CT050-3-M – DATA ANALYTICAL PROGRAMMING (DAP)

MAY-2021-DSBA

SEP 2021

INDIVIDUAL ASSIGNMENT

Date Assigned : Thu-7-Oct-2021

Name: ONG WEI AUN (TP063332)

Lecturer: MR. DHASON PADMA KUMAR

Table of Contents

1. INTRODUCTION

Every day, there are companies and individuals that are borrowing money from banks or other financial institutions to finance their activities like a person buying a car or a business buying new machinery equipment to expand their factory. The borrower thereby incurs a debt, which they must repay with interest and within a specified time frame. However, banks are worried of potential loan defaults, thus they do not approve loan applications easily unless they are confident that the borrowers are able to repay the loan. Loan defaults are also known as non-performing loan (NPL), where the borrower did not pay to the bank for 90 days or more.

Therefore, every bank always tries to identify the risk of NPL from the very beginning. However, when the banks avoid risk too much, it can cause income loss from the customers who are able to repay the loans. This is where a good tool is needed to analyze a customer and identify whether they are good or bad customers. Bank loan application filtration process are time consuming and may increase the risk of misidentification.

Nowadays, artificial intelligence (AI) is a rapidly developing technology. AI are now widely used in solving many real-world challenges. Machine learning is a type of AI that is particularly beneficial in prediction systems. Machine learning builds a model based on training data and makes prediction. The algorithm trains the system with a small portion of data and test it with the remaining data. Similarly, this algorithm can be utilized to help banks to analyze the applications to help the banks make better decisions in approving loan applications.

The aim of this assignment is to analyze past data set obtained from past customers and build a most accurate model to predict the approval process as approved or rejected. In the work flow process, a dataset sample will be collected from past history to study the customers backgrounds. Data exploration and modification will be done where necessary before building a model to predict the outcome of loan approval status.

2. BACKGROUND STUDY

The company chosen in this assignment is Lasiandra Finance Inc. (LFI) which is located in New York, USA. It is a leading private financing company which provides funding to Small and Medium Enterprises (SME). By making their loaning process tailor-made and suitable to the customers, they are able to give those business dreams injection of boost. Through the development of internet, it has tremendously increased their business expansion and provide more funding to more SMEs. In order to speed up the process of loan approval, it needs automation to help process loan eligibility based on the customer portfolio entered online. However, loan approval process is complicated as it requires a lot of verification and validation so that they can give the loans to the most deserving applicants and reduce loan default rate.

3. ASSUMPTION & JUSTIFICATION

It is assumed that the dataset used in this study are actual data from loan applicants to help support the accuracy of the model run in this study.

4. LITERATURE REVIEW

This literature review introduces investigation and discussion for related work on reducing risk for non-performing loans (NPL) or also known as loan defaults.

4.1 Credit Scoring model

In the past works by other researchers, they have used different methods as credit scoring assessment to evaluate whether applications should be approved or not. Imtiaz & J. used variables like gender, education, marital status, age and past payment records to be input their machine learning models as credit risk assessment. (Imtiaz & J., 2017) Other than that, Chen & Xiang even included more detailed variables like loan purpose, loan amount, employment duration, debt service ratio etc. to help filter the applications during risk assessment process. (Chen & Xiang, 2017)

4.2 Applications of Machine Learning in Loan Defaults risk assessment

Thavarith & Liangrokapart (2019) did study on finding the rootcause of NPL and suggested a method about reducing the risks of NPL based on the data from one of the largest banks in Cambodia. They combined the application of both Six Sigma and credit scoring model to help better filter potential clients to reduce the risk of loan defaults. FMEA method are used to analyze the risk priority number for the potential failure causes. Besides, through the credit scoring model, they classified customers into 4 different risk classes. By blending both six sigma and credit scoring model, they are able to reduce the level of RPN by 32.5%, which is from 1446 to 975. (Thavarith & Liangrokapart, 2019)

Furthermore, Coşer et al investigates a database of customers who were unable to repay their loans and got into loan defaults. Predictive models like LightGBM, XGBoost, Logistic

Regression and Random Forest were used to calculate the probability of a customer loan turning default status. Model comparison were done to identify the best model by considering the model performance metrics like AUC score, precision, recall and accuracy. The best results obtained was using the Random Forest model which has a representative AUC of 0.89. (Coşer et al., 2019)

Besides, Figini et al. observe the credit risk in small and medium enterprises by using boosting, bagging and random forest. Multivariate outlier detection techniques like Local Outlier Factor (LOF) were mentioned in the study. Unlike univariate outlier detection techniques, the LOF technique is a multivariate technique which is consistent on high dimensional data without resorting to strong assumptions about the distribution. Thus, the authors proposed to improve the out of sample performance of parametric and non-parametric models for credit risk estimation. (Figini et al., 2017)

Other than that, Butaru et al. apply machine learning techniques to predict delinquency in the credit card industry. The authors combined consumer tradeline, credit bureau and macroeconomic variables as part of the model's prediction. Besides, they also found out that decision trees and random forests outperform the logistic regression method in forecasting the credit card delinquencies. (Butaru et al., 2016)

On the other hand, Sudhamathy used decision tree model in R package to help analyze the credibility of the bank loans applicants. They find the correlation between features and rank the features according to importance before building a decision tree model. (Sudhamathy, 2016)

Chen & Xiang also constructed a credit scoring model based on Group Lasso Logistic Regression to manage credit risks. Lasso (Least Absolute Shrinkage and Selection Operator) regression performs both variable selection and regularization to enhance the predictive accuracy and interpretability of the statistical model it produces. For variable selection, the selection of tuning parameter λ is very important and usually the Akaike Information Criterion(AIC), Bayesian Information Criterion(BIC) and Cross Validation prediction errors will determine the tuning parameter. The final results indicated that the Group Lasso method is better than backward elimination in both interpretability and prediction accuracy. (Chen & Xiang, 2017)

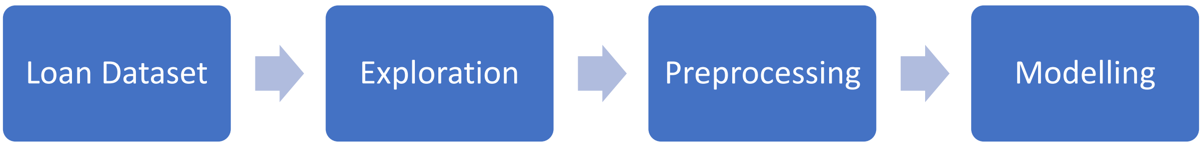
5. DATA EXPLORATION

The dataset used in this assignment contains 614 observations and 13 different variables.

Name of variable	Description	Data Type	Length	Sample Data
SME_LOAN_ID_NO	Loan application number	Char	8	LP001002/LP001003
GENDER	Gender of the applicant	Char	6	Female; Male
MARITAL_STATUS	Marital Status of the applicant; Married or Not Married	Char	11	Married; Not Married
FAMILY_MEMBERS	Number of family members of the applicant	Char	2	1, 2, 3+
QUALIFICATION	Education Qualification of the applicant	Char	14	Graduate; Under Graduate
EMPLOYMENT	Employment Status of the applicant	Char	3	Yes; No
CANDIDATE_INCOME	Income of the applicant	Numeric	5	5849, 4583, 3000
GUARANTEE_INCOME	Income of Joint Applicant	Numeric	5	1508, 2358, 4196
LOAN_AMOUNT	Loan amount	Numeric	5	128,66,120

LOAN_DURATION	Duration of Loan Tenure	Numeric	3	71,360
LOAN_HISTORY	Loan History of the applicant	Numeric	1	0; 1
LOAN_LOCATION	Location of the application	Char	7	City, Village, Town
LOAN_APPROVAL_STATUS	Approval Status of Loan Application	Char	1	Y; N (Y=Yes, N=No)

6. METHODOLOGY

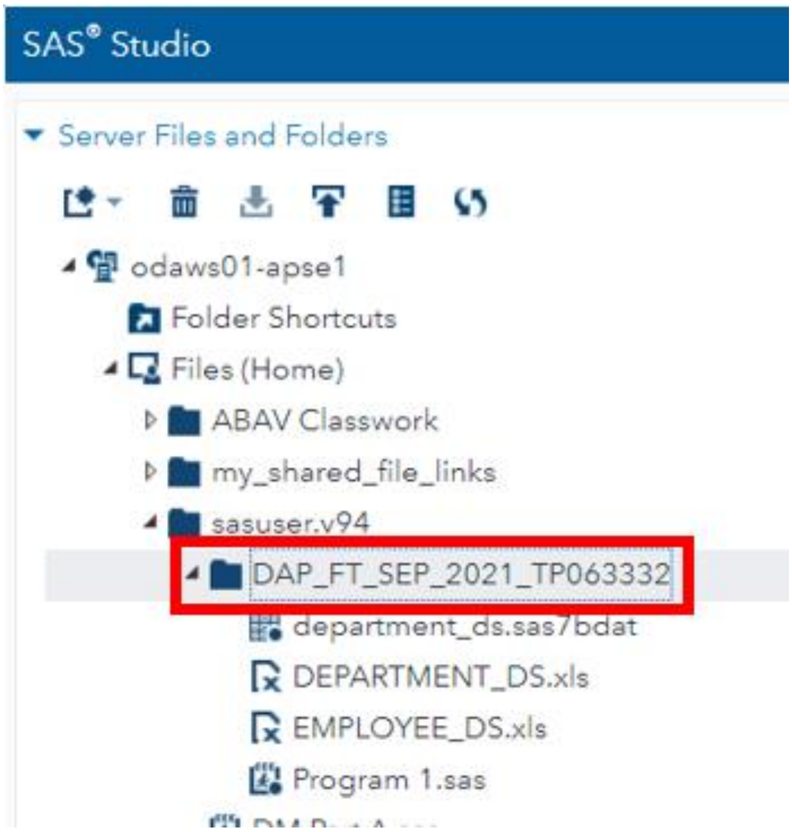


First of all, a dataset consisting of the details of the applicants is obtained. Univariate and bivariate analysis are done on the variables to explore the data. Pre-processing and imputation are done to the missing data in the dataset. Finally, a logistic regression model is created to determine the loan eligibility of the applicants.

7. EXPERIMENTATION

7.1 Create a folder on SAS

<u>Screenshot(s)</u>

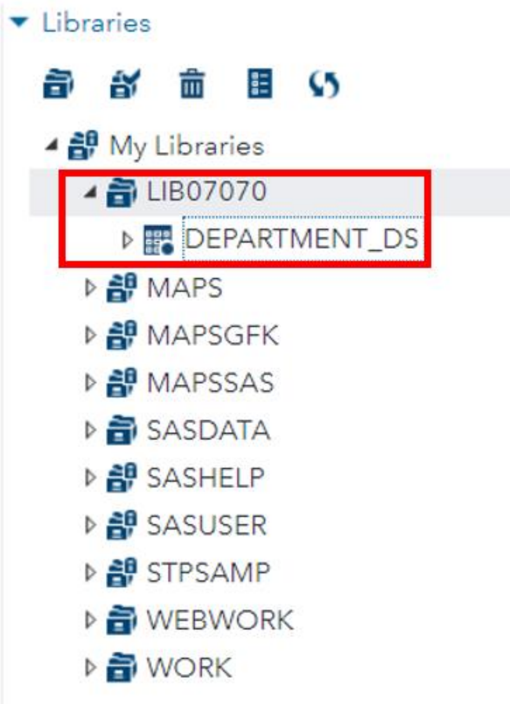


<u>Explanation</u>

- A new folder is created to store all the datasets and SAS programs for the study so that it will not be mixed up with all the other projects.

7.2 Create a permanent library on SAS

Screenshot

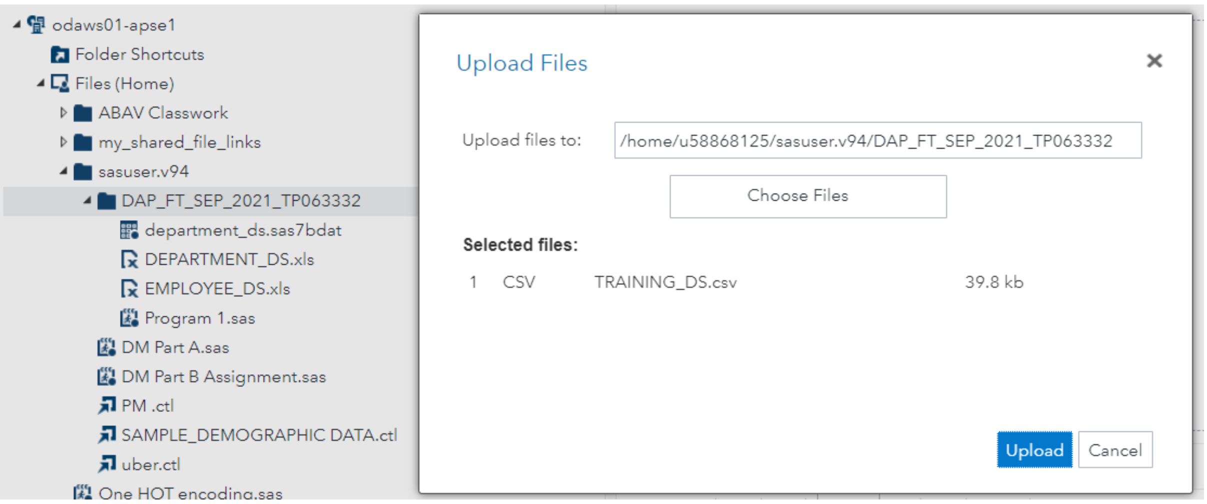


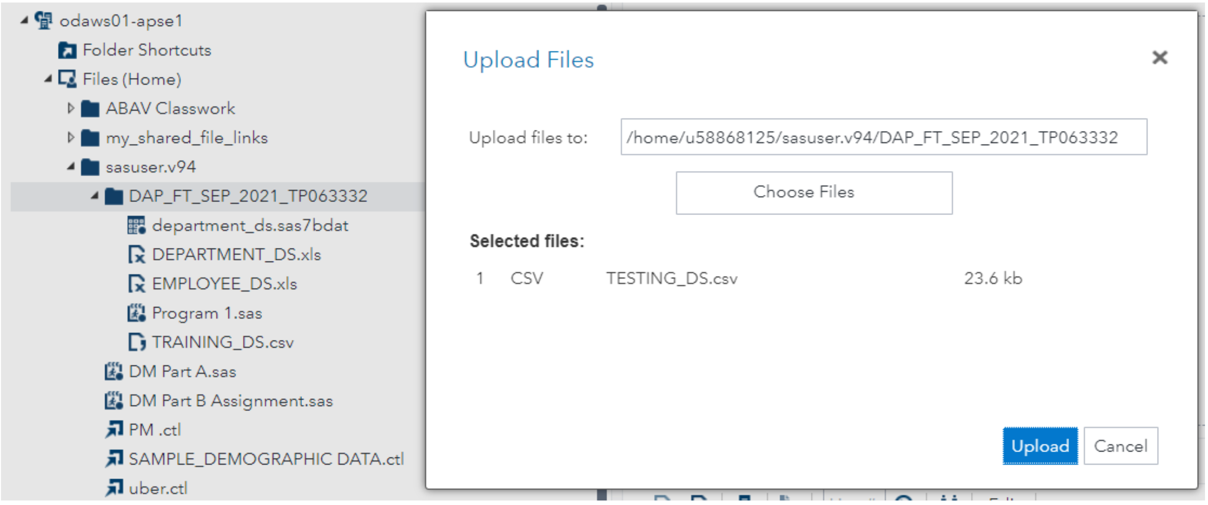
Explanation

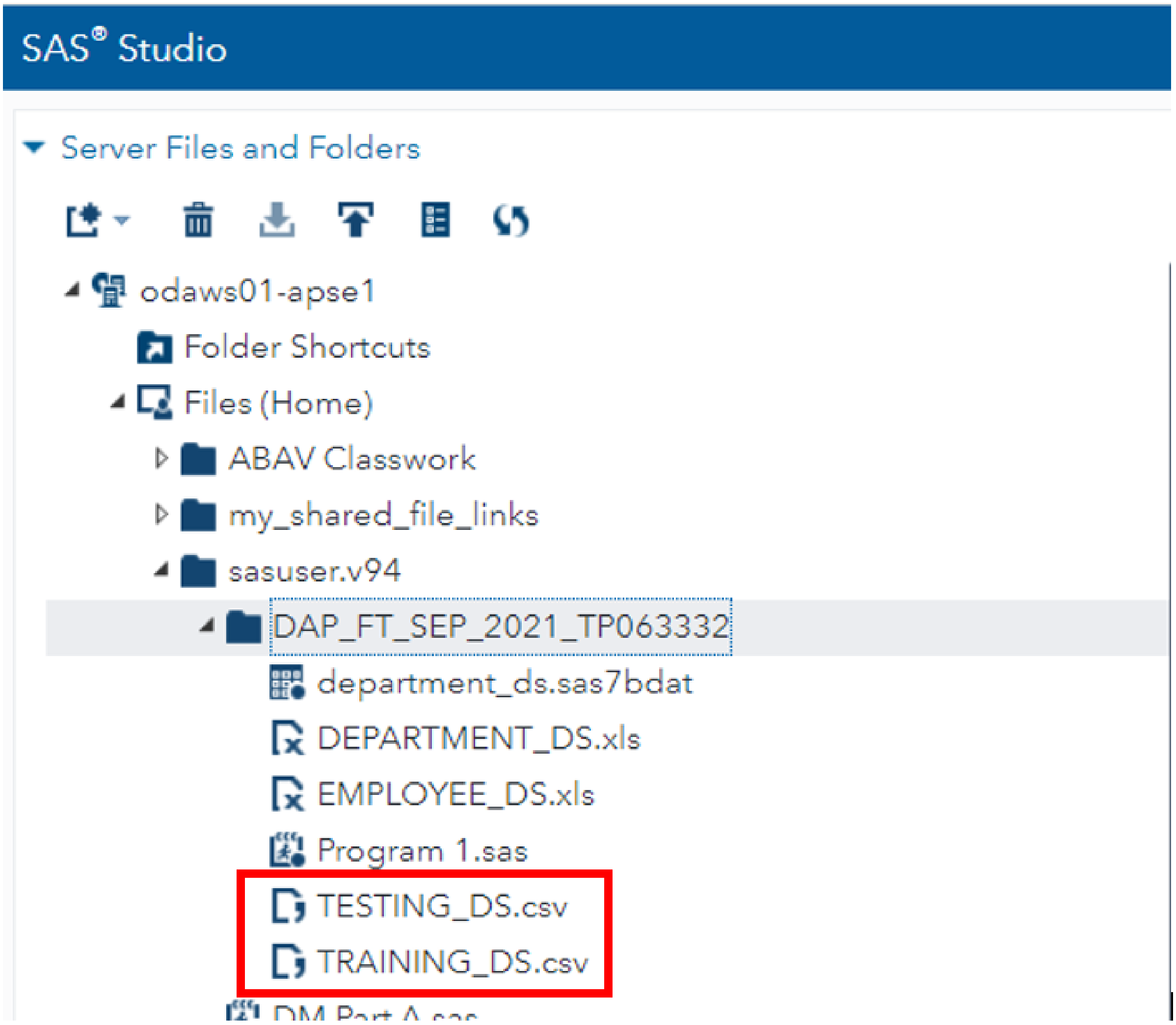
- A library is created to store the data schema imported to SAS Studio.

7.3 Upload the dataset TRAINING_DS and TESTING_DS to the folder DAP_FT_SEP_2021_TP063332

Screenshot(s)





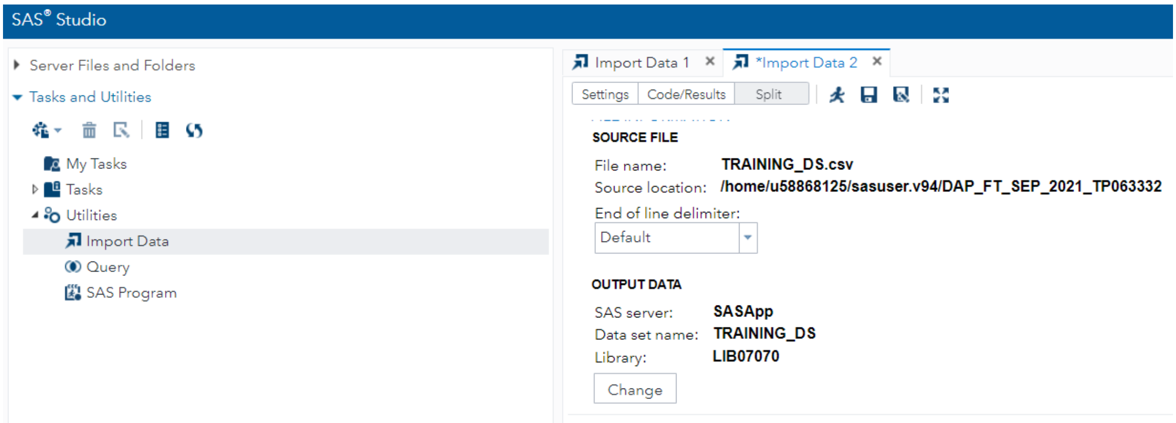


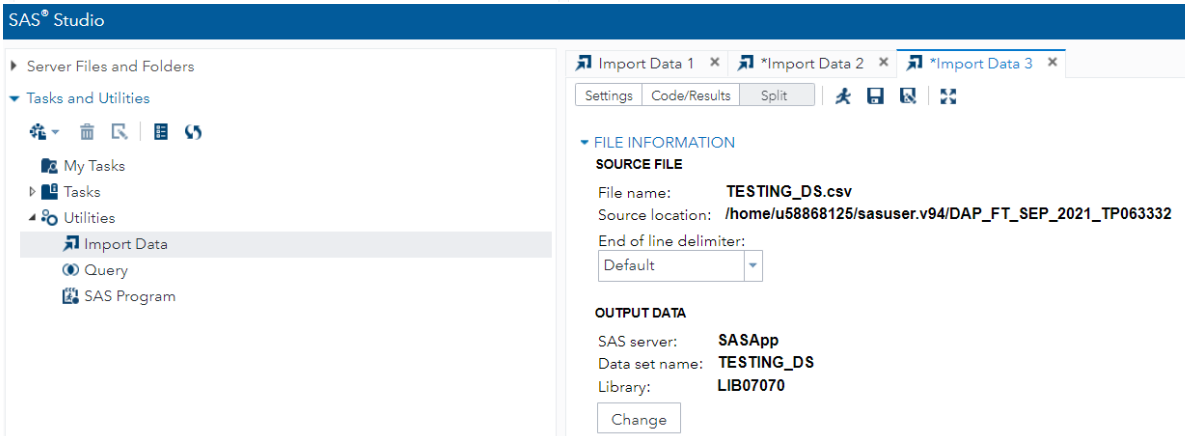
<u>Explanation</u>

- TRAINING_DS and TESTING_DS datasets are uploaded to SAS Studio.

7.4 Import the dataset TRAINING_DS and TESTING_DS to the library LIB07070

<u>Screenshot(s)</u>





<u>Explanation</u>

- TRAINING_DS and TESTING_DS datasets are imported into the library created in previous step. These datasets serve as the database for the SQL programs to run.

7.5 Initial Exploration of the Dataset

<u>Screenshot(s)</u>

```
TITLE1 'Structure of the Dataset (Data Dictionary)';
PROC CONTENTS DATA = LIB07070.TRAINING_DS;
RUN;
```

Structure of the Dataset (Data Dictionary)

The CONTENTS Procedure

Data Set Name	LIB07070.TRAINING_DS	Observations	614
Member Type	DATA	Variables	13
Engine	V9	Indexes	0
Created	11/10/2021 17:25:41	Observation Length	96
Last Modified	11/10/2021 17:25:41	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)		

Engine/Host Dependent Information	
Data Set Page Size	131072
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	1363
Obs in First Data Page	614
Number of Data Set Repairs	0
Filename	/home/u58868125/sasuser.v94/DAP_FT_SEP_2021_TP063332/training_ds.sas7bdat
Release Created	9.0401M6
Host Created	Linux
Inode Number	235407138
Access Permission	rw-r--r--
Owner Name	u58868125
File Size	256KB
File Size (bytes)	262144

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.

Explanation

- Initial understanding of the dataset is done by running the program. There are 5 numerical variables and 8 string variables.

7.6 Univariate Analysis of categorical variables found in the dataset LIB07070.TRAINING_DS

7.6.1 Univariate Analysis of the categorical variable MARITAL STATUS

<u>SAS Codes</u>

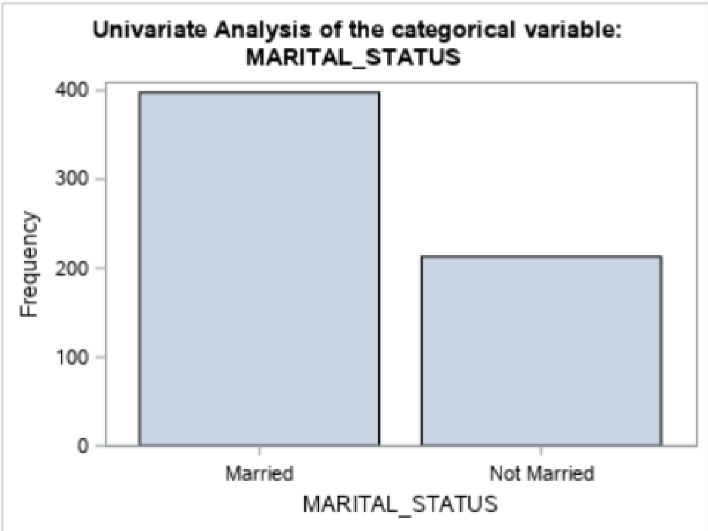
```
TITLE 'Univariate Analysis of the categorical variable: MARITAL_STATUS';
PROC FREQ DATA = LIB07070.TRAINING_DS;
TABLE MARITAL_STATUS;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB07070.TRAINING_DS;
VBAR MARITAL_STATUS;
TITLE 'Univariate Analysis of the categorical variable: MARITAL_STATUS';
RUN;
```

<u>Screenshot(s)</u>

Univariate Analysis of the categorical variable: MARITAL_STATUS

The FREQ Procedure

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				



<u>Explanation</u>

There are 398 married applicants and 213 applicants who are not married. However, there are 3 missing values in the MARITAL_STATUS variable.

7.6.2 Univariate Analysis of the categorical variable GENDER

SAS Codes

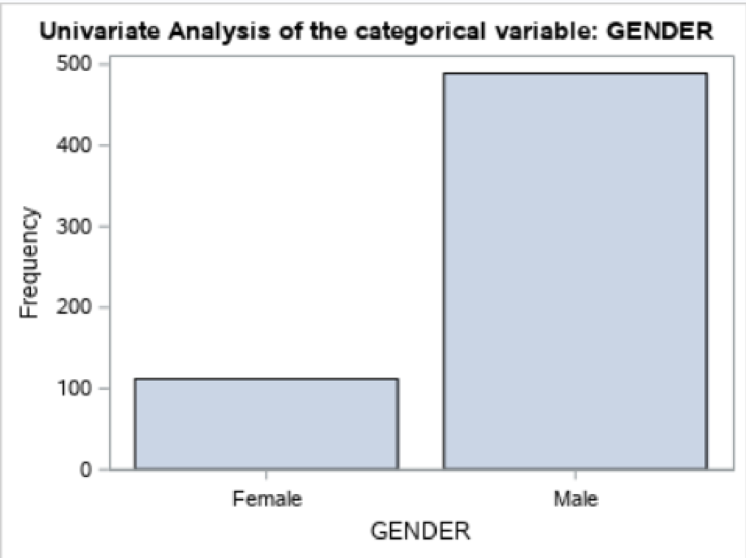
```
TITLE 'Univariate Analysis of the categorical variable: GENDER';
PROC FREQ DATA = LIB07070.TRAINING_DS;
TABLE GENDER;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB07070.TRAINING_DS;
VBAR GENDER;
TITLE 'Univariate Analysis of the categorical variable: GENDER';
RUN;
```

Screenshot(s)

Univariate Analysis of the categorical variable: GENDER

The FREQ Procedure

GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Female	112	18.64	112	18.64
Male	489	81.36	601	100.00
Frequency Missing = 13				



Explanation

There are mostly male in the dataset, consisting as high as 81.36% of the dataset. Other than that, there are also 13 missing data in this GENDER variable.

7.6.3 Univariate Analysis of the categorical variable LOAN_LOCATION

SAS Codes

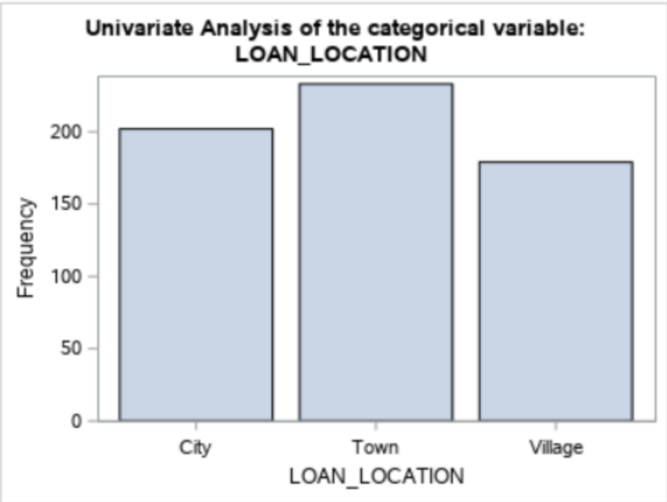
```
TITLE 'Univariate Analysis of the categorical variable: LOAN_LOCATION';
PROC FREQ DATA = LIB07070.TRAINING_DS;
TABLE LOAN_LOCATION;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB07070.TRAINING_DS;
VBAR LOAN_LOCATION;
TITLE 'Univariate Analysis of the categorical variable: LOAN_LOCATION';
RUN;
```

Screenshot(s)

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



Explanation

The loan applicants are mainly from town area, amounting to 233 of them. City area have 202 loan applicants while village have the least number of applicants, only 179 of them. This variable has no missing data.

7.6.4 Univariate Analysis of the categorical variable QUALIFICATION

SAS Codes

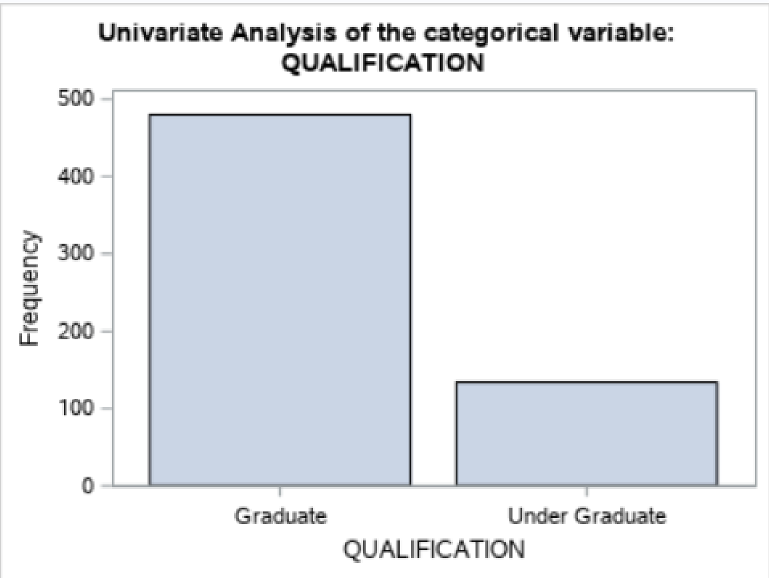
```
TITLE 'Univariate Analysis of the categorical variable: QUALIFICATION';
PROC FREQ DATA = LIB07070.TRAINING_DS;
TABLE QUALIFICATION;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB07070.TRAINING_DS;
VBAR QUALIFICATION;
TITLE 'Univariate Analysis of the categorical variable: QUALIFICATION';
RUN;
```

Screenshot(s)

Univariate Analysis of the categorical variable: QUALIFICATION

The FREQ Procedure

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



Explanation

A total of 78.18% (480 of them) from the loan applicants are graduates while only 134 of them are under graduates. This QUALIFICATION variable does not have any missing data as well.

7.6.5 Univariate Analysis of the categorical variable FAMILY_MEMBERS

SAS Codes

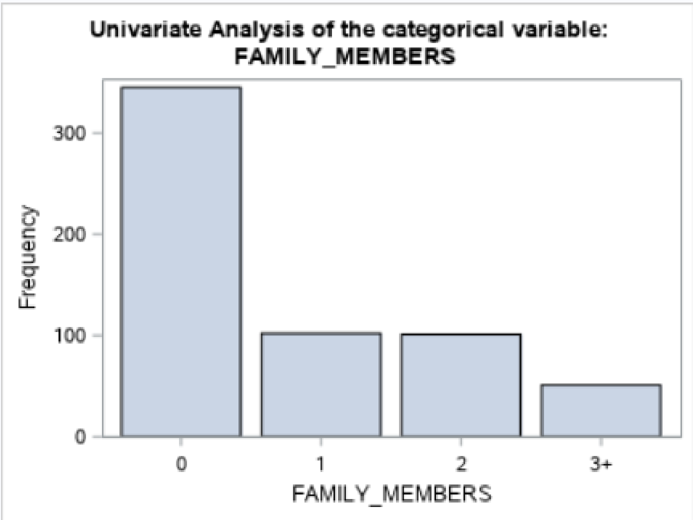
```
TITLE 'Univariate Analysis of the categorical variable: FAMILY_MEMBERS';
PROC FREQ DATA = LIB07070.TRAINING_DS;
TABLE FAMILY_MEMBERS;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB07070.TRAINING_DS;
VBAR FAMILY_MEMBERS;
TITLE 'Univariate Analysis of the categorical variable: FAMILY_MEMBERS';
RUN;
```

Screenshot(s)

Univariate Analysis of the categorical variable: FAMILY_MEMBERS

The FREQ Procedure

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				



Explanation

Most of the loan applicants (57.60%) do not have any family members. This actually indicates that they may be lesser financial commitments with lesser dependent family members. On the other hand, there are 51 of them who have more than 2 family members. This may affect their loan eligibility as banks may prefer individuals with lesser financial commitments reduce loan defaults rate.

7.7 Univariate Analysis of continuous variables found in the dataset LIB07070.TRAINING_DS

7.7.1 Univariate Analysis of the continuous variable CANDIDATE_INCOME

SAS Codes

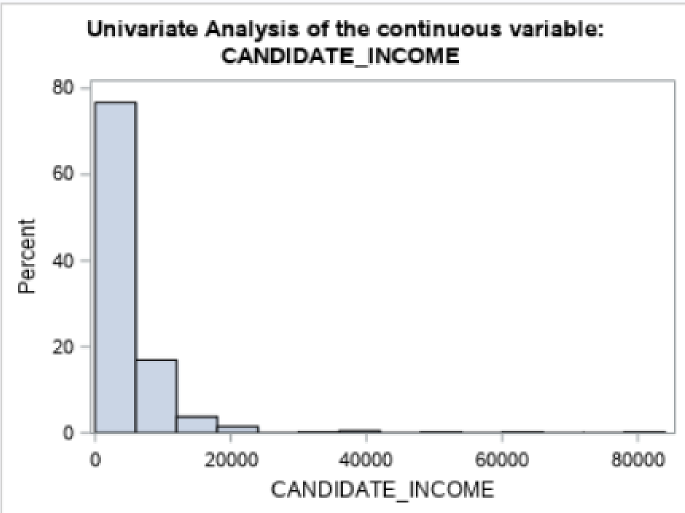
```
PROC MEANS DATA = LIB07070.TRAINING_DS N NMISS MIN MAX MEAN MEDIAN STD;
VAR CANDIDATE_INCOME;
TITLE 'Univariate Analysis of the continuous variable: CANDIDATE_INCOME';
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB07070.TRAINING_DS;
HISTOGRAM CANDIDATE_INCOME;
TITLE 'Univariate Analysis of the continuous variable: CANDIDATE_INCOME';
RUN;
```

Screenshot(s)

Univariate Analysis of the continuous 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



Explanation

As observed from the distribution of the histogram above, most of the candidates have an income of below $20000 with a mean of $5403, although the maximum income recorded is $81000.

7.7.2 Univariate Analysis of the continuous variable LOAN_AMOUNT

<u>SAS Codes</u>

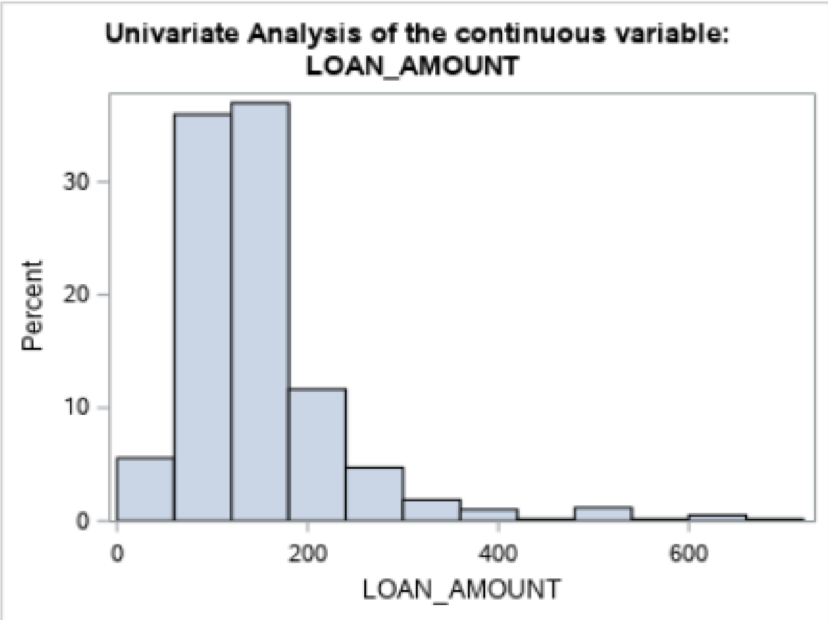
```
PROC MEANS DATA = LIB07070.TRAINING_DS N NMISS MIN MAX MEAN MEDIAN STD;
VAR LOAN_AMOUNT;
TITLE 'Univariate Analysis of the continuous variable: LOAN_AMOUNT';
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB07070.TRAINING_DS;
HISTOGRAM LOAN_AMOUNT;
TITLE 'Univariate Analysis of the continuous variable: LOAN_AMOUNT';
RUN;
```

<u>Screenshot(s)</u>

Univariate Analysis of the continuous 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



<u>Explanation</u>

There are 22 missing data in the LOAN_AMOUNT variable. It can be observed that most of the applicants have loan amount of less than $200 with a median of $128.

7.7.3 Univariate Analysis of the continuous variable GUARANTEE_INCOME

SAS Codes

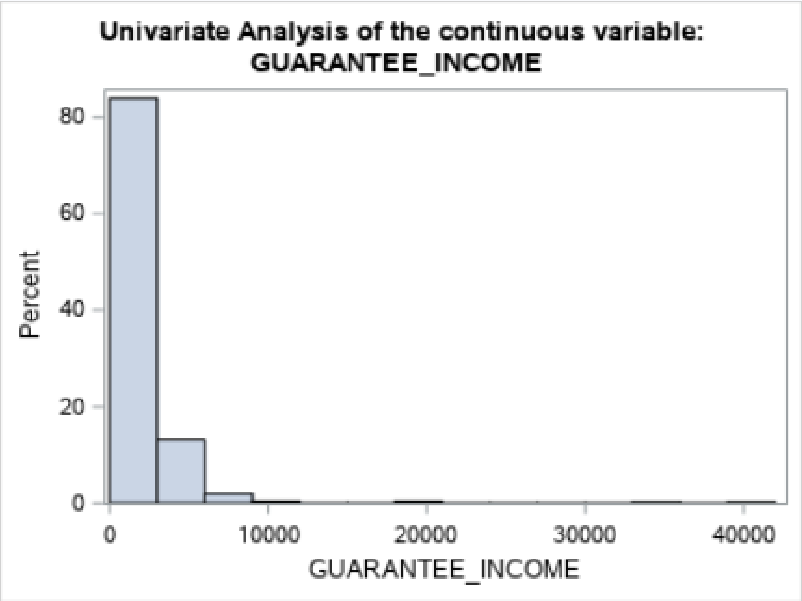
```
PROC MEANS DATA = LIB07070.TRAINING_DS N NMISS MIN MAX MEAN MEDIAN STD;
VAR GUARANTEE_INCOME;
TITLE 'Univariate Analysis of the continuous variable: GUARANTEE_INCOME';
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB07070.TRAINING_DS;
HISTOGRAM GUARANTEE_INCOME;
TITLE 'Univariate Analysis of the continuous variable: GUARANTEE_INCOME';
RUN;
```

Screenshot(s)

Univariate Analysis of the continuous 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



Explanation

More than 80% of the applicants have guarantee income of less than $5000.

7.7.4 Univariate Analysis of the continuous variable LOAN_DURATION

SAS Codes

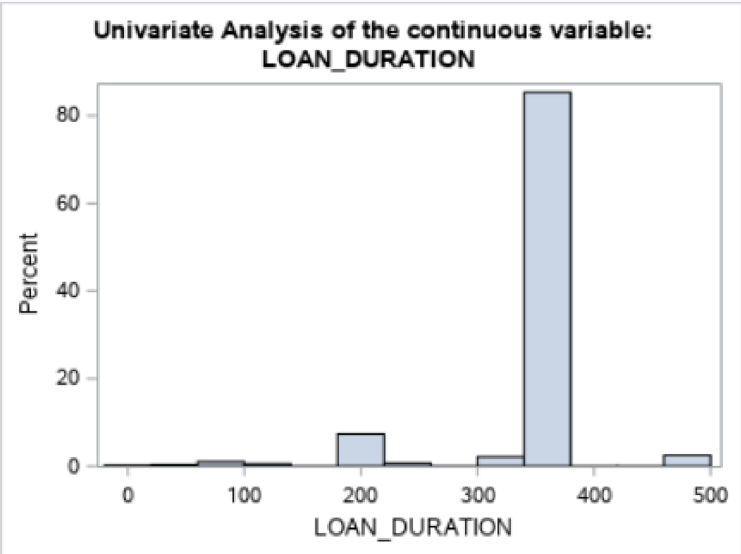
```
PROC MEANS DATA = LIB07070.TRAINING_DS N NMISS MIN MAX MEAN MEDIAN STD;
VAR LOAN_DURATION;
TITLE 'Univariate Analysis of the continuous variable: LOAN_DURATION';
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = LIB07070.TRAINING_DS;
HISTOGRAM LOAN_DURATION;
TITLE 'Univariate Analysis of the continuous variable: LOAN_DURATION';
RUN;
```

Screenshot(s)

Univariate Analysis of the continuous variable: LOAN_DURATION

The MEANS Procedure

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



Explanation

There are 14 missing data in the LOAN_DURATION variable. Other than that, over 80% of the applicants are having a loan duration of 360 months.

7.8 Bivariate Analysis of variables found in the dataset LIB07070.TRAINING_DS

7.8.1 Bivariate Analysis of the variables (LOAN_LOCATION - Categorical variable vs CANDIDATE_INCOME – Continuous variable) found in the LIB07070.TRAINING_DS

SAS Codes

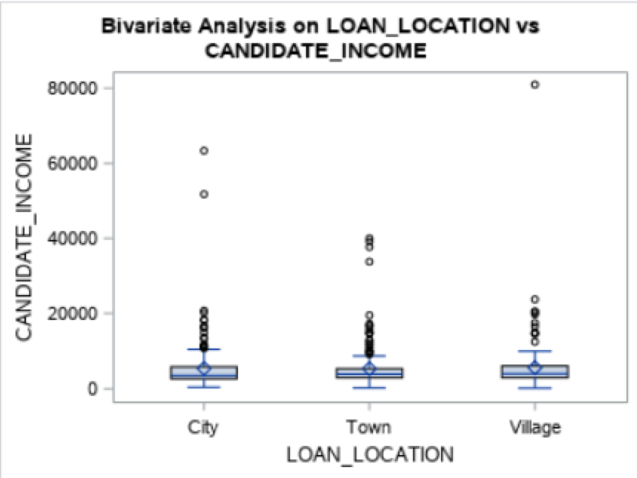
```
TITLE1 'Bivariate Analysis of the variables (Categorical vs Continuous) found in the LIB07070.TRAINING_DS';
TITLE1 'Bivariate Analysis of the variables (LOAN_LOCATION vs CANDIDATE_INCOME)';

PROC MEANS DATA = LIB07070.TRAINING_DS;
CLASS LOAN_LOCATION; /* CHAR */
VAR CANDIDATE_INCOME; /*NUMERIC*/
RUN;

PROC SGPLOT DATA = LIB07070.TRAINING_DS;
VBOX CANDIDATE_INCOME / CATEGORY=LOAN_LOCATION;
/*LL X-AXIS CI Y-AXIS */
TITLE 'Bivariate Analysis on LOAN_LOCATION vs CANDIDATE_INCOME';
RUN;
```

Screenshot(s)

Bivariate Analysis of the variables (LOAN_LOCATION vs CANDIDATE_INCOME)

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
LOAN_LOCATION	N Obs	N	Mean	Std Dev	Minimum	Maximum
City	202	202	5398.25	6392.93	416.0000000	63337.00
Town	233	233	5292.26	5279.63	210.0000000	39999.00
Village	179	179	5554.08	6782.66	150.0000000	81000.00



Explanation

From the boxplot above, we can see that the distribution of data between LOAN_LOCATION and CANDIDATE_INCOME is quite similar across the different loan locations. There are also a few observations that lies as outliers in the dataset.

7.8.2 Bivariate Analysis of the variables (LOAN_HISTORY – categorical variable vs LOAN_APPROVAL_STATUS – categorical variable) found in the LIB07070.TRAINING_DS

SAS Codes

```
TITLE1 'Bivariate Analysis of the variables (Categorical vs Categorical) found in the LIB07070.TRAINING_DS';
TITLE2 'Bivariate Analysis of the variables (LOAN_HISTORY vs LOAN_APPROVAL_STATUS)';
FOOTNOTE '-----END-----';
PROC FREQ DATA = LIB07070.TRAINING_DS;
TABLE LOAN_HISTORY * LOAN_APPROVAL_STATUS /
PLOTS = FREQPLOT ( TWOWAY = STACKED SCALE =GROUPPCT );
RUN;
```

Screenshot(s)

Bivariate Analysis of the variables (LOAN_HISTORY vs LOAN_APPROVAL_STATUS)

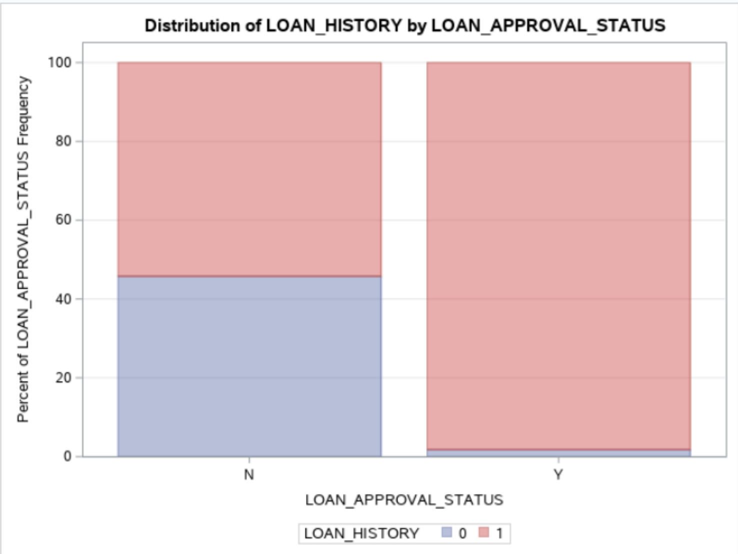
The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of LOAN_HISTORY by LOAN_APPROVAL_STATUS

LOAN_HISTORY	LOAN_APPROVAL_STATUS N	LOAN_APPROVAL_STATUS Y	Total
0	82 14.54 92.13 45.81	7 1.24 7.87 1.82	89 15.78
1	97 17.20 20.42 54.19	378 67.02 79.58 98.18	475 84.22
Total	179 31.74	385 68.26	564 100.00

Frequency Missing = 50



-----END-----

Explanation

Through the graph plot above, it is observed that a large percentage of LOAN_APPROVAL_STATUS showing Y are also showing 1 in the LOAN_HISTORY. This is a very important finding where it suggested that loan history is quite likely to affect the outcome of the loan approval status.

7.8.3 Bivariate Analysis of the variables (GENDER – categorical variable vs LOAN_APPROVAL_STATUS – categorical variable) found in the LIB07070.TRAINING_DS

<u>SAS Codes</u>

```
TITLE1 'Bivariate Analysis of the variables (Categorical vs Continuous) found in the LIB07070.TRAINING_DS';
TITLE1 'Bivariate Analysis of the variables (GENDER vs LOAN_APPROVAL_STATUS)';
FOOTNOTE '-----END-----';
PROC FREQ DATA = LIB07070.TRAINING_DS;
TABLE GENDER * LOAN_APPROVAL_STATUS /
PLOTS = FREQPLOT ( TWOWAY = STACKED SCALE =GROUPPCT );
RUN;
```

<u>Screenshot(s)</u>

Bivariate Analysis of the variables (GENDER vs LOAN_APPROVAL_STATUS)

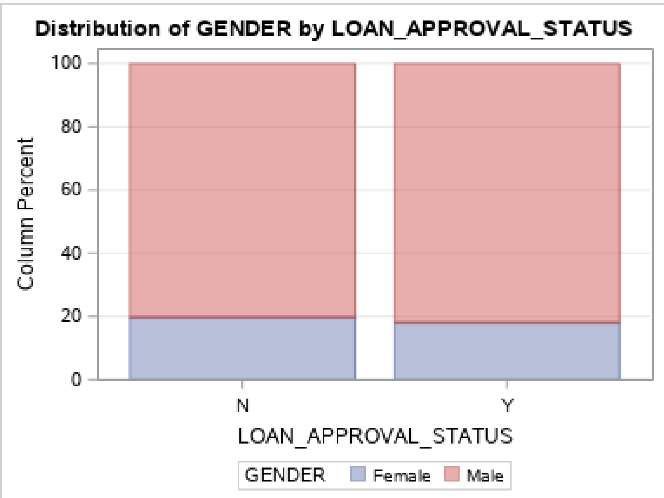
The FREQ Procedure

Frequency Percent Row Pct Col Pct

Table of GENDER by LOAN_APPROVAL_STATUS

GENDER	LOAN_APPROVAL_STATUS N	Y	Total
Female	37 6.16 33.04 19.79	75 12.48 66.96 18.12	112 18.64
Male	150 24.96 30.67 80.21	339 56.41 69.33 81.88	489 81.36
Total	187 31.11	414 68.89	601 100.00

Frequency Missing = 13



-----END-----

<u>Explanation</u>

From the graph above, it is observed that number of loan approved between male and female are quite similar, which indicates that the gender does not really contribute to the outcome of the loan approval status.

7.8.4 Bivariate Analysis of the variables (MARITAL_STATUS – categorical variable vs LOAN_APPROVAL_STATUS – categorical variable) found in the LIB07070.TRAINING_DS

SAS Codes

```
TITLE1 'Bivariate Analysis of the variables (Categorical vs Categorical) found in the LIB07070.TRAINING_DS';
TITLE2 'Bivariate Analysis of the variables (MARITAL_STATUS vs LOAN_APPROVAL_STATUS)';
FOOTNOTE '-----END-----';
PROC FREQ DATA = LIB07070.TRAINING_DS;
TABLE MARITAL_STATUS * LOAN_APPROVAL_STATUS /
PLOTS = FREQPLOT ( TWOWAY = STACKED SCALE =GROUPPCT );
RUN;
```

Screenshot(s)

Bivariate Analysis of the variables (Categorical vs Categorical) found in the LIB07070.TRAINING_DS
Bivariate Analysis of the variables (MARITAL_STATUS vs LOAN_APPROVAL_STATUS)

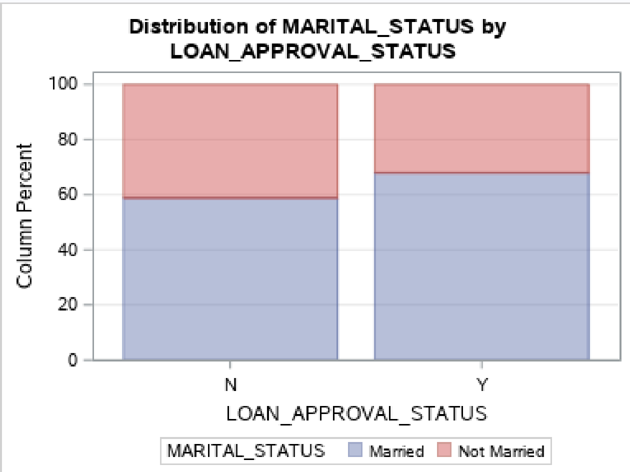
The FREQ Procedure

Frequency Percent Row Pct Col Pct

Table of MARITAL_STATUS by LOAN_APPROVAL_STATUS

MARITAL_STATUS	LOAN_APPROVAL_STATUS: N	Y	Total
Married	113 18.49 28.39 58.85	285 46.64 71.61 68.02	398 65.14
Not Married	79 12.93 37.09 41.15	134 21.93 62.91 31.98	213 34.86
Total	192 31.42	419 68.58	611 100.00

Frequency Missing = 3



-----END-----

Explanation

From the graph above, it is observed that number of loan approved between married and not married are a little different, which indicates that there are more married applicants who got their loan approved as compared to the not married ones.

7.9 Finding the variables with missing values and data imputation.

SAS Codes

```
TITLE 'Find the Categorical and Continuous variables with missing values';
PROC FORMAT;
VALUE $missfmt ' ' = 'Missing' others = 'Not missing';
VALUE  missfmt .   = 'Missing' others = 'Not missing';
RUN;

PROC FREQ DATA=LIB07070.TRAINING_DS;
FORMAT _CHAR_ $missfmt.;
FORMAT _NUMERIC_ missfmt.;

TABLE _CHAR_ / missing nocum nopercent;
TABLE _NUMERIC_ / missing nocum nopercent;
RUN;
```

Screenshot(s)

GENDER	Frequency
Missing	13
Female	112
Male	489

MARITAL_STATUS	Frequency
Missing	3
Married	398
Not Married	213

FAMILY_MEMBERS	Frequency
Missing	15
0	345
1	102
2	101
3+	51

EMPLOYMENT	Frequency
Missing	32
No	500
Yes	82

LOAN_AMOUNT	Frequency
Missing	22
Not missing	592

LOAN_DURATION	Frequency
Missing	14
Not missing	600

LOAN_HISTORY	Frequency
Missing	50
Not missing	564

Explanation

There is total 7 variables with missing values, with the highest being 50 in the LOAN_HISTORY variable. These missing data must be pre-processed first and imputed before performing the next step.

7.9.1 Imputing the missing values in GENDER variable

STEP 1: Making a copy of the dataset: LIB07070.TRAINING_DS and listing the missing values in GENDER variable

SAS Codes

```
TITLE 'STEP 1: Make a copy of the dataset LIB07070.TRAINING_DS before imputing missing values';
PROC SQL;
CREATE TABLE LIB07070.TRAINING_DS_FI AS
SELECT * FROM LIB07070.TRAINING_DS;
QUIT;


TITLE 'LIST THE OBSERVATIONS WITH MISSING VALUES IN GENDER VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( GENDER IS MISSING );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( GENDER IS MISSING );
QUIT;
```

Screenshot(s)

LIST THE OBSERVATIONS WITH MISSING VALUES IN GENDER VARIABLE

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
LP001050		Married	2	Under Graduate	No	3365	1917	112	360	0	Village	N
LP001448		Married	3+	Graduate	No	23803	0	370	360	1	Village	Y
LP001585		Married	3+	Graduate	No	51763	0	700	300	1	City	Y
LP001644		Married	0	Graduate	Yes	674	5296	168	360	1	Village	Y
LP002024		Married	0	Graduate	No	2473	1843	159	360	1	Village	N
LP002103		Married	1	Graduate	Yes	9833	1833	182	180	1	City	Y
LP002478		Married	0	Graduate	Yes	2083	4083	160	360	.	Town	Y
LP002501		Married	0	Graduate	No	16692	0	110	360	1	Town	Y
LP002530		Married	2	Graduate	No	2873	1872	132	360	0	Town	N
LP002625		Not Married	0	Graduate	No	3583	0	96	360	1	City	N
LP002872		Married	0	Graduate	No	3087	2210	136	360	0	Town	N
LP002925		Not Married	0	Graduate	No	4750	0	94	360	1	Town	Y
LP002933		Not Married	3+	Graduate	Yes	9357	0	292	360	1	Town	Y

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
13

-----END-----

Explanation

Before imputing all the variables, the dataset should be duplicated before imputation is done so that the imputation is only done on the duplicate dataset and have the original one as backup in case of any necessary situations. (This step is only done once in this section.)

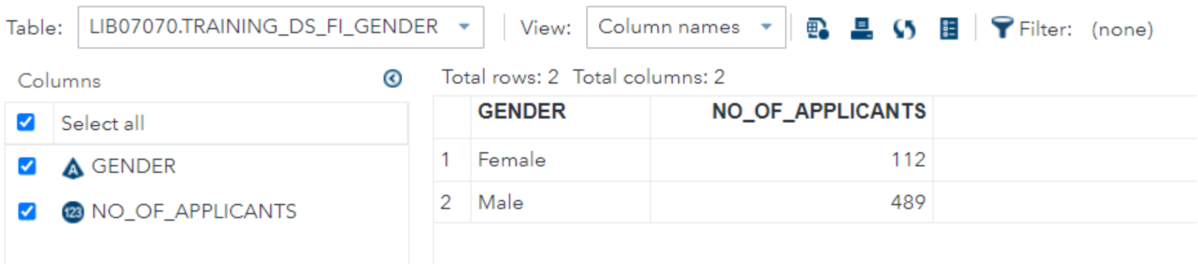
After that, the GENDER variable is checked for missing data and the rows with missing values in the variable are listed.

STEP 2: Create a dataset to hold the gender and number of applicants

SAS Codes

```
TITLE 'STEP 2: Create a dataset to hold the gender and number of applicants';
PROC SQL;
CREATE TABLE LIB07070.TRAINING_DS_FI_GENDER AS
SELECT GENDER, COUNT (*) AS NO_OF_APPLICANTS
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( GENDER IS NOT MISSING ) OR
        ( GENDER IS NOT NULL ) OR
        ( GENDER NE '' ) )
GROUP BY GENDER;

QUIT;
```

Screenshot(s)



Explanation

Since GENDER is a binary categorical variable, mode imputation will be used to impute the missing values. To do so, a secondary table is required to tabulate the frequency of the variable.

STEP 3: Display the contents of the dataset LIB07070.TRAINING_DS_FI_GENDER

SAS Codes

```
TITLE 'STEP 3: Display the contents of the dataset LIB07070.TRAINING_DS_FI_GENDER';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI_GENDER;

QUIT;
```

Screenshot(s)

STEP 3: Display the contents of the dataset LIB07070.TRAINING_DS_FI_GENDER

GENDER	NO_OF_APPLICANTS
Female	112
Male	489

<u>Explanation</u>

After creating the secondary table, the contents are viewed before doing the imputation.

<u>*STEP 4: Impute the missing values found in the GENDER variable.*</u>

<u>SAS Codes</u>

```
TITLE 'STEP 4: Impute the missing values found in the GENDER variable';
PROC SQL;

UPDATE LIB07070.TRAINING_DS_FI
SET GENDER = ( SELECT GENDER
               FROM LIB07070.TRAINING_DS_FI_GENDER
               WHERE NO_OF_APPLICANTS EQ ( SELECT MAX(NO_OF_APPLICANTS) Label 'NO OF APPLICANTS'
                                           FROM LIB07070.TRAINING_DS_FI_GENDER ) )
                                           /*It is a sub-program to find the highest no of applicants*/
WHERE ( ( GENDER IS NOT MISSING ) OR
        ( GENDER IS NOT NULL ) OR
        ( GENDER NE '' ) );
QUIT;
```

<u>Screenshot(s)</u>

```
TITLE 'STEP 4: Impute the missing values found in the GENDER variable';
PROC SQL;

UPDATE LIB07070.TRAINING_DS_FI
SET GENDER = ( SELECT GENDER
   FROM LIB07070.TRAINING_DS_FI_GENDER
   WHERE NO_OF_APPLICANTS EQ ( SELECT MAX(NO_OF_APPLICANTS) Label 'NO OF APPLICANTS'
   FROM LIB07070.TRAINING_DS_FI_GENDER ) )
   /*It is a sub-program to find the highest no of applicants*/
WHERE ( ( GENDER IS MISSING ) OR
( GENDER IS NULL ) OR
( GENDER EQ '' ) );
NOTE: 13 rows were updated in LIB07070.TRAINING_DS_FI.
```

<u>Explanation</u>

The missing values in GENDER variables are imputed using the mode, through the secondary table that is created in the previous step.

<u>*STEP 5: After imputing missing values, list the observations with missing values in GENDER variable*</u>

<u>SAS Codes</u>

```
TITLE 'STEP 5: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN GENDER VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( GENDER IS MISSING );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( GENDER IS MISSING );
QUIT;
```

<u>Screenshot(s)</u>

STEP 5: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN GENDER VARIABLE

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
0

-----END-----

<u>Explanation</u>

After imputation is done, the GENDER variable is double-checked for missing data to ensure there are no more missing data.

7.9.2 Imputing the missing values in MARITAL_STATUS variable

STEP 1: Listing the missing values in MARITAL_STATUS variable

SAS Codes

```
TITLE 'LIST THE OBSERVATIONS WITH MISSING VALUES IN MARITAL_STATUS VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( MARITAL_STATUS IS MISSING );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( MARITAL_STATUS IS MISSING );
QUIT;
```

Screenshot(s)

LIST THE OBSERVATIONS WITH MISSING VALUES IN MARITAL_STATUS VARIABLE

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
LP001357	Male			Graduate	No	3816	754	160	360	1	City	Y
LP001760	Male			Graduate	No	4758	0	158	480	1	Town	Y
LP002393	Female			Graduate	No	10047	0	.	240	1	Town	Y

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
13

-----END-----

Explanation

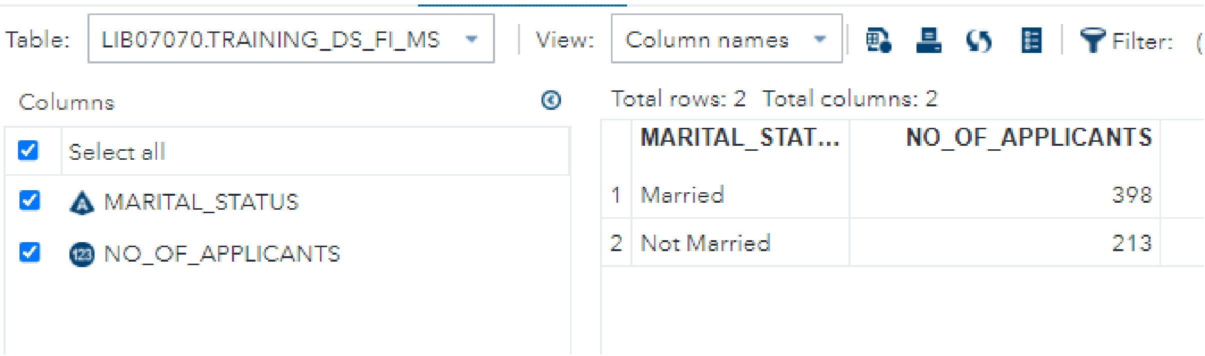
The MARITAL_STATUS variable is checked for missing data and the rows with missing values in the variable are listed.

STEP 2: Create a dataset to hold the marital status and number of applicants

SAS Codes

```
TITLE 'STEP 2: Create a dataset to hold the gender and number of applicants';
PROC SQL;
CREATE TABLE LIB07070.TRAINING_DS_FI_MS AS
SELECT MARITAL_STATUS, COUNT (*) AS NO_OF_APPLICANTS
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( MARITAL_STATUS IS NOT MISSING ) OR
        ( MARITAL_STATUS IS NOT NULL ) OR
        ( MARITAL_STATUS NE '' ) )
GROUP BY MARITAL_STATUS;

QUIT;
```

Screenshot(s)



Explanation

Since MARITAL_STATUS is a binary categorical variable, mode imputation will be used to impute the missing values. To do so, a secondary table is required to tabulate the frequency of the variable.

STEP 3: Display the contents of the dataset LIB07070.TRAINING_DS_FI_MS

SAS Codes

```
TITLE 'STEP 3: Display the contents of the dataset LIB07070.TRAINING_DS_FI_MS';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI_MS;

QUIT;
```

Screenshot(s)

STEP 3: Display the contents of the dataset LIB07070.TRAINING_DS_FI_MS

MARITAL_STATUS	NO_OF_APPLICANTS
Married	398
Not Married	213

Explanation

After creating the secondary table, the contents are viewed before doing the imputation.

STEP 4: Impute the missing values found in the MARITAL_STATUS variable.

SAS Codes

```
TITLE 'STEP 4: Impute the missing values found in the GENDER variable';
PROC SQL;

UPDATE LIB07070.TRAINING_DS_FI
SET MARITAL_STATUS = ( SELECT MARITAL_STATUS
                       FROM LIB07070.TRAINING_DS_FI_MS
                       WHERE NO_OF_APPLICANTS EQ ( SELECT MAX(NO_OF_APPLICANTS) Label 'NO OF APPLICANTS'
                                                   FROM LIB07070.TRAINING_DS_FI_MS ) )
                                                   /*It is a sub-program to find the highest no of applicants*/
WHERE ( ( MARITAL_STATUS IS MISSING ) OR
        ( MARITAL_STATUS IS NULL ) OR
        ( MARITAL_STATUS EQ '' ) );
QUIT;
```

Screenshot(s)

```
TITLE 'STEP 4: Impute the missing values found in the GENDER variable';
PROC SQL;

UPDATE LIB07070.TRAINING_DS_FI
SET MARITAL_STATUS = ( SELECT MARITAL_STATUS
      FROM LIB07070.TRAINING_DS_FI_MS
     WHERE NO_OF_APPLICANTS EQ ( SELECT MAX(NO_OF_APPLICANTS) Label 'NO OF APPLICANTS'
      FROM LIB07070.TRAINING_DS_FI_MS ) )
      /*It is a sub-program to find the highest no of applicants*/
WHERE ( ( MARITAL_STATUS IS MISSING ) OR
( MARITAL_STATUS IS NULL ) OR
( MARITAL_STATUS EQ '' ) );
NOTE: 3 rows were updated in LIB07070.TRAINING_DS_FI.
```

Explanation

The missing values in MARITAL_STATUS variables are imputed using the mode, through the secondary table that is created in the previous step.

STEP 5: After imputing missing values, list the observations with missing values in MARITAL STATUS variable

SAS Codes

```
TITLE 'STEP 5: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN MARITAL_STATUS VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( MARITAL_STATUS IS MISSING );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( MARITAL_STATUS IS MISSING );
QUIT;
```

Screenshot(s)

STEP 5: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN MARITAL_STATUS VARIABLE

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
0

-----END-----

<u>Explanation</u>

After imputation is done, the MARITAL_STATUS variable is double-checked for missing data to ensure there are no more missing data.

7.9.3 Imputing the missing values in EMPLOYMENT variable

STEP 1: Listing the missing values in EMPLOYMENT variable

SAS Codes

```
/************EMPLOYMENT************/
TITLE 'STEP 1: Make a copy of the dataset LIB07070.TRAINING_DS before imputing missing values';
PROC SQL;
CREATE TABLE LIB07070.TRAINING_DS_FI AS
SELECT * FROM LIB07070.TRAINING_DS;
QUIT;


TITLE 'LIST THE OBSERVATIONS WITH MISSING VALUES IN EMPLOYMENT VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( EMPLOYMENT IS MISSING );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( EMPLOYMENT IS MISSING );
QUIT;
```

Screenshot(s)

LIST THE OBSERVATIONS WITH MISSING VALUES IN EMPLOYMENT VARIABLE

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
LP001027	Male	Married	2	Graduate		2500	1840	109	360	1	City	Y
LP001041	Male	Married	0	Graduate		2600	3500	115	.	1	City	Y
LP001052	Male	Married	1	Graduate		3717	2925	151	360	.	Town	N
LP001087	Female	Not Married	2	Graduate		3750	2083	120	360	1	Town	Y
LP001091	Male	Married	1	Graduate		4166	3369	201	360	.	City	N
LP001326	Male	Not Married	0	Graduate		6782	0	.	360	.	City	N
LP001370	Male	Not Married	0	Under Graduate		7333	0	120	360	1	Village	N
LP001387	Female	Married	0	Graduate		2929	2333	139	360	1	Town	Y
LP001398	Male	Not Married	0	Graduate		5050	0	118	360	1	Town	Y
LP001546	Male	Not Married	0	Graduate		2980	2083	120	360	1	Village	Y
LP001581	Male	Married	0	Under Graduate		1820	1769	95	360	1	Village	Y
LP001732	Male	Married	2	Graduate		5000	0	72	360	0	Town	N
LP001768	Male	Married	0	Graduate		3716	0	42	180	1	Village	Y
LP001786	Male	Married	0	Graduate		5746	0	255	360	.	City	N
LP001883	Female	Not Married	0	Graduate		3418	0	135	360	1	Village	N
LP001949	Male	Married	3+	Graduate		4416	1250	110	360	1	City	Y
LP002101	Male	Married	0	Graduate		63337	0	490	180	1	City	Y
LP002110	Male	Married	1	Graduate		5250	688	160	360	1	Village	Y
LP002128	Male	Married	2	Graduate		2583	2330	125	360	1	Village	Y
LP002209	Female	Not Married	0	Graduate		2764	1459	110	360	1	City	Y
LP002226	Male	Married	0	Graduate		3333	2500	128	360	1	Town	Y
LP002237	Male	Not Married	1	Graduate		3667	0	113	180	1	City	Y
LP002319	Male	Married	0	Graduate		6256	0	160	360	.	City	Y
LP002386	Male	Not Married	0	Graduate		12876	0	405	360	1	Town	Y

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
32

-----END-----

Explanation

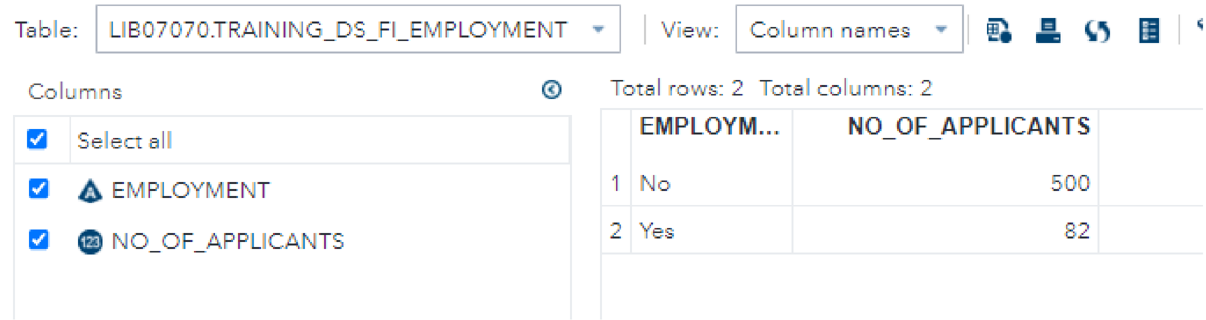
The EMPLOYMENT variable is checked for missing data and the rows with missing values in the variable are listed.

STEP 2: Create a dataset to hold the employment and number of applicants

SAS Codes

```
TITLE 'STEP 2: Create a dataset to hold the EMPLOYMENT and number of applicants';
PROC SQL;
CREATE TABLE LIB07070.TRAINING_DS_FI_EMPLOYMENT AS
SELECT EMPLOYMENT, COUNT (*) AS NO_OF_APPLICANTS
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( EMPLOYMENT IS NOT MISSING ) OR
        ( EMPLOYMENT IS NOT NULL ) OR
        ( EMPLOYMENT NE '' ) )
GROUP BY EMPLOYMENT;

QUIT;
```

Screenshot(s)



	EMPLOYM...	NO_OF_APPLICANTS
1	No	500
2	Yes	82

Explanation

Since EMPLOYMENT is a binary categorical variable, mode imputation will be used to impute the missing values. To do so, a secondary table is required to tabulate the frequency of the variable.

STEP 3: Display the contents of the dataset LIB07070.TRAINING_DS_FI_EMPLOYMENT

SAS Codes

```
TITLE 'STEP 3: Display the contents of the dataset LIB07070.TRAINING_DS_FI_EMPLOYMENT';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI_EMPLOYMENT;

QUIT;
```

Screenshot(s)

STEP 3: Display the contents of the dataset LIB07070.TRAINING_DS_FI_EMPLOYMENT

EMPLOYMENT	NO_OF_APPLICANTS
No	500
Yes	82

Explanation

After creating the secondary table, the contents are viewed before doing the imputation.

STEP 4: Impute the missing values found in the EMPLOYMENT variable.

SAS Codes

```
TITLE 'STEP 4: Impute the missing values found in the EMPLOYMENT variable';
PROC SQL;

UPDATE LIB07070.TRAINING_DS_FI
SET EMPLOYMENT = ( SELECT EMPLOYMENT
                   FROM LIB07070.TRAINING_DS_FI_EMPLOYMENT
                   WHERE NO_OF_APPLICANTS EQ ( SELECT MAX(NO_OF_APPLICANTS) Label 'NO OF APPLICANTS'
                                                FROM LIB07070.TRAINING_DS_FI_EMPLOYMENT ) )
                                                /*It is a sub-program to find the highest no of applicants*/
WHERE ( ( EMPLOYMENT IS MISSING ) OR
        ( EMPLOYMENT IS NULL ) OR
        ( EMPLOYMENT EQ '' ) );
QUIT;
```

Screenshot(s)

```
             WHERE NO_OF_APPLICANTS EQ ( SELECT MAX(NO_OF_A
             FROM LIB07070.TRAINING_DS_FI_EMPLOYMENT ) )
             /*It is a sub-program to find the highest no c
          WHERE ( ( EMPLOYMENT IS MISSING ) OR
          ( EMPLOYMENT IS NULL ) OR
          ( EMPLOYMENT EQ '' ) );
NOTE: 32 rows were updated in LIB07070.TRAINING_DS_FI.

          QUIT;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.01 seconds
      user cpu time       0.00 seconds
      system cpu time     0.00 seconds
      memory              5888.18k
```

Explanation

The missing values in EMPLOYMENT variable are imputed using the mode, through the secondary table that is created in the previous step.

STEP 5: After imputing missing values, list the observations with missing values in EMPLOYMENT variable

SAS Codes

```
TITLE 'STEP 5: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN EMPLOYMENT VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( EMPLOYMENT IS MISSING );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( EMPLOYMENT IS MISSING );
QUIT;
```

Screenshot(s)

STEP 5: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN EMPLOYMENT VARIABLE

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
0

-----END-----

Explanation

After imputation is done, the EMPLOYMENT variable is double-checked for missing data to ensure there are no more missing data.

7.9.4 Imputing the missing values in LOAN_HISTORY variable

STEP 1: Listing the missing values in LOAN_HISTORY variable

SAS Codes

```
TITLE 'STEP 1: Make a copy of the dataset LIB07070.TRAINING_DS before imputing missing values';
PROC SQL;
CREATE TABLE LIB07070.TRAINING_DS_FI AS
SELECT * FROM LIB07070.TRAINING_DS;
QUIT;


TITLE 'LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_HISTORY VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( LOAN_HISTORY IS MISSING );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( LOAN_HISTORY IS MISSING );
QUIT;
```

Screenshot(s)

LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_HISTORY VARIABLE

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
LP001034	Male	Not Married	1	Under Graduate	No	3596	0	100	240	.	City	Y
LP001052	Male	Married	1	Graduate		3717	2925	151	360	.	Town	N
LP001091	Male	Married	1	Graduate		4166	3369	201	360	.	City	N
LP001123	Male	Married	0	Graduate	No	2400	0	75	360	.	City	Y
LP001264	Male	Married	3+	Under Graduate	Yes	3333	2166	130	360	.	Town	Y
LP001273	Male	Married	0	Graduate	No	6000	2250	265	360	.	Town	N
LP001280	Male	Married	2	Under Graduate	No	3333	2000	99	360	.	Town	Y
LP001326	Male	Not Married	0	Graduate		6782	0	.	360	.	City	N
LP001405	Male	Married	1	Graduate	No	2214	1398	85	360	.	City	Y
LP001443	Female	Not Married	0	Graduate	No	3692	0	93	360	.	Village	Y
LP001465	Male	Married	0	Graduate	No	6080	2569	182	360	.	Village	N
LP001469	Male	Not Married	0	Graduate	Yes	20166	0	650	480	.	City	Y
LP001541	Male	Married	1	Graduate	No	6000	0	160	360	.	Village	Y
LP001634	Male	Not Married	0	Graduate	No	1916	5063	67	360	.	Village	N
LP001643	Male	Married	0	Graduate	No	2383	2138	58	360	.	Village	Y
LP001671	Female	Married	0	Graduate	No	3416	2816	113	360	.	Town	Y
LP001734	Female	Married	2	Graduate	No	4283	2383	127	360	.	Town	Y

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
50

-----END-----

Explanation

The LOAN_HISTORY variable is checked for missing data and the rows with missing values in the variable are listed.

STEP 2: Display Median value

SAS Codes

```
TITLE 'STEP 2: DISPLAY MEDIAN';
PROC SQL;
SELECT
MEDIAN(LOAN_HISTORY) LABEL = 'MEDIAN-LOAN HISTORY'
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( LOAN_HISTORY IS NOT MISSING ) OR
        ( LOAN_HISTORY NE . ) );
QUIT;
```

Screenshot(s)

STEP 2: DISPLAY MEDIAN

MEDIAN-LOAN HISTORY
1

Explanation

Since LOAN_HISTORY is a numeric categorical variable, no secondary table is required. Median can be used for the imputation of the missing data.

STEP 3: Impute the missing values found in the LOAN_HISTORY variable

SAS Codes

```
TITLE 'STEP 3: Impute the missing values found in the LOAN_HISTORY variable';

PROC SQL;
CREATE TABLE LIB07070.TRAINING_DS_FI_LH AS
SELECT *
FROM LIB07070.TRAINING_DS_FI;

QUIT;

PROC SQL;

UPDATE LIB07070.TRAINING_DS_FI_LH
SET LOAN_HISTORY = ( SELECT MEDIAN(ti.LOAN_HISTORY) Label 'Loan Median'
                     FROM LIB07070.TRAINING_DS_FI ti
                     WHERE ( ( ti.LOAN_HISTORY IS NOT MISSING ) OR
                             ( ti.LOAN_HISTORY NE . ) ) ) /* It is a sub-program to find median value */

WHERE ( ( LOAN_HISTORY IS MISSING ) OR
        ( LOAN_HISTORY EQ . ) );

QUIT;
```

Screenshot(s)

```
PROC SQL;

UPDATE LIB07070.TRAINING_DS_FI_LH
SET LOAN_HISTORY = ( SELECT MEDIAN(ti.LOAN_HISTORY) Label 'Loan Median'
FROM LIB07070.TRAINING_DS_FI ti
WHERE ( ( ti.LOAN_HISTORY IS NOT MISSING ) OR
( ti.LOAN_HISTORY NE . ) ) ) /* It is a sub-program to find median value */

WHERE ( ( LOAN_HISTORY IS MISSING ) OR
( LOAN_HISTORY EQ . ) );
NOTE: 50 rows were updated in LIB07070.TRAINING_DS_FI_LH.


QUIT;
```

Explanation

The missing values in EMPLOYMENT variable are imputed using the median. 50 rows were updated in this imputation step.

STEP 4: After imputing missing values, list the observations with missing values in LOAN_HISTORY variable

SAS Codes

```
TITLE 'STEP 4: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_HISTORY VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;


SELECT *
FROM LIB07070.TRAINING_DS_FI_LH
WHERE ( ( LOAN_HISTORY IS MISSING ) OR
        ( LOAN_HISTORY EQ . ) );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI_LH
WHERE ( ( LOAN_HISTORY IS MISSING ) OR
        ( LOAN_HISTORY EQ . ) );
QUIT;
```

Screenshot(s)

STEP 4: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_HISTORY VARIABLE

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
0

-----END-----

Explanation

After imputation is done, the LOAN_HISTORY variable is double-checked for missing data to ensure there are no more missing data.

7.9.5 Imputing the missing values in LOAN_AMOUNT variable

STEP 1: Listing the missing values in the LOAN_AMOUNT variable

SAS Codes

```
TITLE 'STEP 1: BEFORE IMPUTING THE MISSING VALUES, LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_HISTORY VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( LOAN_AMOUNT IS MISSING ) OR
        ( LOAN_AMOUNT EQ . ) );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( LOAN_AMOUNT IS MISSING ) OR
        ( LOAN_AMOUNT EQ . ) );
QUIT;
```

Screenshot(s)

STEP 1: BEFORE IMPUTING THE MISSING VALUES, LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_HISTORY VARIABLE

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
LP001002	Male	Not Married	0	Graduate	No	5849	0	.	360	1	City	Y
LP001106	Male	Married	0	Graduate	No	2275	2067	.	360	1	City	Y
LP001213	Male	Married	1	Graduate	No	4945	0	.	360	0	Village	N
LP001266	Male	Married	1	Graduate	Yes	2395	0	.	360	1	Town	Y
LP001326	Male	Not Married	0	Graduate		6782	0	.	360	.	City	N
LP001350	Male	Married		Graduate	No	13650	0	.	360	1	City	Y
LP001356	Male	Married	0	Graduate	No	4652	3583	.	360	1	Town	Y
LP001392	Female	Not Married	1	Graduate	Yes	7451	0	.	360	1	Town	Y
LP001449	Male	Not Married	0	Graduate	No	3865	1640	.	360	1	Village	Y
LP001682	Male	Married	3+	Under Graduate	No	3992	0	.	180	1	City	N
LP001922	Male	Married	0	Graduate	No	20667	0	.	360	1	Village	N
LP001990	Male	Not Married	0	Under Graduate	No	2000	0	.	360	1	City	N
LP002054	Male	Married	2	Under Graduate	No	3601	1590	.	360	1	Village	Y
LP002113	Female	Not Married	3+	Under Graduate	No	1830	0	.	360	0	City	N
LP002243	Male	Married	0	Under Graduate	No	3010	3136	.	360	0	City	N
LP002393	Female			Graduate	No	10047	0	.	240	1	Town	Y
LP002401	Male	Married	0	Graduate	No	2213	1125	.	360	1	City	Y
LP002533	Male	Married	2	Graduate	No	2947	1603	.	360	1	City	N
LP002697	Male	Not Married	0	Graduate	No	4680	2087	.	360	1	Town	N
LP002778	Male	Married	2	Graduate	Yes	6633	0	.	360	0	Village	N
LP002784	Male	Married	1	Under Graduate	No	2492	2375	.	360	1	Village	Y
LP002960	Male	Married	0	Under Graduate	No	2400	3800	.	180	1	City	N

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
22

-----END-----

Explanation

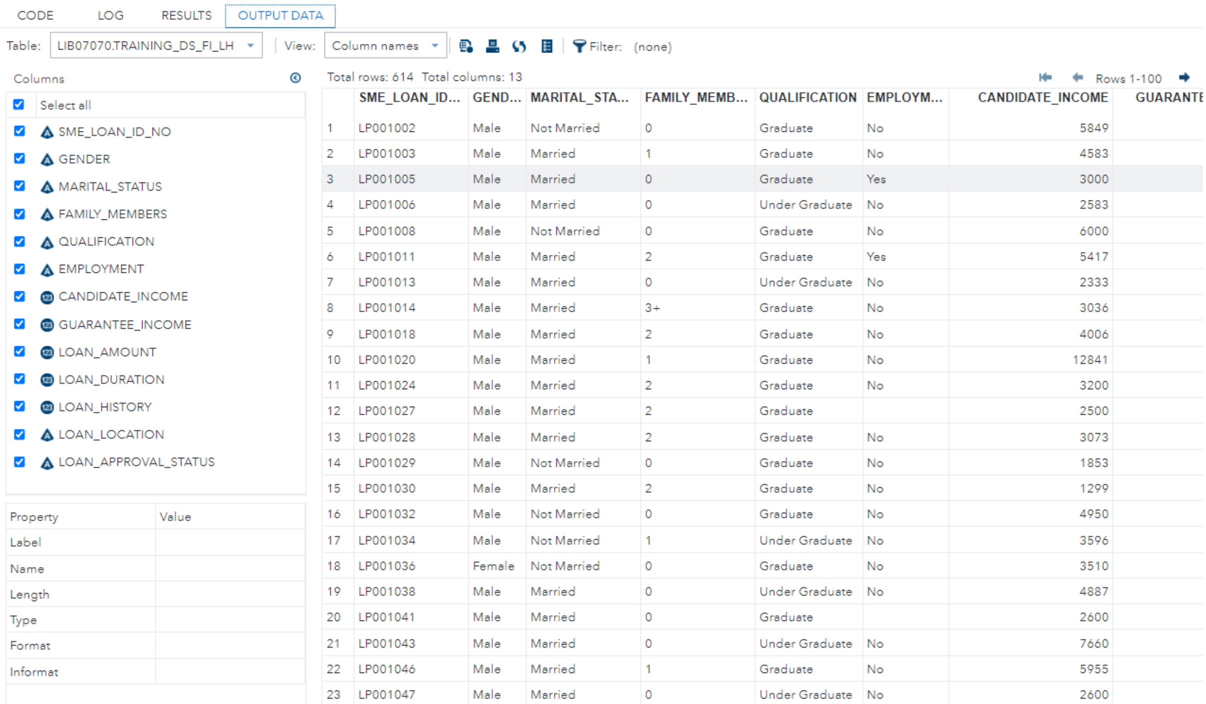
The LOAN_AMOUNT variable is checked for missing data and the rows with missing values in the variable are listed.

STEP 2: Impute missing values with mean

SAS Codes

```
TITLE 'STEP 2: IMPUTE THE MISSING VALUES IN LOAN_AMOUNT';

PROC STDIZE DATA=LIB07070.TRAINING_DS_FI_LH REPONLY

METHOD=MEAN OUT=LIB07070.TRAINING_DS_FI_LH;
VAR LOAN_AMOUNT;

QUIT;
```

Screenshot(s)



	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTE
1	LP001002	Male	Not Married	0	Graduate	No	5849	
2	LP001003	Male	Married	1	Graduate	No	4583	
3	LP001005	Male	Married	0	Graduate	Yes	3000	
4	LP001006	Male	Married	0	Under Graduate	No	2583	
5	LP001008	Male	Not Married	0	Graduate	No	6000	
6	LP001011	Male	Married	2	Graduate	Yes	5417	
7	LP001013	Male	Married	0	Under Graduate	No	2333	
8	LP001014	Male	Married	3+	Graduate	No	3036	
9	LP001018	Male	Married	2	Graduate	No	4006	
10	LP001020	Male	Married	1	Graduate	No	12841	
11	LP001024	Male	Married	2	Graduate	No	3200	
12	LP001027	Male	Married	2	Graduate		2500	
13	LP001028	Male	Married	2	Graduate	No	3073	
14	LP001029	Male	Not Married	0	Graduate	No	1853	
15	LP001030	Male	Married	2	Graduate	No	1299	
16	LP001032	Male	Not Married	0	Graduate	No	4950	
17	LP001034	Male	Not Married	1	Under Graduate	No	3596	
18	LP001036	Female	Not Married	0	Graduate	No	3510	
19	LP001038	Male	Married	0	Under Graduate	No	4887	
20	LP001041	Male	Married	0	Graduate		2600	
21	LP001043	Male	Married	0	Under Graduate	No	7660	
22	LP001046	Male	Married	1	Graduate	No	5955	
23	LP001047	Male	Married	0	Under Graduate	No	2600	

Explanation

As LOAN_AMOUNT is a numeric continuous variable, mean imputation will be used to impute the missing values in the variable.

STEP 3: After imputing missing values, list the observations with missing values in LOAN_AMOUNT variable

SAS Codes

```
TITLE 'STEP 3: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_AMOUNT VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;


SELECT *
FROM LIB07070.TRAINING_DS_FI_LH
WHERE ( ( LOAN_AMOUNT IS MISSING ) OR
        ( LOAN_AMOUNT EQ . ) );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI_LH
WHERE ( ( LOAN_AMOUNT IS MISSING ) OR
        ( LOAN_AMOUNT EQ . ) );
QUIT;
```

Screenshot(s)

STEP 3: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_AMOUNT VARIABLE

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
0

-----END-----

Explanation

After imputation is done, the LOAN_AMOUNT variable is double-checked for missing data to ensure there are no more missing data.

7.9.6 Imputing the missing values in LOAN_DURATION variable

STEP 1: Listing the missing values in the LOAN_DURATION variable

SAS Codes

```
TITLE 'STEP 1: BEFORE IMPUTING THE MISSING VALUES, LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_AMOUNT VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( LOAN_AMOUNT IS MISSING ) OR
        ( LOAN_AMOUNT EQ . ) );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( LOAN_AMOUNT IS MISSING ) OR
        ( LOAN_AMOUNT EQ . ) );
QUIT;
```

Screenshot(s)

STEP 1: BEFORE IMPUTING THE MISSING VALUES, LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_DURATION VARIABLE

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
LP001041	Male	Married	0	Graduate		2600	3500	115	.	1	City	Y
LP001109	Male	Married	0	Graduate	No	1828	1330	100	.	0	City	N
LP001136	Male	Married	0	Under Graduate	Yes	4695	0	96	.	1	City	Y
LP001137	Female	Not Married	0	Graduate	No	3410	0	88	.	1	City	Y
LP001250	Male	Married	3+	Under Graduate	No	4755	0	95	.	0	Town	N
LP001391	Male	Married	0	Under Graduate	No	3572	4114	152	.	0	Village	N
LP001574	Male	Married	0	Graduate	No	3707	3166	182	.	1	Village	Y
LP001669	Female	Not Married	0	Under Graduate	No	1907	2365	120	.	1	City	Y
LP001749	Male	Married	0	Graduate	No	7578	1010	175	.	1	Town	Y
LP001770	Male	Not Married	0	Under Graduate	No	3189	2598	120	.	1	Village	Y
LP002106	Male	Married		Graduate	Yes	5503	4490	70	.	1	Town	Y
LP002188	Male	Not Married	0	Graduate	No	5124	0	124	.	0	Village	N
LP002357	Female	Not Married	0	Under Graduate	No	2720	0	80	.	0	City	N
LP002362	Male	Married	1	Graduate	No	7250	1667	110	.	0	City	N

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
14

-----END-----

Explanation

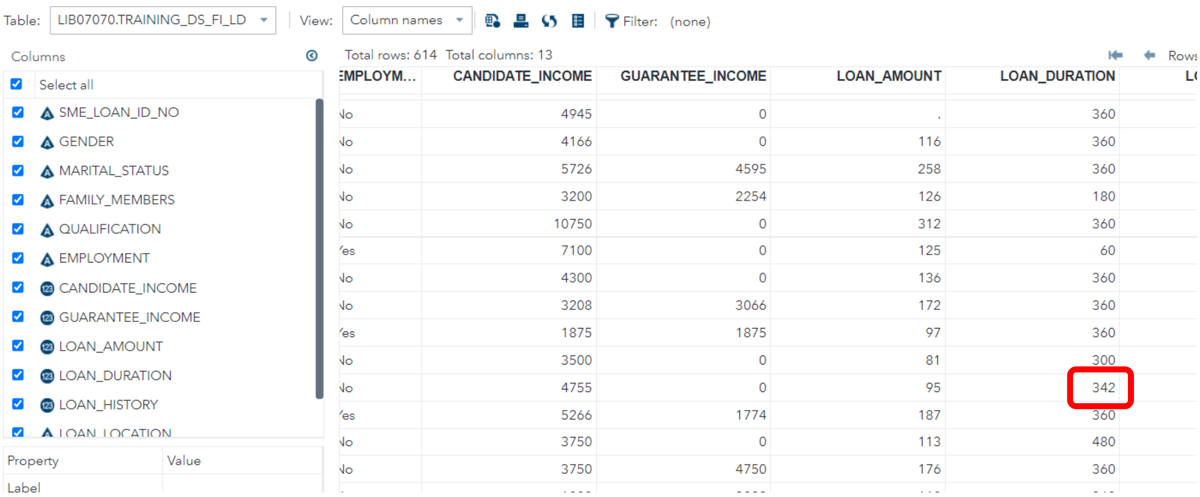
The LOAN_DURATION variable is checked for missing data and the rows with missing values in the variable are listed.

STEP 2: Impute missing values with mean

SAS Codes

```
TITLE 'STEP 2: IMPUTE THE MISSING VALUES IN LOAN_DURATION';

PROC STDIZE DATA=LIB07070.TRAINING_DS_FI REPONLY

METHOD=MEAN OUT=LIB07070.TRAINING_DS_FI_LD;
VAR LOAN_DURATION;

QUIT;
```

Screenshot(s)



Explanation

As LOAN_DURATION is a numeric continuous variable, mean imputation will be used to impute the missing values in the variable.

STEP 3: After imputing missing values, list the observations with missing values in LOAN_DURATION variable

SAS Codes

```
TITLE 'STEP 3: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_DURATION VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;


SELECT *
FROM LIB07070.TRAINING_DS_FI_LD
WHERE ( ( LOAN_DURATION IS MISSING ) OR
        ( LOAN_DURATION EQ . ) );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI_LD
WHERE ( ( LOAN_DURATION IS MISSING ) OR
        ( LOAN_DURATION EQ . ) );
QUIT;
```

Screenshot(s)

STEP 3: AFTER IMPUTING MISSING VALUES: LIST THE OBSERVATIONS WITH MISSING VALUES IN LOAN_DURATION VARIABLE

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
0

-----END-----

Explanation

After imputation is done, the LOAN_DURATION variable is double-checked for missing data to ensure there are no more missing data.

7.9.7 Imputing the missing values in FAMILY_MEMBERS variable

STEP 1: List the observations with missing values in FAMILY_MEMBERS variable

SAS Codes

```
TITLE 'STEP 1: LIST THE OBSERVATIONS WITH MISSING VALUES IN FAMILY_MEMBERS VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( FAMILY_MEMBERS IS MISSING ) OR
        ( FAMILY_MEMBERS IS NULL ) OR
        ( FAMILY_MEMBERS EQ '' ) );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( FAMILY_MEMBERS IS MISSING ) OR
        ( FAMILY_MEMBERS IS NULL ) OR
        ( FAMILY_MEMBERS EQ '' ) );
QUIT;
```

Screenshot(s)

STEP 1: LIST THE OBSERVATIONS WITH MISSING VALUES IN FAMILY_MEMBERS VARIABLE

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
LP001350	Male	Married		Graduate	No	13650	0	.	360	1	City	Y
LP001357	Male			Graduate	No	3816	754	160	360	1	City	Y
LP001426	Male	Married		Graduate	No	5667	2667	180	360	1	Village	Y
LP001754	Male	Married		Under Graduate	Yes	4735	0	138	360	1	City	N
LP001760	Male			Graduate	No	4758	0	158	480	1	Town	Y
LP001945	Female	Not Married		Graduate	No	5417	0	143	480	0	City	N
LP001972	Male	Married		Under Graduate	No	2875	1750	105	360	1	Town	Y
LP002100	Male	Not Married		Graduate	No	2833	0	71	360	1	City	Y
LP002106	Male	Married		Graduate	Yes	5503	4490	70	.	1	Town	Y
LP002130	Male	Married		Under Graduate	No	3523	3230	152	360	0	Village	N
LP002144	Female	Not Married		Graduate	No	3813	0	116	180	1	City	Y
LP002393	Female			Graduate	No	10047	0	.	240	1	Town	Y
LP002682	Male	Married		Under Graduate	No	3074	1800	123	360	0	Town	N
LP002847	Male	Married		Graduate	No	5116	1451	165	360	0	City	N
LP002943	Male	Not Married		Graduate	No	2987	0	88	360	0	Town	N

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
15

-----END-----

Explanation

The FAMILY_MEMBERS variable is checked for missing data and the rows with missing values in the variable are listed.

STEP 2: Display the details of applicants with 3+ family members

SAS Codes

```
TITLE 'STEP 2 : DISPLAY THE DETAILS OF APPLICANTS WITH 3+ FAMILY MEMBERS';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( SUBSTR(FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;


PROC SQL;
SELECT COUNT(*) Label 'No of Applicants'
FROM LIB07070.TRAINING_DS_FI
WHERE ( SUBSTR(FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;
```

Screenshot(s)

STEP 2 : DISPLAY THE DETAILS OF APPLICANTS WITH 3+ FAMILY MEMBERS

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
LP001014	Male	Married	3+	Graduate	No	3036	2504	158	360	0	Town	N
LP001100	Male	Not Married	3+	Graduate	No	12500	3000	320	360	1	Village	N
LP001206	Male	Married	3+	Graduate	No	3029	0	99	360	1	City	Y
LP001238	Male	Married	3+	Under Graduate	Yes	7100	0	125	60	1	City	Y
LP001250	Male	Married	3+	Under Graduate	No	4755	0	95	.	0	Town	N
LP001253	Male	Married	3+	Graduate	Yes	5266	1774	187	360	1	Town	Y

STEP 2 : DISPLAY THE DETAILS OF APPLICANTS WITH 3+ FAMILY MEMBERS

No of Applicants
51

-----END-----

Explanation

The observations of the applicants with 3 or more family members are listed out and counted. There are total 51 applicants with 3 or more family members

STEP 3 : Replace 3+ with 3

SAS Codes

```
TITLE 'STEP 3 : Replace 3+ with 3';

PROC SQL;
UPDATE LIB07070.TRAINING_DS_FI
SET FAMILY_MEMBERS = SUBSTR(FAMILY_MEMBERS,1,1)
WHERE ( SUBSTR(FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;
```

Screenshot(s)

```
PROC SQL;
UPDATE LIB07070.TRAINING_DS_FI
SET FAMILY_MEMBERS = SUBSTR(FAMILY_MEMBERS,1,1)
WHERE ( SUBSTR(FAMILY_MEMBERS,2,1) EQ '+' );
NOTE: 51 rows were updated in LIB07070.TRAINING_DS_FI.

QUIT;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.00 seconds
      user cpu time       0.01 seconds
```

Explanation

The value 3+ have to be replaced with 3 because with the + symbol, it makes the variable become string variable while the other values are numeric. By converting this, it makes the variable as numerical variable before proceeding to the next step.

STEP 4: AFTER REPLACING THE 3+ WITH 3

SAS Codes

```
TITLE 'STEP 4 : After replacing 3+ with 3';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( SUBSTR(FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;


PROC SQL;
SELECT COUNT(*) Label 'No of Applicants'
FROM LIB07070.TRAINING_DS_FI
WHERE ( SUBSTR(FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;
```

Screenshot(s)

STEP 4 : After replacing 3+ with 3

-----END-----

STEP 4 : After replacing 3+ with 3

No of Applicants
0

-----END-----

Explanation

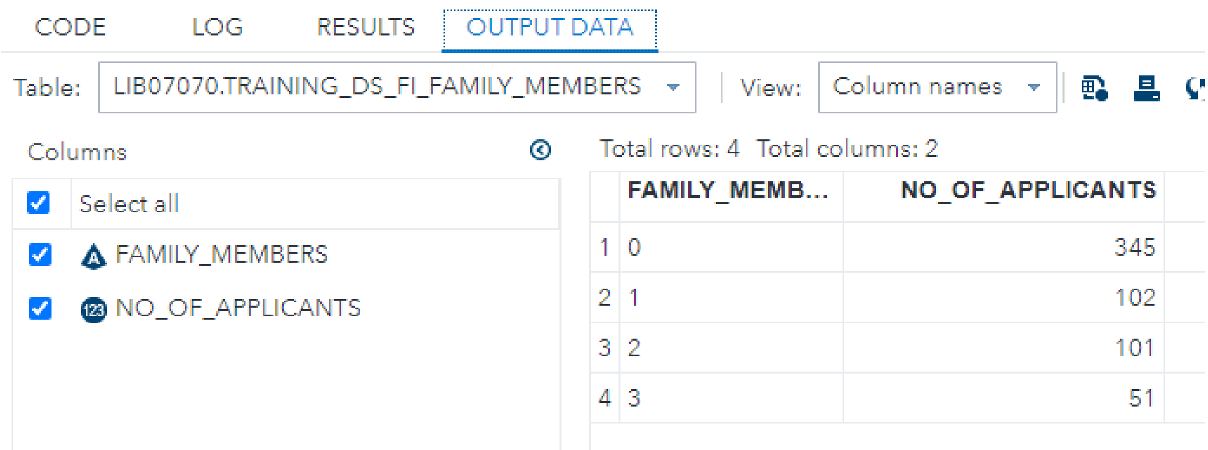
All the 3+ values are replaced with number 3.

STEP 5: Create a dataset to hold the family members and number of applicants

SAS Codes

```
TITLE 'STEP 5: Create a dataset to hold the family members and number of applicants';

PROC SQL;

CREATE TABLE LIB07070.TRAINING_DS_FI_FAMILY_MEMBERS AS
SELECT FAMILY_MEMBERS, COUNT (*) AS NO_OF_APPLICANTS
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( FAMILY_MEMBERS IS NOT MISSING ) OR
        ( FAMILY_MEMBERS IS NOT NULL ) OR
        ( FAMILY_MEMBERS NE '' ) )
GROUP BY FAMILY_MEMBERS;

QUIT;
```

Screenshot(s)



Explanation

A secondary data table is created is tabulate the number of applicants according to the number of their family members.

STEP 6: Display the contents of the dataset LIB07070.TRAINING_DS_FI_FAMILY_MEMBERS

SAS Codes

```
TITLE 'STEP 6: Display the contents of the dataset LIB07070.TRAINING_DS_FI_FAMILY_MEMBERS';

PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI_FAMILY_MEMBERS;

QUIT;
```

Screenshot(s)

STEP 6: Display the contents of the dataset LIB07070.TRAINING_DS_FI_FAMILY_MEMBERS

FAMILY_MEMBERS	NO_OF_APPLICANTS
0	345
1	102
2	101
3	51

Explanation

The secondary data table is viewed after being created. The highest frequency for the variable is zero family members.

STEP 7: Create a dataset to hold the family members and number of applicants

SAS Codes

```
TITLE 'STEP 7: Impute the missing values found in the FAMILY_MEMBERS variable';
PROC SQL;

UPDATE LIB07070.TRAINING_DS_FI
SET FAMILY_MEMBERS = ( SELECT FAMILY_MEMBERS
                       FROM LIB07070.TRAINING_DS_FI_FAMILY_MEMBERS
                       WHERE NO_OF_APPLICANTS EQ ( SELECT MAX(NO_OF_APPLICANTS) Label 'NO OF APPLICANTS'
                                                    FROM LIB07070.TRAINING_DS_FI_FAMILY_MEMBERS ) )
                                                    /*It is a sub-program to find the highest no of applicants*/
WHERE ( ( FAMILY_MEMBERS IS MISSING ) OR
        ( FAMILY_MEMBERS IS NULL ) OR
        ( FAMILY_MEMBERS EQ '' ) );
QUIT;
```

Screenshot(s)

```
78          WHERE ( ( FAMILY_MEMBERS IS MISSING ) OR
79          ( FAMILY_MEMBERS IS NULL ) OR
80          ( FAMILY_MEMBERS EQ '' ) );
NOTE: 15 rows were updated in LIB07070.TRAINING_DS_FI.

81          QUIT;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.01 seconds
```

Explanation

By using the mode imputation, the missing values are imputed using the 0 value from the secondary data table in previous step.

STEP 8: After imputing missing values, list the observations with missing values in FAMILY_MEMBERS variable

SAS Codes

```
TITLE 'STEP 8: AFTER IMPUTING THE MISSING VALUES, LIST THE OBSERVATIONS WITH MISSING VALUES IN FAMILY_MEMBERS VARIABLE';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT *
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( FAMILY_MEMBERS IS MISSING ) OR
        ( FAMILY_MEMBERS IS NULL ) OR
        ( FAMILY_MEMBERS EQ '' ) );
QUIT;

TITLE 'NUMBER OF OBSERVATIONS WITH MISSING VALUES';
FOOTNOTE '-----END-----';
PROC SQL;

SELECT COUNT (*) LABEL 'NUMBER OF OBSERVATIONS WITH MISSING VALUES'
FROM LIB07070.TRAINING_DS_FI
WHERE ( ( FAMILY_MEMBERS IS MISSING ) OR
        ( FAMILY_MEMBERS IS NULL ) OR
        ( FAMILY_MEMBERS EQ '' ) );
QUIT;
```

Screenshot(s)

STEP 8: AFTER IMPUTING THE MISSING VALUES, LIST THE OBSERVATIONS WITH MISSING VALUES IN FAMILY_MEMBERS VARIABLE

-----END-----

NUMBER OF OBSERVATIONS WITH MISSING VALUES

NUMBER OF OBSERVATIONS WITH MISSING VALUES
0

-----END-----

Explanation

After imputation is done, the FAMILY_MEMBERS variable is double-checked for missing data to ensure there are no more missing data.

7.10 SAS MACRO

7.10.1 Univariate Analysis of the categorical variable using SAS MACRO

<u>SAS Codes</u>

```
/* MACRO MACRO_FOR_UNIVARIATE ANALYSIS BEGINS HERE */

%MACRO MACRO_UVA_LIB07070_TESTING_DS(PDS_NAME, PVARI_NAME, PTITLE_NAME);
PROC FREQ DATA = &PDS_NAME;

TABLE &PVARI_NAME;
TITLE &PTITLE_NAME;

QUIT;

%MEND MACRO_UVA_LIB07070_TESTING_DS;

/* MACRO MACRO_FOR_UNIVARIATE ANALYSIS ENDS HERE */

/*CALL/RUN THE SAS MACRO */

%MACRO_UVA_LIB07070_TESTING_DS (LIB07070.TESTING_DS, EMPLOYMENT, "UNIVARIATE ANALYSIS OF THE CATEGORICAL VARIABLE - EMPLOYMENT");
%MACRO_UVA_LIB07070_TESTING_DS (LIB07070.TESTING_DS, GENDER, "UNIVARIATE ANALYSIS OF THE CATEGORICAL VARIABLE - GENDER");
%MACRO_UVA_LIB07070_TESTING_DS (LIB07070.TESTING_DS, QUALIFICATION, "UNIVARIATE ANALYSIS OF THE CATEGORICAL VARIABLE - QUALIFICATION");
%MACRO_UVA_LIB07070_TESTING_DS (LIB07070.TESTING_DS, MARITAL_STATUS, "UNIVARIATE ANALYSIS OF THE CATEGORICAL VARIABLE - MARITAL_STATUS");
```

<u>Screenshot(s)</u>

UNIVARIATE ANALYSIS OF THE CATEGORICAL VARIABLE - EMPLOYMENT

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				

UNIVARIATE ANALYSIS OF THE CATEGORICAL VARIABLE - GENDER

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				

UNIVARIATE ANALYSIS OF THE CATEGORICAL VARIABLE - QUALIFICATION

The FREQ Procedure

QUALIFICATION	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Graduate	283	77.11	283	77.11
Under Graduate	84	22.89	367	100.00

INIVARIATE ANALYSIS OF THE CATEGORICAL VARIABLE - MARITAL_STATUS

The FREQ Procedure

MARITAL_STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Married	233	63.49	233	63.49
Not Married	134	36.51	367	100.00

7.10.2 Univariate Analysis of the continuous variable using SAS MACRO

<u>SAS Codes</u>

```
/* SAS MACRO FOR UNIVARIATE ANALYSIS OF CONTINUOUS VARIABLES*/

%MACRO MACRO_UNIV_CONT_VARI(PDS_NAME, PVARI_NAME, PTITLE_1, PTITLE_NAME_2);

TITLE &PTITLE_1;
PROC MEANS DATA = &PDS_NAME N NMISS MIN MAX MEAN MEDIAN STD;
VAR &PVARI_NAME;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = &PDS_NAME;
HISTOGRAM &PVARI_NAME;
TITLE &PTITLE_NAME_2;
RUN;

%MEND MACRO_UNIV_CONT_VARI;

/* To call the SAS MACRO MACRO_UNIV_CONT_VARI */
%MACRO_UNIV_CONT_VARI (LIB07070.TESTING_DS, LOAN_AMOUNT,
'Figure 7.8.3 Univariate Analysis variable: LOAN_AMOUNT',
'Figure 7.8.3 Univariate Analysis variable: LOAN_AMOUNT');

%MACRO_UNIV_CONT_VARI (LIB07070.TESTING_DS, CANDIDATE_INCOME,
'Figure 7.8.3 Univariate Analysis variable: CANDIDATE_INCOME',
'Figure 7.8.3 Univariate Analysis variable: CANDIDATE_INCOME');

%MACRO_UNIV_CONT_VARI (LIB07070.TESTING_DS, GUARANTEE_INCOME,
'Figure 7.8.3 Univariate Analysis variable: GUARANTEE_INCOME',
'Figure 7.8.3 Univariate Analysis variable: GUARANTEE_INCOME');
```

Explanation

SAS MACROS is a programming feature inside SAS studio. It can help the programmer to save a lot of time doing coding because it can help to run repetitive sections of codes without needing to repeat coding.

In this case, SAS MACROS are used to run univariate analysis of the variables in the dataset.

7.10.3 Bivariate Analysis of the categorical variable using SAS MACRO

SAS Codes

```
/* SAS MACRO FOR BIVARIATE ANALYSIS OF CATEGORICAL VARIABLES*/
%MACRO MACRO_BVAR_CATEG_VARI_TP063332(PDS_NAME,PVARI_1,PVARI_2,PTITLE_1,PTITLE_2);

PROC FREQ DATA = &PDS_NAME;

TABLE &PVARI_1 * &PVARI_2 /
PLOTS=FREQPLOT(TWOWAY=STACKED SCALE=GROUPPCT);
TITLE1 &PTITLE_1;
TITLE2 &PTITLE_2;

RUN;

%MEND MACRO_BVAR_CATEG_VARI_TP063332;

/* To call the macro - MACRO_BVAR_CATEG_VARI_TP063332*/

%MACRO_BVAR_CATEG_VARI_TP063332(LIB07070.TESTING_DS,
MARITAL_STATUS,LOAN_LOCATION,"BIVARIATE ANALYSIS OF CATEGORICAL VARIABLES","MARITAL_STATUS-Categorical vs LOAN_LOCATION-Categorical")

%MACRO_BVAR_CATEG_VARI_TP063332(LIB07070.TESTING_DS,
EMPLOYMENT,LOAN_HISTORY,"BIVARIATE ANALYSIS OF CATEGORICAL VARIABLES","EMPLOYMENT-Categorical vs LOAN_HISTORY-Categorical")

%MACRO_BVAR_CATEG_VARI_TP063332(LIB07070.TESTING_DS,
GENDER,FAMILY_MEMBERS,"BIVARIATE ANALYSIS OF CATEGORICAL VARIABLES","GENDER-Categorical vs FAMILY_MEMBERS-Categorical")

%MACRO_BVAR_CATEG_VARI_TP063332(LIB07070.TESTING_DS,
GENDER,LOAN_LOCATION,"BIVARIATE ANALYSIS OF CATEGORICAL VARIABLES","GENDER-Categorical vs LOAN_LOCATION-Categorical")
```

Explanation

SAS MACROS are also used to carry out the bivariate analysis on the variables in the dataset to find any relationships between the variables.

7.11 Variables with missing values found in the LIB07070.TESTING_DS

7.11.1 Finding the variables with missing data before imputation

SAS Codes

```
TITLE 'Before imputing the missing values, find the categorical and continuous variables with missing values';
PROC FORMAT;

VALUE $missfmt ' ' = 'Missing' others = 'Not missing';
VALUE  missfmt .   = 'Missing' others = 'Not missing';

RUN;

PROC FREQ DATA=LIB07070.TESTING_DS;

FORMAT _CHAR_ $missfmt.;
FORMAT _NUMERIC_ missfmt.;

TABLE _CHAR_ / missing nocum nopercent;
TABLE _NUMERIC_ / missing nocum nopercent;

RUN;
```

Screenshot(s)

LOAN_AMOUNT	Frequency
Missing	5
Not missing	362

LOAN_DURATION	Frequency
Missing	6
Not missing	361

GENDER	Frequency
Missing	11
Female	70
Male	286

FAMILY_MEMBERS	Frequency
Missing	10
0	200
1	58
2	59
3+	40

LOAN_HISTORY	Frequency
Missing	29
Not missing	338

EMPLOYMENT	Frequency
Missing	23
No	307
Yes	37

LOAN_APPROVAL_STATUS	Frequency
Missing	367

Explanation

Similar to the TRAINING dataset, TESTING dataset also have missing values, which will be imputed with the same methods as the TRAINING dataset.

7.11.2 Checking all variables to make sure all missing data are imputed

SAS Codes

```
TITLE 'Before imputing the missing values, find the categorical and continuous variables with missing values';
PROC FORMAT;

VALUE $missfmt ' ' = 'Missing' others = 'Not missing';
VALUE  missfmt .   = 'Missing' others = 'Not missing';

RUN;

PROC FREQ DATA=LIB07070.TESTING_DS;

FORMAT _CHAR_ $missfmt.;
FORMAT _NUMERIC_ missfmt.;

TABLE _CHAR_ / missing nocum nopercent;
TABLE _NUMERIC_ / missing nocum nopercent;

RUN;
```

Screenshot(s)

GENDER	Frequency
Female	70
Male	297

MARITAL_STATUS	Frequency
Married	233
Not Married	134

FAMILY_MEMBERS	Frequency
0	210
1	58
2	59
3	40

QUALIFICATION	Frequency
Graduate	283
Under Gradu	84

EMPLOYMENT	Frequency
No	330
Yes	37

LOAN_LOCATION	Frequency
City	140
Town	116
Village	111

CANDIDATE_INCOME	Frequency
Not missing	367

GUARANTEE_INCOME	Frequency
Not missing	367

LOAN_AMOUNT	Frequency
Not missing	367

LOAN_DURATION	Frequency
Not missing	367

LOAN_HISTORY	Frequency
Not missing	367

Explanation

Similar as the TRAINING dataset, all the missing values in the TESTING dataset are made sure imputed before proceeding to build the logistic regression model in the next step.

7.12 Building a Logistic Regression Model

7.13.1 Build a logistic regression model using the dataset LIB07070.TRAINING_DS_FI_LH

SAS Codes

```
/*****BUILD LOGISTIC REGRESSION*******/
PROC LOGISTIC DATA=LIB07070.TRAINING_DS_FI_LH OUTMODEL=LIB07070.TRAINING_DS_FI_LH_MODEL;
CLASS
GENDER
LOAN_HISTORY
MARITAL_STATUS
QUALIFICATION
LOAN_LOCATION
FAMILY_MEMBERS
EMPLOYMENT;

/* Above are categorical variables */
MODEL LOAN_APPROVAL_STATUS = /*place here all independent variables */
/* LOAN_APPLICATION_STATUS is a dependent variable */

GENDER
LOAN_LOCATION
MARITAL_STATUS
QUALIFICATION
FAMILY_MEMBERS
LOAN_HISTORY
EMPLOYMENT
CANDIDATE_INCOME
GUARANTEE_INCOME
LOAN_AMOUNT
LOAN_DURATION;

OUTPUT OUT = LIB07070.TRAINING_DS_FI_LH_OUT P = PRED_PROB;
/*PRED_PROB ->PRedicted probability - variable to hold predicted probability
OUT -> the output will be stored in the dataset
Akaike Information criterion must ( AIC ) < SC (Schwarz Criterion)
*/
RUN;
```

Output(s)

The LOGISTIC Procedure

Model Information	
Data Set	LIB07070.TRAINING_DS_FI_LH
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 number of observations read and used are matched with both showing 614 observations, this indicated that the training dataset is imputed well and has no missing data. It also predicted that 192 applications were rejected while 422 applications were accepted.

Model Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	764.891	587.154
SC	769.311	653.454
-2 Log L	762.891	557.154

To validate that model created is valid, Akaike Information Criterion (AIC) value must be lower than Schwarz Criterion (SC). The convergence criterion is also satisfied.

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
LOAN_LOCATION	City	1	0.1559	0.1519	1.0538	0.3046
LOAN_LOCATION	Town	1	-0.5313	0.1575	11.3806	0.0007
MARITAL_STATUS	Married	1	-0.2915	0.1264	5.3173	0.0211
QUALIFICATION	Graduate	1	-0.2052	0.1299	2.4952	0.1142
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
LOAN_HISTORY	0	1	1.9696	0.2106	87.4798	<.0001
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

If Pr > ChiSq is <=0.05, it means that the independent variable is an important variable for the dependent variable prediction. In this case, LOAN_LOCATION, MARITAL_STATUS and LOAN_HISTORY is important for the prediction.

7.12.1 Predict the Approval Status using the logistic regression model created

SAS Codes

```
/**********PREDICTION MODEL USING LRA*****************/
TITLE 'Prediction Model Using the Logistic Regression';
FOOTNOTE '-----------END-----------';

PROC LOGISTIC INMODEL=LIB07070.TRAINING_DS_FI_LH_MODEL;

SCORE DATA=LIB07070.TESTING_DS_FI
OUT=LIB07070.TESTING_DS_FI_PREDICTION;

QUIT;
```

```
TITLE 'Number of Loans Approved';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label "NUMBER OF OBSERVATIONS WITH 'Y'"
FROM LIB07070.TESTING_DS_FI_PREDICTION
WHERE ( I_LOAN_APPROVAL_STATUS EQ 'Y' );

QUIT;

TITLE 'Number of Loans Rejected';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label "NUMBER OF OBSERVATIONS WITH 'N'"
FROM LIB07070.TESTING_DS_FI_PREDICTION
WHERE ( I_LOAN_APPROVAL_STATUS EQ 'N' );

QUIT;
```

Output(s)

From: LOAN_APPROVAL_STATUS	Into: LOAN_APPROVAL_STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
	Y	0.15823	0.84177
	Y	0.257444	0.742556
	Y	0.158193	0.841807
	Y	0.140894	0.859106
	Y	0.329375	0.670625
	Y	0.28222	0.71778
	Y	0.272703	0.727297
	N	0.93692	0.06308
	Y	0.131183	0.868817
	Y	0.236009	0.763991

Number of Loans Approved

NUMBER OF OBSERVATIONS WITH 'Y'
306

-----END-----

Number of Loans Rejected

NUMBER OF OBSERVATIONS WITH 'N'
61

-----END-----

<u>Explanation</u>

Through the logistic regression model, the loan approval status is predicted. As we can observed from the screenshot above, when the probability of N is more than 0.5, the loan approval status will be N, which means rejected. Otherwise, when the probability of Y is more than 0.5, the loan approval status will be Y, indicating that the loan will be approved. A total of 306 applications were predicted to be approved while only 61 of the applications were predicted to be rejected.

7.13 Output Delivery System (ODS)

In SAS Studio, the SAS output are only designed like a traditional typewriter. This output has some limitations where not everyone is able to access easily. By using the Output Delivery System (ODS) in SAS, it is a method of delivering the outputs in a number of formats. Some of the formats included are like Portable Document Format (PDF) and HTML etc.

After creating the predicted dataset, the outputs can then be delivered to the library folder as PDF for easy access or even create a view for the other data scientists.

7.13.1 Creating VIEW for other users

SAS Codes

```
PROC SQL;
CREATE VIEW LIB07070.VIEW_FOR_WAYNE AS
SELECT SME_LOAN_ID_NO,
GENDER,
FAMILY_MEMBERS,
EMPLOYMENT,
QUALIFICATION
FROM LIB07070.TESTING_DS_FI_PREDICTION;

QUIT;
```

Creating a VIEW for another user

```
PROC DATASETS library=LIB07070 memtype=VIEW;
RUN;
```

Output(s)

Directory	
Libref	LIB07070
Engine	V9
Physical Name	/home/u58868125/sasuser.v94/DAP_FT_SEP_2021_TP063332
Filename	/home/u58868125/sasuser.v94/DAP_FT_SEP_2021_TP063332
Inode Number	235438221
Access Permission	rwxr-xr-x
Owner Name	u58868125
File Size	4KB
File Size (bytes)	4096

#	Name	Member Type	File Size	Last Modified
1	VIEW_FOR_WAYNE	VIEW	136KB	12/17/2021 18:26:37

The user are able to view the VIEW created for him/her

SME_LOAN_ID_NO	GENDER	FAMILY_MEMBERS	EMPLOYMENT	QUALIFICATION
LP001015	Male	0	No	Graduate
LP001022	Male	1	No	Graduate
LP001031	Male	2	No	Graduate
LP001035	Male	2	No	Graduate
LP001051	Male	0	No	Under Graduate
LP001054	Male	0	Yes	Under Graduate
LP001055	Female	1	No	Under Graduate
LP001056	Male	2	No	Under Graduate
LP001059	Male	2	No	Graduate
LP001067	Male	0	No	Under Graduate
LP001078	Male	0	No	Under Graduate

The user will only be able to view the variables included in the VIEW dataset.

7.13.2 Creating PDF for other users

<u>SAS Codes</u>

```
ODS HTML CLOSE;
ODS PDF CLOSE;

PDS PDF FILE="/home/u58868125/sasuser.v94/DAP_FT_SEP_2021_TP063332/REPORT.pdf";
OPTIONS NOBYLINE NODATE;
TITLE1 "Bank Loan Approval Status Predicted";
TITLE2 "APU,TPM";
FOOTNOTE '-----End of Report-----';

PROC REPORT DATA=LIB07070.TESTING_DS_FI_PREDICTION NOWINDOWS;

BY SME_LOAN_ID_NO; /* To separate each by SME LOAN ID NO */
/* COLUMN SME_LOAN_ID_NO I_LOAN_APPROVAL_STATUS;*/
DEFINE SME_LOAN_ID_NO / GROUP 'LOAN ID';
DEFINE I_LOAN_APPROVAL_STATUS / GROUP 'LOAN APPROVAL STATUS';
FOOTNOTE '-----End of Report-----';

RUN;
OPTIONS BYLINE;
```

<u>Output(s)</u>

Bank Loan Approval Status Predicted
APU,TPM

LOAN ID	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS	LOAN APPROVAL STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
LP001015	Male	Married	0	Graduate	No	5720	0	110	360	1	City			Y	0.1582297	0.8417703

-----End of Report-----

Bank Loan Approval Status Predicted
APU,TPM

LOAN ID	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS	LOAN APPROVAL STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
LP001022	Male	Married	1	Graduate	No	3076	1500	126	360	1	City			Y	0.2574445	0.7425555

-----End of Report-----

TION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS	LOAN APPROVAL STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
360	1	City			Y	0.1582297	0.8417703

TION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS	LOAN APPROVAL STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
360	1	City			Y	0.2574445	0.7425555

The observations are exported together with the predicted probability loan approval status and the outcome. This report can then be passed to the person-in-charge to process the loan documents.

8. CONCLUSION

In a nutshell, 2 datasets were used in this study to predict the loan approval status of the applicants, which are TRAINING_DS and TESTING_DS. The datasets are first explored by doing univariate and bivariate analysis. After that, since missing values are found in the datasets, the missing values are imputed by using either mean, median or mode imputation. After cleaning the data, logistic regression model is run on the training dataset to create the model. The predicted dataset on the testing dataset results in 422 applications approved and 192 applications rejected. The model also suggested that loan history, marital status and loan location plays an important role in predicting the outcome of the application approval status.

9. PERSONAL REFLECTION

At the end of this report, the researcher is satisfied that the prediction of the loan approval status is done using the logistic regression model in SAS studio. Compared to the beginning of the module, a better understanding of SQL programming skills and workflow was obtained through the progress of the assignment. This assignment also provided an opportunity to work on a real-life application in loan approvals. Lastly, sincere gratitude also goes to Mr. Dhason who provided his guidance during the study.

10. REFERENCES

Butaru, F., Chen, Q., Clark, B., Das, S., & Lo, A. W. (2016). Risk and risk management in the credit card industry R. *Journal of Banking and Finance*, *72*, 218–239. https://doi.org/10.1016/j.jbankfin.2016.07.015

Chen, H., & Xiang, Y. (2017). The Study of Credit Scoring Model Based on Group Lasso. *Procedia Computer Science*, *122*, 677–684. https://doi.org/10.1016/j.procs.2017.11.423

Coşer, A., Maer-Matei, M. M., & Albu, C. (2019). Predictive models for loan default risk assessment. *Economic Computation and Economic Cybernetics Studies and Research*, *53*(2), 149–165. https://doi.org/10.24818/18423264/53.2.19.09

Figini, S., Bonelli, F., & Giovannini, E. (2017). Solvency prediction for small and medium enterprises in banking. *Decision Support Systems*, *102*, 91–97. https://doi.org/10.1016/j.dss.2017.08.001

Imtiaz, S., & J., A. (2017). A Better Comparison Summary of Credit Scoring Classification. *International Journal of Advanced Computer Science and Applications*, *8*(7). https://doi.org/10.14569/ijacsa.2017.080701

Sudhamathy, G. (2016). Credit risk analysis and prediction modelling of bank loans using R.

International Journal of Engineering and Technology, *8*(5), 1954–1966. https://doi.org/10.21817/ijet/2016/v8i5/160805414

Thavarith, V., & Liangrokapart, J. (2019). The blend of credit scoring model for individual in the dmaic process for reducing non-performing loan risk. *ACM International Conference Proceeding Series*, 195–202. https://doi.org/10.1145/3335550.3335583