begin with loading package

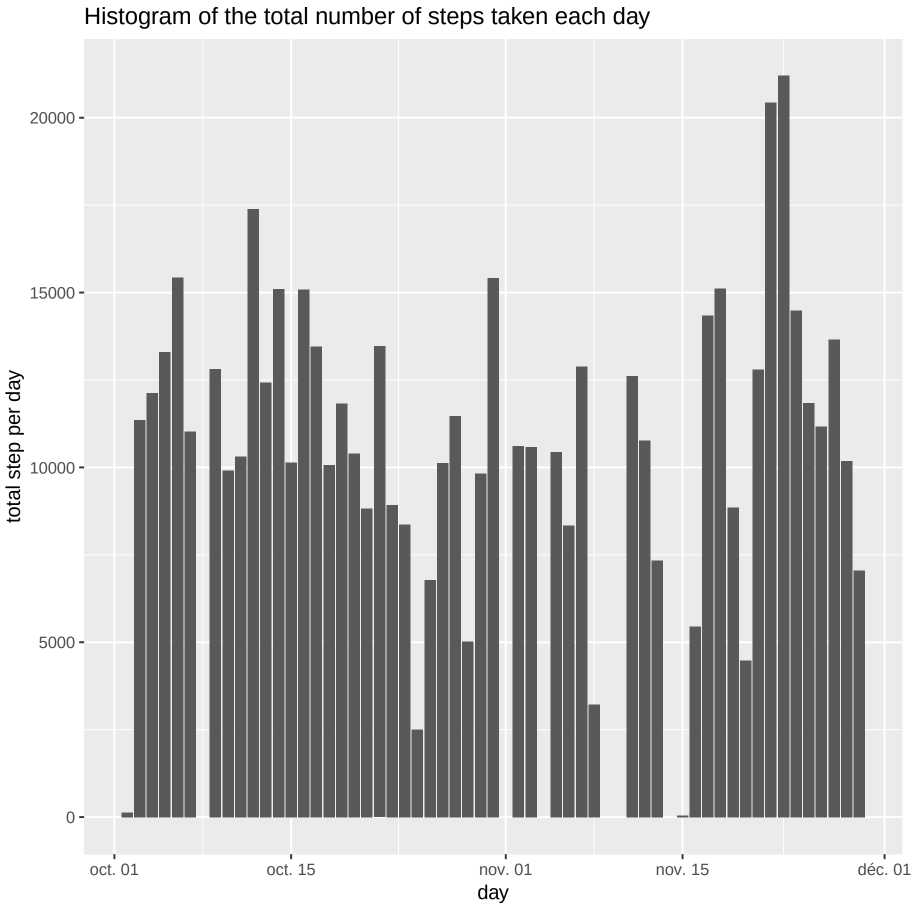
```
library(readr)
library(ggplot2)
library(scales)

##
## Attaching package:  ’scales’
## The following object is masked from ’package:readr’:
##
##    col_factor

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
```

Code for reading in the dataset and/or processing the data. BE CAREFUL, THIS IS WORKING WITH FRENCH DATE

```
setwd("C:/Users/giffonev/Dropbox/TPR/reproducible-research-1")

activity <- read_csv("activity.csv", col_types = cols(steps = col_integer()))
activity$date <- as.Date(activity$date,format="%Y/%m/%d")
g <- ggplot(data=activity,aes(x=activity$date, y=activity$steps))
h <- ggplot(data=activity,aes(x=activity$interval, y=activity$steps))
activity$week <-



activity$week <- sapply(activity$date,weekdays)
 for (i in 1:length(activity$steps)) {
    if (activity$week[i] == "samedi") {
         activity$week[i] <- "weekend"
    } else if (activity$week[i] == "dimanche") {
         activity$week[i] <- "weekend"
    } else {
         activity$week[i] <- "weekday"
    }
 }
activity$week <- as.factor(activity$week)
```

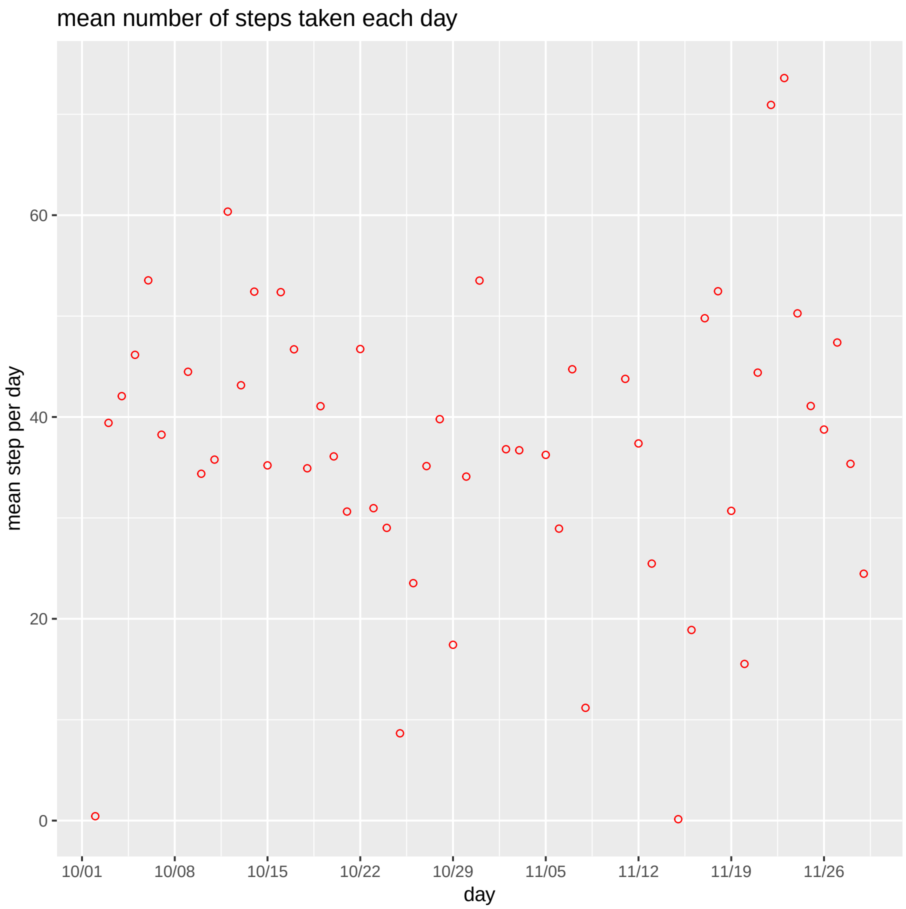
Histogram of the total number of steps taken each day

```
g + geom_bar(stat = "identity") + labs(title = "Histogram of the total number of steps taken

## Warning:  Removed 2304 rows containing missing values (position_stack).
```

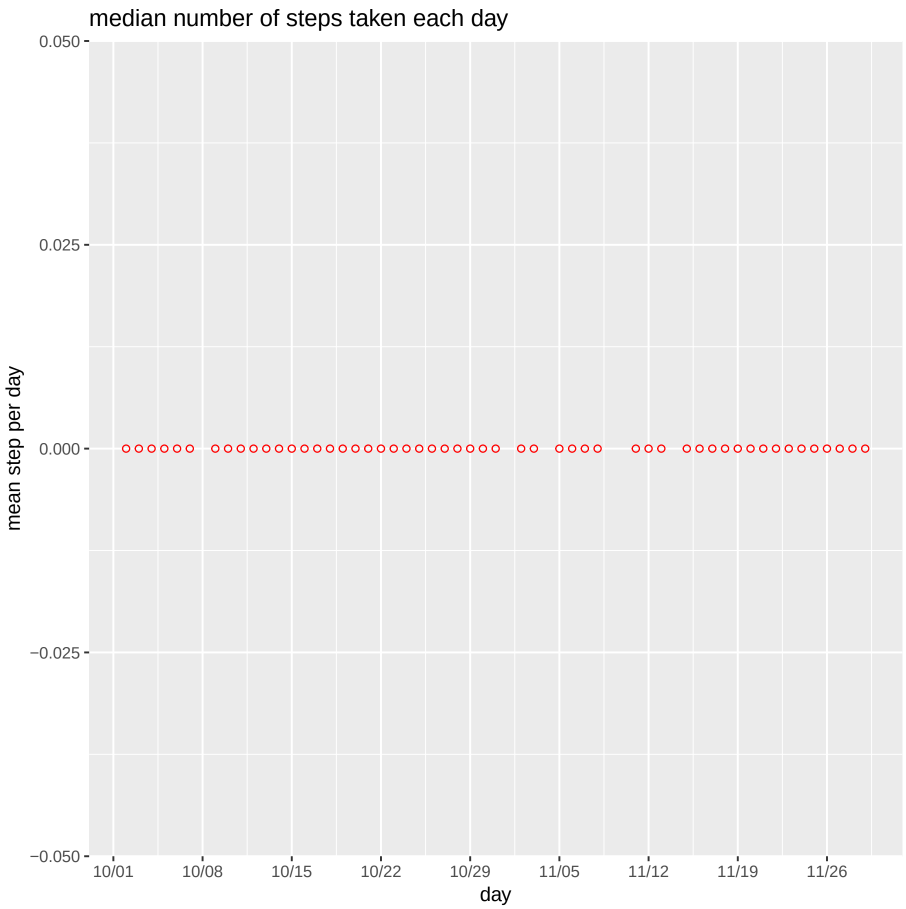


Mean and median number of steps taken each day

```
g + stat_summary(fun.y=mean, geom="point", shape=1, size = 1.5, col = "red") + labs(title =

## Warning:  Removed 2304 rows containing non-finite values (stat_summary).
```

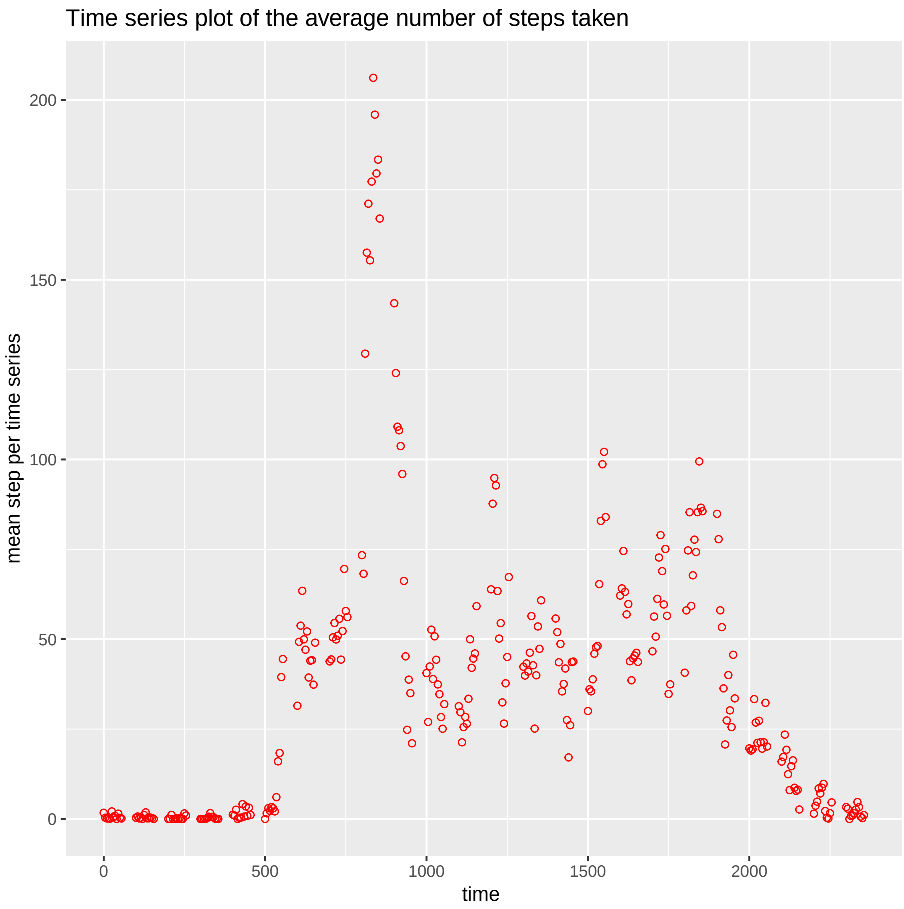


```
g + stat_summary(fun.y=median, geom="point", shape=1, size = 1.5, col = "red") + labs(title

## Warning:  Removed 2304 rows containing non-finite values (stat_summary).
```



Time series plot of the average number of steps taken

```
h + stat_summary(fun.y=mean, geom="point", shape=1, size = 1.5, col = "red") + labs(title =

## Warning:  Removed 2304 rows containing non-finite values (stat_summary).
```



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

```
test <- tapply(activity$steps,activity$interval,mean, na.rm=TRUE)
max(test)

## [1] 206.1698
```

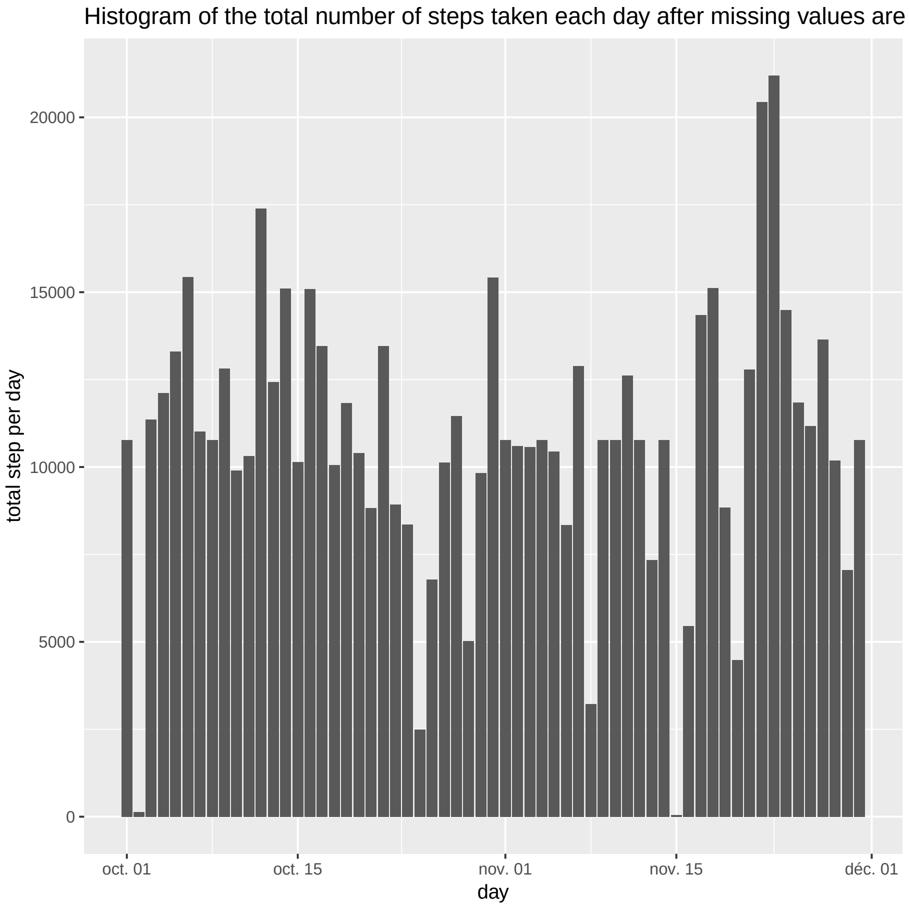
Code to describe and show a strategy for imputing missing data

```
#number of NA :
sum(is.na(activity$steps))

## [1] 2304

#replacing NA with the mean :
activity$steps[is.na(activity$steps)] <- mean(activity$steps, na.rm  = TRUE)
sum(is.na(activity$steps))
```

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

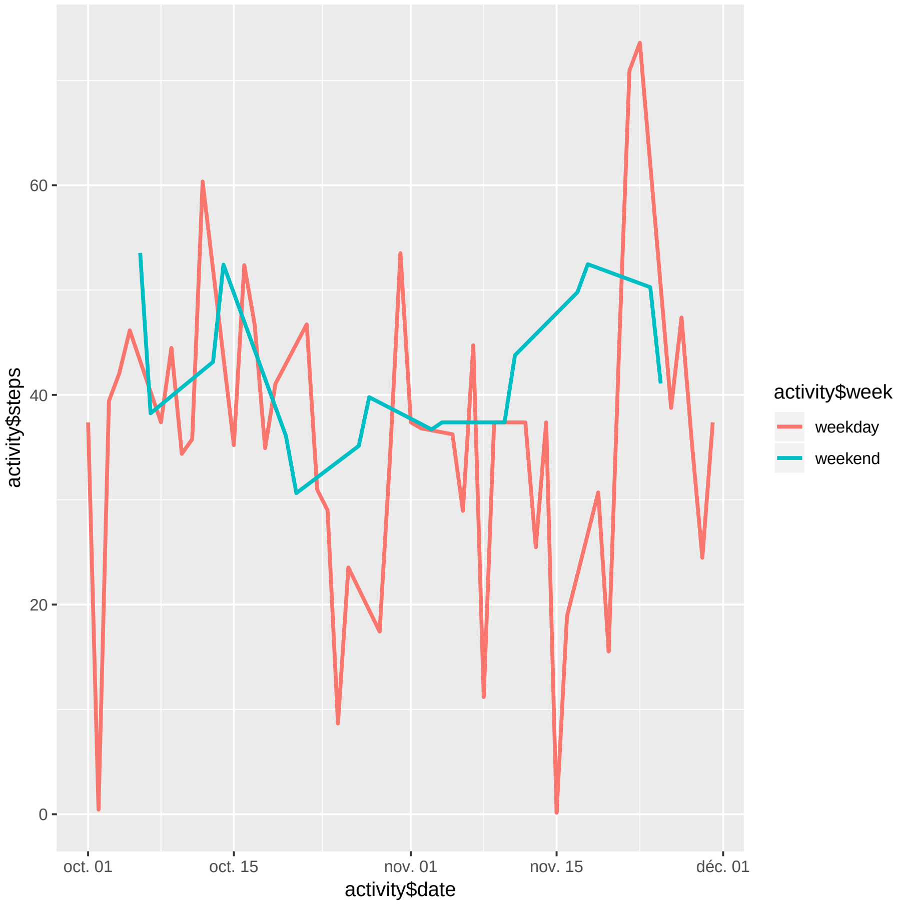
Histogram of the total number of steps taken each day after missing values are imputed

```
j <- ggplot(data=activity,aes(x=activity$date, y=activity$steps))
j + geom_bar(stat = "identity") + labs(title = "Histogram of the total number of steps taken
```



Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
k <- ggplot(data=activity,aes( x = activity$date, y=activity$steps, color = activity$week))
k + stat_summary(fun.y=mean, geom="line", size = 1)
```



All of the R code needed to reproduce the results (numbers, plots, etc.) in the report

setwd(”C:/Users/giffonev/Dropbox/TPR/reproducible-research-1”)

activity ¡- read$_{c}sv(”activity.csv”, col_{t}ypes = cols(steps = col_{i}nteger()))activity$date ¡- as.Date(activity$date, format = ”g < -ggplot(data = activity, aes(x = activity$date, y=activity$steps))h < -ggplot(data = activity, aes(x = activity$interval, y=activity$steps))activity$week ¡-

activity$week < -sapply(activity$date,weekdays) for (i in 1:length(activity$steps))if(activity$week[i] == ”samedi”) activity$week[i] < -”weekend”elseif(activity$week[i] == ”dimanche”) activity$week[i] < -”weekend”elseactivity$week[i] ¡- ”weekday” activity$week < -as.factor(activity$week)

Histogram of the total number of steps taken each day

g + geom$_{b}ar(stat = ”identity”)+labs(title = ”Histogramofthetotalnumberofstepstakeneachday”, x = ”day”, y = ”totalstepperday”)$

Mean and median number of steps taken each day

g + stat$_{s}ummary(fun.y = mean, geom = ”point”, shape = 1, size = 1.5, col = ”red”)+labs(title = ”meannumberofstepstakeneachday”, x = ”day”, y =$

$"meanstepperday") + scale_{xd}ate(labels = date_format("$

g + stat$_{s}ummary(fun.y = median, geom = "point", shape = 1, size = 1.5, col = "red") + labs(title = "mediannumberofstepstakeneachday", x = "day", y = "meanstepperday") + scale_{xd}ate(labels = date_format("$

Time series plot of the average number of steps taken

h + stat$_{s}ummary(fun.y = mean, geom = "point", shape = 1, size = 1.5, col = "red") + labs(title = "Timeseriesplotoftheaveragenumberofstepstaken", x = "time", y = "meansteppertimeseries")$

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

test ¡- tapply(activity$steps, activity$interval,mean, na.rm=TRUE) max(test)

Code to describe and show a strategy for imputing missing data

number of NA : sum(is.na(activity$steps$))

replacing NA with the mean : activity$steps[is.na(activity$steps)] ¡- mean(activity$steps, na.rm = TRUE)sum(is.na(activity$steps))

Histogram of the total number of steps taken each day after missing values are imputed

j ¡- ggplot(data=activity,aes(x=activity$date, y = activity$steps)) j + geom$_{b}ar(stat = "identity") + labs(title = "Histogramofthetotalnumberofstepstakeneachdayaftermissingvaluesareimputed$
$"day", y = "totalstepperday")$

Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

k ¡- ggplot(data=activity,aes(x = activity$date, y = activity$steps, color = activity$week))k + stat_{s}ummary(fun.y = mean, geom = "line", size = 1)$@