

PA1 Assignment

janneke.js

13-7-2020

```
knitr::opts_chunk$set(echo = TRUE)
```

Loading and preprocessing the data.

Read in the data:

```
data <- read.csv("D:/R Course/Data Science Specialisation (JHU)/Reproducible Research/repdata_data_acti  
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
```

Remove missing values:

```
data.complete <- na.omit(data)
```

What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day.

```
library(tidyverse)
```

```
## -- Attaching packages -----  
  
## v ggplot2 3.2.1    v purrr  0.3.4  
## v tibble  2.1.3    v dplyr  0.8.3  
## v tidyr   1.0.0    v stringr 1.4.0  
## v readr   1.3.1    v forcats 0.4.0  
  
## Warning: package 'purrr' was built under R version 3.6.3  
  
## -- Conflicts -----  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()    masks stats::lag()
```

```
activity_day <- group_by(data.complete, date)
activity_day <- summarize(activity_day, steps=sum(steps))
activity_day
```

```
## # A tibble: 53 x 2
##   date      steps
##   <fct>    <int>
## 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
## 7 2012-10-09  12811
## 8 2012-10-10   9900
## 9 2012-10-11  10304
## 10 2012-10-12 17382
## # ... with 43 more rows
```

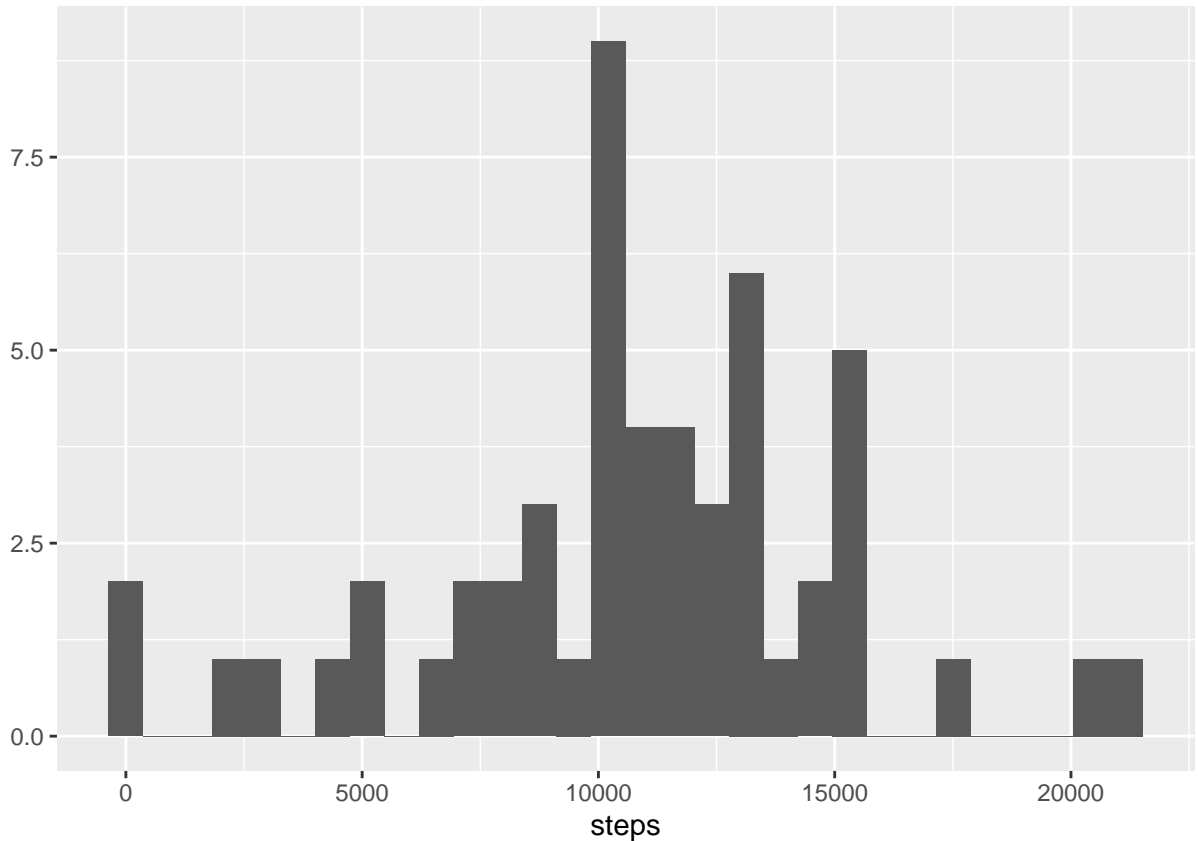
```
summary(activity_day)
```

```
##           date      steps
## 2012-10-02: 1   Min.    :  41
## 2012-10-03: 1   1st Qu.: 8841
## 2012-10-04: 1   Median :10765
## 2012-10-05: 1   Mean    :10766
## 2012-10-06: 1   3rd Qu.:13294
## 2012-10-07: 1   Max.    :21194
## (Other)      :47
```

2. Make a histogram of the total number of steps taken each day.

```
Image1 <- qplot(steps, data=activity_day)
Image1
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



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

```
meansteps_day <- mean(activity_day$steps)
meansteps_day
```

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

```
mediansteps_day <- median(activity_day$steps)
mediansteps_day
```

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

The mean number of steps per day is 10766.19 The median number of steps per day is 10765

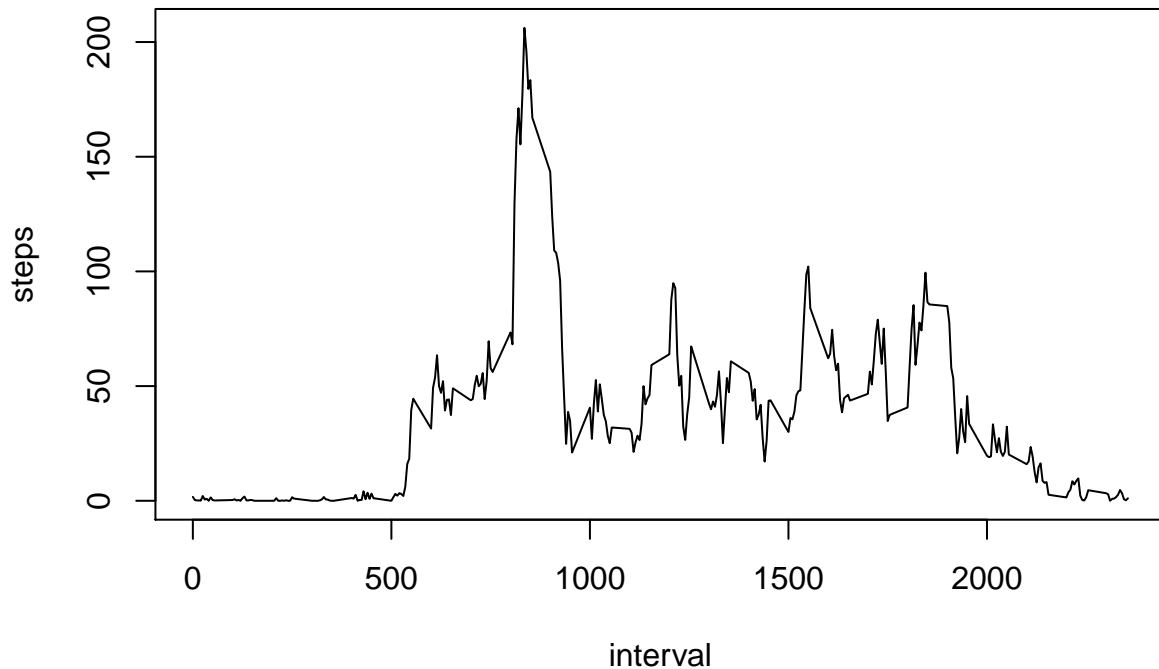
What is the average daily activity pattern?

1. Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis).

```
activity_interval <- group_by(data.complete, interval)
activity_interval <- summarize(activity_interval, steps=mean(steps))
```

```
library(ggplot2)

plot(steps~interval, data = activity_interval, type = "l")
```



2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
interval_maxsteps <- activity_interval[which.max(activity_interval$steps),]$interval
interval_maxsteps
```

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

The maximum number of steps was at the 835th 5-minute interval. ## Imputing missing values ##### 1. Calculate and report the total number of missing values in the dataset.

```
NA_data <- sum(is.na(data))
NA_data
```

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

There are 2304 missing values in the dataset.

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.

The missing values in the dataset will be filled up with the average number of steps of each interval. This will be done by the following code:

```
meansteps_interval <-function(interval){  
  activity_interval[activity_interval$interval==interval,]$steps  
}
```

3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

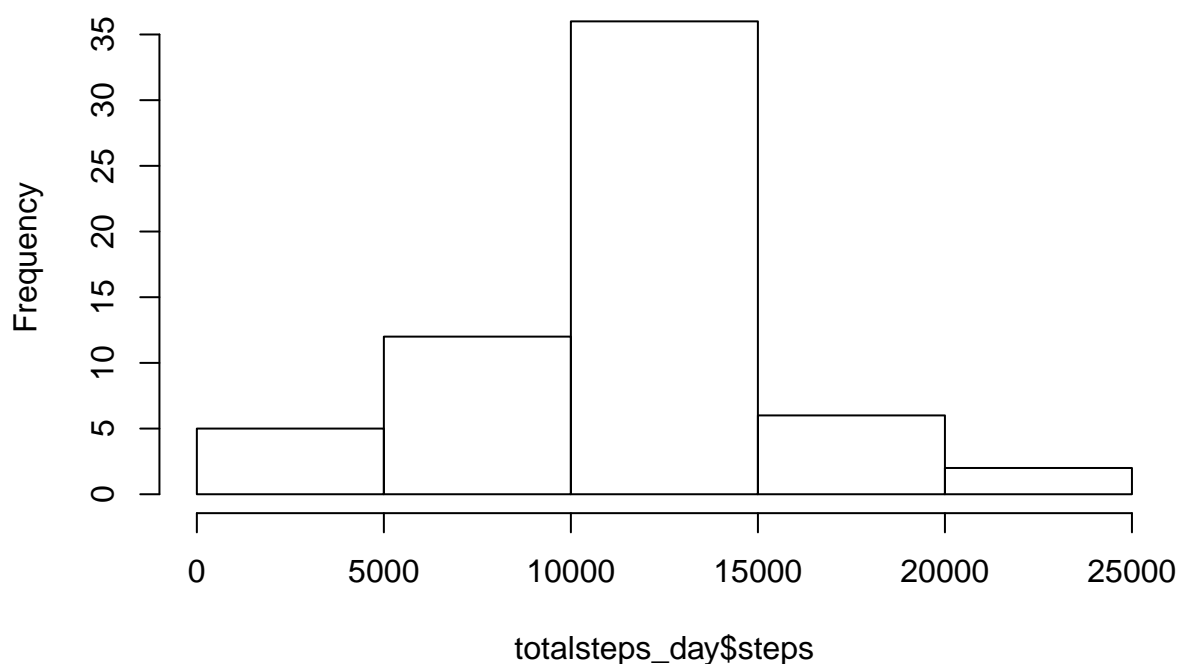
The following code fills in the mean number of steps for each NA value found in the dataset.

```
for(i in 1:nrow(data)){  
  if(is.na(data[i,]$steps)){  
    data[i,]$steps <- meansteps_interval(data[i,]$interval)  
  }  
}
```

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. 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?

```
totalsteps_day <- aggregate(steps ~ date, data=data, sum)  
hist(totalsteps_day$steps)
```

Histogram of totalsteps_day\$steps



```
totalsteps_day_mean <- mean(totalsteps_day$steps)
totalsteps_day_mean
```

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

```
meansteps_day
```

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

```
totalsteps_day_median <- median(totalsteps_day$steps)
totalsteps_day_median
```

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

```
mediansteps_day
```

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

```
totalsteps_day_median - mediansteps_day
```

```
## [1] 1.188679
```

The mean steps/day did not change after imputing the mean value replacing the missing values. The median differed 1.188679 points.

Are there differences in activity patterns between weekdays and weekends?

1. 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.

```
totalsteps_day$weekdays <- weekdays(as.Date(totalsteps_day$date))
totalsteps_day$weekend <- as.factor(totalsteps_day$weekdays=="zaterdag" | totalsteps_day$weekdays=="zondag")
levels(totalsteps_day$weekend) <- c("Weekday", "Weekend")
```

2. Make a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

I tried but was unable to perform this last assignment. I somehow can't get to produce the figure since my R is giving me the message that it doesn't recognise the valuable “interval” and when I tried renaming the variables I had to define xlim values, but I can't figure out what it means or how to do it. I tried the following (can't put this in my R chunk because the system can't knit errors): `plot(steps ~ interval | date, data = totalsteps_day, type = "l")`

```
View(totalsteps_day)

names(totalsteps_day) <- c("date", "steps", "weekdays", "interval")
View(totalsteps_day)

weekday_data <- totalsteps_day[totalsteps_day$weekdays=="Weekday",]
weekend_data <- totalsteps_day[totalsteps_day$weekend=="Weekend",]
```