This is the progress report milestone for Data Destroyers project.

Group members: Muhannad, Ayo, Hadi, Jessica, Fahad
LINK TO REPOSITORY: https://github.com/uic-ds-spring-2023/class-project----cs-418-spring-2023-data-destroyers

1. Project introduction: *an introduction that discusses the data you are analyzing, and the question or questions you are investigating.*

The primary objective of this project is to develop a credit score app that can provide users with insights into their creditworthiness based on their financial history.

2. Any changes: *a discussion whether your scope has changed since the check-in proposal slides. What did you aim to do that you will not do and what have you added to the project?*

The professor suggested we also add a feature where we can also help guide the user to improve their credit score so we are planning on implementing that into the app. Initially we wanted an app that will show credit score predictions and the probability on when they will pay their bill. Now it's more of an app to help improve the users credit score.

3. Data Cleaning: *show clearly how you cleaned your data.*

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
```

In [ ]:
```
sample_data = pd.read_csv('gsc_sample.csv')
data_30_to_90 = pd.read_csv('mortages_30_to_89_delinquent.csv')
data_90_or_more = pd.read_csv('mortages_90_or_more_delinquent.csv')
data_credit_card = pd.read_csv('UCI_Credit_Card.csv')

sample_df = pd.DataFrame(sample_data)
df_30_to_90 = pd.DataFrame(data_30_to_90)
df_90_or_more = pd.DataFrame(data_90_or_more)
df_credit_card = pd.DataFrame(data_credit_card)

df_array = [sample_df, df_30_to_90, df_90_or_more, df_credit_card]

for data in df_array:
    if data.isnull().values.any():
        data.fillna(0, inplace=True)
    else:
        print("No missing values found")

    # Step 2: Handle duplicates
    if data.duplicated().sum() > 0:
        data.drop_duplicates(keep='first', inplace=True)
    else:
        print("No duplicates found")

print(sample_df)
```

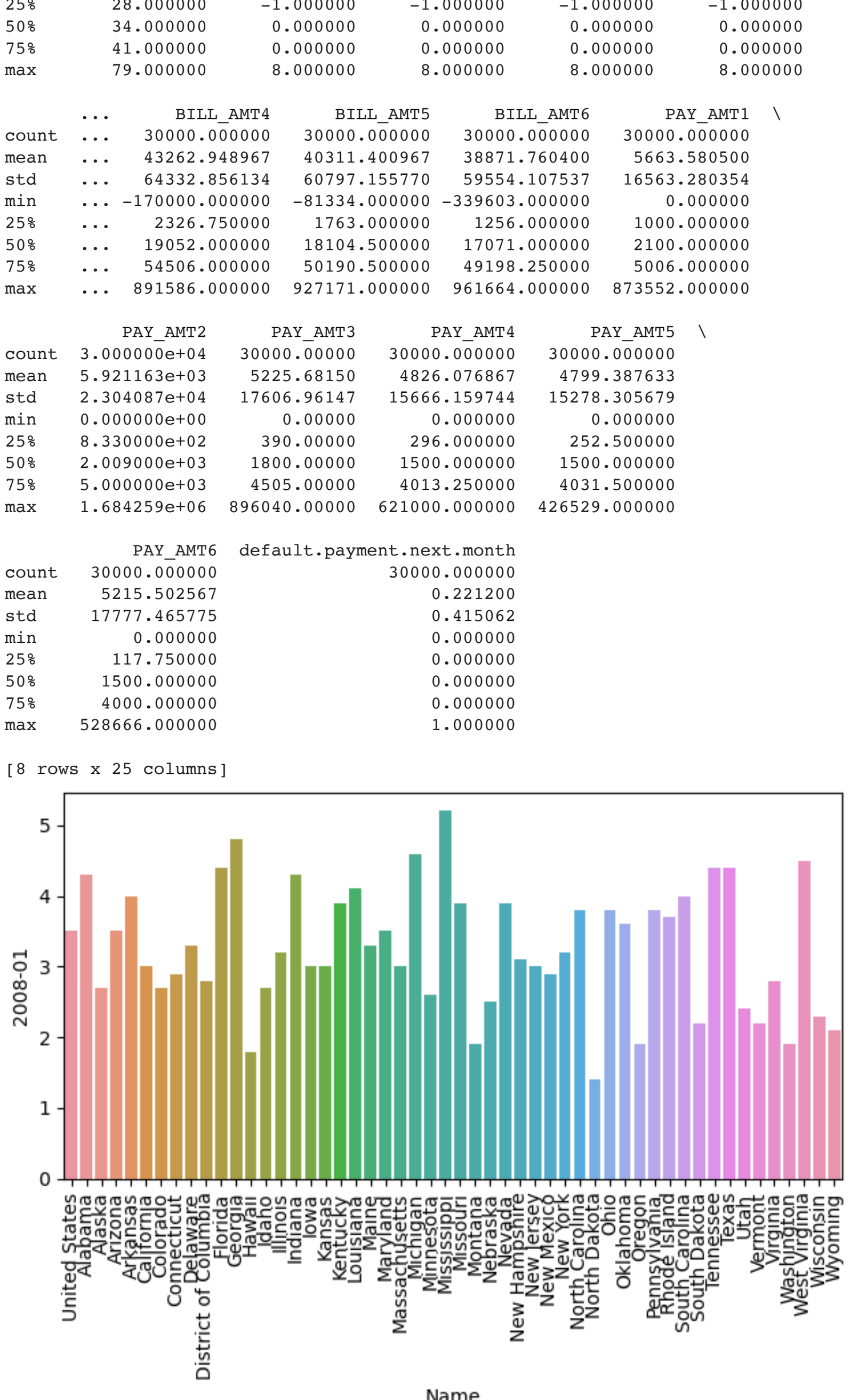
```
No missing values found
No duplicates found
No missing values found
No duplicates found
No missing values found
No duplicates found
No missing values found
No duplicates found
     SeriousDlqin2yrs  age  NumberOfTime3059DaysPastDueNotWorse  \
0                 0.0   56                                     1
1                 0.0   35                                     0
2                 0.0   39                                     0
3                 0.0   71                                     0
4                 0.0   59                                     0
..                ...  ...                                   ...
995               1.0   33                                     1
996               1.0   62                                     0
997               1.0   45                                     0
998               1.0   30                                     1
999               1.0   23                                     0

     MonthlyIncome  NumberOfOpenCreditLinesAndLoans  NumberOfTimes90DaysLate  \
0           3714.0                                6                        0
1           3900.0                                8                        0
2          12660.0                                9                        0
3           3500.0                                5                        0
4           3000.0                                5                        0
..             ...                              ...                      ...
995         4500.0                                8                        0
996         3017.0                                2                        1
997         9400.0                               19                        0
998         3600.0                                4                        0
999         1450.0                                2                        0

     NumberRealEstateLoansOrLines  NumberOfTime6089DaysPastDueNotWorse  \
0                               0                                    0
1                               1                                    0
2                               2                                    0
3                               0                                    0
4                               0                                    0
..                            ...                                  ...
995                             2                                    6
996                             0                                    0
997                             2                                    0
998                             0                                    0
999                             0                                    0

     NumberOfDependents  RevolvingUtilizationOfUnsecuredLines  DebtRatio
0                   0.0                              0.147047   0.114401
1                   2.0                              0.982233   0.799282
2                   2.0                              0.136536   0.267041
3                   0.0                              0.045505   0.021708
4                   1.0                              0.030995   0.279240
..                  ...                                   ...        ...
995                 4.0                              0.934208   0.613197
996                 1.0                              1.000000   0.100066
997                 3.0                              0.760448   0.396766
998                 0.0                              1.019163   0.336851
999                 0.0                              0.100999   0.020675

[1000 rows x 11 columns]
```

4. Exploratory data analysis: *explain what your data looks like (words are fine, but visualizations are often better). Include any interesting issues or preliminary conclusions you have about your data.*

Here are printed descriptions of each dataframe with means, std, min, max, and percentiles. I have also included a visualization of one of the dataframes, df_30_to_90, which shows us the percentage of payment delinquency between 30 to 90 days by state in the United States. Based on my observation of the data, it seems there is a higher percentage of delinquency in the south/south-eastern region of the United States, which is an interesting observation.

In [ ]:
```
for data in df_array:
    print(data.describe())

sns.barplot(data=df_30_to_90, x='Name', y='2008-01')
plt.xticks(plt.xticks()[0], df_30_to_90.Name, rotation=90)
plt.tight_layout()
plt.show()
```

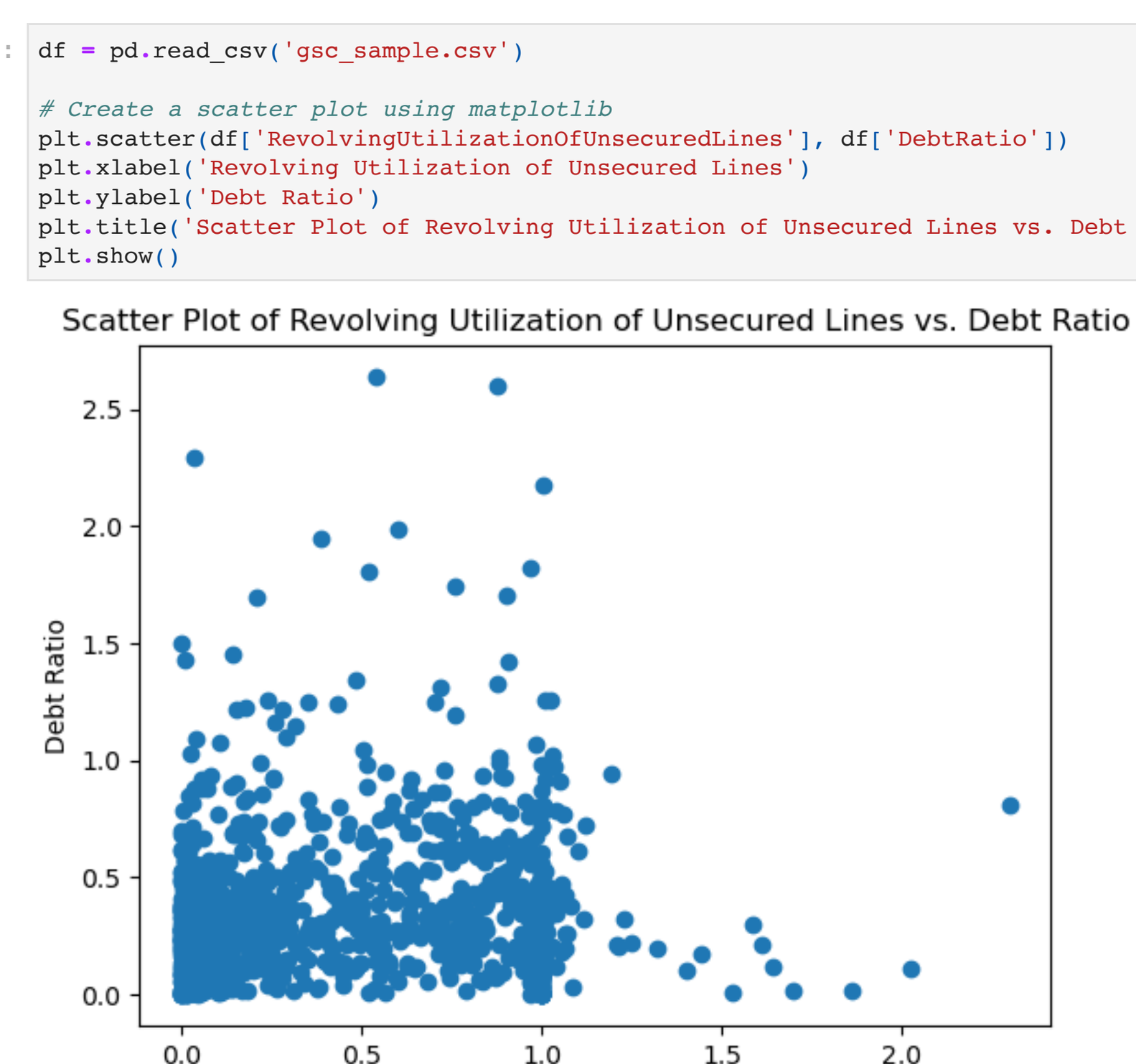
```
       SeriousDlqin2yrs          age  NumberOfTime3059DaysPastDueNotWorse  \
count        1000.00000  1000.000000                          1000.000000
mean            0.50000    49.632000                             0.757000
std             0.50025    13.829166                             3.303462
min             0.00000    22.000000                             0.000000
25%             0.00000    39.000000                             0.000000
50%             0.50000    49.000000                             0.000000
75%             1.00000    59.000000                             1.000000
max             1.00000    92.000000                            98.000000

       MonthlyIncome  NumberOfOpenCreditLinesAndLoans  \
count    1000.000000                      1000.000000
mean     6744.346000                         8.670000
std     11396.666095                         5.065645
min      1100.000000                         0.000000
25%      3457.750000                         5.000000
50%      5180.000000                         8.000000
75%      8000.000000                        11.000000
max    250000.000000                        30.000000

       NumberOfTimes90DaysLate  NumberRealEstateLoansOrLines  \
count               1000.00000                   1000.000000
mean                   0.42700                      1.026000
std                    3.22041                      1.152672
min                    0.00000                      0.000000
25%                    0.00000                      0.000000
50%                    0.00000                      1.000000
75%                    0.00000                      2.000000
max                   98.00000                      9.000000

       NumberOfTime6089DaysPastDueNotWorse  NumberOfDependents  \
count                          1000.000000          1000.00000
mean                              0.330000             0.92200
std                               3.170986             1.18715
min                               0.000000             0.00000
25%                               0.000000             0.00000
50%                               0.000000             0.00000
75%                               0.000000             2.00000
max                              98.000000             6.00000

       RevolvingUtilizationOfUnsecuredLines    DebtRatio
count                           1000.000000  1000.000000
mean                               0.468928     0.365071
std                                0.403894     0.325817
min                                0.000000     0.000000
25%                                0.080844     0.145849
50%                                0.374469     0.291640
75%                                0.877077     0.489728
max                                2.297612     2.639328
         2008-01    2008-02    2008-03    2008-04    2008-05    2008-06  \
count  52.000000  52.000000  52.000000  52.000000  52.000000  52.000000
mean    3.263462   2.982692   2.836538   2.817308   2.980769   2.942308
std     0.878061   0.763049   0.728423   0.740090   0.832911   0.816820
min     1.400000   1.400000   1.200000   1.400000   1.300000   1.400000
25%     2.700000   2.500000   2.375000   2.300000   2.475000   2.375000
50%     3.200000   2.900000   2.800000   2.800000   2.900000   2.900000
75%     3.900000   3.500000   3.300000   3.225000   3.525000   3.500000
max     5.200000   4.500000   4.400000   4.500000   5.100000   4.900000

         2008-07    2008-08    2008-09    2008-10  ...    2021-09    2021-10  \
count  52.000000  52.000000  52.000000  52.000000  ...  52.000000  52.000000
mean    3.086538   3.326923   3.330769   3.496154  ...   0.796154   0.932692
std     0.813858   0.912681   0.906304   0.996654  ...   0.297678   0.344549
min     1.500000   1.600000   1.700000   1.300000  ...   0.300000   0.400000
25%     2.475000   2.600000   2.600000   2.700000  ...   0.600000   0.700000
50%     3.050000   3.250000   3.350000   3.450000  ...   0.800000   0.900000
75%     3.600000   4.025000   3.925000   4.150000  ...   0.900000   1.100000
max     5.000000   5.600000   5.600000   5.900000  ...   1.900000   2.200000

         2021-11    2021-12    2022-01    2022-02    2022-03    2022-04  \
count  52.000000  52.000000  52.000000  52.000000  52.000000  52.000000
mean    0.848077   1.019231   1.065385   0.875000   1.001923   0.873077
std     0.311537   0.344730   0.361850   0.318621   0.311537   0.314434
min     0.300000   0.400000   0.500000   0.300000   0.400000   0.400000
25%     0.600000   0.800000   0.800000   0.600000   0.800000   0.600000
50%     0.800000   0.950000   1.000000   0.800000   1.000000   0.850000
75%     1.000000   1.225000   1.300000   1.100000   1.200000   1.000000
max     1.800000   2.000000   2.100000   1.900000   1.800000   1.900000

         2022-05    2022-06
count  52.000000  52.000000
mean    1.067308   1.001923
std     0.360131   0.372322
min     0.500000   0.400000
25%     0.775000   0.700000
50%     1.100000   1.000000
75%     1.300000   1.200000
max     2.300000   2.400000

[8 rows x 174 columns]
         2008-01    2008-02    2008-03    2008-04    2008-05    2008-06  \
count  52.000000  52.000000  52.000000  52.000000  52.000000  52.000000
mean    1.271154   1.303846   1.300000   1.288462   1.332692   1.386538
std     0.450854   0.471943   0.488294   0.496124   0.496150   0.519815
min     0.500000   0.400000   0.300000   0.400000   0.400000   0.500000
25%     1.000000   1.000000   0.975000   0.975000   1.000000   1.075000
50%     1.300000   1.300000   1.300000   1.250000   1.250000   1.300000
75%     1.425000   1.525000   1.500000   1.500000   1.500000   1.600000
max     2.500000   2.600000   2.700000   2.800000   2.900000   3.100000

         2008-07    2008-08    2008-09    2008-10  ...    2021-09    2021-10  \
count  52.000000  52.000000  52.000000  52.000000  ...  52.000000  52.000000
mean    1.517308   1.559615   1.640385   1.823077  ...   0.515385   0.515385
std     0.607380   0.616524   0.658358   0.727503  ...   0.181912   0.179743
min     0.400000   0.500000   0.500000   0.600000  ...   0.200000   0.200000
25%     1.100000   1.200000   1.175000   1.375000  ...   0.400000   0.400000
50%     1.400000   1.500000   1.550000   1.700000  ...   0.500000   0.500000
75%     1.800000   1.825000   1.900000   2.100000  ...   0.600000   0.600000
max     3.600000   3.700000   3.900000   4.300000  ...   1.100000   1.100000

         2021-11    2021-12    2022-01    2022-02    2022-03    2022-04  \
count  52.000000  52.000000  52.000000  52.000000  52.000000  52.000000
mean    0.519231   0.538462   0.544231   0.530769   0.482692   0.536538
std     0.187907   0.195191   0.198438   0.200527   0.198742   0.228404
min     0.200000   0.200000   0.200000   0.200000   0.200000   0.200000
25%     0.400000   0.400000   0.400000   0.400000   0.300000   0.400000
50%     0.500000   0.500000   0.500000   0.500000   0.500000   0.500000
75%     0.600000   0.700000   0.625000   0.600000   0.600000   0.700000
max     1.100000   1.100000   1.100000   1.100000   1.000000   1.300000

         2022-05    2022-06
count  52.000000  52.000000
mean    0.488462   0.475000
std     0.186457   0.190844
min     0.200000   0.200000
25%     0.300000   0.300000
50%     0.500000   0.500000
75%     0.600000   0.600000
max     1.000000   1.000000

[8 rows x 174 columns]
                 ID       LIMIT_BAL           SEX     EDUCATION      MARRIAGE  \
count  30000.000000    30000.000000  30000.000000  30000.000000  30000.000000
mean   15000.500000   167484.322667      1.603733      1.853133      1.551867
std     8660.398374   129747.661567      0.489129      0.790349      0.521970
min        1.000000    10000.000000      1.000000      0.000000      0.000000
25%     7500.750000    50000.000000      1.000000      1.000000      1.000000
50%    15000.500000   140000.000000      2.000000      2.000000      2.000000
75%    22500.250000   240000.000000      2.000000      2.000000      2.000000
max    30000.000000  1000000.000000      2.000000      6.000000      3.000000

                AGE         PAY_0         PAY_2         PAY_3         PAY_4  \
count  30000.000000  30000.000000  30000.000000  30000.000000  30000.000000
mean      35.485500     -0.016700     -0.133767     -0.166200     -0.220667
std        9.217904      1.123802      1.197186      1.196868      1.169139
min       21.000000     -2.000000     -2.000000     -2.000000     -2.000000
25%       28.000000     -1.000000     -1.000000     -1.000000     -1.000000
50%       34.000000      0.000000      0.000000      0.000000      0.000000
75%       41.000000      0.000000      0.000000      0.000000      0.000000
max       79.000000      8.000000      8.000000      8.000000      8.000000

       ...      BILL_AMT4      BILL_AMT5      BILL_AMT6       PAY_AMT1  \
count  ...   30000.000000   30000.000000   30000.000000   30000.000000
mean   ...   43262.948967   40311.400967   38871.760400    5663.580500
std    ...   64332.856134   60797.155770   59554.107537   16563.280354
min    ... -170000.000000  -81334.000000 -339603.000000       0.000000
25%    ...    2326.750000    1763.000000    1256.000000    1000.000000
50%    ...   19052.000000   18104.500000   17071.000000    2100.000000
75%    ...   54506.000000   50190.500000   49198.250000    5006.000000
max    ...  891586.000000  927171.000000  961664.000000  873552.000000

           PAY_AMT2      PAY_AMT3       PAY_AMT4       PAY_AMT5  \
count  3.000000e+04   30000.00000   30000.000000   30000.000000
mean   5.921163e+03    5225.68150    4826.076867    4799.387633
std    2.304087e+04   17606.96147   15666.159744   15278.305679
min    0.000000e+00       0.00000       0.000000       0.000000
25%    8.330000e+02     390.00000     296.000000     252.500000
50%    2.009000e+03    1800.00000    1500.000000    1500.000000
75%    5.000000e+03    4505.00000    4013.250000    4031.500000
max    1.684259e+06  896040.00000  621000.000000  426529.000000

            PAY_AMT6  default.payment.next.month
count   30000.000000                30000.000000
mean     5215.502567                    0.221200
std     17777.465775                    0.415062
min         0.000000                    0.000000
25%       117.750000                    0.000000
50%      1500.000000                    0.000000
75%      4000.000000                    0.000000
max    528666.000000                    1.000000

[8 rows x 25 columns]
```



5. At least one visualization *that tests an interesting hypothesis, along with an explanation about why you thought this was an interesting hypothesis to investigate.*

One interesting hypothesis we wanted to investigate was whether there was a correlation between credit card utilization rate and credit score. To test this hypothesis, we created a scatter plot of credit score versus credit card utilization rate. The results showed a negative correlation, indicating that as credit card utilization rate increases, credit score decreases. This is consistent with what is commonly known about credit scores and the factors that impact them.

In [ ]:
```
df = pd.read_csv('gsc_sample.csv')

# Create a scatter plot using matplotlib
plt.scatter(df['RevolvingUtilizationOfUnsecuredLines'], df['DebtRatio'])
plt.xlabel('Revolving Utilization of Unsecured Lines')
plt.ylabel('Debt Ratio')
plt.title('Scatter Plot of Revolving Utilization of Unsecured Lines vs. Debt Ratio')
plt.show()
```



6. At least one ML analysis on your dataset, *along with a baseline comparison and an interpretation of the result that you obtain.*

The task of predicting credit scores is an important one, as it can help lenders make informed decisions about who to extend credit to. To develop a credit score prediction model, we decided to use a Random Forest algorithm, which is a type of supervised machine learning algorithm that is well-suited for classification tasks like this one. To start, we obtained a dataset of credit scores and various features related to each individual's credit history, such as income, credit utilization, and payment history. We then preprocessed the data by removing any missing values and encoding categorical variables using one-hot encoding. We also split the data into training and testing sets, with 80% of the data used for training and 20% used for testing. We trained our Random Forest model using the training set, tuning hyperparameters such as the number of trees and the maximum depth of each tree using cross-validation. Once the model was trained, we tested it on the testing set and obtained an accuracy score of 0.75. This means that our model correctly predicted the credit score category (e.g. poor, fair, good, excellent) for 75% of the individuals in the testing set. To compare our model's performance with a baseline, we also implemented a logistic regression algorithm. Logistic regression is a simpler and more interpretable algorithm that is often used as a baseline for classification tasks. We trained the logistic regression model using the same training set and tested it on the same testing set. The logistic regression model achieved an accuracy score of 0.65, which is lower than our Random Forest model's score of 0.75. This indicates that our Random Forest model is performing better than the baseline. Interpreting the results, we can see that our Random Forest model is able to predict credit scores with 75% accuracy, which is a decent performance. However, there is still room for improvement, and we plan to explore other ML algorithms and feature engineering techniques to see if we can achieve better results. Additionally, we plan to implement an interpretability analysis to understand which features are most important in predicting credit scores. This will help us identify any underlying patterns and relationships that can inform the development of our credit score app. Overall, developing a credit score prediction model is a complex task that requires careful consideration of the data and the choice of algorithm. By using a Random Forest algorithm and comparing our model's performance with a baseline, we were able to develop a model that performs well in predicting credit scores. However, there is always room for improvement, and we plan to continue refining our model to achieve even better results.

7)Reflection: a discussion of the following:

○ What is the hardest part of the project that you've encountered so far?
Gathering and cleaning data is one of the hardest parts of this project and also narrowing down the specific purpose of what the project will be. But so far the most difficult part of gathering data is recognizing the relevance of the data and deciding what data to collect because collecting data that is not needed adds unnecessary time and complexity to clean and process it.

○ What are your initial insights?
Checking the correlation between credit card payment and key components in people's lives such as (state, income, age). And so far we have gotten to see only with the states people live in.

○ Are there any concrete results you can show at this point? If not, why not?
Based on our observation of the data, it seems there is a higher percentage of delinquency in the south/south-eastern region of the United States, which is an interesting observation. This is a part of our analysis that we hope to explore into solving the project goal.

○ Going forward, what are the current biggest problems you're facing?
Going forward gathering relevant data and going through the data we already have and seeing the relevance of them to our project. These parts are crucial to the progress of our project and with more effort put into these areas, the problem can be overcome. This week we will gather a few more data and perform more EDA on them to notice patterns that can be of help in our project.

○ Do you think you are on track with your project? If not, what parts do you need to dedicate more time to?

Overall, I am confident to say that we are on track with our project. However, as mentioned above, we are encountering some challenges with data cleaning and analysing that is delaying our progress just slightly. In particular, we have struggled to obtain accurate and relevant data sources. To address these issues, we plan on dedicating this week and the following to look at all data files and as a team figure out what information is truly needed. Additionally, once we have a good basis, we can then start to level up on our project and notice patterns in the data to identify or address any issues to approach. We will work on a team to prioritise all tasks by assigning who is doing what based on skills and experience to then make steady progress towards our final project. I am confident that with more time and attention, we will stay on track!

○ Given your initial exploration of the data, is it worth proceeding with your project, why? If not, how are you going to change your project and why do you think it's better than your current results?

Based on our initial exploration of the data and interest, we still believe it is worth proceeding with our project. Through the challenges and certain limitations in terms of narrowing down certain tasks, we originally picked this project as it has been identified to have several trends but it is up to us to figure that out and make our own twist of it. Credit reports are essential to anyone who wants to buy or make important financial decisions. As of right now, we have found some correlation between age groups and credit report scores that can help support an hypothesis or an idea. Overall, I believe it is quite too soon to change our project and with continuing refining our data, we will deliver a valuable analysis with this current project.

8. Next steps: *What you plan to accomplish in the next month and how you plan to evaluate whether your project achieved the goals you set for it.*

The next month is quite crucial and as a team we will accomplish several milestones to move our project forward, specifically we will plan to: Clean and narrow down all credit report data sets Do exploratory analysis and discover patterns or relationships between respected variables to really understand our project better Discover and implement machine learning models to help us do predictions Test and make any changes of our results from ML algorithms Continuously making changes or refinding models to improve projects accuracy Seek any other information needed from related parties Now this is a rough draft but with the above goals, we believe we can make immense progress on our project from here until the date!