.rm=T))
```

What is the average daily activity pattern?

Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
avg_steps <- raw %>% group_by(interval) %>% summarise(avg_steps=mean(steps,na.rm=T))
```

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

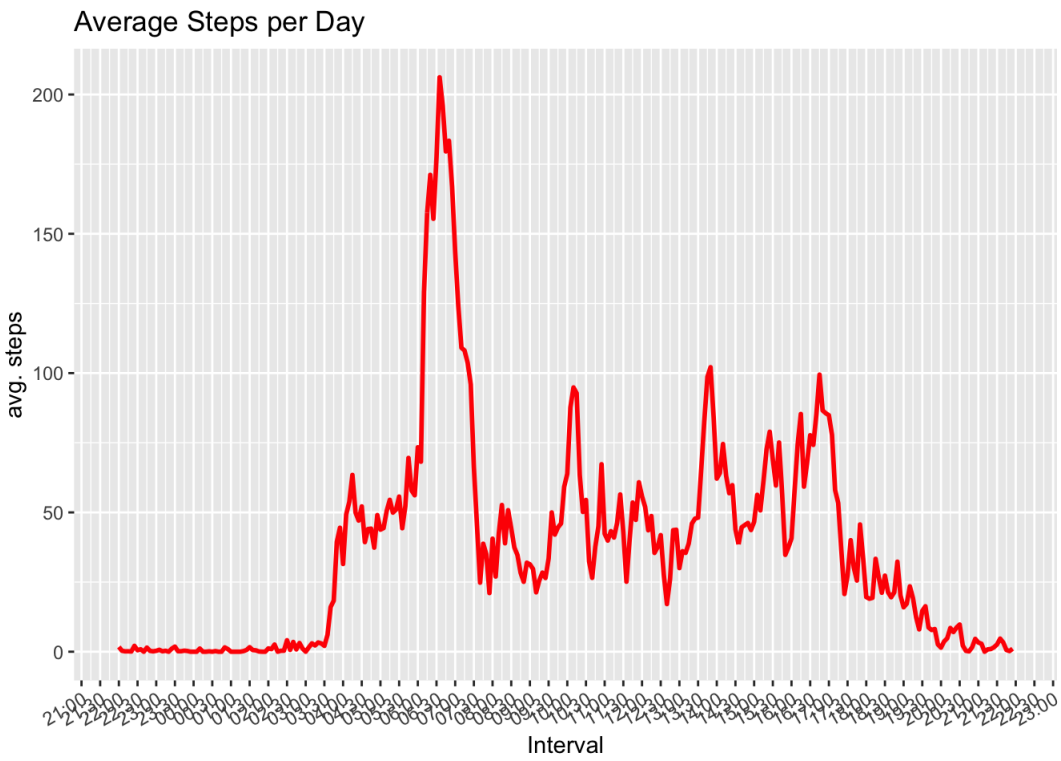
```
# "interval" column is string type and must be temporarily converted to POSIXct to plot a readable time series plot
install.packages("scales",repos = 'http://cran.us.r-project.org')
```

```
##
## The downloaded binary packages are in
##  /var/folders/5x/5xkzjtpx77l47xk83kvx7fl00000gn/T//RtmphYJrIA/downloaded_packages
```

```
library(scales)

raw_copy <- raw
raw_copy$interval <- as.POSIXct(raw_copy$interval,format="%H:%M")
avg_steps_copy <- raw_copy %>% group_by(interval) %>% summarise(avg_steps=mean(steps,na.rm=T))
```

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



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

```
max_steps <- avg_steps[which.max(avg_steps$avg_steps),][[1]]
```

Imputing missing values

calculate and report the total number of missing values in the dataset

```
table(is.na(raw))
```

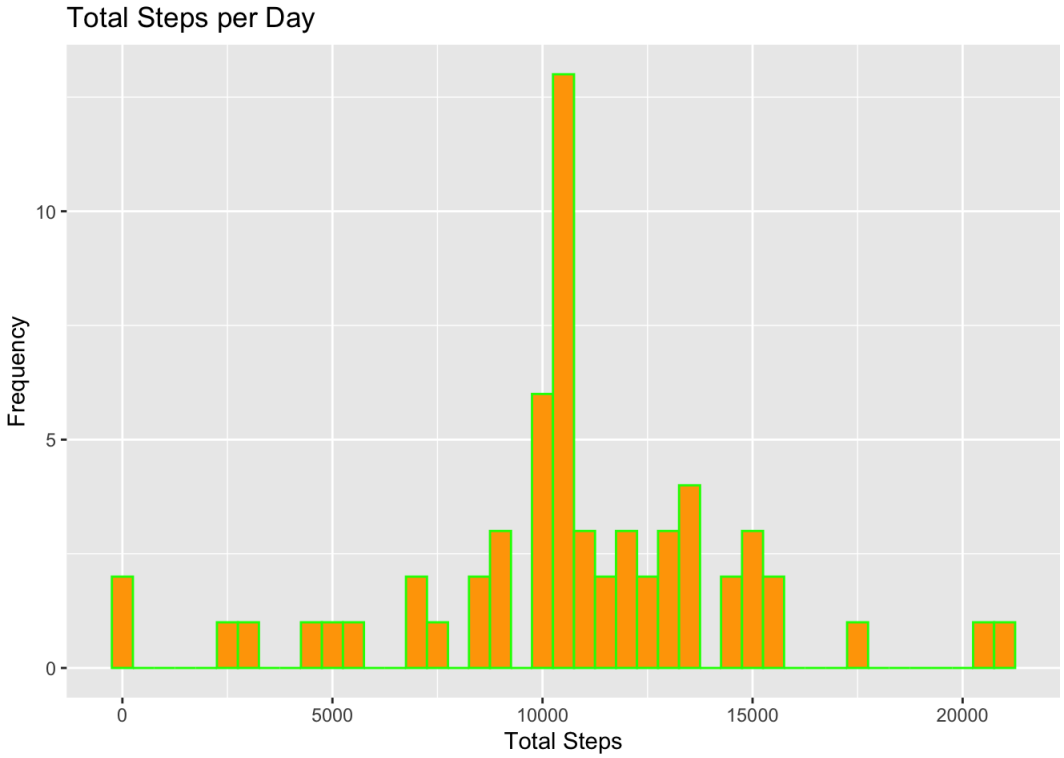
```
##
## FALSE  TRUE
## 50400  2304
```

```
table(is.na(raw[,2:3])) # no missing values in "date" and "interval" columns
```

```
##
## FALSE
## 35136
```

```
# Median of all measured days will be used to replace the missing values in df(total_steps).
total_steps_new <- total_steps
for (i in 1:length(total_steps$date)){
   if (total_steps_new$total_step[[i]]==0) {
   total_steps_new$total_step[[i]] <- mean_median$median
   }
}
```

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?



Since one day has 288 (=12*24)

intervals of 5 minutes, we can replace all the missing step values with median of the day/288.

```
mean_median_new <- total_steps_new %>% summarise(mean=mean(total_step),median=median(total_step))
mean_median_new
```

```
## # A tibble: 1 x 2
##     mean median
##    <dbl>  <int>
## 1 10718.  10395
```

```
# vs.
mean_median
```

```
## # A tibble: 1 x 2
##    mean median
##   <dbl>  <int>
## 1 9354.  10395
```

```
missing_total_steps <- filter(total_steps,total_step==0) # 8 days with missing steps

raw_new <- raw
for (j in 1:length(raw$steps)) {
   if(is.na(raw_new)[j]==T) {
      raw_new$steps[[j]] <- mean_median$median/288
   }
} # create a new data frame raw_new, with all missing values filled
```

Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays()weekdays() function may be of some help here.

```
# 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.
raw_new <- raw_new %>% mutate(weekday=weekdays(date,abbreviate=T))

# Create a factor with two levels ("weekend", "weekday")
weekend=c("Sa","So")
raw_new <- raw_new %>% mutate(label=factor((weekday %in% weekend),levels=c(FALSE,TRUE),labels=c("weekday","weekend")))
```

Make a panel plot containing a time series plot (i.e. type = "l"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).

```
raw_new$interval <- as.POSIXct(raw_new$interval,format="%H:%M")
avg_steps_new <- raw_new %>% group_by(interval,label) %>%
   summarise(avg_steps_new=mean(steps))
```

```
## `summarise()` regrouping output by 'interval' (override with `.groups` argument)
```