Reproducible research: Project 1

Daniel V.

25/8/2020

About The assignment

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

The data for this assignment can be downloaded from the course web site: https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip

```
data<-read.csv("./activity.csv", sep=",", header= T);
head(data)
```

```
##   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
```

The variables included in this dataset are:

1. **steps:** Number of steps taking in a 5-minute interval (missing values are coded as NA)
2. **date:** The date on which the measurement was taken in YYYY-MM-DD format
3. **interval:** Identifier for the 5-minute interval in which measurement was taken

```
str(data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

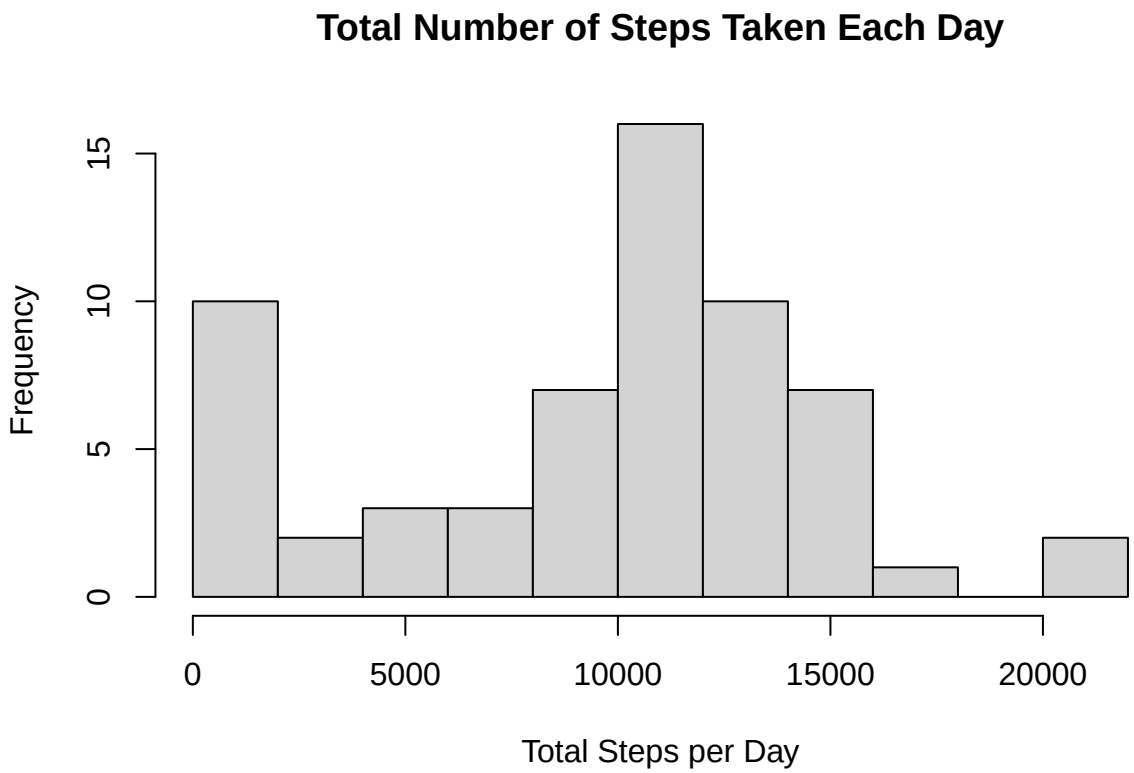
Processing the data

First I changed the data type from *chr* to Date in column "**date**" , then I removed all the NAs in the data set.

```
data$date<- as.Date(data$date, format('%Y-%m-%d'))
workdays<- unique(data$date)
```

1. Plot a Histogram of the total number of steps taken each day.

```
data_day_steps<- with(data,tapply(steps, date, sum, na.rm=TRUE))
hist(data_day_steps, breaks=9, xlab="Total Steps per Day", main= "Total Number of Steps Taken Each Day")
```



2. Mean and median number of steps taken each day.

The mean of number steps taken each day is:

```
mean(data_day_steps)
```

```
## [1] 9354.23
```

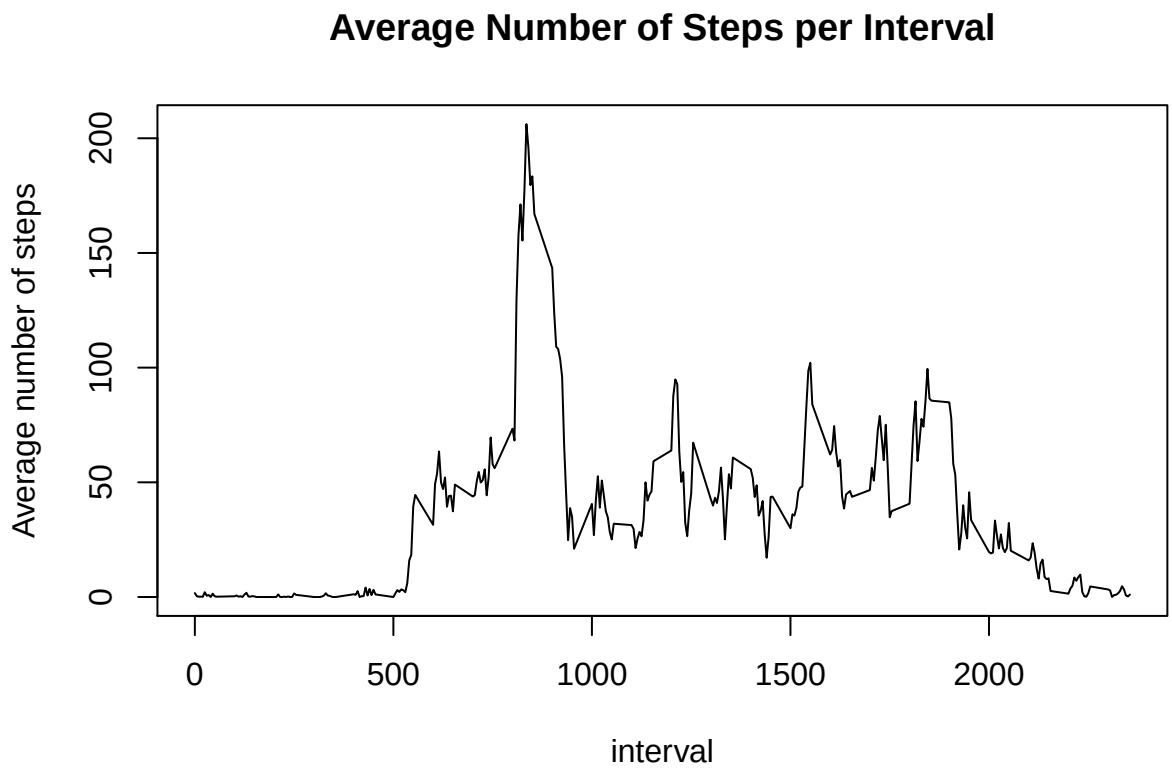
The median of number steps taken each day is:

```
median(data_day_steps)
```

```
## [1] 10395
```

3. Time series plot of the average number of steps taken.

```
data_day_mean<- with(data,aggregate(steps, by=list(interval), mean, na.rm=TRUE))
names(data_day_mean)<-c("interval","mean")
with(data_day_mean, plot(interval,mean,type="l", ylab="Average number of steps", main="Average Number o
```



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

Here is the interval, and the mean for that interval.

```
data_day_mean[which.max(data_day_mean$mean),]
```

```
##     interval     mean
## 104      835 206.1698
```

5. Code to describe and show a strategy for imputing missing data.

calculated the amount of missing values:

```
sum(is.na(data));sum(is.na(data$steps))
```

```
## [1] 2304
```

```
## [1] 2304
```

As we can see, all the missing values are in column **steps**, now how much the missing values are in proportion to the data:

```
sum(is.na(data$steps))/nrow(data)
```

```
## [1] 0.1311475
```

Now a simple strategy for imputing missing data using the `zoo library`:

```
library(zoo)
```

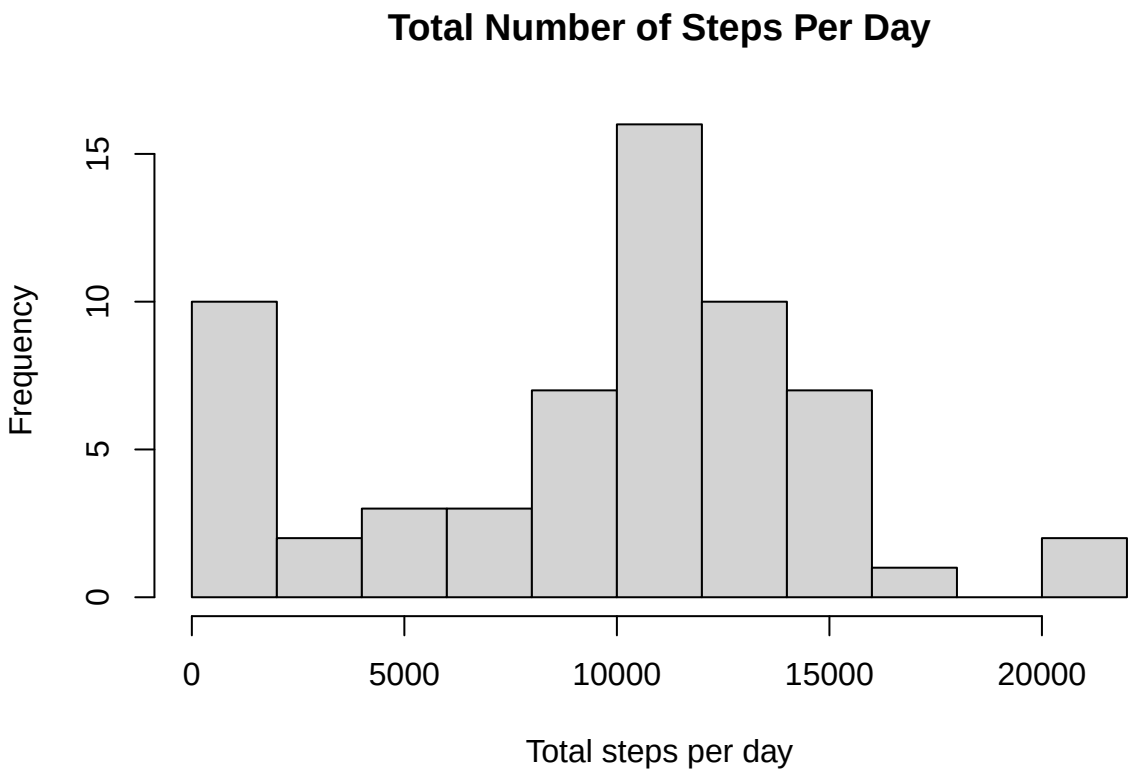
```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
data_imputed<- data
data_imputed$steps<-na.fill(data$steps, c("extend", NA))
```

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

```
Tot_steps_imputed<- with(data_imputed, tapply(steps, date, sum, na.rm=TRUE))
hist(Tot_steps_imputed, breaks=10, xlab="Total steps per day", main="Total Number of Steps Per Day")
```



7. Mean and median number of steps taken each day.

The new mean of number steps taken each day is:

```
mean(Tot_steps_imputed)
```

```
## [1] 9354.23
```

The new median of number steps taken each day is:

```
median(Tot_steps_imputed)
```

```
## [1] 10395
```

So, there are no differences between old mean, median and new mean, median.

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

Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
data$days<- weekdays(data$date)
data$days<-replace(data$days,data$days=="sábado"|data$days=="domingo","weekend")
#table(data$days)
data$days<-replace(data$days,data$days=="lunes"|data$days=="martes"|data$days=="miércoles"  |data$days==
table(data$days)
```

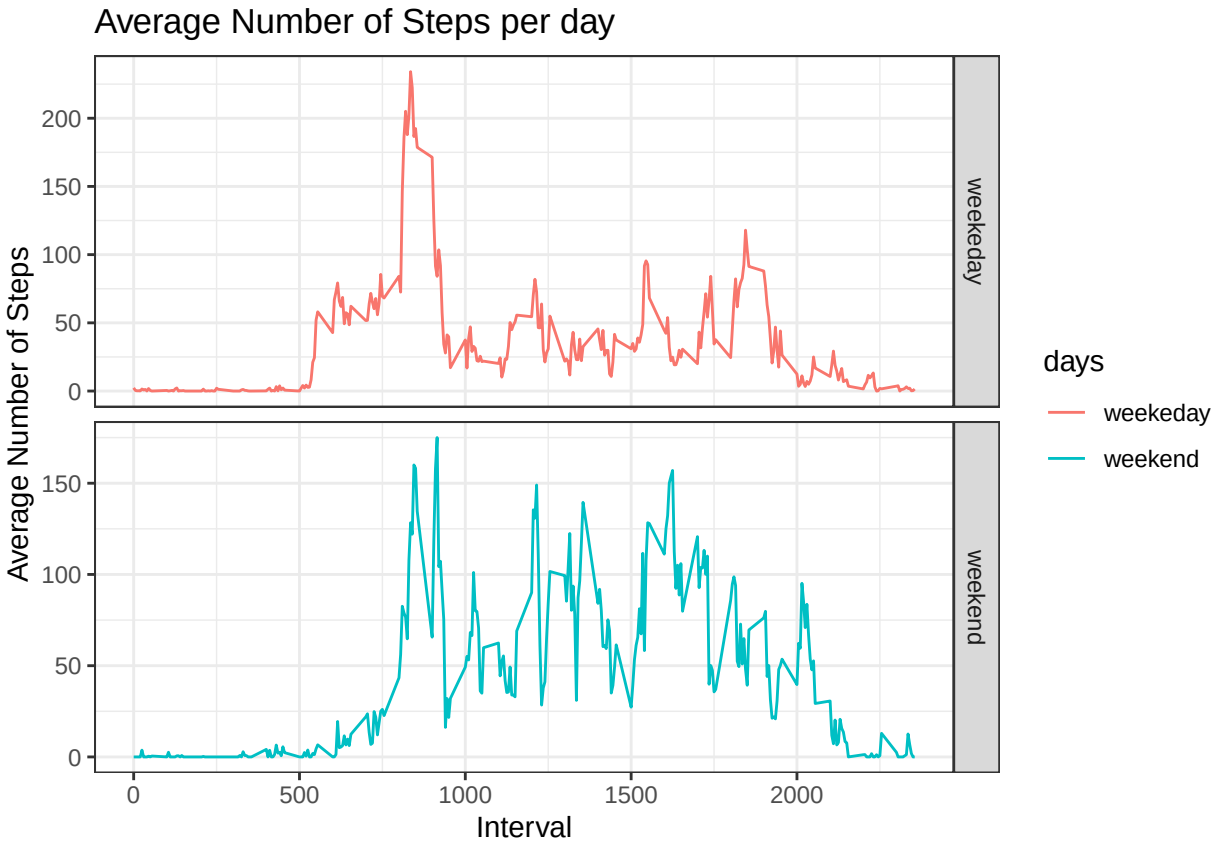
```
##
## weekeday  weekend
##    12960     4608
```

```
data$days<- as.factor(data$days)
str(data)
```

```
## 'data.frame':    17568 obs. of  4 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ days    : Factor w/ 2 levels "weekeday","weekend": 1 1 1 1 1 1 1 1 1 1 ...
```

Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
New_data <- with(data, aggregate(steps~interval + days,FUN = mean, na.rm = TRUE))
library(ggplot2)
g<- ggplot(New_data, aes(x=interval, y= steps, color= days))+geom_line()
g+facet_grid(days~., scales="free")+labs(x= "Interval", y="Average Number of Steps", title="Average Numb
```


Average Number of Steps per day
Average Number of Steps
200
150
100
50
0
weekeday
150
100
50
0
weekend
0
500
1000
1500
2000
Interval
days
weekeday
weekend