Project Summary

Exploratory Data Analysis of Ad Click-Through Rate (CTR)

Matthew Joel

Introduction

Question of interest.

The question of interest I would like to answer is what variables have the most effect on ad click through rate (CTR). Additionally, is there any way we can predict an ads performance based on these variables.

Importance

This topic is important because online advertising is a generally low yield endeavor. Industry average CTRs are between 0.46% - 8% depending on the domain. Running each ad costs some amount of time and money, and to maximize results we want our ads to be as efficient as possible (high CTR). This practically means that we want our ads to be targeted, and only shown to the most relevant of audiences.

Background

The data set I will be using is titled **CTR In Advertisement.** It can be found freely on Kaggle here.

Some context from Kaggle: *A company wants to know the CTR (Click Through Rate) in order to identify whether spending their money on digital advertising is worth or not. A higher CTR represents more interest in that specific campaign, whereas a lower CTR can show that the ad may not be as relevant.*

My analysis of this data-set would allow a potential business to better know their target audience, and thus achieve higher CTRs for future ad campaigns.

Variables

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.3      v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ggplot2)

ads <- read.csv("Ad_click_prediction_train.csv")
str(ads)
```

```
## 'data.frame':    463291 obs. of  15 variables:
##  $ session_id            : int  140690 333291 129781 464848 90569 151475 17583 461128 390699 353607
##  $ DateTime              : chr  "2017-07-02 00:00" "2017-07-02 00:00" "2017-07-02 00:00" "2017-07-02
##  $ user_id               : int  858557 243253 243253 1097446 663656 509591 1091463 469098 611906 418
##  $ product               : chr  "C" "C" "C" "I" ...
##  $ campaign_id           : int  359520 105960 359520 359520 405490 359520 405490 360936 105960 36093
##  $ webpage_id            : int  13787 11085 13787 13787 60305 13787 60305 13787 11085 13787 ...
##  $ product_category_1    : int  4 5 4 3 3 2 3 3 5 2 ...
##  $ product_category_2    : num  NA NA NA NA NA ...
##  $ user_group_id         : num  10 8 8 3 2 1 9 4 NA 4 ...
##  $ gender                : chr  "Female" "Female" "Female" "Male" ...
##  $ age_level             : num  4 2 2 3 2 1 3 4 NA 4 ...
##  $ user_depth            : num  3 2 2 3 3 3 3 3 NA 3 ...
##  $ city_development_index: num  3 NA NA 2 2 NA 4 4 NA 4 ...
##  $ var_1                 : int  0 0 0 1 1 0 0 0 0 0 ...
##  $ is_click              : int  0 0 0 0 0 0 0 0 0 0 ...
```

name	type	description
is_click	numeric (discrete)	Binary indicator of ad click (0 = No, 1 = Yes)
session_id	numeric (continous)	Unique identifier for each session
DateTime	character	Date and time of the session
user_id	numeric (discrete)	Unique identifier for each user
product	character	Product identifier
campaign_id	numeric (discrete)	Campaign identifier
webpage_id	numeric (discrete)	Webpage identifier
product_category_1	numeric (discrete)	Category of the product (numeric)
product_category_2	numeric (continous)	Additional category of the product (numeric)
user_group_id	numeric (discrete)	User group identifier (numeric)
gender	character	Gender of the user
age_level	numeric (continous)	Age level of the user (numeric)
user_depth	numeric (continous)	Depth of the user
city_development_index	numeric (continous)	City development index (numeric)
var_1	numeric (discrete)	Binary variable (0 or 1)

Data cleaning

1. I started of by removing the `product_category_2` column. This column was not tidy, and had many N/As and out-liers. Additionally, it was not relevant to my analysis.
2. Then, I removed any incomplete rows with `na.omit()`. While this operation removed quite a few, this data set is so large that we are still left with 338,162 observations!
3. Next, I converted `DateTime` from a `chr` type to a date type. Because this variable was date AND time, I couldn't use the `as.Date()` function. Instead I used `as.POSIXct()`. Now that `DateTime` is stored as this data type, I can easily split the date and time into separate columns `date` and `time`. `time` is an int which makes it easier to use as a continuous variable. Finally I relocate the columns and then remove the original `DateTime`.

4. This data-set is almost all categorical in nature, meaning I needed to factor most of the variables. The only exceptions were of course the unique, individual identifies, such as `session_id` and `user_id`, and the dates/times.

5. Lastly, I created subsets for each of the 10 products. Why? Well, because for some analyses we will be interested in *Product-CTR pairs*. We don't want to compare apples and oranges, but rather for some questions we will be rather interested in the metrics for a specific product.

Numeric summaries

```
# my code
meanCTRa <- mean(productA$is_click == "1")
meanCTRb <- mean(productB$is_click == "1")
meanCTRc <- mean(productC$is_click == "1")
meanCTRd <- mean(productD$is_click == "1")
meanCTRe <- mean(productE$is_click == "1")
meanCTRf <- mean(productF$is_click == "1")
meanCTRg <- mean(productG$is_click == "1")
meanCTRh <- mean(productH$is_click == "1")
meanCTRi <- mean(productI$is_click == "1")
meanCTRj <- mean(productJ$is_click == "1")

prodCTR <- c(meanCTRa, meanCTRb, meanCTRc, meanCTRd, meanCTRe, meanCTRf, meanCTRg, meanCTRh, meanCTRi, m

meanCTR <- mean(prodCTR)
meanCTR
```

```
## [1] 0.0646957
```

```
fivenum(prodCTR)
```

```
## [1] 0.04613500 0.05417249 0.06556571 0.06987793 0.09428951
```

Because my response variable is binary (clicked or didn't), we will instead look at the CTR per product. The mean CTR is 6.76%. The fivenum is:

- **Minimum (Min):** 0.04613500
- **First Quartile (Q1):** 0.05417249
- **Median (Q2):** 0.06556571
- **Third Quartile (Q3):** 0.06987793
- **Maximum (Max):** 0.09428951

We can see that all products have closely packed CTRs. There are not many out-liers, variability is low, and there are no unusual values either.

Numeric summary of `time`

```r
clicked <- ads %>% filter(is_click == 1)

convert_to_regular_time <- function(time_int) {
  hours <- floor(time_int / 60)
  minutes <- time_int %% 60

  if (hours >= 12) {
    period <- "PM"
    if (hours > 12) {
      hours <- hours - 12
    }
  } else {
    period <- "AM"
    if (hours == 0) {
      hours <- 12
    }
  }

  formatted_time <- sprintf("%02d:%02d %s", as.integer(hours), as.integer(minutes), period)
  return(formatted_time)
}

mean_time_clicked <- mean(clicked$time)
mean_time_clicked
```

```
## [1] 815.4099
```

```r
(fivenum(clicked$time))
```

```
## [1]    0  552  808 1109 1439
```

```r
convert_to_regular_time(mean_time_clicked)
```

```
## [1] "01:35 PM"
```

```r
convert_to_regular_time(0)
```

```
## [1] "12:00 AM"
```

```r
convert_to_regular_time(552)
```

```
## [1] "09:12 AM"
```

```r
convert_to_regular_time(808)
```

```
## [1] "01:28 PM"
```

```r
convert_to_regular_time(1109)
```

```
## [1] "06:29 PM"
```

```r
convert_to_regular_time(1439)
```

```
## [1] "11:59 PM"
```

```r
# my code
```

The mean time for an ad click is 1:35pm. The fivenum is:

- **Minimum (Min):** 12:00 AM
- **First Quartile (Q1):** 9:12 AM
- **Median (Q2):** 1:28 AM
- **Third Quartile (Q3):** 6.29 PM
- **Maximum (Max):** 11:59 PM

As we would expect, the minimum time is 12:00 AM and maximum time is 11:59. The mean and median time are close together, suggesting that the data is symmetrically distributed. There also appears to be very little skewing, as the Q1 is about as close to the median as Q3 is.

Numeric summary of `age_level`

```r
ageGroup1 <- subset(ads, age_level == "1")
ageGroup2 <- subset(ads, age_level == "2")
ageGroup3 <- subset(ads, age_level == "3")
ageGroup4 <- subset(ads, age_level == "4")
ageGroup5 <- subset(ads, age_level == "5")
ageGroup6 <- subset(ads, age_level == "6")
ageGroup0 <- subset(ads, age_level == "0")


meanCTRage1 <- mean(ageGroup1$is_click == "1")
print(meanCTRage1)
```

```
## [1] 0.07353462
```

```r
meanCTRage2 <- mean(ageGroup2$is_click == "1")
print(meanCTRage2)
```

```
## [1] 0.07067268
```

```r
meanCTRage3 <- mean(ageGroup3$is_click == "1")
print(meanCTRage3)
```

```
## [1] 0.06571972
```

```r
meanCTRage4 <- mean(ageGroup4$is_click == "1")
print(meanCTRage4)
```

```
## [1] 0.06003841
```

```r
meanCTRage5 <- mean(ageGroup5$is_click == "1")
print(meanCTRage5)
```

```
## [1] 0.07296501
```

```r
meanCTRage6 <- mean(ageGroup6$is_click == "1")
print(meanCTRage6)
```

```
## [1] 0.07717215
```

```r
meanCTRage0 <- mean(ageGroup0$is_click == "1")
print(meanCTRage0)
```

```
## [1] 0.09375
```

Here we have the mean CTR by age level. We can see that group 0 has the highest CTR at 9.3%, while group 4 has the lowest at 6.0%. They are all close together, with the median being higher than the mean, suggesting a skew right.

Numeric summary of gender

```r
propClicksFemale <- mean(clicked$gender == "Female")

propClicksMale <- mean(clicked$gender == "Male")

print(paste("Proportion of clicks for Females:", propClicksFemale))
```

```
## [1] "Proportion of clicks for Females: 0.107225699155324"
```

```r
print(paste("Proportion of clicks for Males:", propClicksMale))
```

```
## [1] "Proportion of clicks for Males: 0.892774300844676"
```

```r
propClicksFemale+propClicksMale
```

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

```r
female <- subset(ads, gender == "Female")
nrow(female)
```

```
## [1] 38493
```

```r
male <- subset(ads, gender == "Male")
nrow(male)
```

```
## [1] 299669
```

```r
mean(female$is_click=="1")
```

```
## [1] 0.06364794
```

```r
mean(male$is_click=="1")
```

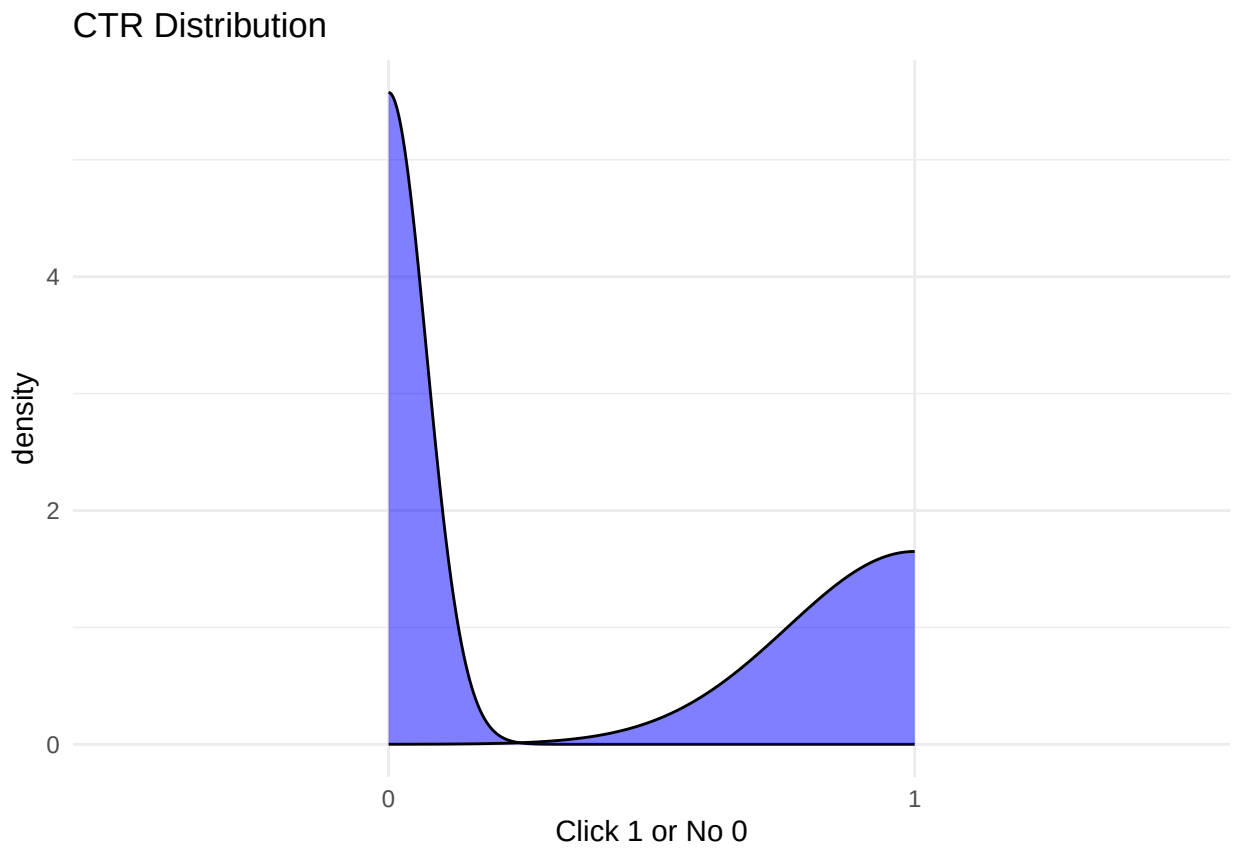
```
## [1] 0.06807177
```

Since gender is a binary variable, we won't do the typical summary on it. Instead, we will look at proportions of clicks. From this, we see that only around 10% or clicks are from females, while almost 90% are from males. From just this information, we might conclude that males are more likely to click ads. However, this doesn't tell the full story. When we count total males and females, we see there are about 8 times as many rows with males vs females, thus explaining the discrepancy (299669 and 38493 respectively). Lastly, when we compare the actual rates, we see 0.06364794 for women and 0.06807177 for men, significantly closer together.

Univariate graphics

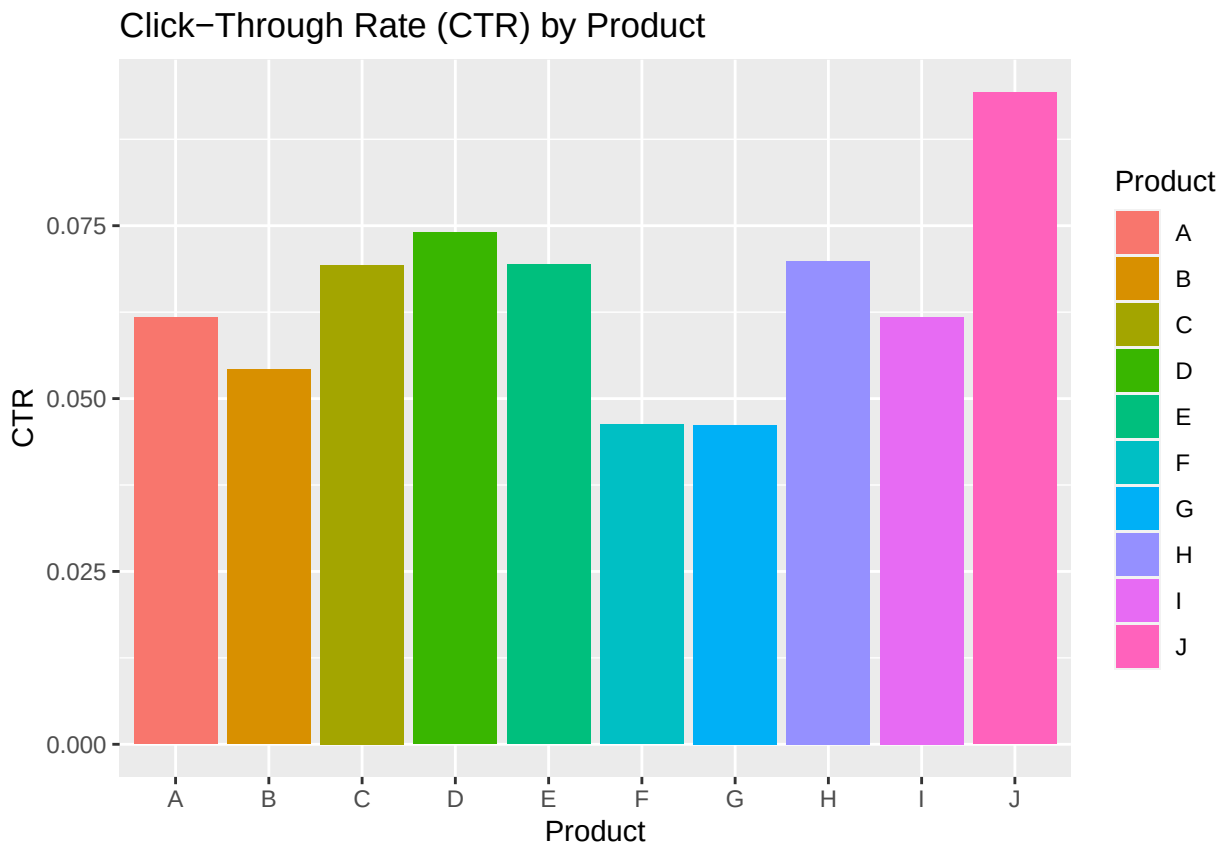
Density plot of `is_click`

```r
ggplot(ads, aes(x = is_click)) +
geom_density(fill = "blue", alpha = 0.5) +
labs(title = "CTR Distribution", x = "Click 1 or No 0") +
theme_minimal()
```



This graph doesn't tell us much, as `is_click` is a binary variable. Instead, lets look at CTR's per product:

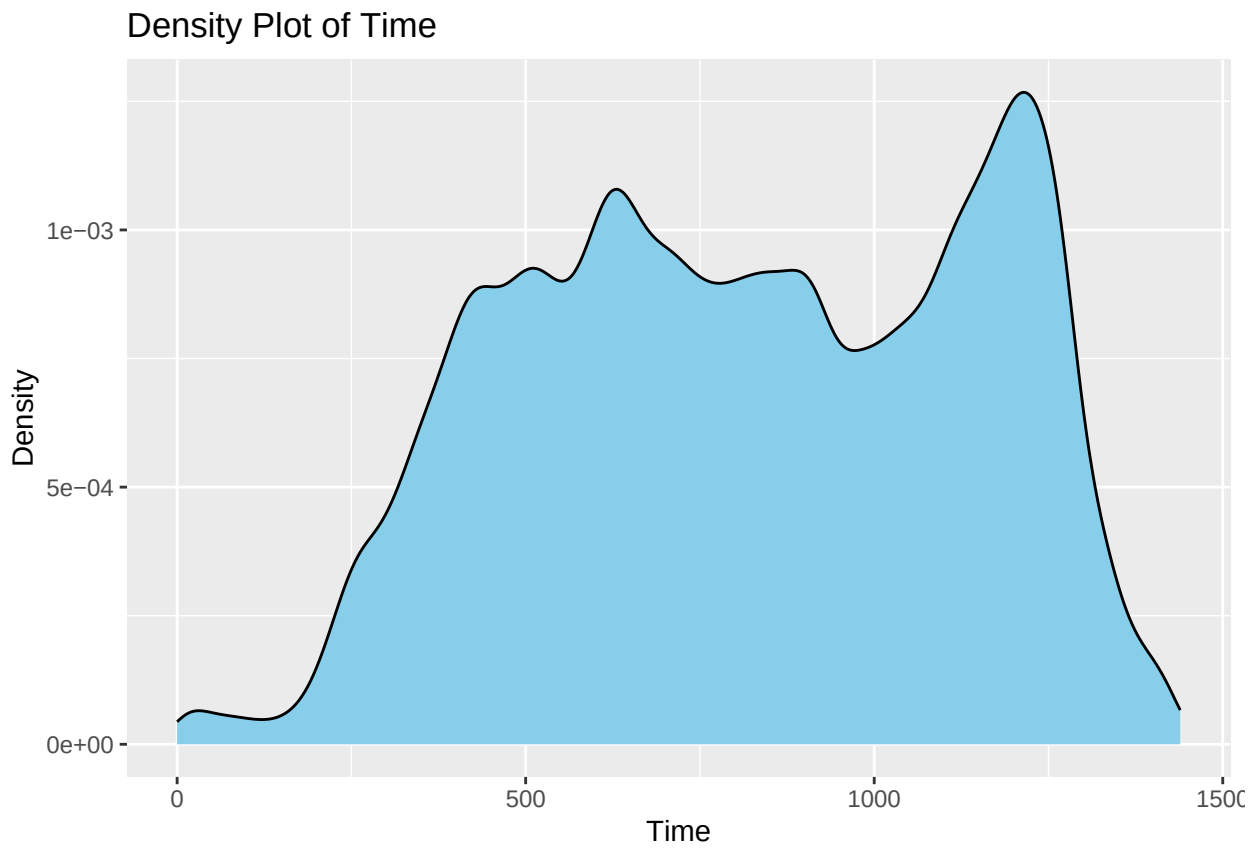
```
ctr_data <- data.frame(Product = c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J"),CTR = prodCTR)
ggplot(ctr_data, aes(x = Product, y = CTR, fill = Product)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Click-Through Rate (CTR) by Product",
       x = "Product",
       y = "CTR")
```



Here, we can see that J has the best CTR, while G is the lowest. This data would most closely follow a uniform distribution, as all products have a similar CTR.

Density plot of `time`

```
ggplot(ads, aes(x = time)) +
  geom_density(fill = "skyblue", color = "black") +
  labs(title = "Density Plot of Time",
       x = "Time",
       y = "Density")
```




```
convert_to_regular_time(600)
```

```
## [1] "10:00 AM"
```

```
convert_to_regular_time(1100)
```

```
## [1] "06:20 PM"
```

This density plot tells us quite a bit about the distribution of time. We can see that it is bimodal, with the first peak around 600 (10:00AM) and the second peak at 1100 (6:20PM). While there is most traffic in the morning and evening, the mean time is around 1:35PM, between the two. If we looked at just the mean, we would not know this part of the story.

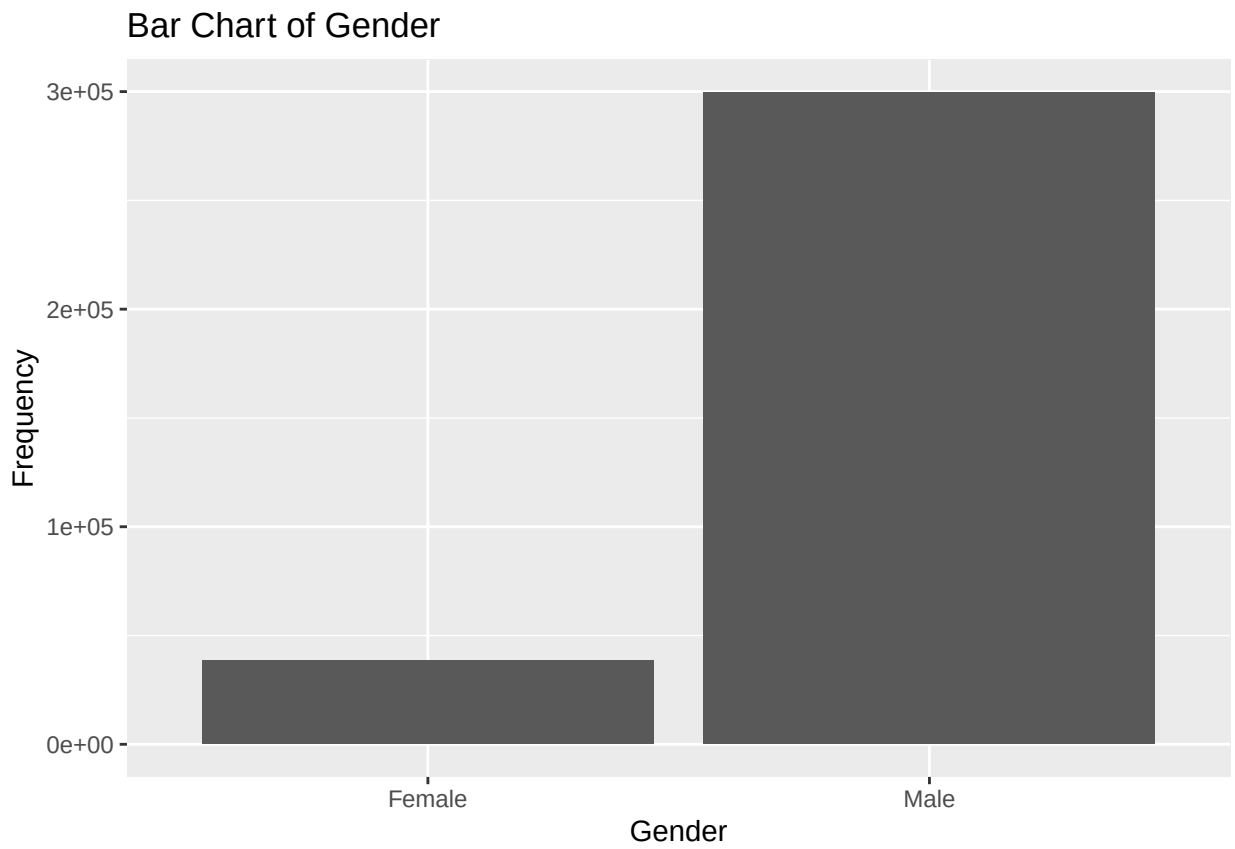
Density plot of `age_level`

```
ggplot(ads, aes(x = (age_level), fill = (age_level))) +
  geom_density(alpha = 0.7) +
  labs(title = "Density Plot of Age Level",
       x = "Age Level",
       y = "Density")
```

We can now see that most web traffic belongs to levels 1 and 2, and the least belongs to group 6. The groups seem to be normally distributed, and thus unimodal. Though, they do have a large skew right. This suggest that the mode is to the left of the mean.

Bar chart of `gender`

```
ggplot(ads, aes(x = gender)) +
  geom_bar(position = "dodge") +
  labs(title = "Bar Chart of Gender",
       x = "Gender",
       y = "Frequency")
```

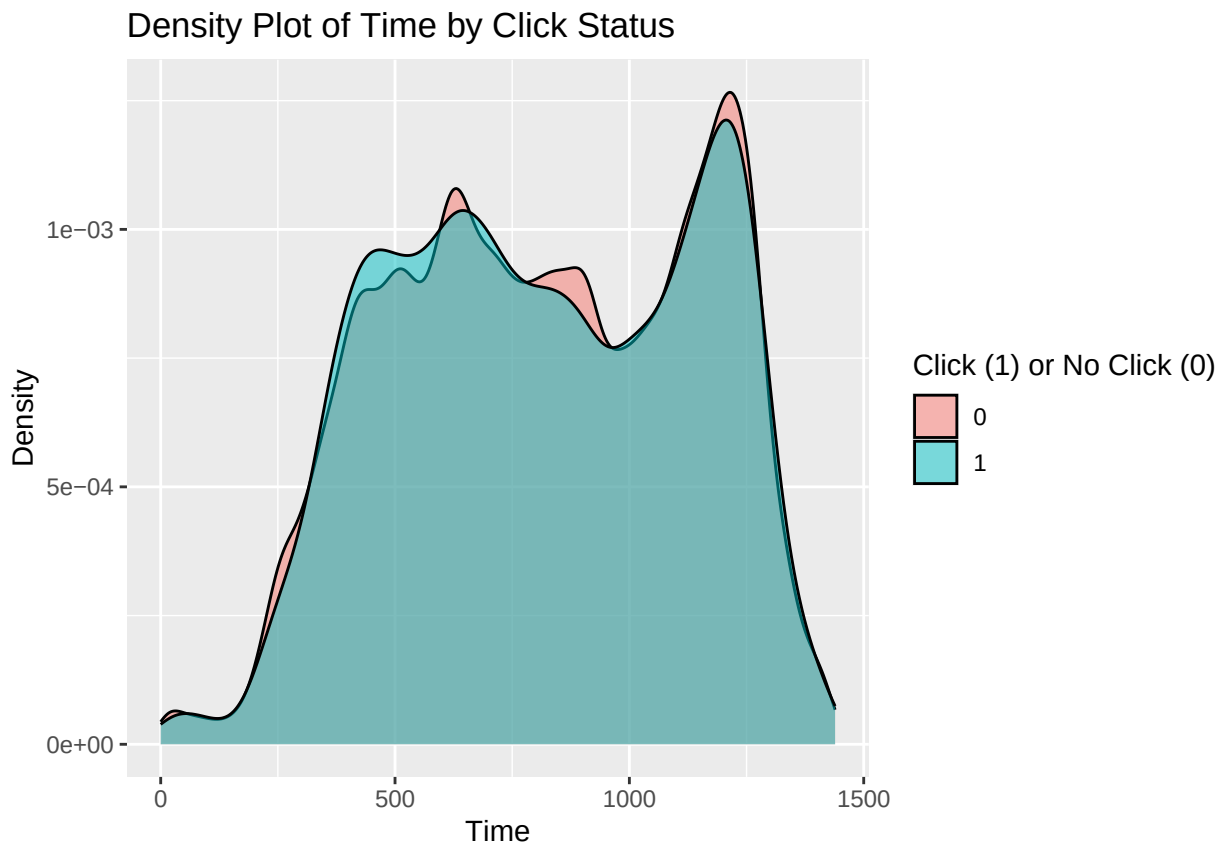


We can see that there is significantly more male web traffic than female in this data set. As this is univariate, we are not told anything about their respective click through rates. Rather, we can only say what proportion of web traffic is male or female. It is not clear why this data set has so few female observation, as there is almost 8 times as many male observations.

Bivariate graphics

Density plot of `is_click` versus `time`

```
ggplot(ads, aes(x = time, fill = as.factor(is_click))) +
  geom_density(alpha = 0.5, color = "black") +
  labs(title = "Density Plot of Time by Click Status",
       x = "Time",
       y = "Density",
       fill = "Click (1) or No Click (0)")
```




```
convert_to_regular_time(850)
```

```
## [1] "02:10 PM"
```

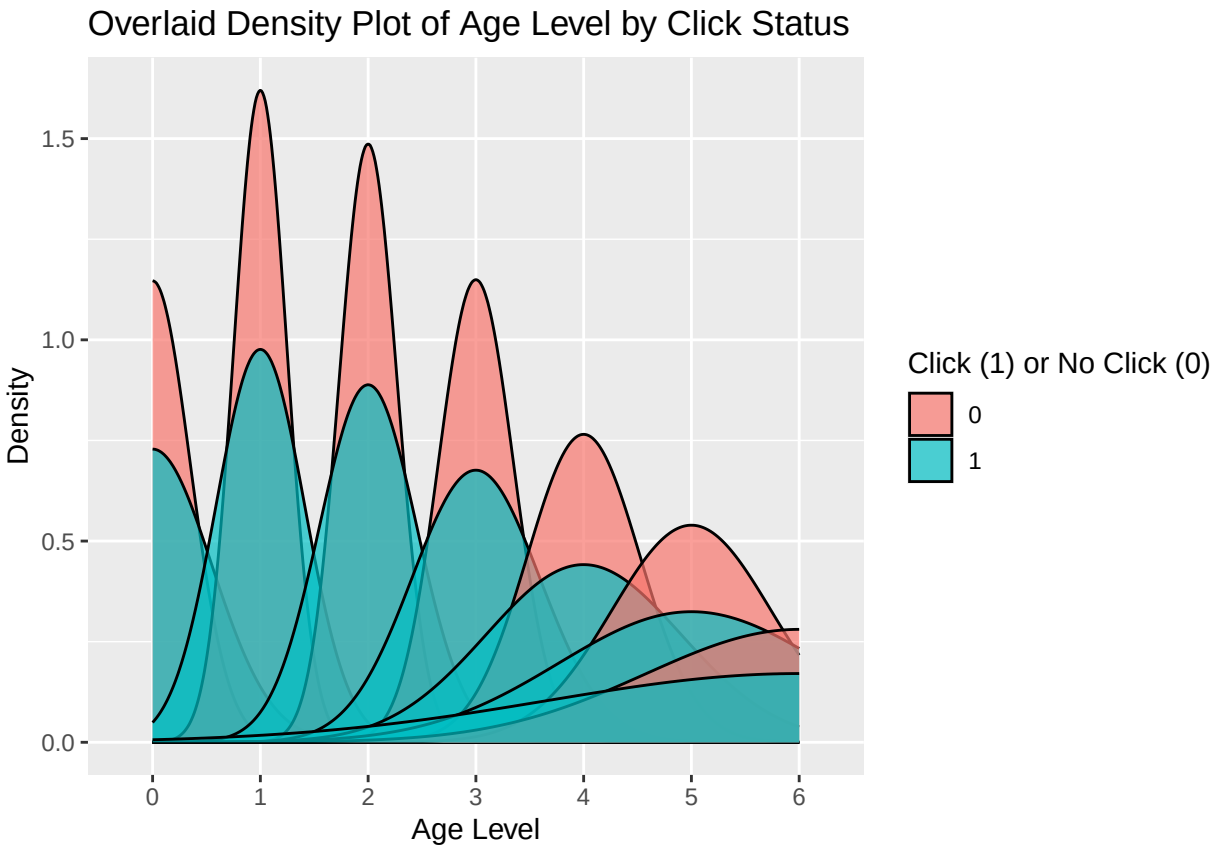
```
convert_to_regular_time(325)
```

```
## [1] "05:25 AM"
```

From this graph, we can see that CTR generally follows web traffic. In practical terms, this means that we expect clicks to go up as traffic goes up. In line with what we'd think. There are a few exceptions however. At 600 (10:00AM), 850 (2:10PM), and at 1100 (6:20PM), we see that CTR actually slows down compared to overall web traffic, indicated by the orange being above the blue area. The opposite is true at 325 (5:25AM), where the click rate briefly exceeds web traffic.

Density plot of `is_click` versus `age_level`

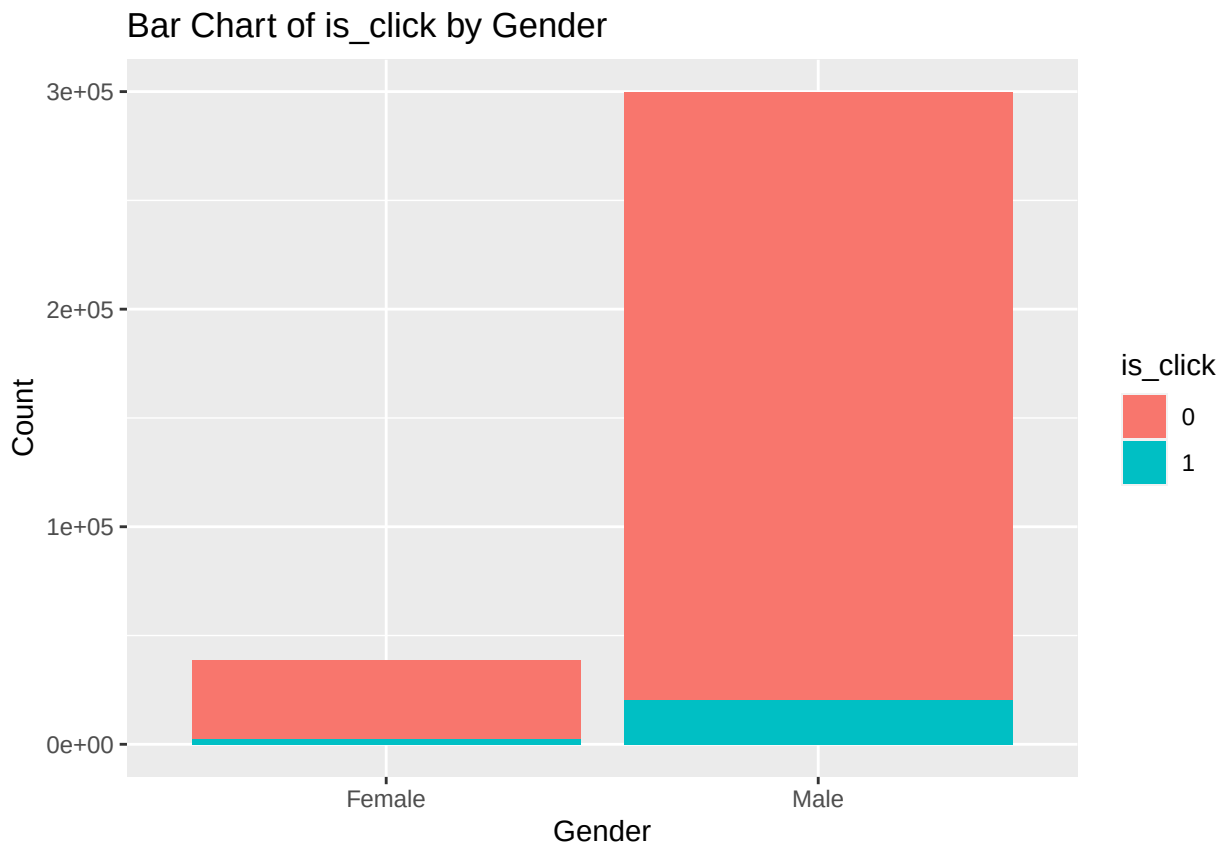
```
ggplot(ads, aes(x = age_level, fill = (is_click))) +
  geom_density(alpha = 0.7, position = "identity") +
  labs(title = "Overlaid Density Plot of Age Level by Click Status",
       x = "Age Level",
       y = "Density",
       fill = "Click (1) or No Click (0)")
```



This graph is rather uneventful. We can see that all age levels have about the same ratios among their CTR's. In other words, CTR is not directly correlated with age level.

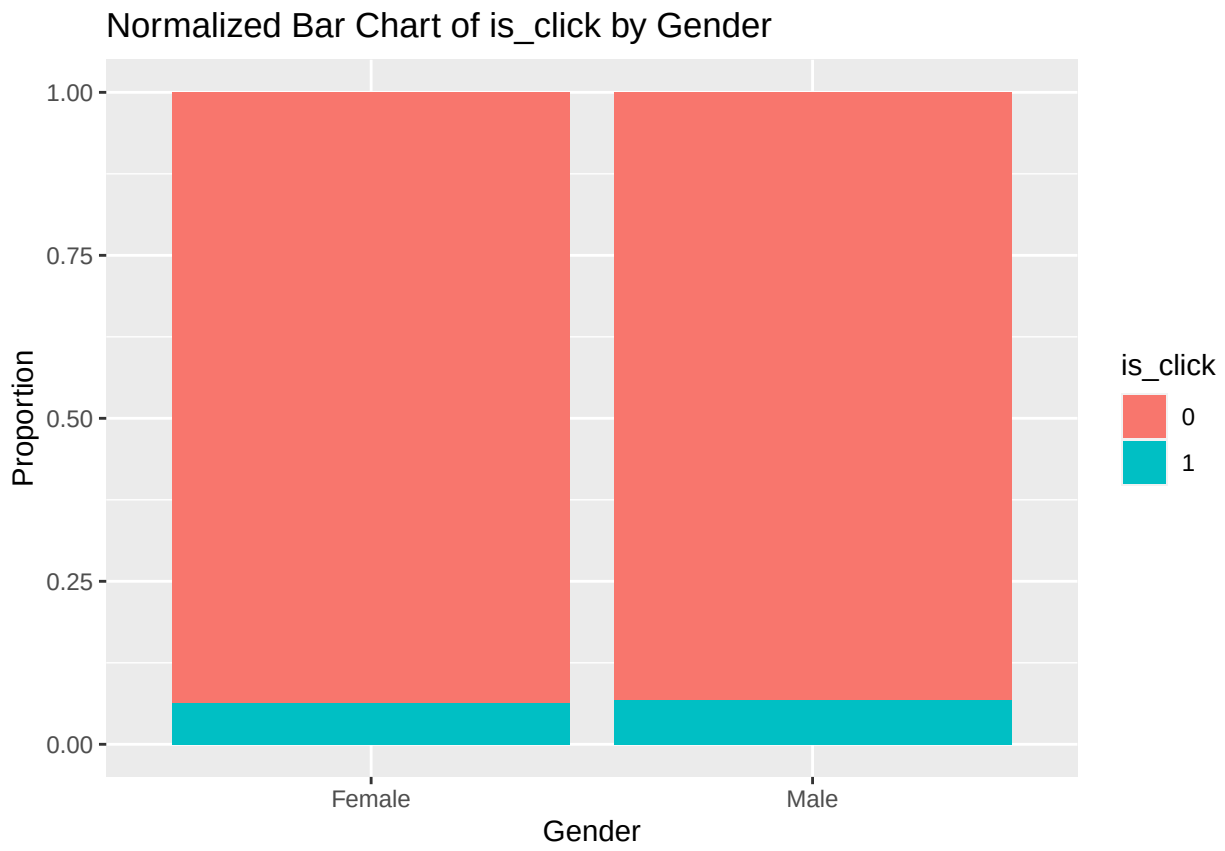
Barplot of `is_click` versus `gender`

```
ggplot(ads, aes(x = gender, fill = (is_click))) +
  geom_bar(position = "stack") +
  labs(title = "Bar Chart of is_click by Gender",
       x = "Gender",
       y = "Count",
       fill = "is_click")
```



This graph shows us about the same info as the univariate did. It shows that males have significantly more observations than females. The only difference is we can also see CTRs as well. To get a better idea, let's normalize the graph:

```
ggplot(ads, aes(x = gender, fill = (is_click))) +
  geom_bar(position = "fill") +
  labs(title = "Normalized Bar Chart of is_click by Gender",
       x = "Gender",
       y = "Proportion",
       fill = "is_click")
```



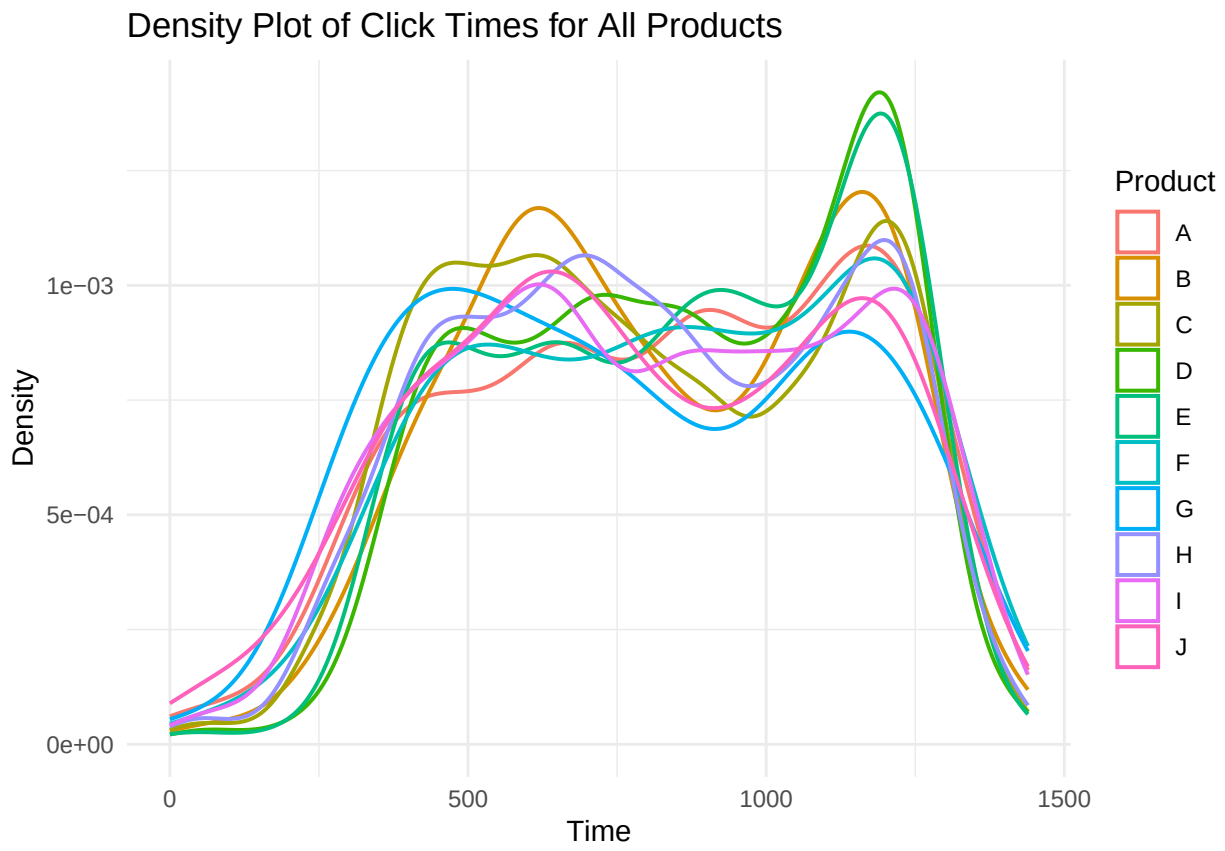
Now that its normalized, we can view the proportion of male and female clicks. And from that, we can see that there is little difference between their respective CTR's. Males and females, on average, click around 6-7%.

Multivariate graphics

Multivariate graphic 1: Response = is_click, Variables = time and product

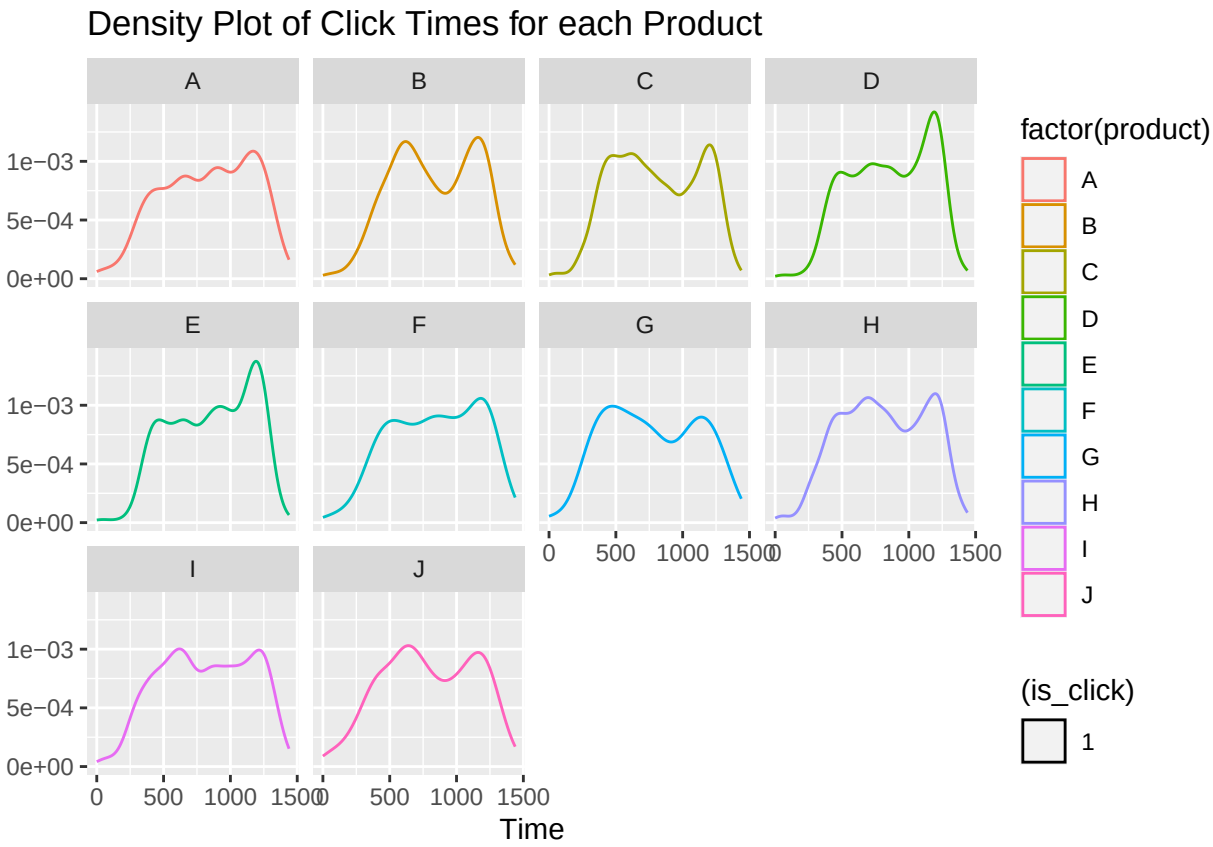
```
ggplot(clicked, aes(x = time, color = factor(product), linetype = (is_click))) +
  geom_density(size = .7, linetype = "solid") +
  labs(title = "Density Plot of Click Times for All Products",
       x = "Time",
       y = "Density",
       color = "Product") + theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
```



This multivariate graphic shows us the density of click times for all products (A-J). In other words, we can see *when* a given product is clicked most often. For the most part, it seems like all products follow a similar bimodal distribution, with peaks around 600 (10:00AM) and the second peak at 1100 (6:20PM). Let's take a closer look to see if this is truly the case:

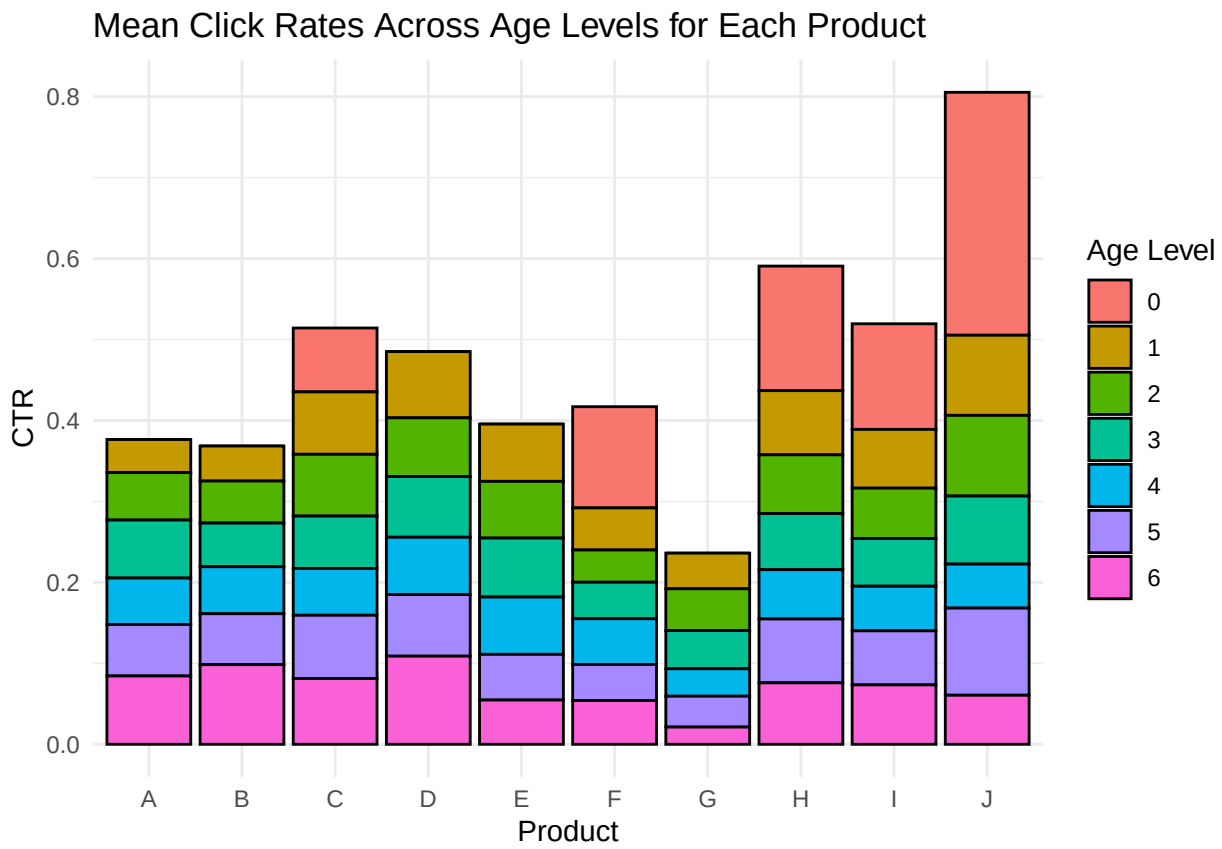
```
ggplot(clicked, aes(x = time, color = factor(product), linetype = (is_click))) +
  geom_density(alpha = 0.0) +
  facet_wrap(~ product) +
  labs(title = "Density Plot of Click Times for each Product",
       x = "Time",
       y = "Density"+ theme_minimal())
```



While this supplementary graph is not quite multivariate, it is crucial in order to understand the products in more detail. For example in almost all graphs, the second peak has greater magnitude than the first. This is not the case for G and J. These products have greater CTR in the morning time. Analyzing this data would allow a business/advertiser to most efficiently list their ad in regards to time.

Multivariate graphic 2: Response = is_click, Variables = age_level and product

```
ads2 <- ads
ads2$is_click <- as.numeric(as.character(ads2$is_click))
ggplot(ads2, aes(x = factor(product), y = is_click, fill = factor(age_level))) +
  geom_bar(stat = "summary", fun = "mean", position = "stack", color = "black") +
  labs(title = "Mean Click Rates Across Age Levels for Each Product",
       x = "Product",
       y = "CTR",
       fill = "Age Level") +
  theme_minimal()
```

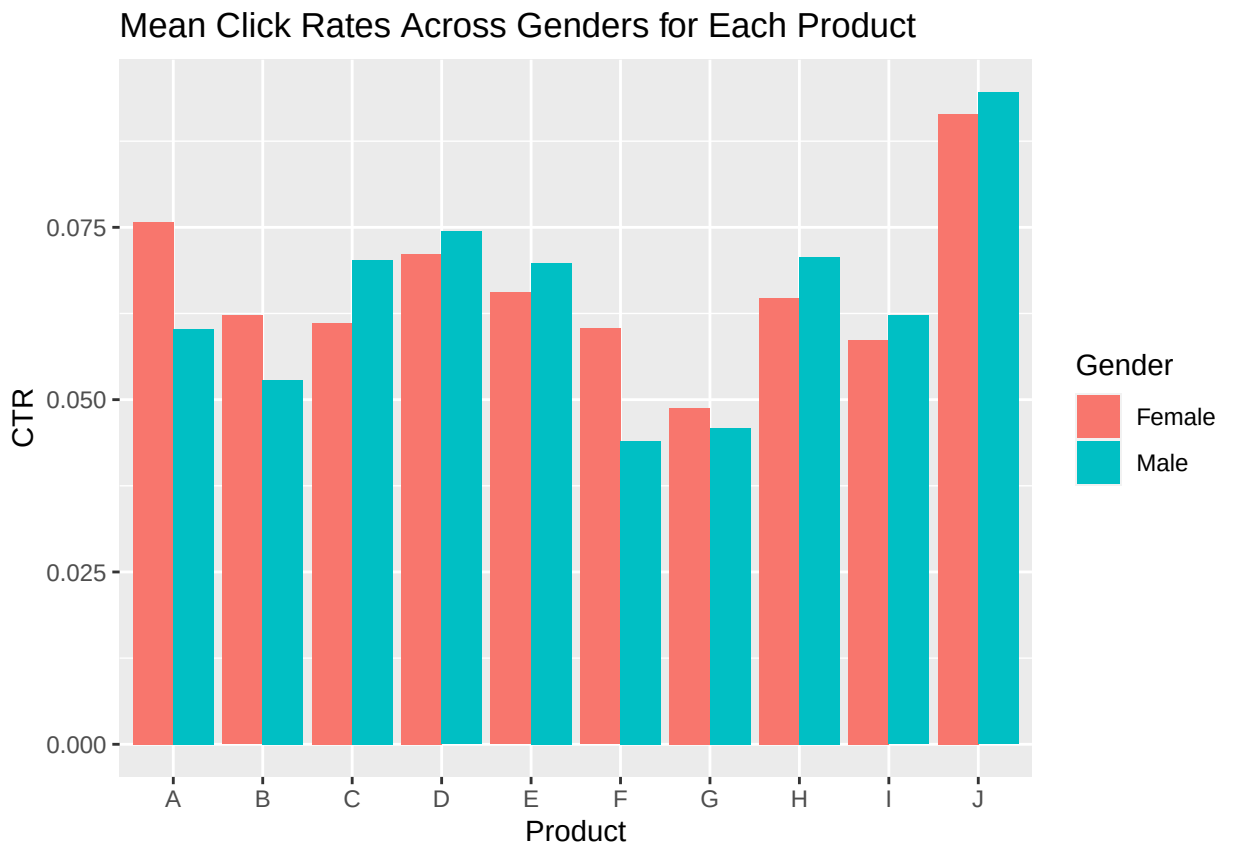


This multivariate graph shows us which `age_level` clicks the most(`is_click`) for a given `product`. In other words, we can see which ages are interested in a specific product. For example, we can see that product J has its majority of clicks from `age_level` 0. Perhaps J is a kids toy or show, and thus most popular with the lowest age level. Using this information, we could come up with the optimal distribution of age groups to advertise a given product, thus maximizing our CTR. For example, product B should be shown equally to `age_levels` 0 through 5, and shown a little more to `age_level` 6. Awesome!

Multivariate graphic 2: Response = is_click, Variables = gender and product

```
mean_clicks <- aggregate(is_click ~ product + gender, data = ads, FUN = function(x) mean(as.numeric(as.

ggplot(mean_clicks, aes(x = product, y = is_click, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge")+
  labs(title = "Mean Click Rates Across Genders for Each Product",
       x = "Product",
       y = "CTR",
       fill = "Gender")
```



This plot allows us to see who (male or female) is clicking on a given product. For example, we can now see that product A has more interest by females than males, and the reverse is true for H. This would allow us to more efficiently allocate advertisements to the more relevant gender

Conclusions

While click through rate depends on many variables, taking a multi-faceted approach can help us analyze our target audience and thus foster data-driven decision making. For example, if I was told to run an advertisement for `Product J`, I would:

- run the ads mainly in the morning, and secondarily in the evening
- push the ad mainly to younger audiences, like `age_level` 0
- target the ad slightly more towards men

Insights like this help maximize our CTR and make our ads more effective. Thanks to the data analysis and graphics I now have, you could give me any product A - J and I could tell you when, what age, and what gender to target the ad at.

CTR is hard to accurately predict using only a single metric. However if I had to pick one, it would probably be `time`. For almost all products, CTR was maximized around the same time frames. This could be simply due to the fact that most people use their phones/electronic devices around the same times, or it could be due to something far more complex. While the analysis I conducted most certainly can't tell us the *whys,* it can likely tell us the *whats*.

In light of these findings, I believe that phone-usage patterns, age demographics, and gender preferences should be further explored in the context of advertising. While not covered in my analysis, I additionally think visual components of an ad should be further examined: what fonts do people like to see, which colors captivate the most attention, etc. Most of advertising history has gone down as a story of brute force, where the ad is pushed to as many people to maximize conversion. I hope that one day we can say goodbye to crappy, irrelevant ads and instead see ones that are interesting and might even make us laugh a bit.

Appendix

```
ads <- subset(ads, select = -product_category_2)
ads <- na.omit(ads)

ads$DateTime <- as.POSIXct(ads$DateTime, format = "%Y-%m-%d %H:%M")
ads$date <- as.Date(ads$DateTime)
ads <- ads %>% relocate(date, .after = DateTime)
ads$hour <- as.integer(format(ads$DateTime, "%H"))
ads$minute <- as.integer(format(ads$DateTime, "%M"))
ads$time <- ads$hour * 60 + ads$minute
ads <- ads %>% relocate(time, .after = date)
ads <- select(ads, -DateTime, -hour, -minute)

ads$product <- as.factor(ads$product)
ads$campaign_id <- as.factor(ads$campaign_id)
ads$webpage_id <- as.factor(ads$webpage_id)
ads$product_category_1 <- as.factor(ads$product_category_1)
ads$user_group_id <- as.factor(ads$user_group_id)
ads$gender <- as.factor(ads$gender)
ads$age_level <- as.factor(ads$age_level)
ads$user_depth <- as.factor(ads$user_depth)
ads$city_development_index <- as.factor(ads$city_development_index)
ads$var_1 <- as.factor(ads$var_1)
ads$is_click <- as.factor(ads$is_click)

str(ads)

productA <- subset(ads, product == "A")
productB <- subset(ads, product == "B")
productC <- subset(ads, product == "C")
productD <- subset(ads, product == "D")
productE <- subset(ads, product == "E")
productF <- subset(ads, product == "F")
productG <- subset(ads, product == "G")
productH <- subset(ads, product == "H")
productI <- subset(ads, product == "I")
productJ <- subset(ads, product == "J")
```