

Reproducible Research: Peer Assessment 1

Cristina Angel

16/10/2019

Peer Assessment 1

Loading and preprocessing the data

Load the data and process/transform the data (if necessary) into a format suitable for your analysis:

```
library(tidyverse)
```

```
Sys.setlocale("LC_TIME", "C")
```

```
## [1] "C"
```

```
amd <- read_delim('./data/raw/activity.csv', delim = ',')
```

What is mean total number of steps taken per day?

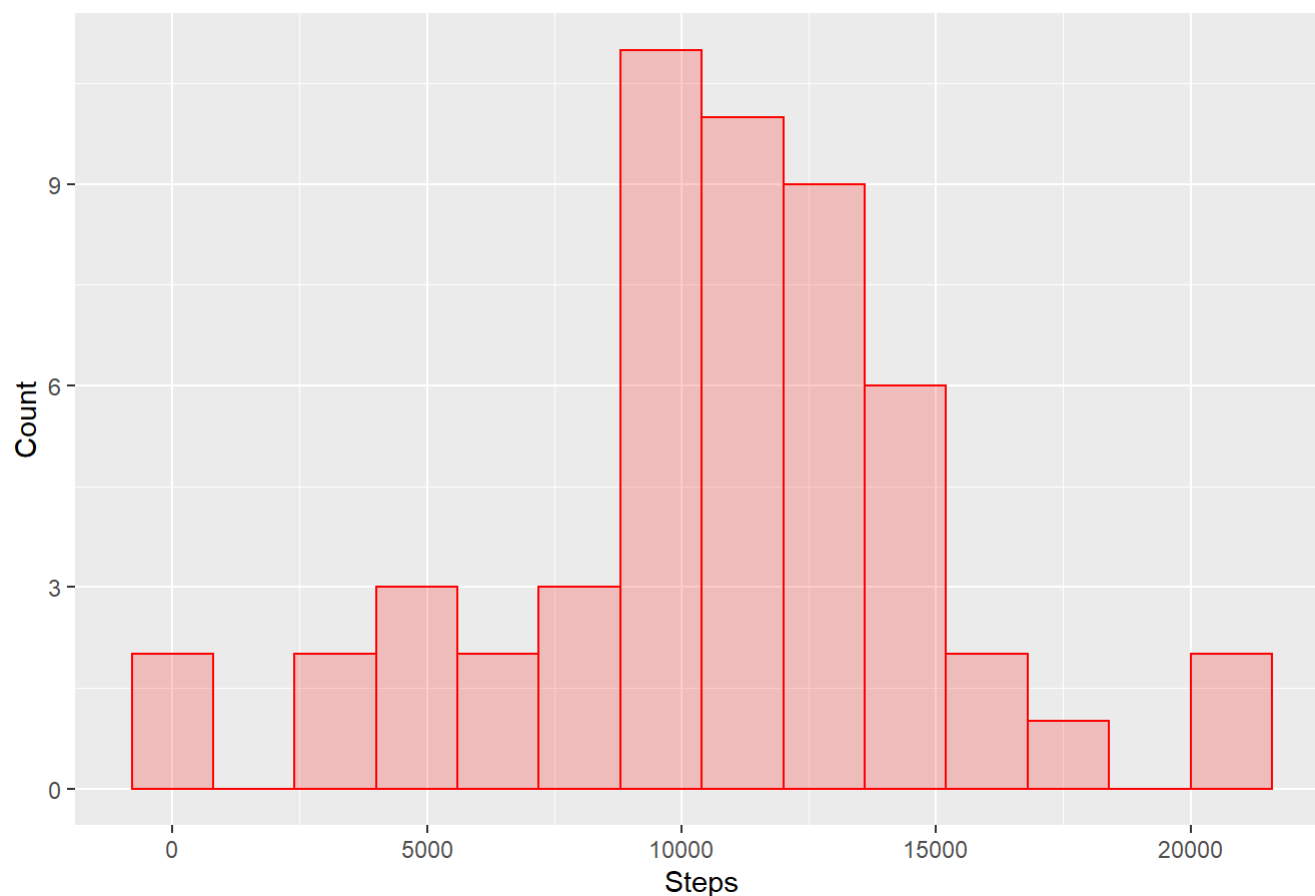
For this part of the assignment, I ignore the missing values in the dataset

Calculate the total number of steps taken per day:

```
daily <- amd %>% filter(!is.na(steps)) %>%  
  group_by(date) %>%  
  summarise(daily_steps = sum(steps))
```

Histogram of the total number of steps taken each day:

Histogram of Daily Steps



Mean and median number of steps taken each day:

```
median(daily$daily_steps)
```

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

```
mean(daily$daily_steps)
```

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

As you see the Median and Mean daily steps are 10765 and 10766

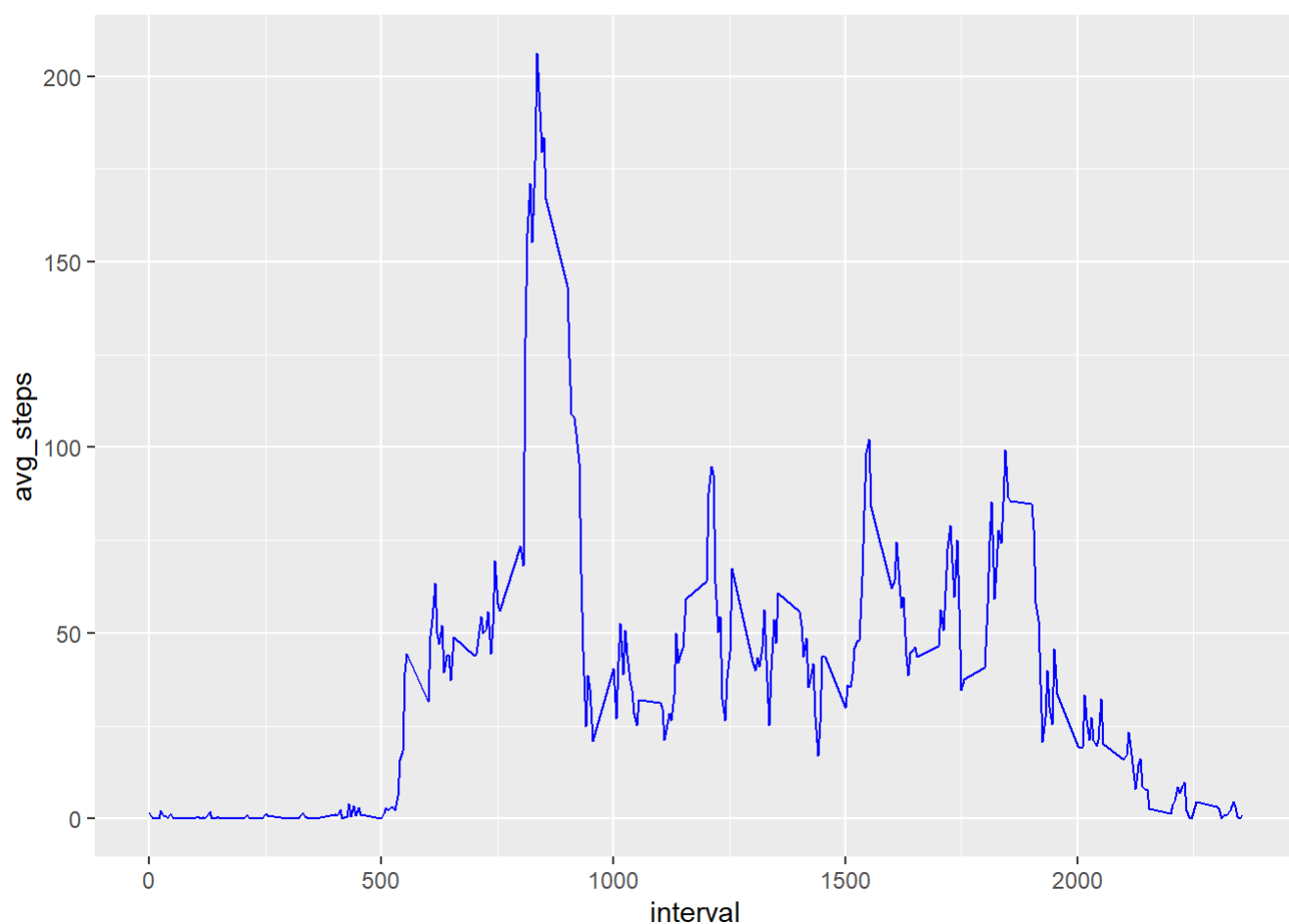
What is mean total number of steps taken per day?

```
interval <- amd %>%  
  filter(!is.na(steps)) %>%  
  group_by(interval) %>%  
  summarise(avg_steps = mean(steps))  
  
head(interval)
```

```
## # A tibble: 6 x 2
##   interval avg_steps
##   <dbl>     <dbl>
## 1      0      1.72
## 2      5      0.340
## 3     10      0.132
## 4     15      0.151
## 5     20      0.0755
## 6     25      2.09
```

What is the average daily activity pattern?

Time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis):



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

```
max <- interval %>% filter(max(avg_steps) == avg_steps)

max
```

```
## # A tibble: 1 x 2
##   interval avg_steps
##   <dbl>     <dbl>
## 1     835       206.
```

The interval with the maximum number of steps is 835

Imputing missing values

Calculate and report the total number of missing values in the dataset:

```
amd %>% filter(is.na(steps)) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  2304
```

The total number of missing values is 2304

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 and create a new dataset that is equal to the original dataset but with the missing data filled in:

```
new_amd <- amd %>%
  group_by(interval) %>%
  mutate(steps = replace(steps, is.na(steps), mean(steps, na.rm = T))) %>%
  ungroup()

head(new_amd)
```

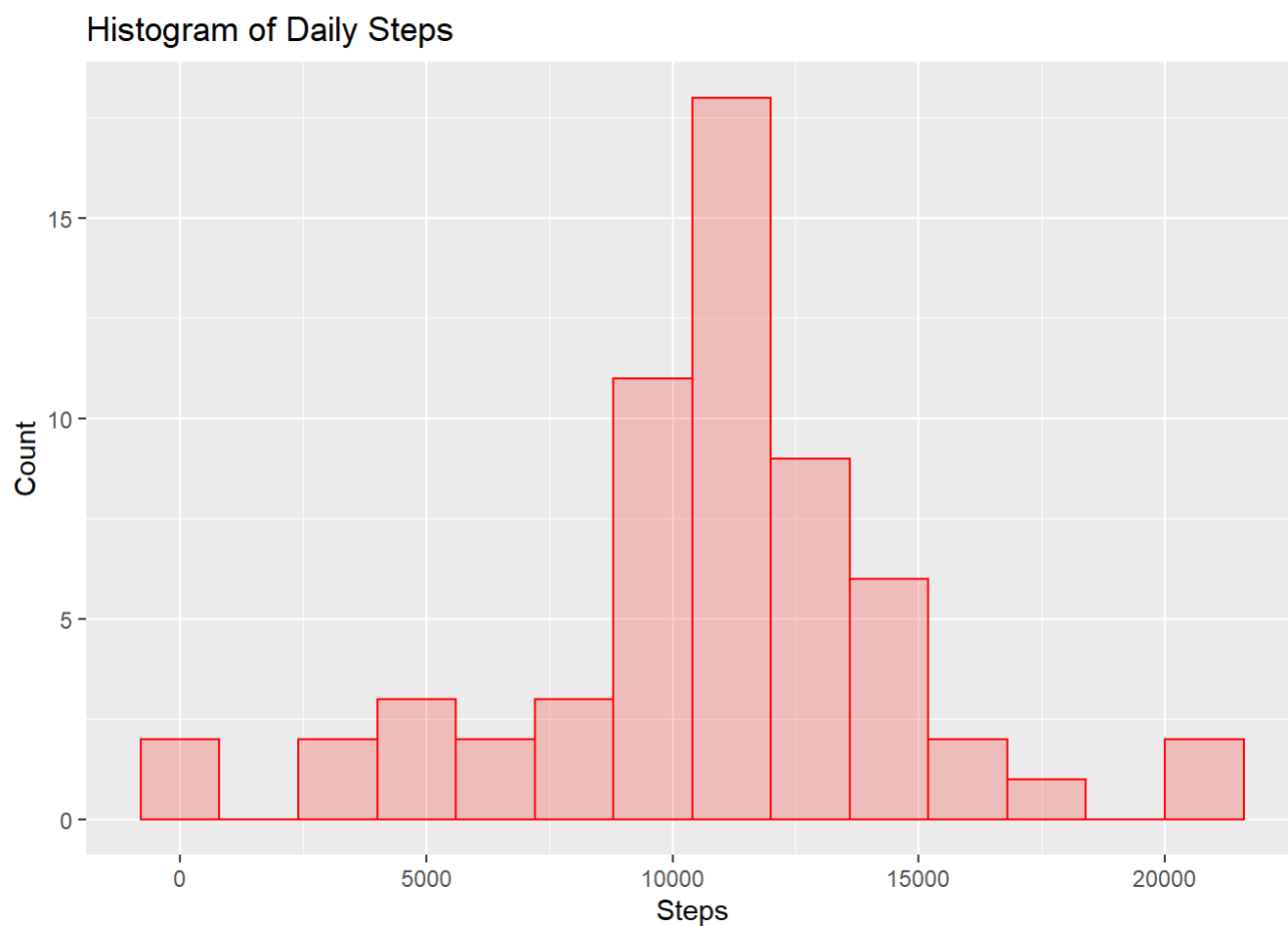
```
## # A tibble: 6 x 3
##   steps date      interval
##   <dbl> <date>     <dbl>
## 1 1.72  2012-10-01      0
## 2 0.340 2012-10-01      5
## 3 0.132 2012-10-01     10
## 4 0.151 2012-10-01     15
## 5 0.0755 2012-10-01     20
## 6 2.09  2012-10-01     25
```

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?

Calculate the total number of steps taken per day:

```
new_daily <- new_amd %>% filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarise(daily_steps = sum(steps))
```

Histogram of the total number of steps taken each day:



Mean and median number of steps taken each day:

```
median(new_daily$daily_steps)
```

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

```
mean(new_daily$daily_steps)
```

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

As you see the Median and Mean daily steps are 10766 and 10766

```
(median(new_daily$daily_steps) - median(daily$daily_steps))/median(daily$daily_steps)*100
```

```
## [1] 0.01104207
```

```
(mean(new_daily$daily_steps) - mean(daily$daily_steps))/mean(daily$daily_steps)*100
```

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

When we input missing values with the mean of each interval, just the median changes in 0.011%

Are there differences in activity patterns between weekdays and weekends?

```
new_amd <- new_amd %>% mutate(weekdays = weekdays(date),
                             day_class = as_factor(if_else(weekdays %in% c('Saturday', 'Sunday'),
                             'Weekend' , 'Weekday')))
```

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

```
ggplot(new_amd %>% group_by(day_class, interval) %>%
  summarise(avg_steps = mean(steps)),
  aes(x = interval , y = avg_steps)) +
  geom_line(color = 'blue') +
  facet_grid(day_class ~ .) +
  ylab("The Average Number of Steps Taken") +
  xlab("the 5-minute Interval") +
  ggtitle("The Average Number of Steps Taken on the 5-minute Interval")
```

The Average Number of Steps Taken on the 5-minute Interval

