

# Reproducible research project 1

Christophe Imbert de la Platière

8/13/2020

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

## 1. Code for reading in the dataset and/or processing the data

```
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", destfile = "activity.zip")
# unzip and read dataset
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
```

### load magrittr and dplyr packages

```
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

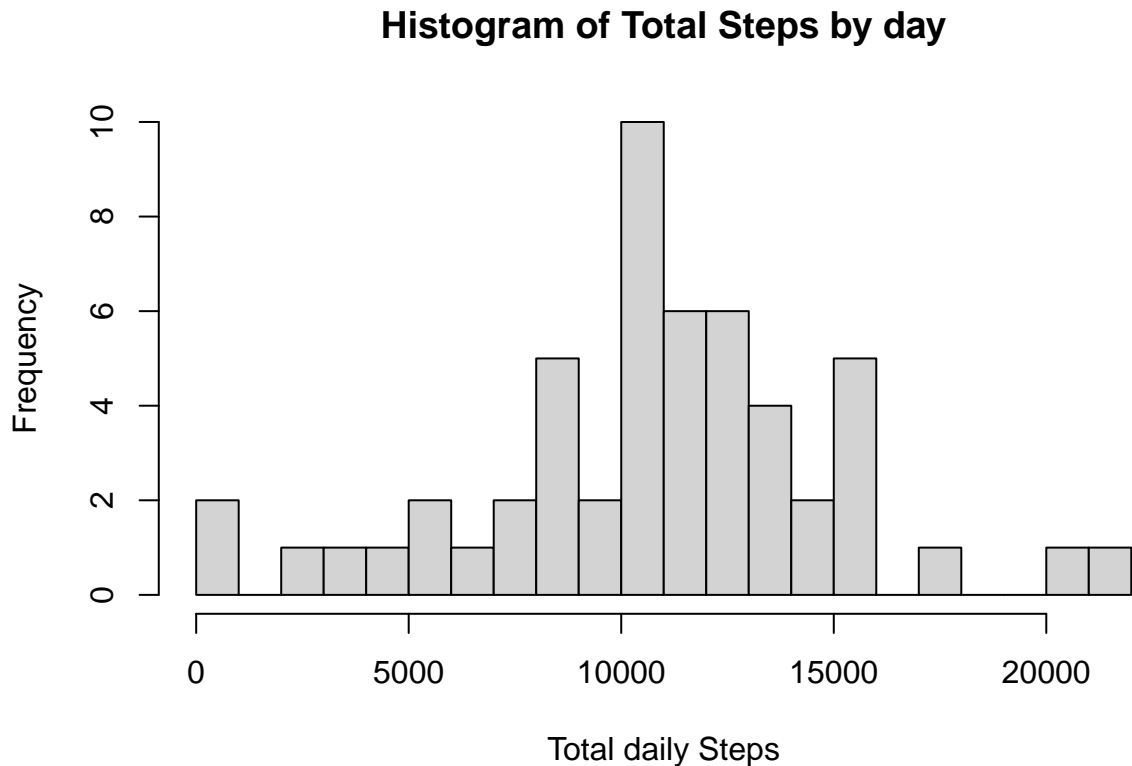
## calculate number of steps by date --> stepsbydate variable

stepsbydate <- stepdata %>% select(date, steps) %>% group_by(date) %>% summarize(tsteps= sum(steps)) %>%
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# 2. display histogram of the total number of steps each day
```

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



```
# 3a. mean value of steps by date
```

```
mean(stepsbydate$steps)
```

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

```
# 3b. median value of steps by date
```

```
median(stepsbydate$steps)
```

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

```
## install package ggplot2
```

```
library(ggplot2)
```

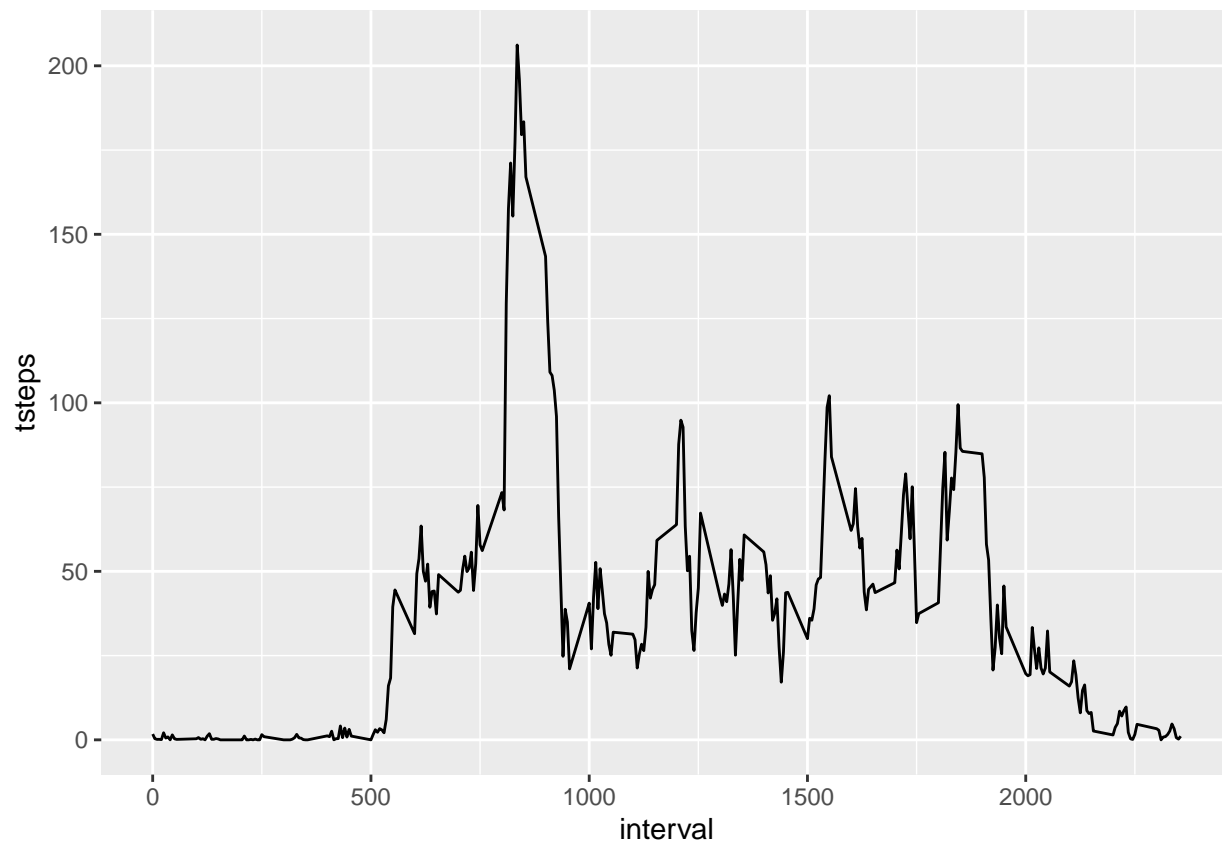
```
## calculate average number of steps taken
```

```
stepsbyinterval <- stepdata %>% select(interval, steps) %>% na.omit() %>% group_by(interval) %>% summarise(
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# 4. time series plot display
```

```
ggplot(stepsbyinterval, aes(x=interval, y=tsteps)) + geom_line()
```



```
# 5. The 5-minute interval that, on average, contains the maximum number of steps
stepsbyinterval[which(stepsbyinterval$tsteps== max(stepsbyinterval$tsteps)),]
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
## # A tibble: 1 x 2
##   interval tsteps
##   <int>   <dbl>
## 1     835    206.
```

```
## imputing missing values NAs
missingVals <- sum(is.na(data))
```

```
## Warning in is.na(data): is.na() applied to non-(list or vector) of type
## 'closure'
```

```
## display missing values
missingVals
```

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

```
# 6. Code to describe and show a strategy for imputing missing data
library(magrittr)
library(dplyr)
```

```

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

```

```

## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?

```

```

## # A tibble: 6 x 3
## # Groups:   interval [6]
##   steps date      interval
##   <dbl> <chr>      <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

```

```

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

```

```

names(FullSummedDataByDay)[1] ="date"
names(FullSummedDataByDay)[2] ="totalsteps"
head(FullSummedDataByDay,15)

```

```

##           date totalsteps
## 1 2012-10-01  10766.19
## 2 2012-10-02   126.00
## 3 2012-10-03  11352.00
## 4 2012-10-04  12116.00
## 5 2012-10-05  13294.00
## 6 2012-10-06  15420.00
## 7 2012-10-07  11015.00
## 8 2012-10-08  10766.19
## 9 2012-10-09  12811.00
## 10 2012-10-10   9900.00
## 11 2012-10-11  10304.00
## 12 2012-10-12  17382.00
## 13 2012-10-13  12426.00
## 14 2012-10-14  15098.00
## 15 2012-10-15  10139.00

```

```

## Summary of new data : mean & median
summary(FullSummedDataByDay)

```

```

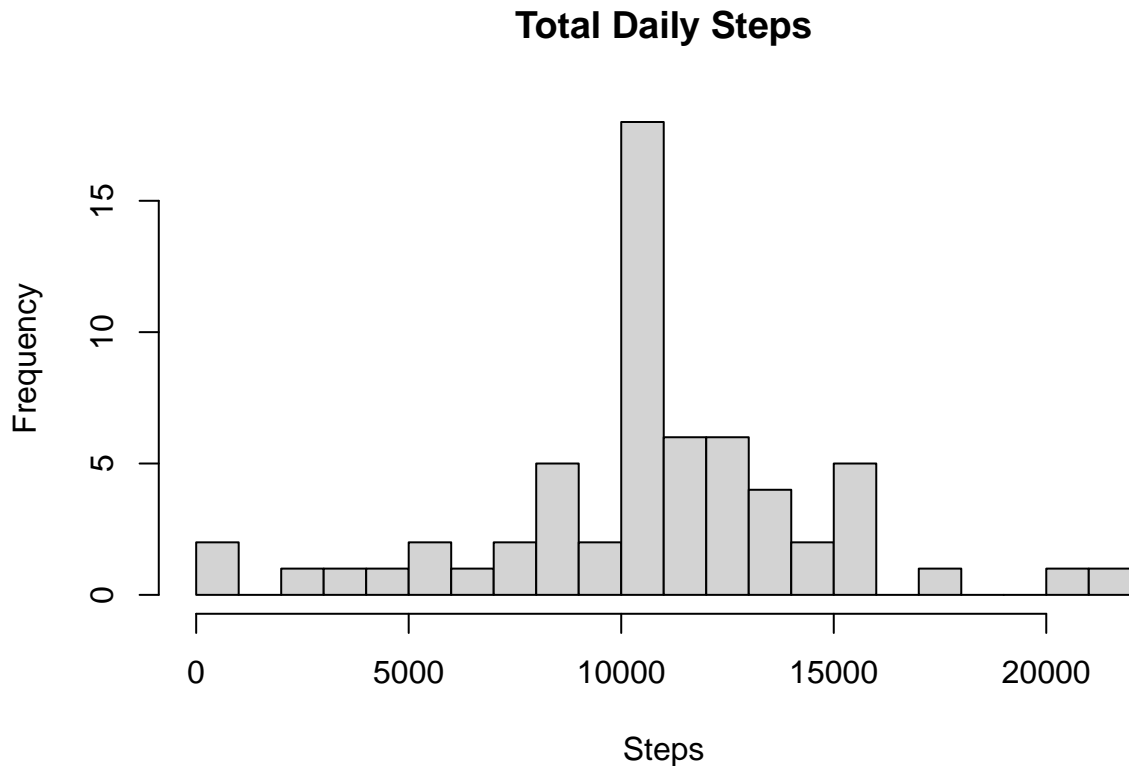
##           date      totalsteps
## Length:61      Min.   :  41
## Class :character 1st Qu.: 9819
## Mode  :character Median :10766
##                               Mean  :10766
##                               3rd Qu.:12811

```

```
##                               Max.      :21194
```

```
# 7. Histogram of the total number of steps taken each day after missing values are imputed
```

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



## 8. compare mean and median of these new data vs initial dataset

```
oldmean <- mean(databydatetsteps, na.rm = TRUE) newmean <- mean(FullSummedDataByDay$totalsteps)
```

```
# Old mean and New mean oldmean newmean
```

```
oldmedian <- median(databydatetsteps, na.rm = TRUE) newmedian <- median(FullSummedDataByDay$totalsteps)
```

```
# Old median and New median oldmedian newmedian
```

```
meandata$date <- as.Date(meandata$date) meandata$weekday <- weekdays(meandata$date) meandata$weekend <-  
ifelse(meandata$weekday=="Saturday" | meandata$weekday=="Sunday", "Weekend", "Weekday")
```

```
## 9. Panel plot comparing the average number of steps taken per 5-minute interval across 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")
```

```
library(ggplot2)
```

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

