

Reproducible Research

hamada zidan

5/2/2022

Reproducible Research Week 2 Project

Loading and preprocessing the data

Load the data Process/transform the data (if necessary) into a format suitable for your analysis

```
# download file from web
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", destfile = "activity.zip", mode="wb")
# 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
```

1. Calculate total number of steps taken each day

```
library(magrittr)
library(dplyr)

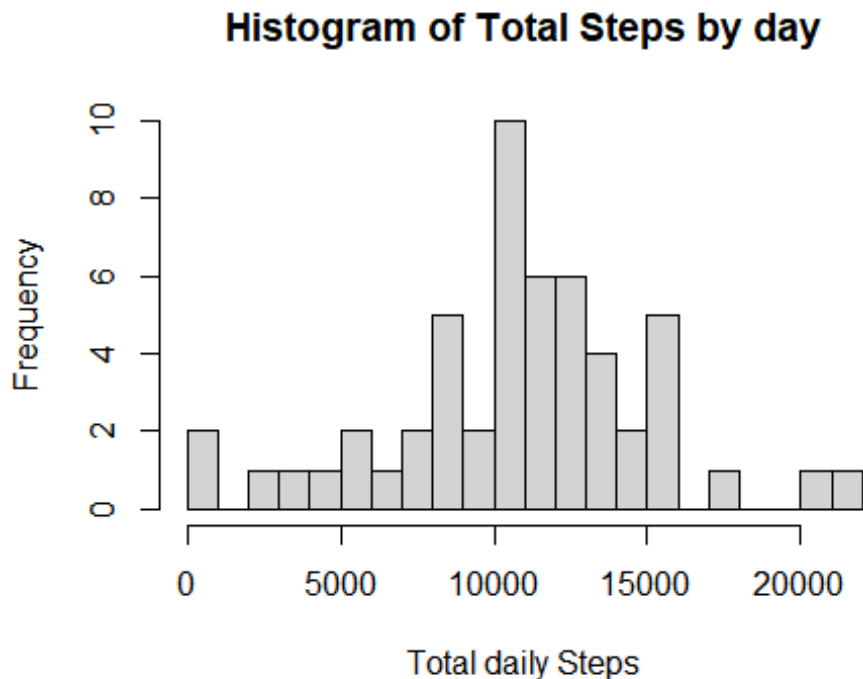
##
## Attaching package: 'dplyr'

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

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

databydate <- stepdata %>% select(date, steps) %>% group_by(date) %>%
summarize(tsteps= sum(steps)) %>%na.omit()
```

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



2. Calculate and report the mean and median of the total number of steps taken per day

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

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

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

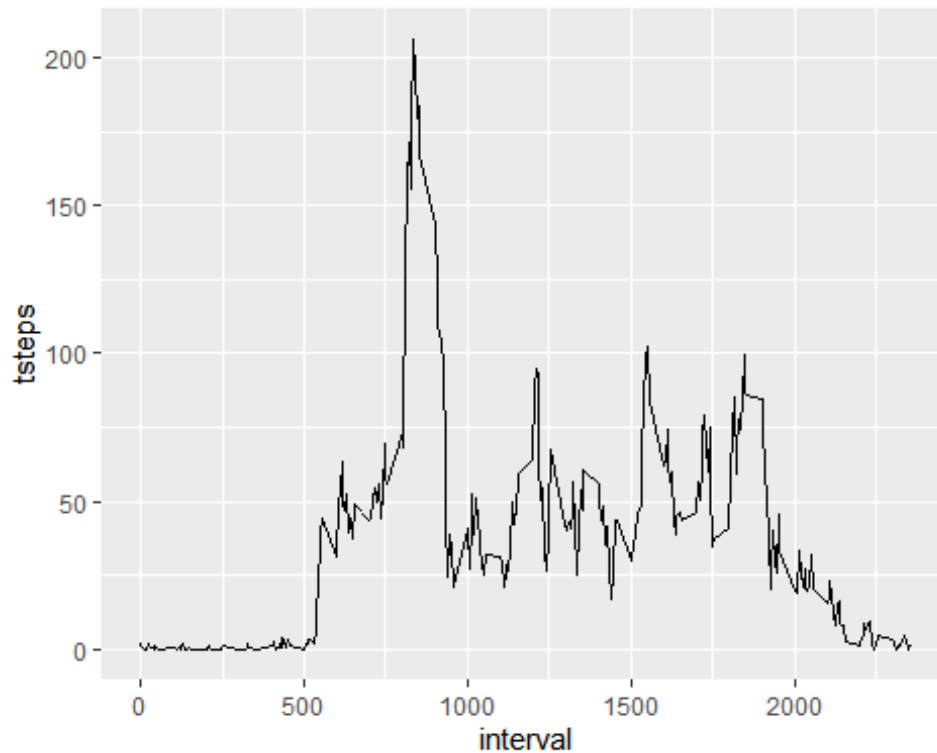
```
## [1] 10766.19
```

4. Time series plot

```
library(ggplot2)
```

```
databyinterval <- stepdata%>% select(interval, steps) %>% na.omit() %>%  
group_by(interval) %>% summarize(tsteps= mean(steps))
```

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



5.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.
```

Imputing missing values

1. 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() applied to non-(list or vector) of
type
## 'closure'
```

```
missingVals
```

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

2. 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.

I will 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

```
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)

## # 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
```

4 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.

```
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
```

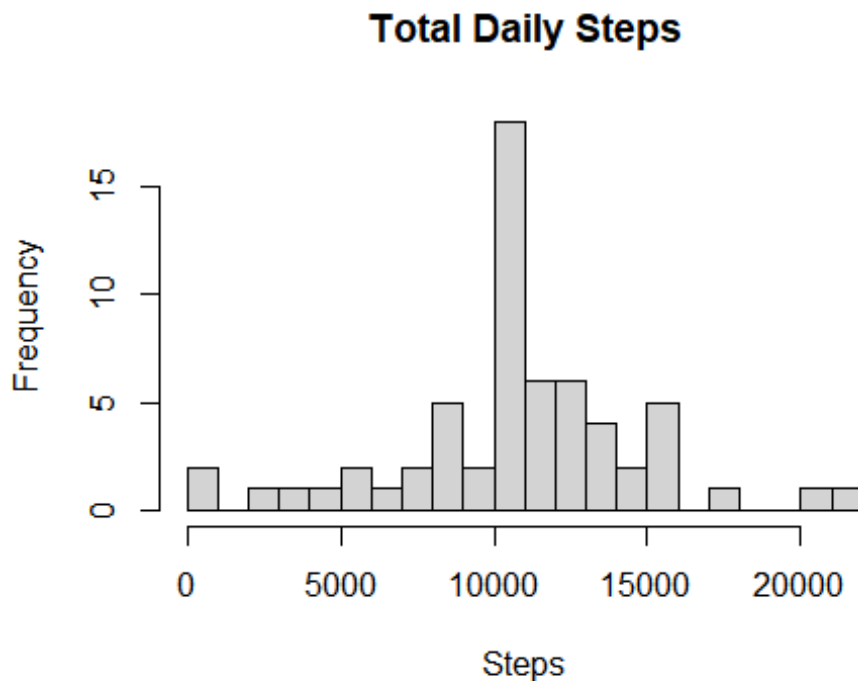
Summary of new data : mean & median

```
summary(FullSummedDataByDay)
```

```
##      date      totalsteps
## Length:61      Min.   :  41
## Class :character 1st Qu.: 9819
## Mode  :character Median :10766
##                      Mean  :10766
##                      3rd Qu.:12811
##                      Max.   :21194
```

Making a histogram

```
hist(FullSummedDataByDay$totalsteps, xlab = "Steps", ylab =
"Frequency", main = "Total Daily Steps", breaks = 20)
```



4C Compare the mean and median of Old and New data

```
oldmean <- mean(databydate$steps, na.rm = TRUE)
newmean <- mean(FullSummedDataByDay$totalsteps)
# Old mean and New mean
oldmean

## [1] 10766.19

newmean

## [1] 10766.19

oldmedian <- median(databydate$steps, na.rm = TRUE)
newmedian <- median(FullSummedDataByDay$totalsteps)
# Old median and New median
oldmedian

## [1] 10765

newmedian

## [1] 10766.19
```

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$weekday=="Sunday", "Weekend", "Weekday" )
library(ggplot2)
meandataweekendweekday <- aggregate(meandata$steps , by=
list(meandata$weekend, meandata$interval), na.omit(mean))
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

