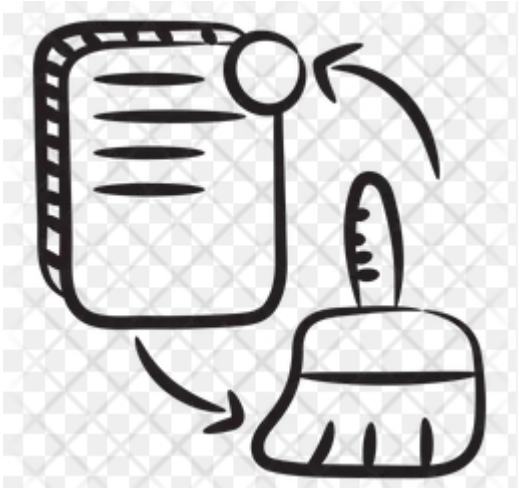
Lending case study



Data Cleaning

Fixing Rows and Columns

Change the Header data

If any multiple rows for a header exist, it can be short formed which are abbreviated and put it in a single row.

Remove summary rows if exist.

Create unique column by merging multiple columns where there is repetive data columns available

Delete unncessary columns

Arrange data in the same data type such as if column has ‘NA’, ‘XXX’ it can be replaced with empty / 0 (if necessary) columns.

```
#Remove NA from the numberic column
df = df.dropna(axis=1)
print(df['annual_inc'])
```

Selecting variables for Defining Defaults

Loan related variables

Experience - Emp_length
Owning the house – home_ownership
Annual Income - annual_inc
purpose – purpose – (It can be hypothicated, appriciable / non appriciable values)
delinq_2yrs - Deliquency
revol_bal – Revolving balance (check whether it is consistent for each cycle)
revol_util - % pf revolving balance
chargeoff_within_12_mths
collection_recovery_fee
Number of mortgage accounts - mort_acc
verification_status
Total number of credit lines – total_acc

..continued

verification_status

Total number of credit lines – total_acc

open_rv_12m – number of revolving trades opened in 12 months

open_rv_24m – number of revolving trades in 24 months

pub_rec - Number of derogatory public records

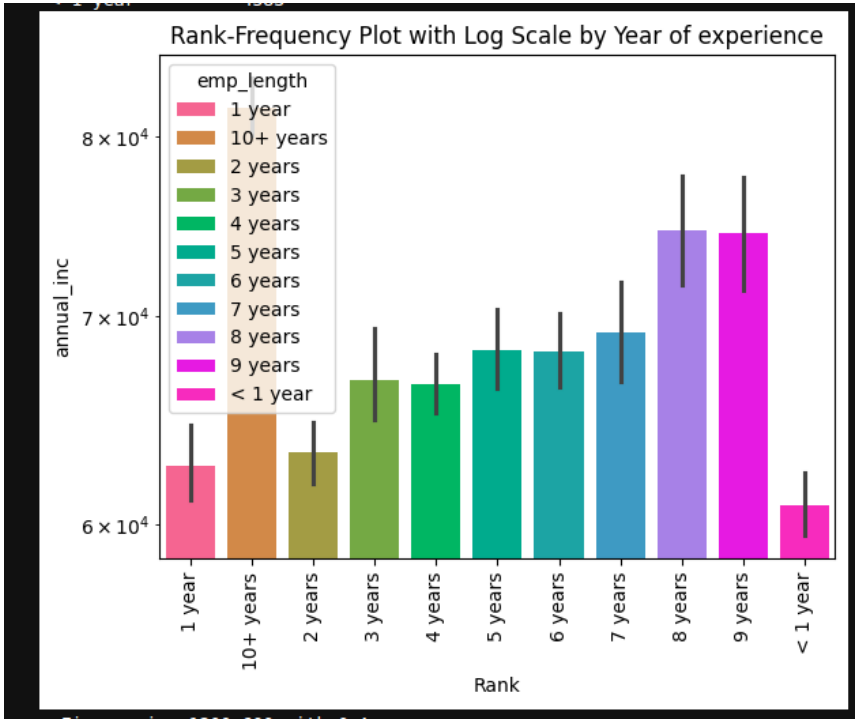
pub_rec_bankruptcies - Number of public record bankruptcies

total_rev_hi_lim - Total revolving high credit/credit limit

#Unordered Variables – Rank frequency Plots with log scale

```
def analysis_charged_off():
    return(df[['tot_cur_bal','tot_coll_amt','chargeoff_within_12_mths']])
def analysis_Current():
    return(df[['tot_cur_bal','tot_coll_amt','chargeoff_within_12_mths','delinq_amnt','inq_last_12m']])
def analysis_Fully_paid():
    return(df[['tot_cur_bal','tot_coll_amt']])
```

Log scale


Rank-Frequency Plot with Log Scale by Year of experience
emp_length
1 year
10+ years
2 years
3 years
4 years
5 years
6 years
7 years
8 years
9 years
< 1 year
annual_inc
8 × 10^4
7 × 10^4
6 × 10^4
1 year
10+ years
2 years
3 years
4 years
5 years
6 years
7 years
8 years
9 years
< 1 year
Rank

Univariate Analysis

Univariate Analysis

Unordered Variables – Rank frequency Plots with log scale

Ordered Variables -

Quantitative Variables – IQR – box plot Median,mode,mean, SD

Segmented Univariate – Grouping data by dimensions -

Quick way of segmentation – Categorical variables in one axis , numerical variables on the other axis

```
print(df[column_name])

sns.set(style="whitegrid")
ax = sns.boxplot(y=df[column_name])

ax.set_title(f'Box Plot for {column_name}')
ax.set_ylabel('Values')

# Calculate and print Interquartile Range (IQR)
Q1 = df[column_name].quantile(0.25)
Q3 = df[column_name].quantile(0.75)
IQR = Q3 - Q1
print("Interquartile Range (IQR):", IQR)

plt.show()
```

..continued

Segmented Univariate – Grouping data by dimensions -
Quick way of segmentation – Categorical variables in one axis , numerical variables on the other axis

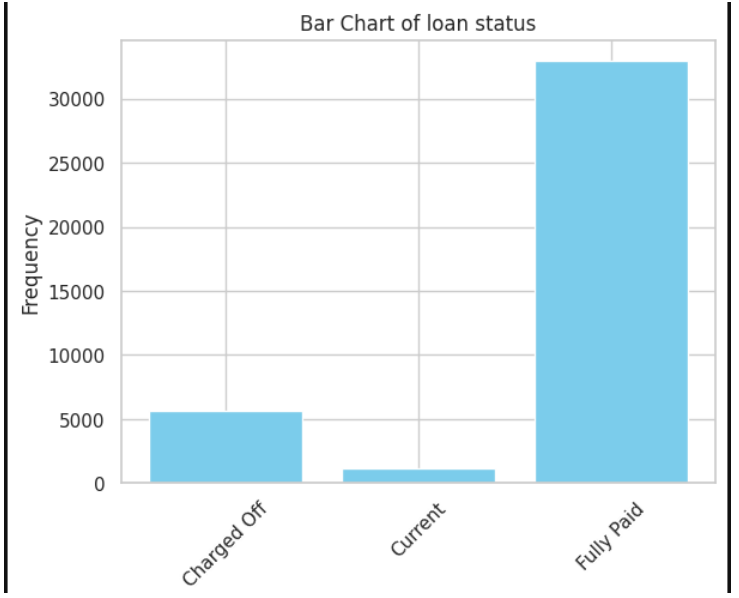
```
column_name = 'loan_status'

loan_status = df[column_name].value_counts()

loan_status.sort_index(inplace=True)

plt.bar(loan_status.index, loan_status.values, color='skyblue')


plt.title(f'Bar Chart of loan status')
plt.xlabel('Values')
plt.ylabel('Frequency')
```



Bivariate Analysis

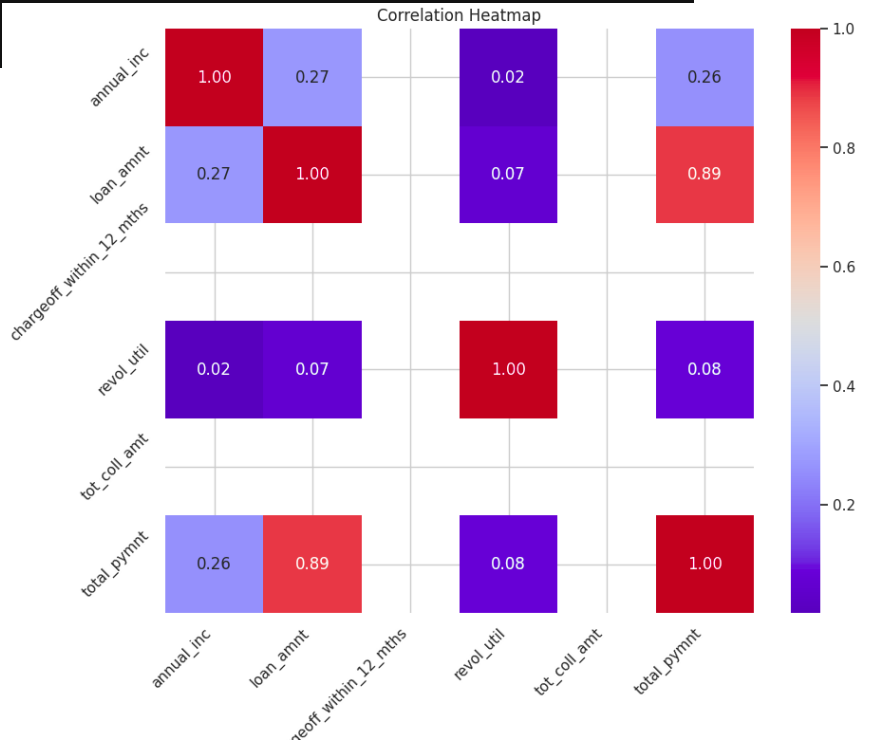
Comprehension – Correlation Matrix
pairs of categorical variables - relationship between categorical and continuous variables.
Distribution of two categorical variables.
annual_inc
loan_amnt
chargeoff_within_12_mths
delinq_amnt

```python
#  correlation matrix for the below columns
df['revol_util'] = df['revol_util'].str.rstrip('%').astype(float) / 100.0
correlation_matrix = df[['annual_inc', 'loan_amnt', 'chargeoff_within_12_mths', 'revol_util','tot_coll_amt','total_pymnt']].corr()

plt.figure(figsize=(10, 8))
heatmap = sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")

# 45 rotate for proper label display
heatmap.set_xticklabels(heatmap.get_xticklabels(), rotation=45, horizontalalignment='right')
heatmap.set_yticklabels(heatmap.get_yticklabels(), rotation=45, horizontalalignment='right')

plt.title('Correlation Heatmap')
plt.show()
```



Derived Metrix

Type-driven metrics

Steven’s typology classifies variables into four types — nominal, ordinal, interval and ratio

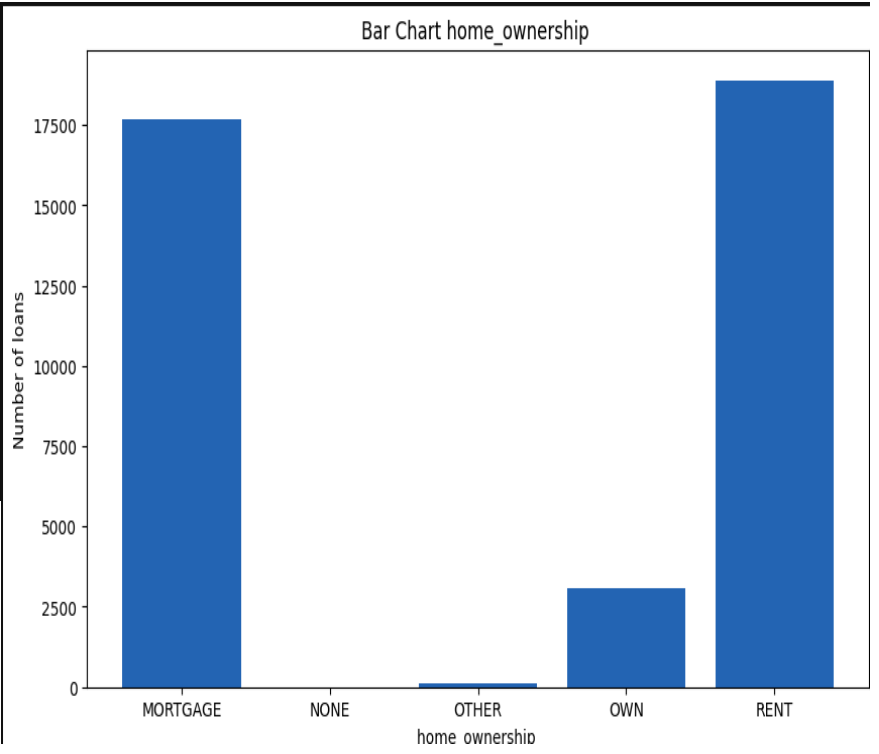
Business-driven metrics
Data-driven metrics – Arriving new data and analysis

```python
filtered_df = df.groupby('home_ownership')['home_ownership'].size().to_frame('count')
filtered_df.reset_index(inplace=True)
filtered_df.rename(columns={'count': 'group_count'}, inplace=True)
print(filtered_df)
x = filtered_df['home_ownership']
y = filtered_df['group_count']

fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(x, y)
ax.set_title('Bar Chart Home Ownership ')
ax.set_xlabel('home_ownership ')
ax.set_ylabel('Number of loans')
plt.show()
```

	home_ownership	group_count
0	MORTGAGE	17659
1	NONE	3
2	OTHER	98
3	OWN	3058
4	RENT	18899

Group by home_ownership



Target

Default/Risk on loan
Fully paid -
Current
Charged off
Ignore loans rejected

```
# 91306721 - Charged Off ; 1296599 - Fully paid; 920821 - Current ;

p_member_id = 920821
df = df[df['member_id']==p_member_id]
print(df)

def analysis_charged_off():
    return(df[['tot_cur_bal','tot_coll_amt','chargeoff_within_12_mths']])
def analysis_Current():
    return(df[['tot_cur_bal','tot_coll_amt','chargeoff_within_12_mths','delinq_amnt','inq_last_12m']])
def analysis_Fully_paid():
    return(df[['tot_cur_bal','tot_coll_amt']])

if df['loan_status'].iloc[0]=='Charged Off':
    review = analysis_charged_off()
elif df['loan_status'].iloc[0]=='Current':
    review = analysis_Current()
    print('analysis_charged_off')
elif df['loan_status'].iloc[0]=='Fully paid':
    review = analysis_Fully_paid()
    print('analysis_charged_off')
else:
    review = 'Others'
    print('Others')

print(df.loc[df['member_id'] ==p_member_id, ['loan_status']])
print(review)
```