Reproducible-Research-Course-Project-1

Loading all necessary libraries:

```
library(ggplot2)
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
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.4     v purrr   0.3.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lattice)
```

Reading data:

```
csvAMD <- unzip("activity.zip")
AMD <- read.csv("activity.csv", sep = ",")
```

Change date to date format:

```
AMD$date <- as.Date(AMD$date)
```

Calculating the total number of steps taken per day:

```
sumAMD <- aggregate(steps ~ date, data = AMD, FUN = sum)
head((sumAMD))
```

```
##         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
```

Calculating mean and median of the total number of steps taken per day:

```
meanAMD <- mean(sumAMD$steps)
medianAMD <- median(sumAMD$steps)
meanAMD
```

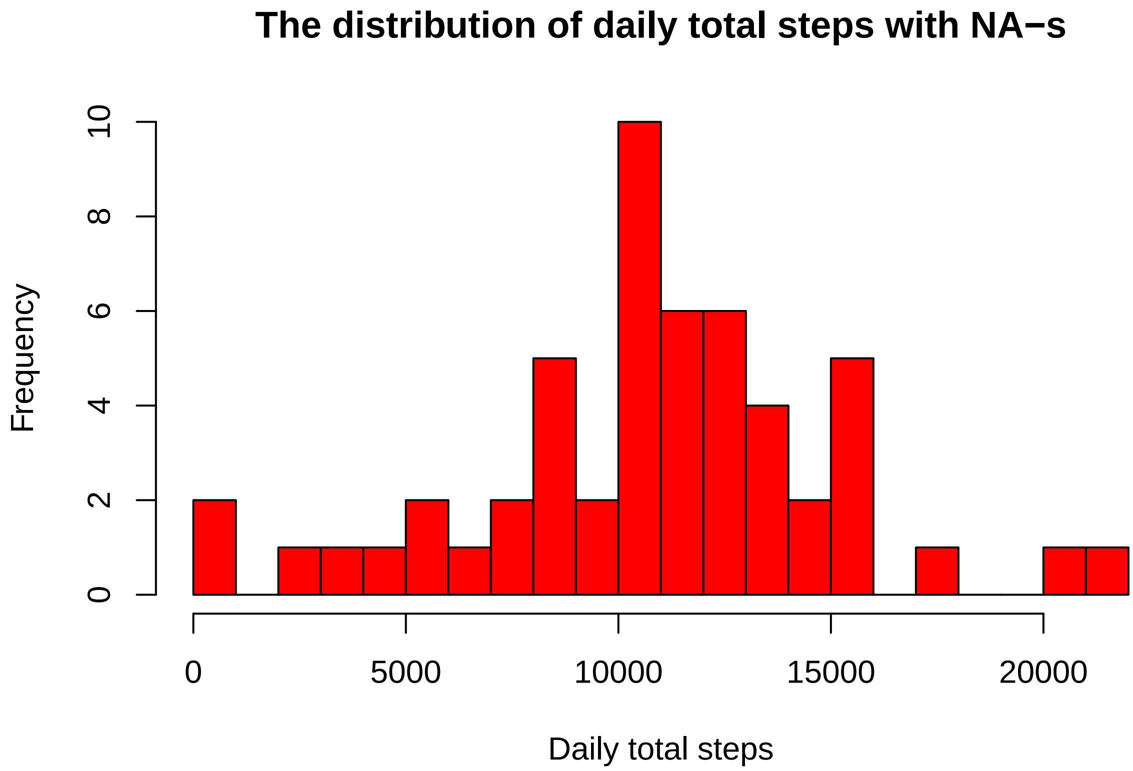
```
## [1] 10766.19
```

```
medianAMD
```

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

Creating a histogram from previously calculated data:

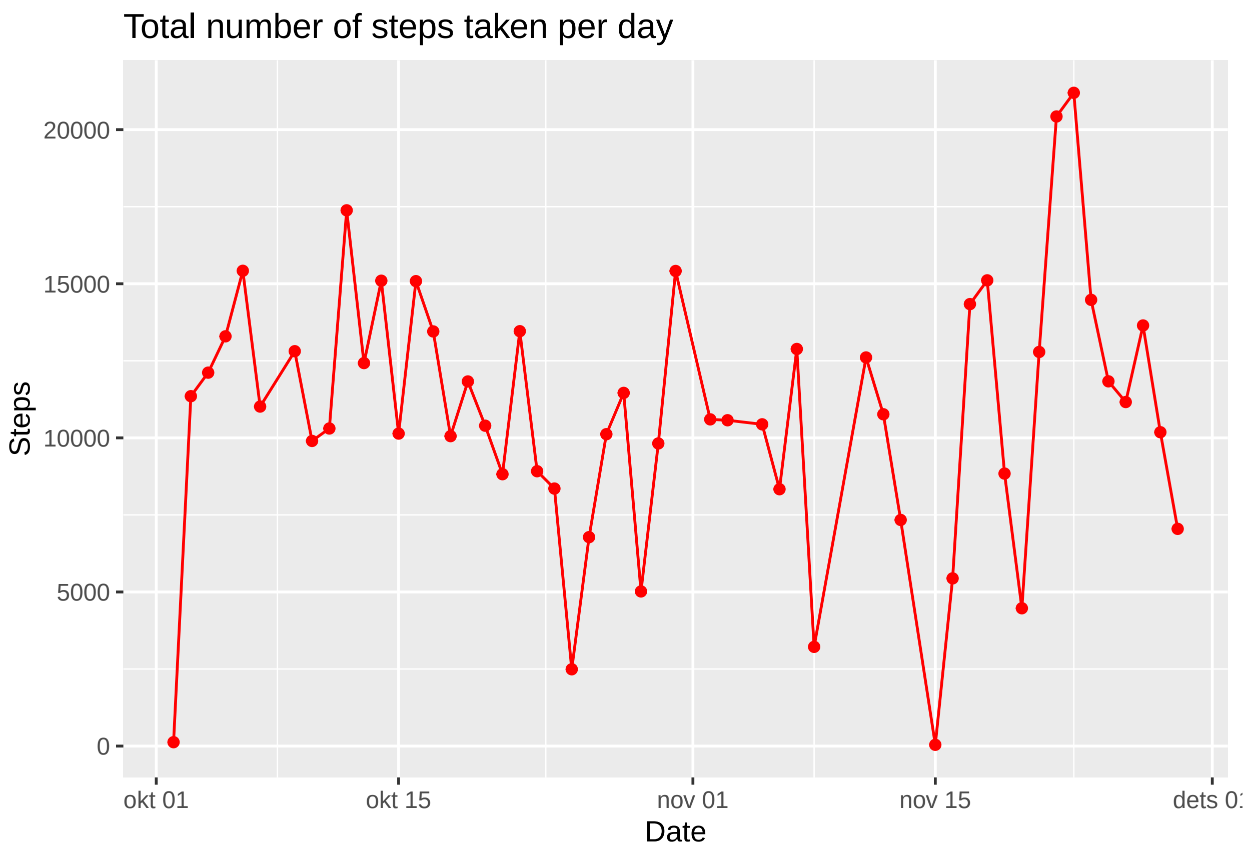
```
hist(x=sumAMD$steps,
     col="red",
     breaks=20,
     xlab="Daily total steps",
     ylab="Frequency",
     main="The distribution of daily total steps with NA-s")
```



This is not required for the assignment, but creating a graph with “ggplot2” for visual aid:

```
g <- ggplot(data = sumAMD, aes(x = date, y = steps))
g <- g + geom_line(color = "red")+geom_point(color = "red") +
          ylab("Steps") +
          xlab("Date") +
          ggtitle("Total number of steps taken per day")
```

```
g
```



What is the average daily activity pattern?

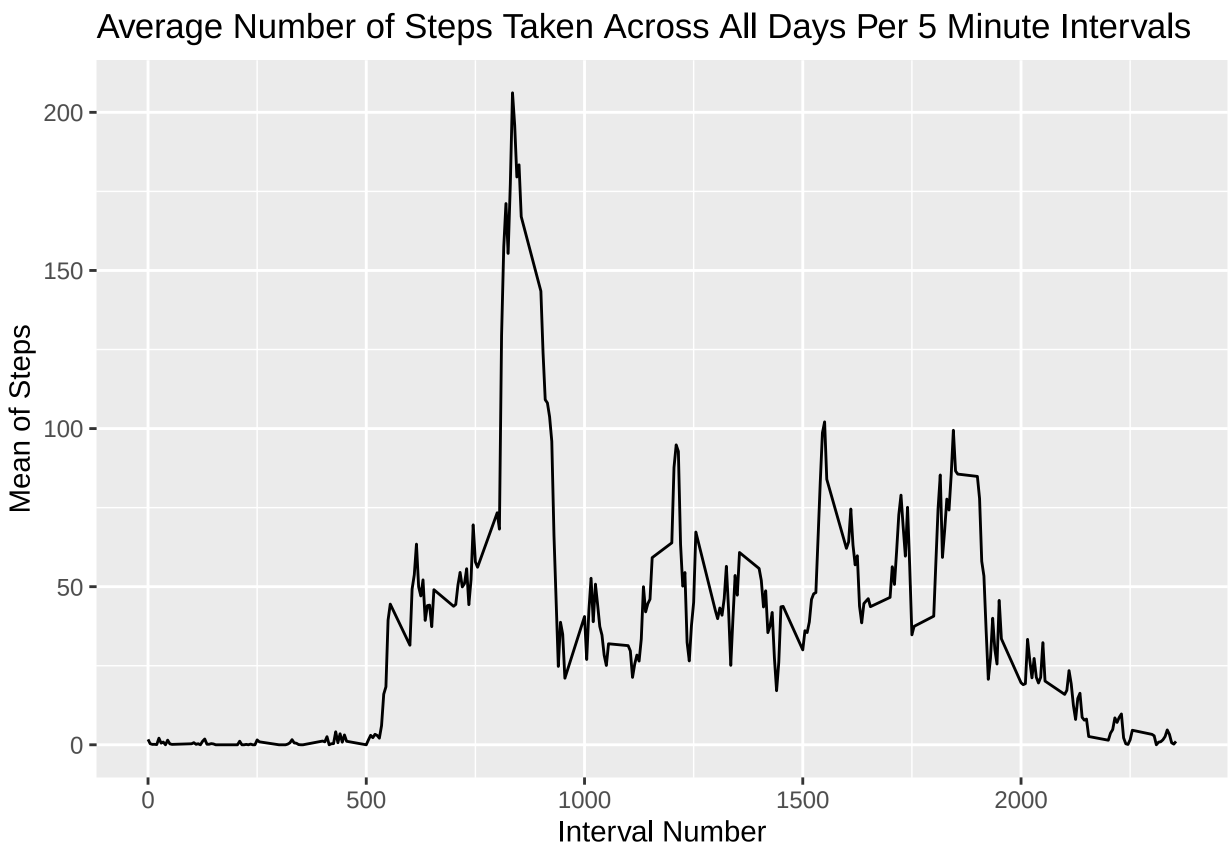
Getting the average steps across all days for each interval:

```
meanInterval <- aggregate(steps ~ interval, data = AMD, FUN = mean)
```

Creating the plot with “ggplot2”:

```
g2 <- ggplot(data = meanInterval, aes(x = interval, y= steps))
g2 <- g2 + geom_line() +
           ylab("Mean of Steps") +
           xlab("Interval Number") +
           ggtitle("Average Number of Steps Taken Across All Days Per 5 Minute Intervals")
```

```
g2
```



Which Interval on average across all days in the dataset contains the maximum number of steps?

```
whichMax <- which.max(meanInterval$steps)
whichMax
```

```
## [1] 104
```

What is the value of the interval?

```
meanInterval$steps[whichMax]
```

```
## [1] 206.1698
```

Imputing missing values

How many rows with NA-s?

```
summary(complete.cases(AMD))
```

```
##    Mode   FALSE    TRUE
## logical    2304   15264
```

Subsetting missing values

My strategy is to take mean across all days for given interval and replace NA-s with the mean:

1)Making a join with a original dataset("AMD") and calculated mean steps for intervals ("meanInterval")

```
JoinAMD <- left_join(AMD, meanInterval, by = "interval", all.x = TRUE)
```

2)Replacing missing values for steps with mean steps across all days("meanInterval")

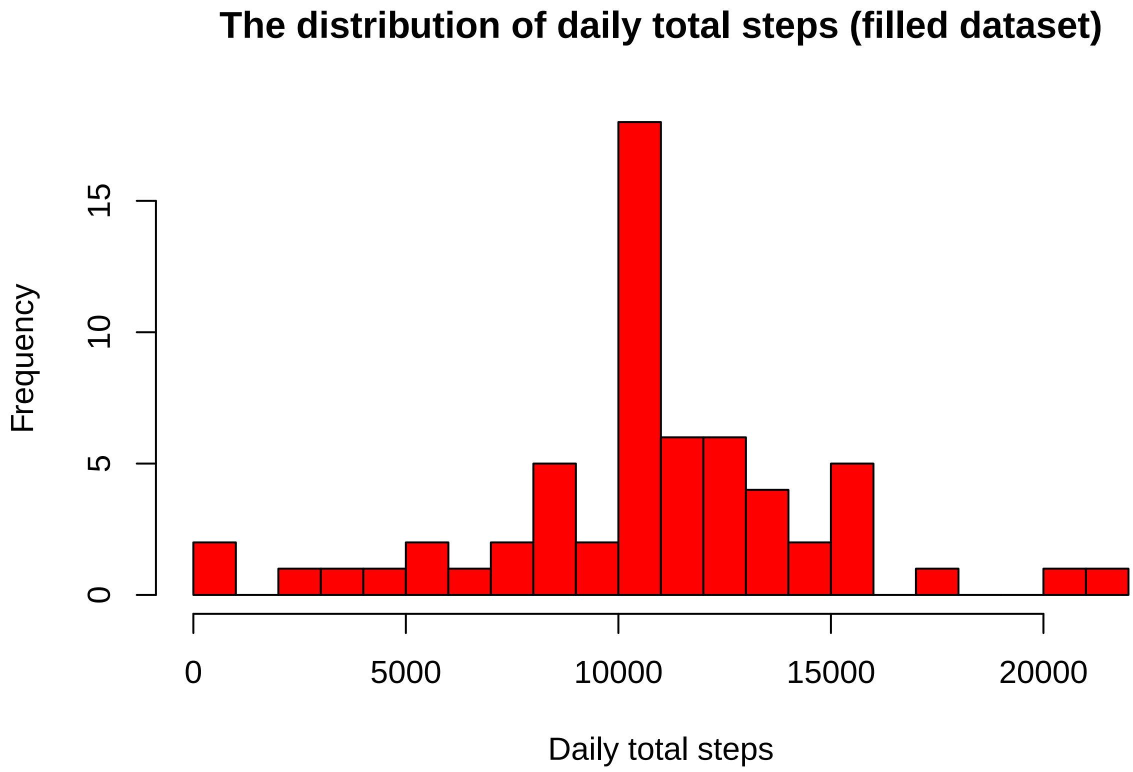
```
JoinAMD$steps.x <- coalesce(JoinAMD$steps.x, JoinAMD$steps.y)
```

Then we need to calculate the sum of each day for new dataset:

```
sumJoinAMD <- aggregate(steps.x ~ date, data = JoinAMD, FUN = sum)
```

Creating the histogram:

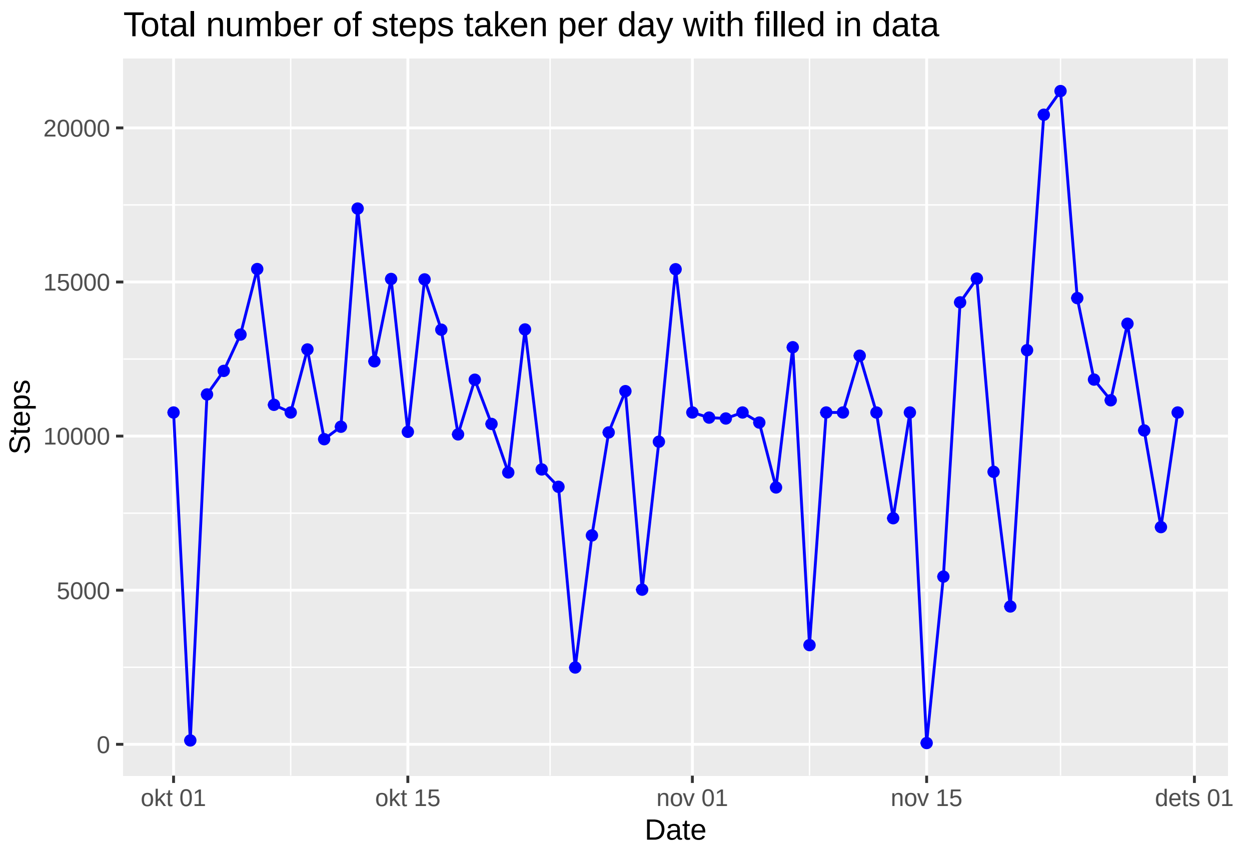
```
hist(x=sumJoinAMD$steps,
     col="red",
     breaks=20,
     xlab="Daily total steps",
     ylab="Frequency",
     main="The distribution of daily total steps (filled dataset)")
```



Again, just for my own curiosity lets see how it looks if we plot the new data with "ggplot2":

```
g3 <- ggplot(data = sumJoinAMD, aes(x = date, y = steps.x))
g3 <- g3 + geom_line(color ="blue")+geom_point(color ="blue") +
          ylab("Steps") +
          xlab("Date")
g3title <- g3 + ggtitle("Total number of steps taken per day with filled in data")
```

```
g3title
```



Calculating mean and median of the total number of steps taken per day:

```
meanJoinAMD <- mean(sumJoinAMD$steps.x)
medianJoinAMD <- median(sumJoinAMD$steps.x)

meanJoinAMD
```

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

```
medianJoinAMD
```

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

Comparing to original dataset:

```
meanAMD
```

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

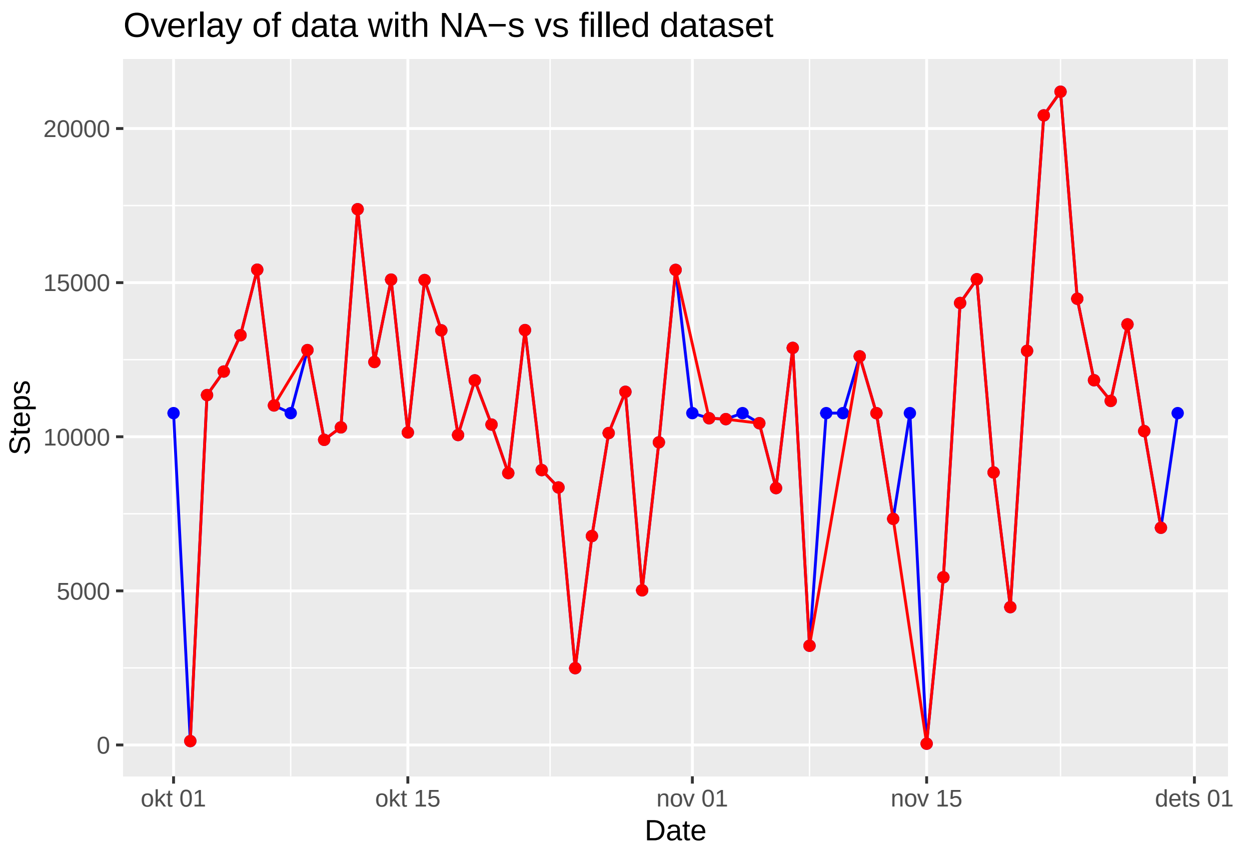
```
medianAMD
```

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

How does original dataset with NA-s (AMD) compare to new filled in dataset (JoinAMD)?

```
g4 <- g3 + geom_line(data = sumAMD, aes(x = date, y = steps), color= "red") +
           geom_point(data = sumAMD, aes(x = date, y = steps), color= "red") +
           ggtitle("Overlay of data with NA-s vs filled dataset")
```

```
g4
```



Are there differences in activity patterns between weekdays and weekends?

Creating a new factor weekday using weekdays()

```
JoinAMD$weekday <- as.factor(weekdays(JoinAMD$date))
```

Adding a new column “DayType” as 2 level factor c(“weekday”,"weekend)

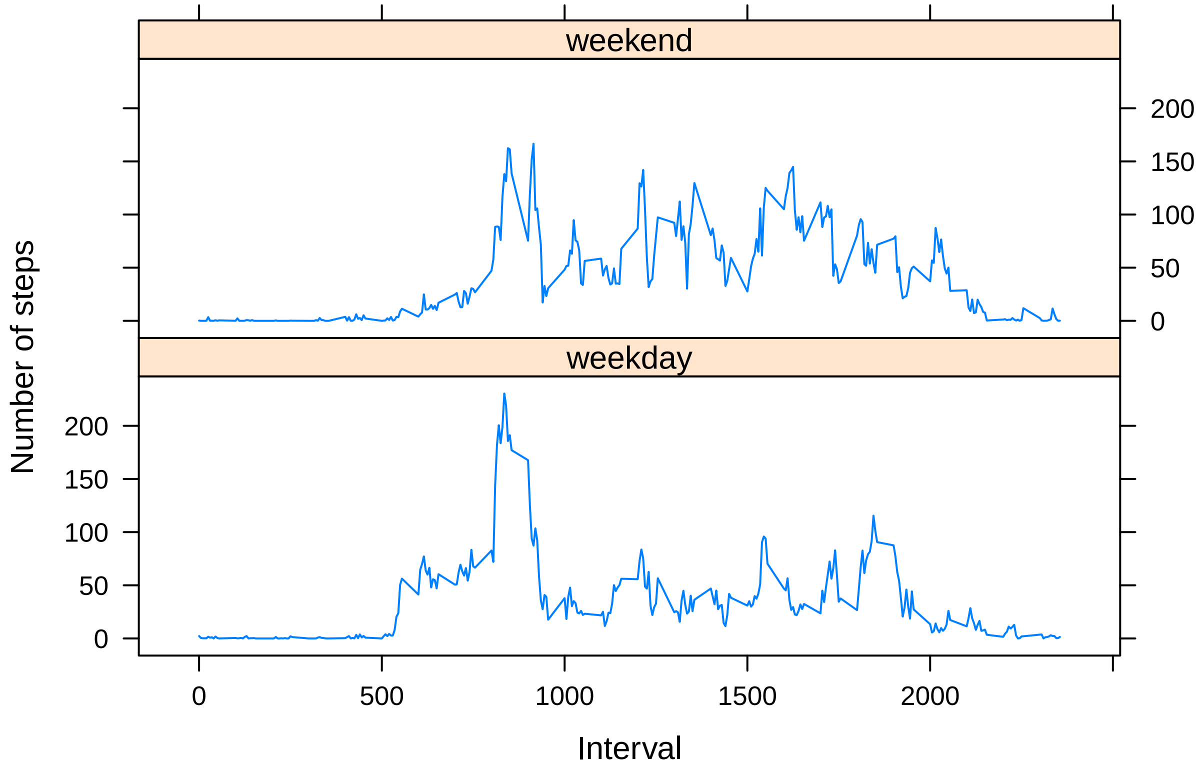
```
JoinAMD$DayType <- as.factor(ifelse(JoinAMD$weekday==’laupäev’ | JoinAMD$weekday==’pühapäev’, ’weekend’
```

Calculating the means for factors and creating a new data frame:

```
WkAMD <- aggregate(steps.x ~ DayType+interval, data=JoinAMD, FUN=mean)
```

Creating the plot using “lattice” library:

```
xyplot(steps.x ~ interval | DayType,
       layout = c(1, 2),
       xlab="Interval",
       ylab="Number of steps",
       type="l",
       lty=1,
       data=WkAMD)
```


weekend
weekday
Number of steps
Interval