PA1_template

Samuel Anang

2/13/2021

Loading and preprocessing the data

Show any code that is needed to 1.Load the data (i.e. read.csv()) 2.Process/transform the data (if necessary) into a format suitable for your analysis

```
# load all packages used in this exploratory analysis
library(knitr)
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
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
library(ggplot2)
opts_chunk$set(echo = TRUE)
# set up working directory
setwd('C:/Personalem/R/RepData_PeerAssessment1/activity')


# load data
data_row <- read.csv('activity.csv')
```

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset. 1.Make a histogram of the total number of steps taken each day 2.Calculate and report the mean and median total number of steps taken per day

```
# remove NA in data
data <- data_row[ with (data_row, { !(is.na(steps)) } ), ]

# print out first 20 rows
head(data,20)
```

```
##     steps       date interval
## 289     0 2012-10-02        0
## 290     0 2012-10-02        5
## 291     0 2012-10-02       10
## 292     0 2012-10-02       15
## 293     0 2012-10-02       20
## 294     0 2012-10-02       25
## 295     0 2012-10-02       30
## 296     0 2012-10-02       35
## 297     0 2012-10-02       40
## 298     0 2012-10-02       45
## 299     0 2012-10-02       50
## 300     0 2012-10-02       55
## 301     0 2012-10-02      100
## 302     0 2012-10-02      105
## 303     0 2012-10-02      110
## 304     0 2012-10-02      115
## 305     0 2012-10-02      120
## 306     0 2012-10-02      125
## 307     0 2012-10-02      130
## 308     0 2012-10-02      135
```

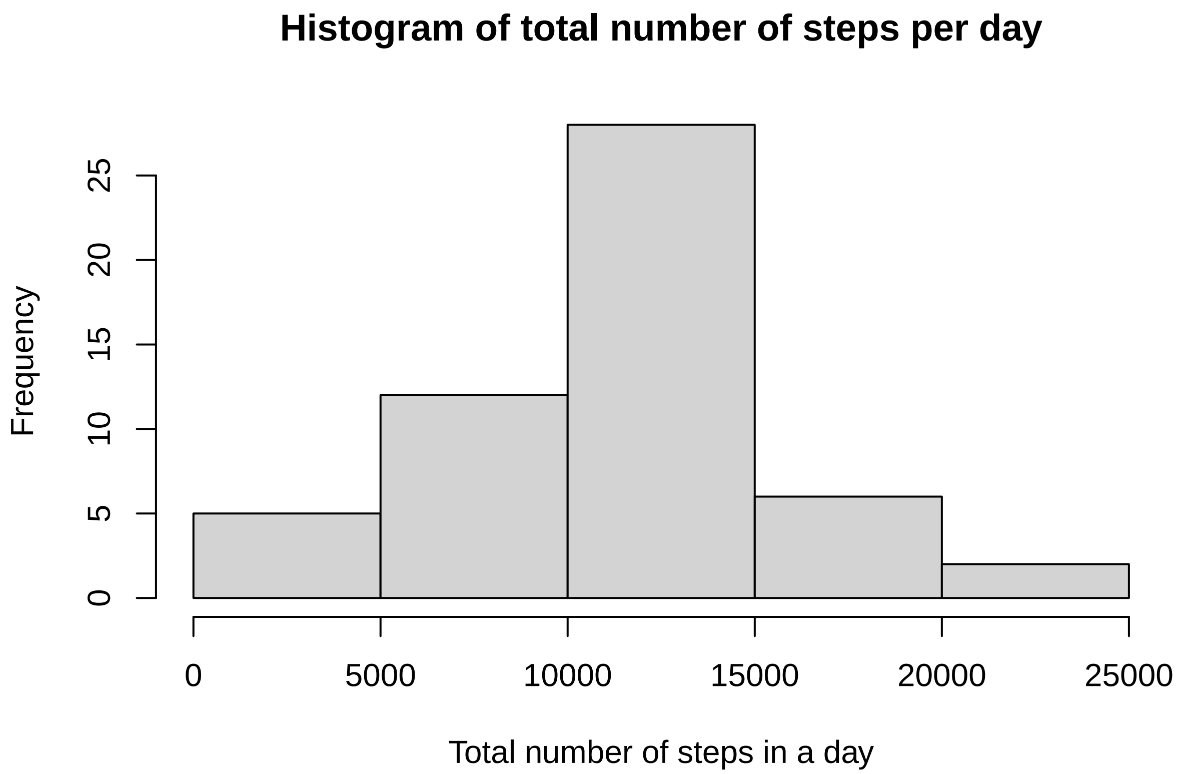
```
by_day <- group_by(data, date)
steps_by_day <- summarise(by_day, total = sum(steps))
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
steps_by_day
```

```
## # A tibble: 53 x 2
##    date       total
##    <chr>      <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
```

```
hist(steps_by_day$total, main="Histogram of total number of steps per day",
     xlab="Total number of steps in a day")
```



```
summary(steps_by_day)
```

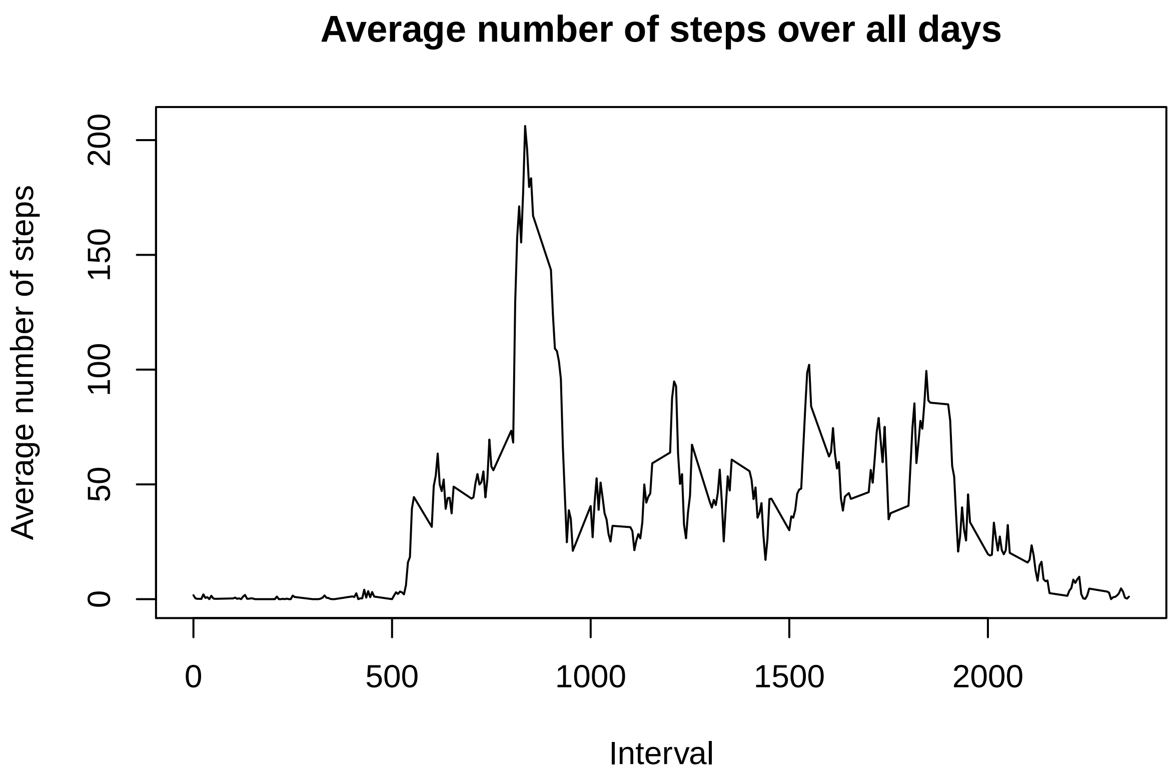
```
##      date               total      
##  Length:53          Min.   :   41  
##  Class :character   1st Qu.: 8841  
##  Mode  :character   Median :10765  
##                     Mean   :10766  
##                     3rd Qu.:13294  
##                     Max.   :21194  
```

What is the average daily activity pattern?

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

```
# preprocessing data for plot
steps_by_interval <- aggregate(steps ~ interval, data, mean)

# create a time series plot
plot(steps_by_interval$interval, steps_by_interval$steps, type='l',
     main="Average number of steps over all days", xlab="Interval",
     ylab="Average number of steps")
```



```
# find row with max of steps
max_steps_row <- which.max(steps_by_interval$steps)

# find interval with this max
steps_by_interval[max_steps_row, ]
```

```
##     interval    steps
## 104      835 206.1698
```

Imputing missing values

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

1.Calculate and report the total number of missing values in the dataset(i.e. the total number of rows with NAs) 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. 3.Create a new dataset that is equal to the original dataset but with the missing data filled in. 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?

```
sum(is.na(data_row))
```

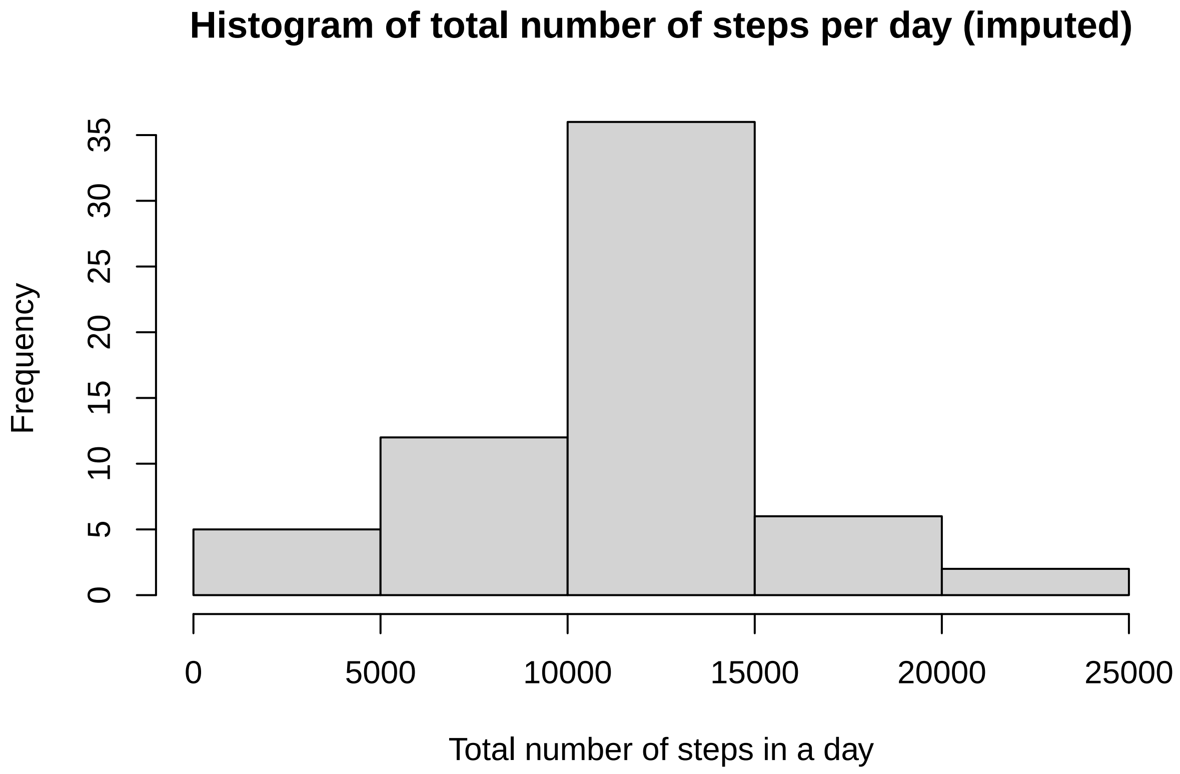
```
## [1] 2304
```

```
data_imputed <- data_row
for (i in 1:nrow(data_imputed)) {
  if (is.na(data_imputed$steps[i])) {
    interval_value <- data_imputed$interval[i]
    steps_value <- steps_by_interval[
      steps_by_interval$interval == interval_value,]
    data_imputed$steps[i] <- steps_value$steps
  }
}

# calculate  total number of steps taken each day
df_imputed_steps_by_day <- aggregate(steps ~ date, data_imputed, sum)
head(df_imputed_steps_by_day)
```

```
##         date    steps
## 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
```

```
hist(df_imputed_steps_by_day$steps, main="Histogram of total number of steps per day (imputed)",
     xlab="Total number of steps in a day")
```



```
# get mean and median of imputed data
mean(df_imputed_steps_by_day$steps)
```

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

```
median(df_imputed_steps_by_day$steps)
```

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

```
# get mean and median of data without NA's
mean(steps_by_day$total)
```

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

```
median(steps_by_day$total)
```

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

Are there differences in activity patterns between weekdays and weekends?

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

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. 2.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). The plot should look something like the following, which was creating using simulated data:

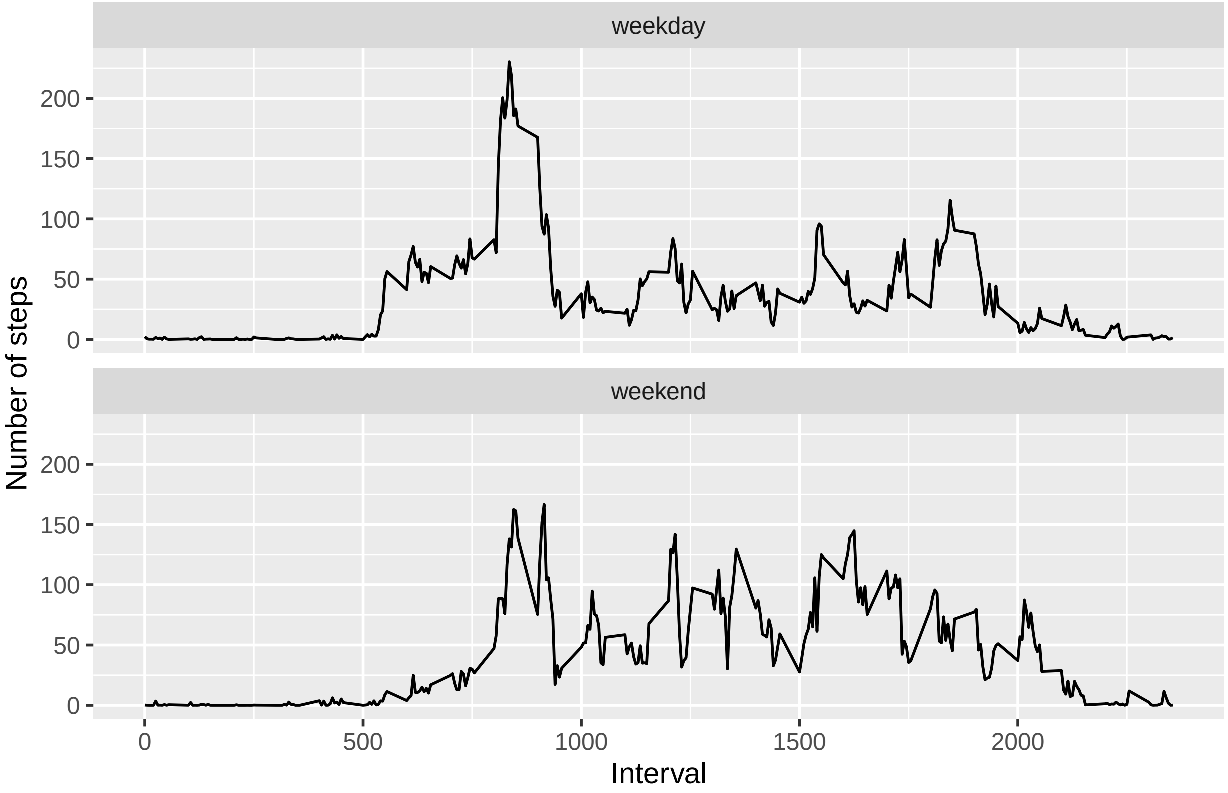
Your plot will look different from the one above because you will be using the activity monitor data. Note that the above plot was made using the lattice system but you can make the same version of the plot using any plotting system you choose.

```
data_imputed['type_of_day'] <- weekdays(as.Date(data_imputed$date))
data_imputed$type_of_day[data_imputed$type_of_day  %in% c('Saturday','Sunday') ] <- "weekend"
data_imputed$type_of_day[data_imputed$type_of_day != "weekend"] <- "weekday"
# convert type_of_day from character to factor
data_imputed$type_of_day <- as.factor(data_imputed$type_of_day)

# calculate average steps by interval across all days
df_imputed_steps_by_interval <- aggregate(steps ~ interval + type_of_day, data_imputed, mean)

# creat a plot
qplot(interval,
      steps,
      data = df_imputed_steps_by_interval,
      type = 'l',
      geom=c("line"),
      xlab = "Interval",
      ylab = "Number of steps",
      main = "") +
  facet_wrap(~ type_of_day, ncol = 1)
```

```
## Warning: Ignoring unknown parameters: type
```


weekday
weekend
Number of steps
0
50
100
150
200
0
50
100
150
200
0
500
1000
1500
2000
Interval