

Reproducible Research Project 1

Luis Felipe

8/7/2020

R Markdown

This is an R Markdown document. For accomplishing the project 1 of reproducible research.

```
# download file from web
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", destfile = "activity.zip")
# unzip data and read
unzip("activity.zip")
stepdata <- read.csv("activity.csv", header = TRUE)
head(stepdata)
```

```
##      steps      date interval
## 1      NA 2012-10-01         0
## 2      NA 2012-10-01         5
## 3      NA 2012-10-01        10
## 4      NA 2012-10-01        15
## 5      NA 2012-10-01        20
## 6      NA 2012-10-01        25
```

Including Plots

Then the calculation of the total number of steps is done

```
library(magrittr)
library(dplyr)

##
## Attaching package: 'dplyr'

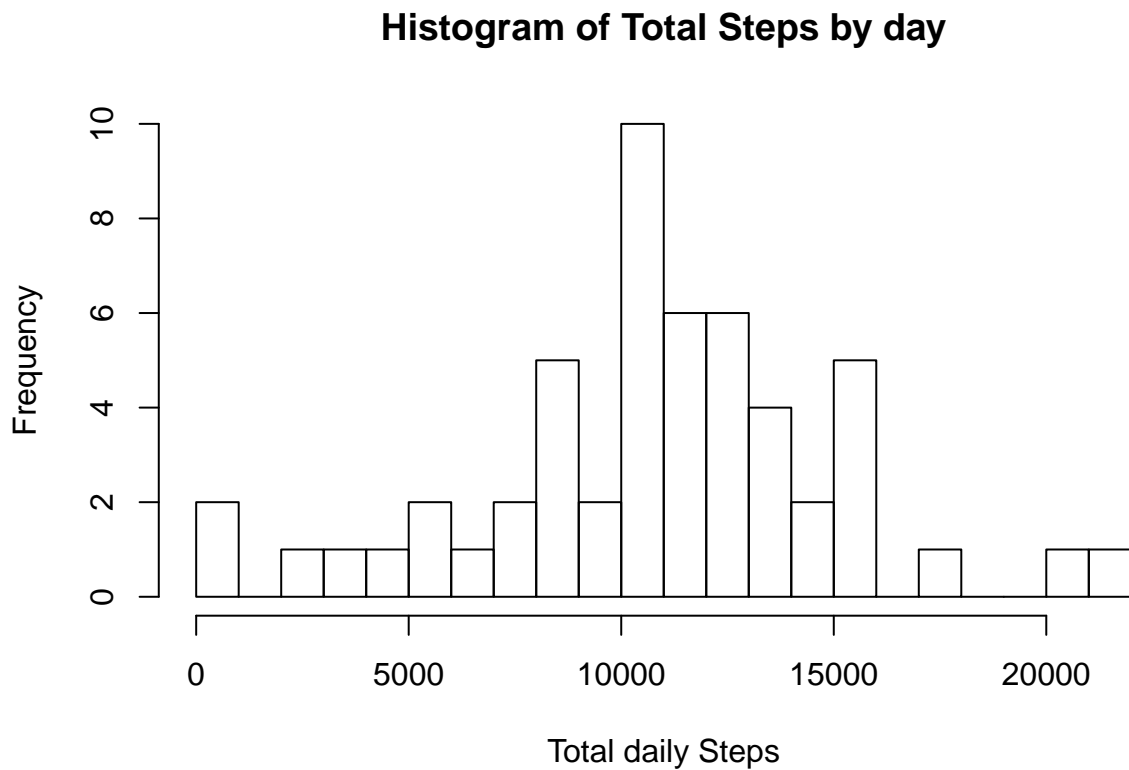
## The following objects are masked from 'package:stats':
##
##      filter, lag

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

library(ggplot2)
library(magrittr)
library(dplyr)
datebydate <- stepdata %>% select(date, steps) %>% group_by(date) %>% summarize(tsteps= sum(steps)) %>%
```

Then the histogram is plotted

```
hist(databydate$tsteps, xlab = "Total daily Steps",main="Histogram of Total Steps by day", breaks = 20)
```



Then the calculation of the mean and median of the total number of steps taken per day.

```
mean(databydate$tsteps)
```

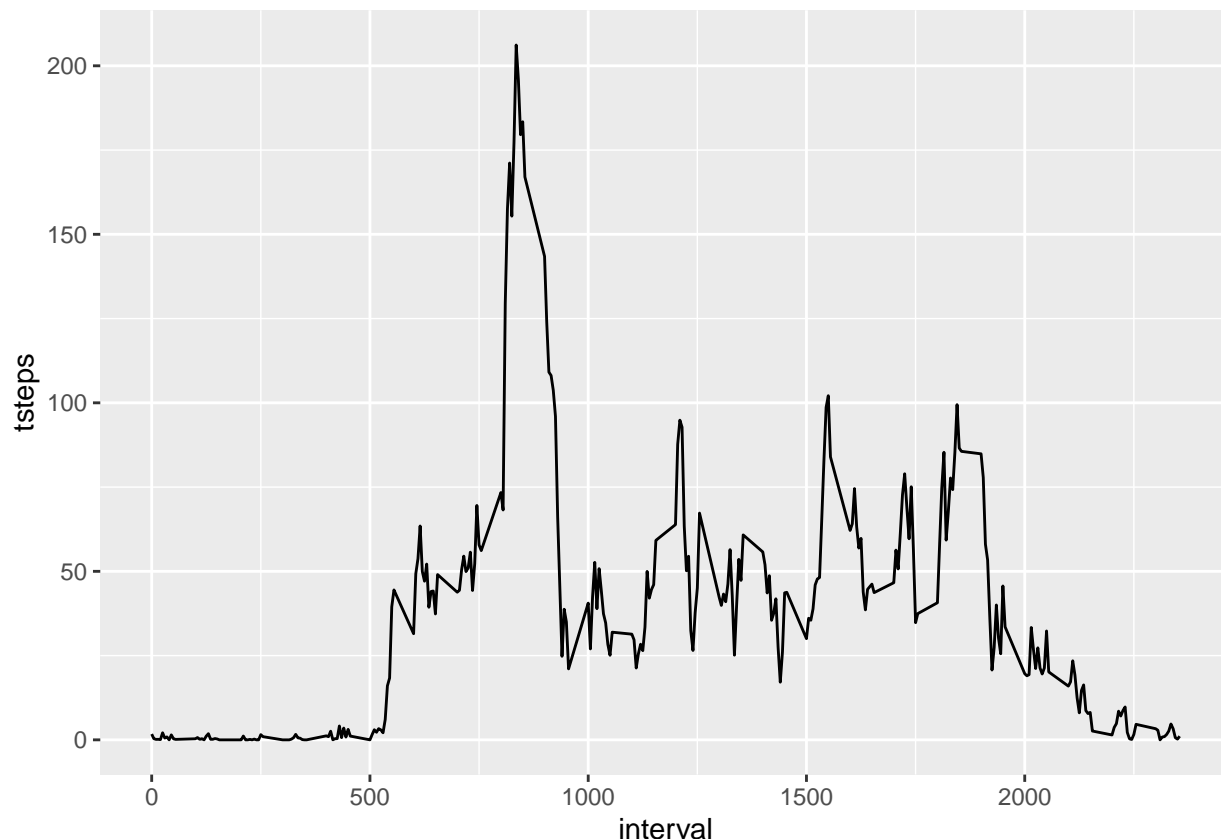
```
## [1] 10766.19
```

```
median(databydate$tsteps)
```

```
## [1] 10765
```

Then the time series are plotted:

```
databyinterval <- stepdata%>% select(interval, steps) %>% na.omit() %>% group_by(interval) %>% summarize(
  ggplot(databyinterval, aes(x=interval, y=tsteps))+ geom_line()
```



Then the 5-minute interval that, on average, contains the maximum number of steps:

```
databyinterval[which(databyinterval$tsteps == max(databyinterval$tsteps)),]
```

```
## # A tibble: 1 x 2
##   interval tsteps
##   <int>   <dbl>
## 1     835    206.
```

Then the strategy for imputing missing data: First is to calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
# generate listing of NA's
missingVals <- sum(is.na(data))
```

```
## Warning in is.na(data): is.na() aplicado a un objeto que no es (lista o vector)
## de tipo 'closure'
missingVals
```

```
## [1] 0
```

Then I use the mean for that 5 -minute interval to replace all the missing values in the dataset. At the end, I will check if all the NAs have been replaced

```
replacewithmean <- function(x) replace(x, is.na(x), mean(x, na.rm = TRUE))
meandata <- stepdata %>% group_by(interval) %>% mutate(steps = replacewithmean(steps))
head(meandata)
```

```
## # A tibble: 6 x 3
## # Groups:   interval [6]
```

```
##      steps date      interval
##      <dbl> <fct>      <int>
## 1  1.72   2012-10-01         0
## 2  0.340  2012-10-01         5
## 3  0.132  2012-10-01        10
## 4  0.151  2012-10-01        15
## 5  0.0755 2012-10-01        20
## 6  2.09   2012-10-01        25
```

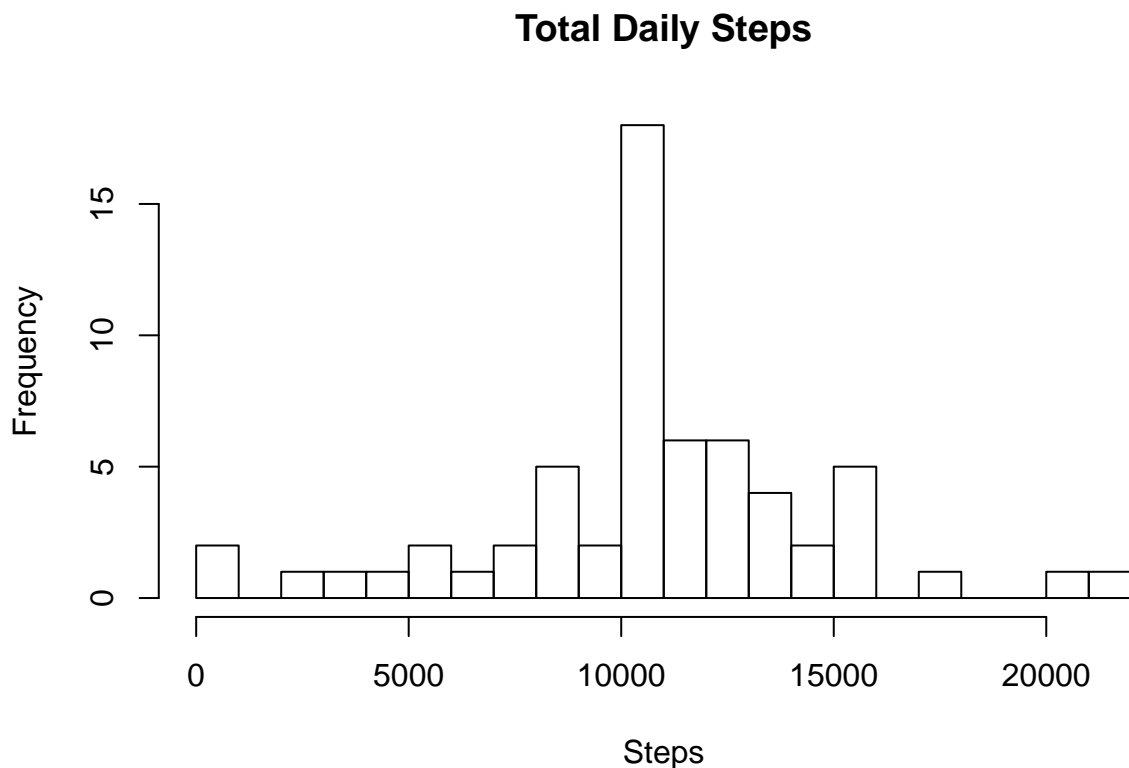
Then the histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day

```
FullSummedDataByDay <- aggregate(meandata$steps, by=list(meandata$date), sum)
```

```
names(FullSummedDataByDay)[1] = "date"
```

```
names(FullSummedDataByDay)[2] = "totalsteps"
```

```
hist(FullSummedDataByDay$totalsteps, xlab = "Steps", ylab = "Frequency", main = "Total Daily Steps", br
```



Then is answered: Are there differences in activity patterns between weekdays and weekends?

```
meandata$date <- as.Date(meandata$date)
```

```
meandata$weekday <- weekdays(meandata$date)
```

```
meandata$weekend <- ifelse(meandata$weekday=="Saturday" | meandata$weekday=="Sunday", "Weekend", "Weekday")
```

```
meandataweekendweekday <- aggregate(meandata$steps, by= list(meandata$weekend, meandata$interval), na.rm=T)
```

```
names(meandataweekendweekday) <- c("weekend", "interval", "steps")
```

```
ggplot(meandataweekendweekday, aes(x=interval, y=steps, color=weekend)) + geom_line() +
facet_grid(weekend ~.) + xlab("Interval") + ylab("Mean of Steps") +
ggtitle("Comparison of Average Number of Steps in Each Interval")
```

Comparison of Average Number of Steps in Each Interval

