Regression Models Course Project

Regression Models Course Project

Reading the dataset and processing the data

```
activity <- read.csv("activity.csv")
str(activity)
```

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

```
activity$date <- as.Date(activity$date)
head(activity)
```

```
##   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
```

Dropping NA rows

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'forcats' was built under R version 4.1.2
```

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

```
activity <- drop_na(activity, steps)
sum(is.na(activity$steps))
```

```
## [1] 0
```

Histogram of the total number of steps taken each day

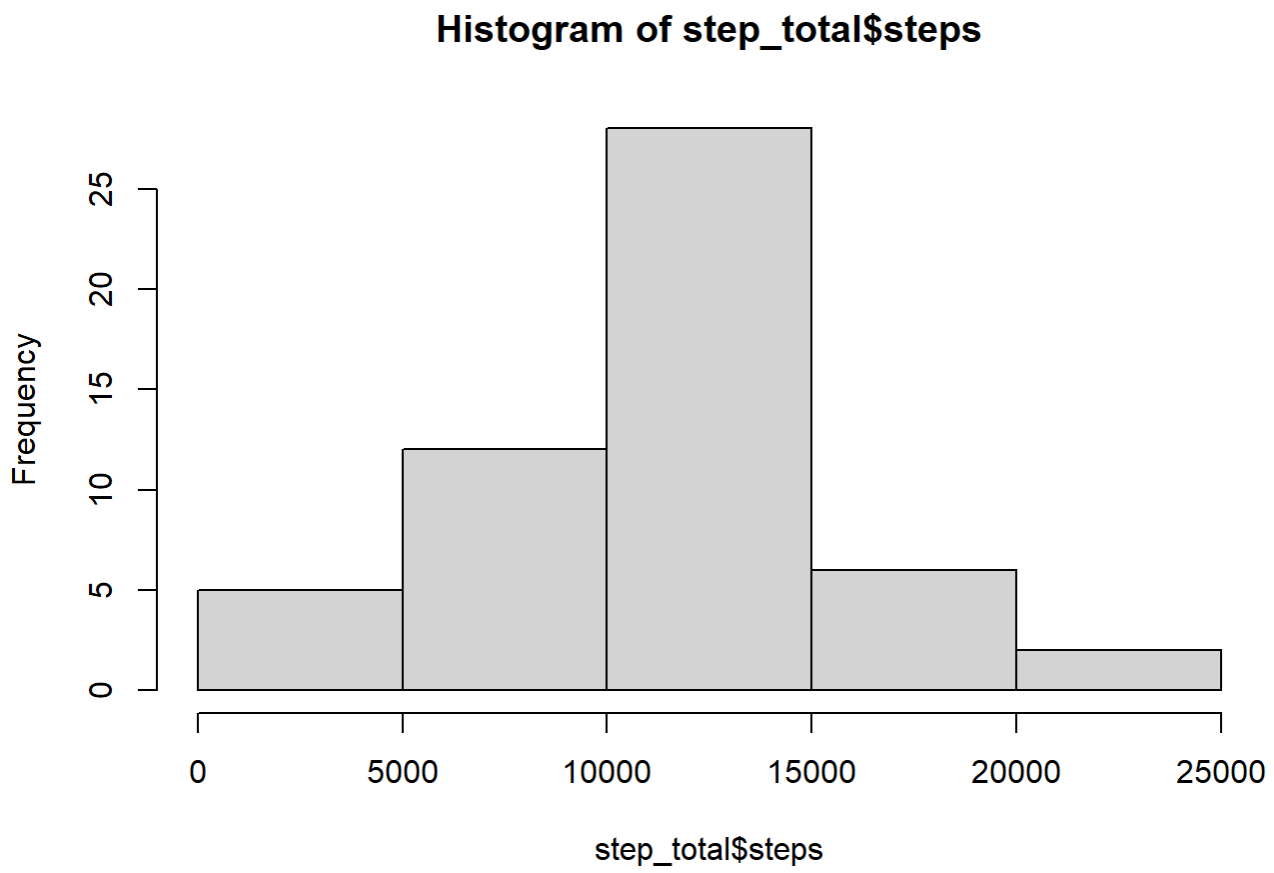
```
step_total <- aggregate(steps~date,activity,sum)
head(step_total)
```

```
##         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
```

```
dim(step_total)
```

```
## [1] 53  2
```

```
hist(step_total$steps)
```



Mean and median number of steps taken each day

```
mean_steps <- aggregate(steps~date,activity,mean)
mean_steps
```

```
##          date      steps
## 1  2012-10-02  0.4375000
## 2  2012-10-03 39.4166667
## 3  2012-10-04 42.0694444
## 4  2012-10-05 46.1597222
## 5  2012-10-06 53.5416667
## 6  2012-10-07 38.2465278
## 7  2012-10-09 44.4826389
## 8  2012-10-10 34.3750000
## 9  2012-10-11 35.7777778
## 10 2012-10-12 60.3541667
## 11 2012-10-13 43.1458333
## 12 2012-10-14 52.4236111
## 13 2012-10-15 35.2048611
## 14 2012-10-16 52.3750000
## 15 2012-10-17 46.7083333
## 16 2012-10-18 34.9166667
## 17 2012-10-19 41.0729167
## 18 2012-10-20 36.0937500
## 19 2012-10-21 30.6284722
## 20 2012-10-22 46.7361111
## 21 2012-10-23 30.9652778
## 22 2012-10-24 29.0104167
## 23 2012-10-25  8.6527778
## 24 2012-10-26 23.5347222
## 25 2012-10-27 35.1354167
## 26 2012-10-28 39.7847222
## 27 2012-10-29 17.4236111
## 28 2012-10-30 34.0937500
## 29 2012-10-31 53.5208333
## 30 2012-11-02 36.8055556
## 31 2012-11-03 36.7048611
## 32 2012-11-05 36.2465278
## 33 2012-11-06 28.9375000
## 34 2012-11-07 44.7326389
## 35 2012-11-08 11.1770833
## 36 2012-11-11 43.7777778
## 37 2012-11-12 37.3784722
## 38 2012-11-13 25.4722222
## 39 2012-11-15  0.1423611
## 40 2012-11-16 18.8923611
## 41 2012-11-17 49.7881944
## 42 2012-11-18 52.4652778
## 43 2012-11-19 30.6979167
## 44 2012-11-20 15.5277778
## 45 2012-11-21 44.3993056
## 46 2012-11-22 70.9270833
## 47 2012-11-23 73.5902778
## 48 2012-11-24 50.2708333
## 49 2012-11-25 41.0902778
## 50 2012-11-26 38.7569444
## 51 2012-11-27 47.3819444
```

```
## 52 2012-11-28 35.3576389
## 53 2012-11-29 24.4687500
```

```
median_steps <- aggregate(steps~date,activity,median)
median_steps
```

```
##          date steps
## 1  2012-10-02     0
## 2  2012-10-03     0
## 3  2012-10-04     0
## 4  2012-10-05     0
## 5  2012-10-06     0
## 6  2012-10-07     0
## 7  2012-10-09     0
## 8  2012-10-10     0
## 9  2012-10-11     0
## 10 2012-10-12     0
## 11 2012-10-13     0
## 12 2012-10-14     0
## 13 2012-10-15     0
## 14 2012-10-16     0
## 15 2012-10-17     0
## 16 2012-10-18     0
## 17 2012-10-19     0
## 18 2012-10-20     0
## 19 2012-10-21     0
## 20 2012-10-22     0
## 21 2012-10-23     0
## 22 2012-10-24     0
## 23 2012-10-25     0
## 24 2012-10-26     0
## 25 2012-10-27     0
## 26 2012-10-28     0
## 27 2012-10-29     0
## 28 2012-10-30     0
## 29 2012-10-31     0
## 30 2012-11-02     0
## 31 2012-11-03     0
## 32 2012-11-05     0
## 33 2012-11-06     0
## 34 2012-11-07     0
## 35 2012-11-08     0
## 36 2012-11-11     0
## 37 2012-11-12     0
## 38 2012-11-13     0
## 39 2012-11-15     0
## 40 2012-11-16     0
## 41 2012-11-17     0
## 42 2012-11-18     0
## 43 2012-11-19     0
## 44 2012-11-20     0
## 45 2012-11-21     0
## 46 2012-11-22     0
## 47 2012-11-23     0
## 48 2012-11-24     0
## 49 2012-11-25     0
## 50 2012-11-26     0
## 51 2012-11-27     0
```

```
## 52 2012-11-28     0
## 53 2012-11-29     0
```

Time series plot of the average number of steps taken based on 5-minute interval

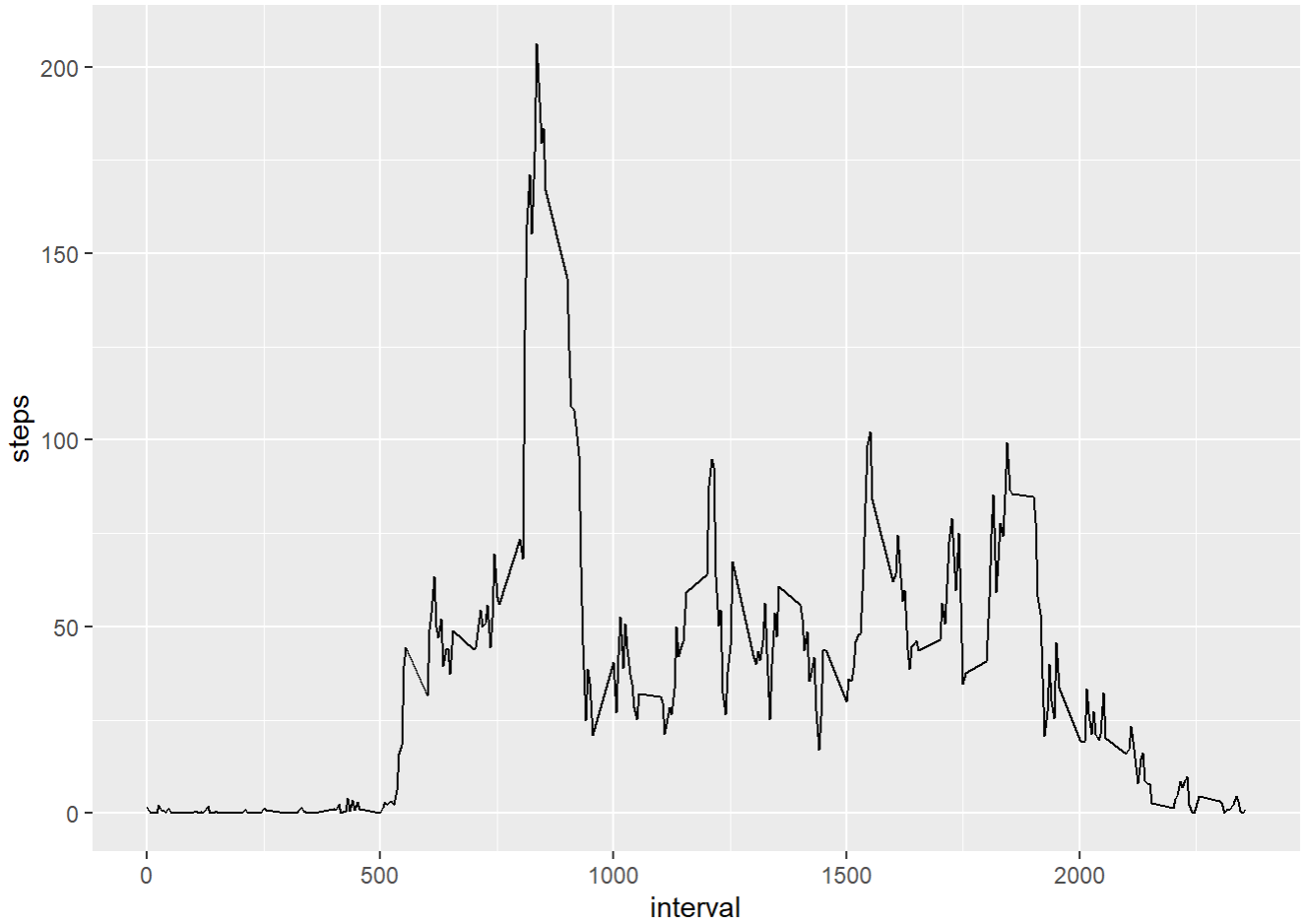
```
library(ggplot2)
tsdata <- aggregate(steps~interval,activity,mean)
head(tsdata)
```

```
##   interval     steps
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396
```

```
dim(tsdata)
```

```
## [1] 288   2
```

```
ggplot(data = tsdata, aes(x = interval, y = steps)) + geom_line()
```



The time interval between 750 - 1000 had the most number of steps.

Code to describe and show a strategy for imputing missing data

```
imputed_activity <- activity
```

Replacing all the NA values with the mean of steps w.r.t their corresponding intervals

```
imputed_activity[which(is.na(imputed_activity$steps))] <- tsdata$steps[imputed_activity$interval
== tsdata$interval]

sum(is.na(imputed_activity))
```

```
## [1] 0
```

```
head(imputed_activity)
```

```
##   steps       date interval
## 1     0 2012-10-02        0
## 2     0 2012-10-02        5
## 3     0 2012-10-02       10
## 4     0 2012-10-02       15
## 5     0 2012-10-02       20
## 6     0 2012-10-02       25
```

```
dim(imputed_activity)
```

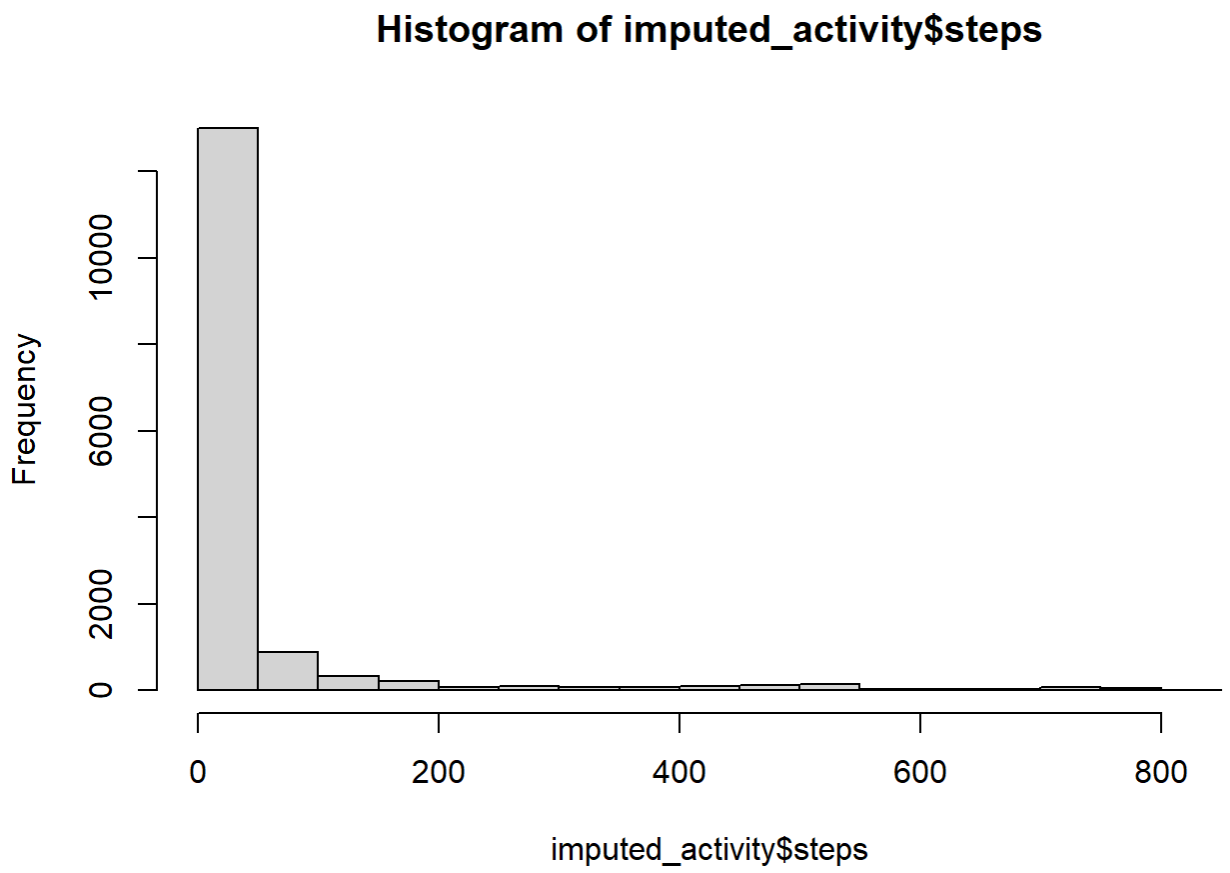
```
## [1] 15264     3
```

```
imputed_activity$steps <- ifelse(is.na(imputed_activity$steps), tsdata$steps, imputed_activity$s
teps)
sum(is.na(imputed_activity$steps))
```

```
## [1] 0
```

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

```
hist(imputed_activity$steps)
```



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

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.1.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
imputed_activity$day <- weekdays(activity$date)
```

Creating dataframes grouped by weekends and weekdays to create a panel plot

```
weekend_data <- subset(imputed_activity, day == c("Saturday", "Sunday"), select = c(steps, date,
interval, day))
head(weekend_data)
```

```
##      steps       date interval      day
## 1153     0 2012-10-06        0 Saturday
## 1155     0 2012-10-06       10 Saturday
## 1157     0 2012-10-06       20 Saturday
## 1159     0 2012-10-06       30 Saturday
## 1161     0 2012-10-06       40 Saturday
## 1163     0 2012-10-06       50 Saturday
```

```
unique(weekend_data$day)
```

```
## [1] "Saturday" "Sunday"
```

```
weekday_data <- subset(imputed_activity, day == c("Monday", "Tuesday", "Wednesday", "Thursday",
"Friday"), select = c(steps, date, interval, day))
```

```
## Warning in day == c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"):
## longer object length is not a multiple of shorter object length
```

```
head(weekday_data)
```

```
##    steps       date interval     day
## 2      0 2012-10-02        5 Tuesday
## 7      0 2012-10-02       30 Tuesday
## 12     0 2012-10-02       55 Tuesday
## 17     0 2012-10-02      120 Tuesday
## 22     0 2012-10-02      145 Tuesday
## 27     0 2012-10-02      210 Tuesday
```

```
unique(weekday_data$day)
```

```
## [1] "Tuesday"   "Wednesday" "Thursday"  "Friday"    "Monday"
```

Panel plot

```
par(mfrow=c(2,1))
plot(y = weekend_data$steps, x = weekend_data$interval, type = "l")
plot(y = weekday_data$steps, x = weekday_data$interval, type = "l")
```