Table of Contents

4. Data Dictionary

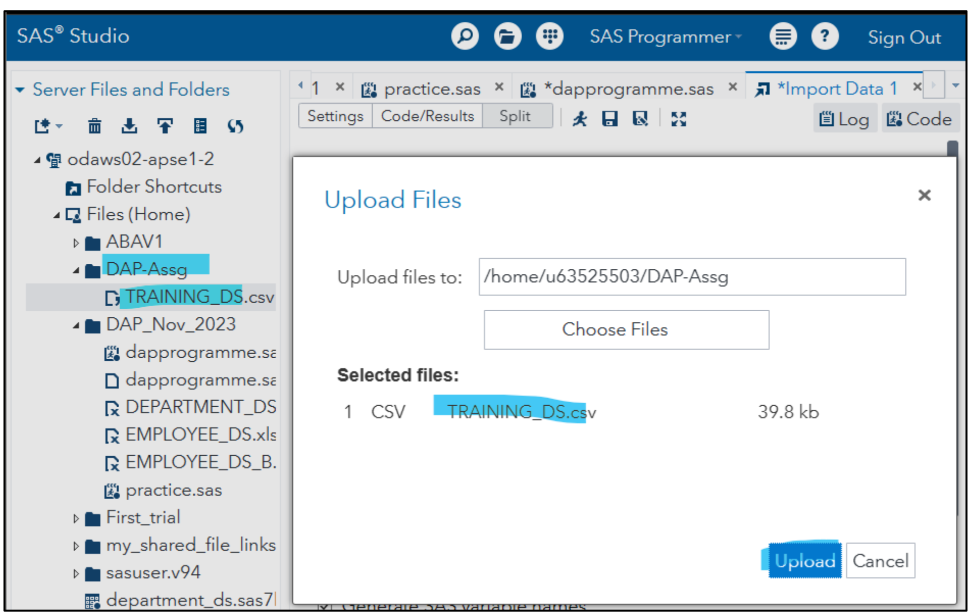
Name of variable	Description	Data Type	Length	Sample Data
SME_LOAN_ID_NO	Reference number	Char	8	LP002555, LP002571, LP002624, LP002625
GENDER	Gender of the applicant	Varchar	6	Female;Male
MARITAL_STATUS	Is the applicant married?	Varchar	11	Married;Not Married
FAMILY_MEMBERS	Total no. of Family Members	Numeric	2	0, 1, 2, 3+
QUALIFICATION	Graduate or Undergraduate	Varchar	14	Under Graduate
EMPLOYMENT	Yes / No	Varchar	3	Yes; No
CANDIDATE_INCOME	Monthly Income of the Applicant	Numeric	5	81000,4547
GUARANTEE_INCOME	Joint Applicant Income	Float	11	10968, 700, 985.79999878
LOAN_AMOUNT	Loan amount in thousands	Numeric	3	128

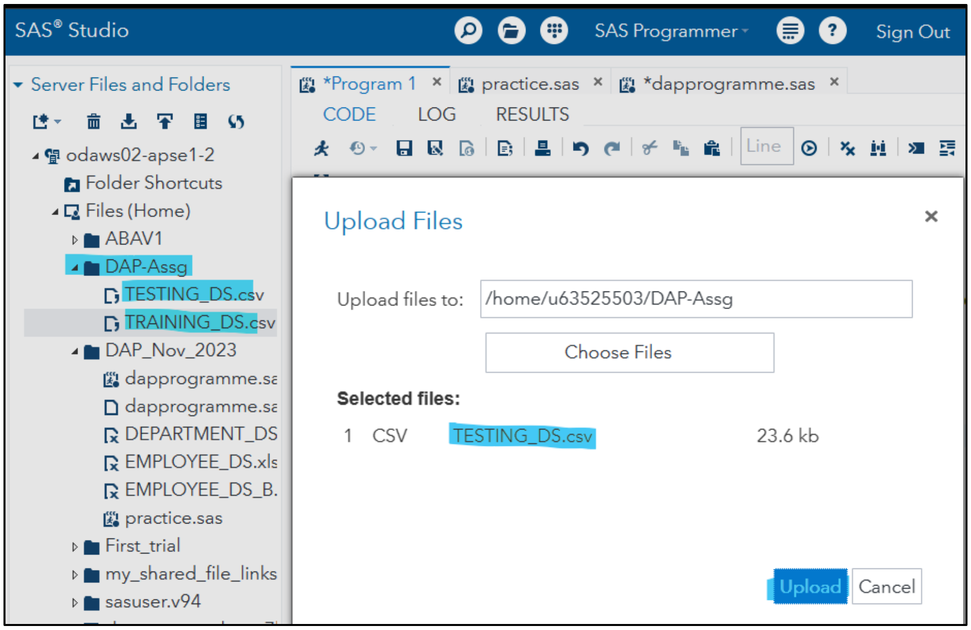
LOAN_DURATION	Repayment duration of loan	Numeric	3	360, 480
LOAN_HISTORY	Past loan records (positive or negative)	Numeric	1	1; 0
LOAN_LOCATION	City / Town / Village	Varchar	7	Village
LOAN_APPROVAL_STATUS	Yes / No	Char	1	Y; N

*Underline means that it is Continuous Variable

4.1 Upload the datasets given to SAS

4.1.1 Screenshots(s)



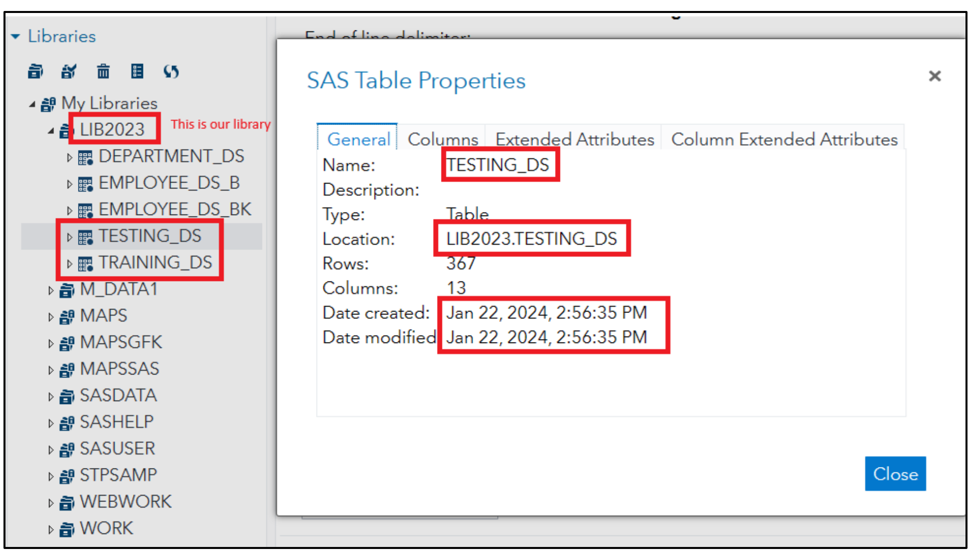


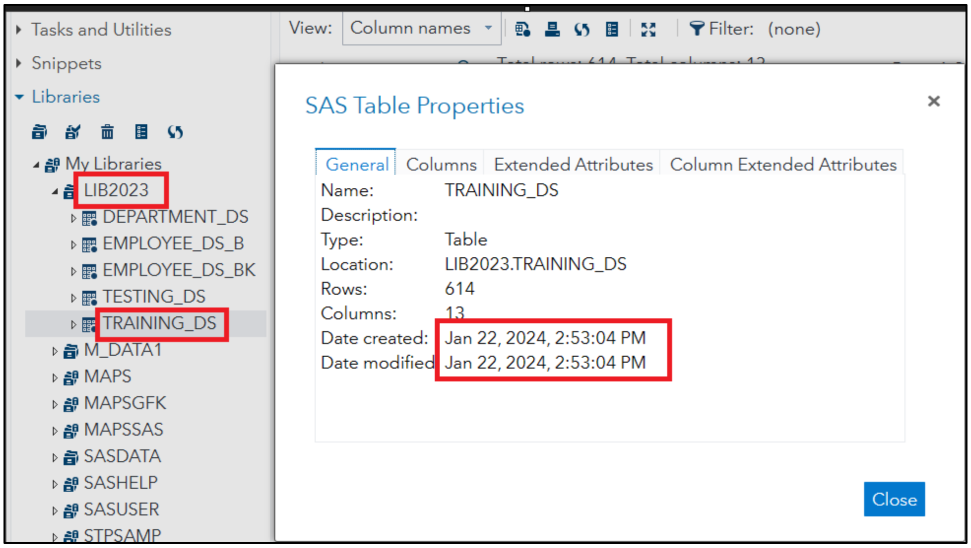
4.1.2 Description

The data is uploaded under the DAP-Assg folder, the dataset of name of TRAINING_DS and TESTING_DS are uploaded to SAS.

4.2 Upload the dataset to LIB2023

4.2.1 Screenshot





4.2.2 Description

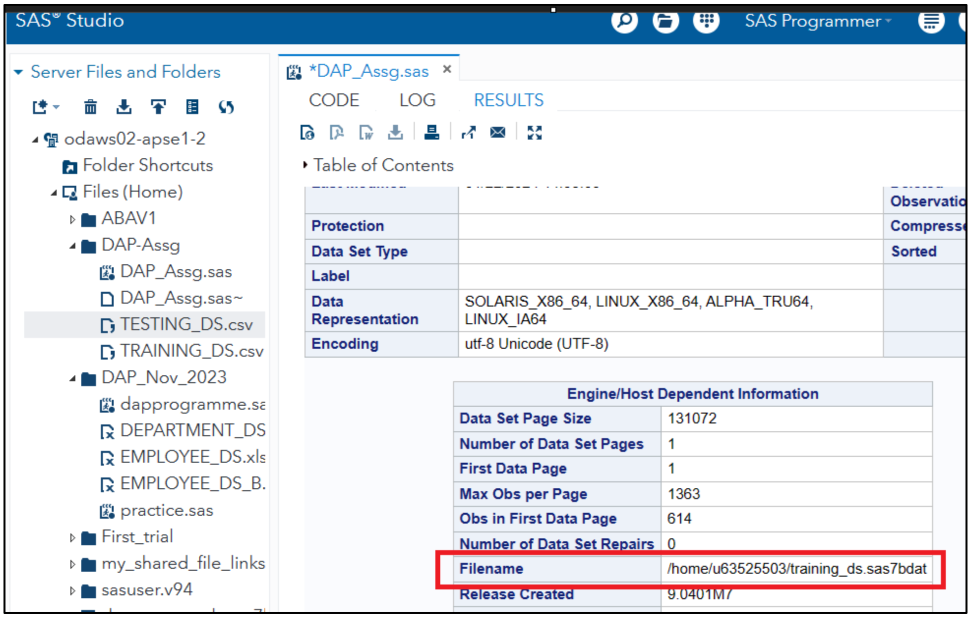
The library that is used to store the datasets is called LIB2023, to call it in the code, we use LIB2023.[dataset] which indicates that the dataset is from the library of name LIB2023. The date that these datasets are uploaded are shown above.

4.3 Data Set Structure

4.3.1 PROC CONTENTS

The CONTENTS Procedure

Data Set Name	LIB2023.TRAINING_DS	**Observations**	614
Member Type	DATA	**Variables**	13
Engine	V9	**Indexes**	0
Created	01/22/2024 14:53:05	**Observation Length**	96
Last Modified	01/22/2024 14:53:05	**Deleted Observations**	0
Protection		**Compressed**	NO
Data Set Type		**Sorted**	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		



Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
7	CANDIDATE_INCOME	Num	8	BEST12.	BEST32.
6	EMPLOYMENT	Char	3	$3.	$3.
4	FAMILY_MEMBERS	Char	2	$2.	$2.
2	GENDER	Char	6	$6.	$6.
8	GUARANTEE_INCOME	Num	8	BEST12.	BEST32.
9	LOAN_AMOUNT	Num	8	BEST12.	BEST32.
13	LOAN_APPROVAL_STATUS	Char	1	$1.	$1.
10	LOAN_DURATION	Num	8	BEST12.	BEST32.
11	LOAN_HISTORY	Num	8	BEST12.	BEST32.
12	LOAN_LOCATION	Char	7	$7.	$7.
3	MARITAL_STATUS	Char	11	$11.	$11.
5	QUALIFICATION	Char	14	$14.	$14.
1	SME_LOAN_ID_NO	Char	8	$8.	$8.

The screenshot in the figures above shows the metadata, such as the path of the dataset, and the details of the variables, such as type, length and format of each variables in a precise manner.

For example, the target variable i.e. LOAN_APPROVAL_STATUS, has length of 1, and the format “$1.” Indicates that it is string with 1 digit only. This matches the fact that it can only be 1 (approved) or 0 (else).

Structure

```
Proc SQL;
Describe table Lib2023.Training_DS;
run;
```

```
create table LIB2023.TRAINING_DS( bufsize=131072 )
  (
   SME_LOAN_ID_NO char(8) format=$8. informat=$8.,
   GENDER char(6) format=$6. informat=$6.,
   MARITAL_STATUS char(11) format=$11. informat=$11.,
   FAMILY_MEMBERS char(2) format=$2. informat=$2.,
   QUALIFICATION char(14) format=$14. informat=$14.,
   EMPLOYMENT char(3) format=$3. informat=$3.,
   CANDIDATE_INCOME num format=BEST12. informat=BEST32.,
   GUARANTEE_INCOME num format=BEST12. informat=BEST32.,
   LOAN_AMOUNT num format=BEST12. informat=BEST32.,
   LOAN_DURATION num format=BEST12. informat=BEST32.,
   LOAN_HISTORY num format=BEST12. informat=BEST32.,
   LOAN_LOCATION char(7) format=$7. informat=$7.,
   LOAN_APPROVAL_STATUS char(1) format=$1. informat=$1.
  );
```

From the “DESCIRBE” function, we can observe the structure of the data set. It is similar to the section 4.3.1, just that it is not that tidy as in a table form. Besides, we can see that the char(8), indicates that SME_LOAN_ID is a character/string with length 8. Another fun fact is that the target variable is not a number, but a string, although it is either “0” or “1”.

CHAPTER 6: Analysis of the variables / EDA

A) Training_DS

6.1 Univariate Analysis

The DS (Data Sciencetist) will perform EDA on the Training and Testing dataset to see if there are any issue in the dataset. The main problem we should focus on will be missing values and noisy data. The missing values can be accessed from the PROC FREQ for Categorical Var.

In this analysis, the DS will identify and take note of the variables that has issue such as missing value, it is because the missing value will not be input into the Logistic regression model, this will effect the prediction of the model. Hence, it will be important to perform EDA before running the model, to ensure that clean data is entered into the model, while ensuring the quality of the predictions.

6.1.1 Categorical variable:

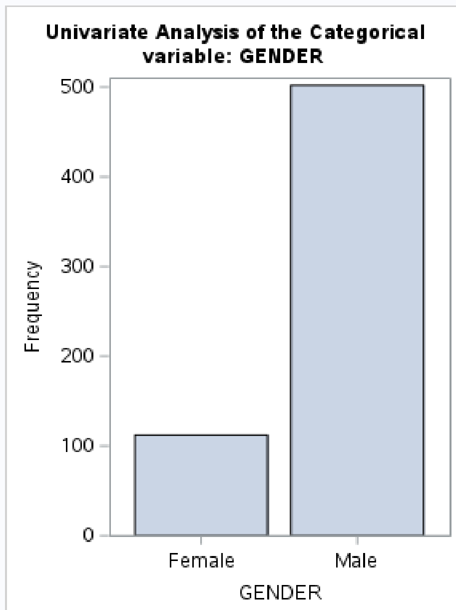
GENDER

```
TITLE 'Figure no 34343- Univariate Analysis of the Categorical variable: GENDER';
Proc FREQ data= LIB2023.TRAINING_DS;
Table GENDER;
run;

ODS GRAPHICS / RESET WIDTH = 3.0 IN HEIGHT = 4.0 IN IMAGEMAP;

PROC SGPLOT DATA = LIB2023.TRAINING_DS;

VBAR GENDER;

TITLE 'Figure no 2323 - Univariate Analysis of the Ctegorical variable: GENDER';

RUN;
```

Univariate Analysis of the Categorical variable: GENDER

The FREQ Procedure

GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Female	112	18.24	112	18.24
Male	502	81.76	614	100.00



The missing value issues are solved in the imputation stage, hence no missing value exists in this dataset. Usually, in the bottom of the first table, we will see “Frequency missing=?” if missing value really exists. We notice an imbalance distribution of Gender in the Training_ds, where the male applicants contributes to 81.76%. While female applicants is only 18.24%.

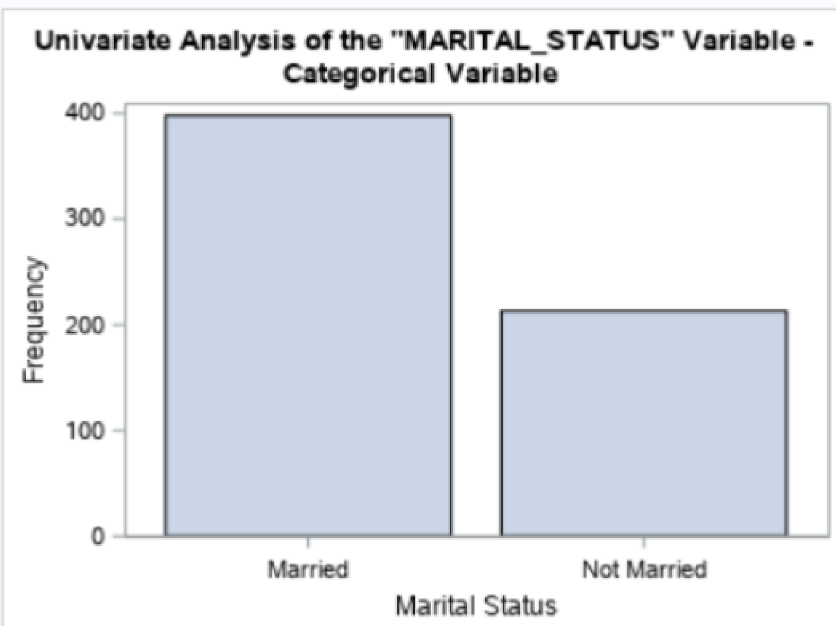
MARITAL STATUS

```
/* MARITAL_STATUS */
TITLE 'Univariate Analysis of the Categorical variable: MARTIAL_STATUS';
PROC FREQ DATA=  LIB2023.TRAINING_DS1;
TABLE MARITAL_STATUS;
RUN;

ODS GRAPHICS / RESET WIDTH = 3.0 IN HEIGHT = 4.0 IN IMAGEMAP;
PROC SGPLOT DATA =  LIB2023.TRAINING_DS1;
VBAR MARITAL_STATUS;
RUN;
```

Univariate Analysis of the "MARITAL_STATUS" Variable - Categorical Variable

The FREQ Procedure

Marital Status				
MARITAL_STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Married	398	65.14	398	65.14
Not Married	213	34.86	611	100.00
Frequency Missing = 3				



- There are 3 missing values in the ds.
- An uneven distribution between the 2 groups is observed.

- Applicants that were married have 65.14%, while the not married ones has 34.86%.

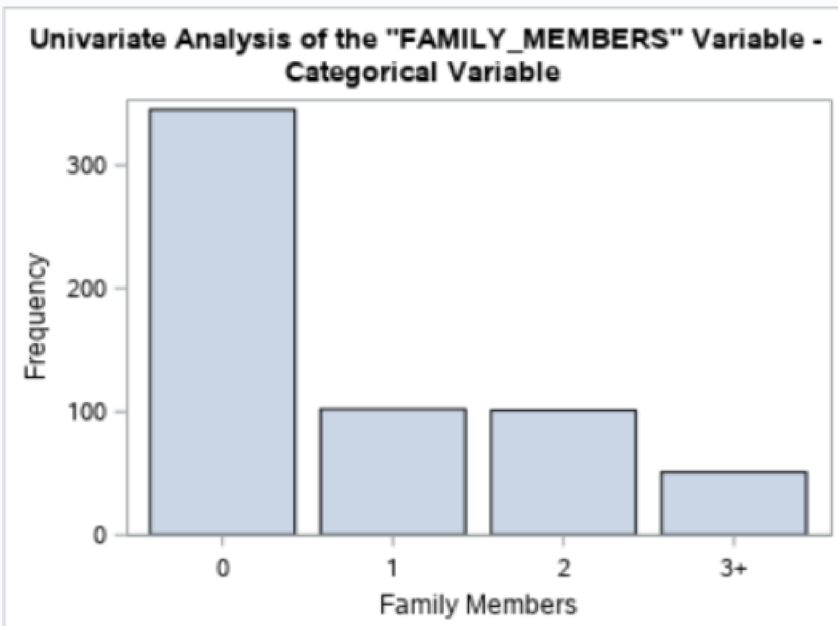
FAMILY MEMBERS

```
/* FAMILY_MEMBERS */
TITLE 'Univariate Analysis of the Categorical Variable: FAMILY_MEMBERS';
/* SAS code to do Univariate Analysis of the "FAMILY_MEMBERS" variable */
PROC FREQ DATA = LIB2023.TRAINING_DS1;
TABLE FAMILY_MEMBERS;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB2023.TRAINING_DS1;
VBAR FAMILY_MEMBERS;
RUN;
```

Univariate Analysis of the "FAMILY_MEMBERS" Variable - Categorical Variable

The FREQ Procedure

Family Members				
FAMILY_MEMBERS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	345	57.60	345	57.60
1	102	17.03	447	74.62
2	101	16.86	548	91.49
3+	51	8.51	599	100.00
Frequency Missing = 15				



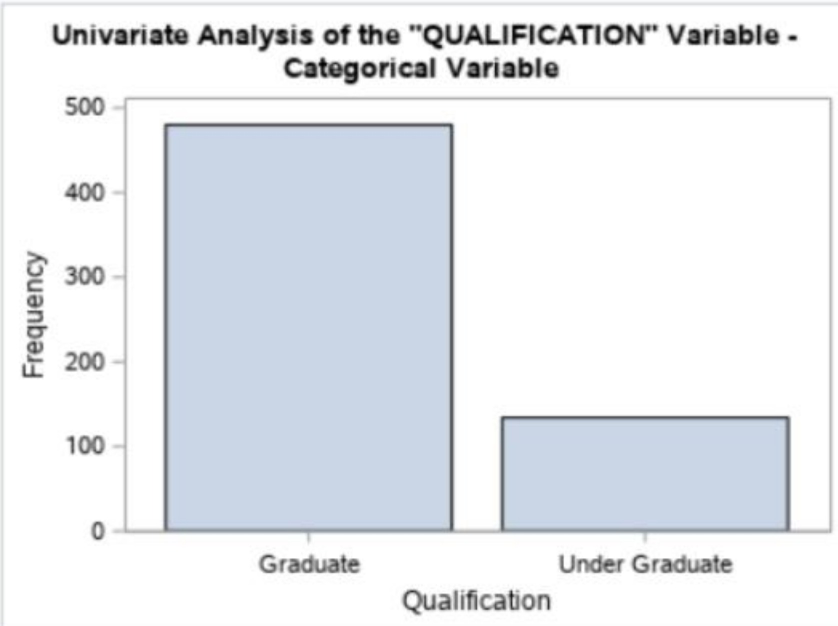
- 15 missing values are found.
- 57.6% of applicants have 0 family members, while 8.51% have 3 or more family members.

Qualification

Univariate Analysis of the "QUALIFICATION" Variable - Categorical Variable

The FREQ Procedure

Qualification				
QUALIFICATION	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Graduate	480	78.18	480	78.18
Under Graduate	134	21.82	614	100.00



- There is no missing value in this variable.
- The applicants are more towards having a graduate degree. (78.18%)

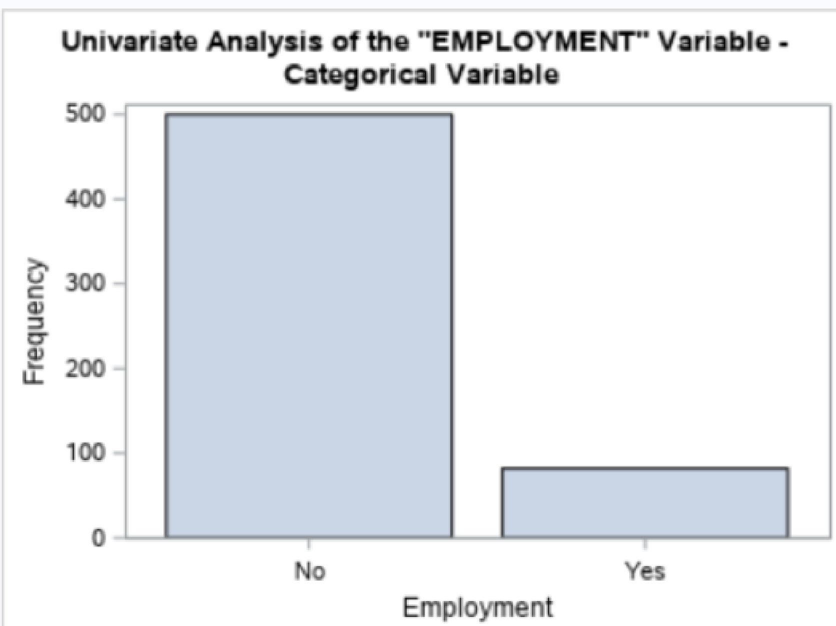
EMPLOYMENT

```
/* EMPLOYMENT */
TITLE 'Univariate Analysis of the Categorical Variable: EMPLOYMENT';
/* SAS code to do Univariate Analysis of the "EMPLOYMENT" variable */
PROC FREQ DATA = LIB2023.TRAINING_DS1;
TABLE EMPLOYMENT;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB2023.TRAINING_DS1;
VBAR EMPLOYMENT;
RUN;
```

Univariate Analysis of the "EMPLOYMENT" Variable - Categorical Variable

The FREQ Procedure

Employment				
EMPLOYMENT	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	500	85.91	500	85.91
Yes	82	14.09	582	100.00
Frequency Missing = 32				



- 32 applicants didn't enter the employment status.
- Most of the applicants are not employed (85.91%).

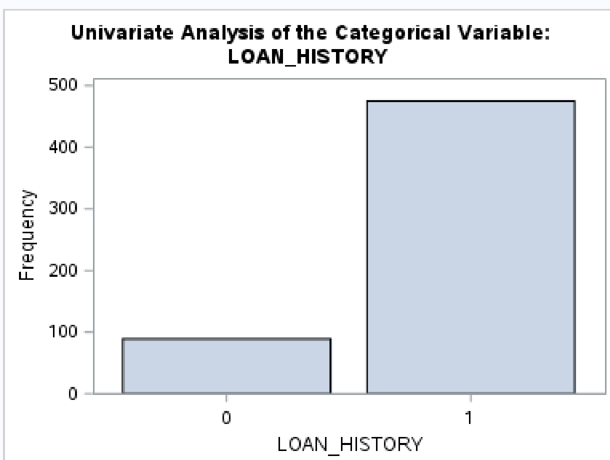
LOAN HISTORY

```
/* LOAN_HISTORY */
TITLE 'Univariate Analysis of the Categorical Variable: LOAN_HISTORY';
/* SAS code to do Univariate Analysis of the "LOAN_HISTORY" variable */
PROC FREQ DATA = LIB2023.TRAINING_DS1;
TABLE LOAN_HISTORY;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB2023.TRAINING_DS1;
VBAR LOAN_HISTORY;
RUN;
```

Univariate Analysis of the Categorical Variable: LOAN_HISTORY

The FREQ Procedure

LOAN_HISTORY	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	89	15.78	89	15.78
1	475	84.22	564	100.00
Frequency Missing = 50				



- 50 loan applicants don’t have a respond for the loan history.
- 84.22% of applicants have applied for a loan in the past.

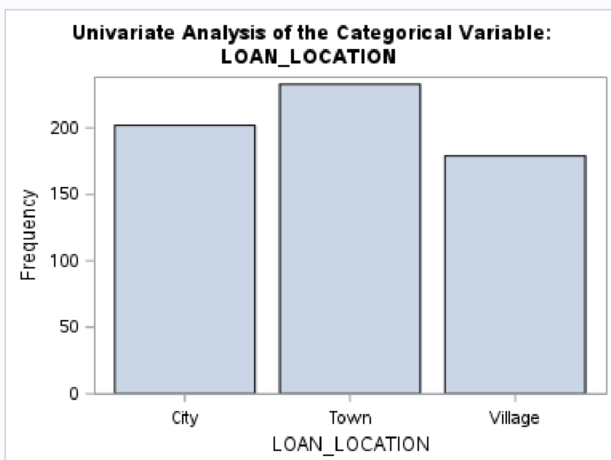
LOAN_LOCATION

```
/* LOAN_LOCATION */
TITLE 'Univariate Analysis of the Categorical Variable: LOAN_LOCATION';
/* SAS code to do Univariate Analysis of the "LOAN_LOCATION" variable */
PROC FREQ DATA = LIB2023.TRAINING_DS1;
TABLE LOAN_LOCATION;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB2023.TRAINING_DS1;
VBAR LOAN_LOCATION;
RUN;
```

Univariate Analysis of the Categorical Variable: LOAN_LOCATION

The FREQ Procedure

LOAN_LOCATION	Frequency	Percent	Cumulative Frequency	Cumulative Percent
City	202	32.90	202	32.90
Town	233	37.95	435	70.85
Village	179	29.15	614	100.00



- There are no applicants with unidentified qualifications in the dataset.
- As many as 32.9% (202 applicants) live in the city, 37.95% (233 applicants) live in the town.
- Moreover, 29.15% (179 applicants) of them live in the village.

LOAN_APPROVAL_STATUS

Univariate Analysis of the Categorical Variable: LOAN_APPROVAL_STATUS

The FREQ Procedure

LOAN_APPROVAL_STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
N	192	31.27	192	31.27
Y	422	68.73	614	100.00

- There are no missing values or no applicants that had unidentified loan approval status in the dataset.
- The dataset has an uneven distribution between approved loans (Y) and rejected loans (N), with the percentage of the approved loan (Y) is 68.73%, and the percentage of the rejected loan (N) is 31.27%.

6.1.2 Continuous variable:

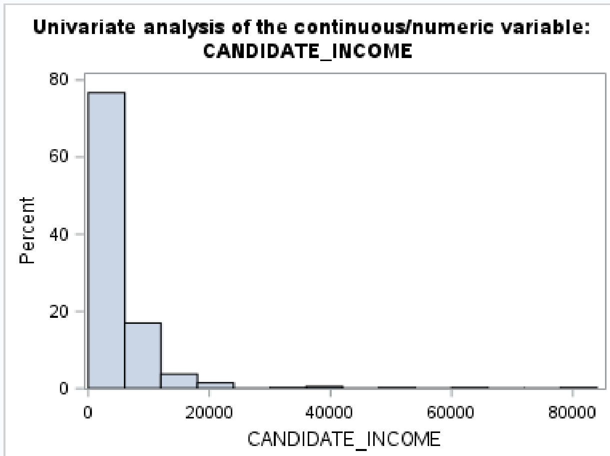
CANDIDATE_INCOME

```
/* CANDIDATE_INCOME  */
TITLE 'Univariate analysis of the continuous/numeric variable: CANDIDATE_INCOME  ';
PROC MEANS DATA =   LIB2023.TRAINING_DS1 N NMISS MIN MAX MEAN MEDIAN STD;
VAR CANDIDATE_INCOME ;
RUN;
ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;
PROC SGPLOT DATA =   LIB2023.TRAINING_DS1;
HISTOGRAM CANDIDATE_INCOME ;
RUN;
```

Univariate analysis of the continuous/numeric variable: CANDIDATE_INCOME

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
614	0	150.0000000	81000.00	5403.46	3812.50	6109.04



- There are no missing values, i.e. no users had unidentified income in the dataset.
- Both the histogram, mean table indicate that the data distribution for this variable is positively skewed, with the median 3,812.5 and mean 5,403.46.
- Noticed that this variable contains extreme outliers because the maximum value 81k is greater than the (mean + 3*sd) value.

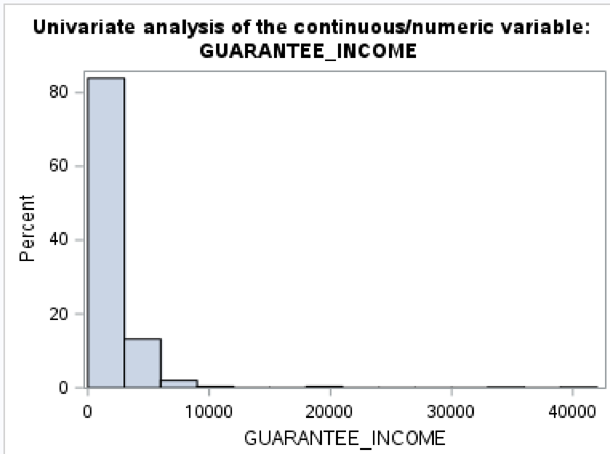
GUARANTEE_INCOME

```
/* GUARANTEE_INCOME  */
TITLE 'Univariate analysis of the continuous/numeric variable: GUARANTEE_INCOME  ';
PROC MEANS DATA =   LIB2023.TRAINING_DS1 N NMISS MIN MAX MEAN MEDIAN STD;
VAR GUARANTEE_INCOME ;
RUN;
ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;
PROC SGPLOT DATA =   LIB2023.TRAINING_DS1;
HISTOGRAM GUARANTEE_INCOME ;
RUN;
```

Univariate analysis of the continuous/numeric variable: GUARANTEE_INCOME

The MEANS Procedure

Analysis Variable : GUARANTEE_INCOME						
N	**N Miss**	**Minimum**	**Maximum**	**Mean**	**Median**	**Std Dev**
614	0	0	41667.00	1621.25	1188.50	2926.25



- There are no missing values for var. guarantee income in the dataset.
- Both the histogram, mean table indicate that the data distribution for this variable is positively skewed, with the median 1188.50 and mean 1621.25.
- Noticed that this variable contains extreme outliers because the maximum value 41k is greater than the (mean + 3*sd) value.

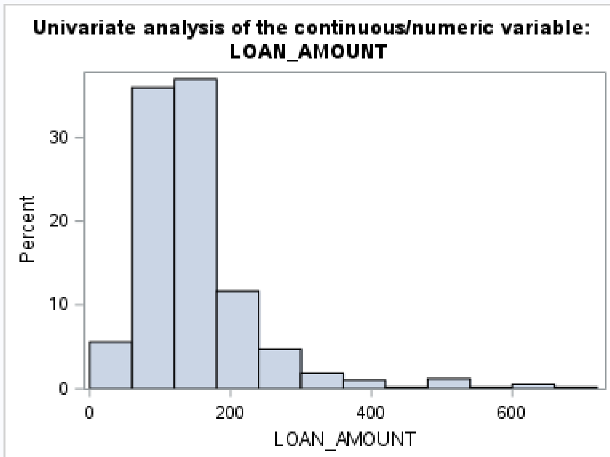
LOAN_AMOUNT

```
/* LOAN_AMOUNT */
TITLE 'Univariate analysis of the continuous/numeric variable: LOAN_AMOUNT ';
PROC MEANS DATA =   LIB2023.TRAINING_DS1 N NMISS MIN MAX MEAN MEDIAN STD;
VAR loan_amount;
RUN;
ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;
PROC SGPLOT DATA =   LIB2023.TRAINING_DS1;
HISTOGRAM loan_amount;
RUN;
```

Univariate analysis of the continuous/numeric variable: LOAN_AMOUNT

The MEANS Procedure

Analysis Variable : LOAN_AMOUNT						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
592	22	9.0000000	700.0000000	146.4121622	128.0000000	85.5873252



- There are 22 missing values.
- The histogram indicates positively skewness, while the median 128 and mean 146.4121622.

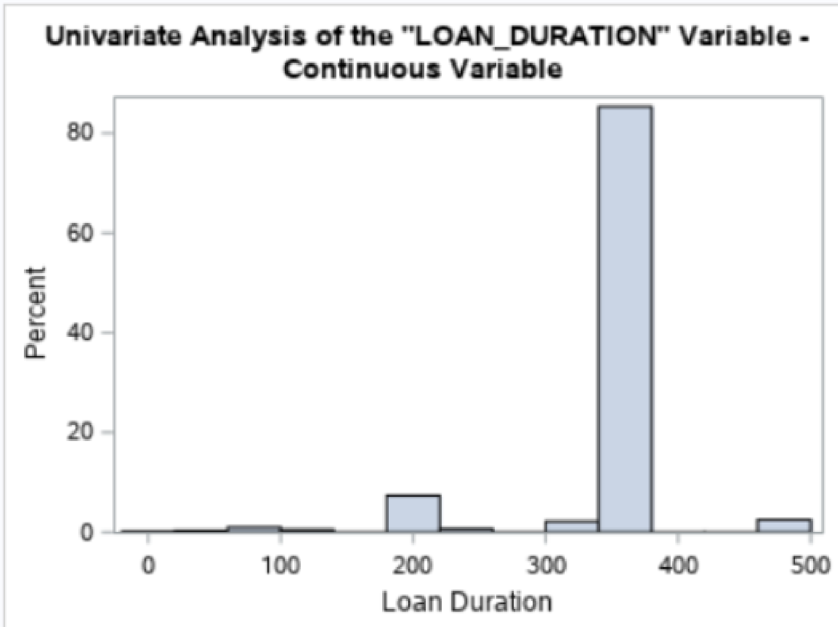
LOAN_DURATION

```
/* LOAN_DURATION */
TITLE 'Univariate analysis of the continuous/numeric variable: LOAN_DURATION ';
PROC MEANS DATA =   LIB2023.TRAINING_DS1 N NMISS MIN MAX MEAN MEDIAN STD;
VAR LOAN_DURATION;
RUN;
ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;
PROC SGPLOT DATA =   LIB2023.TRAINING_DS1;
HISTOGRAM LOAN_DURATION;
RUN;
```

Univariate Analysis of the "LOAN_DURATION" Variable - Continuous Variable

The MEANS Procedure

Analysis Variable : LOAN_DURATION Loan Duration						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
600	14	12.0000000	480.0000000	342.0000000	360.0000000	65.1204099



- There are 14 missing values.
- From the histogram, the distribution is positively skewed, with the median 360 and mean 342.

6.2 Bivariate Analysis

The DS would like to analyze the relationship between 2 variables. From there, we can identify hidden patterns that are helpful for the EDA and Logit Model.

The SAS Macro is used, it is a very powerful syntax that can be used to save time and improve coding experience. It can help to prevent repetitive tasks and increase overall efficiency.

```
/* Macro */
%MACRO MACRO_BIVA_CV( DATASET_NAME, VARIABLE_1, VARIABLE_2, TITLE_1, TITLE_2);
PROC FREQ data= &DATASET_NAME;
TABLE &VARIABLE_1 * &VARIABLE_2 /
PLOTS= FREQPLOT( TWOWAY= STACKED SCALE = GROUPPCT );
TITLE &TITLE_1;
TITLE2 &TITLE_2;
RUN;
%MEND MACRO_BIVA_CV;

/*********************************************************************************************
To run bivariate analysis on Categorical vs Categorical (3 CASES)
**********************************************************************************************/

/* GENDER Vs. MARITAL_STATUS */
%MACRO_BIVA_CV(LIB2023.TRAINING_DS1, GENDER, MARITAL_STATUS,'Bivariate analysis', 'on GENDER(Categorical) Vs. MARITAL_STATUS(Categorical)');

/* GENDER VS. LOAN_APPROVAL_STATUS */
%MACRO_BIVA_CV(LIB2023.TRAINING_DS1, GENDER, LOAN_APPROVAL_STATUS,'Bivariate analysis', 'on GENDER(Categorical) Vs. LOAN_APPROVAL_STATUS(Categorical)');

/*EMPLOYMENT Vs. MARITAL-STATUS */
%MACRO_BIVA_CV(LIB2023.TRAINING_DS1, EMPLOYMENT, MARITAL_STATUS,'Bivariate analysis', 'on EMPLOYMENT(Categorical) Vs. MARITAL_STATUS(Categorical)');
```

- The dataset_name placeholder is for the location of the dataset, in this case, LIB2023.TRAINING_DS.
- Variable 1 and 2 is for the Categorical Var. that we want to analyze.
- Title’s are used to create title for the output.

6.2.1 GENDER Vs MARITAL STATUS

Screenshot of code

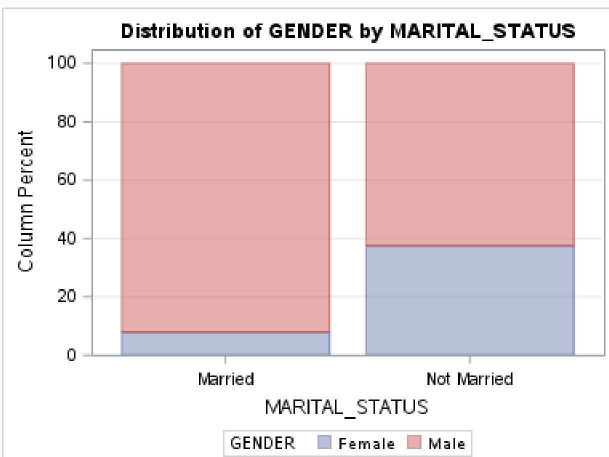
Bivariate analysis
on GENDER(Categorical) Vs. MARITAL_STATUS(Categorical)

The FREQ Procedure

Frequency Percent Row Pct Col Pct

Table of GENDER by MARITAL_STATUS

GENDER	MARITAL_STATUS: Married	MARITAL_STATUS: Not Married	Total
Female	32 5.21 28.57 7.98	80 13.03 71.43 37.56	112 18.24
Male	369 60.10 73.51 92.02	133 21.66 26.49 62.44	502 81.76
Total	401 65.31	213 34.69	614 100.00



- Most male applicants are married (92%) among those who are married. However, 71% of females are not married, this contributes to 37.5% among applicants who are not married.
- The female applicants that are married is 8% among those who are married. However, 26% of males are not married, this contributes to 62.44% among applicants who are not married.
- This can be explained can the uneven distribution of gender in the dataset. Causing a lot of weight towards the Male applicants.

6.2.2 GENDER Vs LOAN_APPROVAL_STATUS

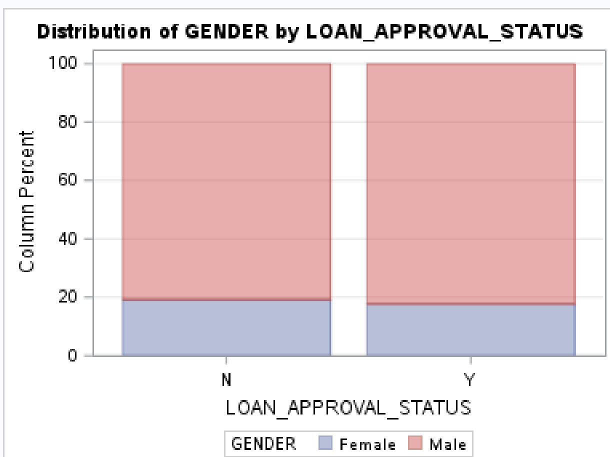
Bivariate analysis
on GENDER(Categorical) Vs. LOAN_APPROVAL_STATUS(Categorical)

The FREQ Procedure

Frequency Percent Row Pct Col Pct

Table of GENDER by LOAN_APPROVAL_STATUS

GENDER	LOAN_APPROVAL_STATUS N	LOAN_APPROVAL_STATUS Y	Total
Female	37 6.03 33.04 19.27	75 12.21 66.96 17.77	112 18.24
Male	155 25.24 30.88 80.73	347 56.51 69.12 82.23	502 81.76
Total	192 31.27	422 68.73	614 100.00



- Most male applicants got Y (82%) among those who got Y. However, 33% of females got N, this contributes to 19.27% among applicants who got N.
- The female applicants that got Y is 17% among those who got Y. However, 30.88% of males got N, this contributes to 80.73% among applicants who got N.
- This can be explained can the uneven distribution of gender in the dataset. Causing a lot of weight towards the Male applicants.

6.2.3 EMPLOYMENT VS MARITAL STATUS

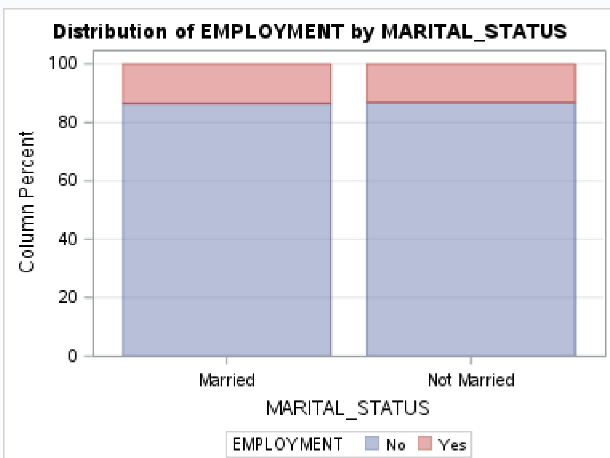
Bivariate analysis
on EMPLOYMENT(Categorical) Vs. MARITAL_STATUS(Categorical)

The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of EMPLOYMENT by MARITAL_STATUS

	MARITAL_STATUS		
EMPLOYMENT	**Married**	**Not Married**	**Total**
No	347 56.51 65.23 86.53	185 30.13 34.77 86.85	532 86.64
Yes	54 8.79 65.85 13.47	28 4.56 34.15 13.15	82 13.36
Total	401 65.31	213 34.69	614 100.00



- 30.13% of applicants are not married and not employed, while 4.56% of all applicants are not married but employed.
- 56.51% of applicants are married but not employed, while 8.79% of applicants are married and employed.
- A huge proportion of applicants are not employed. This can be explained by the uneven distribution of employment var. in the dataset. Causing a lot of weight towards the applicants that are unemployed.

B) Testing_DS

6.3 Univariate

The description will be similar to the one that is used for the training dataset. Where we determine which variables requires imputation to improve the quality of the testing dataset.

6.3.1 Categorical

```
/*****************************************************************************************
Univariate Analysis of the Categorical variables using SAS MACRO in “TESTING_DS”
*****************************************************************************************/
/* Macro Begins here */
OPTIONS mcompilenote=ALL;
%MACRO MMACRO_UVA_TRAINING_DS1(pvariable,pdataset);
TITLE "Univariate Analysis of the categorical variable- &pvariable using SAS MACRO" ;
PROC FREQ DATA= &pdataset;
TABLE &pvariable;
RUN;
%MEND MMACRO_UVA_TRAINING_DS1;
/* Macro ends here */

/*****************************************************************************************
To run Univariate Analysis on Categorical variables in “TESTING_DS”
*****************************************************************************************/
/* Gender */
%MMACRO_UVA_TRAINING_DS1(GENDER, LIB2023.Testing_Ds);

/*Marital_Status*/
%MMACRO_UVA_TRAINING_DS1(MARITAL_STATUS, LIB2023.Testing_Ds);
```

We use SAS Macro to prevent repetitive tasks.

The pdataset is used to identify the location of the file, while the pvairable denotes the name of the variables that will be used.

GENDER

Univariate Analysis of the categorical variable- GENDER using SAS MACRO

The FREQ Procedure

GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Female	70	19.66	70	19.66
Male	286	80.34	356	100.00
Frequency Missing = 11				

- There are 11 records in the testing_ds, who has a missing GENDER in the dataset.
- The distribution is not even, we can see that the percentage of male is 80.34%.

FAMILY_MEMBERS

Univariate analysis on FAMILY_MEMBERS (Categorical) Variable

The FREQ Procedure

FAMILY_MEMBERS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	200	56.02	200	56.02
1	58	16.25	258	72.27
2	59	16.53	317	88.80
3+	40	11.20	357	100.00
Frequency Missing = 10				

- There are 10 records in the testing_ds, who has a missing Family_Members value in the dataset.
- The distribution is not even, we can see that the percentage of applicants with 0 family members is 56%, while 11.20% of it have 3 or more family members.

EMPLOYMENT

Univariate analysis
on EMPLOYMENT (Categorical) Variable

The FREQ Procedure

EMPLOYMENT	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	307	89.24	307	89.24
Yes	37	10.76	344	100.00
Frequency Missing = 23				

- There are 23 records in the testing_ds, who has a missing value in this column of the dataset.
- The distribution is not even, we can see that the percentage of unemployed applicants is 89.24%, while employed applicants only have 10.76%.

LOAN_HISTORY

Univariate analysis
on LOAN_HISTORY (Categorical) Variable

The FREQ Procedure

LOAN_HISTORY	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	59	17.46	59	17.46
1	279	82.54	338	100.00
Frequency Missing = 29				

- There are 29 records in the testing_ds, who has a missing LOAN_HISTORY in the dataset.
- The distribution is not even, we can see that the percentage of applicants with value 0 is 17.46%, while the percentage of applicants with value 1 is 82.54%.

6.3.2 CONTINUOUS VARIABLE

LOAN_AMOUNT

Univariate Analysis of the Continuous Variable- LOAN_AMOUNT using SAS MACRO

The MEANS Procedure

Analysis Variable : LOAN_AMOUNT						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
362	5	28.0000000	550.0000000	136.1325967	125.0000000	61.3666524

It has 5 missing values. The mean is 136.1326, while the median is 125.

LOAN_DURATION

Univariate Analysis of the Continuous Variable- LOAN_DURATION using SAS MACRO

The MEANS Procedure

Analysis Variable : LOAN_DURATION						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
361	6	6.0000000	480.0000000	342.5373961	360.0000000	65.1566434

It has 6 missing values. The mean is 342.5, while the median is 360.

6.4 Bivariate

6.4.1 CATEGORICAL VS CATEGORICAL

```
/*****************************************************************************************************
Bivariate Analysis using SAS MACRO in “TESTING_DS”
/* Categorical vs. Categorical */
*****************************************************************************************************/

/* SAS MACRO begins here */
OPTIONS MCOMPILENOTE=ALL;
%MACRO MACRO_BVA_CATE_CATE(ptitle1,ptitle2,pcate_vari1,pcate_vari2,pdataset);
TITLE1 &ptitle1;
TITLE2 &ptitle2;
PROC FREQ DATA=&pdataset;
TABLE &pcate_vari1 * &pcate_vari2/
PLOTS=FREQPLOT(TWOWAY=STACKED SCALE=GROUPPCT);
RUN;
%MEND MACRO_BVA_CATE_CATE;
/*SAS MACRO ends here */

/* Call the MACRO */
/* GENDER VS LOAN LOCATION */
%Macro_bva_cate_cate('Bivariate Analysis of Variables', 'GENDER VS LOAN LOCATION', gender, loan_location,  LIB2023.Testing_DS);

/* GENDER VS qualification */
%Macro_bva_cate_cate('Bivariate Analysis of Variables', 'GENDER VS qualification', gender, qualification,  LIB2023.Testing_DS);
```

SAS Macro is used to prevent repetitive codes, as can be seen above, instead of running the same code one by one. Macro helps to record the syntax and it only changes the input variables to minimize what that must be typed by the user.

The ptitle1 and 2 helps to form the title on the beginning of the output, while the variables are then input using the pcate_vari1 and pcate_vari2, lastly followed by the pdataset which indictates the location of the dataset.

GENDER VS LOAN LOCATION

Bivariate analysis
on GENDER (Categorical) Vs. LOAN_LOCATION (Categorical)

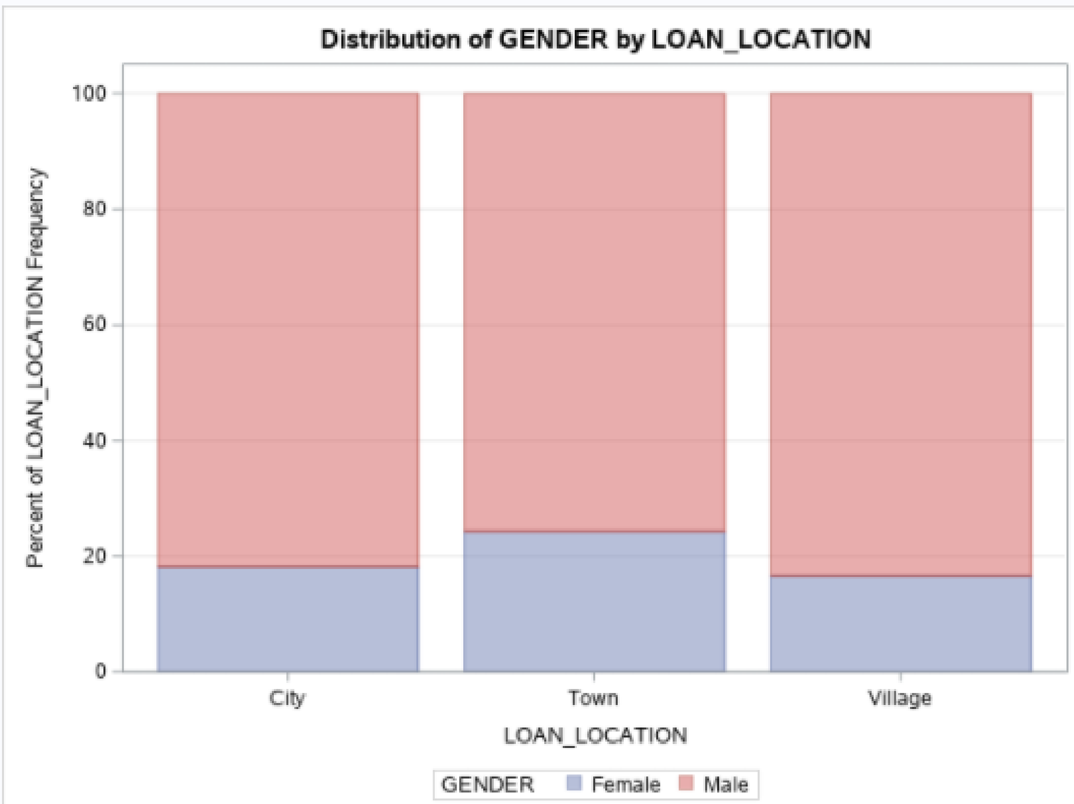
The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of GENDER by LOAN_LOCATION

GENDER	LOAN_LOCATION City	Town	Village	Total
Female	25 7.02 35.71 18.25	27 7.58 38.57 24.32	18 5.06 25.71 16.67	70 19.66
Male	112 31.46 39.16 81.75	84 23.60 29.37 75.68	90 25.28 31.47 83.33	286 80.34
Total	137 38.48	111 31.18	108 30.34	356 100.00

Frequency Missing = 11



- Most male applicants come from the village with a percentage of 83.33%.
- Male applicants among those who come from the city, has 81.75%. While male applicants among who comes from town are 75.68%.
- The majority of female applicants come from a town with a percentage of 24.32%.

- Female applicants among those who come from the city are only 18.25%, female applicants among those who come from the village are only 16.67%.
- There are 11 missing values, which all come from Gender.

GENDER VS qualification

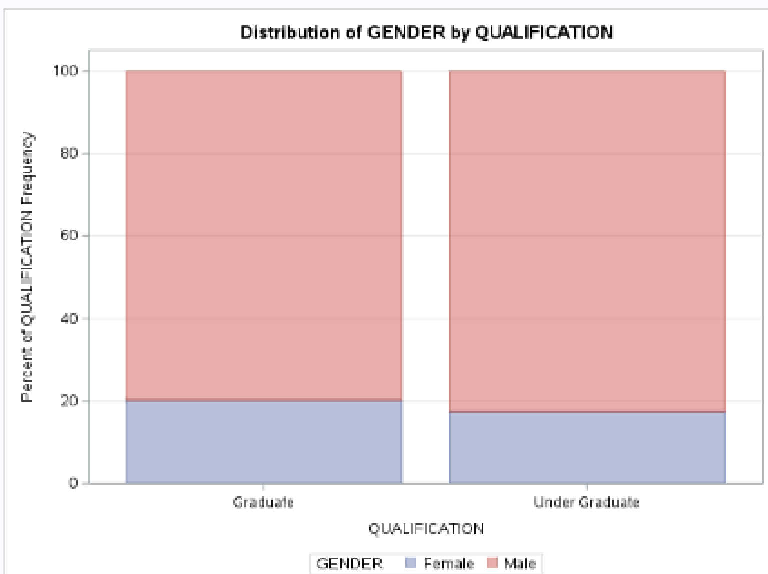
Bivariate Analysis of Variables
GENDER VS qualification

The FREQ Procedure

Frequency Percent Row Pct Col Pct

Table of GENDER by QUALIFICATION

	QUALIFICATION		
GENDER	Graduate	Under Graduate	Total
Female	56 15.73 80.00 20.29	14 3.93 20.00 17.50	70 19.66
Male	220 61.80 76.92 79.71	66 18.54 23.08 82.50	286 80.34
Total	276 77.53	80 22.47	356 100.00
Frequency Missing = 11			



- Among female, 80% of them have graduate degree, the other 20% have under graduate degree.

- Among those who have graduate degree, 79.71% of them are male.
- Among male, 76.92% of them have Graduate degree.
- Among those who have undergrad degree, 82.5% of them are male.
- There are 11 missing values where all are coming from GENDER.

GENDER VS LOAN HISTORY

Bivariate Analysis of Variables
GENDER VS LOAN HISTORY

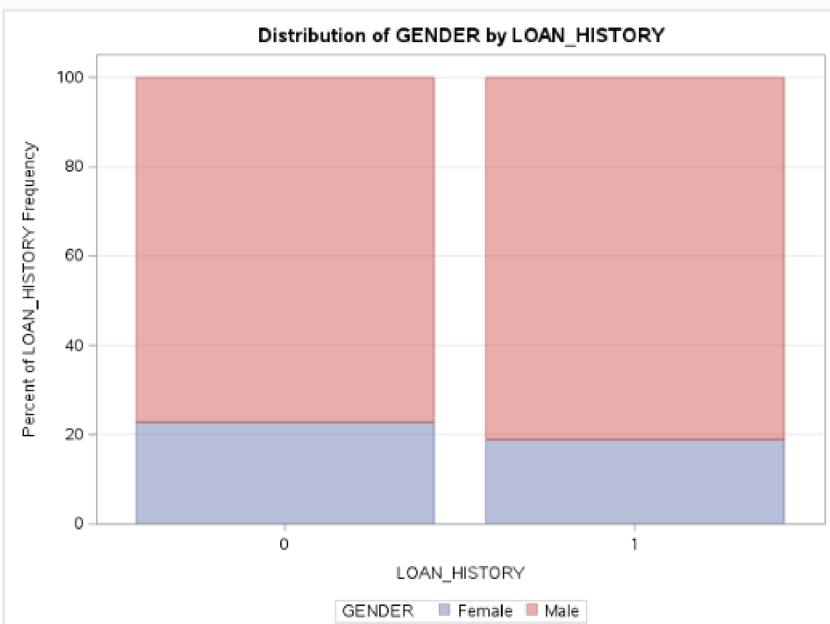
The FREQ Procedure

Frequency Percent Row Pct Col Pct

Table of GENDER by LOAN_HISTORY

GENDER	LOAN_HISTORY 0	LOAN_HISTORY 1	Total
Female	13 3.98 20.31 22.81	51 15.60 79.69 18.89	64 19.57
Male	44 13.46 16.73 77.19	219 66.97 83.27 81.11	263 80.43
Total	57 17.43	270 82.57	327 100.00

Frequency Missing = 40



- Among Female, 79.69% of them has 1 for LOAN HISTORY. While among male, 83.27% of them has 1 for LOAN HISTORY.
- Among those with LOAN HISTORY of 0, 77.19% of them are Male. While among those with LOAN HISTORY OF 1, 81.11% of them are Male.
- There are 40 missing values, 11 from GENDER, 29 from LOAN_HISTORY.

6.4.2 CATEGORICAL VS CONTINOUS

```
/*****************************************************************************************************
Bivariate Analysis using SAS MACRO in "TESTING_DS"
CATEGORICAL VS. CONTINUOUS
*****************************************************************************************************/

/* SAS MACRO begins here */
OPTIONS Mcompilenote=ALL;
%MACRO MACRO_BVA_CATE_CONTI(ptitle1,ptitle2,pcate,pconti,pdataset);
TITLE1 &ptitle1;
TITLE2 &ptitle2;
PROC Means DATA=&pdataset;
    CLASS &pcate;/*  CATE */
    VAR &pconti; /* CONTI */
RUN;
%MEND MACRO_BVA_CATE_CONTI;
/* MACRO ENDS HERE */

/* Gender vs Guarantee_income */
%MACRO_BVA_CATE_CONTI("Bivariate Analysis of Variables",'GENDER vs GUARANTEE_INCOME',
gender, guarantee_income,  LIB2023.Testing_DS);

/* Location vs Candidate_income */
%MACRO_BVa_cate_conti('Bivariate Analysis of Variables','Location vs Candidate Income',
 Loan_location, Candidate_income,  LIB2023.Testing_DS);

/* Marital_status vs Candidate_income */
%macro_bva_cate_conti('Bivariate Analysis of Variables','Marital status vs Candidate Income',
 MARITAL_STATUS, CANDIDATE_INCOME, LIB2023.TESTING_DS);
```

GENDER VS GUARANTEE INCOME

Bivariate Analysis of Variables
GENDER vs GUARANTEE_INCOME

The MEANS Procedure

Analysis Variable : GUARANTEE_INCOME						
GENDER	N Obs	N	Mean	Std Dev	Minimum	Maximum
Female	70	70	1171.96	1979.82	0	11666.00
Male	286	286	1670.87	2433.94	0	24000.00

- It can be seen that guarantee income for male are much higher (mean = 1670) compare to female (mean = 1171).
- By comparing the max with the mean, we see extreme outliers for the dataset.

LOCATION VS CANDIDATE INCOME

Bivariate Analysis of Variables
Location vs Candidate Income

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
LOAN_LOCATION	N Obs	N	Mean	Std Dev	Minimum	Maximum
City	140	140	5038.91	6285.96	1141.00	72529.00
Town	116	116	4745.69	4576.06	0	32000.00
Village	111	111	4573.94	2878.67	0	18840.00

- It can be seen that candidate income for city are the highest (mean = 5038) compare to village (mean = 4574).
- By comparing the max with the mean, we see extreme outliers for the dataset.

MARITAL STATUS VS CANDIDATE INCOME

Bivariate Analysis of Variables
Marital status vs Candidate Income

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
MARITAL_STATUS	N Obs	N	Mean	Std Dev	Minimum	Maximum
Married	233	233	4996.25	5450.00	570.0000000	72529.00
Not Married	134	134	4474.09	3791.41	0	29167.00

- It can be seen that candidate income for married applicants are higher (mean = 5000) compare to female (mean = 4474).

- By comparing the max with the mean, we see extreme outliers for the dataset.

Imputation

Training		Testing	
Mode	Mean	Mode	Mean
Family Members	Loan Amount	Gender	Loan Amount
Marital Status	Loan Duration	Family Members	Loan Duration
Employment		Employment	
Loan History		Loan History	

Table 1. shows the summary for data imputation.

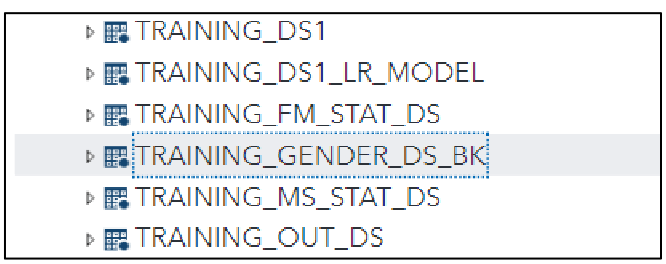
The Data Scientist will Impute the missing values found in the **Categorical** variable using the **mode**, while **continuous** variables are imputed using **mean.**

7.1 Training_DS

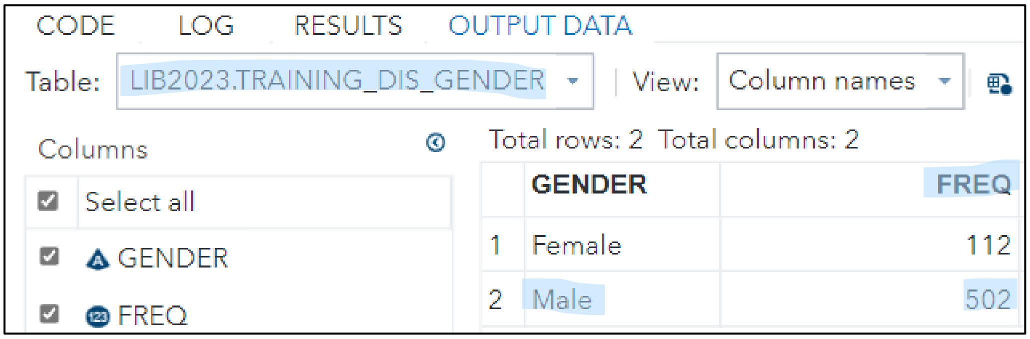
7.1.1 Categorical Variable

Gender

```
/* IMPUTE Gender */

/* Step 1: Make a copy of DS */
Proc SQL;
Create table LIB2023.TRAINING_GENDER_DS_BK AS
SELECT * FROM LIB2023.TRAINING_DS1;
QUIT;

/* Step 2: Find the statistics to get the MOD in Gender */
Proc SQL;
CREATE TABLE LIB2023.TRAINING_DIS_GENDER as
SELECT GENDER, COUNT(*) AS FREQ  FROM LIB2023.TRAINING_DS1
WHERE GENDER IS NOT NULL
GROUP BY GENDER;
QUIT;

/* Step 3: Find the MOD */
PROC SQL;
SELECT GENDER  FROM LIB2023.TRAINING_DIS_GENDER G
WHERE G.FREQ=(SELECT MAX(FREQ) FROM LIB2023.TRAINING_DIS_GENDER);
QUIT;

/* Step 4: Impute using the MOD */
PROC SQL;
UPDATE LIB2023.TRAINING_DS1
SET GENDER =( SELECT GENDER  FROM LIB2023.TRAINING_DIS_GENDER G
            WHERE G.FREQ=(SELECT MAX(FREQ) FROM LIB2023.TRAINING_DIS_GENDER)
            )
WHERE (GENDER EQ '') OR (GENDER IS NULL);
QUIT;

/*STEP 5: CHECK THE CHANGES*/
PROC SQL;
SELECT * FROM LIB2023.TRAINING_DS1
WHERE (GENDER IS NULL) OR (GENDER EQ "");
QUIT;
```



For step 1, the backup is created and the result is shown above. It is named as TRAINING_GENDER_DS_BK, since the variable that is involved is Gender.



For step 2, the table is created to record the distribution and to obtain the mode in the Gender variable. We can see that the mode in Gender is 'Male'.

GENDER
Male

Step 3: This is Mod that is selected from the Proc SQL code.

```
OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

PROC SQL;
UPDATE LIB2023.TRAINING_DS1
SET GENDER =( SELECT GENDER  FROM LIB2023.TRAINING_DIS_GENDER G
WHERE G.FREQ=(SELECT MAX(FREQ) FROM LIB2023.TRAINING_DIS_GENDER)
)
WHERE (GENDER EQ '') OR (GENDER IS NULL);
NOTE: No rows were updated in LIB2023.TRAINING_DS1.
```

Step 4: This is the output window of step 4, noticed that no rows were updated. This is because we have already imputed the missing values once, causing no missing values to be impute on the second try.

Family Members

```
/*******************************
IMPUTE FAMILY_MEMBERS
********************************/
/*Step 1. Count the missing values in FAMILY_MEMBERS */
PROC SQL;
Select Count(*) as No_of_family_missing from Training
where Family_Members is missing;

/*Step 2. Make a copy of the table to keep track of the number of observations for diff. groups*/
Create TABLE  LIB2023.TRAINING_FM_STAT_DS AS
Select Family_members, Count(*) as Counts
From Training
Where Family_members is not missing
Group by Family_members;

/*Shortcut  */
DATA Family_Members_DS;
Set  LIB2023.training_fm_stat_ds;

/*Step 3. Obtain the Mode  */
Select Family_members AS family_members
From Family_Members_DS
Where Counts=(
    Select Max(Counts) as Highest_Count
    From Family_Members_DS);

/*Step 4. Impute missing values with the mode  */
Update Training
Set Family_Members=(
    Select Family_members AS family_members
    From Family_Members_DS
    Where Counts=(
        Select Max(Counts) as Highest_Count  /* Subquery to find highest count in family members*/
        From Family_Members_DS))
Where ( Family_Members eq '');
QUIT;

/*Step 5. Check the imputation results */
** Using Step 1;
PROC SQL;
Select Count(*) as No_of_family_missing from Training
where Family_Members is missing;
```

Step 1:

No_of_family_missing
15

Step 2:

Table: LIB2023.TRAINING_FM_STAT_DS View: Column names

Columns

- Select all
- FAMILY_MEMBERS
- Counts

Total rows: 4 Total columns: 2

	FAMILY_ME...	Counts
1	0	345
2	1	102
3	2	101
4	3+	51

Step 3:

family_members
0

Step 4:

```
OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

PROC SQL;
Update Training
Set Family_Members=(
Select Family_members AS family_members
From Family_Members_DS
Where Counts=(
Select Max(Counts) as Highest_Count  /* Subquery to find highest count in family members*/
From Family_Members_DS))
Where ( Family_Members eq '');
NOTE: 15 rows were updated in WORK.TRAINING.

QUIT;
```

Step 5:

No_of_family_missing
0

The explanation of code is omitted since it quite similar for the GENDER var.

We noticed that 15 rows were updated, and this variable is successfully imputed.

Loan History

```
/*Impute LOAN_HISTORY*/

/* Step 1: List the details of missing values*/
TITLE 'STEP1: List the details of user who dont have loan_history';
PROC SQL;
Select * FROM Training
Where (LOAN_HISTORY EQ '') OR (LOAN_HISTORY IS NULL);
QUIT;

TITLE 'Count the total of user who dont have "LOAN_HISTORY"';
PROC SQL;
Select Count(*) LABEL='Number of applicants'
FROM Training
Where (LOAN_HISTORY EQ '') OR (LOAN_HISTORY IS NULL);
Quit;

/* Step 3: Find the statistics and store the statistics in the dataset */
PROC Sql;
Create TABLE  LIB2023.TRAINING_MS_STAT_DS as
Select LOAN_HISTORY as LOAN_HISTORY, Count(*) as Count
From Training
Where (LOAN_HISTORY Is Not Missing or LOAN_HISTORY ne '')
Group by LOAN_HISTORY;
Quit;

/* Step 4: Find the mod*/
PROC SQL;

Select LOAN_HISTORY as LOAN_HISTORY
From  LIB2023.TRAINING_MS_STAT_DS
WHERE Count eq ( Select Max(Count) Label = 'highest_count'
                    From  LIB2023.TRAINING_MS_STAT_DS);
```

```
/* Step 5: Make a backup copy of dataset- LIB2023.TRAINING_DS1 */
PROC SQL;
Create TABLE  LIB2023.Training_BK
as Select *
From Training;
Quit;

/* Step 6: Impute the missing values OF LOAN_HISTORY */
PROC SQL;
UPDATE Training
Set LOAN_HISTORY=(
    Select LOAN_HISTORY as Loan_History
    From  LIB2023.TRAINING_MS_STAT_DS
    WHERE Count eq ( Select Max(Count) Label = 'highest_count'
                     From  LIB2023.TRAINING_MS_STAT_DS))
WHERE ( LOAN_HISTORY eq '');
QUIT;

/* Step 7: Run Step 1 again to check the result*/
PROC SQL;
Select * FROM Training
Where (LOAN_HISTORY EQ '') OR (LOAN_HISTORY IS NULL);
QUIT;
```

Step 1:

ALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	L
ler Graduate	No	3596	0	100	240	.	City	Y
duate	No	3717	2925	151	360	.	Town	N
duate	No	4166	3369	201	360	.	City	N
duate	No	2400	0	75	360	.	City	Y
ler Graduate	Yes	3333	2166	130	360	.	Town	Y
duate	No	6000	2250	265	360	.	Town	N
ler Graduate	No	3333	2000	99	360	.	Town	Y
duate	No	6782	0	.	360	.	City	N
duate	No	2214	1398	85	360	.	City	Y
duate	No	3692	0	93	360	.	Village	Y
duate	No	6080	2569	182	360	.	Village	N
duate	Yes	20166	0	650	480	.	City	Y
duate	No	6000	0	160	360	.	Village	Y
duate	No	1916	5063	67	360	.	Village	N
duate	No	2383	2138	58	360	.	Village	Y
duate	No	3416	2816	113	360	.	Town	Y
duate	No	4283	2383	127	360	.	Town	Y
duate	No	5746	0	255	360	.	City	N
duate	Yes	3463	0	122	360	.	City	Y
ler Graduate	No	4931	0	128	360	.	Town	N
duate	No	6083	4250	330	360	.	City	Y
ler Graduate	No	4100	0	124	360	.	Village	Y
ler Graduate	No	7667	0	185	360	.	Village	Y
duate	Yes	5746	0	144	84	.	Village	Y
duate	No	2058	2134	88	360	.	City	Y
duate	No	3541	0	112	360	.	Town	Y
duate	No	3166	2985	132	360	.	Village	Y
duate	No	6333	4583	259	360	.	Town	Y

Count the total of user who dont have "LOAN_HISTORY"

Number of applicants
50

Step 3:

CODE LOG RESULTS OUTPUT DATA

Table: LIB2023.TRAINING_MS_STAT_DS View: Column names

Columns

- Select all
- LOAN_HISTORY
- Count

Total rows: 2 Total columns: 2

	LOAN_HISTORY	Count
1	0	89
2	1	475

Step 4:

LOAN_HISTORY
1

As can be seen from step 3, the mode for this categorical variable is 1.

Step 5:

Table: LIB2023.TRAINING_BK   View: Column names   Filter: (none)

Columns

- Select all
- SME_LOAN_ID_NO
- GENDER
- MARITAL_STATUS
- FAMILY_MEMBERS
- QUALIFICATION
- EMPLOYMENT
- CANDIDATE_INCOME
- GUARANTEE_INCOME
- LOAN_AMOUNT
- LOAN_DURATION
- LOAN_HISTORY
- LOAN_LOCATION
- LOAN_APPROVAL_STATUS

Total rows: 614 Total columns: 13   Rows 1-25(

	SME_LOAN...	GE...	MARITAL_S...	FAMILY_ME...	QUALIFIC...	EMPLO...	CAN
1	LP001002	Male	Not Married	0	Graduate	No	
2	LP001003	Male	Married	1	Graduate	No	
3	LP001005	Male	Married	0	Graduate	Yes	
4	LP001006	Male	Married	0	Under Gradua	No	
5	LP001008	Male	Not Married	0	Graduate	No	
6	LP001011	Male	Married	2	Graduate	Yes	
7	LP001013	Male	Married	0	Under Gradua	No	
8	LP001014	Male	Married	3+	Graduate	No	
9	LP001018	Male	Married	2	Graduate	No	
10	LP001020	Male	Married	1	Graduate	No	
11	LP001024	Male	Married	2	Graduate	No	
12	LP001027	Male	Married	2	Graduate	No	
13	LP001028	Male	Married	2	Graduate	No	
14	LP001029	Male	Not Married	0	Graduate	No	
15	LP001030	Male	Married	2	Graduate	No	

Step 6:

```
OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

PROC SQL;
UPDATE Training
Set LOAN_HISTORY=(
Select LOAN_HISTORY as Loan_History
From  LIB2023.TRAINING_MS_STAT_DS
WHERE Count eq ( Select Max(Count) Label = 'highest_count'
 From  LIB2023.TRAINING_MS_STAT_DS))
WHERE ( LOAN_HISTORY eq .);
NOTE: 50 rows were updated in WORK.TRAINING.
```

Employment

We will skip some of the codes and only show the outputs that are useful.

It is because the syntax is almost the same for categorical variables.

Count the total of user who dont have "Employment"

Number of applicants
0

The data scientist noticed that the no. of missing values for this variable is 0. It might be that the data scientist forgot to take the screenshot on the first attempt of imputation. Now, this missing values are all imputed and hence leaving no missing values for us to investigate.

Thus, the future steps will be omitted.

7.1.2 Continuous Variable

We will include the code for 1 of the variables only. Since the code is rather similar to each other, where we just edit the continuous var. that we want to study.

Loan Amount

```
/***********************************************************
IMPUTE Continuous Var. using Mean
***********************************************************/

/* LOAN_AMOUNT (Continuous) */

**Step 1. Count the empty values (if any);
PROC sql;
Select Count(*) Label='Number of loan applicants'
From Training
Where (Loan_amount = .);

**Step 2. Create Backup;
Create TABLE Training_BK as Select * from Training;

**Step 3. Impute missing value;
PROC STDIZE DATA=Training REPONLY  /*Replace only*/
METHOD = MEAN OUT =  LIB2023.TRAINING_DS1;
VAR loan_amount;
QUIT;

**Step 4. Check the results after imputation ;
TITLE 'List the details of the loan applicants who submitted their loan applications without loan amount';
PROC SQL;
SELECT *
FROM  LIB2023.TRAINING_DS1 t
WHERE ( t.loan_amount IS MISSING or t.loan_amount eq . );
QUIT;
```

Proc sql is mainly used to select rows with missing values, create backup tables, and see the results. While Proc STDIZE is used to deal with the missing values.

Number of loan applicants
22

Step 1 counts how many missing values for this var. .

Table: LIB2023.TRAINING_DS1 View: Column names Filter: (none)

Columns

- Select all
- SME_LOAN_ID_NO
- GENDER
- MARITAL_STATUS
- FAMILY_MEMBERS
- QUALIFICATION
- EMPLOYMENT
- CANDIDATE_INCOME
- GUARANTEE_INCOME
- LOAN_AMOUNT
- LOAN_DURATION
- LOAN_HISTORY
- LOAN_LOCATION

Total rows: 614 Total columns: 13 Rows 1-250

IE...	QUALIFIC...	EMPLO...	CANDIDATE_INCO...	GUARANTEE_INC...	LOAN_AMOUNT
	Graduate	No	5849	0	146.41216216
	Graduate	No	4583	1508	128
	Graduate	Yes	3000	0	66
	Under Gradua	No	2583	2358	120
	Graduate	No	6000	0	141
	Graduate	Yes	5417	4196	267
	Under Gradua	No	2333	1516	95
	Graduate	No	3036	2504	158
	Graduate	No	4006	1526	168
	Graduate	No	12841	10968	349
	Graduate	No	3200	700	70
	Graduate	No	2500	1840	109
	Graduate	No	3073	8106	200

This is the output of Step 3, where we can see that the missing values is imputed with the mean of the Loan amount.

List the details of the loan applicants who submitted their loan applications without loan amount

0

After the imputation, we find that there is no missing value in this variable.

Loan Duration

```
/* LOAN_DURATION (Continuous) */

**Step 1. Count the empty values (if any);
PROC sql;
Select Count(*) Label='Number of loan applicants'
From Training
Where (LOAN_DURATION = .);
```

Number of loan applicants
0

This means that there is no missing value to impute for Loan Duration. This might be due to the data scientist has made the first imputation without taking note about it.

7.2 Testing_DS

The screenshot of the coding will be omitted since it is similar to the training dataset, just that the dataset is switched to testing dataset. However, we will show the final screenshot showing that the data is successfully imputed.

7.2.1 Categorical Variable

Gender

```
OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

PROC SQL;
UPDATE LIB2023.TESTING_DS
SET GENDER =( SELECT GENDER  FROM LIB2023.TESTING_DIS_GENDER G
WHERE G.FREQ=(SELECT MAX(FREQ) FROM LIB2023.TESTING_DIS_GENDER)
)
WHERE (GENDER EQ '') OR (GENDER IS NULL);
NOTE: 11 rows were updated in LIB2023.TESTING_DS.
```

We see that a subquery is used to get the mode for Gender to impute in the missing values in the dataset. Besides, 11 rows are updated in TESTING_DS.

```
/*STEP 5: CHECK THE CHANGES*/
PROC SQL;
SELECT * FROM LIB2023.TESTING_DS
WHERE (GENDER IS NULL) OR (GENDER EQ "");
QUIT;
```

After checking the changes, we are sure that there is no missing values for this categorical variable.

Family Members

No_of_family_missing
10

```
OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

Proc SQL;
Update LIB2023.TESTING_DS
Set Family_Members=(
Select Family_members AS family_members
From Family_Members_DS
Where Counts=(
Select Max(Counts) as Highest_Count  /* Subquery to find highest count in family members*/
From Family_Members_DS))
Where ( Family_Members eq '');
NOTE: 10 rows were updated in LIB2023.TESTING_DS.
```

No_of_family_missing
0

After the imputation, the number of rows with missing value for family members is 0. This means that we imputed all rows for this IV(independent variable).

LOAN HISTORY

No_of_missing_LOAN_HISTORY
29

```
OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

PROC SQL;
Update LIB2023.TESTING_DS
Set LOAN_HISTORY=(
Select LOAN_HISTORY AS LOAN_HISTORY
From LOAN_HISTORY_DS
Where Counts=(
Select Max(Counts) as Highest_Count  /* Subquery to find highest count in family members*/
From LOAN_HISTORY_DS))
Where ( LOAN_HISTORY eq .);
NOTE: 29 rows were updated in LIB2023.TESTING_DS.
```

No_of_missing_LOAN_HISTORY
0

After the imputation, the number of rows with missing value for loan history is 0. This means that we imputed all rows for this IV(independent variable).

EMPLOYMENT

Count the total of user who dont have "Employment"

Number of applicants
23

```
OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

PROC SQL;
UPDATE LIB2023.TESTING_DS
Set Employment=(
Select Employment as Employment
From  LIB2023.TESTING_MS_STAT_DS
WHERE Count eq ( Select Max(Count) Label = 'highest_count'
 From  LIB2023.TESTING_MS_STAT_DS))
WHERE ( Employment eq '');
NOTE: 23 rows were updated in LIB2023.TESTING_DS.
```

Count the total of user who dont have "Employment"

Number of applicants
0

After the imputation, the number of rows with missing value for employment is 0. This means that we imputed all rows for this IV (independent variable).

7.2.2 Continuous Variable

LOAN_AMOUNT

Total missing values
5

```
/*Step 3. Impute missing value */
PROC STDIZE DATA=LIB2023.TESTING_DS REPONLY  /*Replace only*/
METHOD = MEAN OUT =  LIB2023.TESTING_DS;
VAR loan_amount;
QUIT;
```

List the details of the loan applicants who submitted their loan applications without loan amount

At first, we have 5 missing values for this IV, and after the imputation by mean, we see that there is no applicants who don't have the loan amount value.

LOAN_DURATION

Total missing values
6

```
/*Step 3. Impute missing value */
PROC STDIZE DATA=LIB2023.TESTING_DS REPONLY  /*Replace only*/
METHOD = MEAN OUT =  LIB2023.TESTING_DS;
VAR LOAN_DURATION;
QUIT;
```

List the details of the loan applicants who submitted their loan applications without LOAN_DURATION

At first, we have 5 missing values for this IV, and after the imputation by mean, we see that there is no applicants who don’t have the loan duration value.

Data Visualization and Prediction

Model implementation.

```
/*********************************************************
Building a Logistic Regression Model
*********************************************************/
PROC LOGistic DATA=  LIB2023.TRAINING_DS1 OUTMODEL= LIB2023.TRAINING_DS1_LR_MODEL;

/*categorical  */
CLASS
    Gender
    Marital_Status
    FAMILY_MEMBERS
    QUALIFICATION
    EMPLOYMENT
    LOAN_HISTORY
    LOAN_LOCATION;

MODEL LOAN_APPROVAL_STATUS =    /*DV*/
    GENDER
    MARITAL_STATUS
    FAMILY_MEMBERS
    QUALIFICATION
    EMPLOYMENT
    CANDIDATE_INCOME
    GUARANTEE_INCOME
    LOAN_AMOUNT
    LOAN_DURATION
    LOAN_HISTORY
    LOAN_LOCATION
    /* Above all are independent variables */
    ;
OUTPUT OUT = LIB2023.TRAINING_OUT_DS P = PPRED_PROB;
/*PRED_PROB ->Predicted probability - variable to hold predicted probability */

RUN;
```

Explanation:

PRED_PROB ->Predicted probability - variable to hold predicted probability.

OUT -> the output will be stored in the dataset

The LOGISTIC Procedure

Model Information	
Data Set	LIB2023.TRAINING_DS1
Response Variable	LOAN_APPROVAL_STATUS
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	614
Number of Observations Used	614

Response Profile		
Ordered Value	LOAN_APPROVAL_STATUS	Total Frequency
1	N	192
2	Y	422

Probability modeled is LOAN_APPROVAL_STATUS='N'.

The dataset has no missing value issue since the num of obs. Used equals to num of obs. Read.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	764.891	585.168
SC	769.311	633.788
-2 Log L	762.891	563.168

AIC < SC

AIC (Akaike Information Criterion) < SC (Schwarz Criterion), hence it is a good fit model.

Type 3 Analysis of Effects			
Effect	DF	Wald Chi-Square	Pr > ChiSq
GENDER	1	0.0100	0.9204
MARITAL_STATUS	1	5.3173	0.0211
FAMILY_MEMBERS	3	4.3866	0.2226
QUALIFICATION	1	2.4952	0.1142
EMPLOYMENT	1	0.0060	0.9384
CANDIDATE_INCOME	1	0.2268	0.6339
GUARANTEE_INCOME	1	2.2688	0.1320
LOAN_AMOUNT	1	1.4294	0.2319
LOAN_DURATION	1	0.5322	0.4657
LOAN_HISTORY	1	87.4798	<.0001
LOAN_LOCATION	2	12.0908	0.0024

≤ 0.05

it indicates good IV.

Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	0.0495	0.6972	0.0050	0.9434
GENDER	Female	1	-0.0149	0.1495	0.0100	0.9204
MARITAL_STATUS	Married	1	-0.2915	0.1264	5.3173	0.0211
FAMILY_MEMBERS	0	1	-0.0394	0.1863	0.0447	0.8326
FAMILY_MEMBERS	1	1	0.4319	0.2258	3.6572	0.0558
FAMILY_MEMBERS	2	1	-0.3310	0.2538	1.6998	0.1923
QUALIFICATION	Graduate	1	-0.2052	0.1299	2.4952	0.1142
EMPLOYMENT	No	1	-0.0123	0.1586	0.0060	0.9384
CANDIDATE_INCOME		1	-0.00001	0.000024	0.2268	0.6339
GUARANTEE_INCOME		1	0.000053	0.000035	2.2688	0.1320
LOAN_AMOUNT		1	0.00191	0.00160	1.4294	0.2319
LOAN_DURATION		1	0.00134	0.00184	0.5322	0.4657
LOAN_HISTORY	0	1	1.9696	0.2106	87.4798	<.0001
LOAN_LOCATION	City	1	0.1559	0.1519	1.0538	0.3046
LOAN_LOCATION	Town	1	-0.5313	0.1575	11.3806	0.0007

If Pr > ChiSq is <= 0.05, it means that that IV has impact on the model and as is significant in predicting the dependent variable. Here, Marital Status, Loan history and loan location are the most significant variables in predicting the DV (Dependent variable). Gender and Employment have a low impact on loan approval status, since the Pr > ChiSq value is very high.

Then, we will use the model created to do prediction on the testing_ds. As shown below:

```
/* Predict the loan approval status using the model created */

PROC LOGISTIC INMODEL =  LIB2023.TRAINING_DS1_LR_MODEL;/* Model that is created */
SCORE DATA= LIB2023.TESTING_DS /* Enter with the Testing Dataset */
OUT= LIB2023.TESTING_LAS_PREDICTED_DS; /* Location of output */
QUIT;
```

Table: LIB2023.TESTING_LAS_PREDICTED_DS View: Column names Filter: (none)

Total rows: 367 Total columns: 17 Rows 1-250

RATION	LOAN_HISTORY	LOAN_LO...	LOAN_APPROVA...	F_LOAN_APPROV...	I_LOAN_APPROVA...	P_N	P_Y
360	1	City			Y	0.1582296819	0.8417703181
360	1	City			Y	0.2574444898	0.7425555102
360	1	City			Y	0.1581934212	0.8418065788
360	1	City			Y	0.1408937614	0.8591062386
360	1	City			Y	0.3293748248	0.6706251752
360	1	City			Y	0.2822197578	0.7177802422
360	1	Town			Y	0.2727031952	0.7272968048
360	0	Village			N	0.9369203432	0.0630796568
240	1	City			Y	0.1311825475	0.8688174525
360	1	Town			Y	0.2360088939	0.7639911061
360	1	City			Y	0.3349456446	0.6650543554
360	1	Town			Y	0.1589053773	0.8410946227
180	1	City			Y	0.1872907809	0.8127092191
360	0	Town			N	0.7893551358	0.2106448642
360	1	Town			Y	0.1459699007	0.8540300993
360	1	City			Y	0.3597430379	0.6402569621
360	1	City			Y	0.1647880759	0.8352119241
360	1	Town			Y	0.0902882227	0.9097117773
360	1	City			Y	0.2831176954	0.7168823046
180	1	Town			Y	0.141781075	0.858218925
360	1	City			Y	0.3147450865	0.6852549135
180	1	City			Y	0.2523766766	0.7476233234
360	1	City			Y	0.252617736	0.747382264
360	1	City			Y	0.3414946076	0.6585053924
360	1	City			Y	0.251406785	0.748593215
360	0	Village			N	0.9888129225	0.0111870775
360	1	City			Y	0.1390035789	0.8609964211
360	1	City			Y	0.2354740399	0.7645259601
360	1	Town			Y	0.0836891009	0.9163108991

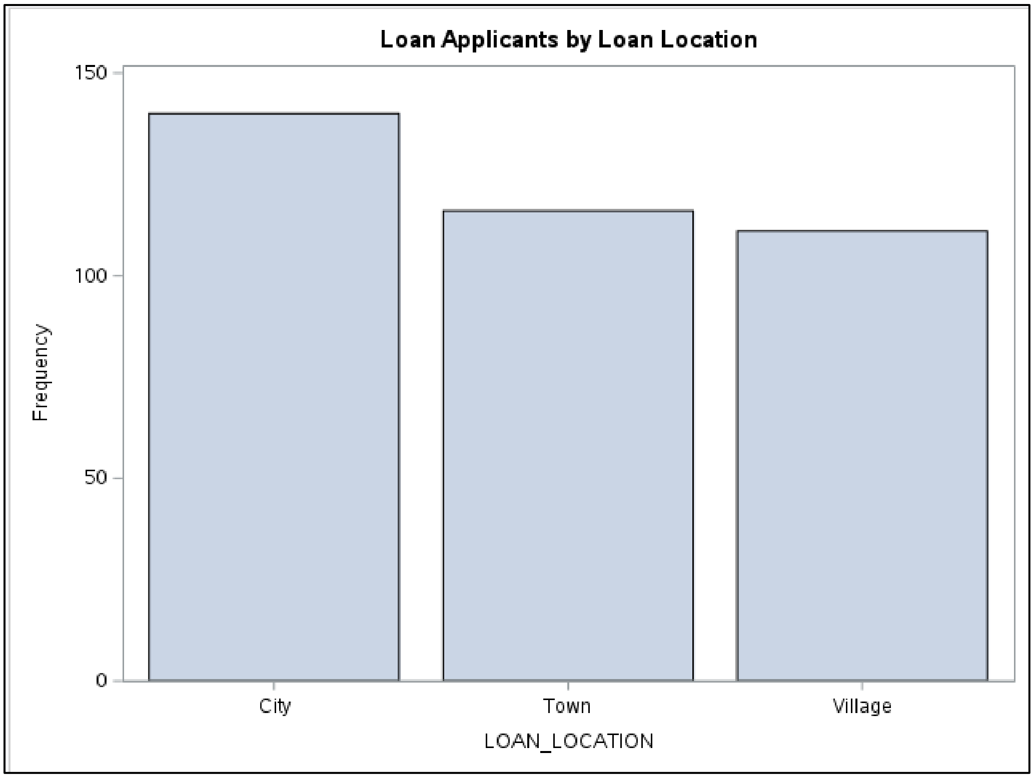
The top left corner shows the name of the dataset. The output of the loan approval status is formed (as shown in the highlighted rectangle). The probability that is used to determine the Y(accepted) and N(rejected) is also formed, shown beside the approval status. We can see that if P_Y that is greater than 0.5, will have a Y in the approval status, for P_N, vice versa.

Data Visualization

Sas CODE Screenshot

```
/* Simple barchart */
PROC SGPLOT DATA = LIB2023.TESTING_LAS_PREDICTED_DS;
VBAR loan_location;
TITLE 'Loan Applicants by Loan Location';
RUN;
```

Screenshot of the Chart



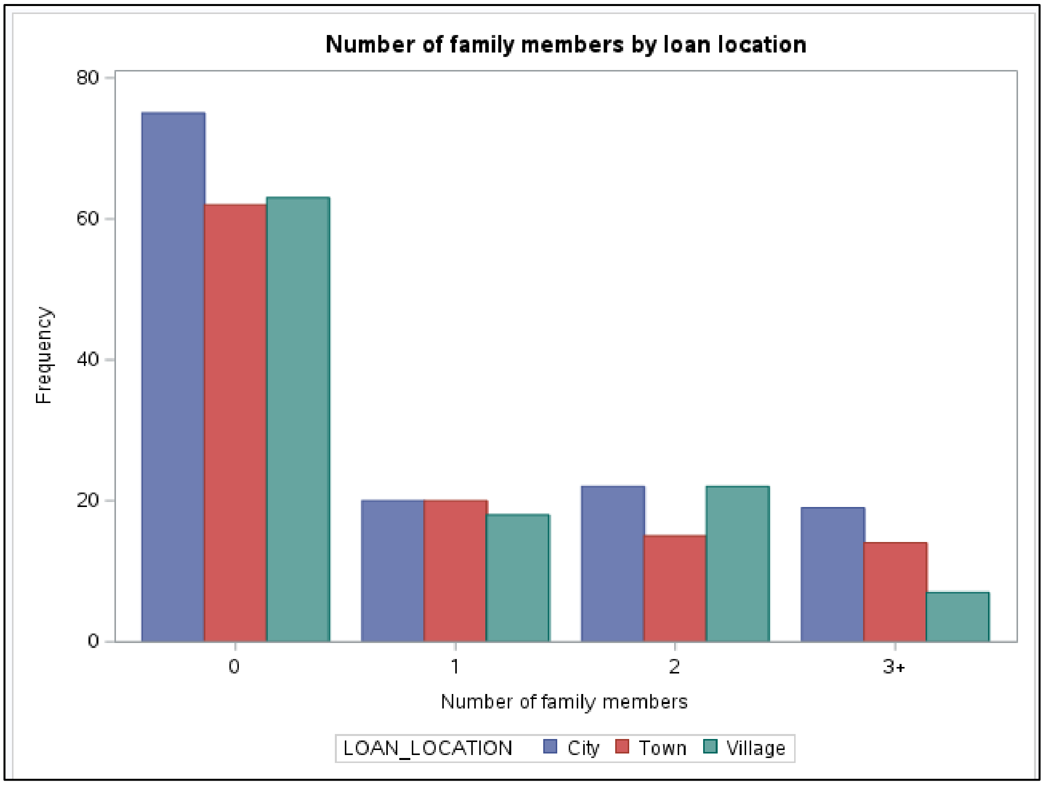
Description:

The mod is city, means that most applicants are from the city region. Least applicants are from the village region.

Code

```
/* Stacked bar chart */
Title 'Number of family members by loan location';
Proc sgplot data= LIB2023.Testing_las_predicted_ds;

vbar family_members / group = loan_location  groupdisplay=cluster;
label family_members ='Number of family members';
run;
```

Chart



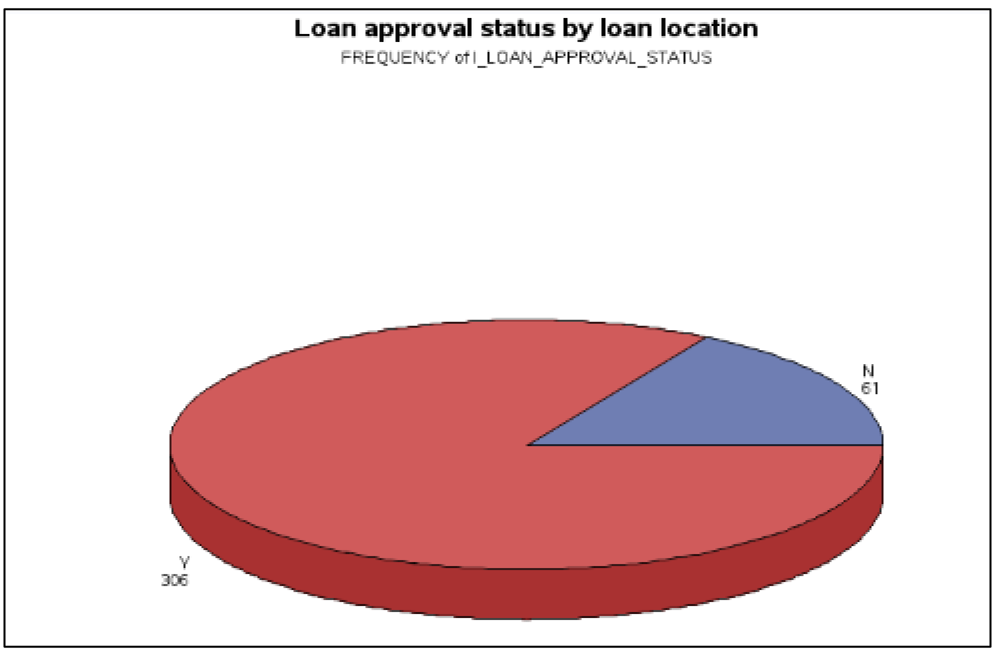
Description

Most of the applicants have 0 family members, despite the location they live. The smaller frequency of applicants are with 3 or more family members despite the location.

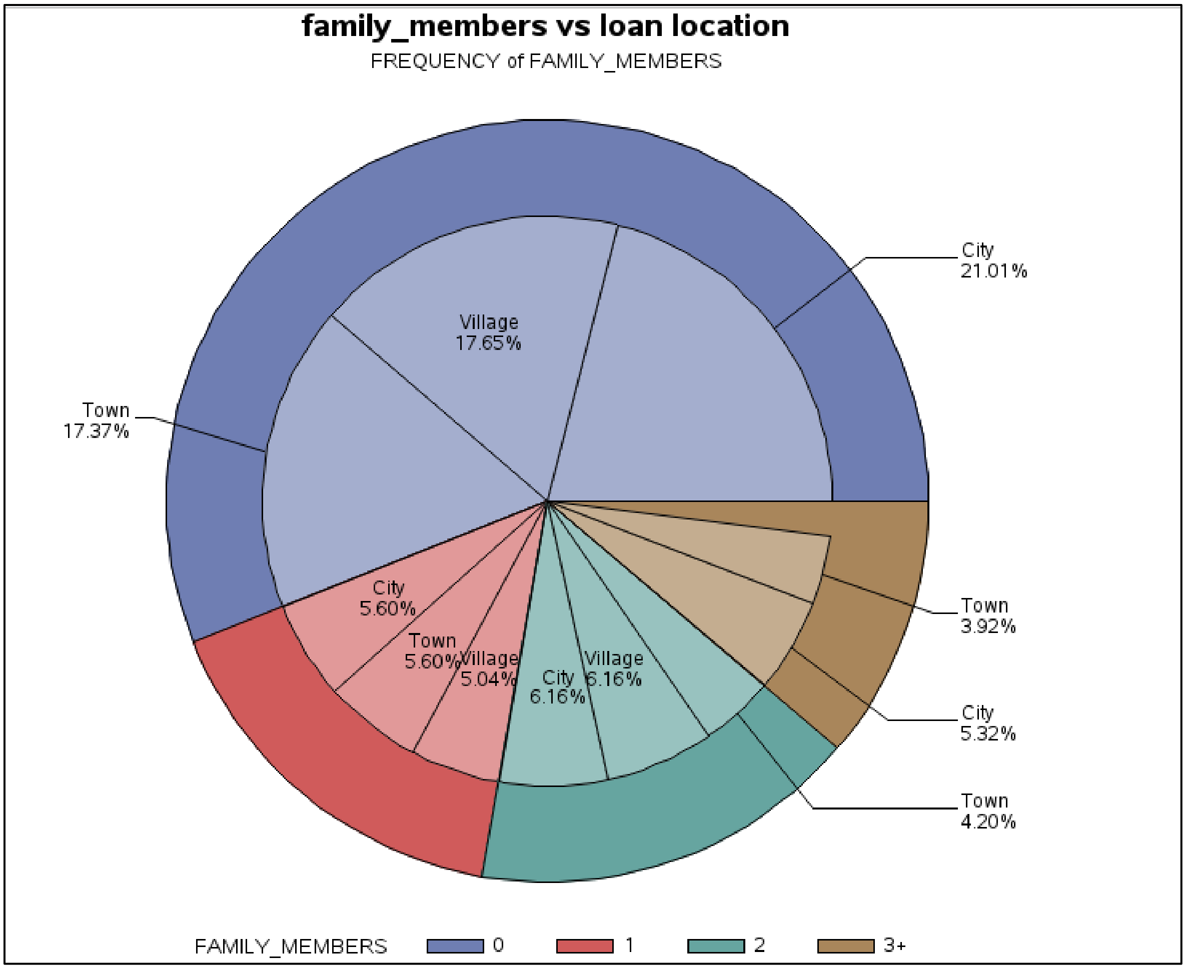
SAS Code

```
/* Pie Chart 3D*/
TITLE 'Loan approval status by loan location';

PROC GCHART data = Lib2023.TESTING_LAS_PREDICTED_DS;
pie3d I_LOAN_APPROVAL_STATUS;
RUN;
QUIT;

/* Pie Chart 2D */
GOPTIONS RESET=ALL BORDER;
TITLE 'family_members vs loan location';
PROC GCHART DATA=Lib2023.TESTING_LAS_PREDICTED_DS;
pie family_members / detail=loan_location
detail_percent=best
detail_value=none
detail_slice=best
detail_threshold=2
legend;
RUN;
```

SAS Screenshot



61 applicants are rjected, while 306 applicants are approved. This is a 83.4% approval rate.



Description

This shows the distribution of applicants in term of family members and loan location. Most applicants are with no family members, despite they are from city, village or town. Which contributes to 21%, 17.7% and 17.3% among all aplicants.

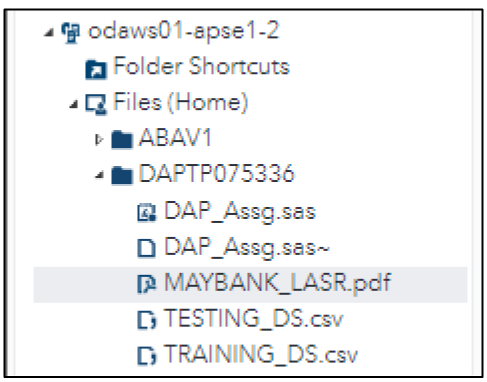
SAS Code

SAS Screenshot

Description

Report generation (Output)

```
/*************************************************************
Generate report using SAS ODS - Output Delivery System
*************************************************************/

ODS HTML CLOSE;
ODS PDF CLOSE;
/* Determine the physical location of pdf */
ODS PDF FILE = "/home/u63525503/DAPTP075336/MAYBK_LASR.pdf";
OPTIONS NODATE;
TITLE1 'Bank Loan Approval Status Predicted';
TITLE2 'APU,TPM';
PROC REPORT DATA =  LIB2023.TESTING_LAS_PREDICTED_DS NOWINDOWS;
BY SME_LOAN_ID_NO;
DEFINE SME_LOAN_ID_NO / GROUP 'LOAN ID';
DEFINE GENDER / GROUP 'GENDER NAME';
DEFINE MARITAL_STATUS / GROUP 'MARITAL STATUS';
DEFINE FAMILY_MEMBERS / GROUP 'FAMILY MEMBERS';
DEFINE CANDIDATE_INCOME / GROUP 'MONTHLY INCOME';
DEFINE GUARANTEE_INCOME / GROUP "CO-APPLICANT'S INCOME";
DEFINE LOAN_AMOUNT / GROUP 'LOAN AMOUNT';
DEFINE LOAN_DURATION / GROUP 'LOAN DURATION';
DEFINE LOAN_HISTORY / GROUP 'LOAN HISTORY';
DEFINE LOAN_LOCATION / GROUP 'LOAN LOCATION';
FOOTNOTE '---End of Report----';

RUN;
```



---End of Report----

Bank Loan Approval Status Predicted
APU,TPM 2

SME_LOAN_ID_NO=LP001022

LOAN ID	GENDER NAME	MARITAL STATUS	FAMILY MEMBERS	QUALIFICATION	EMPLOYMENT	MONTHLY INCOME	CO-APPLICANT'S INCOME	LOAN AMOUNT
LP001022	Male	Married	1	Graduate	No	3076	1500	126

LOAN DURATION	LOAN HISTORY	LOAN LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS
360	1	City		

Into: LOAN_APPROVAL_STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
Y	0.2574445	0.7425555

This shows pg 2 of the MAYBANK_LASR.pdf, we can see a clear detail about how the Logit model is working in behind, by accessing the Predicted Probability, we know that the model is doing a good work without making mistakes.

Besides, all details of the applicants is shown too, this helps the manager to make detail comparison between applicants.

```
/**********************************************************
Generate report carrying the loan approval status (without using SAS ODS)
**********************************************************/

/********************************************************************************
STEP 1: Sort the data found in the dataset - LIB51510.TESTING_LAS_PREDICTED_DS
********************************************************************************/

OPTIONS NODATE;

PROC SORT DATA = LIB2023.TESTING_LAS_PREDICTED_DS OUT = LIB2023.TESTING_LAS_PREDICTED_SORTED_DS;

BY loan_location
   sme_loan_id_no;
RUN;

/*********************************************************************
STEP 2: List the details of the data sorted
*********************************************************************/

PROC SQL;

SELECT *
FROM LIB2023.TESTING_LAS_PREDICTED_SORTED_DS;

QUIT;

PROC SQL;

SELECT COUNT(*)
FROM LIB2023.TESTING_LAS_PREDICTED_SORTED_DS
where into:LOAN_APPROVAL_STATUS eq '';
QUIT;
/*********************************************************************
STEP 3: Generate the report
*********************************************************************/

PROC PRINT DATA = LIB2023.TESTING_LAS_PREDICTED_SORTED_DS SPLIT = '*';
id loan_location;
by loan_location;
var sme_loan_id_no
     candidate_income
     loan_amount
     loan_duration
     i_loan_approval_status;
sum candidate_income loan_amount;
label loan_location = 'LOAN LOCATION*======='
       sme_loan_id_no = 'LOAN ID*======='
       candidate_income = 'CANDIDATE INCOME*================'
       loan_amount = 'LOAN AMOUNT*==========='
       loan_duration = 'LOAN DURATION*============='
       i_loan_approval_status ='LOAN APPROVAL STATUS*====================';
TITLE1 'Bank Loan Approval Status Predicted';
TITLE2 'MAYBANK,TPM';

RUN;
```

The code sorted the output for loan location, which is by the alphabet of location i.e., city, town and then village. Then, the observations are sorted via the sme_loan_id_no. As shown in this 2 screenshots below, where we can see all city entries are grouped together (Fig 3), while in Figure 4, we see that the id is grouped from small to large for a location (grey), then followed by the id of another location (blue).

GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
0	110	360	1	City	
1500	126	360	1	City	
1800	208	360	1	City	
2546	100	360	1	City	
0	78	360	1	City	
3422	152	360	1	City	
0	280	240	1	City	
0	90	360	1	City	
0	40	180	1	City	
0	131	360	1	City	
2916	200	360	1	City	
7916	300	360	1	City	
1620	48	360	1	City	
0	28	180	1	City	
0	101	360	1	City	
0	125	360	1	City	
4380	290	360	1	City	
1250	140	360	1	City	
3750	275	360	1	City	
2382	125	180	1	City	
820	192	360	1	City	
2708	158	360	1	City	
1541	101	360	1	City	
4029	185	180	1	City	
2792	90	360	1	City	
0	116	360	1	City	
1963	138	360	1	City	
818	100	360	1	City	
0	110	360	1	City	
0	84	360	1	City	
3900	185	342.53739612	1	City	
1475	162	360	1	City	
3338	187	342.53739612	1	City	
1707	124	360	1	City	
1000	30	180	1	City	
0	92	360	1	City	
0	130	360	0	City	
292	125	360	1	City	
0	125	360	1	City	

Figure 1

LP002850	Male	Not Married	2	Graduate	No	2400
LP002853	Female	Not Married	0	Under Graduate	No	3015
LP002856	Male	Married	0	Graduate	No	2292
LP002870	Male	Married	1	Graduate	No	4700
LP002878	Male	Married	3+	Graduate	No	8334
LP002885	Male	Not Married	0	Under Graduate	No	2868
LP002890	Male	Married	2	Under Graduate	No	3418
LP002907	Male	Married	0	Graduate	No	5817
LP002932	Male	Married	3+	Graduate	No	7603
LP002935	Male	Married	1	Graduate	No	3791
LP002952	Male	Not Married	0	Graduate	No	2500
LP002965	Female	Married	0	Graduate	No	8550
LP002971	Male	Married	3+	Under Graduate	Yes	4009
LP002975	Male	Married	0	Graduate	No	4158
LP001055	Female	Not Married	1	Under Graduate	No	2226
LP001067	Male	Not Married	0	Under Graduate	No	2400
LP001082	Male	Married	1	Graduate	No	2185
LP001094	Male	Married	2	Graduate	No	12173
LP001096	Female	Not Married	0	Graduate	No	4666
LP001107	Male	Married	3+	Graduate	No	3786
LP001115	Male	Not Married	0	Graduate	No	1300
LP001174	Male	Married	0	Graduate	No	3772
LP001177	Female	Not Married	0	Under Graduate	No	2478
LP001185	Male	Not Married	0	Graduate	No	3268
LP001203	Male	Not Married	0	Graduate	No	3150
LP001226	Male	Married	0	Under Graduate	No	1750
LP001230	Male	Not Married	0	Graduate	No	6500
LP001242	Male	Not Married	0	Under Graduate	No	2356
LP001270	Male	Married	3+	Under Graduate	Yes	8000
LP001287	Male	Married	3+	Under Graduate	No	3500
LP001291	Male	Married	1	Graduate	No	3500

Figure 2

Bank Loan Approval Status Predicted
MAYBANK,TPM

LOAN LOCATION =======	LOAN ID =======	CANDIDATE INCOME ================	LOAN AMOUNT ===========	LOAN DURATION =============	LOAN APPROVAL STATUS ====================
City	LP001015	5720	110	360	Y
	LP001022	3076	126	360	Y
	LP001031	5000	208	360	Y
	LP001035	2340	100	360	Y
	LP001051	3276	78	360	Y
	LP001054	2165	152	360	Y
	LP001059	13633	280	240	Y
	LP001078	3091	90	360	Y
	LP001083	4166	40	180	Y
	LP001099	5667	131	360	Y
	LP001105	4583	200	360	Y
	LP001108	9226	300	360	Y
	LP001121	1888	48	360	Y
	LP001124	2083	28	180	Y
	LP001128	3909	101	360	Y
	LP001135	3765	125	360	Y
	LP001149	5400	290	360	Y
	LP001163	4363	140	360	Y
	LP001169	7500	275	360	Y
	LP001176	2942	125	180	Y
	LP001183	6250	192	360	Y
	LP001187	2783	158	360	Y
	LP001190	2740	101	360	Y
	LP001208	7350	185	180	Y
	LP001210	2267	90	360	Y
	LP001211	5833	116	360	Y
	LP001219	3643	138	360	Y
	LP001220	5629	100	360	Y
	LP001221	3644	110	360	Y
	LP001231	3666	84	360	Y
	LP001232	4260	185	342.53739612	Y
	LP001237	4163	162	360	Y
	LP001268	6792	187	342.53739612	Y
	LP001284	2419	124	360	Y
	LP001298	4116	30	180	Y
	LP001312	5293	92	360	Y
	LP001313	2750	130	360	N
	LP001335	7016	125	360	Y
	LP001348	4490	125	360	Y
	LP001375	4083	139	60	Y
	LP001400	3583	155	360	Y

LOAN LOCATION =======	LOAN ID =======	CANDIDATE INCOME ================	LOAN AMOUNT ===========	LOAN DURATION =============	LOAN APPROVAL STATUS ====================
Town	LP001055	2226	59	360	Y
	LP001067	2400	123	360	Y
	LP001082	2185	162	360	Y
	LP001094	12173	166	360	N
	LP001096	4666	124	360	Y
	LP001107	3786	126	360	Y
	LP001115	1300	100	180	Y
	LP001174	3772	57	360	Y
	LP001177	2478	75	360	Y
	LP001185	3268	152	360	Y
	LP001203	3150	176	360	N
	LP001226	1750	90	360	Y
	LP001230	6500	200	360	Y
	LP001242	2356	108	360	Y
	LP001270	8000	187	360	Y
	LP001287	3500	120	360	Y
	LP001291	3500	160	360	Y
	LP001321	3613	134	180	Y
	LP001323	2779	176	360	N
	LP001324	4720	90	180	Y
	LP001332	2415	110	360	Y

LOAN LOCATION =======	LOAN ID =======	CANDIDATE INCOME ================	LOAN AMOUNT ===========	LOAN DURATION =============	LOAN APPROVAL STATUS ====================
Village	LP001056	3881	147	360	N
	LP001153	0	148	360	N
	LP001317	4402	130	360	Y
	LP001347	2101	108	360	N
	LP001361	2458	188	360	N
	LP001380	3900	232	360	Y
	LP001413	6356	50	360	Y
	LP001445	4136	149	480	N
	LP001446	8449	257	360	Y
	LP001452	4635	102	180	Y
	LP001472	5058	200	360	Y
	LP001475	3188	130	360	Y
	LP001483	13518	390	360	Y
	LP001534	4452	131	360	Y
	LP001548	2687	50	180	Y
	LP001567	4513	120	360	Y
	LP001599	4167	160	360	Y
	LP001611	1516	80	342.53739612	N
	LP001622	724	213	360	N
	LP001650	2333	146	360	Y
	LP001652	2500	187	360	N
	LP001718	3391	132	360	Y
	LP001728	3343	105	360	Y
	LP001742	4500	147	360	Y
	LP001757	2014	120	360	Y
	LP001785	4727	150	360	N
	LP001787	3089	100	240	Y
	LP001794	10890	260	12	Y
	LP001817	8703	199	360	N

From here, we see the final output which sorts the data based on location, followed by loan ID. The loan ID is sorted in ascending order too, and the predicted Dependent Variable (DV) can be clearly seen in figure above.

The "====" used in the figure above is generated using the label function in sas, where we label IV as "IV*===". This forms the beautiful divider for the variables.