Reproducible Research: Peer Assessment 1

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)

## Warning: package 'ggplot2' was built under R version 3.4.3

library(gridExtra)

## Warning: package 'gridExtra' was built under R version 3.4.3

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

## Loading and preprocessing the data

temp <- tempfile()  
temp2 <-tempfile()  
  
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", temp)  
unzip(temp,list=TRUE)

## Name Length Date  
## 1 activity.csv 350829 2014-02-11 10:08:00

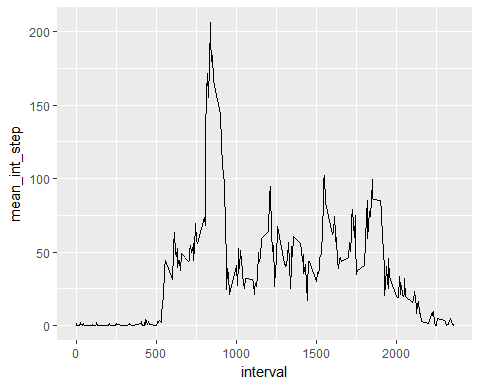
unzip(zipfile=temp, exdir=temp2)  
  
input\_data <- read.csv(file.path(temp2,'activity.csv',fsep="\\"))

|  |
| --- |
| ## What is mean total number of steps taken per day? |
| 1. Calculate the total number of steps taken per day 2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day 3. Calculate and report the mean and median of the total number of steps taken per day |
| ```r Total\_Daily\_Steps <- input\_data %>% group\_by(date) %>% summarize(sum\_steps = sum(steps, na.rm=TRUE)) |
| mean\_daily = round(mean(Total\_Daily\_Stepssum\_steps),0) |
| #qplot(data=Total\_Daily\_Steps, x=sum\_steps) |
| ggplot(Total\_Daily\_Steps,aes(sum\_steps)) + geom\_histogram() + geom\_vline(xintercept = mean\_daily, color=‘red’) + geom\_text(aes(x=mean\_daily, y=10), label=“mean”, color=‘red’, size=3.5, angle=90, hjust=1, vjust=1) + geom\_vline(xintercept = median\_daily, color=‘blue’) + geom\_text(aes(x=median\_daily, y=10),label=“median”, color=‘blue’,size=3.5,angle=90, hjust=1, vjust=1) ``` |
| ## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`. |
|  |
| r #ggplot(Total\_Daily\_Steps,aes(y=sum\_steps, x=date)) + geom\_bar(stat="identity") |
| The mean number of steps taken per day is 9354 The median number of steps taken per day is 1.039510^{4} |

## What is the average daily activity pattern?

1. Make a time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)
2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

Total\_Daily\_Steps$date <- ymd(Total\_Daily\_Steps$date)  
  
  
interval\_mean <- input\_data %>%  
 group\_by(interval)%>%  
 summarize(mean\_int\_step = mean(steps, na.rm=TRUE))  
  
ggplot(interval\_mean, aes(interval, mean\_int\_step)) + geom\_line()



max\_step = interval\_mean[which.max(interval\_mean$mean\_int\_step),'interval']

At interval 835, on average across all the days in the dataset, contains the maximum number of steps.

|  |
| --- |
| ## Imputing missing values 1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs) |
| 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. |
| 3. Create a new dataset that is equal to the original dataset but with the missing data filled in. |
| 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? |
| ```r missing\_cnt = sum(is.na(input\_data)) |
| input\_data2 <- input\_data |
| # fill missing data by the mean of interval for (i in seq(nrow(input\_data2))) { if (is.na(input\_data2[i,‘steps’])) { input\_data2mean\_int\_step[which(interval\_meaninterval[i])] } } |
| Total\_Daily\_Steps2 <- input\_data2 %>% group\_by(date) %>% summarize(sum\_steps = sum(steps, na.rm=TRUE)) |
| mean\_daily2 = round(mean(Total\_Daily\_Steps2sum\_steps),0) |
| #par(mfrow=c(1,2)) p1 <- ggplot(Total\_Daily\_Steps,aes(sum\_steps)) + geom\_histogram() + expand\_limits(y=c(0,15)) + geom\_vline(xintercept = mean\_daily, color=‘red’) + geom\_text(aes(x=mean\_daily, y=15), label=“mean”, color=‘red’, size=3.5, angle=90, hjust=1, vjust=1) + geom\_vline(xintercept = median\_daily, color=‘blue’) + geom\_text(aes(x=median\_daily, y=15),label=“median”, color=‘blue’,size=3.5,angle=90, hjust=1, vjust=1) + ggtitle(“Original Data”) |
| p2<-ggplot(Total\_Daily\_Steps2,aes(sum\_steps)) + geom\_histogram() + expand\_limits(y=c(0,15)) + geom\_vline(xintercept = mean\_daily2, color=‘red’) + geom\_text(aes(x=mean\_daily2, y=15), label=“mean”, color=‘red’, size=3.5, angle=90, hjust=1, vjust=1) + geom\_vline(xintercept = median\_daily2, color=‘blue’) + geom\_text(aes(x=median\_daily2, y=15),label=“median”, color=‘blue’,size=3.5,angle=90, hjust=1, vjust=1) + ggtitle(“Imputed Data”) |
| grid.arrange(p1, p2, ncol=2) ``` |
| ## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`. ## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`. |
| The missing values have impact to the data. |

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

input\_data <- input\_data %>%  
 mutate(day = wday(as.Date(input\_data$date, "%Y-%m-%d"))) %>%  
 mutate(weekdays = day<6)  
  
# split the data by weekdays and weekends, then do a time series   
  
# weekday activity #  
weekday = input\_data %>%  
 filter(input\_data$weekdays == TRUE)  
  
weekday\_interval\_mean <- weekday %>%  
 group\_by(interval)%>%  
 summarize(mean\_int\_step = mean(steps, na.rm=TRUE))  
  
# weekend activity #  
weekend = input\_data %>%  
 filter(input\_data$weekdays == FALSE)  
  
weekend\_interval\_mean <- weekend %>%  
 group\_by(interval)%>%  
 summarize(mean\_int\_step = mean(steps, na.rm=TRUE))  
  
  
p1 <- ggplot(weekday\_interval\_mean, aes(interval, mean\_int\_step)) + geom\_line() + expand\_limits(y=c(0,300)) + ggtitle("Weekday Activity")  
p2 <- ggplot(weekend\_interval\_mean, aes(interval, mean\_int\_step)) + geom\_line() +expand\_limits(y=c(0,300)) + ggtitle("Weekend Activity")  
grid.arrange(p1, p2, ncol=2)

