Course Project 1 of Reproducible Research

Saran

3/28/2022

## Loading and preprocessing the data

Unzip data to obtain a csv file.

library("data.table")

## Warning: package 'data.table' was built under R version 4.1.1

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.1.2

## Reading csv Data into Data.Table.

activity <- data.table::fread(input = "data/activity.csv")

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

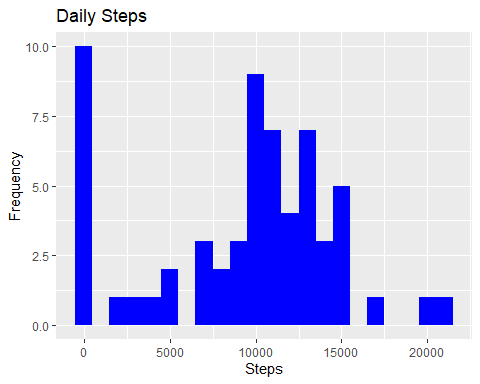
1. Calculate the total number of steps taken per day

Total\_Steps <- activity[, c(lapply(.SD, sum, na.rm = FALSE)), .SDcols = c("steps"), by = .(date)]   
  
head(Total\_Steps, 10)

## date steps  
## 1: 10/1/2012 0  
## 2: 10/2/2012 126  
## 3: 10/3/2012 11352  
## 4: 10/4/2012 12116  
## 5: 10/5/2012 13294  
## 6: 10/6/2012 15420  
## 7: 10/7/2012 11015  
## 8: 10/8/2012 0  
## 9: 10/9/2012 12811  
## 10: 10/10/2012 9900

1. Make a histogram of the total number of steps taken each day.

ggplot(Total\_Steps, aes(x = steps)) +  
 geom\_histogram(fill = "blue", binwidth = 1000) +  
 labs(title = "Daily Steps", x = "Steps", y = "Frequency")



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

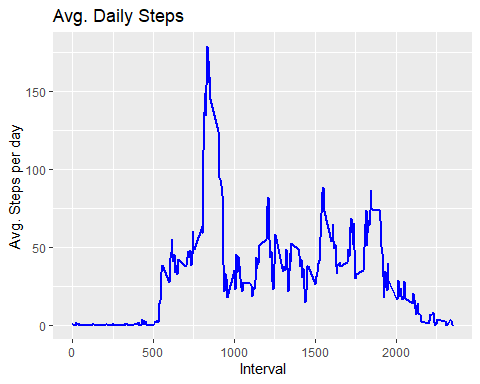
Total\_Steps[, .(Mean\_Steps = mean(steps, na.rm = TRUE), Median\_Steps = median(steps, na.rm = TRUE))]

## Mean\_Steps Median\_Steps  
## 1: 9354.23 10395

## 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)

Interval <- activity[, c(lapply(.SD, mean, na.rm = TRUE)), .SDcols = c("steps"), by = .(interval)]   
ggplot(Interval, aes(x = interval , y = steps)) + geom\_line(color="blue", size=1) + labs(title = "Avg. Daily Steps", x = "Interval", y = "Avg. Steps per day")



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

Interval[steps == max(steps), .(max\_interval = interval)]

## max\_interval  
## 1: 835

## 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)

activity[is.na(steps), .N ]

## [1] 0

1. 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.

# Filling in missing values with median of dataset.   
activity[is.na(steps), "steps"] <- activity[, c(lapply(.SD, median, na.rm = TRUE)), .SDcols = c("steps")]

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

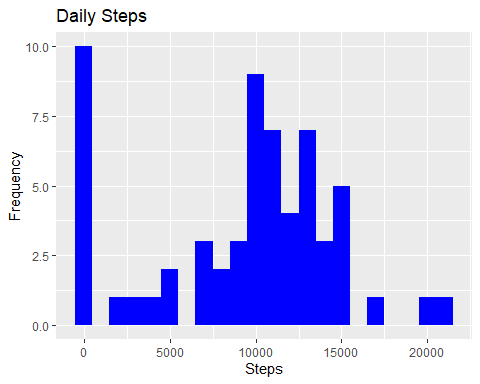
data.table::fwrite(x = activity, file = "data/activity.csv", quote = FALSE)

1. 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?

# total number of steps taken per day  
Total\_Steps <- activity[, c(lapply(.SD, sum)), .SDcols = c("steps"), by = .(date)]   
# mean and median total number of steps taken per day  
Total\_Steps[, .(Mean\_Steps = mean(steps), Median\_Steps = median(steps))]

## Mean\_Steps Median\_Steps  
## 1: 9354.23 10395

ggplot(Total\_Steps, aes(x = steps)) + geom\_histogram(fill = "blue", binwidth = 1000) + labs(title = "Daily Steps", x = "Steps", y = "Frequency")

 First Part (with na) | Mean = 10765 | Median = 10765 Second Part (filling na with median) | Mean = 9354.23 |Median = 10395

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

1. 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.

activity$date <- as.Date(strptime(activity$date, format="%Y-%m-%d"))  
activity$day <- weekdays(activity$date)  
for (i in 1:nrow(activity)) {  
 if (activity[i,]$day %in% c("Saturday","Sunday")) {  
 activity[i,]$day<-"weekend"  
 }  
 else{  
 activity[i,]$day<-"weekday"  
 }  
}  
stepsByDay <- aggregate(activity$steps ~ activity$interval + activity$day, activity, mean)

1. Make a panel plot containing 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 weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

names(stepsByDay) <- c("interval", "day", "steps")  
library(lattice)  
xyplot(steps ~ interval | day, stepsByDay, type = "l", layout = c(1, 2),   
 xlab = "Interval", ylab = "Number of steps")

