Course Project 1

Robson Cruz

20/01/2021

Introduction

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

Loading and preprocessing the data

```
## Load libraries
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.3
```

```
##
## 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(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.3
```

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

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

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.3
```

```
## 1. Load data
activity <- read.csv(file = "./data/activity.csv",
                     header = TRUE,
                     dec = ".",
                     sep = ",")
```

```
## 2. Processing
activity <- activity %>%
        mutate(date = as.Date(date, format = "%Y-%m-%d"),
               day = mday(date))

as_tibble(activity)
```

```
## # A tibble: 17,568 x 4
##    steps date       interval   day
##    <int> <date>        <int> <int>
##  1    NA 2012-10-01        0     1
##  2    NA 2012-10-01        5     1
##  3    NA 2012-10-01       10     1
##  4    NA 2012-10-01       15     1
##  5    NA 2012-10-01       20     1
##  6    NA 2012-10-01       25     1
##  7    NA 2012-10-01       30     1
##  8    NA 2012-10-01       35     1
##  9    NA 2012-10-01       40     1
## 10    NA 2012-10-01       45     1
## # ... with 17,558 more rows
```

What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day

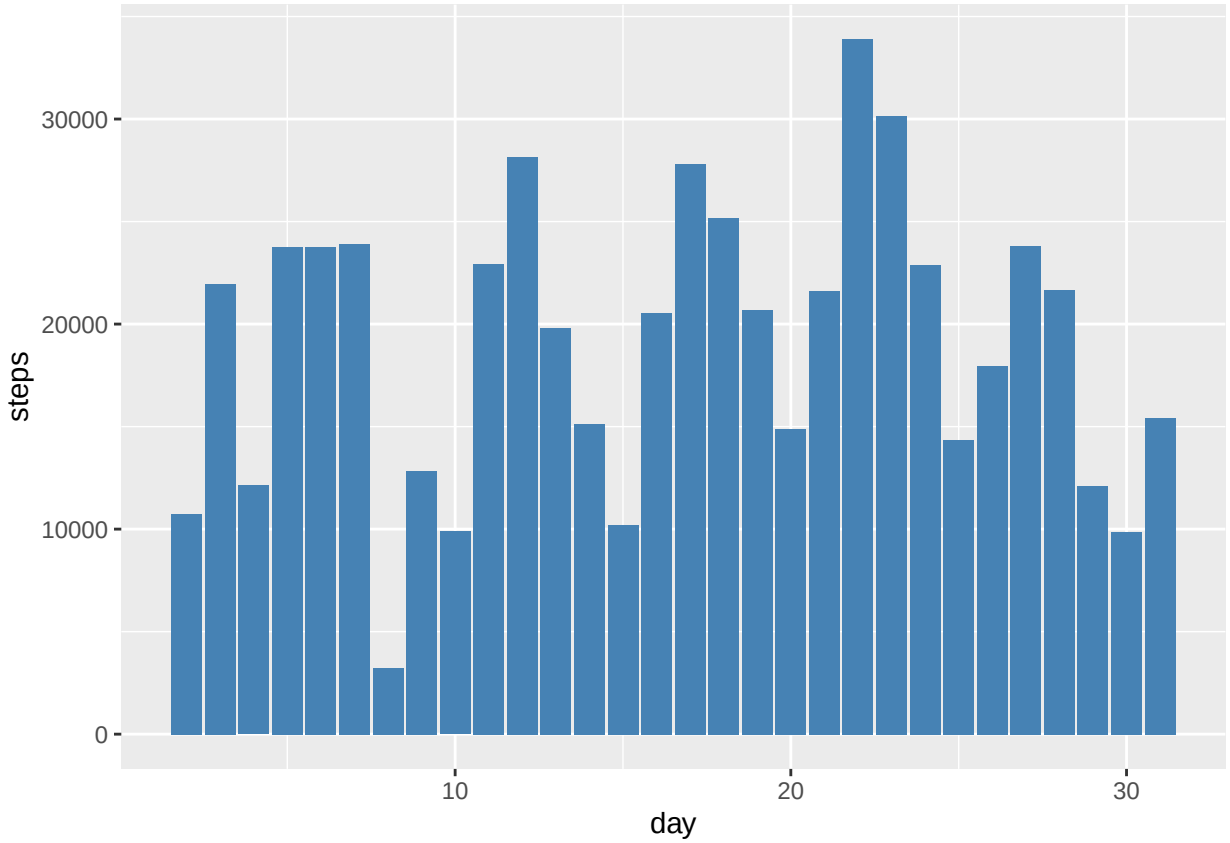
```
activity %>%
        filter(!is.na(steps)) %>%
        group_by(day) %>%
        summarize(avg_steps = mean(steps), Total_steps = n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 30 x 3
##      day avg_steps Total_steps
##    <int>     <dbl>       <int>
##  1     2      18.6         576
##  2     3      38.1         576
##  3     4      42.1         288
##  4     5      41.2         576
##  5     6      41.2         576
##  6     7      41.5         576
##  7     8      11.2         288
##  8     9      44.5         288
##  9    10      34.4         288
## 10    11      39.8         576
## # ... with 20 more rows
```

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

```
activity %>%
        filter(!is.na(steps)) %>%
        group_by(day, steps) %>%
        ggplot(aes(x = day, y = steps)) +
        geom_bar(stat = "identity", fill = "steelblue")
```



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

```
activity %>%
        filter(!is.na(steps)) %>%
        group_by(day, steps) %>%
        summarize(avg_steps = mean(steps), median = median(steps))
```

```
## `summarise()` regrouping output by 'day' (override with `.groups` argument)
```

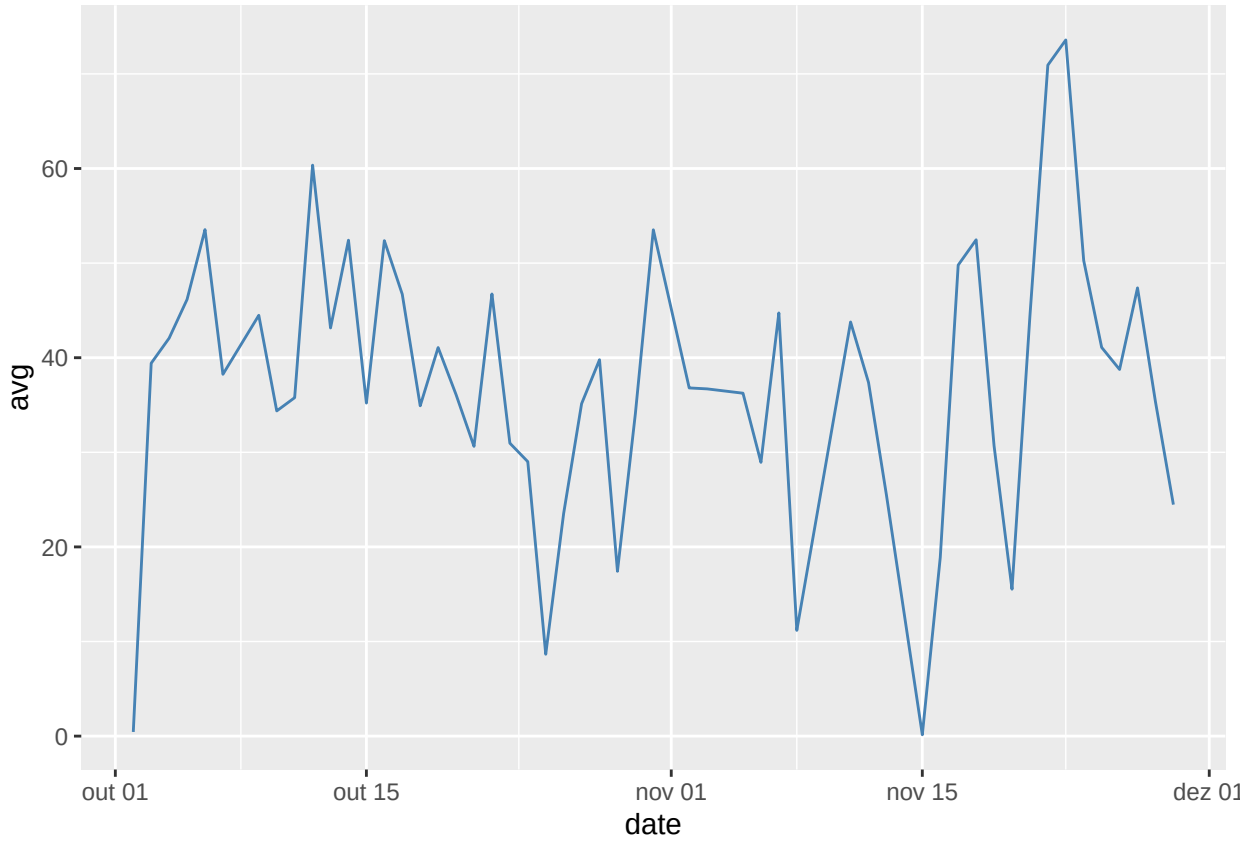
```
## # A tibble: 3,007 x 4
## # Groups:   day [30]
##       day steps avg_steps median
##     <int> <int>     <dbl>  <dbl>
##  1     2     0         0      0
##  2     2     6         6      6
##  3     2     8         8      8
##  4     2     9         9      9
##  5     2    10        10     10
##  6     2    11        11     11
##  7     2    14        14     14
##  8     2    15        15     15
##  9     2    16        16     16
## 10     2    17        17     17
## # ... with 2,997 more rows
```

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)

```
activity %>%
        filter(!is.na(steps), !is.na(date)) %>%
        group_by(date) %>%
        summarize(avg = mean(steps)) %>%
        ggplot(aes(x = date, y = avg)) +
        geom_line(color="steelblue") +
        scale_x_date(date_labels = "%b %d")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



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

```
activity %>%
        mutate(avg = mean(interval, na.rm = TRUE)) %>%
        summarize(max = max(avg, na.rm = TRUE)) %>%
        print()
```

```
##       max
## 1 1177.5
```

Imputing missing values

1. Calculate and report the total number of missing values in the dataset

```
activity %>%
        filter_all(any_vars(is.na(.))) %>%
        group_by(steps) %>%
        summarize(n = n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 1 x 2
##   steps     n
##   <int> <int>
## 1    NA  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.

```
activity %>%
        replace_na(list(steps = 37.38)) %>%
        as_tibble()
```

```
## # A tibble: 17,568 x 4
##    steps date       interval   day
##    <dbl> <date>        <int> <int>
##  1  37.4 2012-10-01        0     1
##  2  37.4 2012-10-01        5     1
##  3  37.4 2012-10-01       10     1
##  4  37.4 2012-10-01       15     1
##  5  37.4 2012-10-01       20     1
##  6  37.4 2012-10-01       25     1
##  7  37.4 2012-10-01       30     1
##  8  37.4 2012-10-01       35     1
##  9  37.4 2012-10-01       40     1
## 10  37.4 2012-10-01       45     1
## # ... with 17,558 more rows
```

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

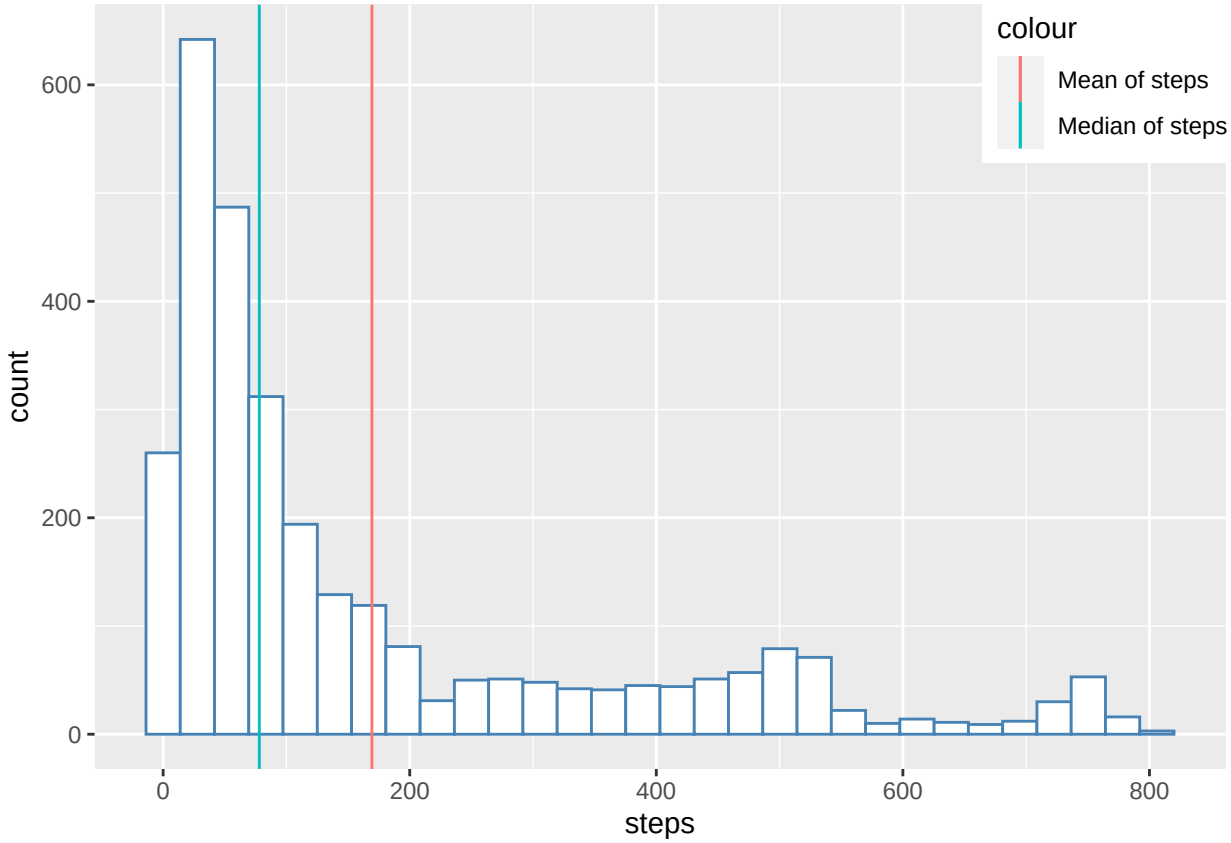
```
no_NA <- activity %>%
        replace_na(list(steps = 37.38)) %>%
        as_tibble()
```

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?

```
no_NA %>%
        group_by(day, steps) %>%
        summarize(mean = mean(steps),
                  median = median(steps)) %>%
        ggplot(aes(x = steps)) +
        geom_histogram(fill="white", color = "steelblue", position="dodge") +
        geom_vline(aes(xintercept = mean(steps), color = "Mean of steps")) +
        geom_vline(aes(xintercept = median(steps), color = "Median of steps")) +
        theme(legend.position = c(0.9, 0.9))
```

```
## `summarise()` regrouping output by 'day' (override with `.groups` argument)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Are there differences in activity patterns between weekdays and weekends? 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.

```
## Set language
Sys.setlocale("LC_ALL","English")
```

```
## [1] "LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MONETARY=English_U
```

```
activity_week <- activity %>%
        filter(!is.na(steps) | steps != 0, !is.na(interval)) %>%
        mutate(day = wday(date, label = TRUE, abbr = TRUE),
               Wday = ifelse(day == "Sun" | day == "Sat", "weekend", "weekday")) %>%
        group_by(interval, Wday) %>%
        summarize(avg = mean(steps, na.rm = TRUE))
```

```
## `summarise()` regrouping output by 'interval' (override with `.groups` argument)
```

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).

```
ggplot(na.omit(activity_week), aes(x = interval, y = avg)) +
        geom_line(stat = "identity", color = "steelblue") +
        facet_wrap(Wday ~ .) +
        theme(plot.title = element_text(hjust = 0.5)) +
        labs(x = "Interval (sec.)",
             y = "Number of Steps",
```

```
        title = "Mean steps over each 5min interval split by weekday/weekend")
```