Project1

Maria Garcia

2024-06-07

Introduction

We will be using a data set obtained from Kaggle that contains customer data from Waze.If you don't already know, Waze is an app that rivals Google Maps and Apple Maps. Their speciality is crowd sourcing information on speed traps, construction and other events drivers often want to know on the road.

This data set contains a column *"label"* which identifies the customer as retained or churned. In this exploratory analysis, I will attempt to identify which factors can aid in predicting customer churn. We will attempt this analysis via visual and statistical methods.

Importing the Data

- We will import the CSV file using read_csv function.
- Using the str() function allows us to preview each columns type and number of rows.
- Using head() allows us to see the first six rows of the file.

There are 13 total variables in this spreadsheet. In my opinion, the labels are a tad bit confusing, and so I've pasted their definitions here, directly from Kaggle.com:

Column Summary

1. **ID**: Unique identifier for each user.
2. **label**: Label indicating user churn status (e.g., churned, retained).
3. **sessions**: Number of sessions logged by the user.
4. **drives**: Number of drives completed by the user.
5. **total_sessions**: Total number of sessions recorded for the user.
6. **n_days_after_onboarding**: Number of days since user onboarding.
7. **total_navigations_fav1**: Total number of navigations using favorite route 1.
8. **total_navigations_fav2**: Total number of navigations using favorite route 2.
9. **driven_km_drives**: Total distance driven by the user in kilometers.
10. **duration_minutes_drives**: Total duration of drives in minutes.
11. **activity_days**: Number of days with user activity recorded.
12. **driving_days**: Number of days with driving activity recorded.
13. **device**: used by the user for navigation

```
waze_data <- read_csv("/Users/mariagarcia/Desktop/DAT301/waze_app_dataset.csv")
```

```
## Rows: 14999 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): label, device
## dbl (11): ID, sessions, drives, total_sessions, n_days_after_onboarding, tot...
##
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(waze_data)
```

```
## spc_tbl_ [14,999 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ID                      : num [1:14999] 0 1 2 3 4 5 6 7 8 9 ...
##  $ label                   : chr [1:14999] "retained" "retained" "retained" "retained" ...
##  $ sessions                : num [1:14999] 283 133 114 49 84 113 3 39 57 84 ...
##  $ drives                  : num [1:14999] 226 107 95 40 68 103 2 35 46 68 ...
##  $ total_sessions          : num [1:14999] 296.7 326.9 135.5 67.6 168.2 ...
##  $ n_days_after_onboarding: num [1:14999] 2276 1225 2651 15 1562 ...
##  $ total_navigations_fav1 : num [1:14999] 208 19 0 322 166 0 185 0 0 72 ...
##  $ total_navigations_fav2 : num [1:14999] 0 64 0 7 5 0 18 0 26 0 ...
##  $ driven_km_drives        : num [1:14999] 2629 13716 3059 914 3950 ...
##  $ duration_minutes_drives: num [1:14999] 1986 3160 1611 587 1220 ...
##  $ activity_days           : num [1:14999] 28 13 14 7 27 15 28 22 25 7 ...
##  $ driving_days            : num [1:14999] 19 11 8 3 18 11 23 20 20 3 ...
##  $ device                  : chr [1:14999] "Android" "iPhone" "Android" "iPhone" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ID = col_double(),
##   ..   label = col_character(),
##   ..   sessions = col_double(),
##   ..   drives = col_double(),
##   ..   total_sessions = col_double(),
##   ..   n_days_after_onboarding = col_double(),
##   ..   total_navigations_fav1 = col_double(),
##   ..   total_navigations_fav2 = col_double(),
##   ..   driven_km_drives = col_double(),
##   ..   duration_minutes_drives = col_double(),
##   ..   activity_days = col_double(),
##   ..   driving_days = col_double(),
##   ..   device = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(waze_data)
```

```
## # A tibble: 6 x 13
##      ID label    sessions drives total_sessions n_days_after_onboarding
##   <dbl> <chr>       <dbl>  <dbl>          <dbl>                   <dbl>
## 1     0 retained      283    226          297.                     2276
## 2     1 retained      133    107          327.                     1225
## 3     2 retained      114     95          136.                     2651
## 4     3 retained       49     40           67.6                      15
## 5     4 retained       84     68          168.                     1562
## 6     5 retained      113    103          280.                     2637
## # i 7 more variables: total_navigations_fav1 <dbl>,
## #   total_navigations_fav2 <dbl>, driven_km_drives <dbl>,
## #   duration_minutes_drives <dbl>, activity_days <dbl>, driving_days <dbl>,
## #   device <chr>
```

Commentary

- Right off, we can see some columns that may not be of use to us such as customer ID. We will take care of these in our preprocessing step.

- I question the usefulness of the column *n_days_after_onboarding*, as this data is static and there is no indication as to when this subset of customers signed up for the app. This could skew the data if not handled correctly.
- We will need to make the assumption that "activity days" indicates that they opened the application, and "driving days" indicates the days that the customer used the app for navigation, as there is nothing specify or indicate otherwise.

Preprocessing

We need to do some manipulation to the data to make it easier for us to understand, and remove the information that is not going to be useful to us.

First, I will rename the column "label" to "customer_status". Because there are so many variables, it will be important to have meaningful names.

Next, we we will remove the ID column, as we will not need it for our analysis. After some consideration, we will leave in *n_days_after_onboarding* for now.

Lastly, because I want to focus on if a customers status is *churned* or *retained*, I want to remove any row that has no value in that column. We will first check if there are any blanks, remove those rows, and perform the same check again.

```
colnames(waze_data)[colnames(waze_data) == "label"] <- "customer_status"
colnames(waze_data)
```

```
##  [1] "ID"                      "customer_status"
##  [3] "sessions"                "drives"
##  [5] "total_sessions"          "n_days_after_onboarding"
##  [7] "total_navigations_fav1"  "total_navigations_fav2"
##  [9] "driven_km_drives"        "duration_minutes_drives"
## [11] "activity_days"           "driving_days"
## [13] "device"
```

```
sapply(waze_data, function(x) sum(is.na(x)))
```

```
##                      ID         customer_status                sessions
##                       0                     700                       0
##                  drives          total_sessions n_days_after_onboarding
##                       0                       0                       0
##  total_navigations_fav1  total_navigations_fav2        driven_km_drives
##                       0                       0                       0
## duration_minutes_drives           activity_days            driving_days
##                       0                       0                       0
##                  device
##                       0
```

```
nrow(waze_data)
```

```
## [1] 14999
```

```
waze_data_cleaned <- waze_data[!is.na(waze_data$customer_status), ]
sapply(waze_data_cleaned, function(x) sum(is.na(x)))
```

```
##                      ID         customer_status                sessions
##                       0                       0                       0
##                  drives          total_sessions n_days_after_onboarding
##                       0                       0                       0
```

```
##  total_navigations_fav1  total_navigations_fav2        driven_km_drives
##                       0                       0                       0
## duration_minutes_drives           activity_days            driving_days
##                       0                       0                       0
##                  device
##                       0
```

```
nrow(waze_data_cleaned)
```

```
## [1] 14299
```

```
#write.csv(waze_data_cleaned, "waze_data_cleaned.csv", row.names = FALSE)

waze_data_cleaned <- waze_data_cleaned %>% select(-ID)
colnames(waze_data_cleaned)
```

```
##  [1] "customer_status"         "sessions"
##  [3] "drives"                  "total_sessions"
##  [5] "n_days_after_onboarding" "total_navigations_fav1"
##  [7] "total_navigations_fav2"  "driven_km_drives"
##  [9] "duration_minutes_drives" "activity_days"
## [11] "driving_days"            "device"
```

Review Retained vs Churned Customers

Looking at our clean data set, I want to check the distribution of *churned* vs *retained* customers. This data set will not be useful to us if we have very little churned customers to base our analysis on!

```
churn_data <- waze_data_cleaned[!is.na(waze_data_cleaned$customer_status), ]
table(waze_data_cleaned$customer_status)
```

```
##
##  churned retained
##     2536    11763
```

```
prop.table(table(waze_data_cleaned$customer_status))
```

```
##
##   churned  retained
## 0.1773551 0.8226449
```

Commentary

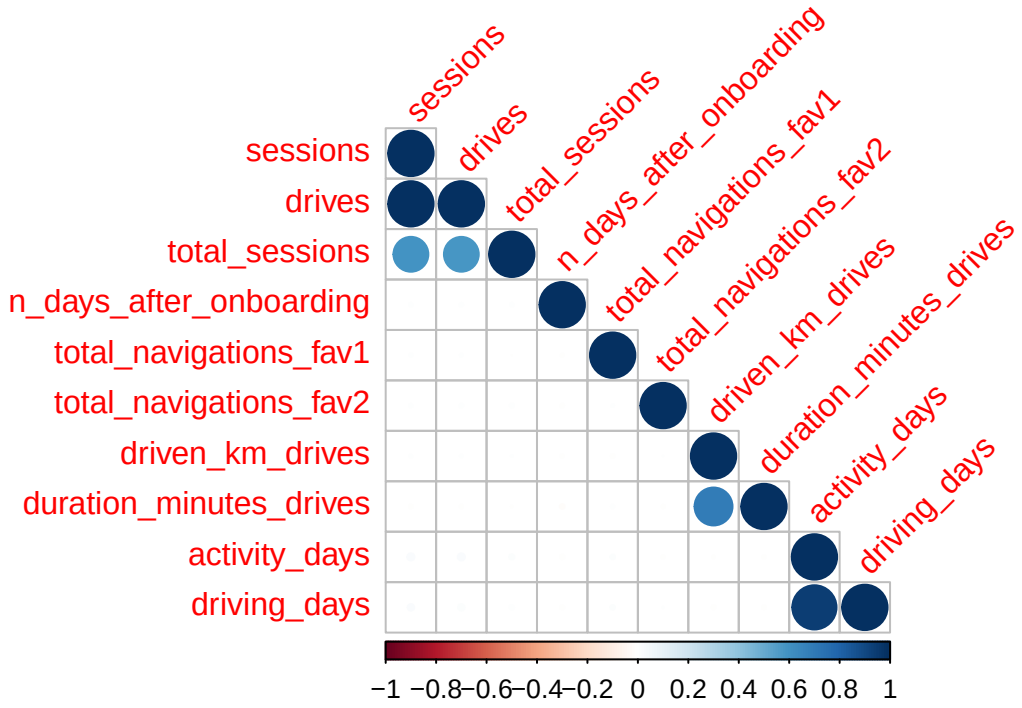
- The distribution of customer statuses is skewed, with 82.3% retained and 17.7% churned. There are ~2,500 churned customers, and ~12,000 retained customers. This will work for our analysis!

Correlation Between Variables

We now know there is a significant size difference in the population of *churned* vs. *retained*.

Of the eleven other variables in this data set, are there are close correlations?

```
churn_data$customer_status <- as.factor(churn_data$customer_status)
numeric_cols <- unlist(lapply(churn_data, is.numeric))
churn_numeric <- churn_data[, numeric_cols]
cor_matrix <- cor(churn_numeric)
corrplot(cor_matrix, method = "circle", type = "lower", tl.coll = "black", tl.srt = 45)
```



Commentrary

- It seems we have some correlations:
  - As to be expected, *total sessions* has a correlation with *total sessions* and *drives*
  - Also to be expected the duration a customer has driven (in minutes) has a correlation to the number of KM driven
- Stronger correlations include: +Number of drives and number of sessions
  - Driving days and activity days.

This does not tell us much about how these variables are related beyond what we can already assume.

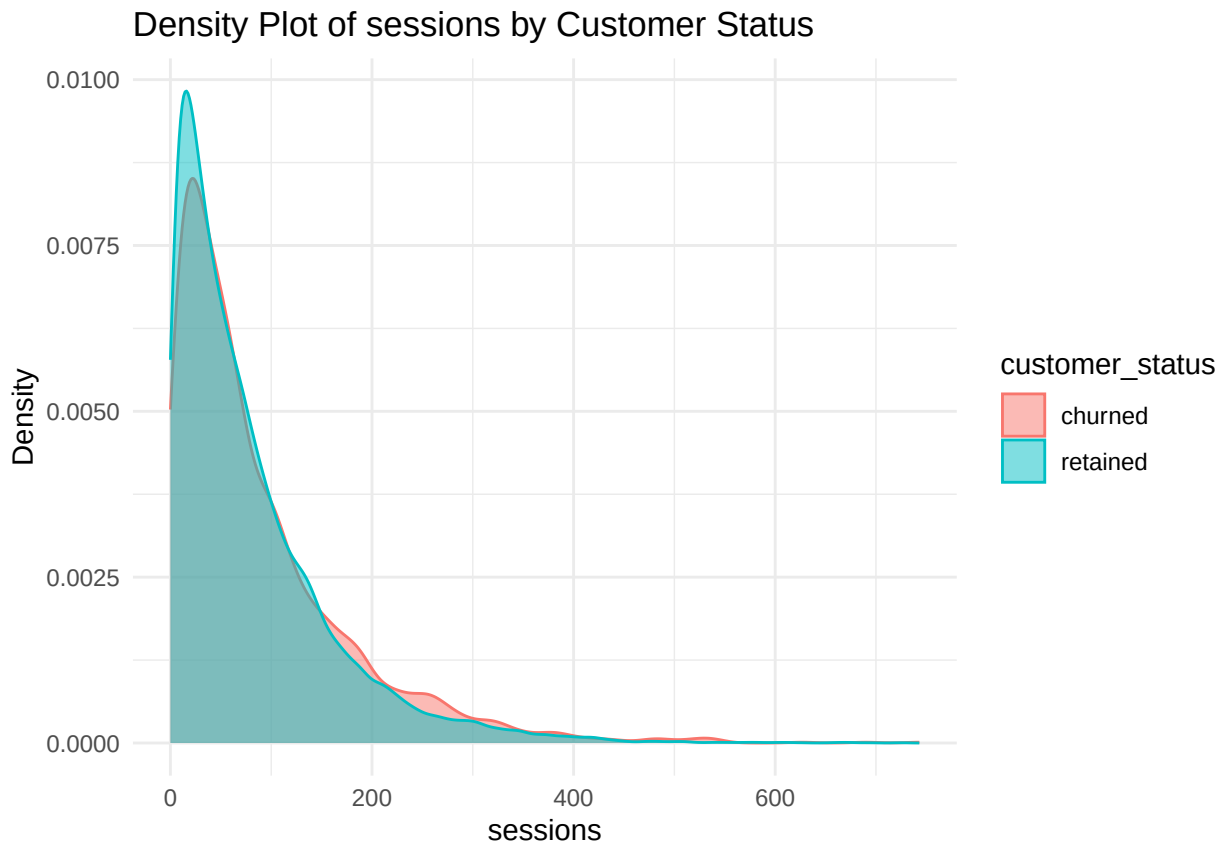
Visualisizing Correlations Using Density Plots

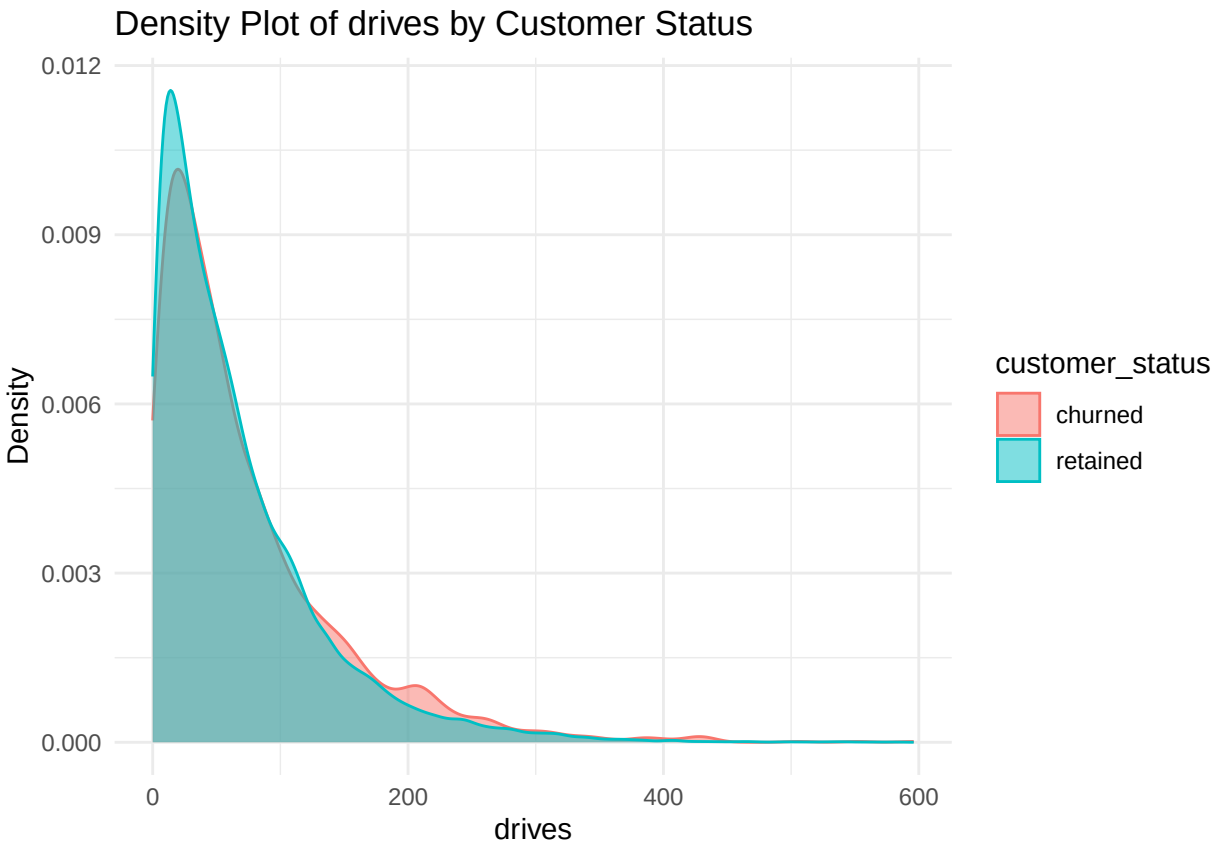
Next, we will try to visualize each variable using density plots. The data will be separated by *churned* and *retained* customers. My goal is to visualize the distribution of various features across churned and retained customers. These plots help in understanding the differences in feature distributions between the two groups.

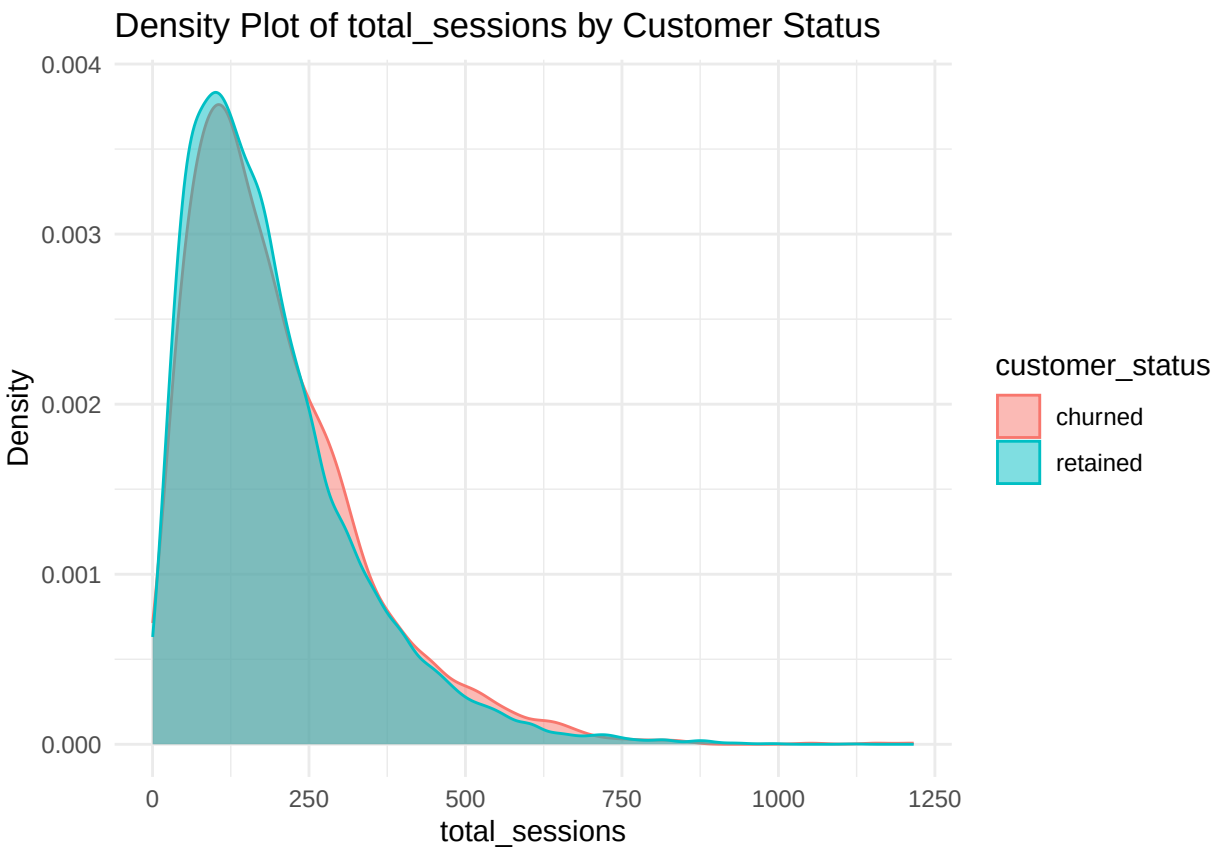
Because I am not sure which variables are important yet, I've created plots for every variable.

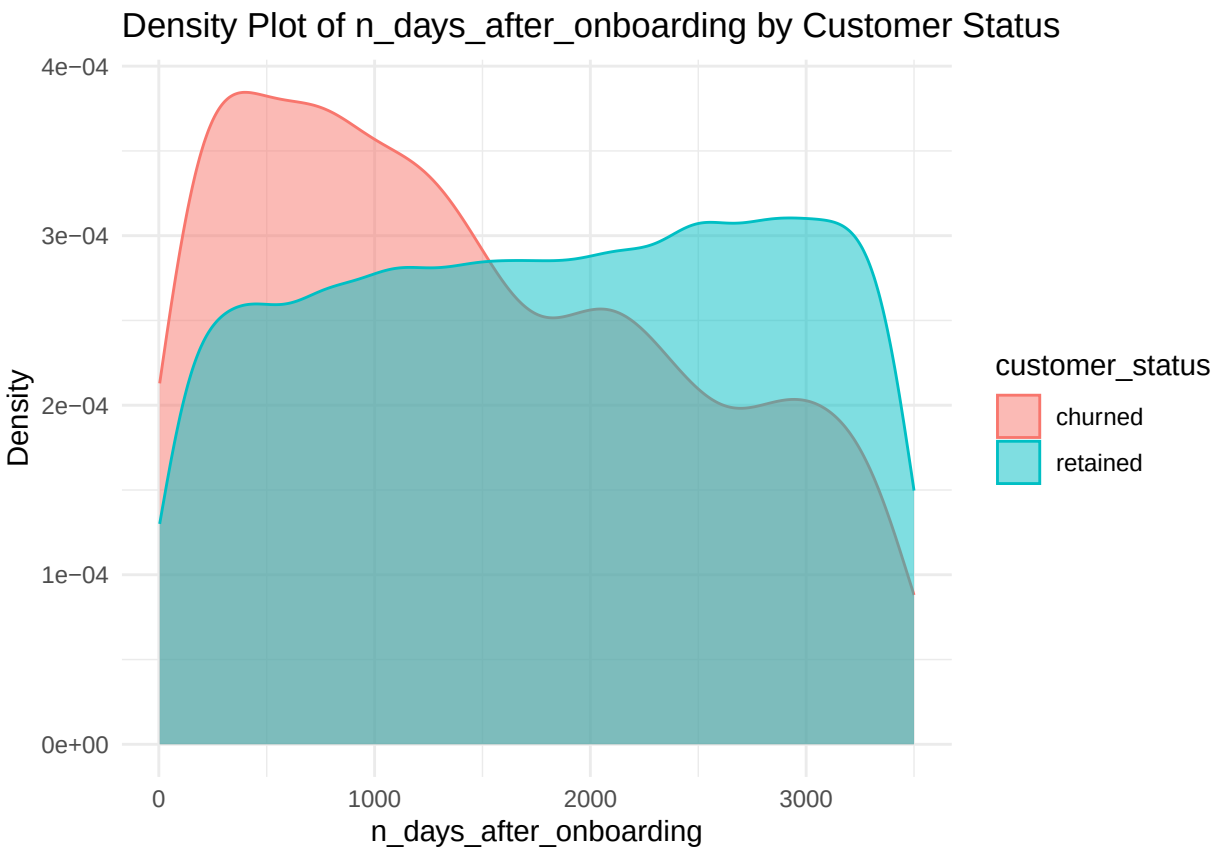
```
numerical_features <- waze_data_cleaned %>% select_if(is.numeric) %>% names()
for (feature in numerical_features) {
  p <- ggplot(waze_data_cleaned, aes(x = !!sym(feature), color = customer_status, fill = customer_status
    geom_density(alpha = 0.5) +
    labs(title = paste("Density Plot of", feature, "by Customer Status"),
         x = feature,
         y = "Density") +
    theme_minimal()
```

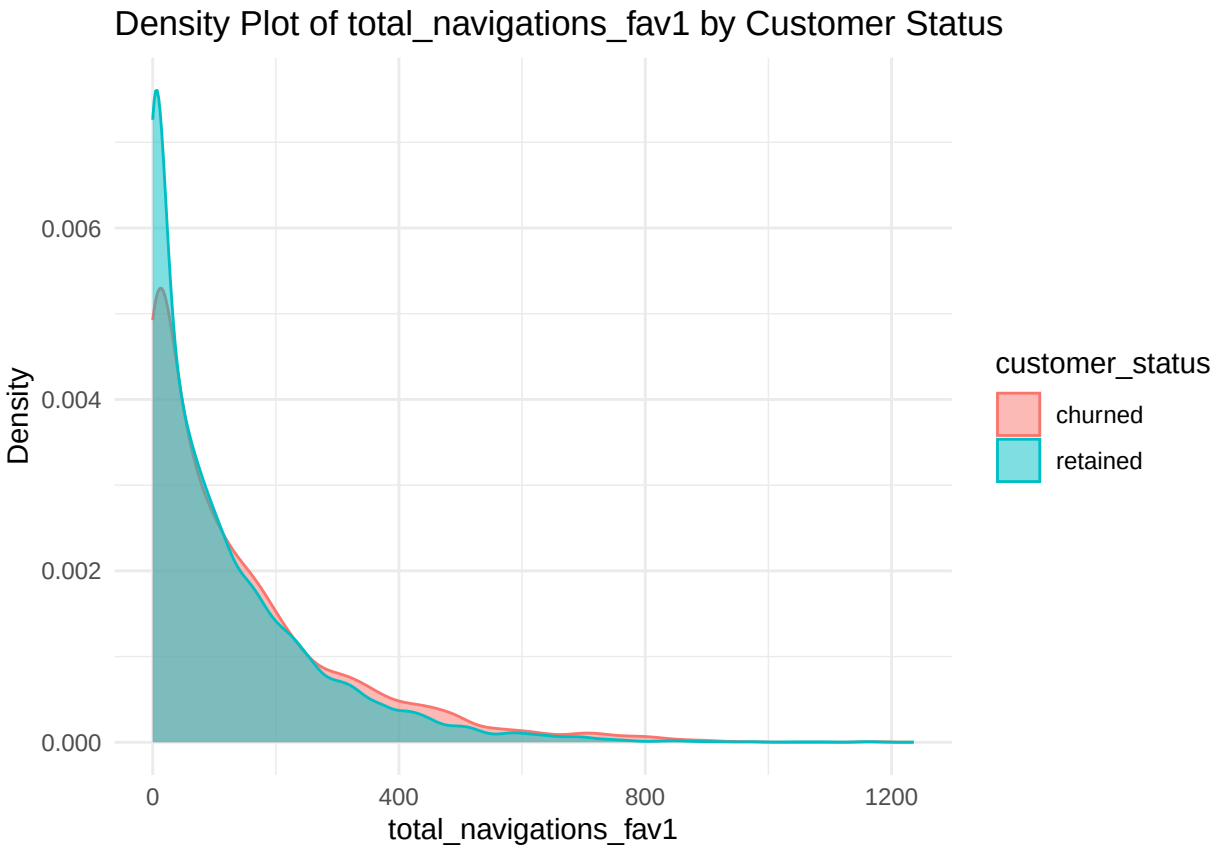
```
  print(p)
}
```

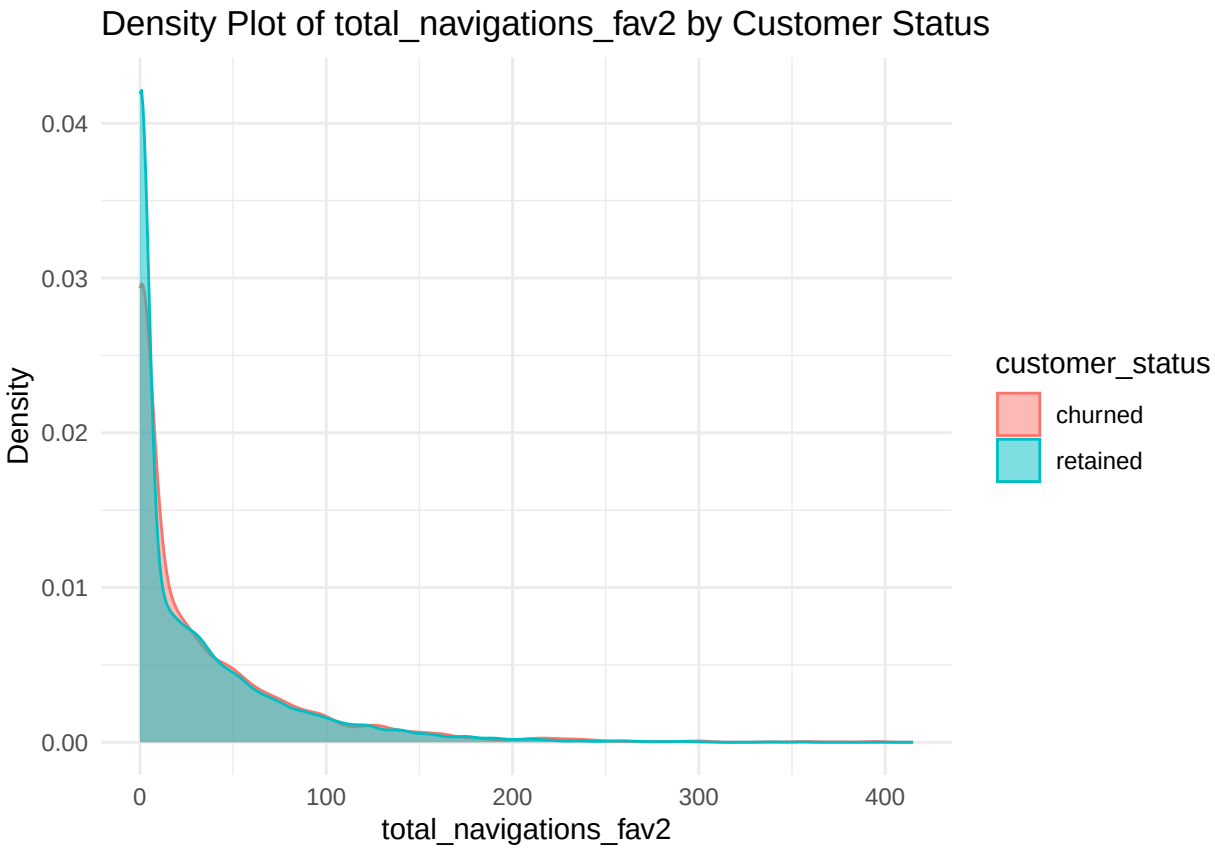

Density Plot of sessions by Customer Status
0.0100
0.0075
0.0050
0.0025
0.0000
Density
0
200
400
600
sessions
customer_status
churned
retained

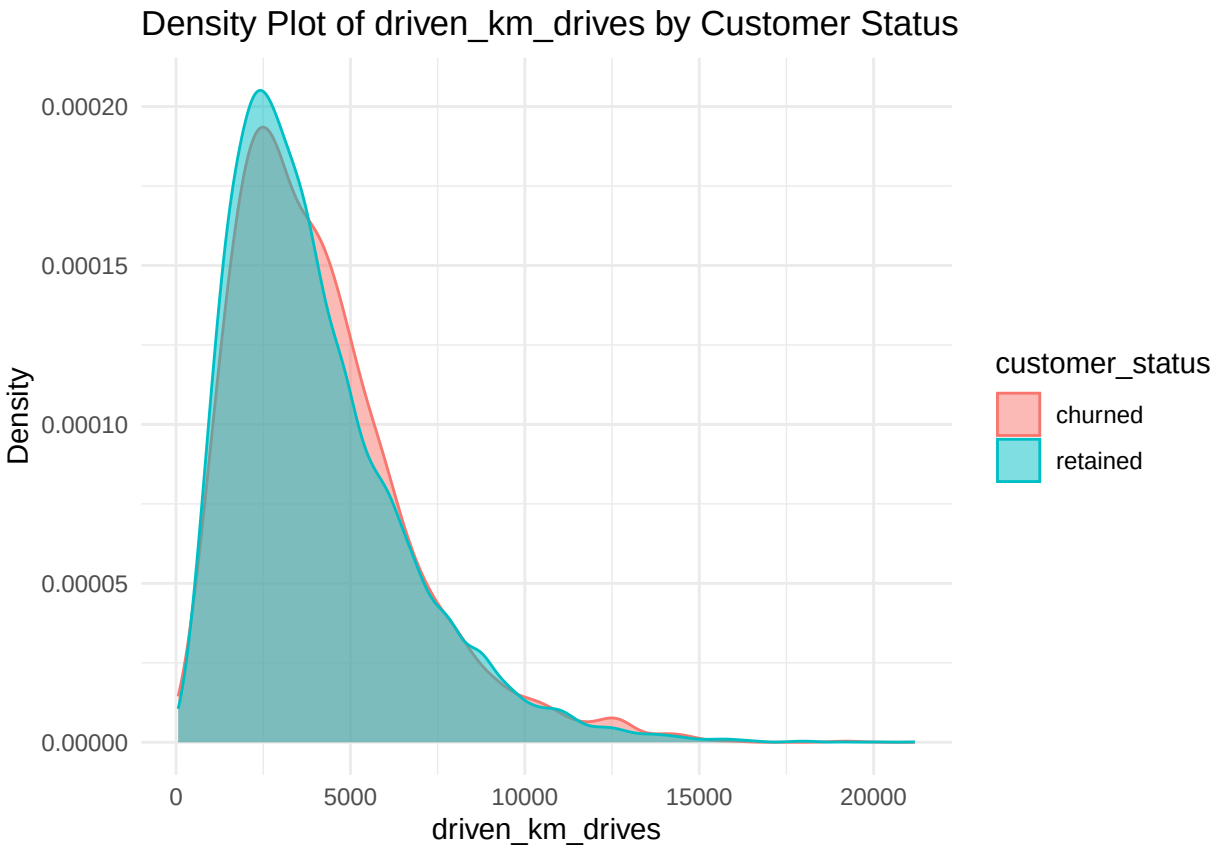

Density Plot of drives by Customer Status
0.012
0.009
0.006
0.003
0.000
Density
0
200
400
600
drives
customer_status
churned
retained

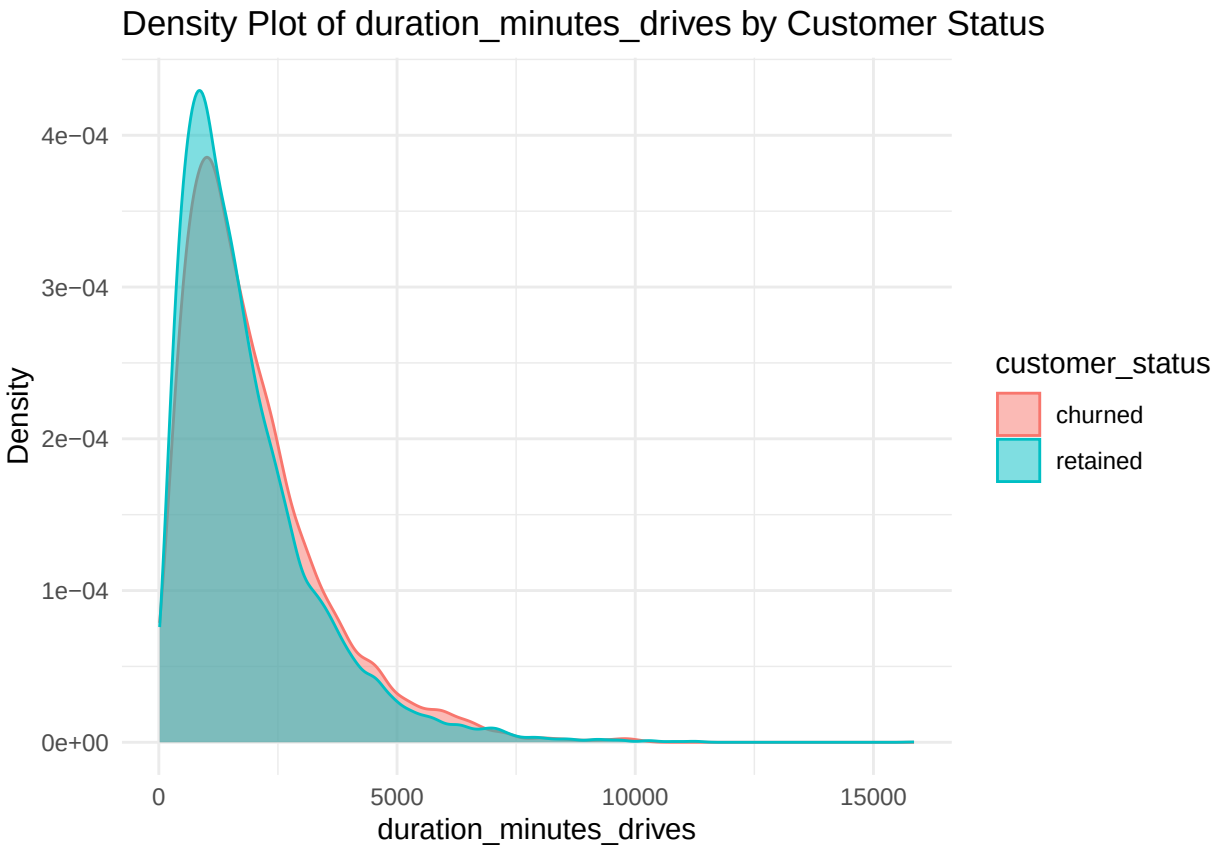

Density Plot of total_sessions by Customer Status
0.004
0.003
0.002
0.001
0.000
Density
0
250
500
750
1000
1250
total_sessions
customer_status
churned
retained

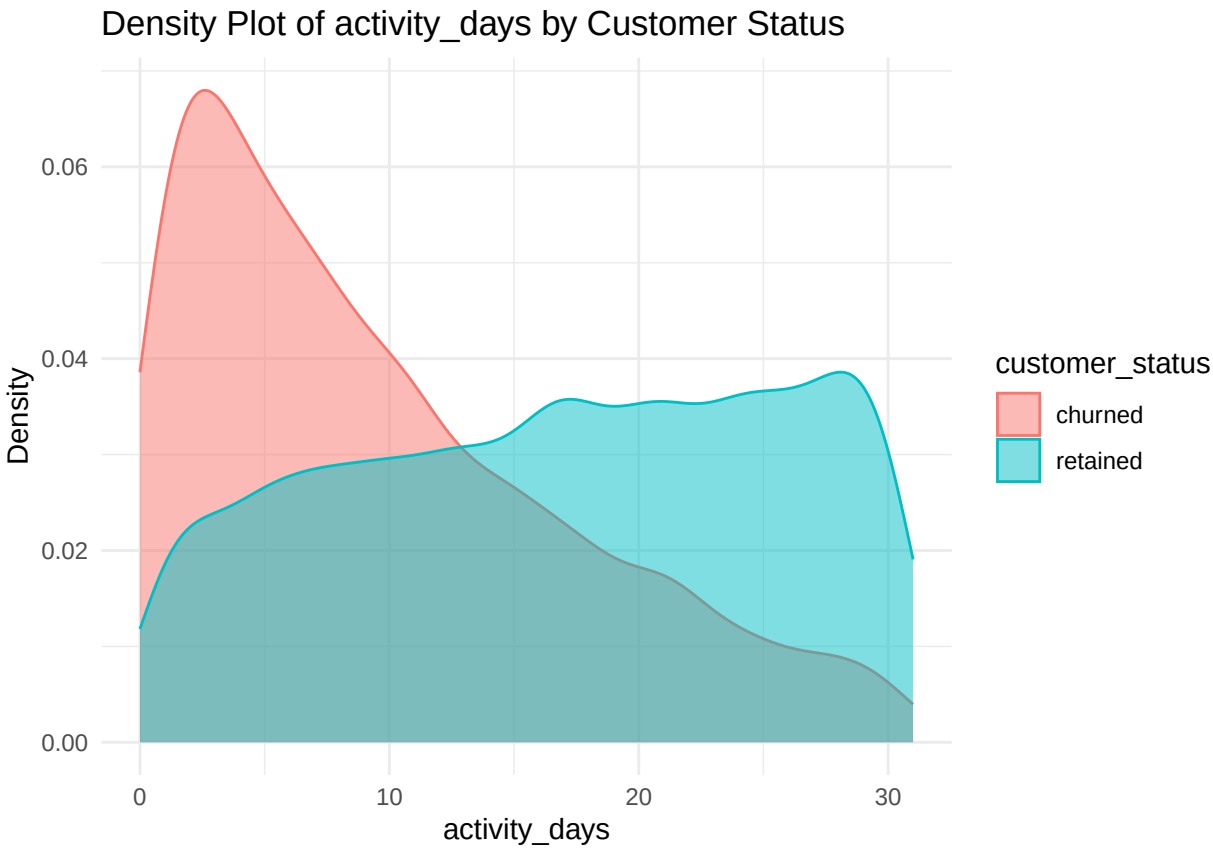

Density Plot of n_days_after_onboarding by Customer Status
Density
4e−04
3e−04
2e−04
1e−04
0e+00
0
1000
2000
3000
n_days_after_onboarding
customer_status
churned
retained

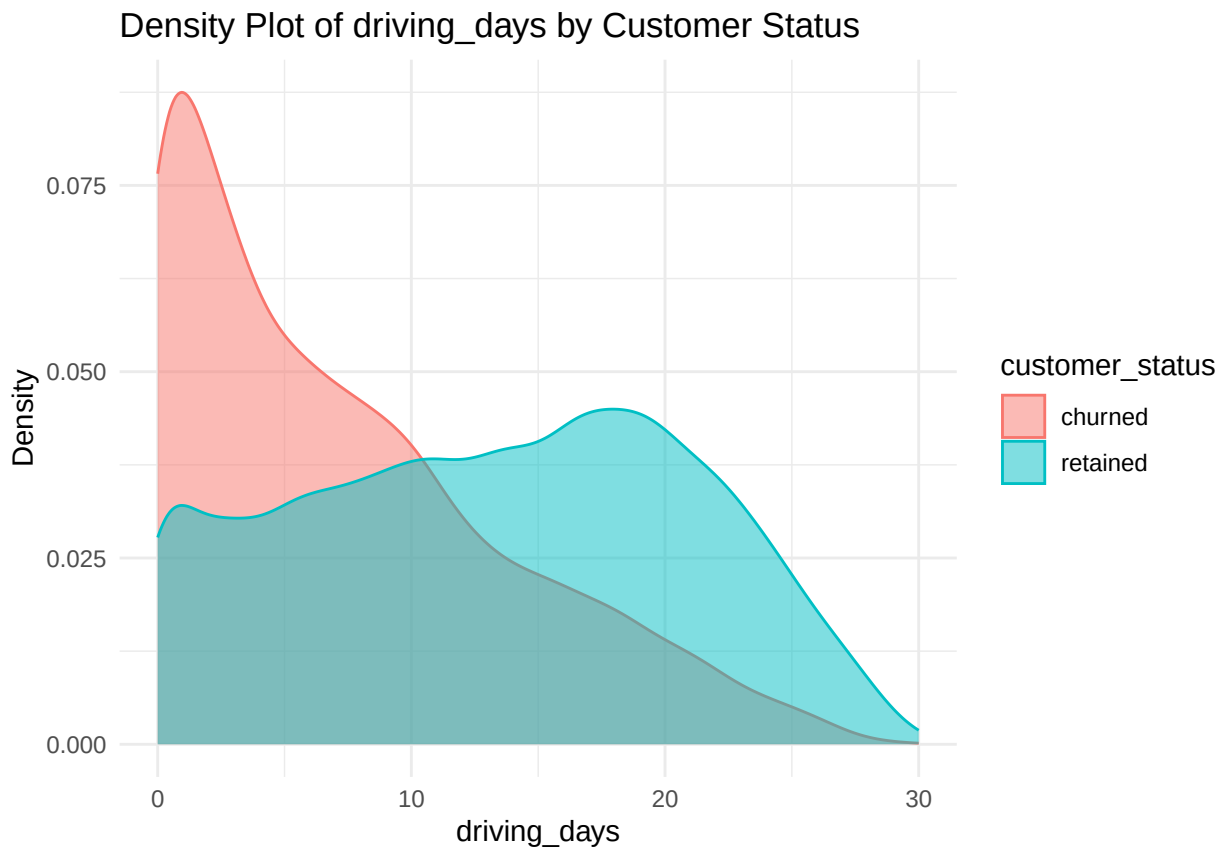

Density Plot of total_navigations_fav1 by Customer Status
Density
0.006
0.004
0.002
0.000
0
400
800
1200
total_navigations_fav1
customer_status
churned
retained


Density Plot of total_navigations_fav2 by Customer Status
Density
0.04
0.03
0.02
0.01
0.00
0
100
200
300
400
total_navigations_fav2
customer_status
churned
retained


Density Plot of driven_km_drives by Customer Status
Density
0.00020
0.00015
0.00010
0.00005
0.00000
0
5000
10000
15000
20000
driven_km_drives
customer_status
churned
retained


Density Plot of duration_minutes_drives by Customer Status
Density
4e−04
3e−04
2e−04
1e−04
0e+00
0
5000
10000
15000
duration_minutes_drives
customer_status
churned
retained


Density Plot of activity_days by Customer Status
Density
0.06
0.04
0.02
0.00
0
10
20
30
activity_days
customer_status
churned
retained



Commentrary on Density Plots

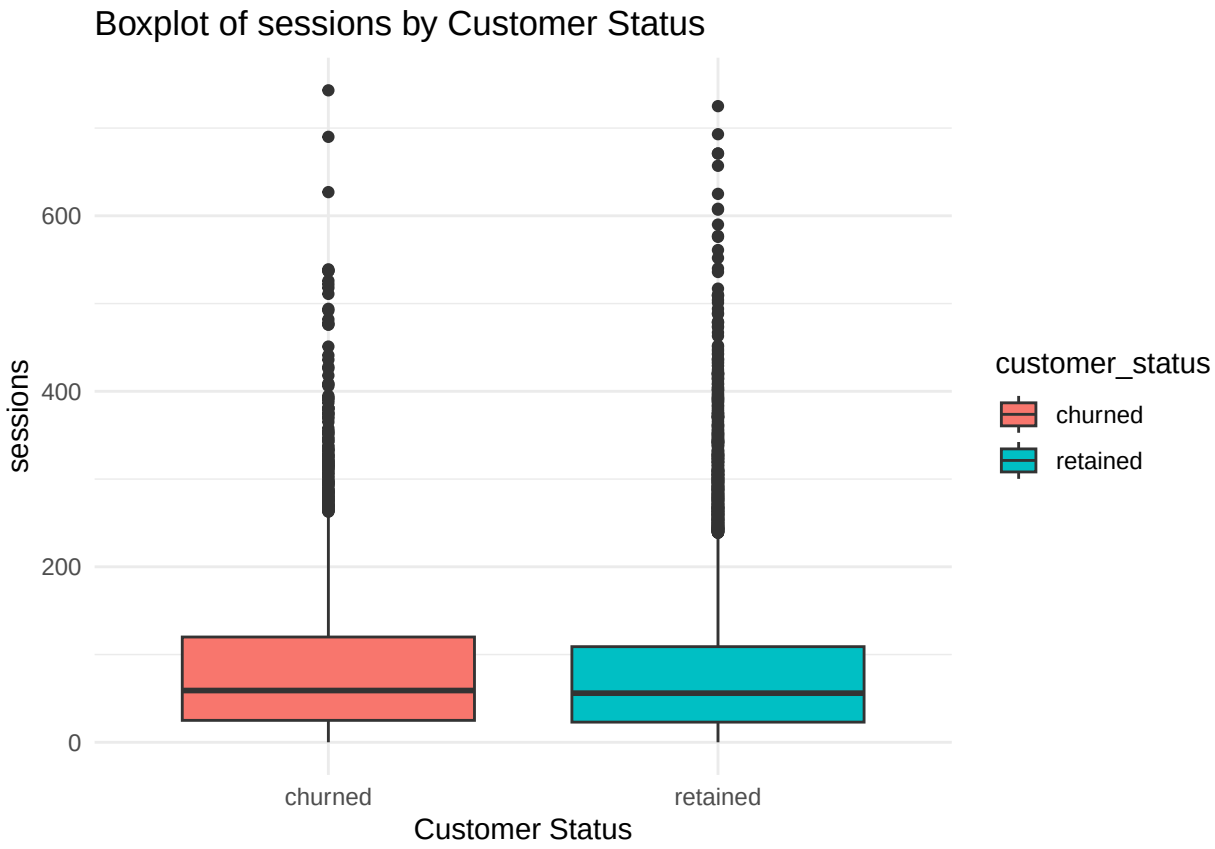
- We can gather a few insights from these plots:
  - Customers in the retained group have used the app for longer, where as customers in the churned group appear to be newer.
  - A greater density of customers who left the app used it for less than 10 days
  - Similar to the activity days, customers who left the app used it for driving for less than 10 days.

The above insights are surprising, given that the plots also reveal similar density in sessions, KM driven and favorite routes. There is more to uncover here!

Visualizing Using Boxplots

I generated boxplots for every variable to compare the distributions of numerical features between churned and retained customers. This visualization provides a clearer comparison of central tendencies and variances.

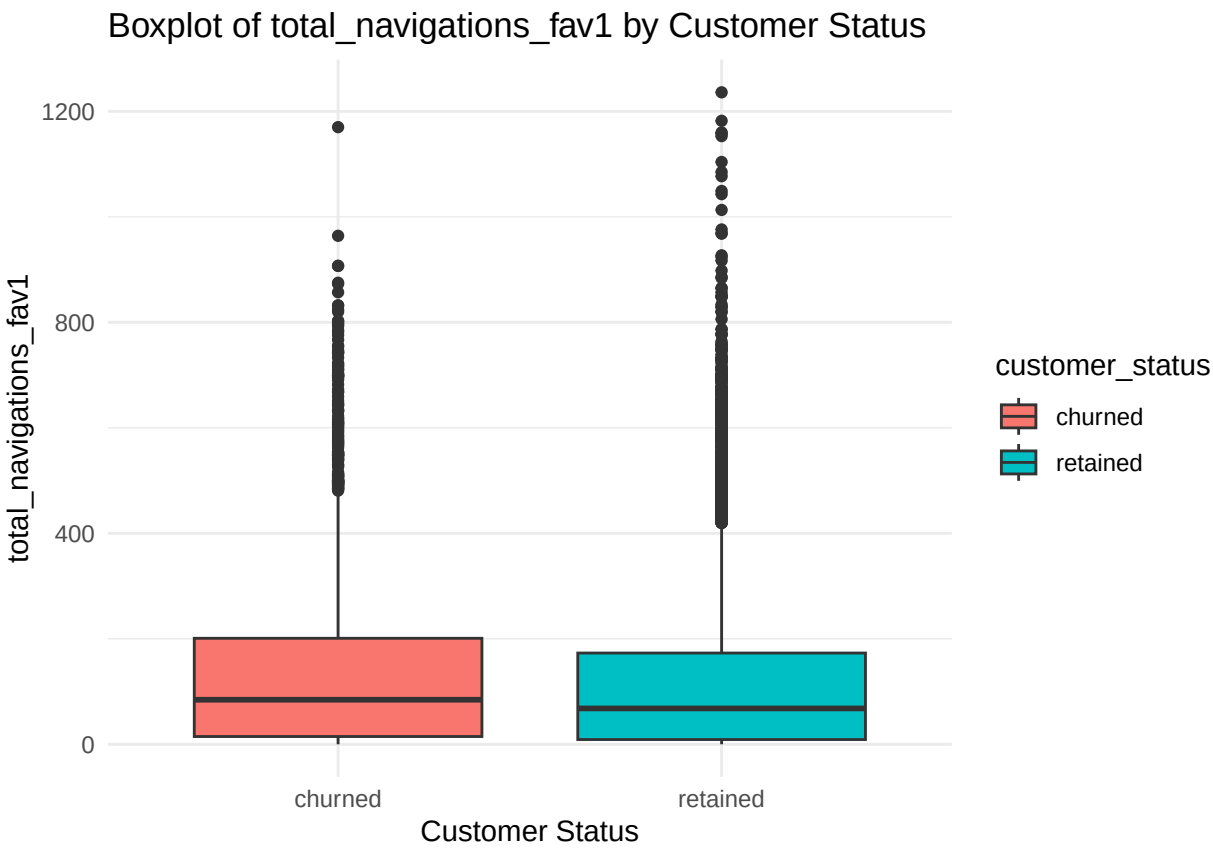
```
numerical_features <- waze_data_cleaned %>% select_if(is.numeric) %>% names()
for (feature in numerical_features) {
  p <- ggplot(waze_data_cleaned, aes(x = customer_status, y = !!sym(feature), fill = customer_status)) +
    geom_boxplot() +
    labs(title = paste("Boxplot of", feature, "by Customer Status"),
         x = "Customer Status",
         y = feature) +
    theme_minimal()
  print(p)
}
```


Boxplot of sessions by Customer Status
sessions
600
400
200
0
churned
retained
Customer Status
customer_status
churned
retained


Boxplot of drives by Customer Status
600
400
200
0
drives
churned
retained
Customer Status
customer_status
churned
retained


Boxplot of total_sessions by Customer Status
1250
1000
750
500
250
0
total_sessions
churned
retained
Customer Status
customer_status
churned
retained


Boxplot of n_days_after_onboarding by Customer Status
n_days_after_onboarding
3000
2000
1000
0
churned
retained
Customer Status
customer_status
churned
retained

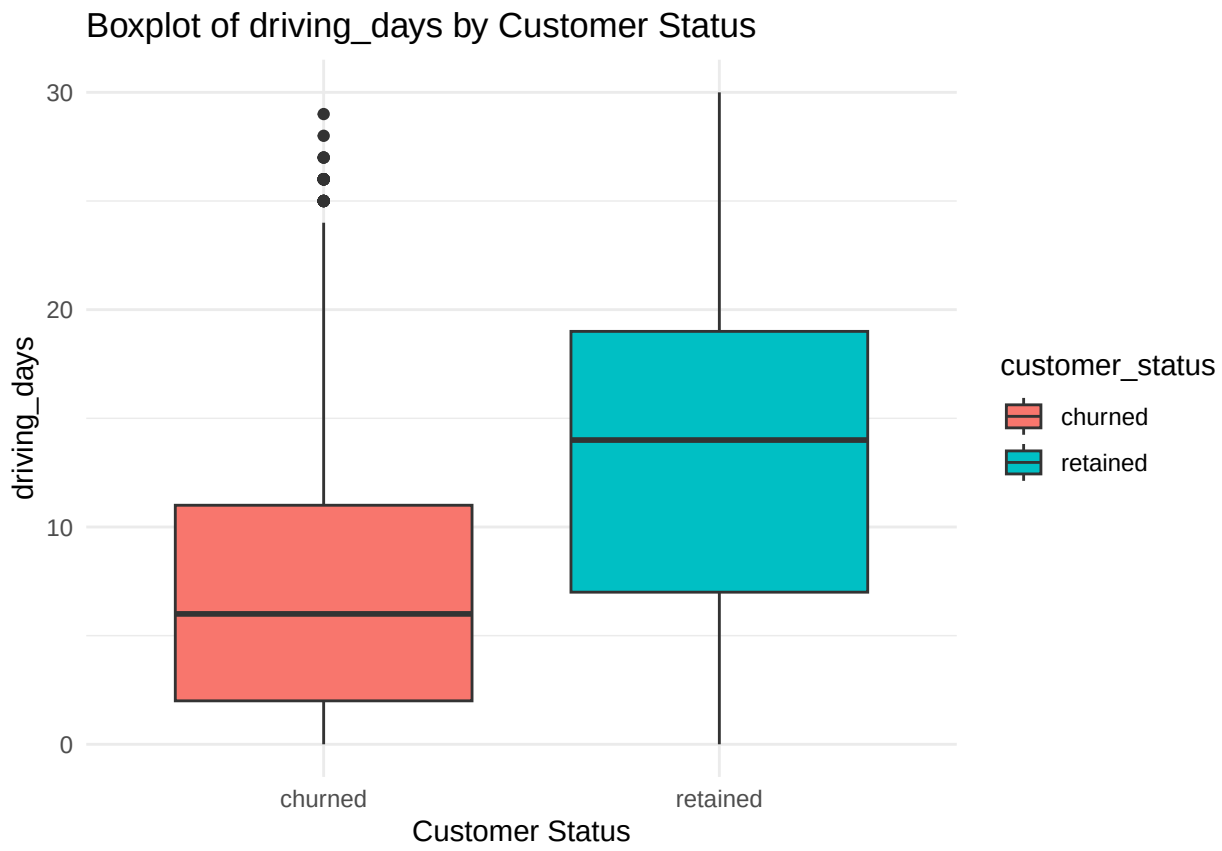

Boxplot of total_navigations_fav1 by Customer Status
total_navigations_fav1
1200
800
400
0
churned
retained
Customer Status
customer_status
churned
retained


Boxplot of total_navigations_fav2 by Customer Status
total_navigations_fav2
400
300
200
100
0
churned
retained
Customer Status
customer_status
churned
retained


Boxplot of driven_km_drives by Customer Status
driven_km_drives
20000
15000
10000
5000
0
churned
retained
Customer Status
customer_status
churned
retained


Boxplot of duration_minutes_drives by Customer Status
duration_minutes_drives
15000
10000
5000
0
churned
retained
Customer Status
customer_status
churned
retained


Boxplot of activity_days by Customer Status
activity_days
30
20
10
0
churned
retained
Customer Status
customer_status
churned
retained



Commentary on boxplots

- From these plots, we can gather:
  - there are a significant amount of outliers for number of sessions, number of drives, minutes duration, KM driven, and favorite routes driven (1 and 2), in both groups. This explains why our density plots revealed so little.
  - Retained customers have a greater mean for driving days, activity days and are active for longer (n_days_after_onboarding), although there are some outliers with churned customers.

Statistic Modeling

I think I've gotten most of what I can from visuals. Let's do some math!

I built a logistic regression model to see if this leads me to identify significant predictors of customer churn. The model includes variables such as sessions, drives, total sessions, n_days_after_onboarding, total_navigations_fav1, total_navigations_fav2, driven_km_drives, duration_minutes_drives, activity_days, driving_days, and device.

```
waze_data_cleaned$customer_status <- as.factor(waze_data_cleaned$customer_status)
model <- glm(customer_status ~ sessions + drives + total_sessions +
             n_days_after_onboarding + total_navigations_fav1 +
             total_navigations_fav2 + driven_km_drives +
             duration_minutes_drives + activity_days +
             driving_days + device,
             data = waze_data_cleaned, family = binomial)
```

```r
summary(model)
```

```
##
## Call:
## glm(formula = customer_status ~ sessions + drives + total_sessions +
##     n_days_after_onboarding + total_navigations_fav1 + total_navigations_fav2 +
##     driven_km_drives + duration_minutes_drives + activity_days +
##     driving_days + device, family = binomial, data = waze_data_cleaned)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.927e-02  8.562e-02  -0.225 0.821924
## sessions                 1.350e-03  3.484e-03   0.388 0.698296
## drives                  -3.440e-03  4.259e-03  -0.808 0.419248
## total_sessions          -1.317e-04  2.114e-04  -0.623 0.533457
## n_days_after_onboarding  3.891e-04  2.384e-05  16.320  < 2e-16 ***
## total_navigations_fav1  -1.099e-03  1.489e-04  -7.377 1.62e-13 ***
## total_navigations_fav2  -1.137e-03  5.012e-04  -2.268 0.023309 *
## driven_km_drives         1.471e-05  1.331e-05   1.105 0.269093
## duration_minutes_drives -8.335e-05  2.239e-05  -3.723 0.000197 ***
## activity_days            8.095e-02  9.012e-03   8.983  < 2e-16 ***
## driving_days             2.774e-02  1.043e-02   2.659 0.007827 **
## deviceiPhone             7.059e-03  4.915e-02   0.144 0.885804
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13366  on 14298  degrees of freedom
## Residual deviance: 11571  on 14287  degrees of freedom
## AIC: 11595
##
## Number of Fisher Scoring iterations: 5
```

Explanation of Model: Phew, that's a lot of numbers. Here's some background on what these numbers mean:

This model estimates the probability that a given input point belongs to a particular class. Each variable is treated like a predictor, and the model calculates four numbers for each:

- Estimate : This is esentially the change in the log-odds of the churn for a one-unit change in the predictor. You may have heard this being called the coefficient for the predictor in stats.
- Std. Error: This measures the variability in the estimate
- z value: The estimate divided by the standard error. It's a test statistic, for now we won't worry about this value.
- P-value : A smaller p-value suggests that the null hypothesis is less likely.

So looking at a few of the first variables...

- **n_days_after_onboarding**: -The longer a customer has been onboarded, the more likely they are to churn. -The p-value is less than 0.001 (***), indicating high statistical significance
- The positive coefficient (0.0003891) suggests that as the number of days since onboarding increases, the log-odds of churn also increase.

- **total_navigations_fav1:**
- Higher navigations to the first favorite location decrease churn likelihood.
- The p-value is less than 0.001 (***), indicating high statistical significance.
- The negative coefficient (-0.001099) suggests that as the number of navigations to the first favorite location increases, the log-odds of churn decrease.

I will skip the lengthy explanations for every variable and move to a summary, as these are arduous and surely (understandably) will not be read

Summary

1. n_days_after_onboarding: Longer onboarding duration increases the likelihood of churn.
2. total_navigations_fav1: More navigations to the first favorite location decrease churn likelihood.
3. total_navigations_fav2: More navigations to the second favorite location decrease churn likelihood.
4. duration_minutes_drives: Longer drive durations reduce churn likelihood.
5. activity_days: More activity days increase retention likelihood.
6. driving_days: More driving days reduce churn likelihood.

Conclusion:

There are six variables that will aid in predicting customer churn. These six variables are based on a logistic regression model. These findings suggest that increased engagement with the app (as indicated by navigations to favorite locations, drive durations, activity days, and driving days) reduces the likelihood of churn. On the other hand, customers who have been onboarded longer are more likely to churn, indicating that maintaining engagement over time is crucial.

Refrences:

- Dataset: https://www.kaggle.com/datasets/raminhuseyn/wase-navigation-app-dataset/data
- Math: https://www.youtube.com/watch?v=C4N3_XJJ-jU
- Correlation Heat Map: https://www.geeksforgeeks.org/how-to-create-correlation-heatmap-in-r/