Reproducible Research: Peer Assessment 1

## Loading and preprocessing the data

I loaded the data Via read.csv function, and then by summary function we can have a taste on our data!

library(lubridate)

##   
## Attaching package: 'lubridate'

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

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

library(ggplot2)  
data <- read.csv("./activity.csv",header = T)  
summary(data)

## steps date interval   
## Min. : 0.00 Length:17568 Min. : 0.0   
## 1st Qu.: 0.00 Class :character 1st Qu.: 588.8   
## Median : 0.00 Mode :character Median :1177.5   
## Mean : 37.38 Mean :1177.5   
## 3rd Qu.: 12.00 3rd Qu.:1766.2   
## Max. :806.00 Max. :2355.0   
## NA's :2304

From above summary, we can notice there are 2304 intervals (observation) missing values of steps. Missing value will be imputed later, and I focused mainly on variable “date” and “interval”, since both of these interval are stored as data class hard to manipulate and plot.

1. Transform “date” from character to date object.
2. Transform “interval” from integer to character with my self-defined int\_time\_converter function, then use as.POSIXct function to transform character to POSIXct class.

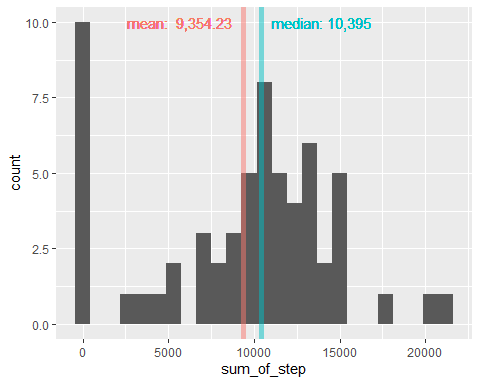
data <- data %>%  
 transform(date = ymd(date))  
  
int\_time\_converter <- function(x){  
 time\_str <- as.character(x)  
 time\_str <- paste0(paste0(rep("0",4 - nchar(time\_str)),collapse = ''),time\_str)  
 return(time\_str)  
}  
  
data$time <- as.POSIXct(unlist(lapply(data$interval,int\_time\_converter)),format = "%H%M")

## What is mean total number of steps taken per day?

To plot histogram of total steps taken per day as well as calculate means and median steps taken per day, I preprocess the data in following way:

1. Group the data by date.
2. Calculate the summation of each data
3. Plot histogram with ggplot2.
4. Draw vertical line to display mean and median

df\_q1 <- data %>%  
 group\_by(date) %>%  
 summarise(sum\_of\_step = sum(steps,na.rm = T))  
  
ggplot(data = df\_q1, aes(x = sum\_of\_step )) +  
 geom\_histogram(bins = 25)+  
   
 geom\_vline(xintercept = mean(df\_q1$sum\_of\_step),color = "#F8766D", size = 2,alpha = 0.5)+  
 geom\_vline(xintercept = median(df\_q1$sum\_of\_step),color = "#00BFC4",size = 2,alpha = 0.5)+  
   
 geom\_text(aes(label = paste("mean: ",format(round(mean(df\_q1$sum\_of\_step),2),big.mark = ",")),x =mean(df\_q1$sum\_of\_step),y=10),hjust = 1.1, vjust = 0.5,color = "#F8766D") +  
 geom\_text(aes(label = paste("median:",format(round(median(df\_q1$sum\_of\_step),2),big.mark = ",")),x =median(df\_q1$sum\_of\_step),y=10),hjust = -0.1, vjust = 0.5,color = "#00BFC4")

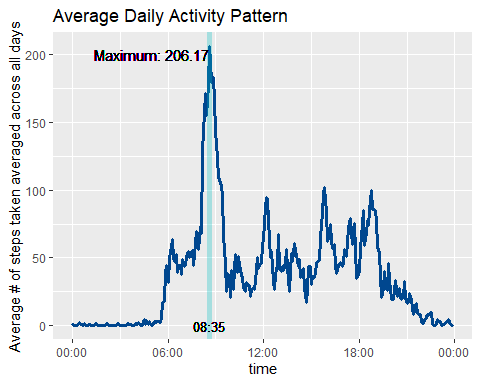


## what is the average daily activity pattern?

To plot time series of daily activity pattern, we first calculate average numbers of steps taken across all days via following steps:

1. Group data by interval
2. Calculate average of each interval
3. Compute the maximum value of averaged step.
4. Find time interval the maximum value lies in.

df\_q2 <- data %>%  
 group\_by(time) %>%  
 summarise(avg\_step = mean(steps, na.rm = T))  
  
max <- max(df\_q2$avg\_step)  
max\_time <- df\_q2$time[df\_q2$avg\_step == max]  
  
ggplot(data = df\_q2, aes(x = time, y = avg\_step))+  
 geom\_line(color = "#00478f",size = 1.2)+  
 geom\_vline(xintercept = max\_time, color = "#00BFC4",size = 2, alpha = 0.3)+  
 geom\_text(aes(label = paste("Maximum:" ,format(round(max,2),big.mark = ",")), x = max\_time, y = 200),hjust = 1 )+  
 geom\_text(aes(label = format(max\_time, "%H:%M"), x = max\_time, y = 0),size = 3.5,color = "black")+  
 scale\_x\_datetime(date\_labels = "%H:%M")+  
 labs(title = "Average Daily Activity Pattern")+  
 ylab("Average # of steps taken averaged across all days")

 ## Imputing missing values

From the summary result in the 1st code chunk, we know that there’s 2304 observation missing “steps”. 2. I will use average mean of 5 minutes interval, which is 37.3826, to fill missing value.

mean\_steps <- mean(data$steps,na.rm = T)  
print(paste0("Average steps of each time interval: ",round(mean\_steps,2)))

## [1] "Average steps of each time interval: 37.38"

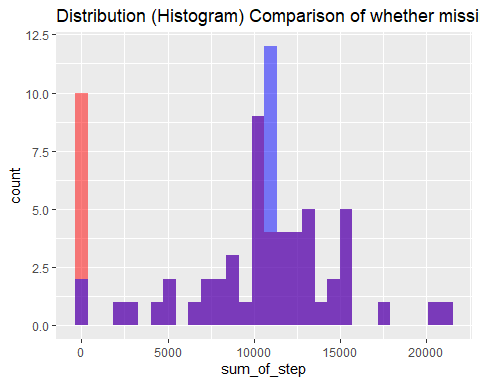
new\_data <- data  
new\_data$steps[is.na(new\_data$steps)] <- mean\_steps

1. As the assignment instructed, I re-plot histogram of new data(with missing value imputed with average.)

* However, instead of plotting new histogram alone, I would like to overlap two histogram (new v.s old) to have a better understanding changes in distribution resulted from missing value imputing.
* Red histogram represent old data set, blue represent new data set. In addition, I set alpha equal 0.5, thus overlapping part would be purple.

df\_q3 <- new\_data %>%  
 group\_by(date) %>%  
 summarise(sum\_of\_step = sum(steps,na.rm = T))  
  
seq <- rep(1:2, each = 61)  
  
df\_q3 <- cbind(rbind(df\_q1,df\_q3),seq)  
df\_q3 <- mutate(df\_q3,seq = recode\_factor(seq,"1" = "No Imputing", "2" = "Imputing"))  
  
  
ggplot(data = df\_q3, aes(x = sum\_of\_step)) +  
 geom\_histogram(data = subset(df\_q3, seq =="No Imputing"),fill = "red",alpha = 0.5)+  
 geom\_histogram(data = subset(df\_q3, seq =="Imputing"),fill = "blue",alpha = 0.5)+  
 labs(title = "Distribution (Histogram) Comparison of whether missing value imputing or not")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

 From above graph, we can noticed that most part of distribution (Purple part.) doesn’t change at all but huge decrease of frequency of 0 steps per day, and sudden increase of frequency of 11,000 steps per day.

It clearing indicating that missing value are concentrated in certain date!(If it’s evenly or randomly distributed in different dates, we supposed to see entire distribution shifting rightward.) As a matter of fact, we can use table to confirm my thoughts. and indeed NA value concentrated in 8 days, and these 8 days are full of NA value.

table(data$date[is.na(data$steps)])

##   
## 2012-10-01 2012-10-08 2012-11-01 2012-11-04 2012-11-09 2012-11-10 2012-11-14   
## 288 288 288 288 288 288 288   
## 2012-11-30   
## 288

Let’s see how mean and median of total number of steps taken per day changes after imputing missing value with average.

table <- df\_q3 %>%  
 group\_by(seq) %>%  
 summarise(Mean = mean(sum\_of\_step),Median = median(sum\_of\_step))  
table

## # A tibble: 2 x 3  
## seq Mean Median  
## <fct> <dbl> <dbl>  
## 1 No Imputing 9354. 10395   
## 2 Imputing 10766. 10766.

We can see that both mean and median increase by 1,400, and mean and median becomes the same value (10766.19).

## Are there differences in activity patterns between weekdays and weekends?

1. To analyze if there’s differences in activity pattern between weekdays and weekends, let first create a new variable differentiate weekend from weekdays.

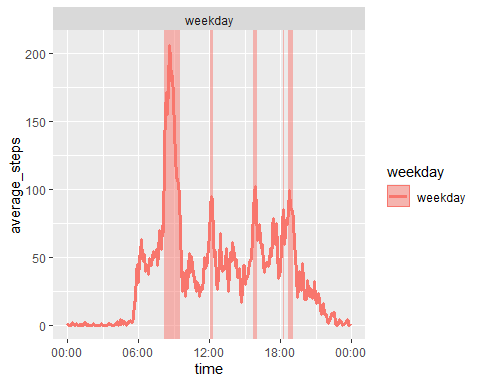
data <- data %>%  
 mutate(weekday = weekdays(date)) %>%  
 mutate(weekday = as.factor(weekday %in% c('Saturday','Sunday'))) %>%  
 mutate(weekday = recode\_factor(weekday,"TRUE" = "weekend","FALSE" = "weekday"))

1. Secondly, we will plot 2 time series of the 5-minute interval and average number of steps take, averaged across all weekday days or weekend days. (Noted that missing value are ignored in the analysis because of concentrated missing value.)
2. In addition to assignment request, I would like to visualize which time interval are the subject most active. I define active time interval as the interval when average steps taken is equal or above 90th percentile of average steps taken per interval.

df\_q4 <- data %>%  
 group\_by(time,weekday) %>%  
 summarise(average\_steps = mean(steps,na.rm = T))%>%  
 ungroup() %>%  
 group\_by(weekday) %>%  
 mutate(active\_interval = (average\_steps >= quantile(average\_steps,probs = 0.9)))

## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.

weekday\_int\_df <- df\_q4 %>%  
 filter(active\_interval == T) %>%  
 mutate(upper = time + 60\*5)  
  
  
ggplot(df\_q4,aes(time,average\_steps,fill = weekday,color = weekday))+  
 geom\_line(size = 1.2)+  
 geom\_rect(data = weekday\_int\_df,  
 aes( x = NULL, y = NULL, xmin = time, xmax = upper, ymin = -Inf,ymax = Inf,color = NULL),   
 alpha = 0.5  
 )+  
 facet\_wrap(.~weekday,nrow = 2)+  
 scale\_x\_datetime(date\_labels = ("%H:%M"))



At last, we can see from the plot that the most active interval in weekday centered at 7 a.m. and 7 p.m. , which is probably the subject’s commute time to school/office. And in weekend, active time intervals are more scattered throughout the day. There’re other interesting active time interval worth more investigation. (ex: why the subject are active around 4 p.m both in weekend and weekday.), but that’s beyond the scope of assignment and maybe required information more than the data set.