hello

Irina

31/05/2020

Loading and preprocessing the data

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

#change date to right format
activity$date <- ymd(activity$date)
```

What is mean total number of steps taken per day?

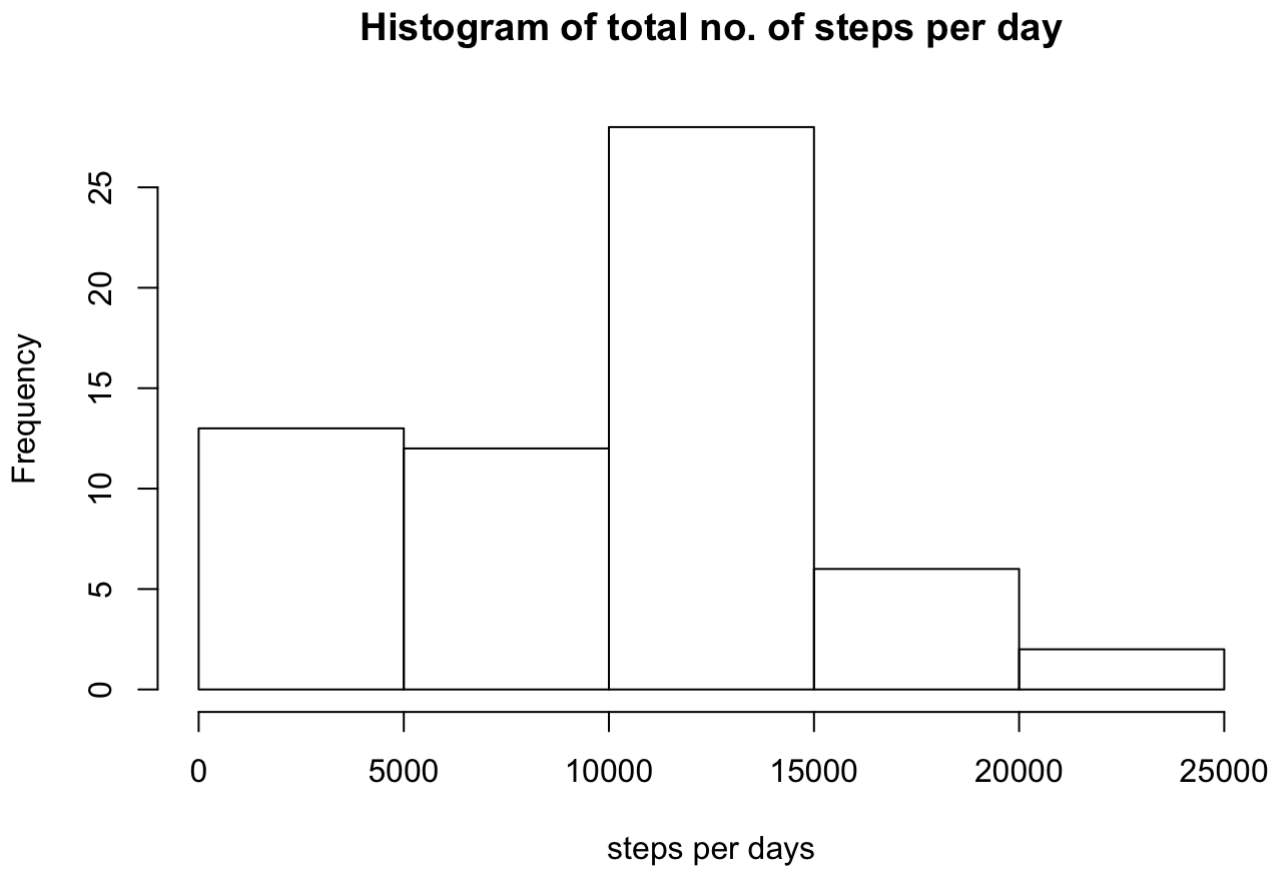
1. Calculate the total number of steps taken per day

```
steps_per_day <- activity %>%
        group_by(date) %>%
        summarise(total_steps= sum(steps, na.rm = T))
steps_per_day
```

```
## # A tibble: 61 x 2
##    date       total_steps
##    <date>           <int>
##  1 2012-10-01           0
##  2 2012-10-02         126
##  3 2012-10-03       11352
##  4 2012-10-04       12116
##  5 2012-10-05       13294
##  6 2012-10-06       15420
##  7 2012-10-07       11015
##  8 2012-10-08           0
##  9 2012-10-09       12811
## 10 2012-10-10        9900
## # … with 51 more rows
```

2. Make a histogram of the total number of steps taken each day

```
hist(steps_per_day$total_steps, main="Histogram of total no. of steps per day", xlab=
"steps per days")
```



3. Calculate and report the mean and median of the total number of steps taken per day

```
mean(steps_per_day$total_steps)
```

```
## [1] 9354.23
```

```
median(steps_per_day$total_steps)
```

```
## [1] 10395
```

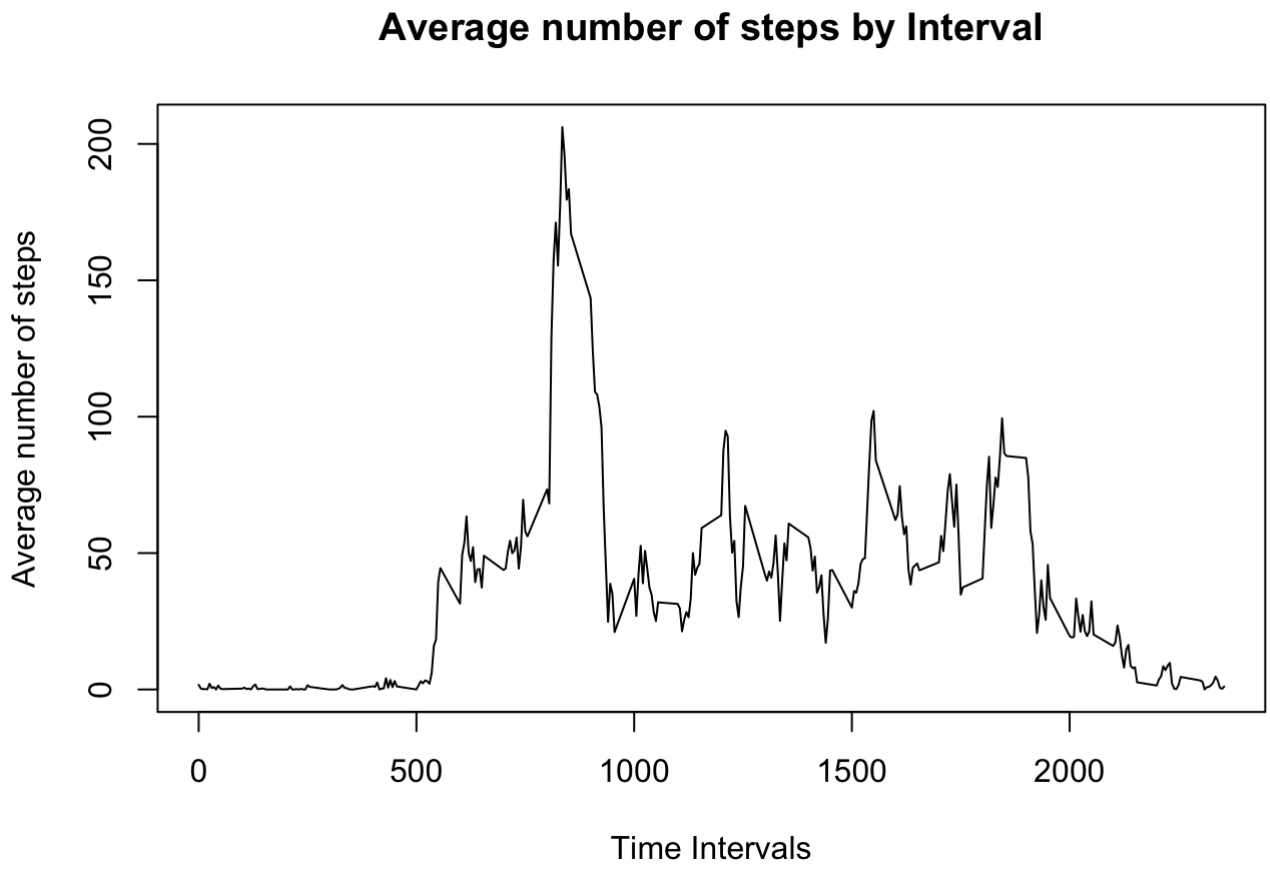
What is the average daily activity pattern?

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)

```
# Calculate average steps per interval for all days
avg_steps_per_interval <- aggregate(steps ~ interval, activity, mean)

# Calculate average steps per day for all intervals - Not required, but for my own sa
ke
avg_steps_per_day <- aggregate(steps ~ date, activity, mean)

# Plot the time series with appropriate labels and heading
plot(avg_steps_per_interval$interval, avg_steps_per_interval$steps, type='l', col=1,
 main="Average number of steps by Interval", xlab="Time Intervals", ylab="Average num
ber of steps")
```



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

```
# Identify the interval index which has the highest average steps
interval_idx <- which.max(avg_steps_per_interval$steps)

# Identify the specific interval and the average steps for that interval
print (paste("The interval with the highest avg steps is ", avg_steps_per_interval[in
terval_idx, ]$interval, " and the no of steps for that interval is ", round(avg_steps
_per_interval[interval_idx, ]$steps, digits = 1)))
```

```
## [1] "The interval with the highest avg steps is  835  and the no of steps for that
interval is  206.2"
```

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)

```
sum(is.na(activity))
```

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

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.

```
global_mean <- avg_steps_per_day$mean_steps %>%
        na.omit() %>%
        mean()
```

```
## Warning in mean.default(.): argument is not numeric or logical: returning NA
```

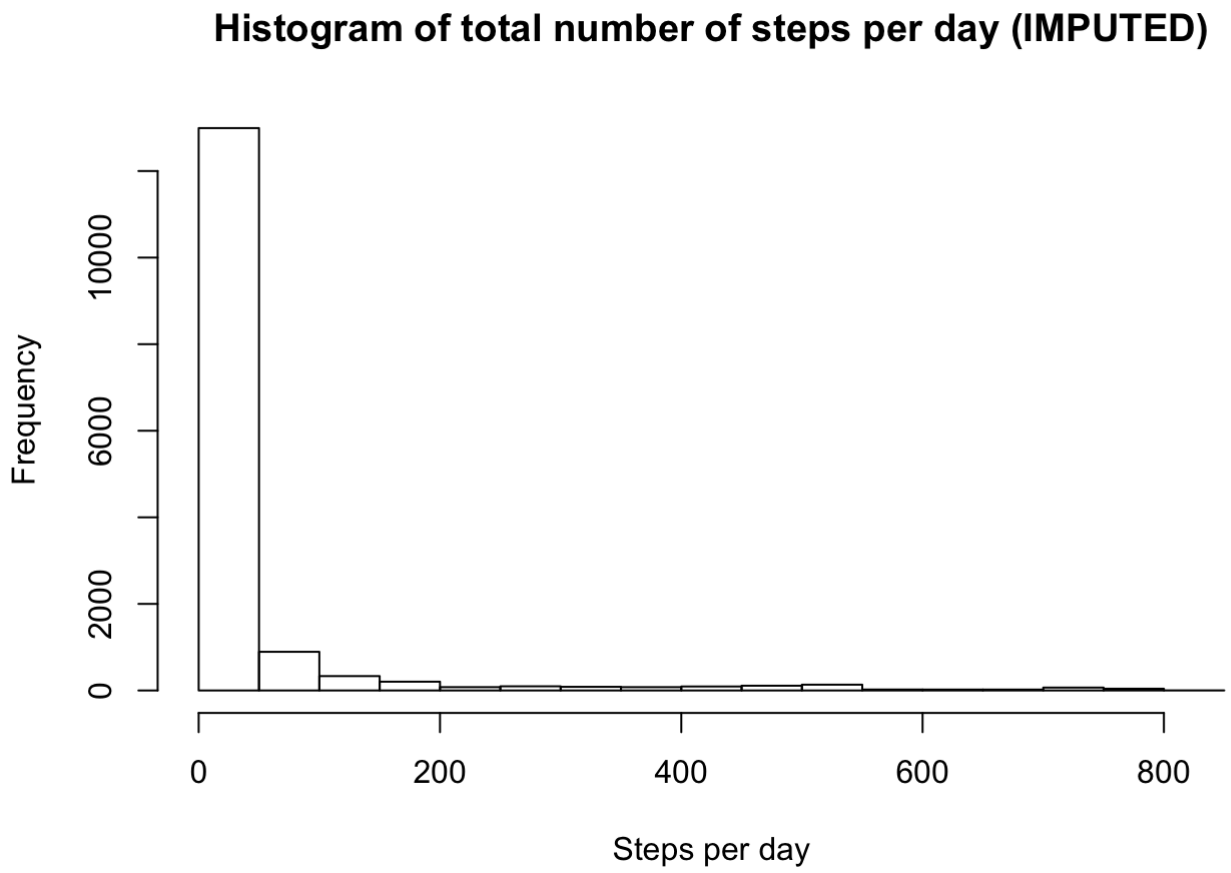
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
#replace all NAs with global_mean
df_replacedNA <- activity %>%
        replace_na(list(steps= global_mean))
head(df_replacedNA,30)
```

```
##    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
## 7     NA 2012-10-01       30
## 8     NA 2012-10-01       35
## 9     NA 2012-10-01       40
## 10    NA 2012-10-01       45
## 11    NA 2012-10-01       50
## 12    NA 2012-10-01       55
## 13    NA 2012-10-01      100
## 14    NA 2012-10-01      105
## 15    NA 2012-10-01      110
## 16    NA 2012-10-01      115
## 17    NA 2012-10-01      120
## 18    NA 2012-10-01      125
## 19    NA 2012-10-01      130
## 20    NA 2012-10-01      135
## 21    NA 2012-10-01      140
## 22    NA 2012-10-01      145
## 23    NA 2012-10-01      150
## 24    NA 2012-10-01      155
## 25    NA 2012-10-01      200
## 26    NA 2012-10-01      205
## 27    NA 2012-10-01      210
## 28    NA 2012-10-01      215
## 29    NA 2012-10-01      220
## 30    NA 2012-10-01      225
```

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?

```
hist(df_replacedNA$steps, main = "Histogram of total number of steps per day (IMPUTE
D)", xlab = "Steps per day")
```



Are there differences in activity patterns between weekdays and weekends?

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.

```
activity <- activity %>%
        mutate(is_weekend = chron::is.weekend(date))
head(activity)
```

```
##   steps       date interval is_weekend
## 1    NA 2012-10-01        0      FALSE
## 2    NA 2012-10-01        5      FALSE
## 3    NA 2012-10-01       10      FALSE
## 4    NA 2012-10-01       15      FALSE
## 5    NA 2012-10-01       20      FALSE
## 6    NA 2012-10-01       25      FALSE
```

```
summary(activity)
```

```
##      steps                date                  interval      is_weekend
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0   Mode :logical
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8   FALSE:12960
##  Median :  0.00   Median :2012-10-31   Median :1177.5   TRUE :4608
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
steps_per_day_impute <- aggregate(steps ~ interval+ is_weekend, activity, mean)

# Create the plot
ggplot(steps_per_day_impute, aes(interval, steps)) +
    geom_line(stat = "identity", aes(colour = is_weekend)) +
    theme_gray() +
    facet_grid(is_weekend ~ ., scales="fixed", space="fixed") +
    labs(x="Interval", y=expression("No of Steps")) +
    ggtitle("No of steps Per Interval by day type")
```