Principal Component Analysis of Lending Club Loan Data

AUTHOR
John Liu and Benson Wang

Lending Club Loan Data

The Lending Club loan dataset provides detailed information on consumer loans issued through the Lending Club platform, including borrower attributes, loan characteristics, payment history, and credit risk indicators, making it a valuable resource for analyzing lending trends, default risk, and financial behaviors.

```
library(tidyverse)
library(corrplot)
library(PerformanceAnalytics)
library(heplots)
library(FactoMineR)
library(dplyr)
```

Preparing data

Data was obtained from Kaggle.

```
accepted <- read.csv("../../accepted_2007_to_2018Q4.csv")
```

Subset selection

To analyze the Lending Club loan data, we select a subset to reduce the number of observations (from 2.2M) to have better interpretability. We first subset the data to include only fully paid loans in Connecticut (CT) for credit card purposes with verified income and individual application type.

```
credit <- subset(accepted,
    loan_status == "Fully Paid" &
    addr_state == "CT" &
    purpose == "credit_card" &
    verification_status == "Verified" &
    application_type == "Individual"
)
```

Converting loan grade and sub-grade to numeric

```r
# Define numeric mapping for sub_grade
sub_grade_levels <- c("A1", "A2", "A3", "A4", "A5",
                      "B1", "B2", "B3", "B4", "B5",
                      "C1", "C2", "C3", "C4", "C5",
                      "D1", "D2", "D3", "D4", "D5",
                      "E1", "E2", "E3", "E4", "E5",
                      "F1", "F2", "F3", "F4", "F5",
                      "G1", "G2", "G3", "G4", "G5")

# Assign numeric values (1 to 35) to sub_grade
credit$sub_grade_num <- as.numeric(factor(credit$sub_grade,
                                          levels = sub_grade_levels))

grade_levels <- c("A", "B", "C", "D", "E", "F", "G")
credit$grade_num <- as.numeric(factor(credit$grade,
                                      levels = grade_levels))
```

Variable Selection

We select a subset of variables that are eligible for Principal Component Analysis (PCA) that are continuous, have less than half of the observations missing, and have meaningful values. Finally, we choose a subset of interesting variables for further analysis.

```r
credit <- credit %>%
  # Keep continuous variables
  select_if(is.numeric) %>%

  # Remove variables with too many NAs (more than half of observations)
  select_if(function(x) sum(is.na(x)) <= nrow(credit)/2) %>%

  # Remove variables where min and max are the same
  select_if(~ min(.x, na.rm = TRUE) != max(.x, na.rm = TRUE)) %>%

  # Remove variables where min and median are the same
  select_if(~ min(.x, na.rm = TRUE) != median(.x, na.rm = TRUE)) %>%

  # Remove specific redundant variables
  select(-funded_amnt_inv,
         -total_pymnt_inv,
         -fico_range_low,
         -last_fico_range_low) %>%

  # Remove variables with high multicollinearity
  select(-installment)

# Find and remove duplicated variables
# duplicated_vars <- which(duplicated(as.matrix(credit), MARGIN = 2))
# print("Duplicate variables:")
```

```r
# print(names(credit)[duplicated_vars])
# credit <- credit[, -duplicated_vars]

# choose interesting variables
credit <- credit %>%
  select(grade_num, sub_grade_num, loan_amnt, dti, annual_inc, fico_range_high,
         total_acc, avg_cur_bal, tot_hi_cred_lim)

# Remove rows with missing values
credit <- credit[complete.cases(credit), ]

head(credit)
```

```
     grade_num sub_grade_num loan_amnt   dti annual_inc fico_range_high
929          1             2     28000 12.67      82000             704
1573         3            13     30000 15.61     115000             664
2424         3            12      1050 25.43     130000             699
2730         2             6     10000 35.56     120000             724
2856         1             5     10000 38.86     162000             694
4927         2             8      6000 22.18      48000             754
     total_acc avg_cur_bal tot_hi_cred_lim
929         29       25361          344526
1573        15       29627          441400
2424        29       34334          569975
2730        54       12129          359901
2856        44       35801         1063828
4927        24       18247          302946
```

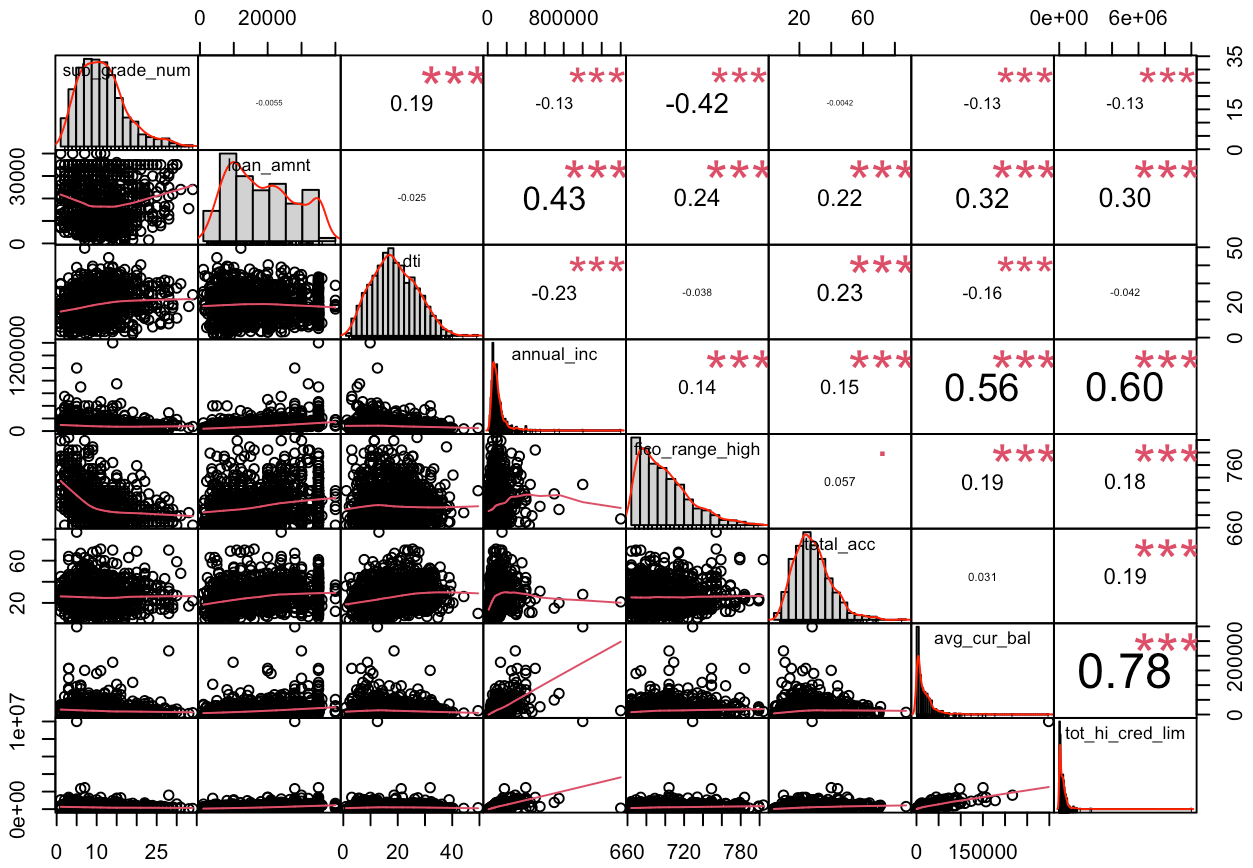
Variable Definitions

Data dictionary for dataset.

- `loan_amnt`: The listed amount of the loan applied for by the borrower.
- `dti`: A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
- `annual_inc`: The self-reported annual income provided by the borrower during registration.
- `fico_range_high`: The upper boundary range the borrower's FICO at loan origination belongs to.
- `total_acc`: The total number of credit lines currently in the borrower's credit file.
- `avg_cur_bal`: Average current balance of all accounts.
- `tot_hi_cred_lim`: Total high credit/credit limit.
- `grade_num`: The numeric value (1-7) of the grade assigned by Lending Club.
- `sub_grade_num`: The numeric value (1-35) of the sub-grade assigned by Lending Club.

Data Distribution (Raw Data)

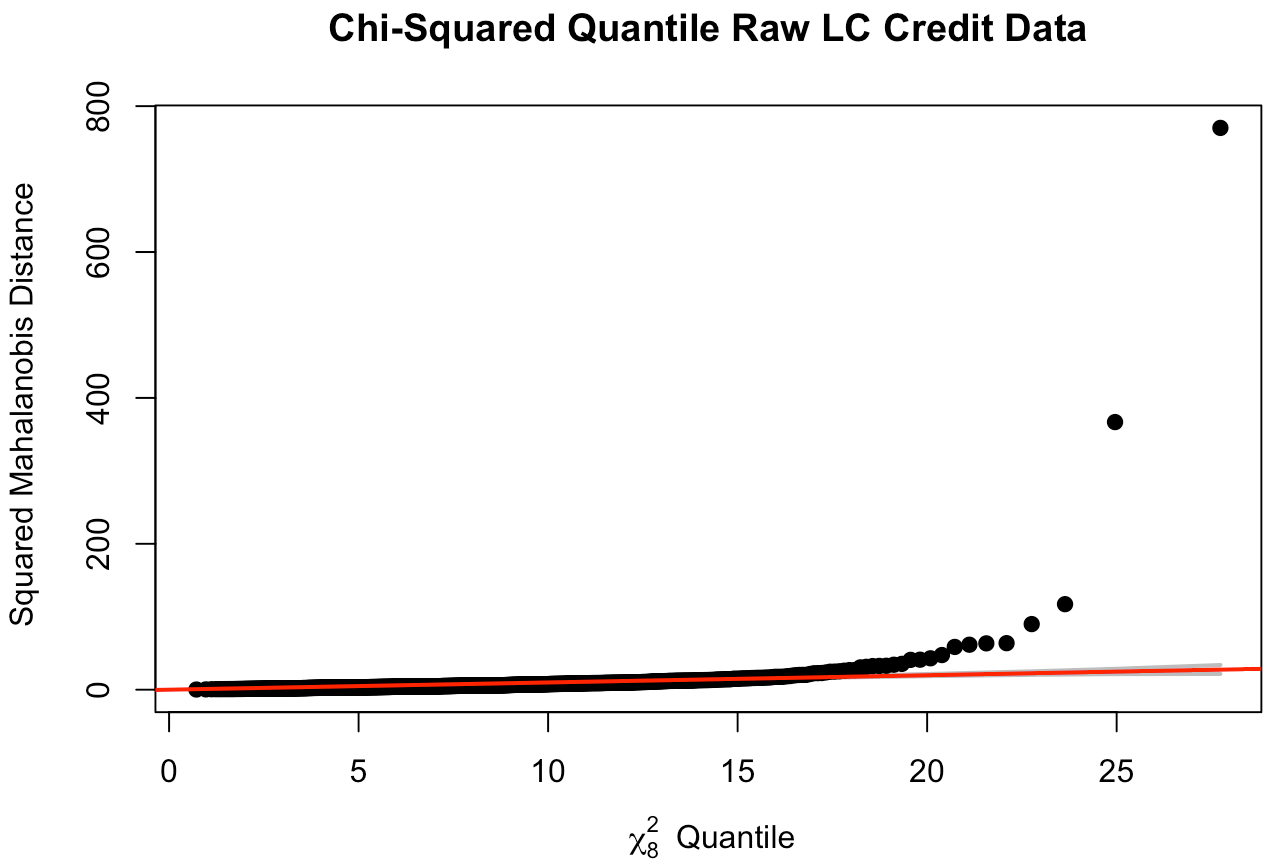
```r
chart.Correlation(credit[, -1])
```



From the correlation graph, we observe that certain variables, such as `annual_inc`, `avg_cur_bal`, `tot_hi_cred_lim`, and `fico_range_high` deviate from a normal distribution. This indicates that the dataset is not multivariate normal, as multivariate normality requires each individual variable to be normally distributed. Additionally, these variables exhibit non-linearity, which may pose a challenge for PCA, as PCA assumes linear relationships among variables.

Let us see if the data is multi-variate normal by plotting the chi-squared quantile plot. (Even though we know it can't be multivariate normal due to the non-normality of the individual variables, we can use this to compare how our transformed data looks.)

```
cqplot(credit[, -1], main = "Chi-Squared Quantile Raw LC Credit Data")
```



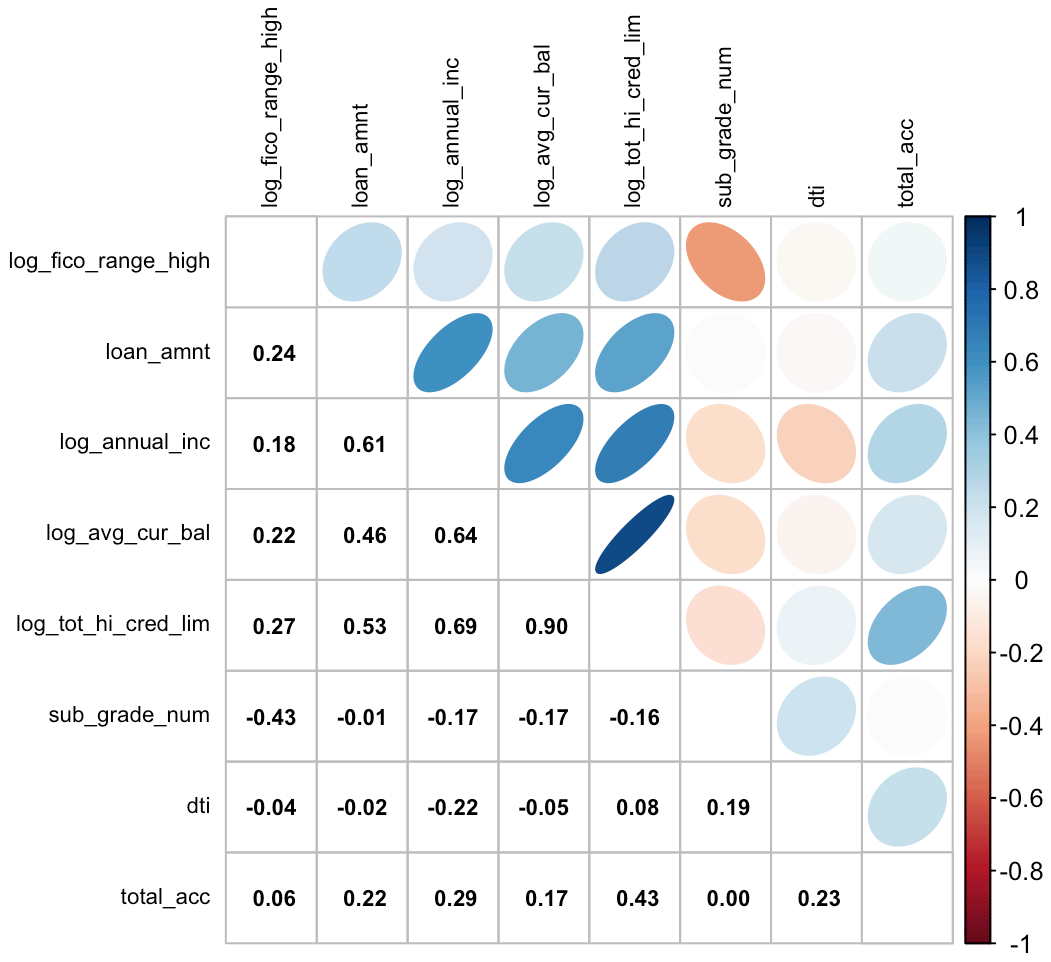
This is not multivariate normal as points right winged points are distant from the confidence band. We will transform the variables to obtain linearity across all variables. Let's see if we could obtain a multivariate normal distribution.

We will log-transform `annual_inc`, `avg_cur_bal`, `tot_hi_cred_lim`, and `fico_range_high`.

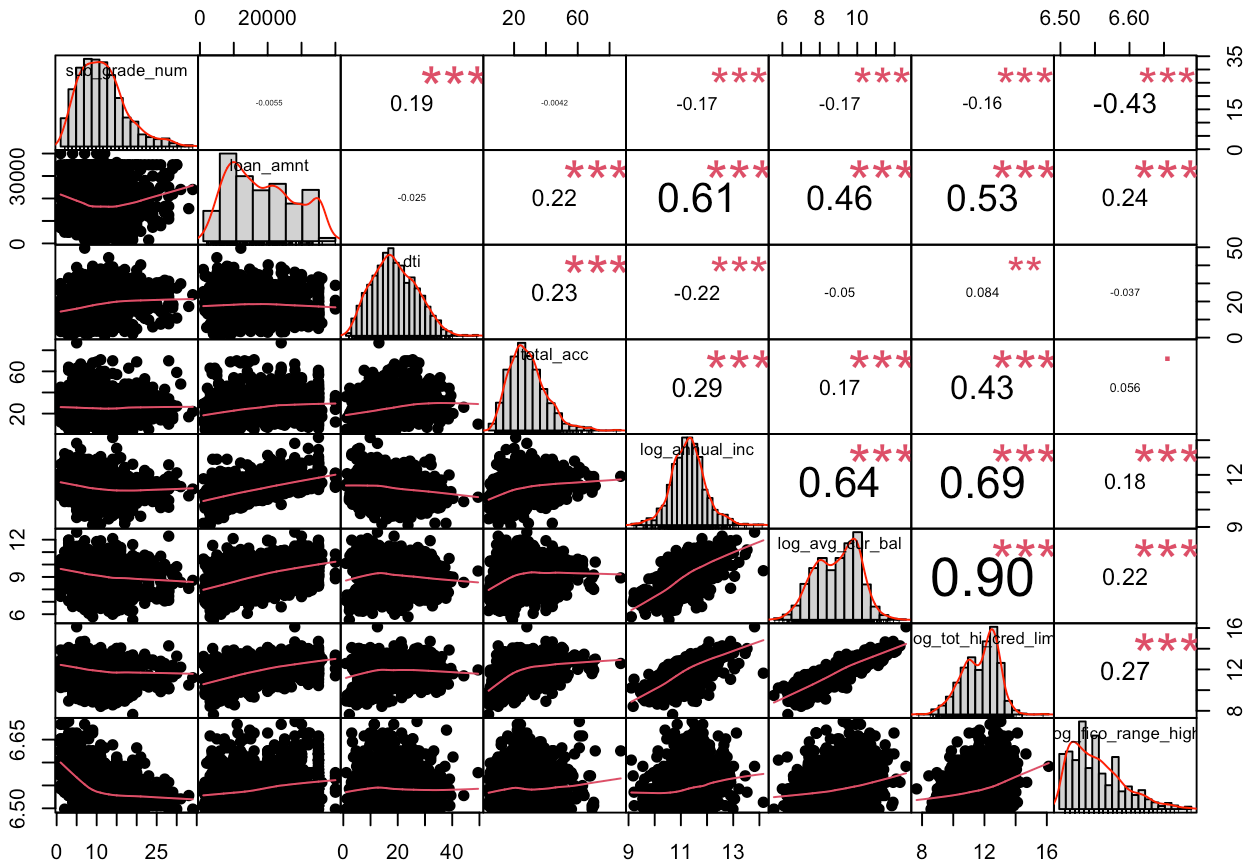
Data Distribution (Transformed Data)

```
credit_trans <- credit %>%
  mutate(log_annual_inc = log(annual_inc),
         log_avg_cur_bal = log(avg_cur_bal+10),
         log_tot_hi_cred_lim = log(tot_hi_cred_lim),
         log_fico_range_high = log(fico_range_high)
         ) %>%
  select(-annual_inc, -avg_cur_bal, -tot_hi_cred_lim, -fico_range_high)
```

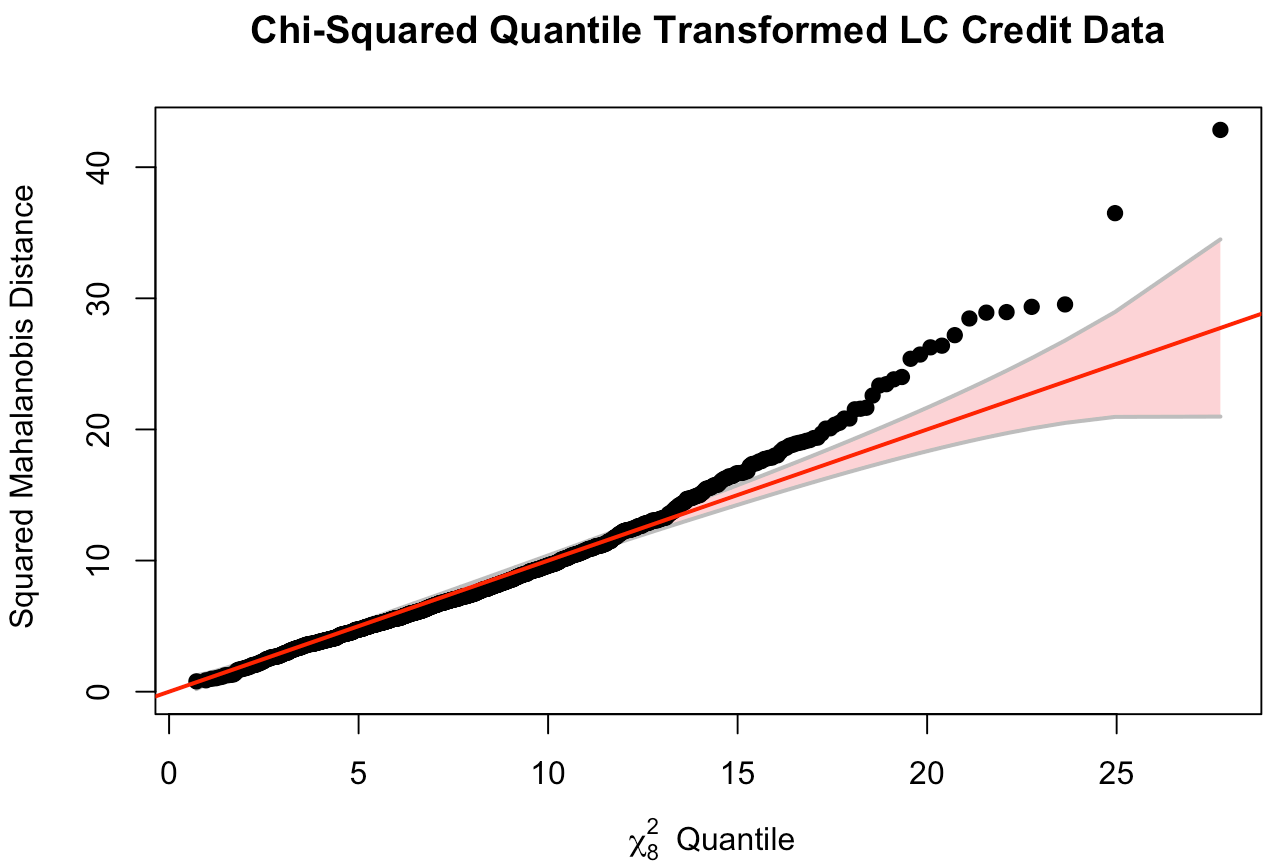
```
corrplot.mixed(cor(credit_trans[, -1]), lower.col = "black", upper = "ellipse",
               tl.col = "black", number.cex=.7, order = "hclust",
               tl.pos = "lt", tl.cex=.7)
```




```
chart.Correlation(credit_trans[, -1], histogram = TRUE, pch = 19)
```




```
cqplot(credit_trans[, -1], main = "Chi-Squared Quantile Transformed LC Credit Dat
```



The correlation between the transformed variables look similar to the raw data. The variables look relatively normally distributed and linear. The chi-squared quantile plot shows that the transformed data is closer to multivariate normality than the raw data, but it still does not look multivariate normal. We will proceed with PCA to further analyze the data.

Analysis of Correlations and PCA Feasability

Raw Data

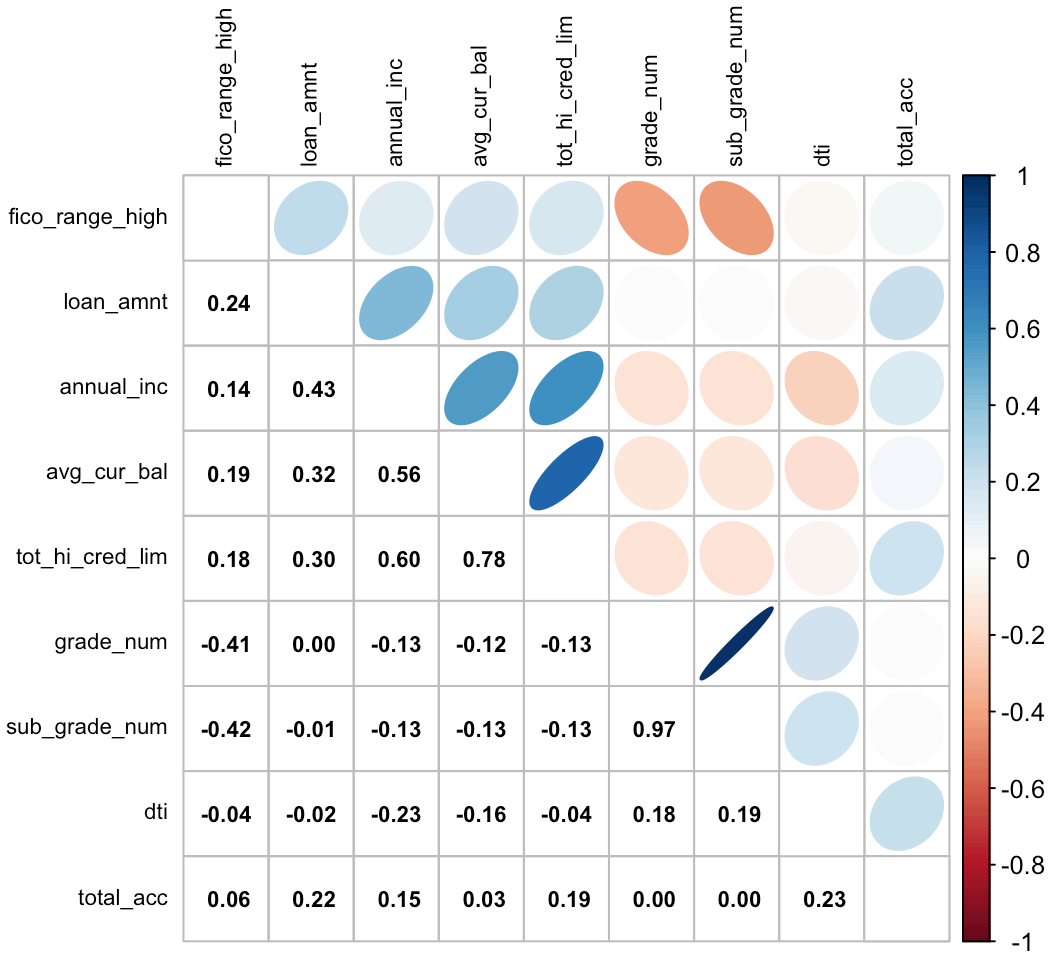
```
round(cor(credit), 2)
```

```
                grade_num sub_grade_num loan_amnt   dti annual_inc
grade_num            1.00          0.97      0.00  0.18      -0.13
sub_grade_num        0.97          1.00     -0.01  0.19      -0.13
loan_amnt            0.00         -0.01      1.00 -0.02       0.43
dti                  0.18          0.19     -0.02  1.00      -0.23
annual_inc          -0.13         -0.13      0.43 -0.23       1.00
fico_range_high     -0.41         -0.42      0.24 -0.04       0.14
total_acc            0.00          0.00      0.22  0.23       0.15
avg_cur_bal         -0.12         -0.13      0.32 -0.16       0.56
tot_hi_cred_lim     -0.13         -0.13      0.30 -0.04       0.60
                fico_range_high total_acc avg_cur_bal tot_hi_cred_lim
```

```
grade_num            -0.41      0.00       -0.12          -0.13
sub_grade_num        -0.42      0.00       -0.13          -0.13
loan_amnt             0.24      0.22        0.32           0.30
dti                  -0.04      0.23       -0.16          -0.04
annual_inc            0.14      0.15        0.56           0.60
fico_range_high       1.00      0.06        0.19           0.18
total_acc             0.06      1.00        0.03           0.19
avg_cur_bal           0.19      0.03        1.00           0.78
tot_hi_cred_lim       0.18      0.19        0.78           1.00
```

```r
# corrplot(cor(credit), method = "ellipse")

corrplot.mixed(cor(credit), lower.col = "black", upper = "ellipse",
               tl.col = "black", number.cex = .7, order = "hclust",
               tl.pos = "lt", tl.cex = .7)
```




```r
dim(credit)
```

```
[1] 949   9
```

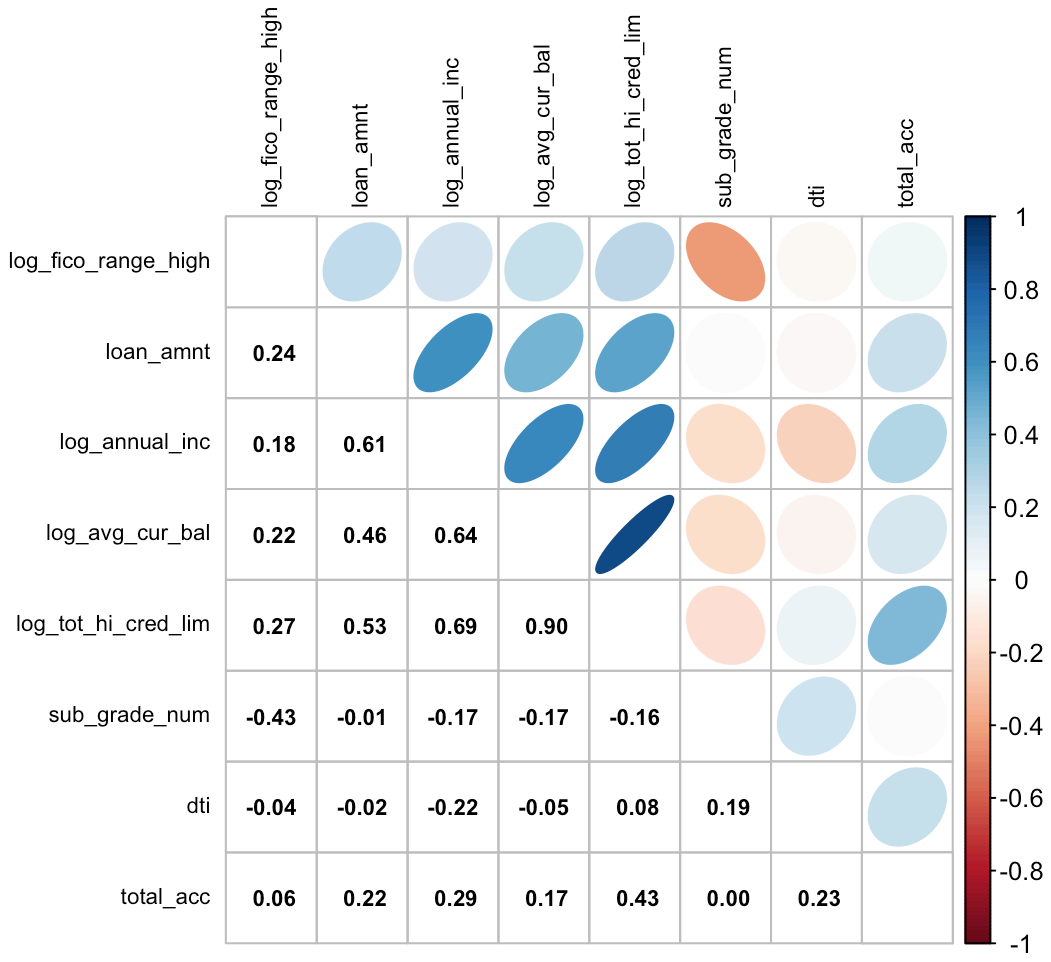
Transformed Data

```
        round(cor(credit_trans[, -1]), 2)
```

```
                   sub_grade_num loan_amnt   dti total_acc log_annual_inc
sub_grade_num               1.00     -0.01  0.19      0.00          -0.17
loan_amnt                  -0.01      1.00 -0.02      0.22           0.61
dti                         0.19     -0.02  1.00      0.23          -0.22
total_acc                   0.00      0.22  0.23      1.00           0.29
log_annual_inc             -0.17      0.61 -0.22      0.29           1.00
log_avg_cur_bal            -0.17      0.46 -0.05      0.17           0.64
log_tot_hi_cred_lim        -0.16      0.53  0.08      0.43           0.69
log_fico_range_high        -0.43      0.24 -0.04      0.06           0.18
                   log_avg_cur_bal log_tot_hi_cred_lim log_fico_range_high
sub_grade_num                -0.17               -0.16               -0.43
loan_amnt                     0.46                0.53                0.24
dti                          -0.05                0.08               -0.04
total_acc                     0.17                0.43                0.06
log_annual_inc                0.64                0.69                0.18
log_avg_cur_bal               1.00                0.90                0.22
log_tot_hi_cred_lim           0.90                1.00                0.27
log_fico_range_high           0.22                0.27                1.00
```

```
        # corrplot(cor(credit), method = "ellipse")

        corrplot.mixed(cor(credit_trans[, -1]), lower.col = "black", upper = "ellipse",
                       tl.col = "black", number.cex = .7, order = "hclust",
                       tl.pos = "lt", tl.cex = .7)
```



We observe some strong correlations, but there are some variables that are not highly correlated with others. PCA works well with strongly correlated variables to reduce dimensionality and we believe that it will be able to work well with this data. We will proceed with PCA to further analyze the data. We have a data set of a sample size of 949 observations with 8 variables. PCA needs enough observations relative to the dimensionality. The data will work well as there is $N \approx 120p > 10p$.

From the correlation matrix, we observe several strong correlations, such as `annual_inc` with `tot_hi_cred_lim` (0.60) and `avg_cur_bal` with `tot_hi_cred_lim` (0.78). These strong relationships suggest that PCA can effectively capture variance in the data by reducing redundancy among highly correlated variables.

However, some variables, like `sub_grade_num` and `total_acc`, exhibit weak correlations with most other features, which may limit their contribution to principal components. Since PCA performs best when variables are strongly correlated, the effectiveness of dimensionality reduction in this dataset will largely depend on how much variance is explained by the first few principal components.

The correlation plot of the transformed data looks similar to the raw data, so we will proceed.

Our dataset consists of 949 observations and 8 variables, which meets the general guideline that PCA requires a sufficient sample size relative to the number of variables. A common rule of thumb suggests $N \approx 120p > 10p$.

Given these factors, we believe PCA will be a useful technique for identifying dominant patterns in the data and reducing dimensionality while retaining essential information.

Principal Component Analysis

Helpful Functions (provided by JDRS)

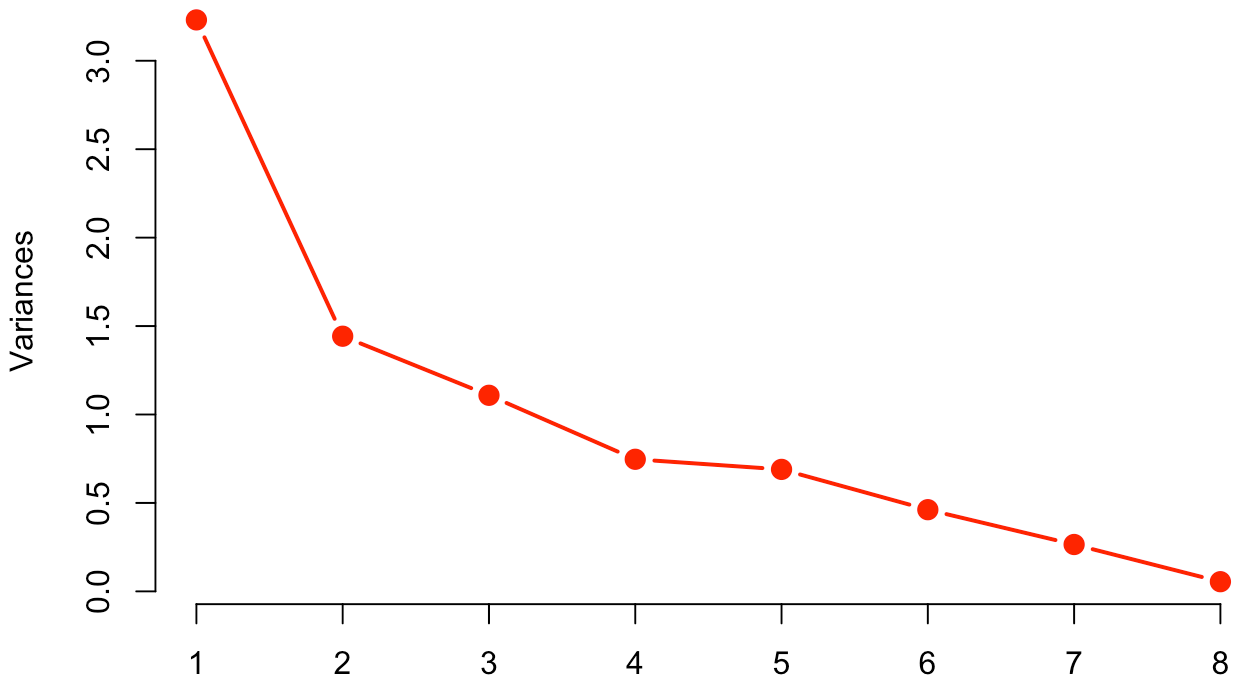
Imported `parallel`, `parallelplot`, and `ciscoreplot` functions from JDRS's R script.

```
summary.PCA.JDRS <- function(x){
  sum_JDRS <- summary(x)$importance
  sum_JDRS[1, ] <- sum_JDRS[1, ]^2
  attr(sum_JDRS, "dimnames")[[1]][1] <- "Eigenvals (Variance)"
  sum_JDRS
}

credit_trans_pca <- prcomp(credit_trans[, -1], scale. = T)
round(summary.PCA.JDRS(credit_trans_pca), 3)
```

```
                        PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
Eigenvals (Variance)   3.231 1.443 1.109 0.747 0.690 0.462 0.265 0.055
Proportion of Variance 0.404 0.180 0.139 0.093 0.086 0.058 0.033 0.007
Cumulative Proportion  0.404 0.584 0.723 0.816 0.902 0.960 0.993 1.000
```

```
screeplot(credit_trans_pca, type = "lines", col = "red", lwd = 2, pch = 19, cex =
          main = "Scree Plot of Transformed Credit Data")
```



To determine the number of principal components to retain, we considered several criteria, including total variance explained, the eigenvalue > 1 rule, and the scree plot elbow method. When examining total variance explained, we set a threshold of 80%. We found that retaining four components would capture 80% of the variance, ensuring that a significant portion of the dataset's variability is preserved while reducing dimensionality. The eigenvalue > 1 criterion, which suggests keeping components with eigenvalues greater than 1, indicated that three components should be retained. Similarly, the scree plot elbow method pointed to either one or three components, depending on where the eigenvalues begin to level off. Since our data is not multivariate normal, we opted not to use parallel analysis, as this method relies on assumptions that are not held in our dataset. Based on these considerations, we decided to retain three principal components, as this choice aligns with both the eigenvalue > 1 rule and the scree plot elbow method, providing a balance between variance retention and dimensionality reduction.

Principal Components

```
        # Obtain loadings
        round(credit_trans_pca$rotation, 3)
```

```
                    PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
sub_grade_num     0.157  0.583 -0.398  0.298  0.228 -0.566  0.126  0.006
loan_amnt        -0.396  0.093 -0.180  0.315  0.591  0.423 -0.417  0.017
dti               0.035  0.539  0.573  0.395 -0.178  0.355  0.248  0.072
```

```
total_acc           -0.238  0.405  0.353 -0.699  0.279 -0.182 -0.145  0.184
log_annual_inc      -0.467 -0.026 -0.278 -0.146  0.111  0.184  0.794  0.072
log_avg_cur_bal     -0.477  0.020 -0.125  0.170 -0.504 -0.184 -0.236  0.620
log_tot_hi_cred_lim -0.513  0.146  0.036  0.023 -0.317 -0.169 -0.128 -0.754
log_fico_range_high -0.227 -0.417  0.512  0.343  0.350 -0.496  0.157  0.045
```

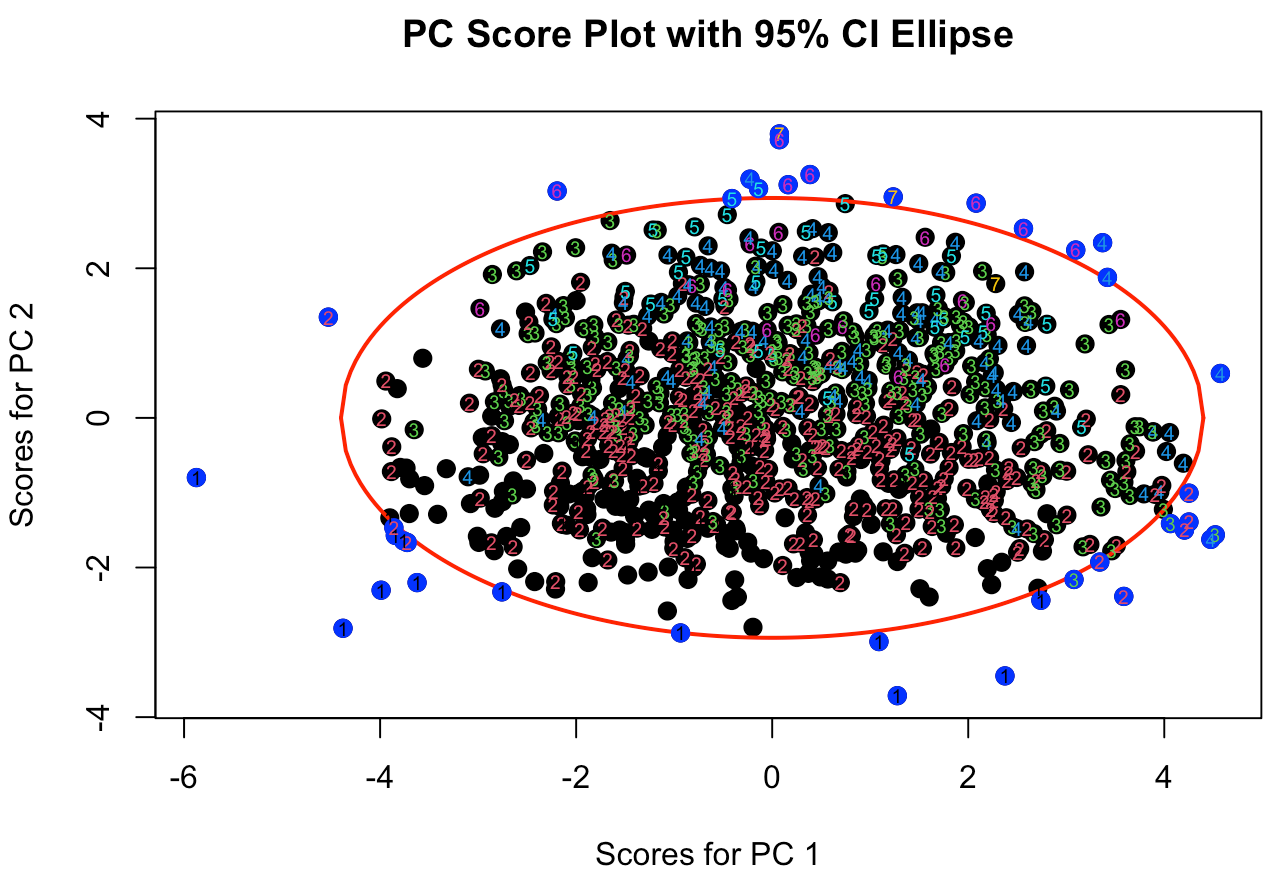
Interpretations of Principal Components

- `PC1` (Financial Strength & Credit Capacity): The first principal component has high negative loadings on `log_annual_inc`, `log_avg_cur_bal`, and `log_tot_hi_cred_lim`, indicating that it primarily captures a borrower's overall financial strength and credit availability. Borrowers with higher values in this component tend to have higher income, larger financial reserves, and greater total credit limits, reflecting stronger financial stability and borrowing capacity.

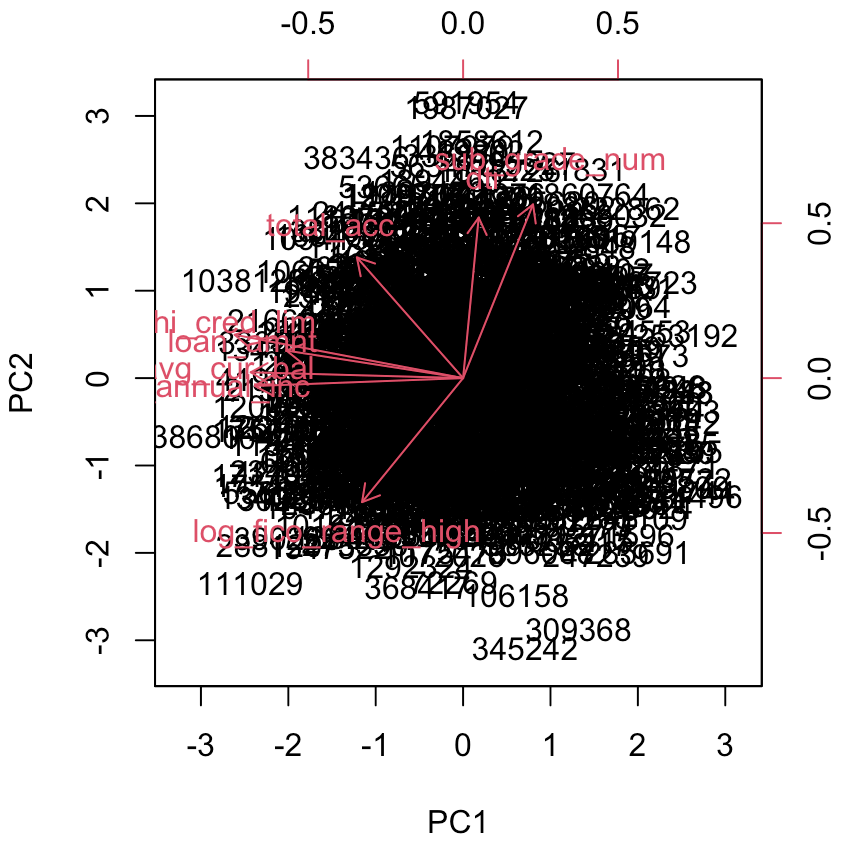
- `PC2` (Credit Risk & Debt Burden): The second principal component has high positive loadings on `sub_grade_num`, `dti`, and `total_acc`, along with a high negative loading on `log_fico_range_high.` This component represents credit risk and debt burden, where higher values indicate lower FICO scores, riskier loan sub-grades, higher debt-to-income (DTI) ratios, and a greater number of total credit accounts. Borrowers scoring high on this component are likely considered higher risk by lenders.

- `PC3` (High Debt, Strong Credit Score): The third principal component has high positive loadings on `dti` and `log_fico_range_high`, suggesting that it captures borrowers who carry high debt loads but still maintain strong credit scores. This could reflect high-income borrowers who strategically leverage credit or individuals with a well-managed but substantial amount of debt.

```r
ciscoreplot(credit_trans_pca, c(1,2), NULL)
text(credit_trans_pca$x[, 1], credit_trans_pca$x[, 2], labels = credit_trans$grad
```




```
biplot(credit_trans_pca, choices = c(1, 2), pc.biplot = T)
```



The score plot visualizes the distribution of observations along the first two principal components (`PC1` and `PC2`), with loan grades labeled (A:1, B:2, ..., G:7). While the plot provides an overview of how observations are spread, no distinct clusters or groupings are apparent.

A 95% confidence interval (CI) ellipse was added to two of the retained components to assess potential outliers. The dataset contains a significant number of outliers (~40), but given that we have close to 1,000 observations, this is not alarming. However, as PCA assumes multivariate normality, which our dataset does not strictly follow, the 95% CI ellipse may not be a reliable method for detecting outliers.

The biplot further enhances interpretability by displaying the loadings of the original variables on `PC1` and `PC2`, offering insights into their contributions. We observe that `log_annual_inc`, `log_avg_cur_bal`, `log_tot_hi_cred_lim`, and `loan_amnt` are strongly aligned with `PC1`, indicating that this component primarily reflects financial strength and credit availability. Meanwhile, `sub_grade_num` and `dti` are more closely associated with `PC2`, reinforcing its interpretation as a credit risk and debt burden component. Additionally, `log_fico_range_high` and `total_acc` contribute to both `PC1` and `PC2`, suggesting that they play a role in both financial stability and borrowing behavior.

While the score plot does not reveal clear groupings, it provides insight into the data's spread and the presence of outliers. The 95% CI ellipse is not a reliable outlier detection method due to the dataset's non-normality. The biplot confirms variable associations, helping to better understand the relationship between borrower attributes and the principal components.

Summary of Findings and PCA Effectiveness

PCA effectively reduced the eight variables into three principal components, capturing 72% of the total variance and providing meaningful insights into borrower financial profiles. The transformed data exhibited linear relationships, making PCA a suitable method for dimensionality reduction. The principal components reflected financial strength (`PC1`), credit risk (`PC2`), and high debt with strong credit scores (`PC3`). While the score plot did not reveal clear groupings, it provided insight into the spread of observations, and the bi-plot helped visualize the interactions between the principal components. Overall, PCA successfully summarized key financial patterns, demonstrating its effectiveness in understanding the relationships between borrower attributes.