Week 2 Project 1

Rod Jones (rwjones97)

5/4/2020

## R Markdown

## Loading the files and exploring the data

library(“data.table”) library(ggplot2)

library("data.table")  
library(ggplot2)  
  
fileUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"  
  
download.file(fileUrl, destfile = paste0(getwd(), '/repdata%2Fdata%2Factivity.zip'), method = "curl")  
  
unzip("repdata%2Fdata%2Factivity.zip",exdir = "data")

% Total % Received % Xferd Average Speed Time Time Time Current Dload Upload Total Spent Left Speed 100 53559 100 53559 0 0 53559 0 0:00:01 –:–:– 0:00:01 83555

## Reading csv Data into Data.Table.

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

## Question: Average total number of steps taken per day

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

## date steps  
## 1: 2012-10-01 NA  
## 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 NA  
## 9: 2012-10-09 12811  
## 10: 2012-10-10 9900

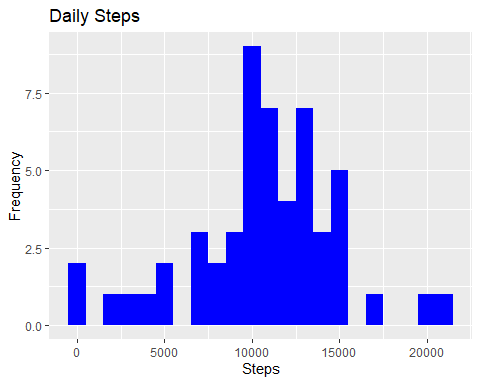
date steps

1: 2012-10-01 NA 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 NA 9: 2012-10-09 12811 10: 2012-10-10 9900

## Question: 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")

## Warning: Removed 8 rows containing non-finite values (stat\_bin).



## Question: Calculate and report the mean and median of the total number of steps taken per day

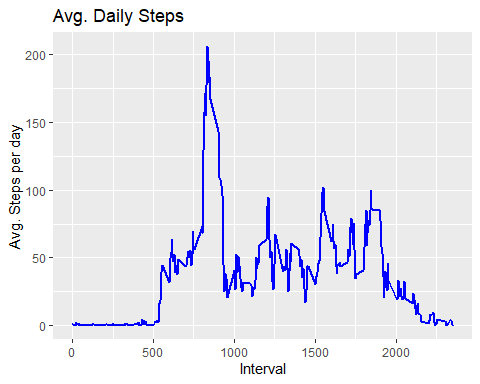
Total\_Steps[, .(Mean\_Steps = mean(steps, na.rm = TRUE), Median\_Steps = median(steps, na.rm = TRUE))]

## Mean\_Steps Median\_Steps  
## 1: 10766.19 10765

Mean\_Steps Median\_Steps 1: 10766.19 10765

## Question: 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)

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



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

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

## max\_interval  
## 1: 835

max\_interval 1: 835

## Question: Calculate and report the total number of missing values in the dataset

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

## [1] 2304

[1] 2304

## Question: 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

activityDT[is.na(steps), "steps"] <- activityDT[, c(lapply(.SD, median, na.rm = TRUE)), .SDcols = c("steps")]

## Question: Create a new dataset that is equal to the original dataset but with the missing data filled in

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

## Question: 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 <- activityDT[, c(lapply(.SD, sum)), .SDcols = c("steps"), by = .(date)]

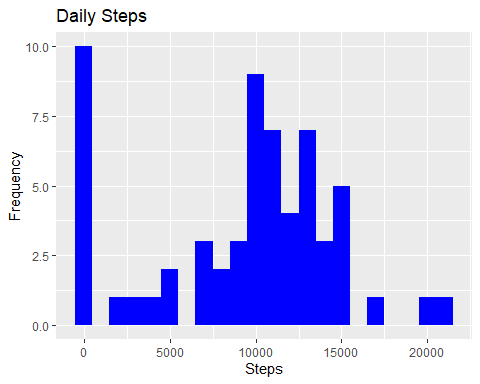
## mean and median total number of steps taken per day

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

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

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")



## Question: 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.

activityDT <- data.table::fread(input = "data/activity.csv")  
activityDT[, date := as.POSIXct(date, format = "%Y-%m-%d")]  
activityDT[, `Day of Week`:= weekdays(x = date)]  
activityDT[grepl(pattern = "Monday|Tuesday|Wednesday|Thursday|Friday", x = `Day of Week`), "weekday or weekend"] <- "weekday"  
activityDT[grepl(pattern = "Saturday|Sunday", x = `Day of Week`), "weekday or weekend"] <- "weekend"  
activityDT[, `weekday or weekend` := as.factor(`weekday or weekend`)]

head(activityDT, 10) steps date interval Day of Week weekday or weekend 1: NA 2012-10-01 0 Monday weekday 2: NA 2012-10-01 5 Monday weekday 3: NA 2012-10-01 10 Monday weekday 4: NA 2012-10-01 15 Monday weekday 5: NA 2012-10-01 20 Monday weekday 6: NA 2012-10-01 25 Monday weekday 7: NA 2012-10-01 30 Monday weekday 8: NA 2012-10-01 35 Monday weekday 9: NA 2012-10-01 40 Monday weekday 10: NA 2012-10-01 45 Monday weekday

## Question: 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.

activityDT[is.na(steps), "steps"] <- activityDT[, c(lapply(.SD, median, na.rm = TRUE)), .SDcols = c("steps")]  
IntervalDT <- activityDT[, c(lapply(.SD, mean, na.rm = TRUE)), .SDcols = c("steps"), by = .(interval, `weekday or weekend`)]   
   
ggplot(IntervalDT , aes(x = interval , y = steps, color=`weekday or weekend`)) + geom\_line() + labs(title = "Avg. Daily Steps by Weektype", x = "Interval", y = "No. of Steps") + facet\_wrap(~`weekday or weekend` , ncol = 1, nrow=2)

