Marketing Email Campaign

Mitul Shah

8/9/2017

Loading the data

```
## Email_table
email_table <- read.csv("email/email_table.csv")

## Emails which were opened
email_opened_table <- read.csv("email/email_opened_table.csv")

## Emails which were clicked
link_clicked_table <- read.csv("email/link_clicked_table.csv")

## Adding new column to emails opened and clicked tables which is equal to 1
email_opened_table$opened <- 1
link_clicked_table$clicked <- 1

## Merge email table with emails opened
data <- merge(email_table, email_opened_table, by = "email_id", all.x = T)

## Setting not opened emails to 0
data$opened <- as.character(data$opened)
data$opened[is.na(data$opened)] = "0"

## Merge clicked emails
data <- merge(data, link_clicked_table, by = "email_id", all.x = T)

## Setting not clicked emails to 0
data$clicked <- as.character(data$clicked)
data$clicked[is.na(data$clicked)] = "0"
```

Checking Data Quality

```
## Are there duplicates?
length(unique(data$email_id)) == length(data$email_id) ## looks good!
```

```
## [1] TRUE
```

```
## Checking whether there were any emails where the user clicked the link without opening it!
nrow(data[which(data$opened == 0 & data$clicked == 1), ]) ## this is an issue
```

```
## [1] 50
```

Let's remove these 50 observations from the data.

```
## Loading dplyr
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
```

```
## Removing the emails where the link was clicked without opening the email
data <- filter(data, (opened == 0 & clicked == 0) | (opened == 1 & clicked == 1) | (opened == 1 & click
```

```
##
str(data)
```

```
## 'data.frame':    99950 obs. of  9 variables:
##  $ email_id           : int  8 33 46 49 65 66 72 73 82 114 ...
##  $ email_text         : Factor w/ 2 levels "long_email","short_email": 2 1 2 1 2 1 2 1 1 2 ...
##  $ email_version      : Factor w/ 2 levels "generic","personalized": 1 2 1 2 1 1 1 1 2 2 ...
##  $ hour               : int  9 6 14 11 8 12 4 18 17 5 ...
##  $ weekday            : Factor w/ 7 levels "Friday","Monday",..: 5 2 6 5 7 7 3 5 5 7 ...
##  $ user_country       : Factor w/ 4 levels "ES","FR","UK",..: 4 4 4 4 3 4 4 2 1 4 ...
##  $ user_past_purchases: int  3 0 3 10 3 0 0 5 0 2 ...
##  $ opened             : chr  "0" "0" "0" "1" ...
##  $ clicked            : chr  "0" "0" "0" "0" ...
```

```
summary(data)
```

```
##     email_id            email_text         email_version
##  Min.   :     8   long_email :50248   generic     :50178
##  1st Qu.:246722   short_email:49702   personalized:49772
##  Median :498442
##  Mean   :498696
##  3rd Qu.:749937
##  Max.   :999998
##
##       hour             weekday      user_country user_past_purchases
##  Min.   : 1.000   Friday   :14165   ES: 9964     Min.   : 0.000
##  1st Qu.: 6.000   Monday   :14358   FR: 9989     1st Qu.: 1.000
##  Median : 9.000   Saturday :14564   UK:19928     Median : 3.000
##  Mean   : 9.059   Sunday   :14374   US:60069     Mean   : 3.879
##  3rd Qu.:12.000   Thursday :14274                3rd Qu.: 6.000
##  Max.   :24.000   Tuesday  :14137                Max.   :22.000
##                   Wednesday:14078
```

```
##     opened            clicked
##  Length:99950       Length:99950
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

What percentage of users opened the email and what percentage clicked on the link within the email ?

```
## Percentage of users who opened the email
(nrow(subset(data, data$opened == 1)) / nrow(data)) * 100
```

```
## [1] 10.35018
```

```
## Percentage of users who clicked on the link within the email
(nrow(subset(data, data$clicked == 1)) / nrow(data)) * 100
```

```
## [1] 2.070035
```

Exploratory Data Analysis

```
## Loading ggplot2
library(ggplot2)

## Changing the mode of opened and clicked to integer
data$opened <- as.integer(data$opened)
data$clicked <- as.integer(data$clicked)

## Data by email text
data_by_email_text <- data %>% group_by(email_text) %>% summarise(mean_opened = mean(opened), mean_click

## Looking at it!
data_by_email_text
```

```
## # A tibble: 2 x 3
##    email_text mean_opened mean_clicked
##        <fctr>       <dbl>        <dbl>
## 1  long_email  0.09122751   0.01799077
## 2 short_email  0.11591083   0.02343970
```

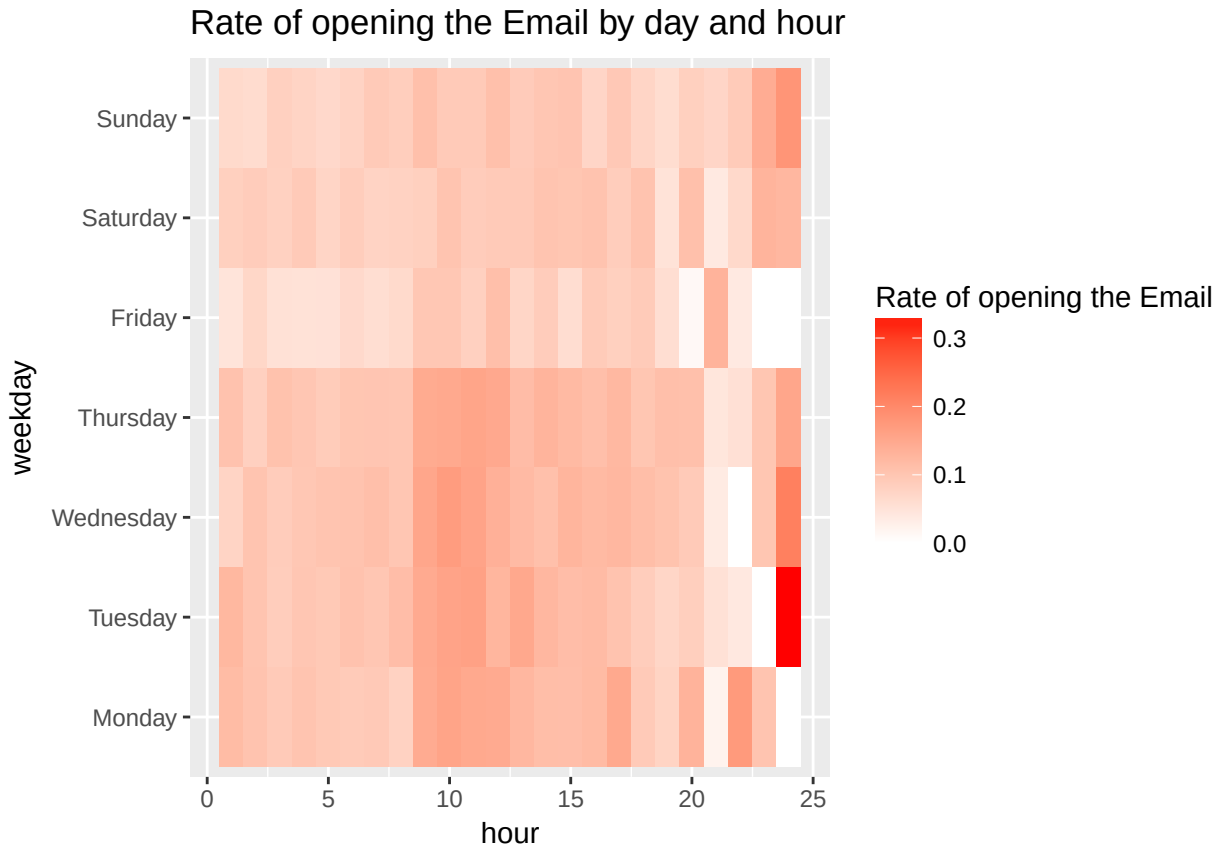
```
## Data by email version
data_by_email_version <- data %>% group_by(email_version) %>% summarise(mean_opened = mean(opened), mean

## Looking at it!
data_by_email_version
```

```
## # A tibble: 2 x 3
##   email_version mean_opened mean_clicked
##           <fctr>       <dbl>        <dbl>
## 1       generic  0.07939735   0.01452828
## 2  personalized  0.12780278   0.02692277
```

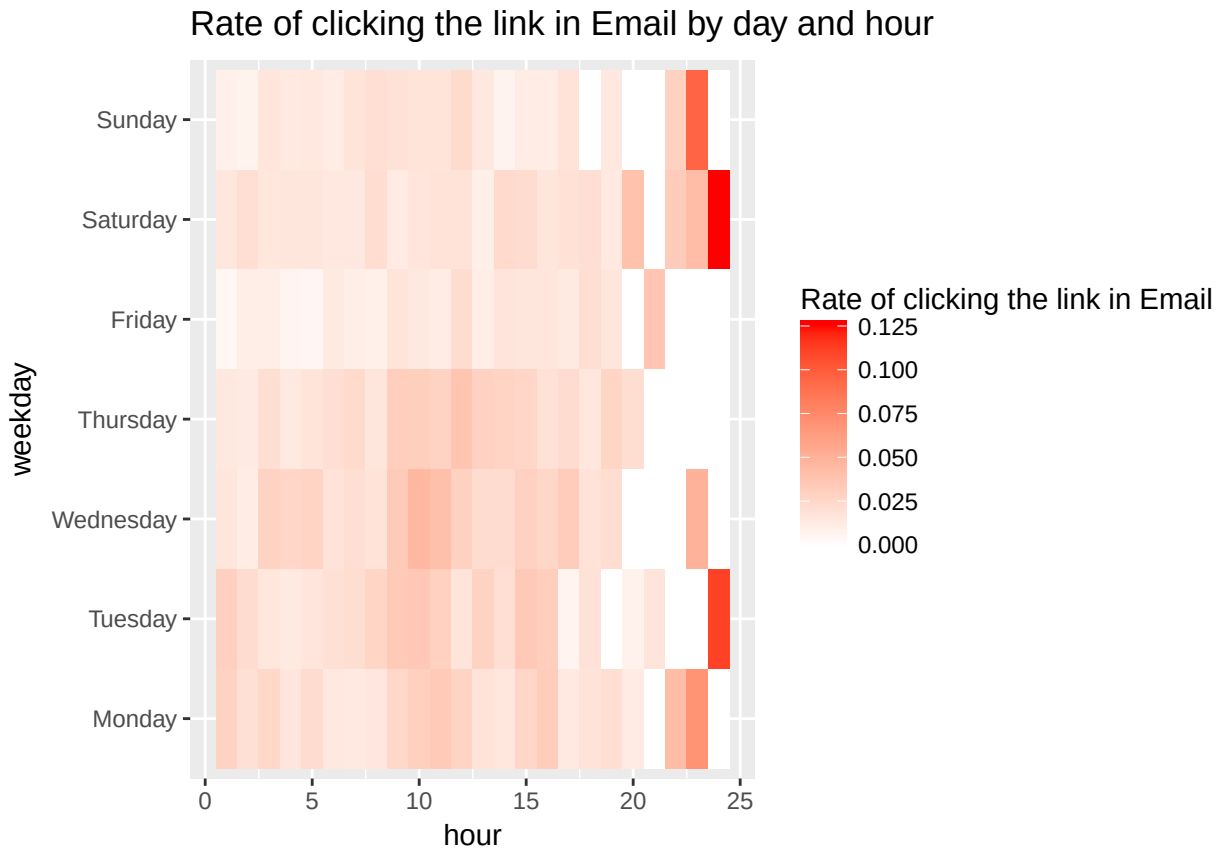
```
## Data by day and hour
data_by_day_and_hour <- data %>% group_by(weekday, hour) %>% summarise(mean_opened = mean(opened), mean_
```

```
## Changing the order of days
data_by_day_and_hour$weekday <- factor(data_by_day_and_hour$weekday, ordered = TRUE, levels=c("Monday",
```

```
## Visualizing the rate of opening an email by Heatmap
ggplot(data_by_day_and_hour, aes(x = hour, y = weekday)) + geom_tile(aes(fill = mean_opened)) + scale_f
```



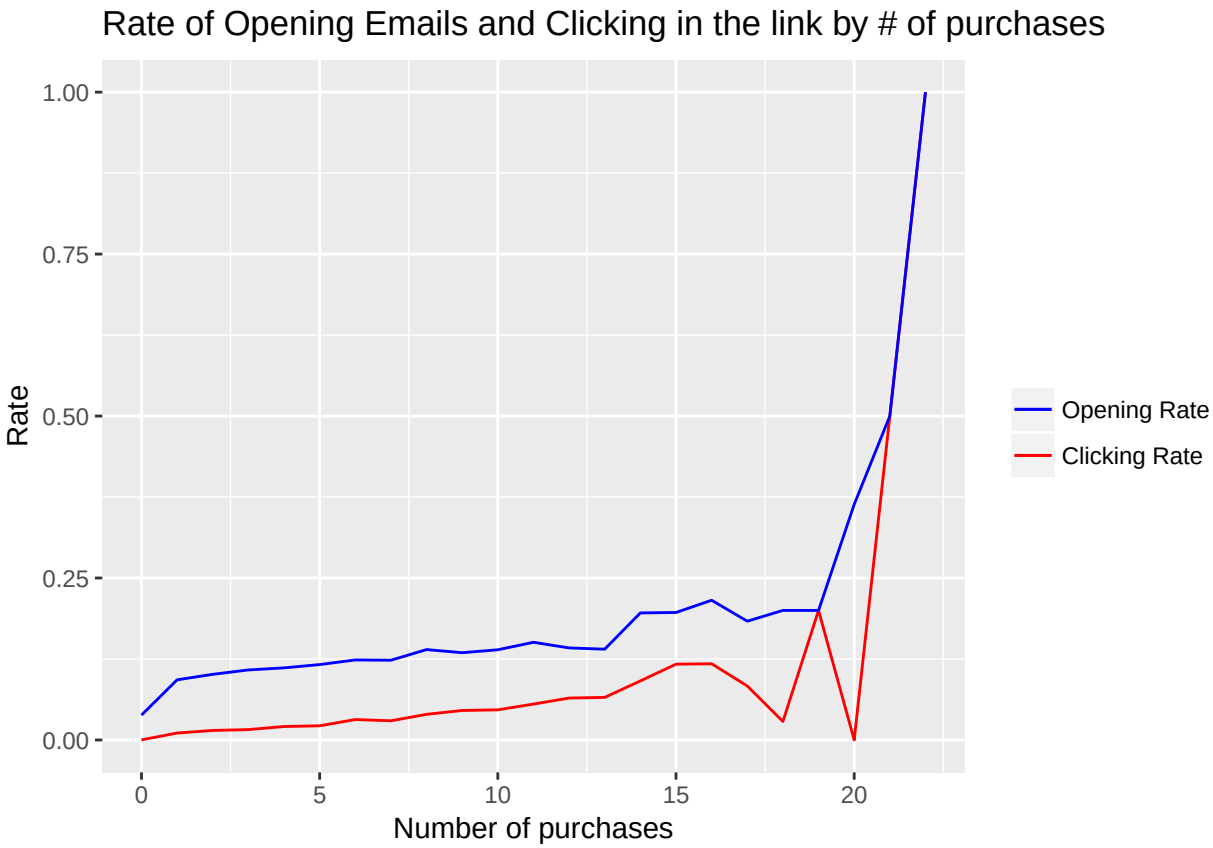

```
## Visualizing the rate of clicking the link in an email by Heatmap
ggplot(data_by_day_and_hour, aes(x = hour, y = weekday)) + geom_tile(aes(fill = mean_clicked)) + scale_
```




```
## Data by country
data_by_country <- data %>% group_by(user_country) %>% summarise(mean_opened = mean(opened), mean_click

## Looking at it!
data_by_country
```

```
## # A tibble: 4 x 3
##   user_country mean_opened mean_clicked
##          <fctr>       <dbl>        <dbl>
## 1           ES  0.03914091  0.008028904
## 2           FR  0.04064471  0.007408149
## 3           UK  0.12023284  0.024136893
## 4           US  0.11907972  0.023872547
```

```
## Data by number of user past purchases
data_by_user_past_purchases <- data %>% group_by(user_past_purchases) %>% summarise(mean_opened = mean(

## Visualizing it!
ggplot(data_by_user_past_purchases, aes(user_past_purchases, mean_clicked, col = "red")) + geom_line() +
```



The VP of marketing thinks that it is stupid to send emails to a random subset and in a random way. Based on all the information you have about the emails that were sent, can you build a model to optimize in future email campaigns to maximize the probability of users clicking on the link inside the email?

In order to optimize this, let's only consider those emails which get opened first.

```
## Loading rpart
library(rpart)

## Opened Emails
opened_emails <- filter(data, opened == 1)

## Decision tree to predict clicked
tree = rpart(clicked ~ ., data = opened_emails, control = rpart.control(minbucket = nrow(data)/100, max

## Looking at the tree
tree
```

```
## n= 10345
##
## node), split, n, deviance, yval
```

```
##       * denotes terminal node
##
## 1) root 10345 1655.2000 0.2000000
##   2) user_past_purchases< 5.5 6680  818.4693 0.1429641 *
##   3) user_past_purchases>=5.5 3665  775.3926 0.3039563 *
```

By how much do you think your model would improve click through rate (defined as # of users who click on the link / total users who received the email). How would you test that?

```
## Subset by # of user past purchases >=6
users_with_more_purchases <- filter(data, user_past_purchases >= 6)

## Expected click through rate (CTR)
(nrow(subset(users_with_more_purchases, users_with_more_purchases$clicked == 1)) / nrow(users_with_more
```

```
## [1] 4.056662
```

```
## Expected Percentage increase
(4.05 - 2.07) / 2.07
```

```
## [1] 0.9565217
```

```
## t-test
t.test(data$clicked, users_with_more_purchases$clicked)
```

```
##
##  Welch Two Sample t-test
##
## data:  data$clicked and users_with_more_purchases$clicked
## t = -15.607, df = 35680, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02236113 -0.01737141
## sample estimates:
##  mean of x  mean of y
## 0.02070035 0.04056662
```

By sending emails to users with atleast 6 purchases, we can almost double the click through rate. We can test this by a t-test.

Did you find any interesting pattern on how the email campaign performed for different segments of users? Explain.

```
## Mean of rate of clicking by country, given the user opened the email
clicked_by_country <- opened_emails %>% group_by(user_country) %>% summarise(mean_clicked = mean(clicke

## Merge the data by country
dat <- merge(data_by_country, clicked_by_country, by = "user_country")

## Renaming columns of dat
colnames(dat)[3:4] <- c("mean_clicked", "mean_clicked_given_opened")

## Looking at dat
dat
```

```
##   user_country mean_opened mean_clicked mean_clicked_given_opened
## 1           ES  0.03914091  0.008028904                 0.2051282
## 2           FR  0.04064471  0.007408149                 0.1822660
## 3           UK  0.12023284  0.024136893                 0.2007513
## 4           US  0.11907972  0.023872547                 0.2004753
```

I notice that the users of countries Spain and France have almost equal chances of clicking on the link in the email as that of US and UK users, once they open it. But due to some reason, they have a very low probability of opening the email as compared to US and UK users. This needs to be investigated further!

Other Interesting Results

1. Short and personalized emails were more effective than long and generic emails.
2. Most users are opening their email in the weekdays in the morning time (9 am to 12 pm). But we also see that many people opened the email on 24th hour of Tuesday.
3. The users are clicking on the link in the email mostly on Tuesday, Saturday or Sunday night. This information can be used in order to decide when the emails should be sent!

```
## Load the library dplyr
library(dplyr)

## Create data for each segment having CTR and number of users for that segment
data_for_each_segment <- data %>% group_by(email_text, email_version, hour, weekday, user_country, user

## CTR
weighted.mean(data_for_each_segment$click_through_rate, data_for_each_segment$number_of_users / sum(data
```

```
## [1] 0.02070035
```

```
## CTR (same as above; just confirming whether I am doing it correctly)
(nrow(subset(data, data$clicked == 1)) / nrow(data))
```

```
## [1] 0.02070035
```

```r
## Estimating maxiumum CTR for each segment (Grouped by country and past purchases as they can't be char
data_to_estimate_max_ctr <- data_for_each_segment %>% group_by(user_country, user_past_purchases) %>% su

## Maximum CTR expected
weighted.mean(data_to_estimate_max_ctr$max_ctr, data_to_estimate_max_ctr$number_of_users / sum(data_to_e
```

```
## [1] 0.6820054
```