Reproducible Research Course Project 1

Shree Ravi

April 15, 2017

Activity Dataset - Data analysis

Let's first load the dataset into RStudio. I have it saved to my working directory, so I'll simply use read.csv() to get the data into RStudio.

```
activity_data <- read.csv("activity.csv", header = TRUE)
str(activity_data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

I can see that the "date" field is a "factor", but what I really want is for it to be in a date format. So I convert the date field into the date format using as.Date() as shown below:

```
activity_data <- within(activity_data, date <- as.Date(date, format = "%Y-%m-%d"))
str(activity_data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

Now we can see that the date field is in the desired format.

Question 1: What is mean total number of steps taken per day?

1. Calculate the total number of steps taking per day.

Using aggregate(), we compute the sum of the "steps" field, by date.

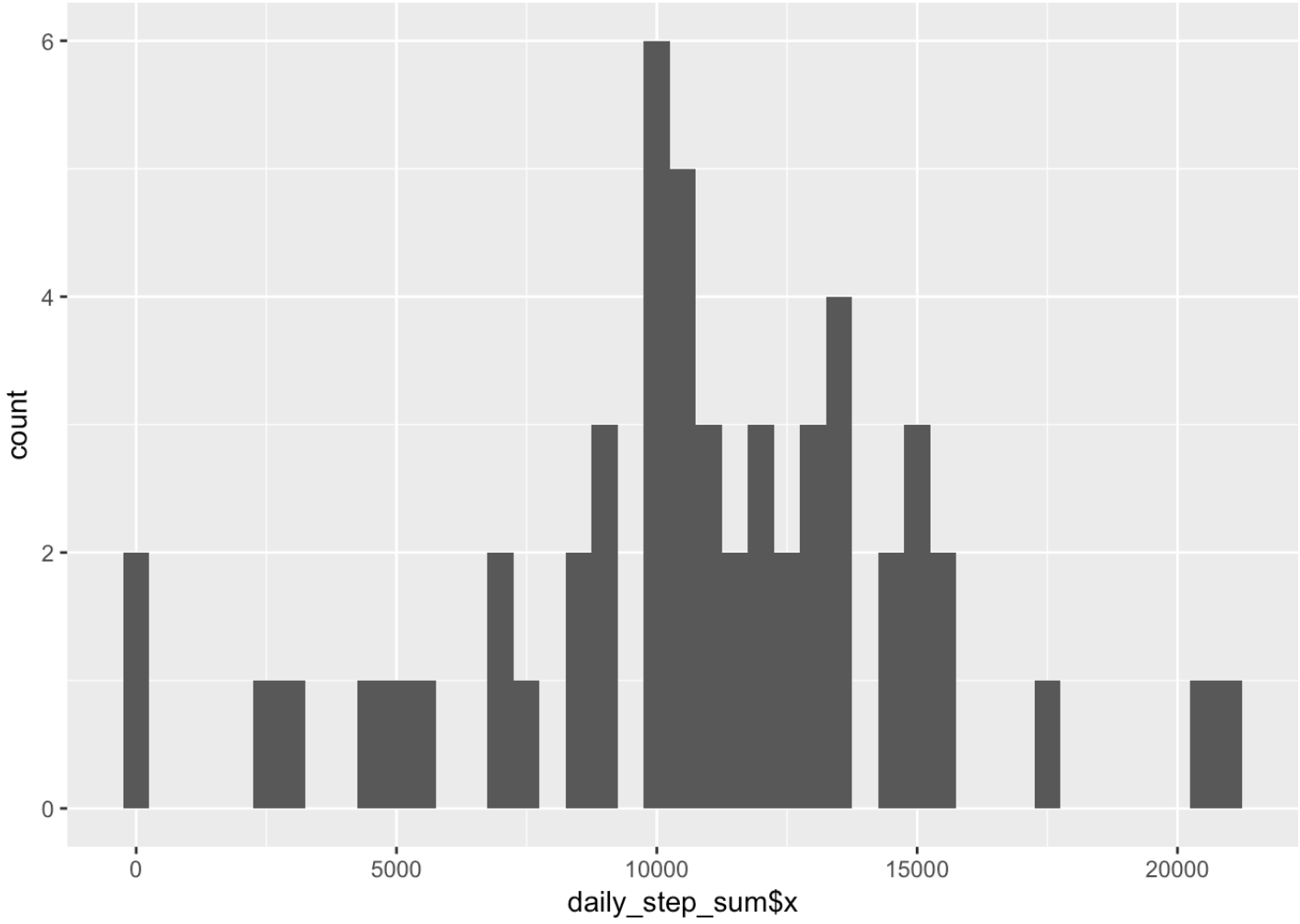
```
daily_step_sum <- aggregate(activity_data$steps, list(date = activity_data$date), sum
)
```

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

Using the ggplot plotting system, we can plot a histogram of the total steps taken each day, as shown below:

```
library(ggplot2)
ggplot(data = daily_step_sum, aes(daily_step_sum$x)) + geom_histogram(binwidth = 500)
```

```
## Warning: Removed 8 rows containing non-finite values (stat_bin).
```



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

```
mean(daily_step_sum$x, na.rm = TRUE)
```

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

```
median(daily_step_sum$x, na.rm = TRUE)
```

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

We can see that the mean and the median are nearly the same.

Question 2: 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)

First, remove the missing values from the activity_data data frame using complete.cases()

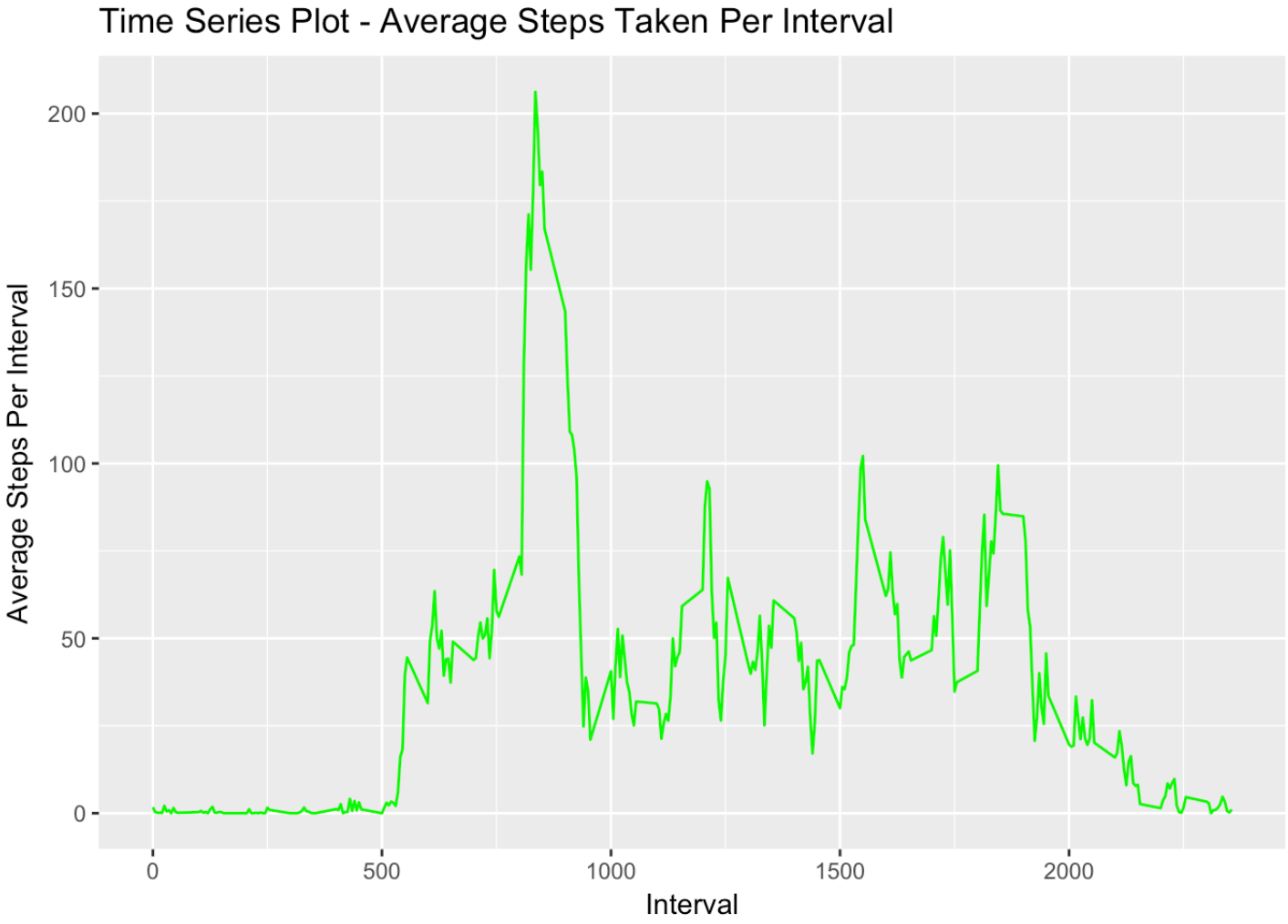
```
activity_data <- activity_data[complete.cases(activity_data), ]
```

Next, compute the average steps taken per day, using aggregate(), as shown below:

```
avg_interval_step <- aggregate(activity_data$steps, list(interval = activity_data$int
erval), mean)
```

Now plot a time series plot, using the ggplot plotting system. I modify the format of the date scale to explicity show the month, day and year.

```
ggplot(data = avg_interval_step, aes(interval, x)) + geom_line(color = "green", na.rm
= TRUE) + labs(x = "Interval", y = "Average Steps Per Interval", title = "Time Series
Plot - Average Steps Taken Per Interval")
```



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

```
avg_interval_step[(which.max(avg_interval_step$x)), ]
```

```
##     interval        x
## 104      835 206.1698
```

From the above code, the interval 835 is the one with the maximum average steps taken.

Question 3: Imputing missing values

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

Since we earlier removed the NAs from the activity_data data frame for Questions 1 & 2, we will reload the .csv file into R:

```
activity_data <- read.csv("activity.csv", header = TRUE)
activity_data <- within(activity_data, date <- as.Date(date, format = "%Y-%m-%d"))
```

Now I can create a new data frame with just the missing values to answer the question.

```
sum(is.na(activity_data$steps))
```

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

We can see that there are 2,304 records with missing or NA values.

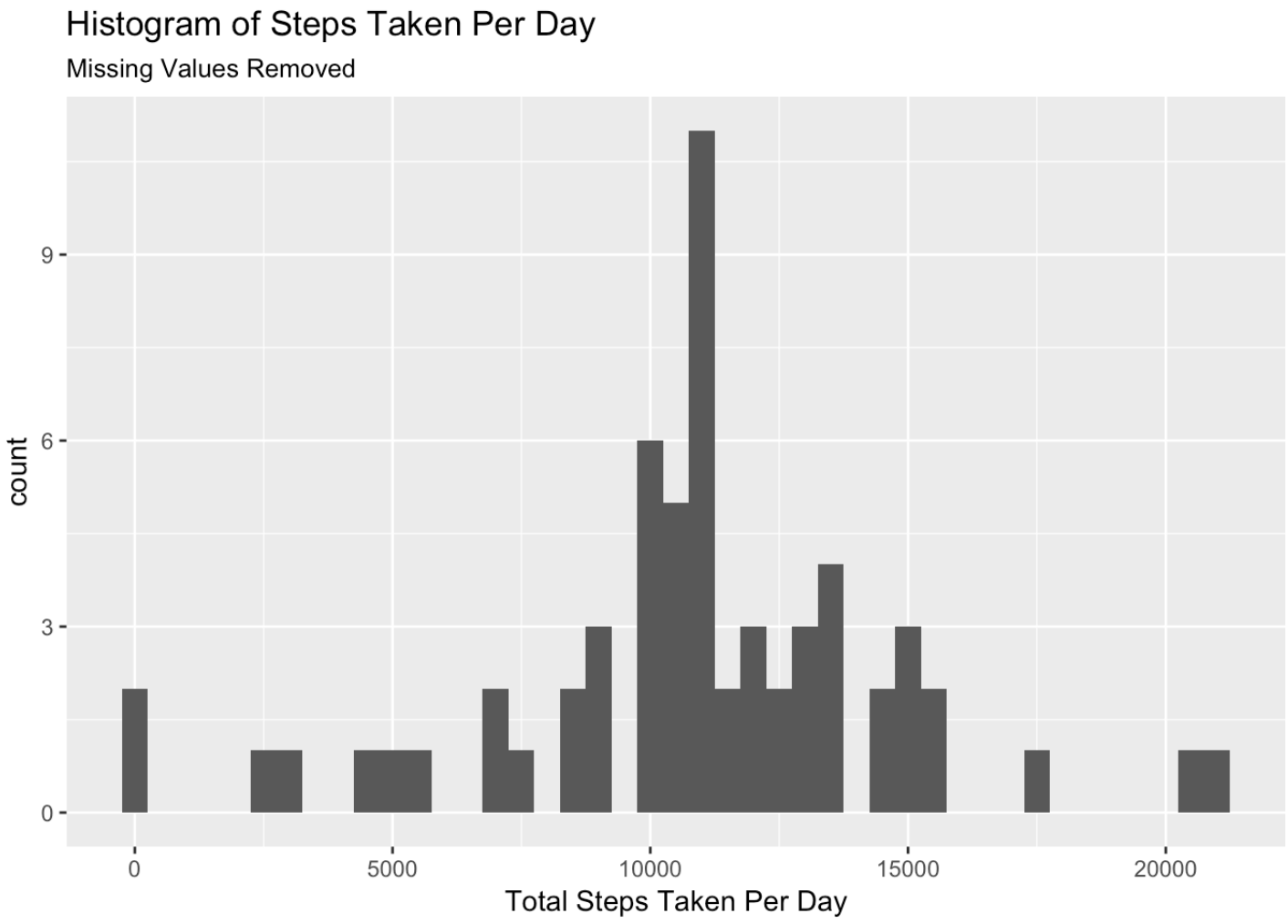
2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
activity_data_2 <- read.csv("activity.csv", header = TRUE)

activity_data_2 <- within(activity_data_2, date <- as.Date(date, format = "%Y-%m-%d")
)

nas <- is.na(activity_data_2$steps) ## Create logical vector of missing values

## Compute mean by interval to fill in missing values
avg_interval <- tapply(activity_data_2$steps, activity_data_2$interval, mean, na.rm =
TRUE, simplify = TRUE)

activity_data_2$steps[nas] <- avg_interval[as.character(activity_data_2$interval[nas]
)]

## Check if there are any missing values
sum(is.na(activity_data_2$steps))
```

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

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?

```
daily_step_sum2 <- aggregate(activity_data_2$steps, list(date = activity_data_2$date)
, sum)
ggplot(data = daily_step_sum2, aes(daily_step_sum2$x)) + geom_histogram(binwidth = 50
0) + labs(x = "Total Steps Taken Per Day", title = "Histogram of Steps Taken Per Day"
, subtitle = "Missing Values Removed")
```




```
## Compute mean and median
mean(daily_step_sum2$x)
```

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

```
median(daily_step_sum2$x)
```

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

We can see from the results of the above code that the mean and median are now equal. ### Question 4: Are there differences in activity patterns between weekdays and weekends?

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

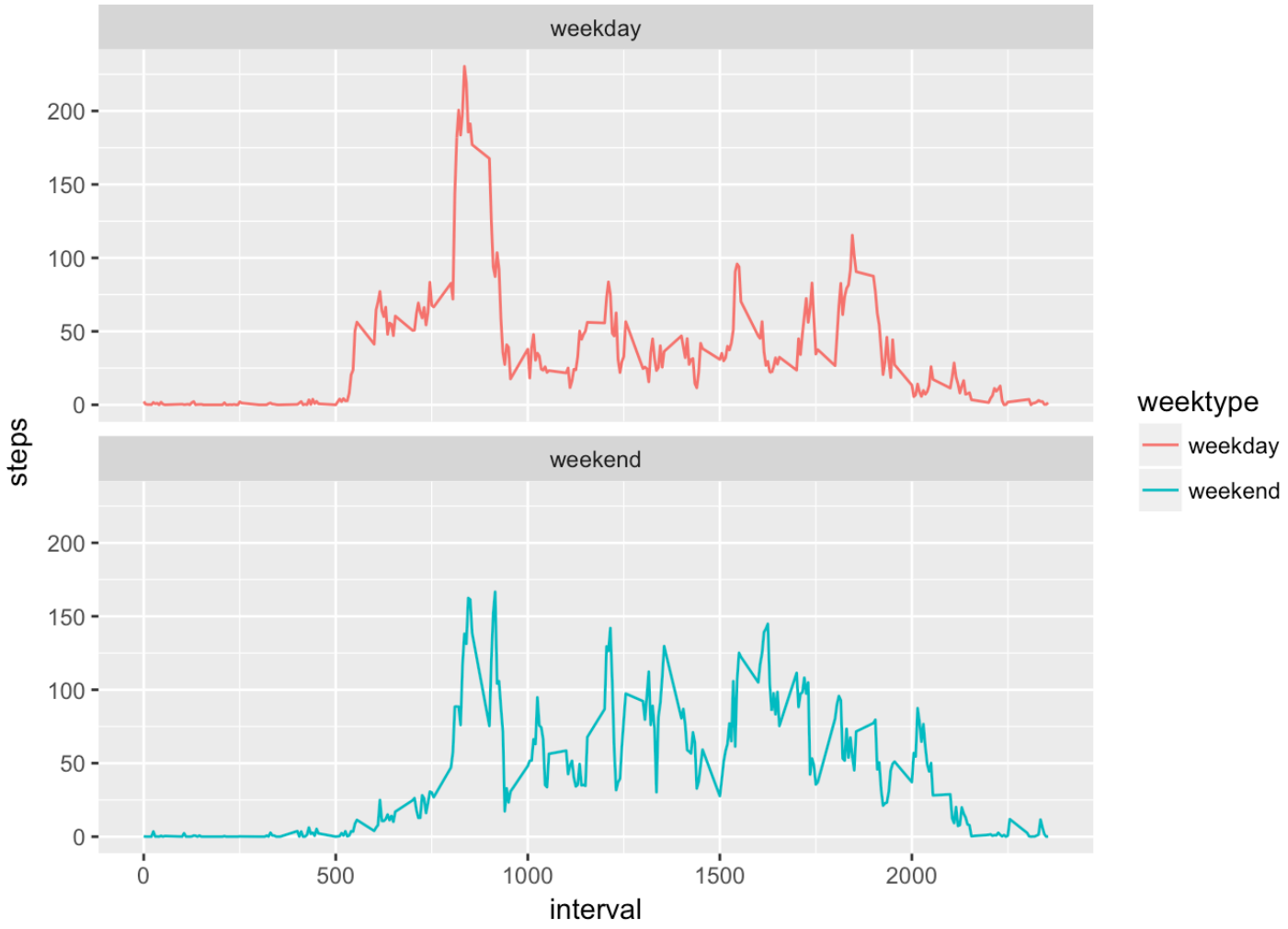
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
activity_data_2 <- mutate(activity_data_2, weektype = ifelse(weekdays(activity_data_2
$date) == "Saturday" | weekdays(activity_data_2$date) == "Sunday", "weekend", "weekda
y"))
activity_data_2$weektype <- as.factor(activity_data_2$weektype)
str(activity_data_2)
```

```
## 'data.frame':    17568 obs. of  4 variables:
##  $ steps   : num  1.717 0.3396 0.1321 0.1509 0.0755 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ weektype: Factor w/ 2 levels "weekday","weekend": 1 1 1 1 1 1 1 1 1 1 ...
```

Now let's plot the panel plot using ggplot plotting system.

```
interval <- activity_data_2 %>%
  group_by(interval, weektype) %>%
  summarise(steps = mean(steps))
s <- ggplot(interval, aes(x=interval, y=steps, color = weektype)) +
  geom_line() +
  facet_wrap(~weektype, ncol = 1, nrow=2)
print(s)
```



We can see that the total steps taken on weekends is more uniform across the intervals compared to weekdays.