Reproducible Research Project 1

Maria

16/08/2021

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.

Project Description

This project is part of the Data Science Specialization Series: Reproducible Research by the John Hopkins University. The objective of this project is to create an R markdown document showing all steps taken to load/process and answer the questions provided for a dataset containing number of steps taken by an individual in 5 minute intervals.

Loading the data

1. Instructions: show any code that is needed to load the data, process/transform the data (if necessary) into a format suitable for your analysis.
2. Code:

```r
unzip("activity.zip") ## unzip folder
activity<-read.csv("activity.csv") ## load data into "activity" variable
str(activity) ## explore the dataset
```

```
## '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 ...
```

```r
##Note that date is shown as a character
activity$date<-as.Date(activity$date)
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 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 ...
## Now, it is in the format that we want.
```

Mean Total Number of Steps per Day

1. Instructions: for this part of the assignment you can ignore missing values in the dataset.

- Calculate the total number of steps taken per day
- Make a histogram of the total number of steps taken each day
- Calculate and report the mean and median of the total number of steps taken per day

2. Code:

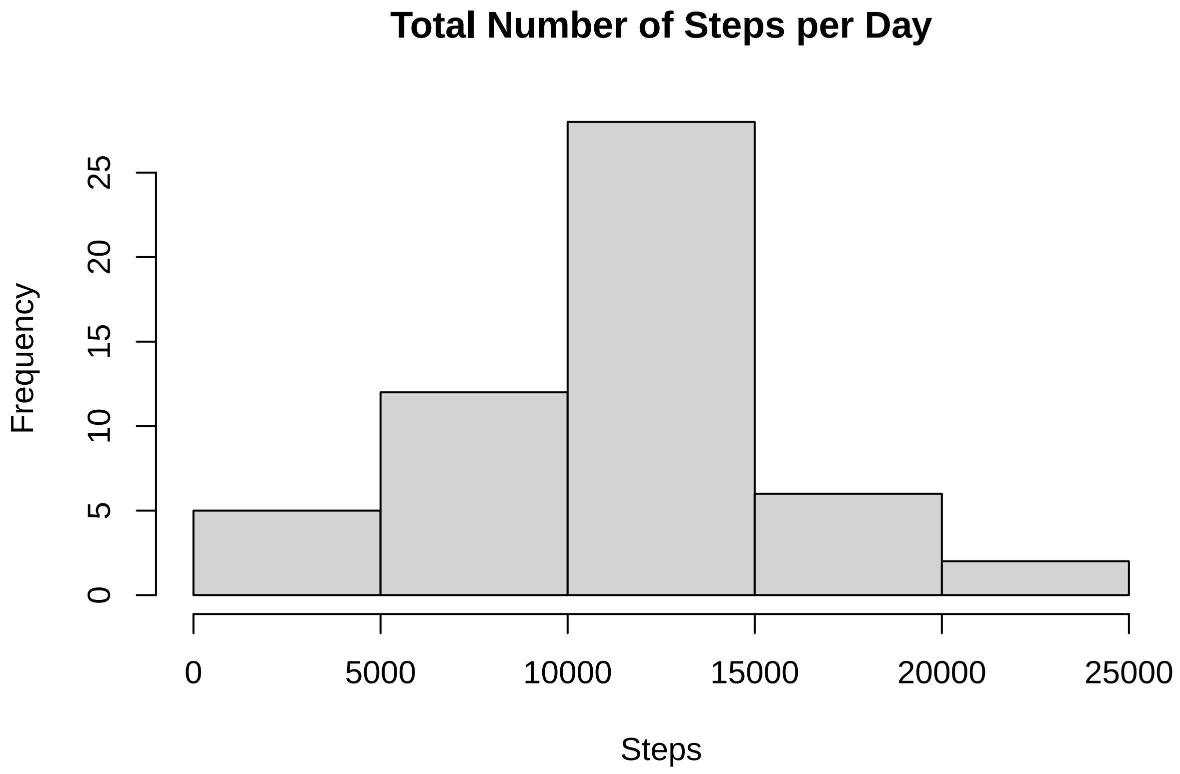
- Calculate the total number of steps taken per day

```
act <- activity[!is.na(activity$steps),] ##remove all NA values
actt<-aggregate(act$steps,by=list(act$date),sum) ##sum all steps per day
colnames(actt)<-c("Date","Steps") ##name columns
head(actt) ##first 6 rows to show an extract of the calculation
```

```
##         Date Steps
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

- Make a histogram of the total number of steps taken each day

```
hist(actt$Steps,xlab="Steps",main="Total Number of Steps per Day") #create histogram of steps per day
```



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

```
mean(actt$Steps) ##Mean of the total number of steps taken per day
```

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

```
median(actt$Steps) ##Median of the total number of steps taken per day
```

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

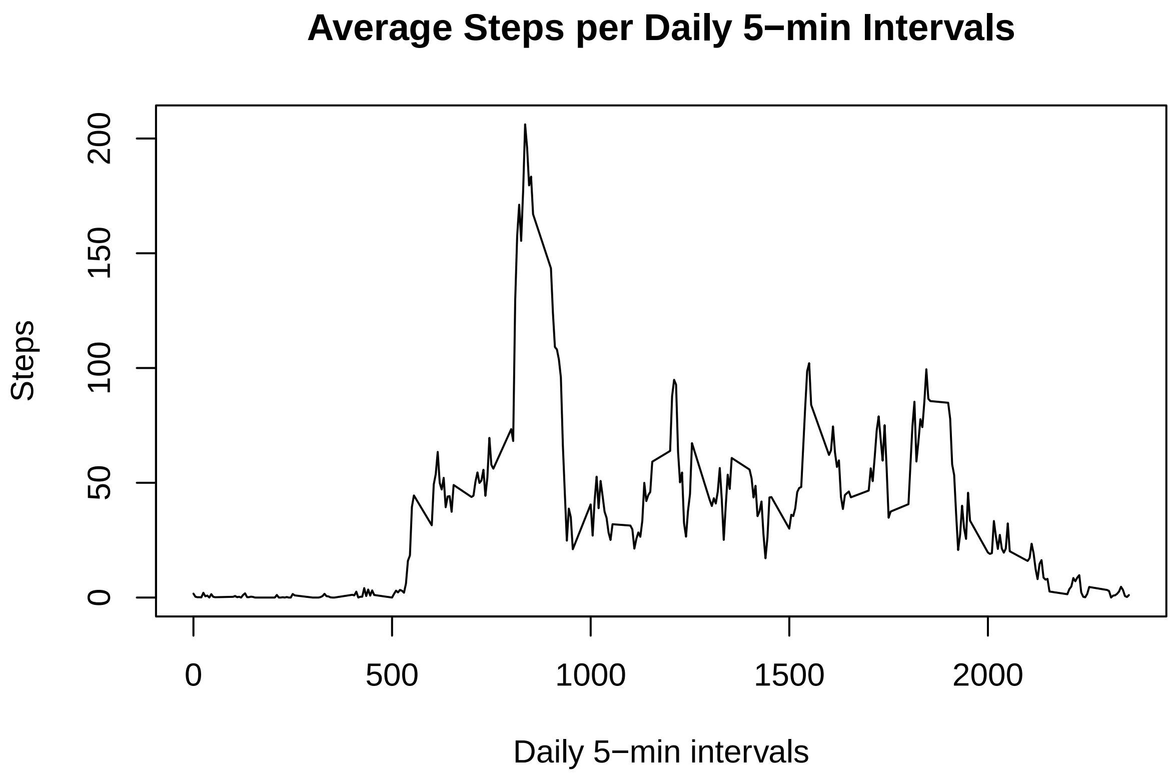
Average Daily Activity Pattern

1. Instructions:

- Make 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 days (y-axis)
- Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

2. Code:

```
act2<-aggregate(act,by=list(act$interval),mean) ##group values by interval number using mean function
act22<-act2[,1:2] ##take out other columns
colnames(act22)<-c("Interval","Steps") #rename columns
plot(act22$Interval,act22$Steps,xlab="Daily 5-min intervals",ylab="Steps",main="Average Steps per Daily
```



```
maxval<-max(act22$Steps)##find max value
maxrow<-filter(act22,act22$Steps==maxval)##filter data by max value
#maxrow ##show interval and steps of max (NOT SURE WHY THIS IS PRINTING AS A LONG LIST OF WEIRD THINGS)
print(835)
```

```
## [1] 835
```

Imputing Missing Values

1. Instructions: note that there are a number of days/intervals where there are missing values (NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

- Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs).
- 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.
- Create a new dataset that is equal to the original dataset but with the missing data filled in.

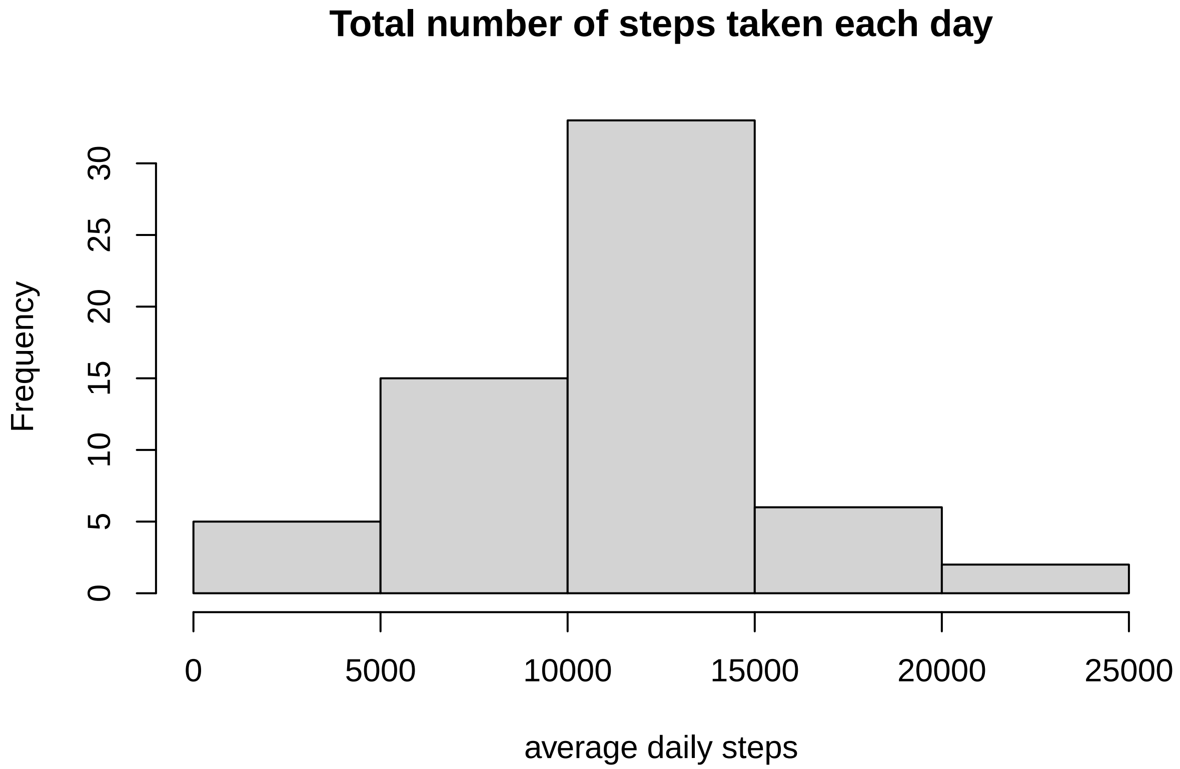
- Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

2. Code:

```
nrow(activity[is.na(activity$steps),])##count number of rows containing NAs
```

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

```
##lets group data by weekday and by interval by the function mean
act[,4]<-weekdays(act$date) ##create a weekday column
colnames(act)[4]<-"day" ##Add column day
naact<-activity[is.na(activity$steps),] ##NA data
naact[,4]<-weekdays(naact$date) #add day column
colnames(naact)[4]<-"day"#name column
act3<-aggregate(steps~interval+day,act,mean) ##group values by weekday
##and interval number using mean function
na<-merge(naact,act3,by=c("day","interval"))#merge NA data with average by interval and day
colnames(na)[5]<-"steps" #rename steps column
na<-na[,c(5,4,2,1)]#reorder columns to match act data
act33<-rbind(act,na) #merge both NA data and nonNA data
daily<-aggregate(steps~date,act33,sum) #sum values per day
hist(daily$steps,xlab="average daily steps",main="Total number of steps taken each day")
```



```
mean(daily$steps)
```

```
## [1] 10821.21
```

```
median(daily$steps)
```

```
## [1] 11015
```

The new mean and median of the steps once NA data was added increased not significantly.

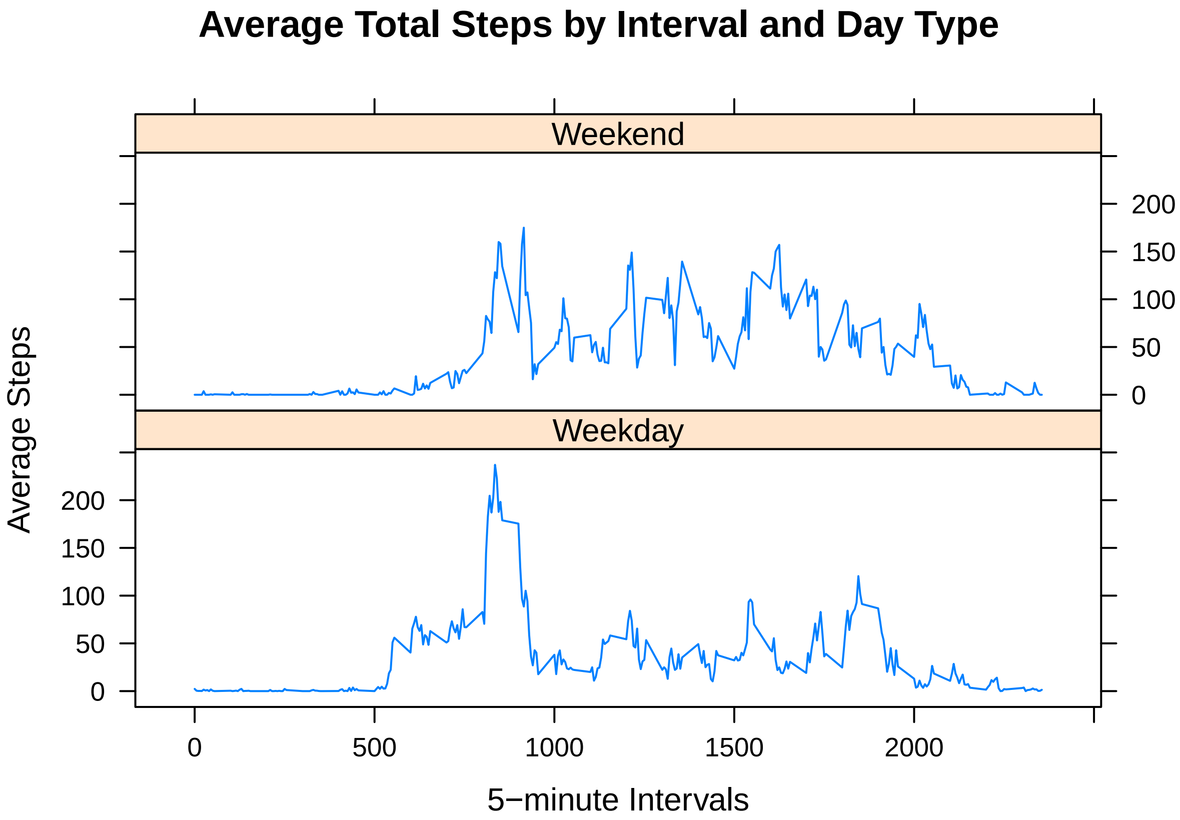
Patterns between Weekdays and Weekends

1. Instructions: For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

- 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.
- 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.

2. Code:

```
#add a column to classify day to weekday/weekend
act33[,5]<-ifelse(act33$day=="Sunday"|act33$day=="Saturday","Weekend","Weekday")
#name the column
colnames(act33)[5]<-"day_type"
act333<-aggregate(steps~interval+day_type,act33,mean)
library(lattice)
xyplot(steps~interval|day_type, data=act333, type="l",  layout = c(1,2),
       main="Average Total Steps by Interval and Day Type",
       ylab="Average Steps", xlab="5-minute Intervals")
```


Average Total Steps by Interval and Day Type
Weekend
Weekday
Average Steps
0
50
100
150
200
0
500
1000
1500
2000
5−minute Intervals