Credit Sesame Consumer Report

Audreya Metz, Calvin Ma, Hari Rajan, Paul Giroud

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:magrittr':
##
##     extract
```

1 Introduction

Determining the appropriate consumer to give loans to is a difficult task that can depend on many factors. Some of which directly impact what a company may think of you such as the amount past due on accounts, your open balance, etc. Other factors are more implicit, but could also determine whether or not you'd be a good candidate to give a loan to. We analyzed data from Credit Sesame, a company that calculates credit scores to determine options for credit cards, mortgage rates and loans. This dataset included three main parts: User Demographics, First Session Information and 30-day User Engagement Data. The user demographics data included mostly credit profile information including loan histories, tradeline details, etc. There were also a few personal features such as gender and location. The First Session data provided action logs of each user's first interaction with the Credit Sesame website. Finally, the 30-day user engagement gave insight into the actions of users in each one's first 30 days. This dataset had similar features to the first session, but was sparser and included various logins for each user.

We believe that identifying accurate credit scores with both financial and non-financial data can be of use to Credit Sesame. If Credit Sesame could approximate credit scores from certain characteristics, they could then provide targeted advertisements for specific credit cards or loans. In addition to targeted offerings and ads, using more statistically significant data when calculating credit score could lead more better predictions of user's credit, which could lead to Credit Sesame making better decision on who to give loans to based on user's credit score. Our ultimate goal was to see what cariables were strong indicators of credit score and delinquencies. We also found it relevant to explore the engagement habits of certain demographics groups with the Credit Sesame website. By analyzing how certain groups differ in website navigation, we could provide insight to Credit Sesame on their customer base. Credit Sesame could use this to develop internal strategies and how to better retain users based on their needs and profiles.

To accomplish these tasks, we used 4 tools: PCA, Linear Regression, Lasso Regression and K-means clustering. PCA, linear regression and lasso regression were used with the user demographics data to explore which features had a significant impact on the credit score, and by how much. The k-mean clustering was used with both the user demographics and 30-day engagement to cluster similar users based on their engagement and then compare the difference between users of differing clusters.

ADD IN SUMMARY OF RESULTS

2 Data set

2.1 Preprocessing

We did different forms of preprocessing for each of the different methods (from working separately on different parts).

```
##    Unnamed..0                                        user_id
##  Min.   :     0   0000485dbdc19e76bedcaf155b4de9491f414a27:     1
##  1st Qu.: 71373   00008fb52ca81c6767db31f58ea37e7229c372e1:     1
##  Median :142745   000091252c68230bd55a9abda98896c8610994bc:     1
##  Mean   :142745   000091d428154729cbf338d4f51aa95af254d342:     1
##  3rd Qu.:214118   0000b28d30dd9cf2424e83822b2877fec98b7de5:     1
##  Max.   :285490   0000b68d91d518befd9d44c3217fd69f4e2fa66c:     1
##                   (Other)                                   :285485
##          user_signup_timestamp     state           zipcode
##  2018-07-23 20:57:07:     5    TX     : 26785   30349  :   182
##  2018-07-01 21:11:33:     4    CA     : 24700   33311  :   175
##  2018-07-02 03:26:03:     4    FL     : 21955   60620  :   147
##  2018-07-02 19:19:32:     4    GA     : 14054   30906  :   145
##  2018-07-02 23:21:37:     4    NY     : 13113   60619  :   143
##  2018-07-03 16:17:21:     4    OH     : 11636   19124  :   141
##  (Other)            :285466    (Other):173248   (Other):284558
##  is_homeowner      gender       tradelines_avg_days_since_opened
##  False:212817         : 34772   Min.   :    3.0
##  True : 72674   Female:118914   1st Qu.:  545.5
##                 Male  : 81854   Median :  919.0
##                 Unisex: 49951   Mean   : 1164.7
##                                 3rd Qu.: 1479.6
##                                 Max.   :19777.0
##                                 NA's   :10593
##  tradelines_max_days_since_opened tradelines_min_days_since_opened
##  Min.   :    3                    Min.   :    2.0
##  1st Qu.:  964                    1st Qu.:   92.0
##  Median : 1750                    Median :  182.0
##  Mean   : 2495                    Mean   :  369.2
##  3rd Qu.: 3268                    3rd Qu.:  397.0
##  Max.   :25797                    Max.   :19777.0
##  NA's   :10593                    NA's   :10593
##  count_tradelines_closed_accounts count_total_tradelines_opened_24_months
##  Min.   :  0.000                  Min.   :  0
##  1st Qu.:  1.000                  1st Qu.:  1
##  Median :  4.000                  Median :  2
##  Mean   :  7.502                  Mean   :  3
##  3rd Qu.: 10.000                  3rd Qu.:  4
##  Max.   :187.000                  Max.   :100
##
##  count_tradelines_cc_opened_24_months
##  Min.   : 0.0000
##  1st Qu.: 0.0000
##  Median : 0.0000
##  Mean   : 0.5937
##  3rd Qu.: 1.0000
##  Max.   :20.0000
##
##  count_tradelines_condition_derogatory
##  Min.   :  0.000
##  1st Qu.:  1.000
##  Median :  3.000
##  Mean   :  4.299
##  3rd Qu.:  6.000
```

```
##  Max.   :282.000
##
##  count_open_installment_accounts_24_months
##  Min.   : 0.0000
##  1st Qu.: 0.0000
##  Median : 0.0000
##  Mean   : 0.6473
##  3rd Qu.: 1.0000
##  Max.   :20.0000
##
##  count_tradelines_open_collection_accounts count_tradelines_open_mortgages
##  Min.   :  0.000                           Min.   : 0.0000
##  1st Qu.:  0.000                           1st Qu.: 0.0000
##  Median :  1.000                           Median : 0.0000
##  Mean   :  2.348                           Mean   : 0.1361
##  3rd Qu.:  3.000                           3rd Qu.: 0.0000
##  Max.   :281.000                           Max.   :35.0000
##
##  count_tradelines_open_student_loans count_tradelines_opened_accounts
##  Min.   : 0.0000                     Min.   :  0.00
##  1st Qu.: 0.0000                     1st Qu.:  2.00
##  Median : 0.0000                     Median :  5.00
##  Mean   : 0.8865                     Mean   :  6.56
##  3rd Qu.: 0.0000                     3rd Qu.:  9.00
##  Max.   :44.0000                     Max.   :281.00
##
##  count_tradelines_open_secured_loans count_tradelines_open_unsecured_loans
##  Min.   : 0.00000                    Min.   : 0.0000
##  1st Qu.: 0.00000                    1st Qu.: 0.0000
##  Median : 0.00000                    Median : 0.0000
##  Mean   : 0.08293                    Mean   : 0.1293
##  3rd Qu.: 0.00000                    3rd Qu.: 0.0000
##  Max.   :26.00000                    Max.   :10.0000
##
##  total_tradelines_amount_past_due total_open_cc_amount_past_due
##  Min.   :     0                   Min.   :    0.000
##  1st Qu.:     0                   1st Qu.:    0.000
##  Median :     0                   Median :    0.000
##  Mean   :  2942                   Mean   :    3.766
##  3rd Qu.:  1269                   3rd Qu.:    0.000
##  Max.   :390690                   Max.   :16607.000
##
##  total_cc_open_balance total_tradelines_open_balance  max_cc_limit
##  Min.   :     0        Min.   :      0               Min.   :     0
##  1st Qu.:     0        1st Qu.:   1677               1st Qu.:     0
##  Median :     0        Median :   9841               Median :     0
##  Mean   :  2191        Mean   :  40200               Mean   :  2395
##  3rd Qu.:  1108        3rd Qu.:  35256               3rd Qu.:  1958
##  Max.   :198182        Max.   :8221813               Max.   :200000
##
##  max_cc_utilization_ratio avg_cc_utilization_ratio
##  Min.   :-1.00            Min.   : 0.00
##  1st Qu.: 0.40            1st Qu.: 0.19
##  Median : 0.88            Median : 0.61
```

```
##  Mean   : 0.71            Mean   : 0.56
##  3rd Qu.: 0.99            3rd Qu.: 0.92
##  Max.   :10.48            Max.   :10.48
##  NA's   :153646           NA's   :153656
##  total_mortgage_loans_amount total_mortgage_loans_balance
##  Min.   :      0             Min.   :      0
##  1st Qu.:      0             1st Qu.:      0
##  Median :      0             Median :      0
##  Mean   :  22458             Mean   :  19179
##  3rd Qu.:      0             3rd Qu.:      0
##  Max.   :7732500             Max.   :7555508
##
##  total_auto_loans_balance total_student_loans_balance
##  Min.   :      0          Min.   :     0
##  1st Qu.:      0          1st Qu.:     0
##  Median :      0          Median :     0
##  Mean   :   6516          Mean   :  7787
##  3rd Qu.:  10030          3rd Qu.:     0
##  Max.   :1777930          Max.   :827594
##
##  count_inquiries_3_months count_inquiries_6_months
##  Min.   : 0.0000          Min.   : 0.000
##  1st Qu.: 0.0000          1st Qu.: 0.000
##  Median : 0.0000          Median : 1.000
##  Mean   : 0.8155          Mean   : 1.414
##  3rd Qu.: 1.0000          3rd Qu.: 2.000
##  Max.   :53.0000          Max.   :97.000
##
##  count_inquiries_12_months         recent_bankruptcy_date
##  Min.   :  0.000                                 :259814
##  1st Qu.:  0.000           2011-10-31 00:00:00:    39
##  Median :  1.000           2017-03-31 00:00:00:    37
##  Mean   :  2.581           2018-01-31 00:00:00:    36
##  3rd Qu.:  3.000           2018-02-28 00:00:00:    36
##  Max.   :137.000           2015-04-30 00:00:00:    33
##                            (Other)            : 25496
##  count_bankruptcy          age_bucket        credit_score_bucket
##  Min.   : 0.0000   (25.0, 30.0]:34237   (520.0, 525.0]: 19314
##  1st Qu.: 0.0000   (30.0, 35.0]:34168   (515.0, 520.0]: 14206
##  Median : 0.0000   (35.0, 40.0]:33043   (550.0, 555.0]: 12005
##  Mean   : 0.1016   (45.0, 50.0]:29663   (510.0, 515.0]: 11167
##  3rd Qu.: 0.0000   (40.0, 45.0]:29388   (505.0, 510.0]:  9074
##  Max.   :10.0000   (50.0, 55.0]:28316   (500.0, 505.0]:  8377
##                    (Other)     :96676   (Other)       :211348
```

- 2.1.1 PCA: To clean the user profile data and make it more manageable, we removed any rows with any NA's and removed users with gender marked as "unisex". We removed the unisex because there was an extremely large number of unisex users, which led us to believe that unisex represented the users who didn't want to choose which sex they were. This cut down on our sample size but we still had quite a lot of data. We also removed the user_signup_timesteamp, state and zipcode features since we decided they were unnecessary to the PCA. For other features, there seemed to be overlapping information that were redundant and colinear- Cases like avg_days vs. max_days were not both necessary since avg_days accounts for max_days, so we removed rows that represented the same information. Also, cases where information could be represented by total tradeline data, but also had information split

between banking, credit, etc. also presented the same information. While these could individually important, we only wanted general credit information for the PCA, so we removed any columns that represented a split of tradeline information (if tradeline info was unavailable, we kept the subset features of tradeline). Finally, in dealing with bucketed age and credit scores, we parsed the min and max of the buckets and had numerical values as the average of the min & max of the bucket. This means that we don't have to deal with numberous dummy variables.

- 2.1.2 Regression: The regression required minimal preprocessing as any variabes that we felt were not useful or colinear could simply be excluded from the regression model. The bulk of the preprocessing done for the regression was in creating dummy variables for the categorical that were to be included in the model and converting categorical bins for age and credit score to numerical variables.
- 2.1.3 Lasso: We took a similar process of preprocessing lasso - we removed the NAs and unisex users, and removed unnecessary features (user_signup_timestamp, state and zipcode). We also parsed the bucketed features like in PCA. However, we decided not to remove redundent, colinear values because lasso naturally does this during the regression.

```
##        X                                                session_id
##  Min.   :      0   000005c3141860cd88c666c83e0387d6203048fd:      1
##  1st Qu.: 294214   00003adaba342e9e7a198f90ae66cc1f6db4f1f3:      1
##  Median : 588429   00006515fc208e834240995702978301a5ad5a5d:      1
##  Mean   : 588429   0000721b3f57fa6b76c2b3a512f36f092f249279:      1
##  3rd Qu.: 882643   00008670c218c4c8796fd32cd57b2d7834256d21:      1
##  Max.   :1176857   000089658f18a2c3da7f3e53071f4834f65bbddd:      1
##                    (Other)                                   :1176852
##                                        user_id        session_length
##  124b2b7812e5cc9142a0f15a3a80bef2855738cc:    323   Min.   :     0.0
##  326eddcf7540902d008271c95a48589c956c03f0:    323   1st Qu.:    16.0
##  cd3ba2499cbad67a1d3203696fa4969acd90ee21:    301   Median :    95.0
##  0fd175df91a54f0c5e7e6143af19656b6718a594:    299   Mean   :   283.2
##  38a79c3548922781ecc8aaeb847290cb3512e16d:    292   3rd Qu.:   318.0
##  affbb2f0b8d03dfd56c4a40b0e74b159a11dffa3:    285   Max.   :929132.0
##  (Other)                                  :1175035
##    view_count      view_aoop_overview_count view_my_credit_count
##  Min.   :  0.000   Min.   : 0.000           Min.   : 0.0000
##  1st Qu.:  1.000   1st Qu.: 0.000           1st Qu.: 0.0000
##  Median :  3.000   Median : 1.000           Median : 0.0000
##  Mean   :  4.772   Mean   : 1.089           Mean   : 0.4457
##  3rd Qu.:  6.000   3rd Qu.: 2.000           3rd Qu.: 1.0000
##  Max.   :164.000   Max.   :88.000           Max.   :24.0000
##
##  view_credit_monitoring_count view_my_recommendations_count
##  Min.   : 0.00000             Min.   : 0.000
##  1st Qu.: 0.00000             1st Qu.: 0.000
##  Median : 0.00000             Median : 0.000
##  Mean   : 0.09018             Mean   : 0.362
##  3rd Qu.: 0.00000             3rd Qu.: 0.000
##  Max.   :36.00000             Max.   :32.000
##
##  view_my_borrowing_power_count view_cc_best_cards_count view_my_debt_count
##  Min.   : 0.0000               Min.   :0                Min.   : 0.0000
##  1st Qu.: 0.0000               1st Qu.:0                1st Qu.: 0.0000
##  Median : 0.0000               Median :0                Median : 0.0000
##  Mean   : 0.2678               Mean   :0                Mean   : 0.1483
##  3rd Qu.: 0.0000               3rd Qu.:0                3rd Qu.: 0.0000
```

```
##  Max.   :32.0000               Max.   :0                Max.   :52.0000
##
##  view_cc_my_borrowing_power_count view_cc_approval_odds_count
##  Min.   : 0.00000                 Min.   : 0.00000
##  1st Qu.: 0.00000                 1st Qu.: 0.00000
##  Median : 0.00000                 Median : 0.00000
##  Mean   : 0.03388                 Mean   : 0.03219
##  3rd Qu.: 0.00000                 3rd Qu.: 0.00000
##  Max.   :16.00000                 Max.   :32.00000
##
##  view_cc_details_count view_my_credit_report_count
##  Min.   : 0.00000      Min.   : 0.0000
##  1st Qu.: 0.00000      1st Qu.: 0.0000
##  Median : 0.00000      Median : 0.0000
##  Mean   : 0.08788      Mean   : 0.0592
##  3rd Qu.: 0.00000      3rd Qu.: 0.0000
##  Max.   :32.00000      Max.   :18.0000
##
##  view_increase_total_credit_limit_count view_cc_marketplace_count
##  Min.   : 0.00000                       Min.   : 0.00000
##  1st Qu.: 0.00000                       1st Qu.: 0.00000
##  Median : 0.00000                       Median : 0.00000
##  Mean   : 0.04314                       Mean   : 0.04313
##  3rd Qu.: 0.00000                       3rd Qu.: 0.00000
##  Max.   :18.00000                       Max.   :36.00000
##
##  view_alert_count  click_count      click_count_credit_card
##  Min.   :0        Min.   :  0.000   Min.   : 0.0000
##  1st Qu.:0        1st Qu.:  0.000   1st Qu.: 0.0000
##  Median :0        Median :  3.000   Median : 0.0000
##  Mean   :0        Mean   :  6.471   Mean   : 0.1452
##  3rd Qu.:0        3rd Qu.:  8.000   3rd Qu.: 0.0000
##  Max.   :0        Max.   :332.000   Max.   :44.0000
##
##  click_count_personal_loan click_count_mortgage click_count_credit_repair
##  Min.   : 0.00000          Min.   :0.000000     Min.   : 0.00000
##  1st Qu.: 0.00000          1st Qu.:0.000000     1st Qu.: 0.00000
##  Median : 0.00000          Median :0.000000     Median : 0.00000
##  Mean   : 0.04999          Mean   :0.003354     Mean   : 0.01522
##  3rd Qu.: 0.00000          3rd Qu.:0.000000     3rd Qu.: 0.00000
##  Max.   :40.00000          Max.   :6.000000     Max.   :12.00000
##
##  click_count_banking click_count_auto_products click_apply_count
##  Min.   :0.00000     Min.   :0.000000          Min.   : 0.0000
##  1st Qu.:0.00000     1st Qu.:0.000000          1st Qu.: 0.0000
##  Median :0.00000     Median :0.000000          Median : 0.0000
##  Mean   :0.00078     Mean   :0.000676          Mean   : 0.2935
##  3rd Qu.:0.00000     3rd Qu.:0.000000          3rd Qu.: 0.0000
##  Max.   :9.00000     Max.   :6.000000          Max.   :78.0000
##
##  click_count_credit_card.1 click_count_personal_loan.1
##  Min.   : 0.0000           Min.   : 0.00000
##  1st Qu.: 0.0000           1st Qu.: 0.00000
##  Median : 0.0000           Median : 0.00000
```

```
##  Mean   : 0.2231           Mean   : 0.03568
##  3rd Qu.: 0.0000           3rd Qu.: 0.00000
##  Max.   :78.0000           Max.   :29.00000
##
##  click_count_mortgage.1 click_count_credit_repair.1 click_count_banking.1
##  Min.   : 0.00000       Min.   : 0.000000           Min.   :0.000000
##  1st Qu.: 0.00000       1st Qu.: 0.000000           1st Qu.:0.000000
##  Median : 0.00000       Median : 0.000000           Median :0.000000
##  Mean   : 0.01809       Mean   : 0.008028           Mean   :0.000354
##  3rd Qu.: 0.00000       3rd Qu.: 0.000000           3rd Qu.:0.000000
##  Max.   :47.00000       Max.   :12.000000           Max.   :7.000000
##
##  click_count_auto_products.1 logged_in_count
##  Min.   : 0.000000           Min.   : 0.0000
##  1st Qu.: 0.000000           1st Qu.: 0.0000
##  Median : 0.000000           Median : 1.0000
##  Mean   : 0.002792           Mean   : 0.8591
##  3rd Qu.: 0.000000           3rd Qu.: 1.0000
##  Max.   :20.000000           Max.   :28.0000
##
##         session_start_timestamp          city            state
##  2018-07-17 16:43:18:      8               :105144   TX     :118303
##  2018-07-29 22:56:46:      8    Chicago    : 29657   CA     :108108
##  2018-07-31 19:08:39:      8    Houston    : 24797   FL     :100399
##  2018-07-13 01:40:15:      7    Los Angeles: 21083   GA     : 62958
##  2018-07-17 17:55:19:      7    Orlando    : 17660   NY     : 57072
##  2018-07-19 17:43:04:      7    Brooklyn   : 17598          : 56953
##  (Other)            :1176813    (Other)    :960919   (Other):673065
##         browser_name    browser_version        app_version
##  Chrome Mobile:444009   Min.   :   0.2               :878817
##  Unknown      :308217   1st Qu.:  11.0   3.1.15      :146980
##  Mobile Safari:225195   Median :  63.0   3.1.14      : 79971
##  Chrome       : 87207   Mean   :  48.2   3.1.16      : 40244
##               : 30581   3rd Qu.:  67.0   App_Version2: 30581
##  Apple WebKit : 29896   Max.   :9696.0   3.1.12      :    49
##  (Other)      : 51753   NA's   :364845   (Other)     :   216
##    login_platform       os_name
##  Mobile App:339559           :  4098
##  Mobile Web:714899   Android :583434
##  Web       :122400   iOS     :202075
##                      Linux   :  1294
##                      LiveArea:    33
##                      OS X    :284536
##                      Windows :101388
```

- 2.1.4 k-Means: The k-means clustering was meant as a way to cluster users based on their engagement. We decided that the best way to gauge engagement is based on the number of clicks per page. So, we took the 30 day engagement dataset and only took the features that represented clicks per various pages like credit cards, loans, etc. Although the dataset was very sparse, we didn't need to preprocess it any further.

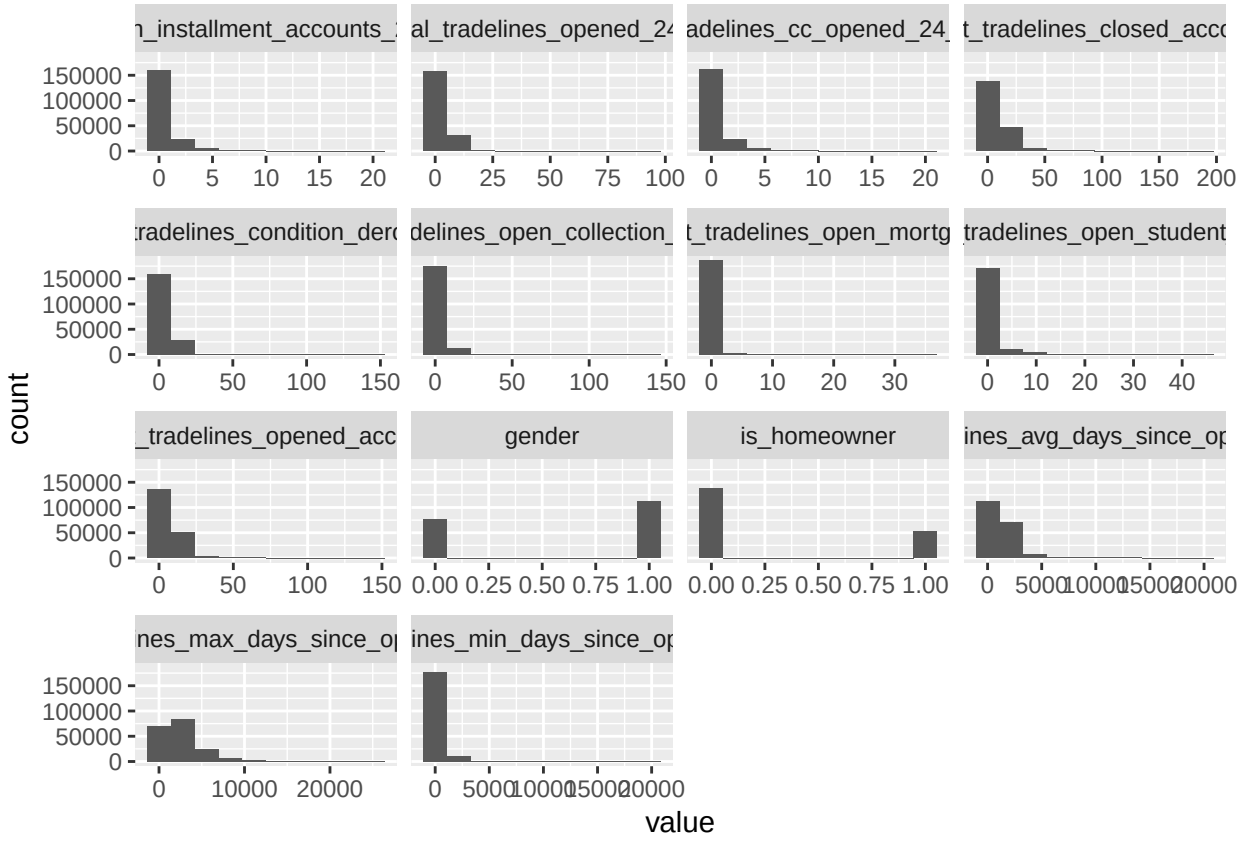
2.2 Exploratory Data Analysis

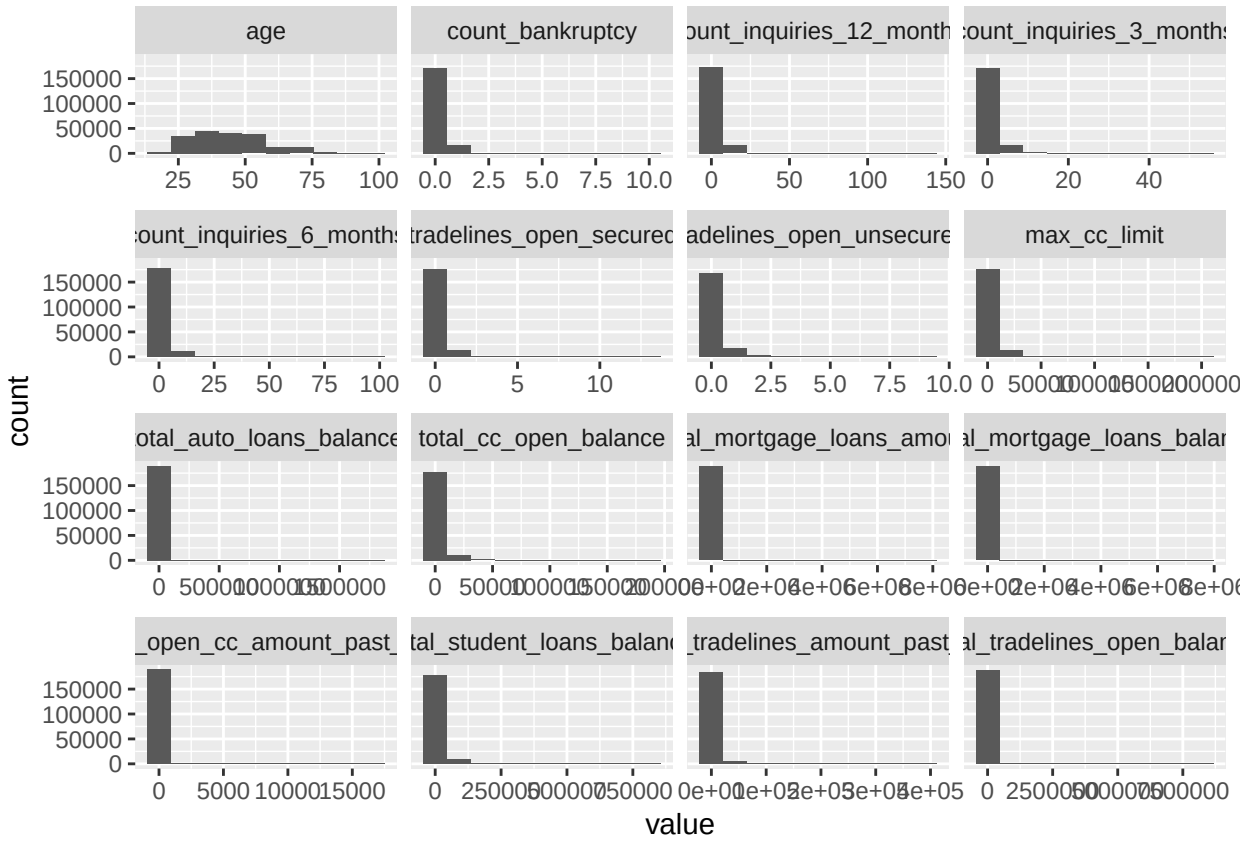
HARI, PAUL, CALVIN ADD IN EDA. (We should mention that some variables are very correlated - the correlation matrix is easy to make)

We started our exploratory data analysis by first looking at the distribution of each of the variables in the dataset, as shown below.

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
##       key                                     value
## 1 user_id 50991631a5e7fafd8b5856fc15e3d1a3af5dcf98
## 2 user_id cabee62f0c4f26bb088f4a48d9ca5efa3a4f96e3
## 3 user_id 6da929725c76c01aa151d97060df2e6bd051e31e
## 4 user_id e8a6717452a88ec8d699c0a4181637c67d247e84
## 5 user_id 03c209fbb349633c40826a83874f92e302382b13
## 6 user_id ae0ebe7492c5af1fec00c8ecd59f83cc5a659fb2
```





As shown by the histograms of the column variables above, we noticed that almost all the non-categorical data was heavily skewed to the right as many of the variables had a high frequency for lower values and significantly lower frequency for higher values. The only non-cateogrical variables that were not heavily skewed were `age` and `credit_score`. This was important to keep in mind as we proceeded in our analysis as the distribution of each variable can impact the results of our analysis methods.

3 Methods

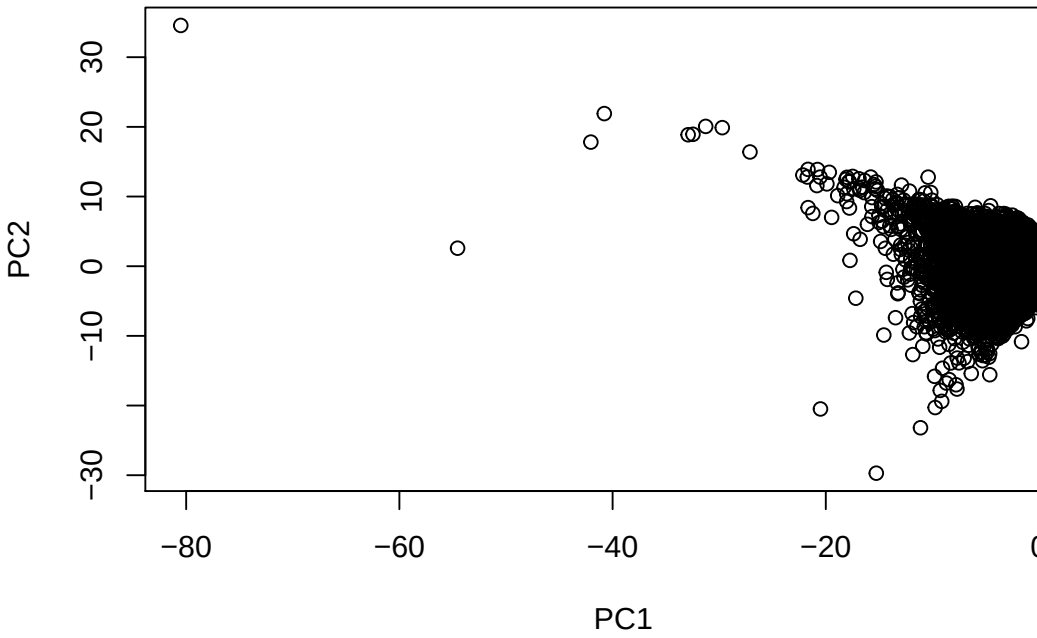
- 3.1 PCA: We used Principle Component Analysis because it is a powerful tool that can be used for both exploratory data analysis and also making conclusions about the dataset. PCA continually takes the axis of greatest variance and calculates the correlation coefficients along this axis as loadings. We wanted some method to visualize the relationship between data points, but the dataset's dimensionality was too large to plot. Using PCA, we could plot two or three of the Principle Components in order to have some sort of visualization of the data, and can offer some intuition of the relationships between data. In plotting it, we were hoping to see two or more distinct clusters of datapoints, but we were left with a large clump. However, the biplot and loadings revealed interesting information that led to other discoveries. Interpreting the PCA is difficult, but by analyzing the loadings, we can see how correlated various features are within each principle component, which tells us which features are related to each other.
- 3.2.2 Linear Regression: Based on our EDA and PCA results, we believed that we could attempt to model a regression on credit score using the given demographic variables. Using our PCA results, we were able to to load the initial model with variables thappeared to be correlated with credit score. This initial model had a decent R-squared of about .36, which we believed to reasonable for real world variables, and therefore we continued with the method and began to remove the least significant

predictors to create a leaner model and ensure there was no colinearity. Every time a variable was removed we would ensure that the R-squared had not dropped past .30. Finally, we reviewed our residual plot and found that we had some clustering indicative of colinearity in the model and used a matrix plot to find the source of the colinearity and removed one of the variables causing it.

- 3.2.3 Lasso: Regular linear regression cannot work properly when features are highly correlated. However, Lasso is a form of linear regression that allows us to input a dataset with correlated variables. Lasso will account for these variables and remove them based on the lambda regularization. We used this method because our process of removing features by hand using the R-squared was slow and had a lot of room of error. We also had to do a lot of preprocessing for linear regression that could have removed potentially important features. Using lasso, we were free to input the entire dataset without worrying about correlated variables. Overally, we implemented lasso as an alternative to linear regression that would work better with the dataset due to the correlated variables.
- 3.2.4 k-Means: K-means clustering allows us to find groups of datapoints together based on their distances from each other. By dictating a number of clusters we expect to see, the algorithm will find the closest subset of datapoints for each of clusters. This model is useful because it gives us a definitive way to find which datapoints are related based on a variety of features. For our dataset, we can take datapoints for different clusters and view them as a collective, and find the demographics of the clusters based on its datapoints.

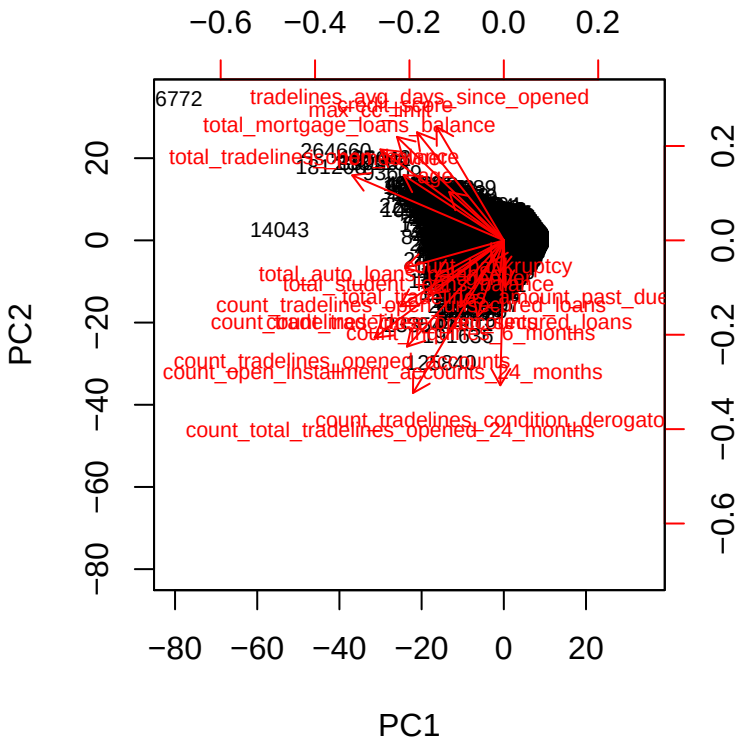
4 Applications

- 4.1 Applications for PCA: After doing PCA on the user_demographics, the first thing that we looked at

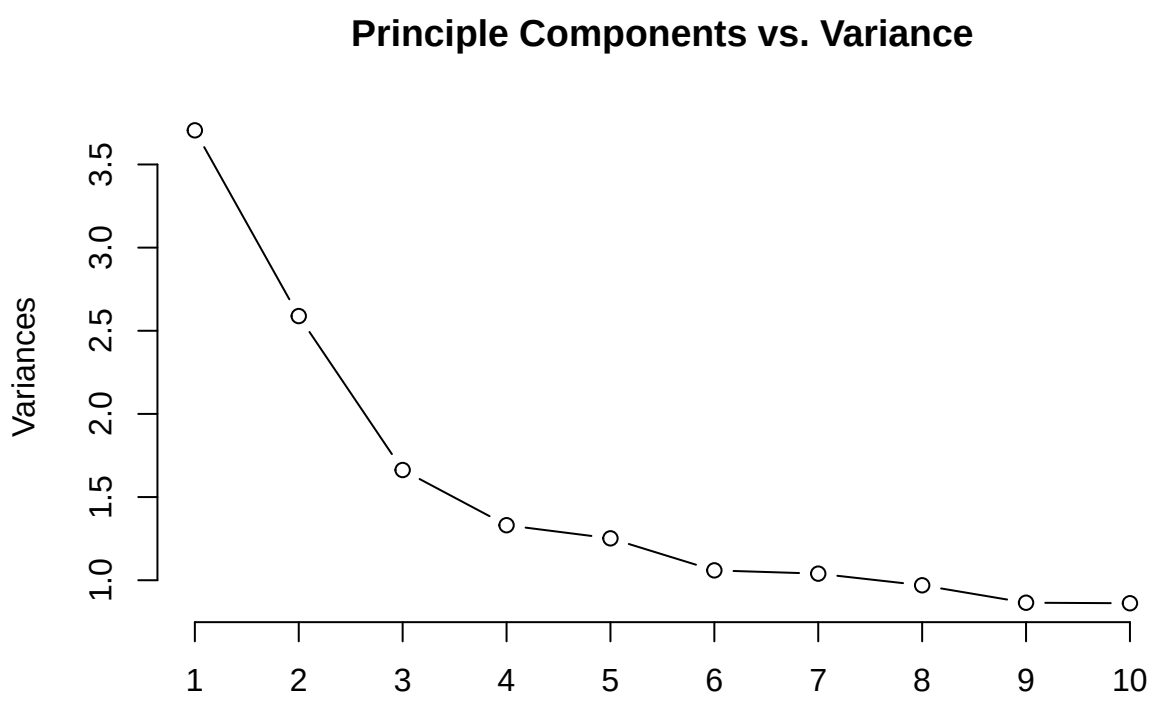


was the plot of PC1 vs. PC2.

Initially, we were hoping that we could find distinct clusters of users based on their traits. However, the plot is clearly just a large clump of datapoints. To find some meaning from it, we also plotted the biplot of the PCA.



The biplot reveals much more information about the loadings and relationships between values. We can see that the entire first loading is negative. The second loading is much more interesting - there is a split between the PC2 loadings. Either the value is relatively positive or negative, there are only a couple features with a moderate loadings. It is difficult to interpret the meaning of the loadings as a whole, but we can see that the positive correlations easily differentiable from the negative loadings. Next, we looked at the loadings themselves and scree plot of the PCs.



From the scree plot, we can see that there are large differences in variance from PCs 1 to 4. Once we get to PC4, we see a small change in the decrease of variation, which means PC4 is the last significant PC that provides a large decrease in variance. Finally, we interpreted the Principle Components loadings by looking at PC1 through PC4.

```
##                                              PC1         PC2
## is_homeowner                          -0.265749640  0.17305440
## gender                                -0.010274389 -0.08327323
## tradelines_avg_days_since_opened      -0.178215179  0.30098061
## count_tradelines_closed_accounts      -0.274483231 -0.17461720
## count_total_tradelines_opened_24_months -0.240645508 -0.40437433
## count_tradelines_condition_derogatory -0.009301021 -0.38383500
## count_open_installment_accounts_24_months -0.256623494 -0.28157493
## count_tradelines_opened_accounts      -0.342340376 -0.25795940
## count_tradelines_open_secured_loans   -0.115586155 -0.17626423
## count_tradelines_open_unsecured_loans -0.196337473 -0.13971221
## total_tradelines_amount_past_due       0.004046851 -0.12392095
## total_tradelines_open_balance         -0.401728005  0.17239936
## max_cc_limit                          -0.283081287  0.27432410
## total_mortgage_loans_balance          -0.327026202  0.24094675
## total_auto_loans_balance              -0.261479615 -0.07301565
## total_student_loans_balance           -0.180515221 -0.09485020
## count_inquiries_6_months              -0.069988959 -0.20048977
## count_bankruptcy                      -0.033354933 -0.05813271
## age                                   -0.144567316  0.12806396
## credit_score                          -0.229915247  0.28650043
##                                              PC3         PC4
```

```
## is_homeowner                                -0.11400276  0.18499807
## gender                                      -0.09514876 -0.02343193
## tradelines_avg_days_since_opened            -0.24973575  0.11615267
## count_tradelines_closed_accounts            -0.29015770  0.21957401
## count_total_tradelines_opened_24_months      0.06936418 -0.04979841
## count_tradelines_condition_derogatory       -0.46932016 -0.05413354
## count_open_installment_accounts_24_months    0.37651626  0.07721219
## count_tradelines_opened_accounts            -0.09670775 -0.09930531
## count_tradelines_open_secured_loans          0.14120492  0.30923177
## count_tradelines_open_unsecured_loans        0.22036394  0.23597644
## total_tradelines_amount_past_due            -0.41837417 -0.07128415
## total_tradelines_open_balance               -0.05549340 -0.37604110
## max_cc_limit                                 0.01678692  0.03489727
## total_mortgage_loans_balance                -0.07544896 -0.33184781
## total_auto_loans_balance                     0.22543807 -0.07189954
## total_student_loans_balance                 -0.06203506 -0.28662612
## count_inquiries_6_months                     0.14411127  0.01653901
## count_bankruptcy                            -0.15005839  0.32376684
## age                                         -0.25775070  0.47478425
## credit_score                                 0.19549777  0.23517635
```

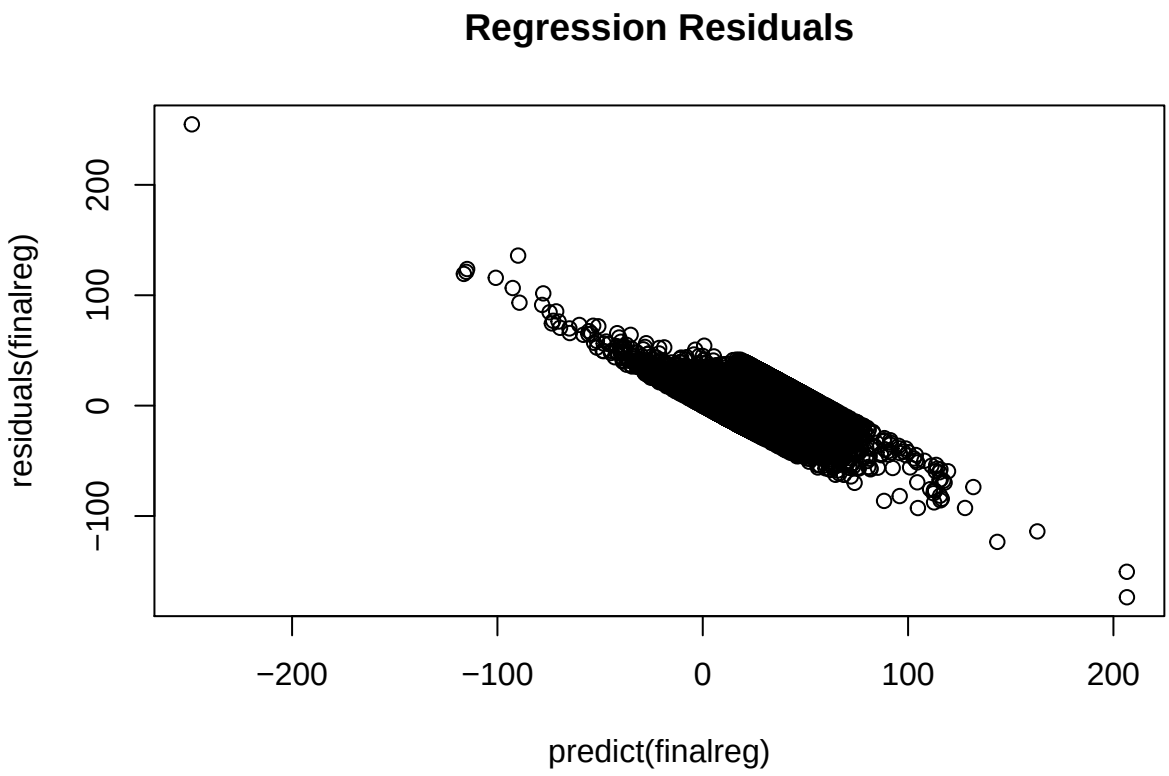
The values of the loadings correspond to the correlation coefficient of the variable with that principle component. What we are looking for in the loadings are values that are related to credit_score. From the first PC, we see that all of the correlation values are negative, some more than others. Credit_score has a loading of -0.23, which means it is moderately related to PC1, but we cannot make any conclusion with respect to the other variables. We could say that values extremely close to 0, meaning that are not very correlated with PC1, probably are not related with credit_score, but its hard to say for certain. PC2 gives us much more information. The correlation for credit_score in PC2 is 0.29, and we can see other variables that are relatively large positive numbers and small negative numbers. We can definitely interpret this principle component. Looking at the large positive correlations, we can see that tradelines_avg_days_since_opened, max_cc_limit, and total_mortgage_loans_balance are strongly positively correlated with credit_score. This means that as these values increase, we would expect to see credit_score increase as well. As for the negative correlation coefficients, we can say that count_total_tradelines_opened_24_months, count_tradelines_condition_derogatory, and count_open_installment_accounts_24_months have an inverse relationship with credit_score. The other negative but lower magnitude coefficients could also have some sort of relationship with credit_score. We can continue down the line with PC3 and PC4, but the interpretability goes down as you analyze more principle components and they become less significant.

From our PCA, we determined which features are correlated with each other, which provided us with useful information that was used for future analysis. Although the interpretability of PCA is lower than other models, we can still determine relationships between variables and datapoints.

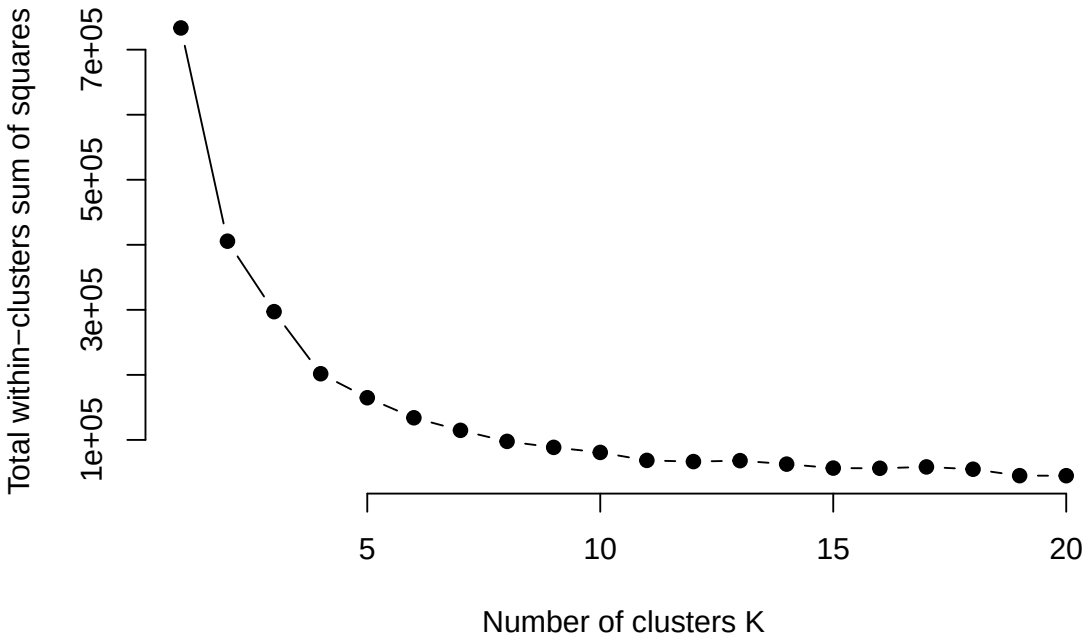
- 4.2 Applications for Regression: Based on our regression model, we know that being a homeowner, your number of open collection accounts, your maximum credit card limit, and the average days since your account has been open, are the most significant indicators of credit score and future delinquent payments. We expected open collections to have a strong predictive relationship with delinquencies and credit score since open collections are caused by missed payments. Our results indicate that each account turned over to a third party for collections leads to a 5 point decrease in credit score. Homeowners tend to have a credit score that is 20 points above non-homeowners. This makes sense because homeowners tend to be more financially stable. Every dollar increase on a credit card limit tends to indicate a 0.0045 point increase in credit score. This is probably explained by the fact that people with better credit scores who are more financially stable are given larger credit card limits. We also found that with every day an individual has an account open, his credit score increases by 0.007135 points. This last finding is also logical because accounts that stay open are accounts that have gone

longer making payments. While these are useful observations for getting a better insight into variables that appear to be correlated with credit score and how they are correlated, we can only use this model with caution is its R-squared is .3227, meaning that the model is only able to account for 32.27% of the variance in credit scores, meaning that these variables serve only as a moderate predictor of quality of credit score and therefore the reliability of the person we could potentially give a loan. The further limits of this regression lie in the skewed overall distribution of this data, which resulted in residual plots for the model that indicate that we must be wary using this model.

```
##
## Call:
## lm(formula = credit_cat ~ home_cat + count_tradelines_open_collection_accounts +
##     max_cc_limit + tradelines_avg_days_since_opened, data = user_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -173.589   -9.141   -1.967    7.520  254.818
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                               1.577e+01  3.965e-02  397.79
## home_cat                                  3.962e+00  5.325e-02   74.41
## count_tradelines_open_collection_accounts -9.482e-01  6.159e-03 -153.95
## max_cc_limit                              9.073e-04  4.441e-06  204.31
## tradelines_avg_days_since_opened          1.427e-03  2.486e-05   57.42
##                                          Pr(>|t|)
## (Intercept)                                <2e-16 ***
## home_cat                                   <2e-16 ***
## count_tradelines_open_collection_accounts  <2e-16 ***
## max_cc_limit                               <2e-16 ***
## tradelines_avg_days_since_opened           <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.57 on 272813 degrees of freedom
##   (12673 observations deleted due to missingness)
## Multiple R-squared:  0.3227, Adjusted R-squared:  0.3227
## F-statistic: 3.25e+04 on 4 and 272813 DF,  p-value: < 2.2e-16
```



- 4.3 Applications for k-means Clustering: From our preprocessed data of clicks on different parts of the website, we start to run the k-means clustering algorithm. First, we need to decide on the number of clusters that we want. To do so, we run k-means for 1 to 20 clusters, calculating the within-sum-of-squares error for each. The within-sum-of-squares error tells us how separated our data is from the mean of the cluster. To have the best closest clusters, we want to minimize this value.



The above graphic tells us that between 5 and 8 number of clusters seems to be the amount where the within-cluster sum of squares difference starts to drop off. Therefore, we will use 8 clusters, which will help with accounting for outliers. We run our k-means clustering on the dataset with k=8.

```
##   click_count_credit_card click_count_personal_loan click_count_mortgage
## 1              2.42372881                0.24250326          0.043024772
## 2              0.00000000                0.00000000          0.001996741
## 3              0.07725445                1.46600373          0.013497990
## 4              0.52521178                5.06978620          0.034288019
## 5              0.08707440                0.08149270          0.017542461
## 6              4.00561366                0.11665641          0.025146109
## 7             10.54363284                0.21913381          0.045248869
## 8              1.26445073                0.03114592          0.009925624
##   click_count_credit_repair click_count_banking click_count_auto_products
## 1                1.40286832        0.0000000000              0.0234680574
## 2                0.00000000        0.0003855379              0.0002551072
## 3                0.03212952        0.0081849512              0.0077900632
## 4                0.14199274        0.0189592578              0.0254134732
## 5                1.21122717        0.0015947692              0.0051829998
## 6                0.02230083        0.0066133497              0.0029221778
## 7                0.07175178        0.0051712993              0.0058177117
## 8                0.00000000        0.0016718225              0.0015665109
```

Looking at these cluster centers, we can see that cluster 7 and 4 have relatively high values for click_count_credit_card and click_count_personal_loan, respectively. This means that the clustered groups tend to click on the these sites more often. The other clusters are all more centered to the origin. This could have to do with the sparse natural of the dataset, which might be having an adverse effect on the

results. Nonetheless, we can see two clearly different clustered groups in cluster 7 and 4. We then compare the user demographic for users of different groups.

```
##                                          click_credit    click_loan
## count_tradelines_condition_derogatory         5.267442     2.2942584
## count_tradelines_open_collection_accounts     2.663848     0.8931419
## total_tradelines_amount_past_due           3731.251586  1217.4928230
## total_open_cc_amount_past_due                 1.661734     0.4170654
## total_cc_open_balance                      1520.512685  4543.9377990
## total_tradelines_open_balance             35731.413319 60459.4736842
## max_cc_limit                               1834.285412  4855.6355662
## total_mortgage_loans_amount               14611.932347 34770.7679426
## total_mortgage_loans_balance              12857.081395 30600.4059011
## total_auto_loans_balance                   6974.323467 10398.5669856
## credit_score                                587.521142   657.9465710
```

The above table is a subset of all of the features that differ between the two groups (check appendix for the full list of features). We can see that in several categories there is quite a large difference between the mean variables of the two clustered groups, and they are make sense. The users who click on the credit pages more often have a lower credit score (they click on the credit pages because they are interested in improving their credit), have inflated values that are associated with good credit (click_credit has lower cc_open_balance, max_cc_limit,count_tradelines_condition_derogatory). The click_loan group tend to have higher loan amounts, and they click on the loan pages more often possibly to learn how to best pay back their loans, or perhaps they clicked on the pages to take a loan.

From the k-means clusters, we were able to determine that the demographics of people who use Credit Sesame's website differently and access different pages are in fact different. Naturally, the people who click more on credit or loans are the people who need help in improving those areas. By using k-means clustering, we were able to look at different people as a group, which meant we didn't have to compare individuals to other individuals, and it gives us a better view of what different groups of people are using Credit Sesame.

5 Discussion