Week 2 project

Sam Channon-Wells

15 October 2019

Introduction:

Welcome to my week 2 project in Reproducible Research! This document contains the code and output required to complete the analyses required for week 2 of this coursera course.

Step 1 - Load and view the data:

Below we load in the data from a local file, and then view the first few rows.

```
library(dplyr); library(data.table);library(ggplot2)
```

```
##
## 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
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
data <- fread("repdata_data_activity/activity.csv")
head(data)
```

```
##    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
```

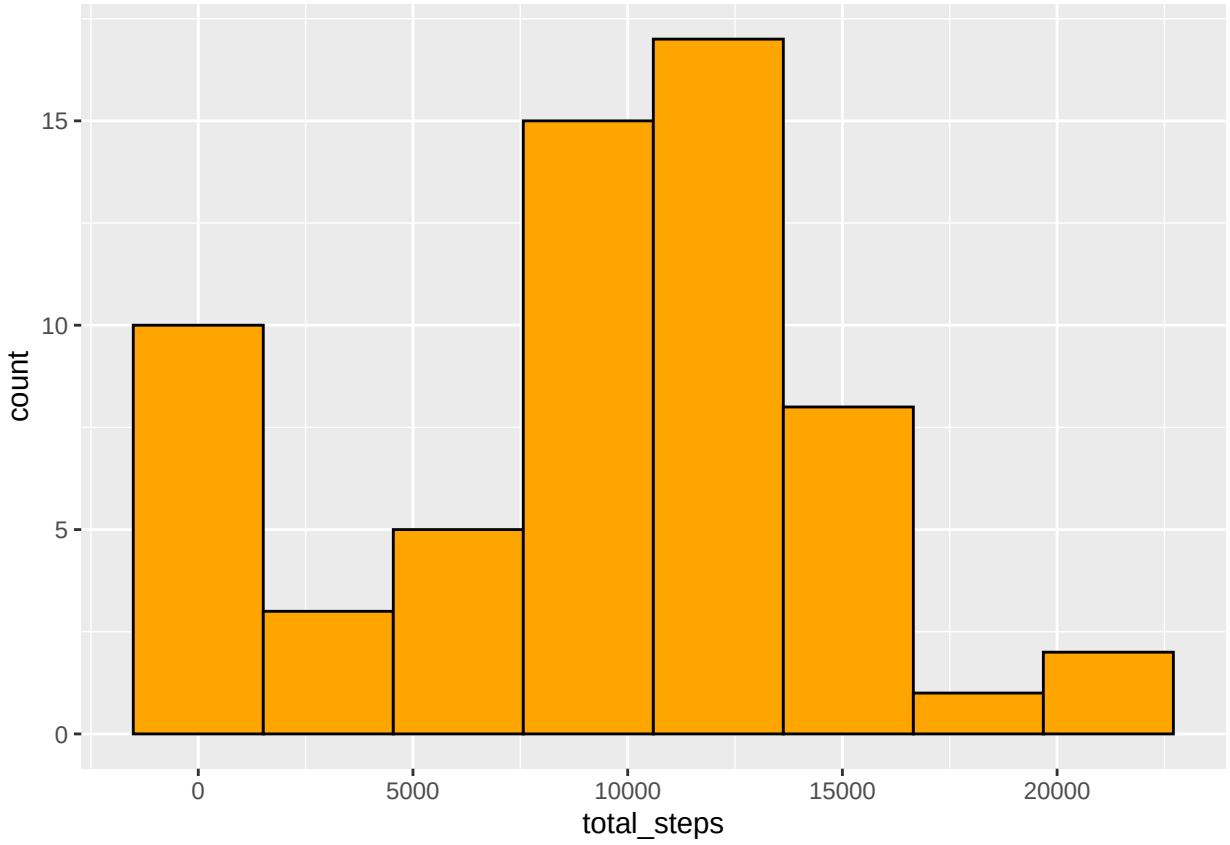
Step 2 - What is mean total number of steps taken per day?

The instructions from the course assignment are:

> For this part of the assignment, you can ignore the missing values in the dataset. Calculate the total number of steps taken per day
> If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day
> Calculate and report the mean and median of the total number of steps taken per day

We perform the required steps in the chunk below, and the output should provide the necessary histogram and statement of medians/means

```r
data2 <- data %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps, na.rm = TRUE))

ggplot(data2, aes(x = total_steps)) + geom_histogram(bins = 8, fill = "orange", col = "black")
```




```r
steps.mean <- mean(data2$total_steps)
steps.median <-median(data2$total_steps)

print(paste0("The mean number of steps per day was: ", steps.mean))
```

```
## [1] "The mean number of steps per day was: 9354.22950819672"
```

```r
print(paste0("The median number of steps per day was: ", steps.median))
```

```
## [1] "The median number of steps per day was: 10395"
```

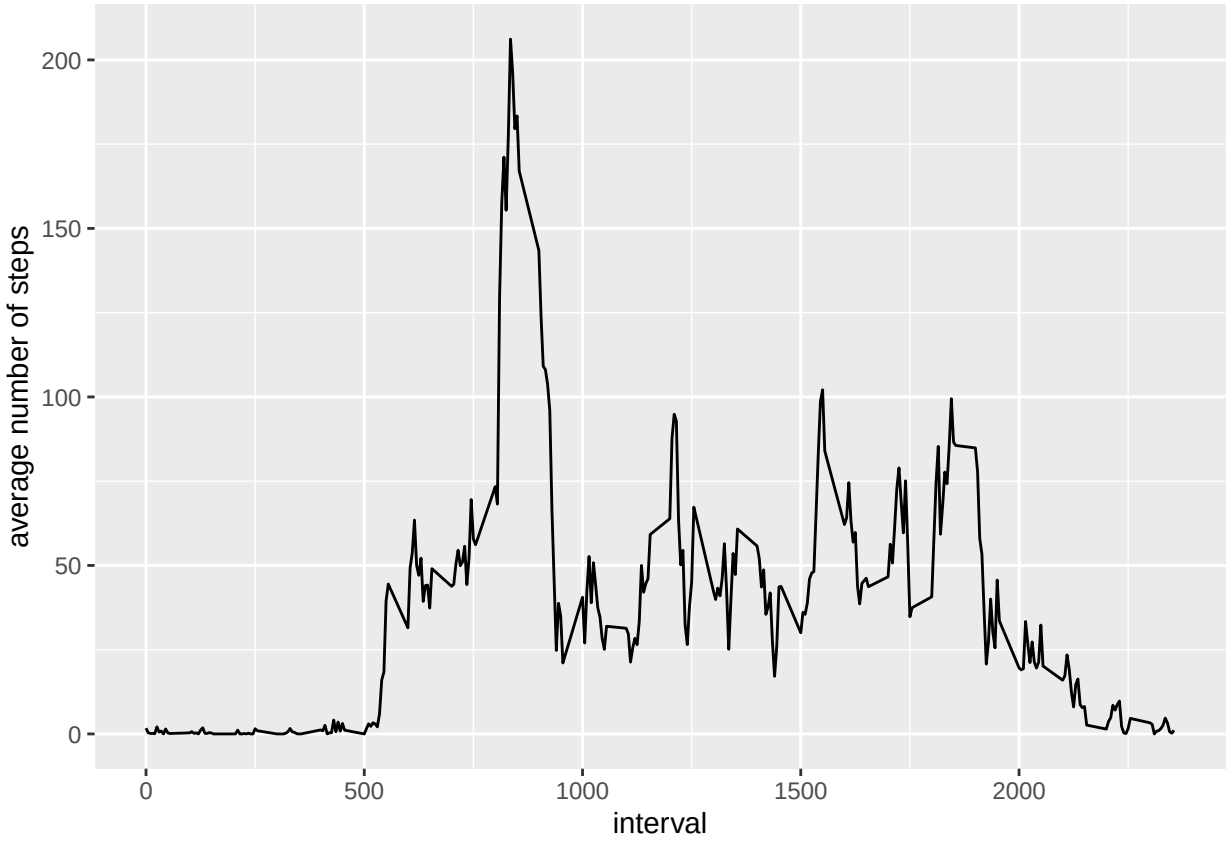
Step 3 - What is the average daily activity pattern?

The instructions from the course assignment are:

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

We perform the required steps in the chunk below, with the answer to the question printed as output.

```r
data3 <- data %>%
  group_by(interval) %>%
  summarise(average_steps_int = mean(steps, na.rm = TRUE))

ggplot(data3, aes(x = interval, y = average_steps_int)) + geom_line(col = "black")+ylab("average number
```




```r
max.steps.int <- which(data3$average_steps_int == max(data3$average_steps_int))
max.steps.int <- data3$interval[max.steps.int]

print(paste0("The interval with the largest average across all the days is from 8:35-8:40am. The averag
```

```
## [1] "The interval with the largest average across all the days is from 8:35-8:40am. The average numbe
```

Step 4 - Imputing missing values:

The instructions from the course assignment are:

> 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.
>
> Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs).
>
> 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. Create a new dataset that is equal to the original dataset but with the missing data filled in. 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?

We perform the required steps in the chunk below.

I have used the mean steps over other intervals to calculate the NA values at a particular interval.

```
# Calculate amount of missing data:

data.na <- apply(data, MAR = 1, FUN = function(x) { sum(is.na(x)) } )
data.na.any <- as.numeric(data.na > 0)

print(paste0("There are ", sum(data.na), " NAs in the whole data, with ", sum(data.na.any), " rows with
```

```
## [1] "There are 2304 NAs in the whole data, with 2304 rows with an NA. This suggests there is at most
```

```
steps.na <- sum(is.na(data$steps))
interval.na <- sum(is.na(data$interval))
date.na <- sum(is.na(data$date))

print(paste0("There are ", steps.na, " missing values in the steps column"))
```

```
## [1] "There are 2304 missing values in the steps column"
```

```
print(paste0("There are ", interval.na, " missing values in the interval column"))
```

```
## [1] "There are 0 missing values in the interval column"
```

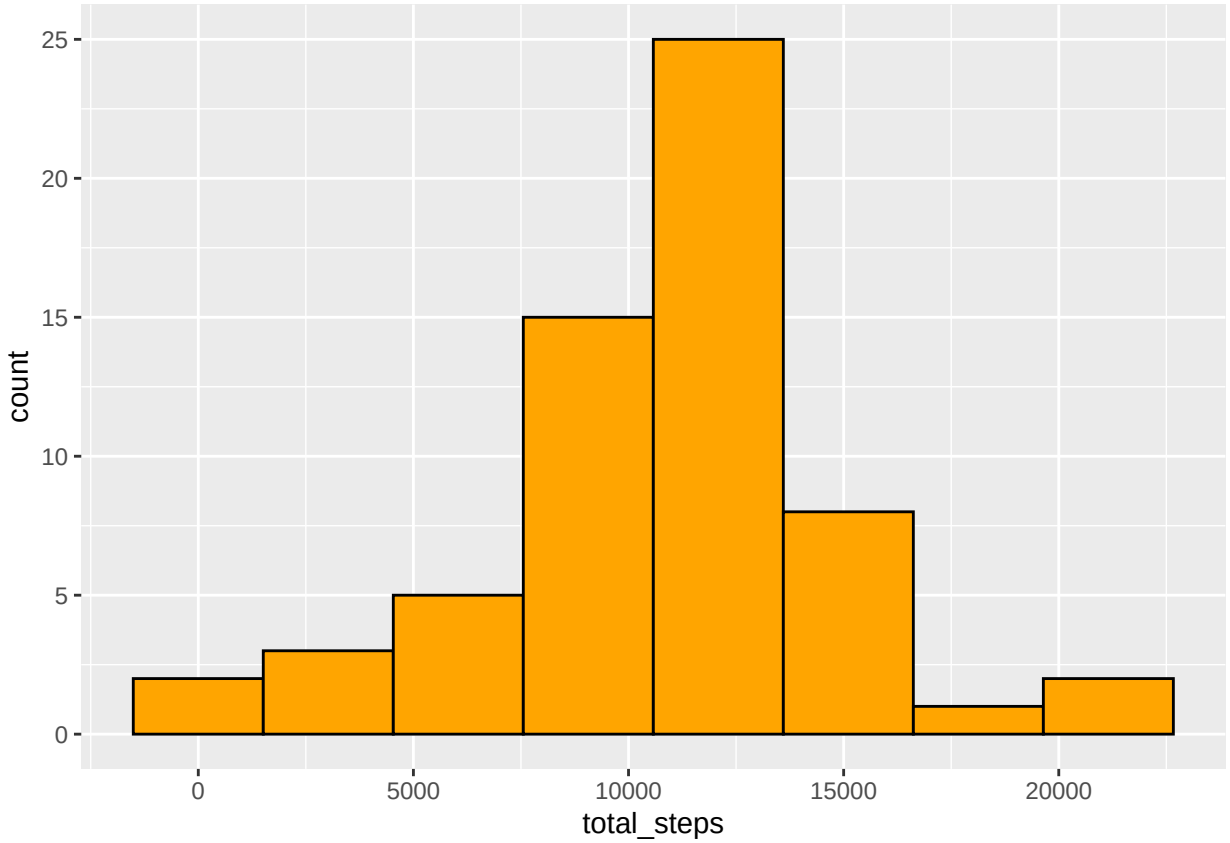
```
print(paste0("There are ", date.na, " missing values in the date column"))
```

```
## [1] "There are 0 missing values in the date column"
```

```
print("This shows that all the NAs are in the steps column")
```

```
## [1] "This shows that all the NAs are in the steps column"
```

```
# Replace missing values:

data3 <- data %>%
  group_by(interval) %>%
  summarise(average_steps_int = mean(steps, na.rm = TRUE))

data.comp <- data
for ( i in which(is.na(data.comp$steps))) {
  int <- data.comp$interval[i]
  impute.val <- data3$average_steps_int[data3$interval == int]
  data.comp$steps[i] <- impute.val
}

# Redraw histogram and recalculate means/medians:

data4 <- data.comp %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps, na.rm = TRUE))

ggplot(data4, aes(x = total_steps)) + geom_histogram(bins = 8, fill = "orange", col = "black")
```




```
steps.mean <- mean(data4$total_steps)
steps.median <-median(data4$total_steps)

print(paste0("The mean number of steps per day was: ", steps.mean))
```

```
## [1] "The mean number of steps per day was: 10766.1886792453"
```

```
print(paste0("The median number of steps per day was: ", steps.median))
```

```
## [1] "The median number of steps per day was: 10766.1886792453"
```

```
print("Yes, these values are different. We have moved both the median and the mean values up.")
```

```
## [1] "Yes, these values are different. We have moved both the median and the mean values up."
```

Step 5 - Are there differences in activity patterns between weekdays and weekends?

The instructions from the course assignment are:

> For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.
>
> 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. 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.

We perform the required steps in the chunk below, and the output should provide the necessary plot.

```
# install.packages("lubridate")
library(lubridate)
```

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

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
data.comp$date <- as.Date(data.comp$date, format="%Y-%m-%d")
data.comp$dow <- weekdays(data.comp$date)
data.comp$dow[data.comp$dow %in% c("Sunday", "Saturday")] <- "weekend"
data.comp$dow[data.comp$dow != "weekend"] <- "weekday"

data5 <- data.comp %>%
  group_by(interval, dow) %>%
  summarise(av_steps = mean(steps))

ggplot(data5, aes(x = interval, y = av_steps, col = dow)) + geom_line() + facet_wrap(. ~ dow, nrow = 2)
```