# Assessment1 - PA1_template

## Task1

Upload data set:

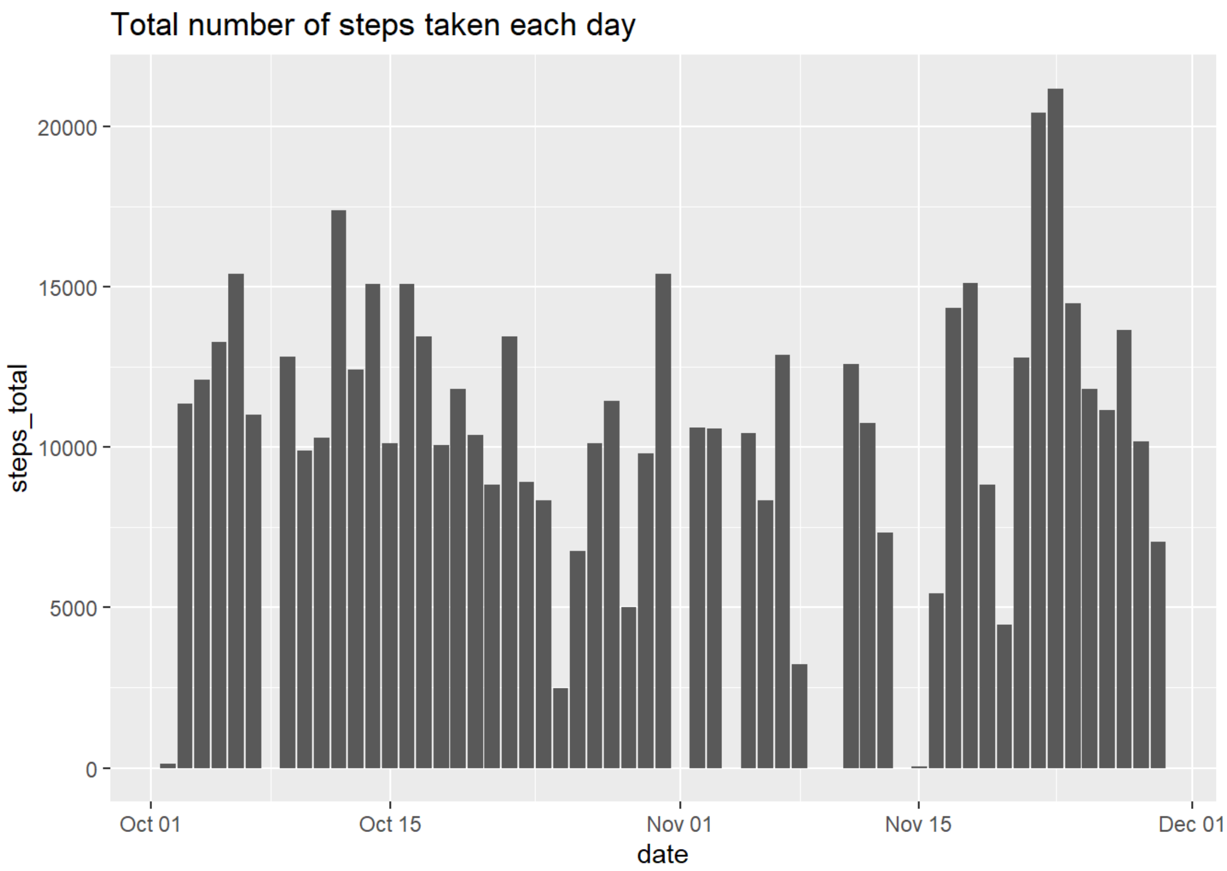
```
directory <- "C:/R Code/Training Materials/Coursera-1"

activity_orig <- read.csv(paste(directory,"activity.csv",sep='/'),
                        header=T,
                        na.strings="")

activity_data <- activity_orig %>%
  filter(steps != "NA")%>%
  mutate(steps = as.numeric (steps),
         date = as.Date(date))
```

## Task 2

Calculate number of steps taken each day:

```
activity_data %>%
  group_by(date) %>%
  dplyr::summarise(steps_total= sum(steps)) %>%
  ggplot(aes(x=date, y=steps_total))+
  geom_bar(stat = "identity") +
  ggtitle('Total number of steps taken each day')
```

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



# Task 3

Calculate mean and median steps per day:

```
activity_data_3 <- activity_data %>%
  group_by(date) %>%
  dplyr::summarise(steps_total= sum(steps)) %>%
  mutate(steps_mean = mean(steps_total),
         steps_median = median(steps_total))
```

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

```
paste("Mean steps per day" , ceiling(activity_data_3$steps_mean[1]))
```

```
## [1] "Mean steps per day 10767"
```

```
paste("Median steps per day" , ceiling(activity_data_3$steps_median[1]))
```

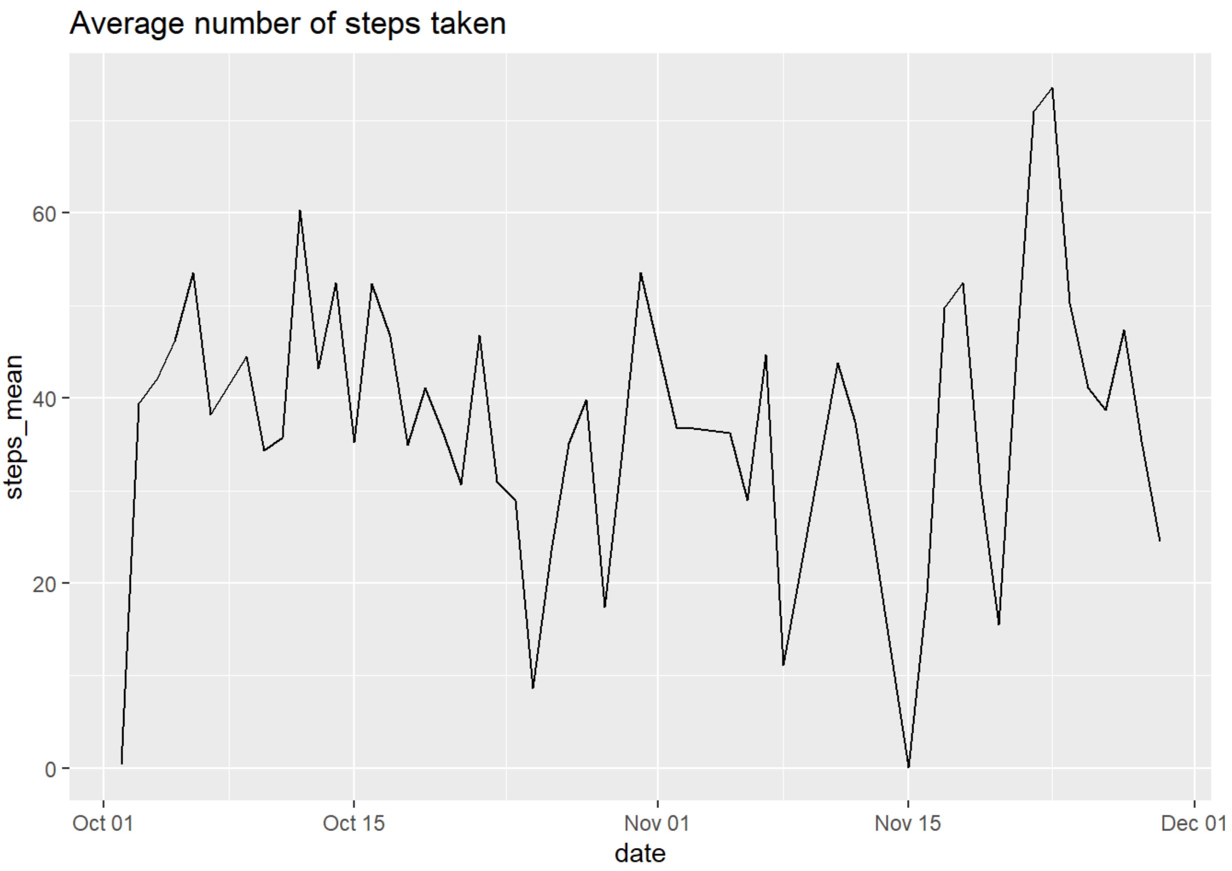
```
## [1] "Median steps per day 10765"
```

# Task 4

Plots average number of steps taken per day:

```
activity_data %>%
  group_by(date) %>%
  dplyr::summarise(steps_mean= mean(steps)) %>%
  ggplot(aes(x=date, y=steps_mean))+
    geom_line(aes(group=1))+
    ggtitle('Average number of steps taken')
```

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



# Task 5

Gives interval where maximum number of steps take place:

```
activity_data_5 <- activity_data %>%
  group_by(interval) %>%
  dplyr::summarise(steps_mean= mean(steps))
```

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

```
subset(activity_data_5, steps_mean == max(steps_mean))
```

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

# Task 6

Missing values on column steps is filled with mean steps:

```
steps_na <- activity_orig %>%
  filter(steps == "NA")%>%
  group_by(steps) %>%
  dplyr::summarise(n=n())
```

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

```
paste("Missing values on column steps", steps_na[2])
```

```
## [1] "Missing values on column steps 2304"
```

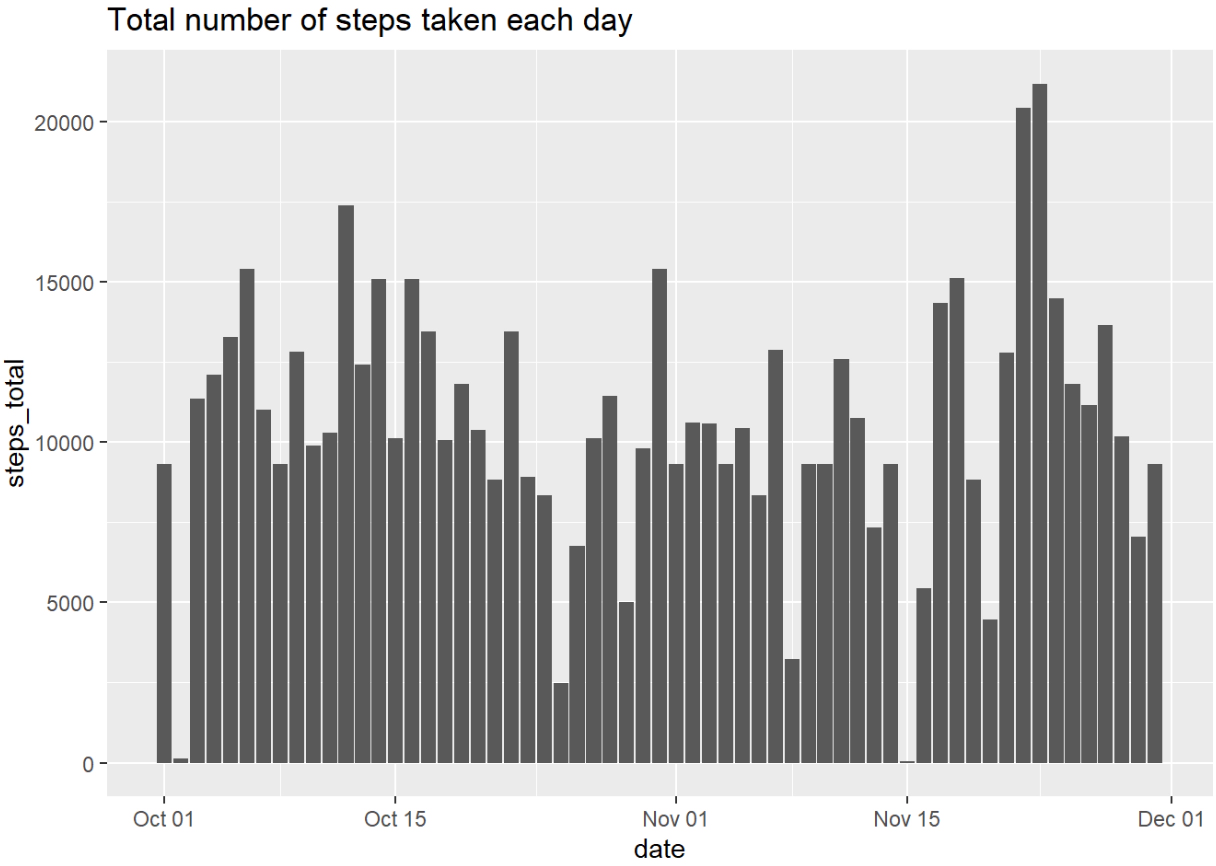
```
activity_data_6 <- activity_orig %>%
  mutate(steps =ifelse(steps == "NA","-1",steps),
         steps = as.numeric (steps),
         date = as.Date(date),
         steps = replace(steps, steps ==-1,mean(steps, na.rm = TRUE)))
```

# Task 7

Number of steps taken per each day when filling missing values:

```
activity_data_6 %>%
  group_by(date) %>%
  dplyr::summarise(steps_total= sum(steps)) %>%
  ggplot(aes(x=date, y=steps_total))+
  geom_bar(stat = "identity") +
  ggtitle('Total number of steps taken each day')
```

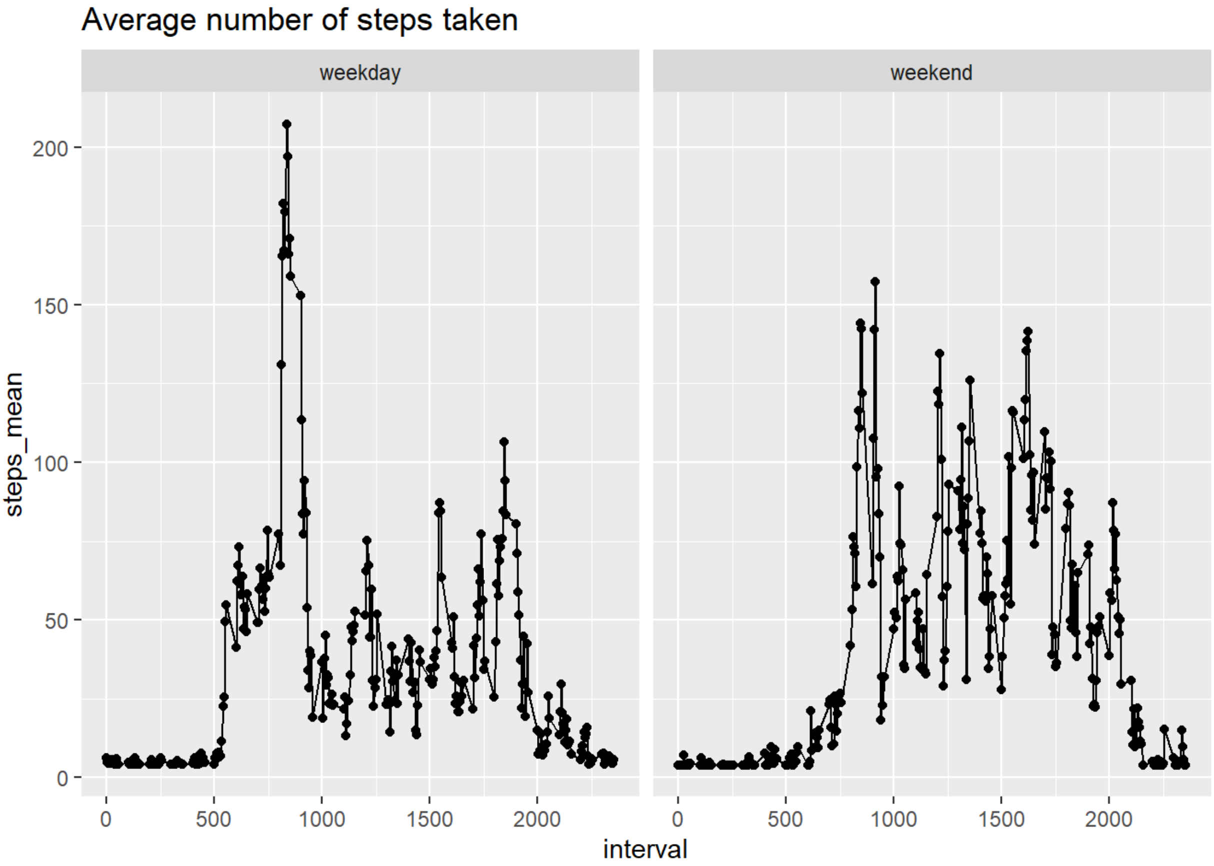
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



# Task 8

```
activity_data_6 <- activity_data_6 %>%
  mutate(weekdays = weekdays(date),
           weekdays_merge = ifelse(weekdays == "Sunday"|weekdays == "Saturda
y",
                                    "weekend","weekday"))

activity_data_6 %>%
  group_by(weekdays_merge,interval) %>%
  dplyr::summarise(steps_mean= mean(steps)) %>%
  ggplot(aes(x=interval, y=steps_mean))+
  geom_point()+
  geom_line(aes(group=1))+
  facet_wrap(~ weekdays_merge)+
  ggtitle('Average number of steps taken')
```

```
## `summarise()` regrouping output by 'weekdays_merge' (override with `.grou
ps` argument)
```


Average number of steps taken
weekday
weekend
steps_mean
200
150
100
50
0
0
500
1000
1500
2000
interval