Project 1

Roshan

10/26/2020

## Loading and preprocessing the data

### 1. Load the data

activity <- read.csv("activity.csv")

### 2. Process/transform the data

activity$date <- as.POSIXct(activity$date, format="%Y-%m-%d")

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

### 1.

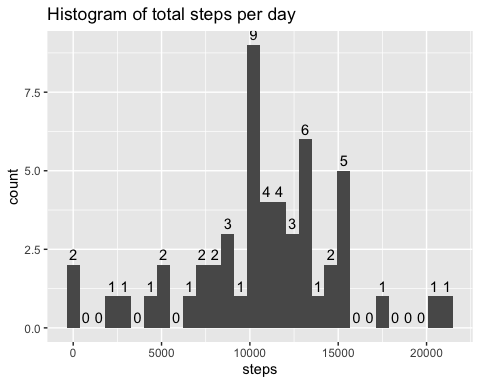
activity\_clean <- activity[complete.cases(activity),]  
total\_steps\_per\_day <- aggregate(activity\_clean["steps"],   
 list(days=cut(activity\_clean$date, "1 days")),  
 sum)  
total\_steps\_per\_day

## days steps  
## 1 2012-10-02 00:00:00 126  
## 2 2012-10-03 00:00:00 11352  
## 3 2012-10-04 00:00:00 12116  
## 4 2012-10-05 00:00:00 13294  
## 5 2012-10-06 00:00:00 15420  
## 6 2012-10-07 00:00:00 11015  
## 7 2012-10-09 00:00:00 12811  
## 8 2012-10-10 00:00:00 9900  
## 9 2012-10-11 00:00:00 10304  
## 10 2012-10-12 00:00:00 17382  
## 11 2012-10-13 00:00:00 12426  
## 12 2012-10-14 00:00:00 15098  
## 13 2012-10-15 00:00:00 10139  
## 14 2012-10-16 00:00:00 15084  
## 15 2012-10-17 00:00:00 13452  
## 16 2012-10-18 00:00:00 10056  
## 17 2012-10-19 00:00:00 11829  
## 18 2012-10-20 00:00:00 10395  
## 19 2012-10-21 00:00:00 8821  
## 20 2012-10-22 00:00:00 13460  
## 21 2012-10-23 00:00:00 8918  
## 22 2012-10-24 00:00:00 8355  
## 23 2012-10-25 00:00:00 2492  
## 24 2012-10-26 00:00:00 6778  
## 25 2012-10-27 00:00:00 10119  
## 26 2012-10-28 00:00:00 11458  
## 27 2012-10-29 00:00:00 5018  
## 28 2012-10-30 00:00:00 9819  
## 29 2012-10-31 00:00:00 15414  
## 30 2012-11-02 00:00:00 10600  
## 31 2012-11-03 00:00:00 10571  
## 32 2012-11-04 23:00:00 10439  
## 33 2012-11-05 23:00:00 8334  
## 34 2012-11-06 23:00:00 12883  
## 35 2012-11-07 23:00:00 3219  
## 36 2012-11-10 23:00:00 12608  
## 37 2012-11-11 23:00:00 10765  
## 38 2012-11-12 23:00:00 7336  
## 39 2012-11-14 23:00:00 41  
## 40 2012-11-15 23:00:00 5441  
## 41 2012-11-16 23:00:00 14339  
## 42 2012-11-17 23:00:00 15110  
## 43 2012-11-18 23:00:00 8841  
## 44 2012-11-19 23:00:00 4472  
## 45 2012-11-20 23:00:00 12787  
## 46 2012-11-21 23:00:00 20427  
## 47 2012-11-22 23:00:00 21194  
## 48 2012-11-23 23:00:00 14478  
## 49 2012-11-24 23:00:00 11834  
## 50 2012-11-25 23:00:00 11162  
## 51 2012-11-26 23:00:00 13646  
## 52 2012-11-27 23:00:00 10183  
## 53 2012-11-28 23:00:00 7047

### 2.

library(ggplot2)  
  
g <- ggplot(total\_steps\_per\_day, aes(x = steps))  
g <- g + geom\_histogram() + stat\_bin(aes(y=..count.., label=..count..),   
 geom="text", vjust=-.5)  
g + ggtitle("Histogram of total steps per day")

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



### 3.

average\_steps\_per\_day <- aggregate(activity\_clean["steps"],   
 list(days=cut(activity\_clean$date, "1 days")),  
 mean)  
average\_steps\_per\_day

## days steps  
## 1 2012-10-02 00:00:00 0.4375000  
## 2 2012-10-03 00:00:00 39.4166667  
## 3 2012-10-04 00:00:00 42.0694444  
## 4 2012-10-05 00:00:00 46.1597222  
## 5 2012-10-06 00:00:00 53.5416667  
## 6 2012-10-07 00:00:00 38.2465278  
## 7 2012-10-09 00:00:00 44.4826389  
## 8 2012-10-10 00:00:00 34.3750000  
## 9 2012-10-11 00:00:00 35.7777778  
## 10 2012-10-12 00:00:00 60.3541667  
## 11 2012-10-13 00:00:00 43.1458333  
## 12 2012-10-14 00:00:00 52.4236111  
## 13 2012-10-15 00:00:00 35.2048611  
## 14 2012-10-16 00:00:00 52.3750000  
## 15 2012-10-17 00:00:00 46.7083333  
## 16 2012-10-18 00:00:00 34.9166667  
## 17 2012-10-19 00:00:00 41.0729167  
## 18 2012-10-20 00:00:00 36.0937500  
## 19 2012-10-21 00:00:00 30.6284722  
## 20 2012-10-22 00:00:00 46.7361111  
## 21 2012-10-23 00:00:00 30.9652778  
## 22 2012-10-24 00:00:00 29.0104167  
## 23 2012-10-25 00:00:00 8.6527778  
## 24 2012-10-26 00:00:00 23.5347222  
## 25 2012-10-27 00:00:00 35.1354167  
## 26 2012-10-28 00:00:00 39.7847222  
## 27 2012-10-29 00:00:00 17.4236111  
## 28 2012-10-30 00:00:00 34.0937500  
## 29 2012-10-31 00:00:00 53.5208333  
## 30 2012-11-02 00:00:00 36.8055556  
## 31 2012-11-03 00:00:00 36.7048611  
## 32 2012-11-04 23:00:00 36.2465278  
## 33 2012-11-05 23:00:00 28.9375000  
## 34 2012-11-06 23:00:00 44.7326389  
## 35 2012-11-07 23:00:00 11.1770833  
## 36 2012-11-10 23:00:00 43.7777778  
## 37 2012-11-11 23:00:00 37.3784722  
## 38 2012-11-12 23:00:00 25.4722222  
## 39 2012-11-14 23:00:00 0.1423611  
## 40 2012-11-15 23:00:00 18.8923611  
## 41 2012-11-16 23:00:00 49.7881944  
## 42 2012-11-17 23:00:00 52.4652778  
## 43 2012-11-18 23:00:00 30.6979167  
## 44 2012-11-19 23:00:00 15.5277778  
## 45 2012-11-20 23:00:00 44.3993056  
## 46 2012-11-21 23:00:00 70.9270833  
## 47 2012-11-22 23:00:00 73.5902778  
## 48 2012-11-23 23:00:00 50.2708333  
## 49 2012-11-24 23:00:00 41.0902778  
## 50 2012-11-25 23:00:00 38.7569444  
## 51 2012-11-26 23:00:00 47.3819444  
## 52 2012-11-27 23:00:00 35.3576389  
## 53 2012-11-28 23:00:00 24.4687500

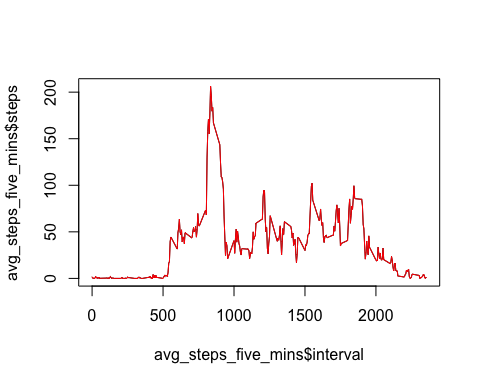
median\_steps\_per\_day <- aggregate(activity\_clean["steps"],   
 list(days=cut(activity\_clean$date, "1 days")),  
 median)  
median\_steps\_per\_day

## days steps  
## 1 2012-10-02 00:00:00 0  
## 2 2012-10-03 00:00:00 0  
## 3 2012-10-04 00:00:00 0  
## 4 2012-10-05 00:00:00 0  
## 5 2012-10-06 00:00:00 0  
## 6 2012-10-07 00:00:00 0  
## 7 2012-10-09 00:00:00 0  
## 8 2012-10-10 00:00:00 0  
## 9 2012-10-11 00:00:00 0  
## 10 2012-10-12 00:00:00 0  
## 11 2012-10-13 00:00:00 0  
## 12 2012-10-14 00:00:00 0  
## 13 2012-10-15 00:00:00 0  
## 14 2012-10-16 00:00:00 0  
## 15 2012-10-17 00:00:00 0  
## 16 2012-10-18 00:00:00 0  
## 17 2012-10-19 00:00:00 0  
## 18 2012-10-20 00:00:00 0  
## 19 2012-10-21 00:00:00 0  
## 20 2012-10-22 00:00:00 0  
## 21 2012-10-23 00:00:00 0  
## 22 2012-10-24 00:00:00 0  
## 23 2012-10-25 00:00:00 0  
## 24 2012-10-26 00:00:00 0  
## 25 2012-10-27 00:00:00 0  
## 26 2012-10-28 00:00:00 0  
## 27 2012-10-29 00:00:00 0  
## 28 2012-10-30 00:00:00 0  
## 29 2012-10-31 00:00:00 0  
## 30 2012-11-02 00:00:00 0  
## 31 2012-11-03 00:00:00 0  
## 32 2012-11-04 23:00:00 0  
## 33 2012-11-05 23:00:00 0  
## 34 2012-11-06 23:00:00 0  
## 35 2012-11-07 23:00:00 0  
## 36 2012-11-10 23:00:00 0  
## 37 2012-11-11 23:00:00 0  
## 38 2012-11-12 23:00:00 0  
## 39 2012-11-14 23:00:00 0  
## 40 2012-11-15 23:00:00 0  
## 41 2012-11-16 23:00:00 0  
## 42 2012-11-17 23:00:00 0  
## 43 2012-11-18 23:00:00 0  
## 44 2012-11-19 23:00:00 0  
## 45 2012-11-20 23:00:00 0  
## 46 2012-11-21 23:00:00 0  
## 47 2012-11-22 23:00:00 0  
## 48 2012-11-23 23:00:00 0  
## 49 2012-11-24 23:00:00 0  
## 50 2012-11-25 23:00:00 0  
## 51 2012-11-26 23:00:00 0  
## 52 2012-11-27 23:00:00 0  
## 53 2012-11-28 23:00:00 0

## What is the average daily activity pattern?

### 1.

avg\_steps\_five\_mins <- aggregate(steps ~ interval, data = activity\_clean, mean)  
  
plot(avg\_steps\_five\_mins$interval, avg\_steps\_five\_mins$steps, type="l")  
lines(avg\_steps\_five\_mins$interval, avg\_steps\_five\_mins$steps, col="red")



### 2. Which 5-minute interval, on average across all the days in the dataset,

contains the maximum number of steps?

avg\_steps\_five\_mins$interval[which.max(avg\_steps\_five\_mins$steps)]

## [1] 835

835, 5-minute interval, on average across all the days in the data set, contains the maximum number of steps.

## Imputing missing values

### 1.

sum(is.na(activity))

## [1] 2304

Total number of missing rows: 2304

### 2.

Filling missing values with the average value of the day.

### 3.

fill\_missing <- function(df1, df2) {  
 n <- nrow(df1)   
 new\_df <- df1  
  
 for(i in 1:ncol(df1)){  
 new\_df[is.na(df1[,"steps"]), "steps"] <- mean(df1[,"steps"], na.rm = TRUE)  
 }  
   
 return(new\_df)  
}  
  
average\_steps\_per\_day$days <- as.POSIXct(average\_steps\_per\_day$days, format="%Y-%m-%d")  
activity\_new <- fill\_missing(activity, average\_steps\_per\_day)  
  
sum(is.na(activity\_new))

## [1] 0

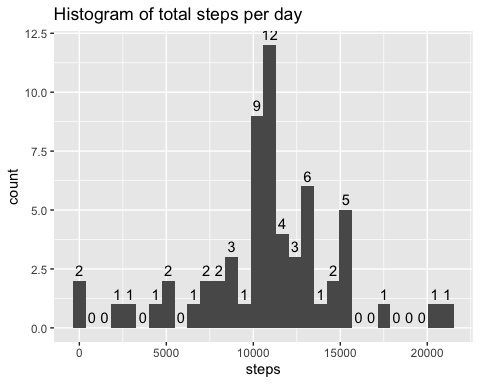
### 4.

total\_steps\_per\_day <- aggregate(activity\_new["steps"],   
 list(days=cut(activity\_new$date, "1 days")),  
 sum)  
head(total\_steps\_per\_day)

## days steps  
## 1 2012-10-01 00:00:00 10766.19  
## 2 2012-10-02 00:00:00 126.00  
## 3 2012-10-03 00:00:00 11352.00  
## 4 2012-10-04 00:00:00 12116.00  
## 5 2012-10-05 00:00:00 13294.00  
## 6 2012-10-06 00:00:00 15420.00

g <- ggplot(total\_steps\_per\_day, aes(x = steps))  
g <- g + geom\_histogram() + stat\_bin(aes(y=..count.., label=..count..),   
 geom="text", vjust=-.5)  
g + ggtitle("Histogram of total steps per day")

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



average\_steps\_per\_day <- aggregate(activity\_new["steps"],   
 list(days=cut(activity\_new$date, "1 days")),  
 mean)  
head(average\_steps\_per\_day)

## days steps  
## 1 2012-10-01 00:00:00 37.38260  
## 2 2012-10-02 00:00:00 0.43750  
## 3 2012-10-03 00:00:00 39.41667  
## 4 2012-10-04 00:00:00 42.06944  
## 5 2012-10-05 00:00:00 46.15972  
## 6 2012-10-06 00:00:00 53.54167

median\_steps\_per\_day <- aggregate(activity\_new["steps"],   
 list(days=cut(activity\_new$date, "1 days")),  
 median)  
median\_steps\_per\_day

## days steps  
## 1 2012-10-01 00:00:00 37.3826  
## 2 2012-10-02 00:00:00 0.0000  
## 3 2012-10-03 00:00:00 0.0000  
## 4 2012-10-04 00:00:00 0.0000  
## 5 2012-10-05 00:00:00 0.0000  
## 6 2012-10-06 00:00:00 0.0000  
## 7 2012-10-07 00:00:00 0.0000  
## 8 2012-10-08 00:00:00 37.3826  
## 9 2012-10-09 00:00:00 0.0000  
## 10 2012-10-10 00:00:00 0.0000  
## 11 2012-10-11 00:00:00 0.0000  
## 12 2012-10-12 00:00:00 0.0000  
## 13 2012-10-13 00:00:00 0.0000  
## 14 2012-10-14 00:00:00 0.0000  
## 15 2012-10-15 00:00:00 0.0000  
## 16 2012-10-16 00:00:00 0.0000  
## 17 2012-10-17 00:00:00 0.0000  
## 18 2012-10-18 00:00:00 0.0000  
## 19 2012-10-19 00:00:00 0.0000  
## 20 2012-10-20 00:00:00 0.0000  
## 21 2012-10-21 00:00:00 0.0000  
## 22 2012-10-22 00:00:00 0.0000  
## 23 2012-10-23 00:00:00 0.0000  
## 24 2012-10-24 00:00:00 0.0000  
## 25 2012-10-25 00:00:00 0.0000  
## 26 2012-10-26 00:00:00 0.0000  
## 27 2012-10-27 00:00:00 0.0000  
## 28 2012-10-28 00:00:00 0.0000  
## 29 2012-10-29 00:00:00 0.0000  
## 30 2012-10-30 00:00:00 0.0000  
## 31 2012-10-31 00:00:00 0.0000  
## 32 2012-11-01 00:00:00 37.3826  
## 33 2012-11-02 00:00:00 0.0000  
## 34 2012-11-03 00:00:00 0.0000  
## 35 2012-11-04 00:00:00 37.3826  
## 36 2012-11-04 23:00:00 0.0000  
## 37 2012-11-05 23:00:00 0.0000  
## 38 2012-11-06 23:00:00 0.0000  
## 39 2012-11-07 23:00:00 0.0000  
## 40 2012-11-08 23:00:00 37.3826  
## 41 2012-11-09 23:00:00 37.3826  
## 42 2012-11-10 23:00:00 0.0000  
## 43 2012-11-11 23:00:00 0.0000  
## 44 2012-11-12 23:00:00 0.0000  
## 45 2012-11-13 23:00:00 37.3826  
## 46 2012-11-14 23:00:00 0.0000  
## 47 2012-11-15 23:00:00 0.0000  
## 48 2012-11-16 23:00:00 0.0000  
## 49 2012-11-17 23:00:00 0.0000  
## 50 2012-11-18 23:00:00 0.0000  
## 51 2012-11-19 23:00:00 0.0000  
## 52 2012-11-20 23:00:00 0.0000  
## 53 2012-11-21 23:00:00 0.0000  
## 54 2012-11-22 23:00:00 0.0000  
## 55 2012-11-23 23:00:00 0.0000  
## 56 2012-11-24 23:00:00 0.0000  
## 57 2012-11-25 23:00:00 0.0000  
## 58 2012-11-26 23:00:00 0.0000  
## 59 2012-11-27 23:00:00 0.0000  
## 60 2012-11-28 23:00:00 0.0000  
## 61 2012-11-29 23:00:00 37.3826

The imputed dates now have different median values, also the mean of each day has changed after imputing missing values.

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

### 1.

activity$weekday <- ifelse(weekdays(activity$date) == "Saturday" | weekdays(activity$date) == "Sunday", "Weekend", "Weekday")

### 2.

g <- ggplot(activity, aes(x = interval, y = steps))  
g <- g + geom\_line() + facet\_grid(vars(weekday))  
g

## Warning: Removed 1 row(s) containing missing values (geom\_path).

