Cluster Analysis of Credit Card Data

Christopher 'Kitt' Burch

9/21/2021

Data

Source

This data set was made available as part of a Kaggle Competition and is licensed under a creative commons license (CC0).

Description

The data set summarizes the usage behavior of about 9000 active credit card holders during a 6 month period. The file is at a customer level with 18 behavioral variables.

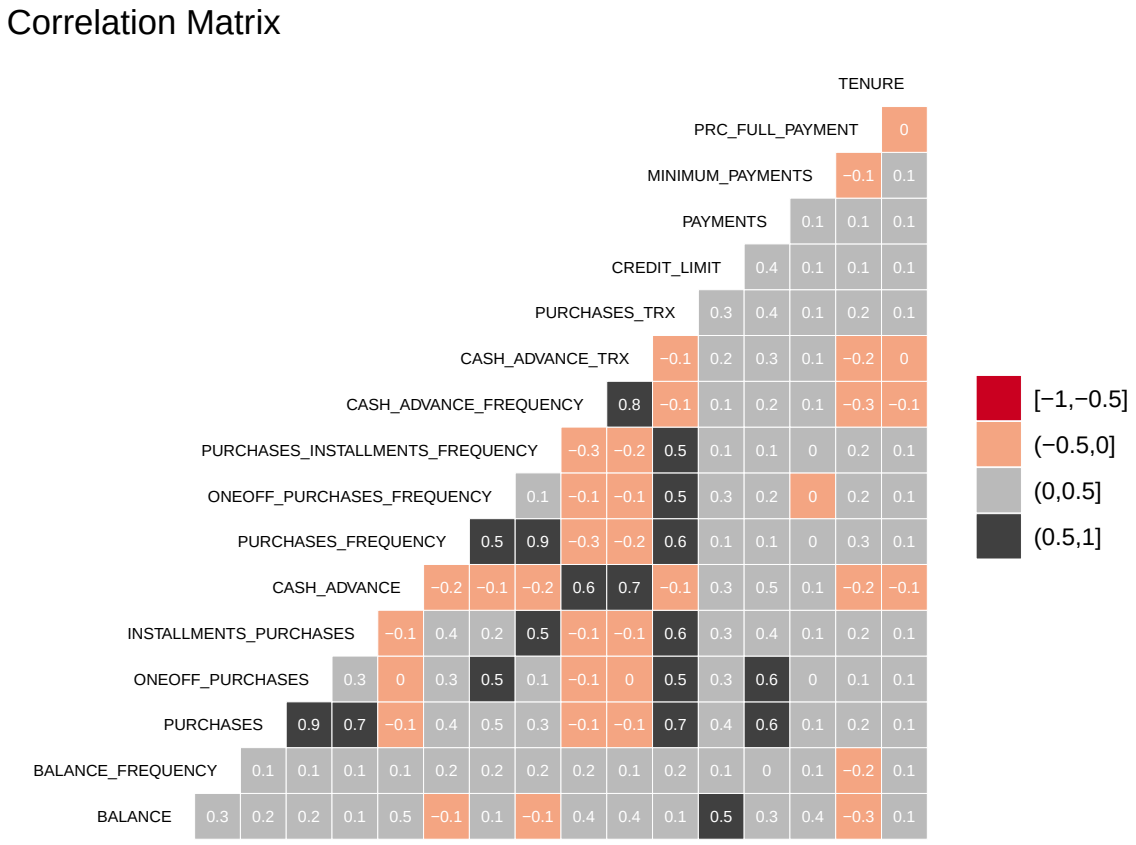
Variables

The data set contains 8950 records with 18 variables. One variable (CUSTID) is a character label. The other 17 variables are numeric.

variable	description
CUSTID	Identification of Credit Card holder (Categorical)
BALANCE	Balance amount left in their account to make purchases
BALANCEFREQUENCY	How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated)
PURCHASES	Amount of purchases made from account
ONEOFFPURCHASES	Maximum purchase amount done in one-go
INSTALLMENTSPURCHASES	Amount of purchase done in installment
CASHADVANCE	Cash in advance given by the user
PURCHASESFREQUENCY	How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased)
ONEOFFPURCHASESFREQUENCY	How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased)
PURCHASESINSTALLMENTSFREQUENCY	How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done)
CASHADVANCEFREQUENCY	How frequently the cash in advance being paid
CASHADVANCETRX	Number of Transactions made with "Cash in Advanced"
PURCHASESTRX	Number of purchase transactions made
CREDITLIMIT	Limit of Credit Card for user
PAYMENTS	Amount of Payment done by user
MINIMUM_PAYMENTS	Minimum amount of payments made by user
PRCFULLPAYMENT	Percent of full payment paid by user
TENURE	Tenure of credit card service for user

Overview

Correlations



Characteristics

Data relating to money often follows a logarithmic or scale-free distribution, meaning that extreme values are expected and the data is skewed to the right. This causes most values to 'bunch' around the lower values with higher values distorting the distribution and complicating statistical analysis.

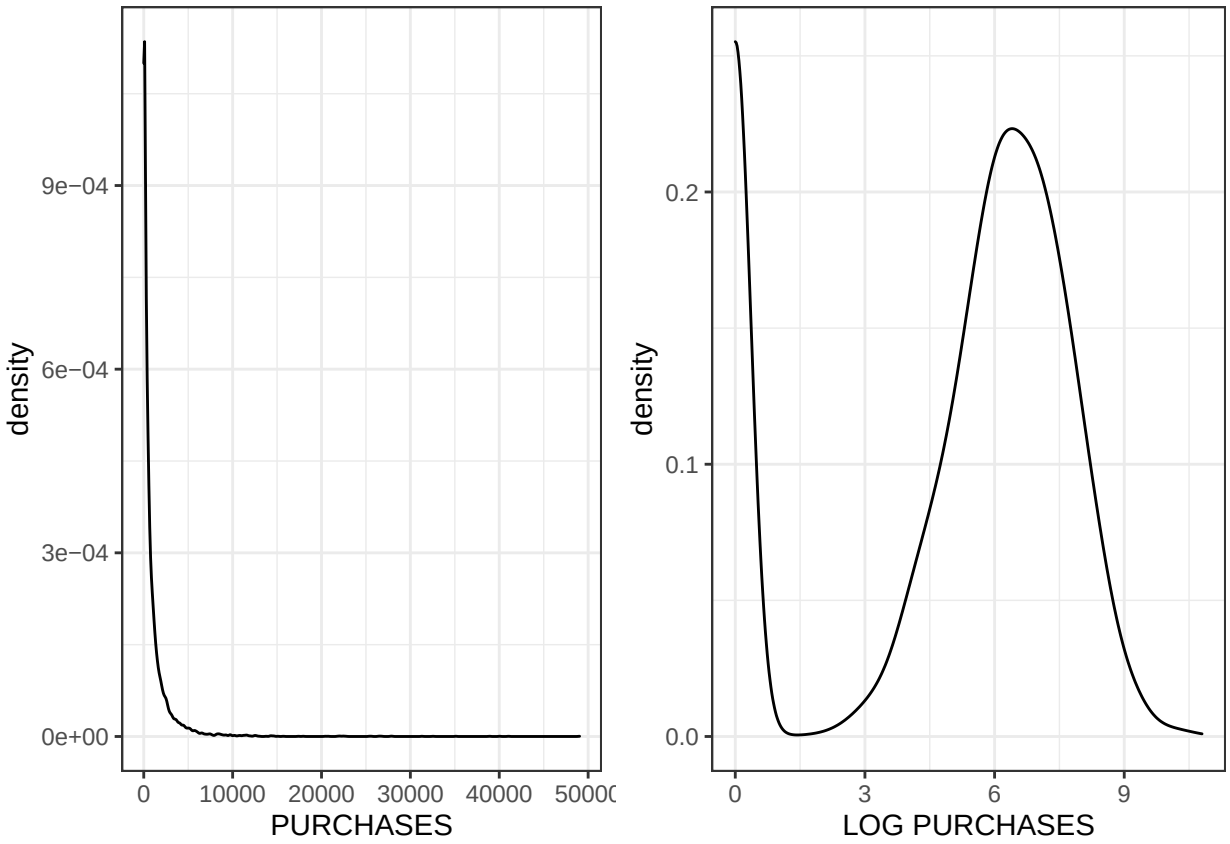
The **skewness** of each variable indicates how each variable is shifted around the mean.

```
##                          BALANCE                  BALANCE_FREQUENCY
##                       2.39298490                       -2.02292641
##                        PURCHASES                ONEOFF_PURCHASES
##                       8.14290404                       10.04339927
##             INSTALLMENTS_PURCHASES                    CASH_ADVANCE
##                       7.29789654                        5.16574312
##                PURCHASES_FREQUENCY          ONEOFF_PURCHASES_FREQUENCY
##                       0.06015415                        1.53535541
## PURCHASES_INSTALLMENTS_FREQUENCY              CASH_ADVANCE_FREQUENCY
##                       0.50911582                        1.82837977
##                  CASH_ADVANCE_TRX                     PURCHASES_TRX
##                       5.72033928                        4.62987914
##                     CREDIT_LIMIT                         PAYMENTS
##                                NA                        5.90662964
##                  MINIMUM_PAYMENTS                PRC_FULL_PAYMENT
##                                NA                        1.94249431
```

```
##                          TENURE
##                     -2.94252402
```

Higher skew indicates more of a shift to an extreme value in the form of a 'tail' to the distribution. In order to combat this tendency, the natural log of the data was analyzed.

For example, the amount purchased on each account varies from 0 to $49,039 with the mean value at $1,003 and the median at $361. The unmodified distribution of values for the PURCHASES variable are right skewed:



In order to counteract this tendency, the data set was scaled to the natural logarithm of the data. Both the log-transformed and non-transformed data were analyzed, with log-transformed data typically performing better in the analysis due to the smoothing effect of log transformations.

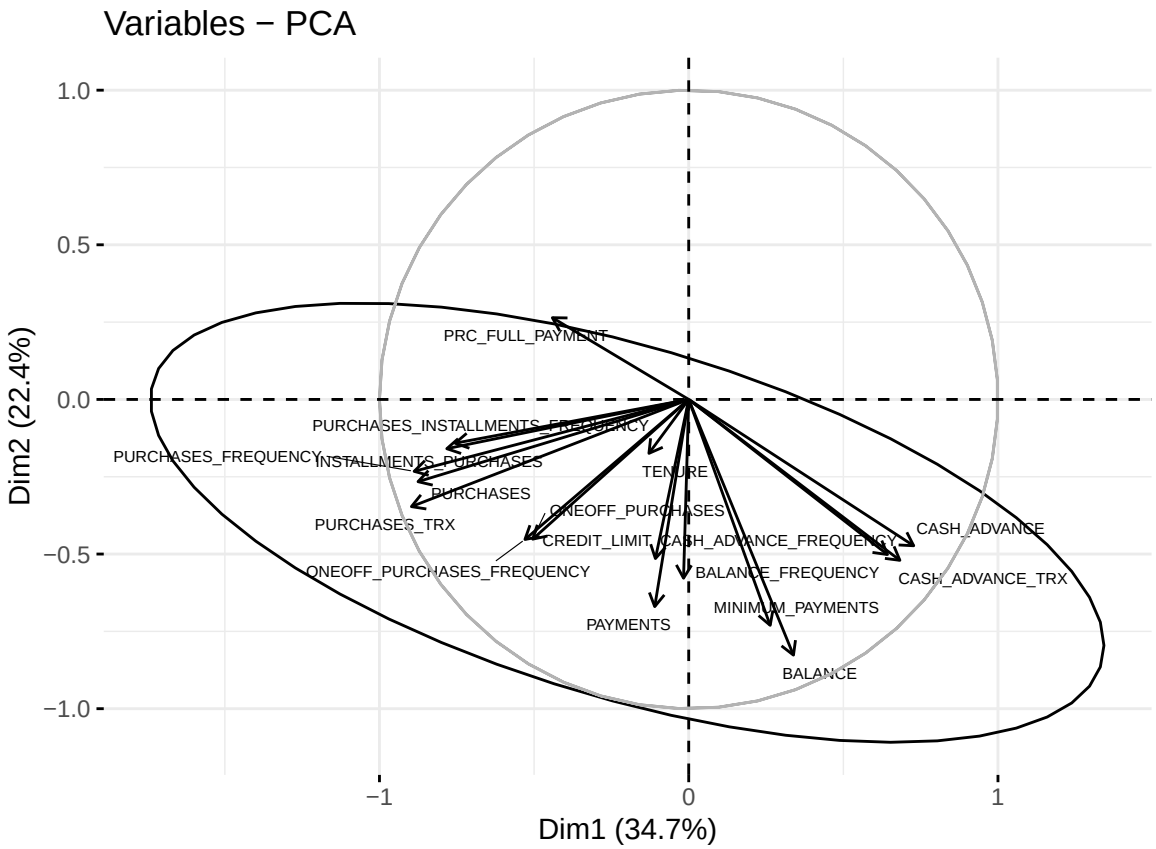
The transformed distribution should be easier to work with.

Analysis

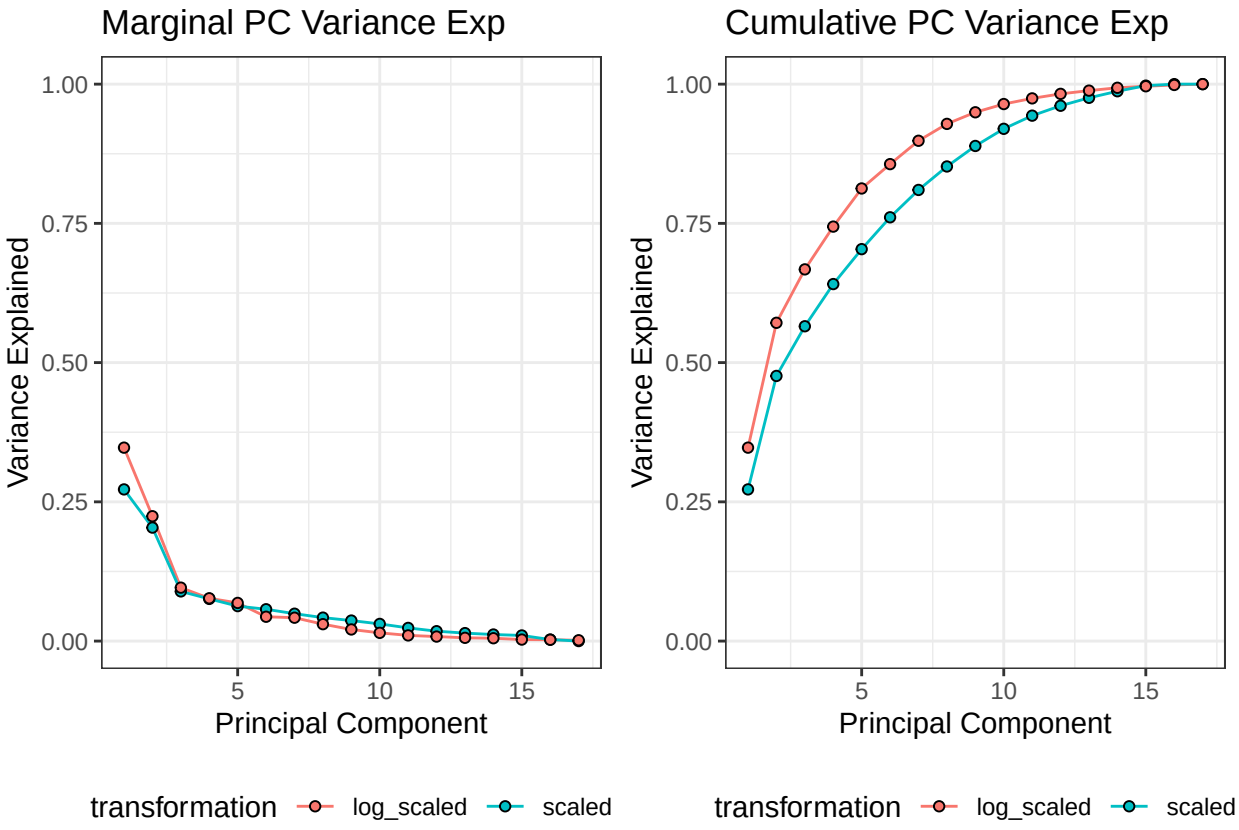
Dimension Reduction

Principal Component Analysis (PCA) is used to reduce the number of dimensions in the data set. High-order data (data with many variables) can cause issues with model convergence. PCA helps combat this by projecting the higher-order data onto a number of smaller "principal components" in the data set. PCA can help significantly reduce dimensionality while keeping most of the data.

When the data are projected on two variables (from 17 numeric variables), the result can look a bit chaotic.


Variables − PCA
1.0
0.5
0.0
−0.5
−1.0
−1
0
1
Dim2 (22.4%)
Dim1 (34.7%)
PRC_FULL_PAYMENT
PURCHASES_INSTALLMENTS_FREQUENCY
PURCHASES_FREQUENCY
INSTALLMENTS_PURCHASES
TENURE
PURCHASES
ONEOFF_PURCHASES
PURCHASES_TRX
CREDIT_LIMIT
CASH_ADVANCE_FREQUENCY
CASH_ADVANCE
ONEOFF_PURCHASES_FREQUENCY
BALANCE_FREQUENCY
CASH_ADVANCE_TRX
MINIMUM_PAYMENTS
PAYMENTS
BALANCE

Principal Components



These scree plots show that taking the natural log of the data prior to deconstructing into Principal Components improves the marginal and cumulative explanatory power of the components.

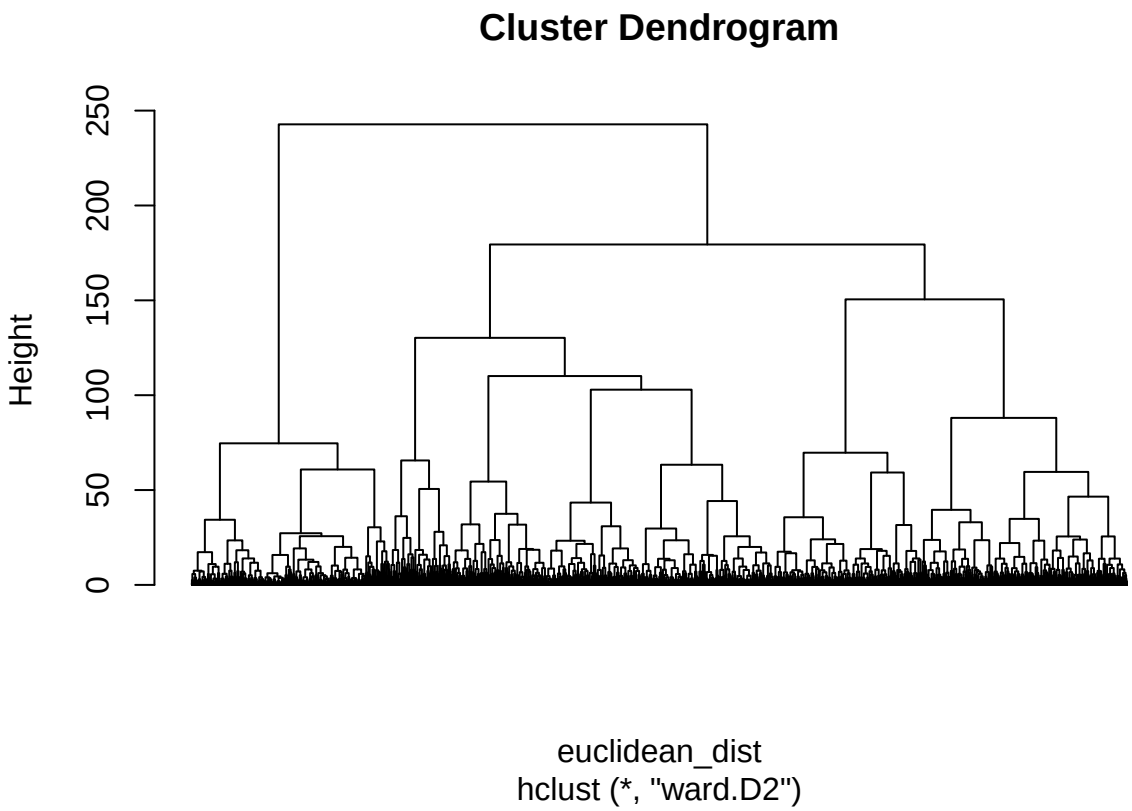
The marginal scree plot shows an 'elbow' at 3 principal components, which suggests that marginal gain in explanatory power declines after projecting the data into more than three PCs.

Clustering

Clustering assigns data points into groups of similar points based off some distance metric. As this data is unlabeled, we do not know how many distinct groups are represented in the data. Several techniques exist that will help determine the appropriate number of clusters to consider.

Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering splits the data into groups based off of the distance between points. HAC calculates trees of clusters of similar points. HAC does not split into a predefined number of groups. Instead, it hierarchically groups points and asks us to choose the appropriate cutoff.



HCA trees are cut into groups based off of height, which is a measure of similarity between points. For each height, the count of points in each group shows how density of clusters will be effected.

At height 150, HCA finds four groups.

```
##
##    1    2    3    4
## 3544 1833 1365 1894
```

At 110, HCA finds six groups.

```
##
##    1    2    3    4    5    6
## 2103 1833 1365  872 1894  569
```

At 100, HCA finds seven groups.

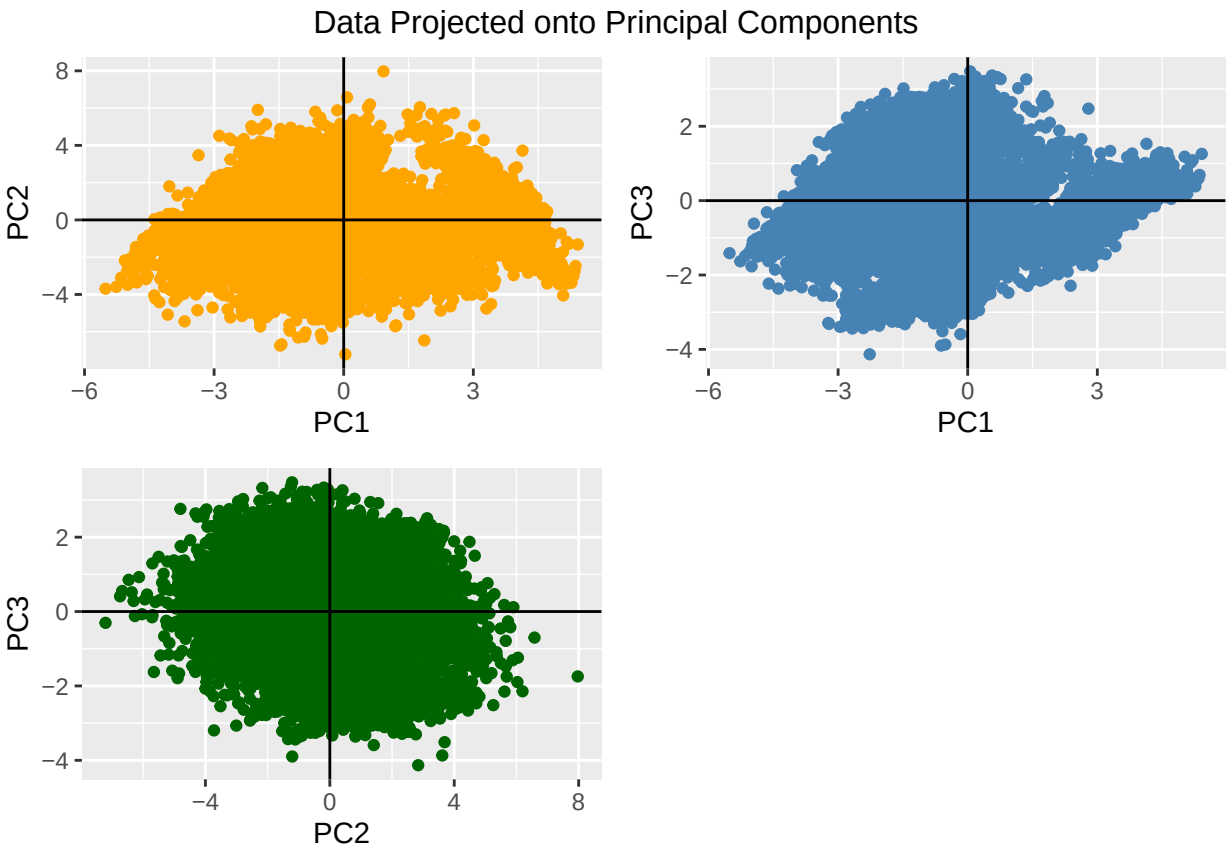
```
##
##    1    2    3    4    5    6    7
## 1225 1833 1365  878  872 1894  569
```

In this case, we choose groups from h = 110 because this results in a relatively even number of points per group. Groups from h = 100 would also be a good choice. Ultimately, both are valid, and should be examined to see which is more applicable to the business problem at hand.

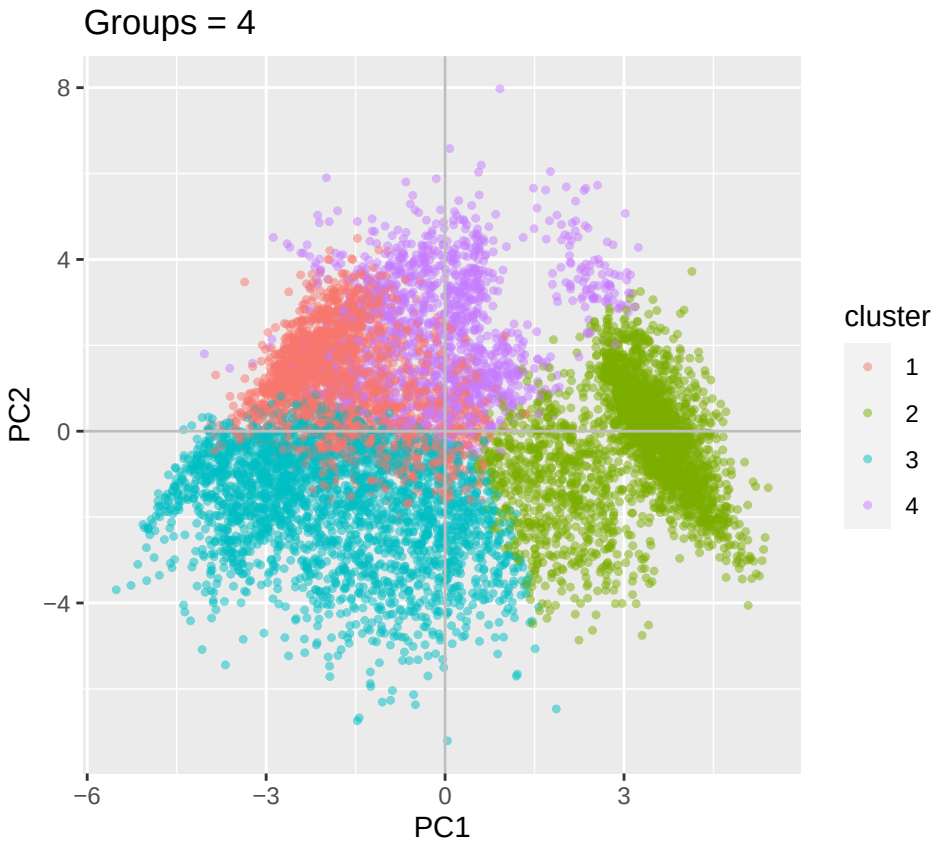
Projected HAC

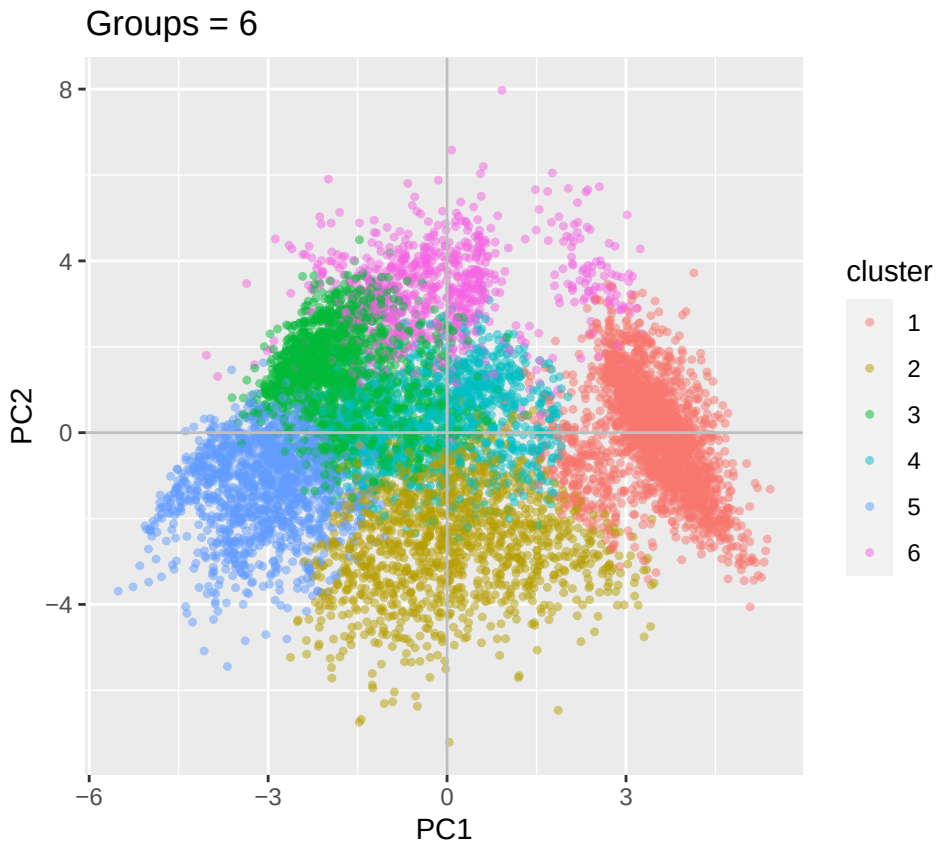
Since only three primary components are necessary to explain most of the variance in the data, projecting the data onto those support vectors will reduce dimensionality while only losing a small portion of data. This may allow for a better segmentation of points into meaningful clusters.

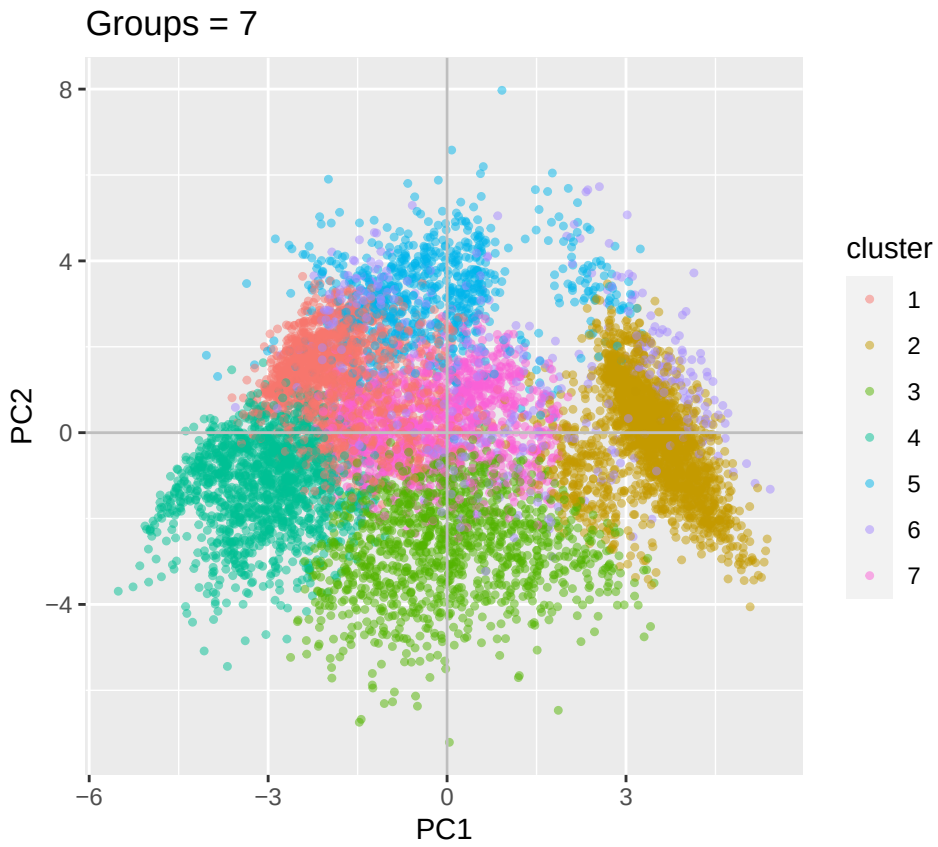
```
##       PC1                PC2                PC3
##  Min.   :-5.513   Min.   :-7.210850   Min.   :-4.13170
##  1st Qu.:-1.963   1st Qu.:-1.290151   1st Qu.:-0.85437
##  Median :-0.437   Median :-0.006021   Median : 0.08285
##  Mean   : 0.000   Mean   : 0.000000   Mean   : 0.00000
##  3rd Qu.: 2.301   3rd Qu.: 1.295578   3rd Qu.: 0.78803
##  Max.   : 5.418   Max.   : 7.971777   Max.   : 3.47567
```



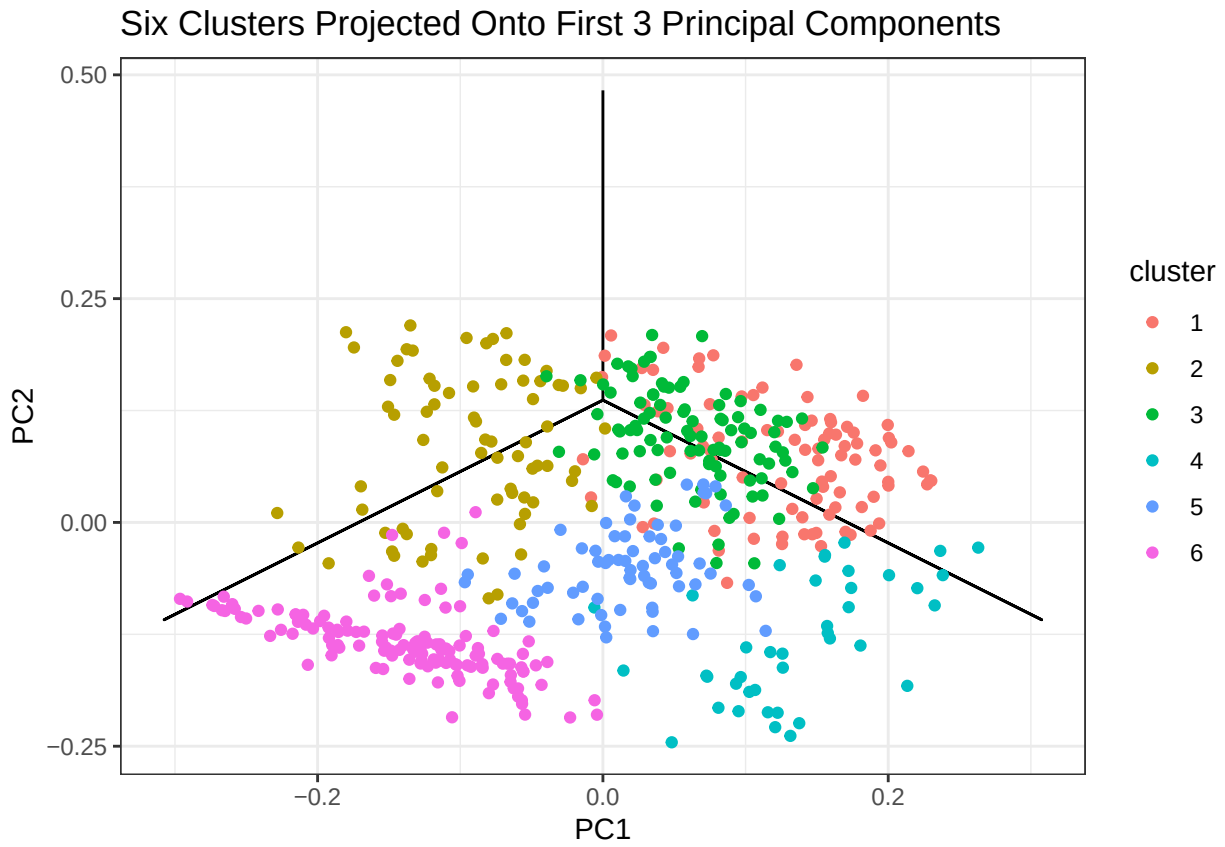
Grouping the Points

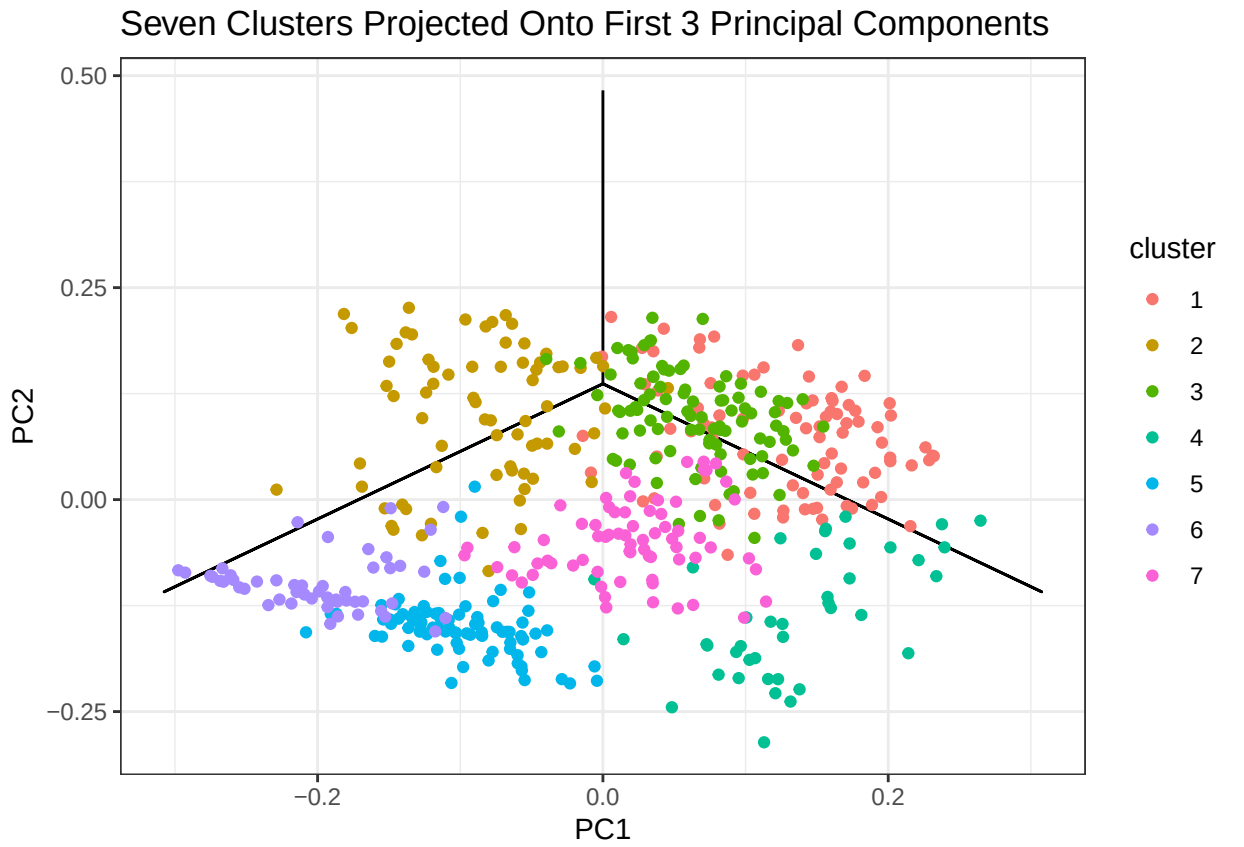



Groups = 6
8
4
0
−4
PC2
−6
−3
0
3
PC1
cluster
1
2
3
4
5
6



Projecting these points onto the first three PCs appears to show spacing between the clusters.


Six Clusters Projected Onto First 3 Principal Components
0.50
0.25
0.00
−0.25
PC2
−0.2
0.0
0.2
PC1
cluster
1
2
3
4
5
6



DBSCAN Clustering

DBSCAN is a popular nonparametric clustering algorithm that can choose the appropriate number of groups on its own. Unfortunately, DBSCAN does not perform well with groups of different densities.

```
##
##    0    1    3    5    7    9   11   13   15   17   19   21   23   25   27   29
## 5459   18   22   17   33   10   15   10   12   10   21  101   10   33   14   56
##   31   37   41   43   46   49   50   51   57   58   61   63   64   66   67   70
##   11   10   18   10   41   14  836   17  147   29   18  117   23   13   38   26
##   71   72   73   74   78   83   84
##   13   18   51   16   28   15 1286
```

DBSCAN and related algorithms (HDBSCAN) cannot separate the data into meaningful clusters due to density. This attempt at a HDBSCAN model produced very small clusters, with the majority of points as unclassified.

Conculsions

There are a number of distinct groups in this data set, an indication of differing but related types of consumer behavior. For marketing and business purposes, it may be useful to segment customers into six (6) or seven (7) groups based on their characteristics and the business problem at hand.

Implications

These groups can be used to target products or services in a marketing context, or may be useful to identify anomalous behavior.

For example, the patterns of purchase amount and balance amount are different for each group. Additional metrics can also be computed for each group - for example, the ratio of purchases to carried balance may provide some insight into purchase patterns for each group.

For six groups:

```
## # A tibble: 7 x 4
##   cluster avg_balance avg_purchases purchase_balance_ratio
##   <fct>         <dbl>         <dbl>                  <dbl>
## 1 1             431.          630.                 1.46
## 2 2            3494.         1401.                 0.401
## 3 3            1540.         3397.                 2.21
## 4 4              41.1         307.                 7.47
## 5 5            1175.          709.                 0.604
## 6 6            2299.           18.4                0.00799
## 7 <NA>          554.          392.                 0.708
```

For seven groups:

```
## # A tibble: 8 x 4
##   cluster avg_balance avg_purchases purchase_balance_ratio
##   <fct>         <dbl>         <dbl>                  <dbl>
## 1 1             422.          620.                 1.47
## 2 2            3311.         1538.                 0.465
## 3 3            1522.         3411.                 2.24
## 4 4              38.4         320.                 8.33
## 5 5            1230.           12.1                0.00984
## 6 6            3954.           78.3                0.0198
## 7 7            1172.          729.                 0.621
## 8 <NA>          554.          392.                 0.708
```

Potential Issues

- The data only covers a 6-month period, which may not be an adequate amount of time for a comprehensive understanding of patterns.

- It is not known how these customers were selected. This may not be a random sample of credit card customers, and if that is the case, the results of this analyisis may not be generalizable outside of the sample population.

Next Steps

Consumer behavior likely changes over the course of a year (holiday shopping, vacation purchases, etc.) It would be beneficial to see if these same groups appear when purchase data for the entire year, or even better, data for a multi-year period is available.

Customer segmentation is likely just the beginning of any useful analysis. The next research step will likely involve applying this analysis to a business problem (increase sales, reduce defaults, identify fraud, etc.)

The code for this analysis can be found at https://github.com/burch-cm/ibm-cert.