Reproducible Research: Peer Assessment 1

Anh

8/20/2021

Loading and preprocessing the data

Load 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(lattice)
```

Read the file

```
activity = read.csv("activity.csv")
```

Process datetime column of the data

```
activity$fixed_date = as.Date(activity$date)
activity$week_day = weekdays(activity$fixed_date)
```

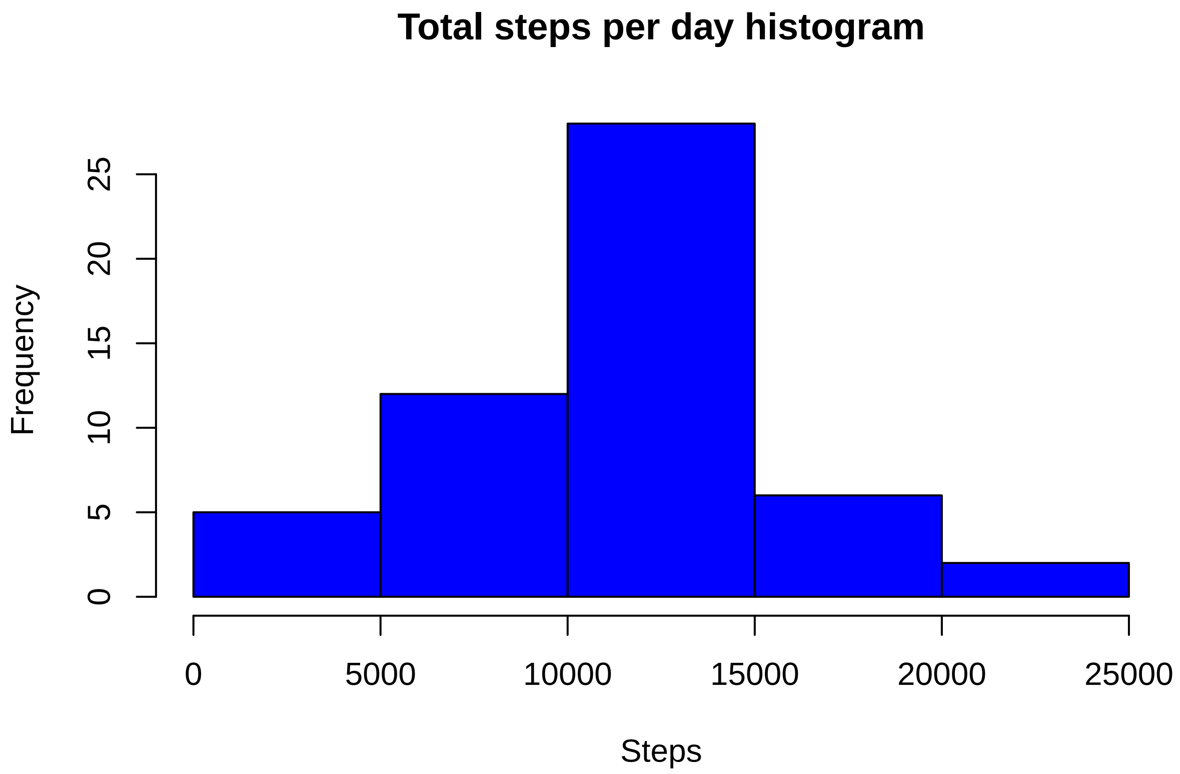
What is mean total number of steps taken per day?

Calculate the total number of steps taken per date (fixed_date)

```
total_steps = activity %>%
                group_by(fixed_date) %>%
                summarise(sum(steps))
```

Create a histogram of the total number of steps per day

```
hist(total_steps$`sum(steps)`, breaks = 5, xlab = "Steps", main =
       "Total steps per day histogram", col = "blue")
```



Calculate the mean and median of the total number of steps per day

```
(mean(total_steps$`sum(steps)`, na.rm = TRUE))
```

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

```
(median(total_steps$`sum(steps)`, na.rm = TRUE))
```

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

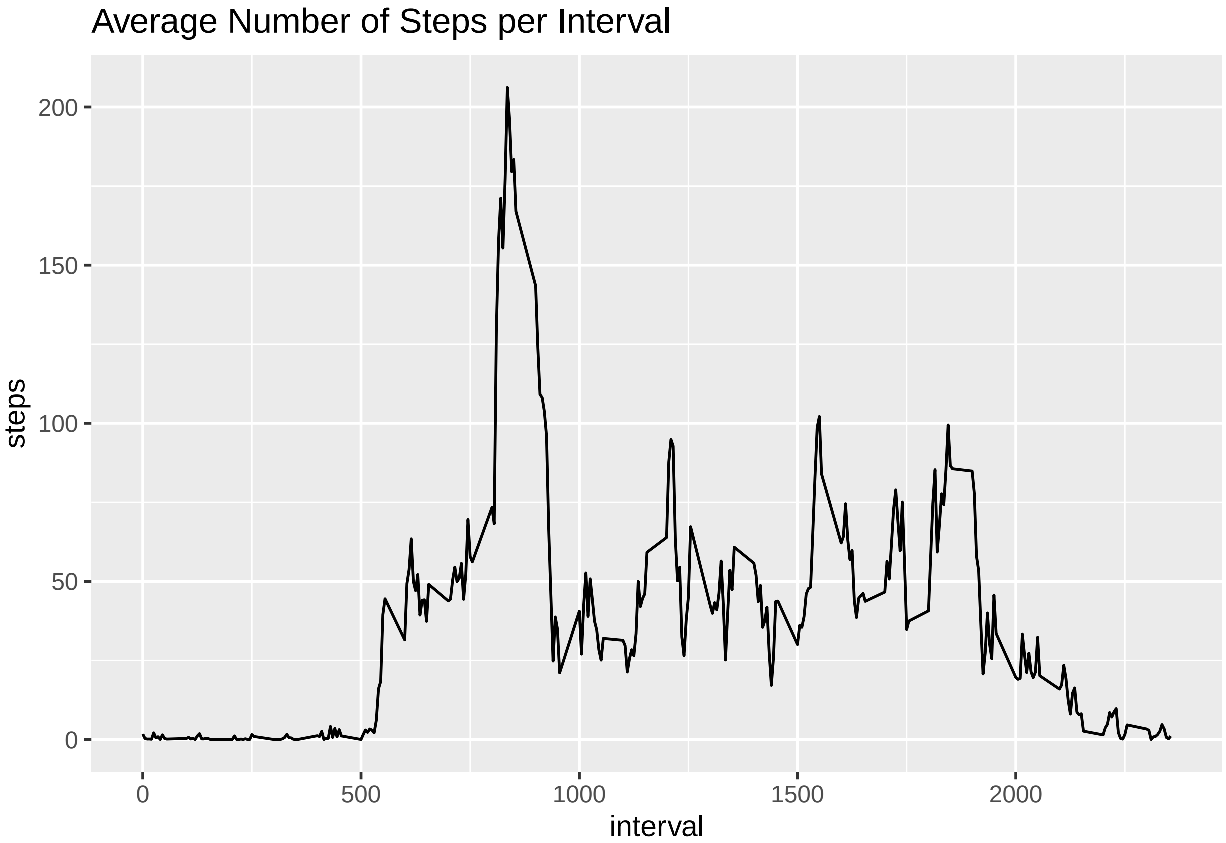
What is the average daily activity pattern?

Create a dataset of steps in 5-minute interval and take their average

```
intervals <- aggregate(steps ~ interval, activity, mean)
```

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)

```
ggplot(intervals, aes(x = interval, y = steps), xlab = "Intervals", ylab =
         "Average number of steps") +
  geom_line() +
  ggtitle("Average Number of Steps per Interval")
```



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

```
max_steps = max(intervals$steps)
(intervals[intervals$steps == max_steps, ])
```

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

Imputing missing values

Calculate the total number of missing values in the dataset

```
nrow(activity[is.na(activity$steps), ])
```

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

For this exercise, I have chosen to impute NA values by the mean of the 5-minute intervals of the weekdays.

```
activity_wo_na = activity %>%
  group_by(week_day, interval) %>%
  mutate(steps = ifelse(is.na(steps), mean(steps, na.rm = TRUE), steps))
```

Aggregate the total number of steps per day in the new dataset with NAs imputed

```
total_steps_wo_na = activity_wo_na %>%
     group_by(fixed_date) %>%
     summarise(sum(steps))
```

```
(mean(total_steps_wo_na$`sum(steps)`))
```

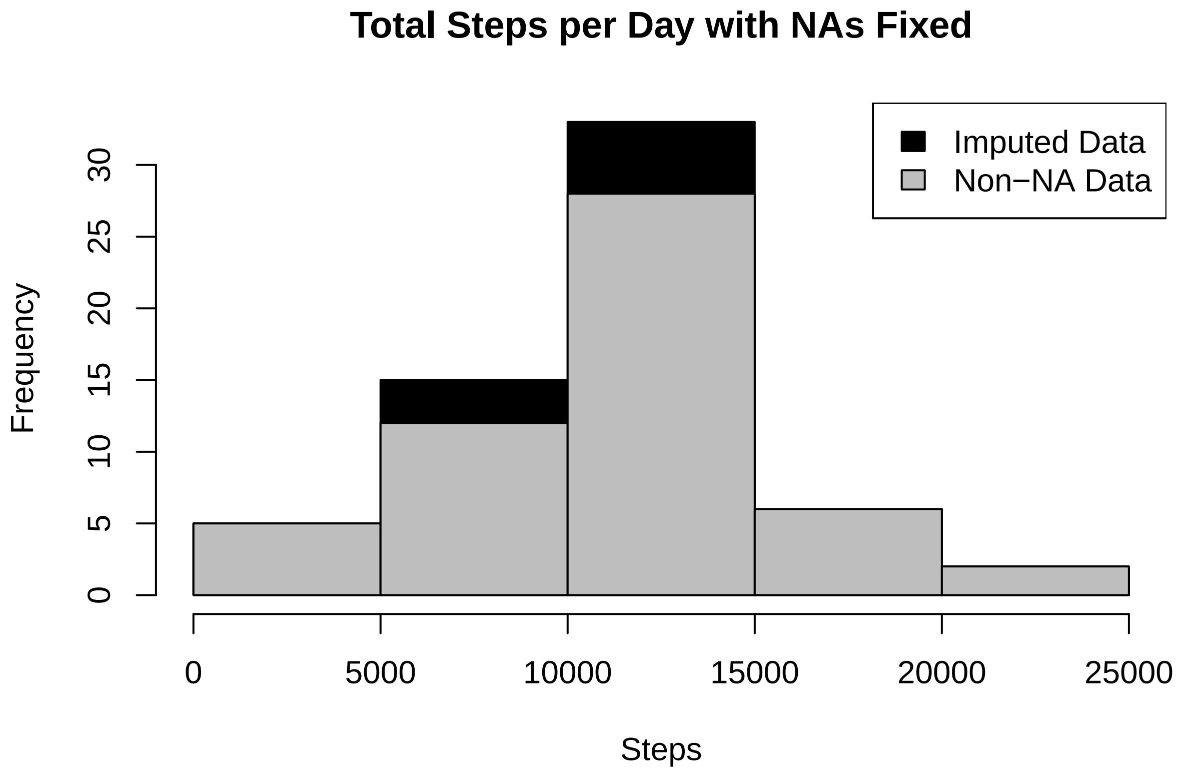
```
## [1] 10821.21
```

```
(median(total_steps_wo_na$`sum(steps)`))
```

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

Create a histogram of total number of steps per day, categorised by two datasets (one with NAs and one with NAs imputed)

```
hist(total_steps_wo_na$`sum(steps)`, breaks = 5, xlab = "Steps",
     main = "Total Steps per Day with NAs Fixed",col = "Black")
hist(total_steps$`sum(steps)`, breaks = 5, xlab = "Steps",
     main = "Total Steps per Day with NAs Fixed", col = "Grey", add = TRUE)
legend("topright", c("Imputed Data", "Non-NA Data"), fill=  c("black", "grey"))
```



Are there differences in activity patterns between weekdays and weekends?

Create a new column to determine if the week_day is weekday or weekend

```
activity_wo_na$week_day_or_weekend = ifelse(activity_wo_na$week_day %in%
                                              c("Saturday", "Sunday"),
                                            "weekend", "weekday")
```

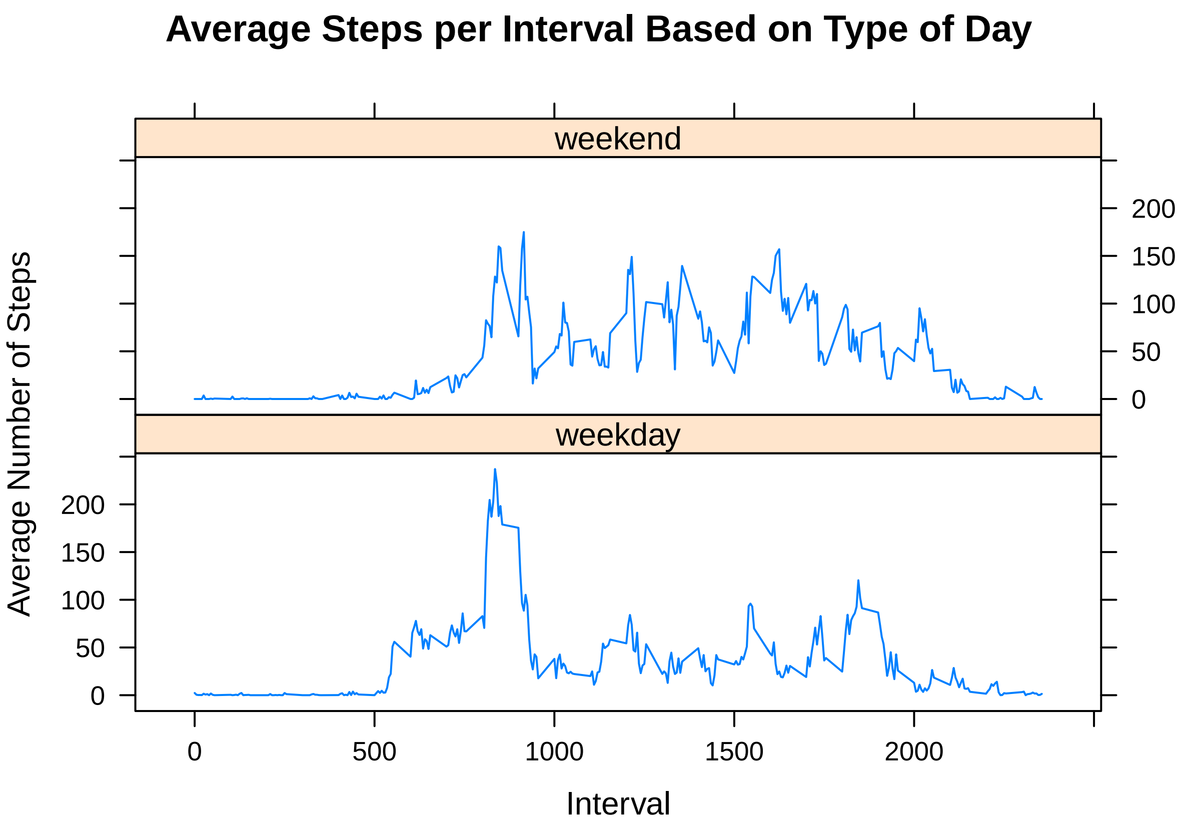
Summarise the number of steps by intervals and weekday category (weekdays or weekend)

```
by_intervals_weekday = activity_wo_na %>%
                        group_by(interval, week_day_or_weekend) %>%
                        summarise(avg = mean(steps))
```

```
## ‘summarise()‘ has grouped output by ’interval’. You can override using the ‘.groups‘ argument.
```

Plot

```
xyplot(avg ~ interval|week_day_or_weekend, data = by_intervals_weekday,
       type = "l", layout = c(1,2),
       main = "Average Steps per Interval Based on Type of Day",
       ylab = "Average Number of Steps", xlab = "Interval")
```


Average Steps per Interval Based on Type of Day
weekend
weekday
Average Number of Steps
Interval
0
50
100
150
200
0
500
1000
1500
2000