INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT050-3-M-DAP

DATA ANALYTICAL PROGRAMMING

APDMF2112DSBA(DE)(PR)

AUGUST 2022

TITLE: AUTOMATION OF LOAN APPROVAL PROCESS USING SAS PROGRAMMING

LEE KEAN LIM

TP065778

LECTURER: MR. DHASON PADMA KUMAR

TABLE OF CONTENTS

TABLE OF CONTENTS ii
LIST OF TABLES vi
LIST OF FIGURES ix
LIST OF ABBREVIATIONS xvi
SECTION 1: INTODUCTION 1
1.1 INTRODUCTION 1
1.2 BACKGROUND 2
1.3 AIM & OBJECTIVES 7
1.3.1 Aim 7
1.3.2 Objectives 7
1.4 ASSUMPTION AND JUSTIFICATION 7
SECTION 2: LITERATURE REVIEW 8
2.1 LEGACY SYSTEM IN LOAN APPROVAL 8
2.2 MACHINE LEARNING IN LOAN APPROVAL 9
2.2.1 Machine Learning Application 9
2.2.2 Challenges in Machine Learning Application 11
SECTION 3: DATASET 14
3.1 INTRODUCTION 14
3.2 DATA DICTONARY 16
SECTION 4: METHODOLOGY 17
4.1 INTRODUCTION 17
4.2 METHODOLOGY 17
SECTION 5: EXPERIMENTATION 19
5.1 INITIAL SETUP 19
5.1.1 Folder Creation on SAS 19

5.1.2 Uploading Datasets 20
5.1.3 Library Creation 21
5.1.4 Import Dataset to Created Library 22
5.1.4 Data Dictionary – Structure of the Dataset 24
5.2 ANALYSIS OF VARIABLES – LIB65778.TRAINING_DS 26
5.2.1 Univariate Analysis of the Categorical Variable – MARITAL_STATUS 26
5.2.2 Univariate Analysis of the Categorical Variable – QUALIFICATION 28
5.2.3 Univariate Analysis of the Categorical Variable – FAMILY_MEMBERS 30
5.2.4 Univariate Analysis of the Categorical Variable – GENDER 32
5.2.5 Univariate Analysis of the Categorical Variable – EMPLOYMENT 34
5.2.6 Univariate Analysis of the Categorical Variable – LOAN_HISTORY 36
5.2.7 Univariate Analysis of the Categorical Variable – LOAN_LOCATION 38
5.2.8 Univariate Analysis of the Categorical Variable – LOAN_APPROVAL_STATUS 40
5.2.9 Univariate Analysis of the Continuous Variable – CANDIDATE_INCOME 42
5.2.10 Univariate Analysis of the Continuous Variable – LOAN_DURATION 44
5.2.11 Univariate Analysis of the Continuous Variable – GUARANTEE_INCOME 46
5.2.12 Univariate Analysis of the Continuous Variable – LOAN_AMOUNT 48
5.2.13 Bivariate Analysis of the Variables (MARITAL_STATUS – Categorical variable versus LOAN_APPROVAL_STATUS – Categorical variable) 50
5.2.14 Bivariate Analysis of the Variables (LOAN_LOCATION – Categorical variable versus LOAN_APPROVAL_STATUS – Categorical variable) 52
5.2.15 Bivariate Analysis of the Variables (LOAN_LOCATION – Categorical variable versus CANDIDATE_INCOME – Continuous variable) 54
5.2.16 Bivariate Analysis of the Variables (LOAN_APPROVAL_STATUS – Categorical variable versus CANDIDATE_INCOME – Continuous variable) 56
5.3 MISSING VALUES IMPUTATION – LIB65778.TRAINING_DS 58

5.3.1 Missing Values Imputation in the Categorical Variable – MARITAL_STATUS 58

5.3.2 Missing Values Imputation in the Categorical Variable – FAMILY_MEMBERS....63

5.3.3 Missing Values Imputation in the Categorical Variable – GENDER....68

5.3.4 Missing Values Imputation in the Categorical Variable – EMPLOYMENT....72

5.3.5 Missing Values Imputation in the Categorical Variable – LOAN_HISTORY .77

5.3.6 Missing Values Imputation in the Numerical Variable – LOAN_AMOUNT...82

5.3.7 Missing Values Imputation in the Numerical Variable – LOAN_DURATION86

5.4 ANALYSIS OF VARIABLES – LIB65778.TRAINING_DS....90

5.4.1 Univariate Analysis of the Categorical Using SAS MACRO 90

5.4.2 Univariate Analysis of the Continuous Variable – CANDIDATE_INCOME ..93

5.4.3 Univariate Analysis of the Continuous Variable – LOAN_DURATION 95

5.4.4 Univariate Analysis of the Continuous Variable – GUARANTEE_INCOME .97

5.4.6 Bivariate Analysis of the Categorical Variables Using SAS MACRO 100

5.4.7 Bivariate Analysis of the Categorical Variable and Numerical Variable Using SAS MACRO....104

5.5 MISSING VALUES IMPUTATION – LIB65778.TESTING_DS....107

5.5.1 Missing Values Imputation in the Categorical Variable – GENDER....107

5.5.2 Missing Values Imputation in the Categorical Variable – FAMILY_MEMBERS....112

5.5.3 Missing Values Imputation in the Continuous Variable – LOAN_AMOUNT117

5.5.4 Missing Values Imputation in the Continuous Variable – LOAN_DURATION 120

SECTION 6: PREDICTION MODEL AND ODS....124

6.1 PREDICTION MODEL....124

6.2 SAS ODS – OUTPUT DELIVERY SYSTEM....128

6.2.1 Output Delivery System Output....128

SECTION 7: CONCLUSION 130

REFERENCES 131

LIST OF TABLES

Table 2.1: Data type used in credit scoring....8
Table 3.1: Data dictionary....16
Table 5.1: Table output of univariate analysis for variable: MARITAL_STATUS....26
Table 5.2: Table output of univariate analysis for variable: QUALIFICATION....28
Table 5.3: Table output of univariate analysis for variable: FAMILY_MEMBERS....30
Table 5.4: Table output of univariate analysis for variable: GENDER....32
Table 5.5: Table output of univariate analysis for variable: EMPLOYMENT....34
Table 5.6: Table output of univariate analysis for variable: LOAN_HISTORY....36
Table 5.7: Table output of univariate analysis for variable: LOAN_LOCATION....38
Table 5.8: Table output of univariate analysis for variable: LOAN_APPROVAL_STATUS 40
Table 5.9: Table output of univariate analysis for variable: CANDIDATE_INCOME....42
Table 5.10: Table output of univariate analysis for variable: LOAN_DURATION....44
Table 5.11: Table output of univariate analysis for variable: GUARANTEE_INCOME....46
Table 5.12: Table output of univariate analysis for variable: LOAN_AMOUNT....48
Table 5.13: Table output of bivariate analysis for variables: MARITAL_STATUS versus LOAN_APPROVAL_STATUS....50
Table 5.14: Table output of bivariate analysis for variables: LOAN_LOCATION versus LOAN_APPROVAL_STATUS....52
Table 5.15: Table output of bivariate analysis for variables: LOAN_LOCATION versus CANDIDATE_INCOME....54
Table 5.16: Table output of bivariate analysis for variables: LOAN_APPROVAL_STATUS versus CANDIDATE_INCOME....56
Table 5.17: Output of quantity of missing values for variable: MARITAL_STATUS....59
Table 5.18: Output of details of missing values for variable: MARITAL_STATUS....60
Table 5.19: Output of missing value after imputation for variable: MARITAL_STATUS....62
Table 5.20: Output of quantity of missing values for variable: FAMILY_MEMBERS....63
Table 5.21: Output of missing value after imputation for variable: FAMILY_MEMBERS...67
Table 5.22: Output of quantity of missing values for variable: GENDER....68
Table 5.23: Output of details of missing values for variable: GENDER....69
Table 5.24: Output of missing value after imputation for variable: GENDER....72
Table 5.25: Output of quantity of missing values for variable: EMPLOYMENT....73

Table 5.26: Output of details of missing values for variable: EMPLOYMENT 74
Table 5.27: Output of missing value after imputation for variable: EMPLOYMENT 76
Table 5.28: Output of quantity of missing values for variable: LOAN_HISTORY 78
Table 5.29: Output of details of missing values for variable: LOAN_HISTORY 78
Table 5.30: Output of missing value after imputation for variable: LOAN_HISTORY 81
Table 5.31: Output of quantity of missing values for variable: LOAN_AMOUNT 82
Table 5.32: Output of details of missing values for variable: LOAN_AMOUNT 83
Table 5.33: Output of missing value after imputation for variable: LOAN_AMOUNT 85
Table 5.34: Output of quantity of missing values for variable: LOAN_DURATION 86
Table 5.35: Output of details of missing values for variable: LOAN_DURATION 87
Table 5.36: Output of missing value after imputation for variable: LOAN_DURATION 89
Table 5.37: Table output of univariate analysis for variable: MARITAL_STATUS 91
Table 5.38: Table output of univariate analysis for variable: QUALIFICATION 91
Table 5.39: Table output of univariate analysis for variable: FAMILY_MEMBERS 91
Table 5.40: Table output of univariate analysis for variable: GENDER 92
Table 5.41: Table output of univariate analysis for variable: EMPLOYMENT 92
Table 5.42: Table output of univariate analysis for variable: LOAN_HISTORY 92
Table 5.43: Table output of univariate analysis for variable: LOAN_LOCATION 93
Table 5.44: Table output of univariate analysis for variable: CANDIDATE_INCOME 94
Table 5.45: Table output of univariate analysis for variable: LOAN_DURATION 95
Table 5.46: Table output of univariate analysis for variable: GUARANTEE_INCOME 97
Table 5.47: Table output of univariate analysis for variable: LOAN_AMOUNT 99
Table 5.48: Table output of bivariate analysis for variables: MARITAL_STATUS versus LOAN_LOCATION 101
Table 5.49: Table output of bivariate analysis for variables: GENDER versus FAMILY_MEMBERS 102
Table 5.50: Table output of bivariate analysis for variables: GENDER versus CANDIDATE_INCOME 104
Table 5.51: Table output of bivariate analysis for variables: GENDER versus LOAN_AMOUNT 105
Table 5.52: Output of quantity of missing values for variable: GENDER 108
Table 5.53: Output of details of missing values for variable: GENDER 109
Table 5.54: Output of missing value after imputation for variable: GENDER 111
Table 5.55: Output of quantity of missing values for variable: FAMILY_MEMBERS 112

Table 5.56: Output of missing value after imputation for variable: FAMILY_MEMBERS.116
Table 5.57: Output of quantity of missing values for variable: LOAN_AMOUNT 117
Table 5.58: Output of details of missing values for variable: LOAN_AMOUNT 118
Table 5.59: Output of missing value after imputation for variable: LOAN_AMOUNT.....120
Table 5.60: Output of quantity of missing values for variable: LOAN_DURATION 121
Table 5.61: Output of details of missing values for variable: LOAN_DURATION.....122
Table 5.62: Output of missing value after imputation for variable: LOAN_DURATION ...123

LIST OF FIGURES

Figure 1.1: Financing status of SME in the United States (Misera et al., 2022) 3
Figure 1.2: General process flow of applying for business loan..... 5
Figure 3.1: Snapshot of dataset from “TRAINING_DS” file..... 14
Figure 3.2: Snapshot of dataset from “TESTING_DS” file 15
Figure 4.1: SEMMA methodology workflow..... 17
Figure 5.1: Folder creation in SAS 19
Figure 5.2: Training dataset upload to SAS..... 20
Figure 5.3: Testing dataset upload to SAS..... 20
Figure 5.4: Uploaded dataset shown in project folder 21
Figure 5.5: Creation of permanent library 22
Figure 5.6: Dataset import to permanent library..... 23
Figure 5.7: Imported dataset in permanent library..... 23
Figure 5.8: SAS code for generating data dictionary..... 24
Figure 5.9: SAS generated summary information of dataset..... 24
Figure 5.10: SAS generated engine dependent information 25
Figure 5.11: SAS generated data dictionary of dataset..... 25
Figure 5.12: SAS code for univariate analysis of variable: MARITAL_STATUS..... 26
Figure 5.13: Graphical output of univariate analysis for variable: MARITAL_STATUS..... 27
Figure 5.14: SAS code for univariate analysis of variable: QUALIFICATION 28
Figure 5.15: Graphical output of univariate analysis for variable: QUALIFICATION 29
Figure 5.16: SAS code for univariate analysis of variable: FAMILY_MEMBERS 30
Figure 5.17: Graphical output of univariate analysis for variable: FAMILY_MEMBERS 31
Figure 5.18: SAS code for univariate analysis of variable: GENDER..... 32
Figure 5.19: Graphical output of univariate analysis for variable: GENDER..... 33
Figure 5.20: SAS code for univariate analysis of variable: EMPLOYMENT 34
Figure 5.21: Graphical output of univariate analysis for variable: EMPLOYMENT 35
Figure 5.22: SAS code for univariate analysis of variable: LOAN_HISTORY..... 36
Figure 5.23: Graphical output of univariate analysis for variable: LOAN_HISTORY..... 37
Figure 5.24: SAS code for univariate analysis of variable: LOAN_LOCATION..... 38
Figure 5.25: Graphical output of univariate analysis for variable: LOAN_LOCATION..... 39
Figure 5.26: SAS code for univariate analysis of variable: LOAN_APPROVAL_STATUS. 40

Figure 5.27: Graphical output of univariate analysis for variable: LOAN_APPROVAL_STATUS 41
Figure 5.28: SAS code for univariate analysis of variable: CANDIDATE_INCOME 42
Figure 5.29: Graphical output of univariate analysis for variable: CANDIDATE_INCOME 43
Figure 5.30: SAS code for univariate analysis of variable: LOAN_DURATION 44
Figure 5.31: Graphical output of univariate analysis for variable: LOAN_DURATION 45
Figure 5.32: SAS code for univariate analysis of variable: GUARANTEE_INCOME 46
Figure 5.33: Graphical output of univariate analysis for variable: GUARANTEE_INCOME 47
Figure 5.34: SAS code for univariate analysis of variable: LOAN_AMOUNT..... 48
Figure 5.35: Graphical output of univariate analysis for variable: LOAN_AMOUNT 49
Figure 5.36: SAS code for bivariate analysis of variables: MARITAL_STATUS versus LOAN_APPROVAL_STATUS 50
Figure 5.37: Graphical output of bivariate analysis for variables: MARITAL_STATUS versus LOAN_APPROVAL_STATUS 51
Figure 5.38: SAS code for bivariate analysis of variables: LOAN_LOCATION versus LOAN_APPROVAL_STATUS 52
Figure 5.39: Graphical output of bivariate analysis for variables: LOAN_LOCATION versus LOAN_APPROVAL_STATUS 53
Figure 5.40: SAS code for bivariate analysis of variables: LOAN_LOCATION versus CANDIDATE_INCOME 54
Figure 5.41: Graphical output of bivariate analysis for variables: LOAN_LOCATION versus CANDIDATE_INCOME 55
Figure 5.42: SAS code for bivariate analysis of variables: LOAN_APPROVAL_STATUS versus CANDIDATE_INCOME 56
Figure 5.43: Graphical output of bivariate analysis for variables: LOAN_APPROVAL_STATUS versus CANDIDATE_INCOME 57
Figure 5.44: SAS code for making a dataset backup prior to imputation for variable: MARITAL_STATUS 58
Figure 5.45: Backup dataset creation output..... 58
Figure 5.46: SAS code for identifying quantity of missing values for variable: MARITAL_STATUS 59
Figure 5.47: SAS code for identifying details of missing values for variable: MARITAL_STATUS 59

Figure 5.48: SAS code for temporary dataset creation for variable: MARITAL_STATUS ... 60
Figure 5.49: Output of temporary dataset for variable: MARITAL_STATUS ... 60
Figure 5.50: SAS code for identifying the mode and perform missing value imputation for variable: MARITAL_STATUS ... 61
Figure 5.51: Output from imputation of variable: MARITAL_STATUS ... 61
Figure 5.52: SAS code to identify missing value after imputation for variable: MARITAL_STATUS ... 62
Figure 5.53: SAS code for making a dataset backup prior to imputation for variable: FAMILY_MEMBERS ... 63
Figure 5.54: SAS code for identifying quantity of missing values for variable: FAMILY_MEMBERS ... 63
Figure 5.55: SAS code for data manipulation for variable: FAMILY_MEMBERS ... 64
Figure 5.56: Output from data manipulation of variable: FAMILY_MEMBERS ... 64
Figure 5.57: SAS code for temporary dataset creation for variable: FAMILY_MEMBERS ... 65
Figure 5.58: Output of temporary dataset for variable: FAMILY_MEMBERS ... 65
Figure 5.59: SAS code for identifying the mode and perform missing value imputation for variable: FAMILY_MEMBERS ... 66
Figure 5.60: Output from imputation of variable: FAMILY_MEMBERS ... 66
Figure 5.61: SAS code to identify missing value after imputation for variable: FAMILY_MEMBERS ... 67
Figure 5.62: SAS code for making a dataset backup prior to imputation for variable: GENDER ... 68
Figure 5.63: SAS code for identifying quantity of missing values for variable: GENDER ... 68
Figure 5.64: SAS code for identifying details of missing values for variable: GENDER ... 69
Figure 5.65: SAS code for temporary dataset creation for variable: GENDER ... 70
Figure 5.66: Output of temporary dataset for variable: GENDER ... 70
Figure 5.67: SAS code for identifying the mode and perform missing value imputation for variable: GENDER ... 70
Figure 5.68: Output from imputation of variable: GENDER ... 71
Figure 5.69: SAS code to identify missing value after imputation for variable: GENDER ... 71
Figure 5.70: SAS code for making a dataset backup prior to imputation for variable: EMPLOYMENT ... 72
Figure 5.71: SAS code for identifying quantity of missing values for variable: EMPLOYMENT ... 73

Figure 5.72: SAS code for identifying details of missing values for variable: EMPLOYMENT 73
Figure 5.73: SAS code for temporary dataset creation for variable: EMPLOYMENT 74
Figure 5.74: Output of temporary dataset for variable: EMPLOYMENT 75
Figure 5.75: SAS code for identifying the mode and perform missing value imputation for variable: EMPLOYMENT 75
Figure 5.76: Output from imputation of variable: EMPLOYMENT 75
Figure 5.77: SAS code to identify missing value after imputation for variable: EMPLOYMENT 76
Figure 5.78: SAS code for making a dataset backup prior to imputation for variable: LOAN_HISTORY 77
Figure 5.79: SAS code for identifying quantity of missing values for variable: LOAN_HISTORY 77
Figure 5.80: SAS code for identifying details of missing values for variable: LOAN_HISTORY 78
Figure 5.81: SAS code for temporary dataset creation for variable: LOAN_HISTORY 79
Figure 5.82: Output of temporary dataset for variable: LOAN_HISTORY 79
Figure 5.83: SAS code for identifying the mode and perform missing value imputation for variable: LOAN_HISTORY 80
Figure 5.84: Output from imputation of variable: LOAN_HISTORY 80
Figure 5.85: SAS code to identify missing value after imputation for variable: LOAN_HISTORY 81
Figure 5.86: SAS code for making a dataset backup prior to imputation for variable: LOAN_AMOUNT 82
Figure 5.87: SAS code for identifying quantity of missing values for variable: LOAN_AMOUNT 82
Figure 5.88: SAS code for identifying details of missing values for variable: LOAN_AMOUNT 83
Figure 5.89: SAS code for performing missing value imputation for variable: LOAN_AMOUNT 84
Figure 5.90: Output from imputation of variable: LOAN_AMOUNT 84
Figure 5.91: SAS code to identify missing value after imputation for variable: LOAN_AMOUNT 85

Figure 5.92: SAS code for making a dataset backup prior to imputation for variable: LOAN_DURATION....86
Figure 5.93: SAS code for identifying quantity of missing values for variable: LOAN_DURATION....86
Figure 5.94: SAS code for identifying details of missing values for variable: LOAN_DURATION....87
Figure 5.95: SAS code for performing missing value imputation for variable: LOAN_DURATION....88
Figure 5.96: Output from imputation of variable: LOAN_DURATION....88
Figure 5.97: SAS code to identify missing value after imputation for variable: LOAN_DURATION....89
Figure 5.98: SAS code to program a macro for univariate analysis of categorical variables..90
Figure 5.99: SAS code for calling the macro of univariate analysis of categorical variables.90
Figure 5.100: SAS code for univariate analysis of variable: CANDIDATE_INCOME....94
Figure 5.101: Graphical output of univariate analysis for variable: CANDIDATE_INCOME....94
Figure 5.102: SAS code for univariate analysis of variable: LOAN_DURATION....95
Figure 5.103: Graphical output of univariate analysis for variable: LOAN_DURATION....96
Figure 5.104: SAS code for univariate analysis of variable: GUARANTEE_INCOME....97
Figure 5.105: Graphical output of univariate analysis for variable: GUARANTEE_INCOME....98
Figure 5.106: SAS code for univariate analysis of variable: LOAN_AMOUNT....99
Figure 5.107: Graphical output of univariate analysis for variable: LOAN_AMOUNT....99
Figure 5.108: SAS code to program a macro for bivariate analysis of categorical variables 100
Figure 5.109: SAS code for calling the macro of bivariate analysis of categorical variables100
Figure 5.110: Graphical output of bivariate analysis for variables: MARITAL_STATUS versus LOAN_LOCATION....101
Figure 5.111: Graphical output of bivariate analysis for variables: GENDER versus FAMILY_MEMBERS....103
Figure 5.112: SAS code to program a macro for bivariate analysis of categorical variable versus numerical variable....104
Figure 5.113: SAS code for calling the macro of bivariate analysis of categorical variable versus numerical variable....104

Figure 5.114: Graphical output of bivariate analysis for variables: GENDER versus CANDIDATE_INCOME 105
Figure 5.115: Graphical output of bivariate analysis for variables: GENDER versus LOAN_AMOUNT 106
Figure 5.116: SAS code for making a dataset backup prior to imputation for variable: GENDER 107
Figure 5.117: Backup dataset creation output 107
Figure 5.118: SAS code for identifying quantity of missing values for variable: GENDER 108
Figure 5.119: SAS code for identifying details of missing values for variable: GENDER... 108
Figure 5.120: SAS code for temporary dataset creation for variable: GENDER 109
Figure 5.121: Output of temporary dataset for variable: GENDER 109
Figure 5.122: SAS code for identifying the mode and perform missing value imputation for variable: GENDER 110
Figure 5.123: Output from imputation of variable: GENDER 110
Figure 5.124: SAS code to identify missing value after imputation for variable: GENDER 111
Figure 5.125: SAS code for making a dataset backup prior to imputation for variable: FAMILY_MEMBERS 112
Figure 5.126: SAS code for identifying quantity of missing values for variable: FAMILY_MEMBERS 112
Figure 5.127: SAS code for data manipulation for variable: FAMILY_MEMBERS 113
Figure 5.128: Output from data manipulation of variable: FAMILY_MEMBERS 113
Figure 5.129: SAS code for temporary dataset creation for variable: FAMILY_MEMBERS 114
Figure 5.130: Output of temporary dataset for variable: FAMILY_MEMBERS 114
Figure 5.131: SAS code for identifying the mode and perform missing value imputation for variable: FAMILY_MEMBERS 115
Figure 5.132: Output from imputation of variable: FAMILY_MEMBERS 115
Figure 5.133: SAS code to identify missing value after imputation for variable: FAMILY_MEMBERS 116
Figure 5.134: SAS code for making a dataset backup prior to imputation for variable: LOAN_AMOUNT 117
Figure 5.135: SAS code for identifying quantity of missing values for variable: LOAN_AMOUNT 117

Figure 5.136: SAS code for identifying details of missing values for variable: LOAN_AMOUNT 118
Figure 5.137: SAS code for performing missing value imputation for variable: LOAN_AMOUNT 118
Figure 5.138: Output from imputation of variable: LOAN_AMOUNT 119
Figure 5.139: SAS code to identify missing value after imputation for variable: LOAN_AMOUNT 119
Figure 5.140: SAS code for making a dataset backup prior to imputation for variable: LOAN_DURATION 120
Figure 5.141: SAS code for identifying quantity of missing values for variable: LOAN_DURATION 121
Figure 5.142: SAS code for identifying details of missing values for variable: LOAN_DURATION 121
Figure 5.143: SAS code for performing missing value imputation for variable: LOAN_DURATION 122
Figure 5.144: Output from imputation of variable: LOAN_DURATION 122
Figure 5.145: SAS code to identify missing value after imputation for variable: LOAN_DURATION 123

LIST OF ABBREVIATIONS

AIC.....Akaike's Information Criterion
CNN.....Convolutional Neural Network
DT .....Decision Tree
GBDT.....Gradient Boosting Decision Tree
LFI.....Lasiandra Finance Inc.
ODS.....Output Delivery System
SAS .....Statistical Analysis System
SC.....Schwarz Criterion
SME .....Small and Medium enterprises
SMOTE.....Synthetic Minority Oversampling Technique

SECTION 1

INTRODUCTION

1.1 INTRODUCTION

Small and Medium enterprises (SME) made up over 99% of all businesses in the United States and employed an approximation of 60.6 million workers (Downing, 2021). Therefore, SME plays a major role in the economy by providing employments, generating tax revenues, and driving innovations. However, majority of SME failed to survive in the first five years of operation. This is due to the difficulty of SME in attracting and raising sufficient capital to fund their operations and growth (Liberto, 2021). Hence, SME relies heavily on the accessibility to loan facilities to finance their business expenses.

Loan facilities are widely available and offered by many financing institutions. However, the SME are facing challenges in acquiring the fundings due to several reasons. SME are typically newly established businesses which lack the credit history thus causing lenders to reject or charge a higher interest rate on the loan. In addition, SME lacked the collateral to offer as a guarantee for the loan. Furthermore, the unsatisfactory business plans and low level of stability and liquidity in SME further reduced the confidence of lenders in providing loans to SME.

A private financing company from New York, Lasiandra Finance Inc. (LFI) understood the challenges faced and the significance of fundings required by SME. Therefore, LFI provides personalized loan facilities to cater the needs of SME. However, the current loan approval process is complex and time-consuming where it is a labor-intensive task to perform verification and validation on customer information just to gather sufficient justifications for the approval of the loan application. In addition, the loan approval process handled by humans are prone to error and misjudgment. Therefore, loans provided to inadequate applicants will lead to increasing credit risk for LFI. While, missing out on providing loans to adequate applicants, causes LFI to lose customers and profits.

LFI has acknowledged the deficiency in their loan approval process and seeking to improve it by automating the process using predictive analytics. LFI has hired and assigned the task to a data scientist to design and develop the automation program with a highly accurate prediction model to predict the loan approval with outcomes as approved or rejected for the loan

applications. Therefore, the main objective of LFI is to improve the speed and accuracy in providing their loan services to SME. This in turn, improves the customer retention rate and customer experience.

This report documents the process of developing a loan approval prediction model using logistic regression. In addition, the program will be developed using Statistical Analysis System (SAS) programming, which is an integrated system of software for advanced analysis, data management, and predictive analytics.

1.2 BACKGROUND

SME defined by Small Business Administration in the United States, are businesses having a certain threshold on number of employees based on different industries. Example, for companies to be considered as SME in the manufacturing sector, the number of employees is a maximum of 500, while for the wholesale trading companies are only allowed up to 100 employees (Corporate Finance Institute, 2022). In general, SME in the United States can be considered as businesses with less than 500 employees.

SME require access to fundings throughout their business life cycle to facilitate start up, development, and growth. Which ultimately produces employment and economic growth. Therefore, SME typically turn to the banks to acquire loans to cater for the financial needs of their businesses. However, it is not a straightforward process for SME to obtain such loan facilities. The failure for SME to access loan facilities would severely hinder the growth opportunities. The following list some of the difficulties in securing loans from the perspective of borrowers, and providing loans from the perspective of lenders. (Cusmano et al., 2018):

From the perspective of lenders:

- Lenders are becoming more risk averse.
- Lenders lacking the expertise to understand the intricacies of diverse sectors of SME, causing lenders unable to gauge the risk involved.
- Lenders having difficulty in forecasting future return of SME with limited documentation provided from SME.
- Legacy system using paper-based process causes loan approval to be time-consuming and labor-intensive.
- Cost of serving SME outweighed potential gain.

From the perspective of borrowers:

- SME to bear higher transaction costs from lenders.
- SME having insufficient collateral as a guarantee to secure the loan.
- SME having incomplete and insufficient credit history.
- SME having weak cash flow and liquidity.
- SME lacking the skills and knowledge to produce complete financial statements.
- SME abandoning the application process halfway due to the difficult and lengthy process to satisfy the requirements from lenders.

With such obstacles to go through by the SME to obtain loans from banks, SME faces a higher rejection rate as compared to large corporations. Therefore, a high number of SME did not manage to secure business loans from traditional banks. SME who got their loans rejected, are forced to rely on internal fundings or to seek short-term cash from friends and families to launch, develop, and continue their businesses. In addition, a study by Cowling et al. (2021), found that 72% of SME who got their loans rejected are scarred from the incident, resulting in reluctant to try again to apply for business loans from the banks. Instead, would seek alternative source of fundings as their expectation that the banks would reject them once again.

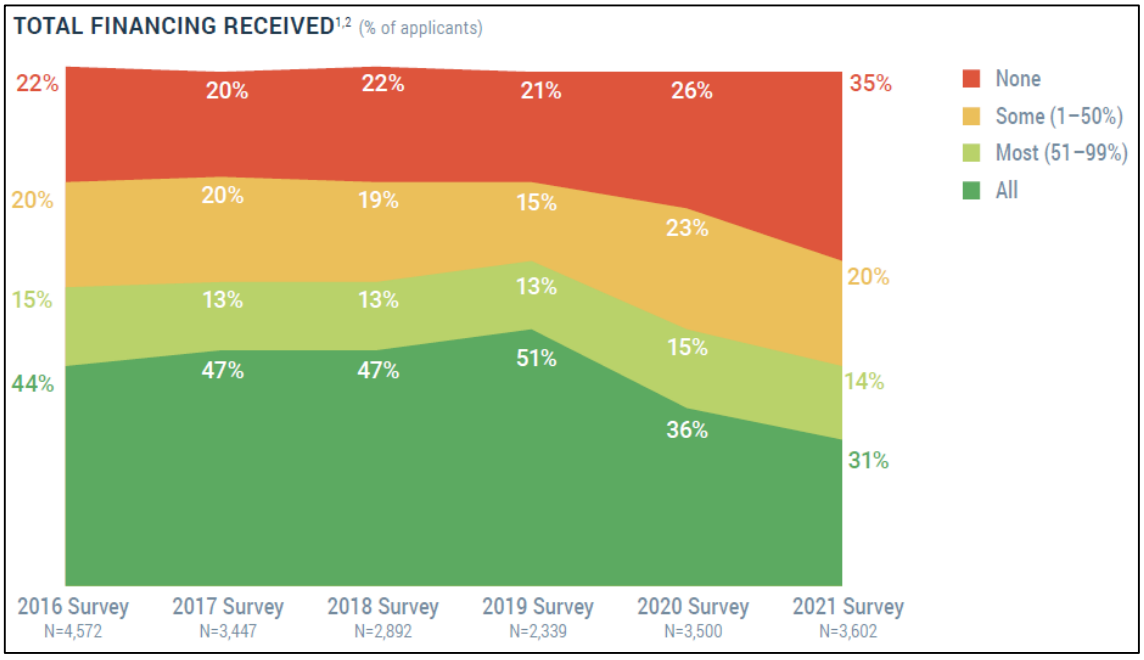


Figure 1.1: Financing status of SME in the United States (Misera et al., 2022)

According to Small Business Credit Survey in the United States, 35% of SME applicants got their loans rejected in year 2021 (Misera et al., 2022) which is an increased from previous years. Figure 1.1 shows the survey of financing outcomes for SME loan applications in the United States from year 2016 to year 2021. It can be observed that from year 2016 to year 2019, the percentage of fully and partially approved loans and fully rejected loans remain relatively similar. However, a decrease in fully and partially approved loans and an increase in fully rejected loans started to increase after the year 2019. This may be due to the COVID-19 pandemic affecting many businesses. In addition, banks may become more risk averse as the economic downturn would likely cause businesses to collapse and unsustainable. Thus, reducing the likelihood of approving loans to businesses without a strong foundation.

Therefore, there exist a large number of SME not benefitting the loan facilities from banks. Which is a significant portion of the market since SME contributes a major share of the economy. This untapped business opportunity is yet to be exploited by the traditional banks. With the rise of fintech, it has provided SME an alternative source of fundings which are more attractive as compared to applying financing from traditional banks. Some benefits of fintech are quick and easy financing procedures, personalized product offerings, and reduced transaction costs. Thus, with the increased competition from newer technologies offering more attractive packages, traditional banks would have to improve and innovate on digital transformation to be on par with the competitions.

Prior to the adoption of fintech and digitalization, the loan approval process would typically take anywhere from one to three months for processing the loan applications by the banks. In addition, the duration mentioned does not consider the time required for the SME to prepare the requested documents in a fully complete and accurate form. Thus, the whole process may take longer than three months to complete. Figure 1.2 shows the general process flow of applying for a business loan, which involve many stages.

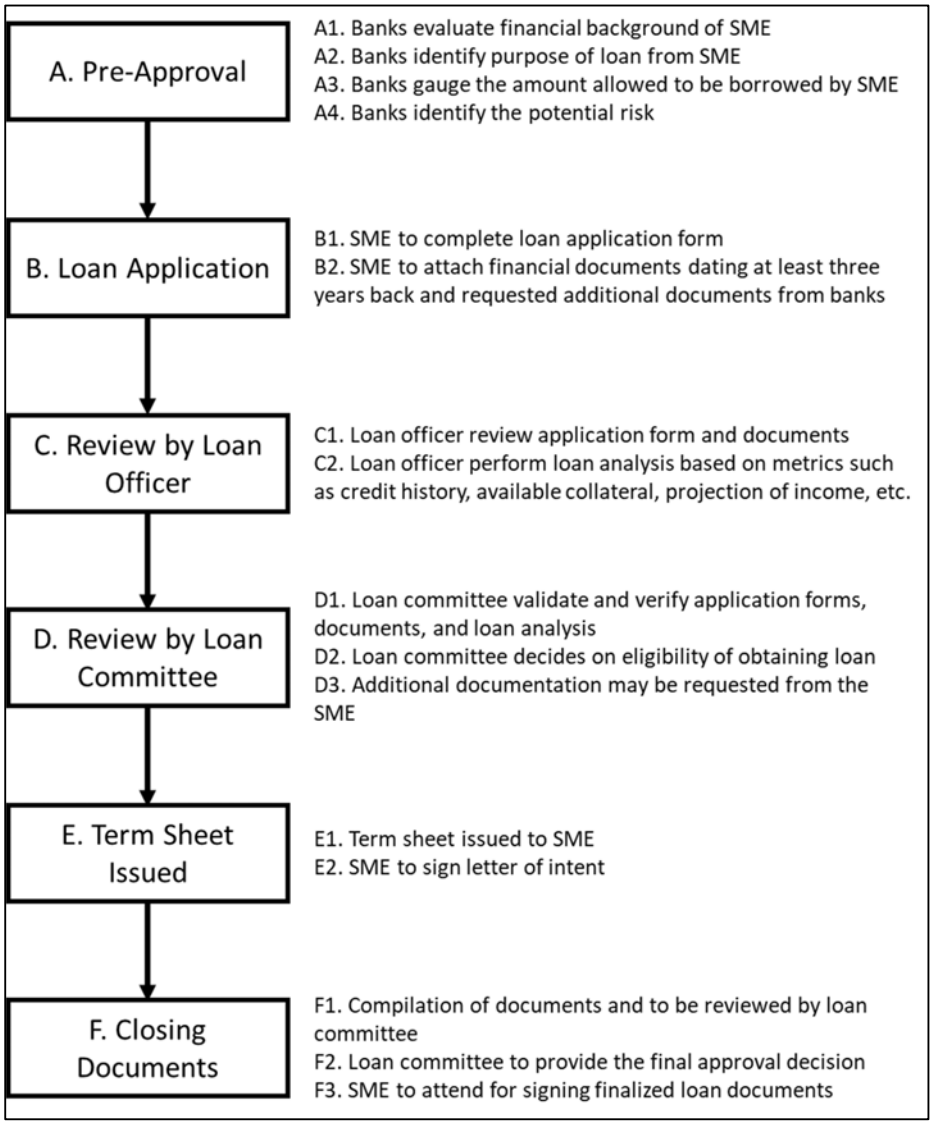


Figure 1.2: General process flow of applying for business loan

The stages involved in approving a business loan can be very tedious and would often involve repetitive verification and evaluation of information and documents. This is to ensure the risk is fully understood by the bank before approving loans to any businesses to protect the bank from credit risks. Consequences of credit risks would include opportunity costs, accounting loss, and transaction costs. In addition, the reputation of the bank would be tarnished and possibly resulting in liquidity risk (Win, 2018). However, some credit risks are unavoidable, as accurately forecasting the future is a difficult task to perform. Therefore, banks often perform multiple stages of verification to minimize the risks to an acceptable level.

LFI is a licensed financial institution providing credit opportunities to SME, which offers short-term cash and long-term debt to the funding needs of SME. Business model of LFI is based on charging interests on the provided loans and services to eligible SME. With the increased competition, LFI is aware of the importance to innovate and digitalized the loan approval process and providing personalized services to capture the untapped market opportunities. Thus, LFI is adopting digitalization using machine learning to offer personalized products and to streamline the loan approval process.

With the increasing number of applications for business loans, LFI would require machine learning to assist in processing the high volume of data to provide a quick and accurate evaluation on the loan applications. The utilization of machine learning in loan approval prediction has been widely researched and applied in the industry. Machine learning models can capture hidden patterns and trends in the data to provide an accurate loan approval prediction at the fraction of time as compared to the legacy approach. In addition, the automation of loan approval process, would significantly reduce the administrative overhead and delays in procedures.

An example case study of machine learning in credit assessment (Parungao, 2020). Amazon using machine learning was able to utilize external data which are not from the banking sector to perform credit assessment on borrowers. Which many borrowers who do not have credit history were able to benefit from the loan facility offered by Amazon. This has allowed Amazon to identify potential borrowers which were once not able to pass the pre-assessment from traditional banks. As a result, Amazon were able to profit from the untapped market opportunities.

1.3 AIM & OBJECTIVES

1.3.1 Aim

To develop a program to automate the loan eligibility process for LFI using predictive analytics with increased accuracy and accelerating the process for loan approvals.

1.3.2 Objectives

The objectives of this study are as followed:

1. To perform descriptive analytics to identify trends and data quality issues present in the dataset.
2. To develop predictive model using the logistic regression algorithm for predicting the loan eligibility of applicants.
3. To develop the program that automates the descriptive and predictive analytics using SAS programming language.

1.4 ASSUMPTION AND JUSTIFICATION

Available resources to be utilized by the data scientist, will be confined to what is provided by LFI. LFI has adopted SAS analytics solutions as part of their transformation movement to provide improved and personalized services to their customers. Therefore, the data scientist will be developing the automation program using only SAS Studio.

SAS is a web-based application for data analysis, complete with a wide range of statistical procedures. In addition, SAS provides an interactive graphical user interface for performing analytics which is very useful to non-programmers. Furthermore, features of SAS include data management, artificial intelligence, and graphical presentation of outputs. Moreover, SAS syntax is easy to learn and provides a comprehensive library for various functions, thus can be easily adapted by different levels of users.

However, licensing for SAS can be very costly as it is offered as a complete software without the option select and pay for specific functions as needed. Therefore, utilization rate of SAS functions will remain low for novice users. In addition, SAS is a closed environment software with no support for open source thus implementation of newer machine learning algorithms into SAS may be delayed. Moreover, SAS lacks the graphical capabilities in customizing plots which require advance understanding of SAS packages to perform plots customization.

SECTION 2

LITERATURE REVIEW

The following section covers the literature survey on the existing system used for evaluating banking loans approval and the modern approach of utilizing machine learning in predicting the loans approval.

2.1 LEGACY SYSTEM IN LOAN APPROVAL

The credit scoring system has been widely used by financial institution to gauge the risk of providing loans to borrowers. It is a quick, consistent, and effective method on deciding the eligibility of loan to an applicant. Therefore, loan applicants are highly dependent on having a good historical credit to be granted to loan facilities. Generally, a higher credit score signifies a higher creditworthiness of an applicant which means a higher future repayment rate is expected from the applicant.

According to the credit scoring guidelines by World Bank, three data types are typically used by financial institution to formulate the credit score (Knutson & L, 2020). Table 2.1 shows the data in each category typically used in credit score assessment.

Table 2.1: Data type used in credit scoring

Data Type	Description
Bank transactional data	Historical payment records, current loan status and amount, credit history, type of loan, loan purpose, loan maturity term, types of credit, length of credit history
Credit bureau checks	Number of loans applied, bankruptcy records
Commercial data	Financial statements, collateral value, number of current loans, payable tax amount

Prior to adoption of machine learning, the credit risk assessment were hand-calculated by professionals and are time-consuming and prone to human error (Xiao & Jiao, 2021). The commonly used techniques to formulate the credit score are statistical discriminant and classification methods. Which includes linear regression, discriminant analysis, logistic regression, judgement-based models.

The use of traditional approaches in assessing the credit risk has provided some difficulties for the banks. This is due to the increasing volume of applicants and the inability to quickly and accurately assess the applications, leading to the increase of loan defaulters and impacting the revenue generation and customer experience (Shoumo et al., 2019). Utilizing the traditional approach is often unable to obtain a complete information about the applicant, as the applicants would not disclose fully the information to the banks (Win, 2018).

Loan defaults significantly impact the financial institution as it consumes excess resources and imposes additional expenses to recover the defaulted loans. Therefore, banks would often avoid such events by critically assessing the credit risk during the application stage. However, it is impossible to eliminate bad debts. Thus, banks would often deploy loan recovery methods to retrieve back the defaulted loans.

The banks can attempt to deploy an in-house or a third-party collection agency to assist in the recovery efforts. In addition, legal action can be taken, where formal demand letters can be issued to defaulters to request payment. However, lenders resulting in loan default typically are experiencing downturn in businesses which the bank can negotiate with the lenders to restructure the loan payment.

2.2 MACHINE LEARNING IN LOAN APPROVAL

The legacy system of loan assessment has been gradually replaced by the use of machine learning methods due to the benefits of improved efficiency and cost effectiveness (Xiao & Jiao, 2021). Various predictive algorithms have been applied to develop loan prediction models to effectively assess credit risks from the SME to reduce the potential of bad debts (Jiang, 2021). In addition, the utilization of machine learning has allowed the use of alternative data for credit scoring models. For example, Yu et al. (2020) has utilized social media data to develop a credit scoring model to assess credit risk and successfully predicted the approval rate with high accuracy. Therefore, the use of machine learning will open more opportunities to the untapped audiences seeking for financing with low or no credit history.

2.2.1 Machine Learning Application

Lusinga et al. (2021) has mentioned that the required loan approval data typically in the developing countries are often incomplete as compared to developed countries. Thus, investigated the application of machine learning technique using data from developing

countries which often consist of high amount of missing data. Several algorithms were applied namely extreme gradient boosting, logistic regression, random forest, support vector machine, and artificial neural network. It was identified that tree-based algorithm, specifically the extreme gradient boosting performed the best with 73.2% accuracy. However, similar machine learning algorithms were used in the works of Shoumo et al. (2019) which the author investigated the application of different dimensionality reduction techniques affecting the loan approval results. Data from a lending institution from the United States were used. It was identified that support vector machines with an accuracy of 94.5%, outperform other tree-based and regression models. This may indicate the difference in dataset quality from the developed and developing countries as proposed by Lusinga et al. (2021). In addition, indicating the effectiveness of applying dimensionality reduction significantly benefitted the prediction outcome.

Wang et al. (2018) utilized the concept of combining random subspace and negative correlation with Artificial Neural Network (ANN) to increase the diversity between base classifiers. The model was able to achieve an amazing accuracy of 98.09%. However, mentioned by the author that the model does not provide interpretability due to the use of neural network algorithm and would be less practical to be used in the industry. Another type of neural network applied, where Dastile and Celik (2021) utilized the Convolutional Neural Network (CNN) technique in assessing the credit score for a loan approval model. The author utilized the method of converting tabular data into images and processed using CNN to obtain weights of evidence. The prediction model was able to achieve an accuracy of 83%.

Haoran and Boyang (2020) investigated the application of gradient boosting decision trees (GBDT) in predicting loan approvals. It was found that GBDT algorithm was robust to imbalanced dataset and noisy data. The model was able to achieve a high prediction accuracy of 87.31%. In addition, the author found that historical repayment record has a significant impact on the outcome of the prediction model. In the works of Fan (2021), a basic Decision Tree (DT) algorithm was used to develop the prediction model and achieved an ideal accuracy of 78.12%. In addition, the author highlighted the benefits of using decision trees where it displays the output in a tree-like structure, which can be transformed into if-then statements. This provides the interpretability of results from the predictive model. Comparing the works of both authors, the GBDT achieved a better prediction accuracy than DT. The better result can

be due to GBDT comprising of an ensemble of DT, as compared to using only a single DT for developing the model.

2.2.2 Challenges in Machine Learning Application

Result Interpretability

A problem with using machine learning techniques to predict a loan approval is the lack in interpretability of the reasoning behind the approval and rejection made by the prediction model (Dastile & Celik, 2021; Knutson & L, 2020; Xiao & Jiao, 2021). Explanation is essential especially when a loan application is rejected, as applicants would often seek the rejection reasoning. In addition, ability to interpret the results would increase confidence in the application of machine learning in financial institution. This is to ensure no discrimination or biases occurred during the evaluation by the prediction models.

Imbalanced data classes

Dataset of loan approvals are typically imbalanced. Where there are more data with approved outcomes as compared to rejected outcomes (Xiao & Jiao, 2021). The consequence of using such dataset is the development of a biased prediction model. Therefore, pre-processing is often required to balance the dataset to achieve a more accurate model. Lusinga et al. (2021) has used random sampling and Synthetic Minority Oversampling Technique (SMOTE) to overcome the imbalanced dataset problem. In addition, mentioned by the author that the combination usage of both random sampling and SMOTE yielded a better result as compared to using each data balancing technique in isolation.

Feature Reduction

Dataset obtained from the loan application process often involve a very high amount of variables (Lusinga et al., 2021). Which the number of variables would make the prediction model much more complex and consumes higher processing resources to compute an accurate result. Feature selection is often performed to reduce the number of variables. Which Lusinga et al. (2021) used Random Forest Boruta to eliminate insignificant features and provided importance values to the high impact variables. This has simplified and reduced the number of features for the model to learn thus achieving a higher efficiency model. Similarly, Shoumo et al. (2019) performed feature reduction by adopting Recursive Feature Elimination with Cross-Validation and Principal component analysis.

In the works of Haoran and Boyang (2020) and Fan (2021), features were provided a weightage using the entropy method which identify the impact of the features onto the prediction model. Features with a low entropy score were not included into the model fitting.

Feature Transformation

Dataset related to the credit risk assessment task often consist of a combination of numerical and categorical variables. Feature transformation, which is the application of mathematical formulae to the feature values are often performed to transform the data into a more usable format facilitating the prediction algorithms.

Feature encoding, which is the transformation of variable type from character into numeric, typically performed for categorical variables. It is performed due to majority of machine learning algorithms can only work with numerical data. As seen in the works of Lusinga et al. (2021) and Shoumo et al. (2019), the authors performed feature encoding using the one-hot encoding technique, which generates additional variables based on the distinct labels from the categorical variables.

Feature scaling, which is the standardization of numeric data within a confined range. It is performed typically on dataset with variables of highly varying magnitude of data values. Benefits of feature scaling include quicker model convergence and reduce biases on variables with higher magnitude of data values. A comparison in feature scaling technique between Z-score standardization and Min-Max normalization by Shoumo et al. (2019), mentioned that support vector machine, logistic regression, and neural network algorithms performed better when the data is scaled using the Z-score standardization technique. However, in the works Wang et al. (2018) who utilized CNN, mentioned that Min-Max normalization is important for neural networks due to the impact of varied value ranges have on the activation functions.

Missing value

Missing values are typically found in the dataset of loan applications. This can be due to a mistake in data entry or by purpose of applicants in leaving the field empty to conceal information. However, dataset of loan approvals are typically huge in volume thus observations with missing data can be removed as the amount of missing data are typically minimal as compared to the entire dataset (Shoumo et al., 2019). Imputation of missing values can be

performed which is seen in the works of Lusinga et al. (2021) and Wang et al. (2018) which imputed missing values with the mean.

Model Overfitting

Researchers who utilized neural networks as the prediction algorithm typically faced with an issue of overfitting of the model due to the strong fitting ability of neural networks (Knutson & L, 2020). Wang et al. (2018) faces similar difficulty and applied regularization penalty to reduce the model overfitting problem. An overfitting model would result in inaccurate loan approval prediction which can lead to losses or missed opportunity faced by the bank.

SECTION 3

DATASET

3.1 INTRODUCTION

This section documents the dataset provided by LFI and description of the data dictionary. The dataset was provided in two comma separated values file named "TESTING_DS" and "TRAINING_DS". The dataset contains the customer portfolio derived from the filling of the online application form.

The "TRAINING_DS" dataset will be used for model fitting, while the "TESTING_DS" dataset will be used for model fitness evaluation. The "TRAINING_DS" dataset contains 614 observations and 13 features, while the "TESTING_DS" dataset contains 367 observations and 13 features. However, the "LOAN_APPROVAL_STATUS" feature in the "TESTING_DS" dataset will contain no values. It is to allow the prediction models to predict the loan approval outcome for evaluation of the model performance. Shown in Figure 3.1 and Figure 3.2, the first five observations from the "TRAINING_DS" and "TESTING_DS" dataset respectively.

Obs	SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME
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

Obs	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
1	0	.	360	1	City	Y
2	1508	128	360	1	Village	N
3	0	66	360	1	City	Y
4	2358	120	360	1	City	Y
5	0	141	360	1	City	Y

Figure 3.1: Snapshot of dataset from "TRAINING_DS" file

Obs	SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME
1	LP001015	Male	Married	0	Graduate	No	5720
2	LP001022	Male	Married	1	Graduate	No	3076
3	LP001031	Male	Married	2	Graduate	No	5000
4	LP001035	Male	Married	2	Graduate	No	2340
5	LP001051	Male	Not Married	0	Under Graduate	No	3276

Obs	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
1	0	110	360	1	City	
2	1500	126	360	1	City	
3	1800	208	360	1	City	
4	2546	100	360	.	City	
5	0	78	360	1	City	

Figure 3.2: Snapshot of dataset from “TESTING_DS” file

Upon initial inspection on the first five observations from the dataset, several observations are identified:

- The “SME_LOAN_ID_NO” can be dropped from the analysis as it does not provide meaningful information to the prediction task.
- A mix of numerical and categorical features are present in the dataset. Some categorical features are in numeric format, while some are in character format.
- The “LOAN_APPROVAL_STATUS” is the dependent variable. Thus, it is identified that the dataset contains 11 independent variables and one dependent variable after excluding the dropped feature.

3.2 DATA DICTONARY

Table 3.1 shows the data dictionary for the dataset, outlining the feature name, feature description, data type, data length, and data samples.

Table 3.1: Data dictionary

Feature Name	Description	Data Type	Length	Data Sample
SME_LOAN_ID_NO	Loan reference number	Character	8	LP001015, LP001008, LP001097, …
GENDER	Gender of applicant	Character	6	Male, Female
MARITAL_STATUS	Marital status of applicant	Character	11	Married, Not Married
FAMILY_MEMBERS	Total number of family members	Character	2	0, 1, 2, 3+
QUALIFICATION	Education level of applicant	Character	14	Graduate, Under Graduate
EMPLOYMENT	Employment status of applicant	Character	3	Yes, No
CANDIDATE_INCOME	Monthly income of the applicant	Numeric	8	0, 570, 72529, …
GUARANTEE_INCOME	Monthly income of the co-applicant	Numeric	8	0, 4288, 24000, …
LOAN_AMOUNT	Loan amount applied in thousands	Numeric	8	28, 90,148, …
LOAN_DURATION	Loan tenure in months	Numeric	8	6, 360, 480, …
LOAN_HISTORY	Historical loan records – where 0 indicates having missed repayment, and 1 indicates no missed repayment	Numeric	8	0, 1
LOAN_LOCATION	Location of applicant	Character	7	Town, City, Village
LOAN_APPROVAL_STATUS	Approval status of the loan	Character	1	Y, N

SECTION 4

METHODOLOGY

4.1 INTRODUCTION

This section documents the experimental methodology for developing the loan eligibility prediction model. The methodology consists of a sequence of steps which guides the data mining process. The methodology consists of four stages, namely **S**ample, **E**xplore, **M**odify, and **M**odel. The following section would describe the stages and specific methods applied in this study.

4.2 METHODOLOGY

This section outlines the methods to be applied in each stage of the SEMM methodology. Figure 4.1 shows the workflow of SEMM methodology applied in this study.

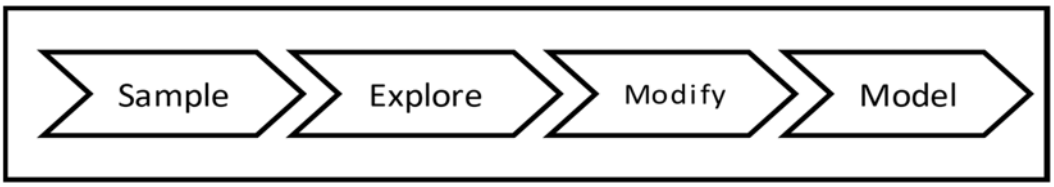


Figure 4.1: SEMM methodology workflow

Stage 1: Sample

This stage consists of partitioning and initial setting to the dataset. Typical data science project would require the partitioning of the dataset into training and testing sets. The training set would be used for model fitting while the testing set will be used by the fitted model to predict the outcomes. Dataset provided by LFI has been partitioned where the "TRAINING_DS" dataset will be used as the training set and the "TESTING_DS" dataset will be the dataset that requires prediction of the loan approval outcome. The dataset would be imported to SAS following the initial settings and verification of data type will be performed.

Stage 2: Explore

This stage consists of data exploration for data understanding, trends and patterns discovery, and anomalies detection. The following list the required processes to be performed as part of the data exploration stage:

- Identifying missing values. Missing values in SAS by default are represented by a single period for numeric features and a blank space for character features.
- Identifying outliers. For numerical features, outliers are data points that significantly differs from the average value of other data points and can be identified using box plots.
- Performing graphical analysis to understand relationships between features. It can be performed using charts such as box plot, histogram, and bar chart.
- Computing summary statistics. Complementing to the graphical analysis, summary statistics provide summaries of the data using central tendency and dispersion measures to provide a deeper understanding to the relationship between features.

Stage 3: Modify

This stage consists of data pre-processing steps to prepare the data in a format usable by the predictive algorithm. The data pre-processing would include missing values imputation. Imputing missing values allow the retaining of data instead of dropping the observations with missing values which may contain useful information. Since the dataset only contains a small number of missing values, where each feature contains less than 5% of missing values. Imputation techniques such as replacing with the mean value for numerical features and replacing with the mode label for categorical features can be utilized for imputing missing values.

Stage 4: Model

This stage consists of development of the predictive model. A prediction model using logistic regression algorithm will be developed. The prediction task is a classification problem with a binary outcome to predict whether the applicant has their loan approved or rejected. The logistic regression algorithm is chosen due to the suitability in classification task and provides high interpretability to understand the how the decisions are justified by the model. The model returns the prediction in a probability value which allows to judge the confidence of the model in the prediction output.

SECTION 5

EXPERIMENTATION

5.1 INITIAL SETUP

This section documents the initial setting up of SAS Studio in preparation for the model development. The processes would include the creation of SAS folder, creation of permanent library, uploading and importing of datasets, and displaying the data dictionary of the dataset.

5.1.1 Folder Creation on SAS

The SAS folders facilitate the storing and organizing of files, which for this study the main components stored under the folder would be the programs containing the prediction model and datasets provided by LFI. A project folder named "DAP_FT_MAY_2022_TP065778" is created for the purpose of storing all related project files, programs, and dataset related to this study. Figure 5.1 shows the SAS interface indicating the project folder has been created and stored under "Files (Home)" tab.

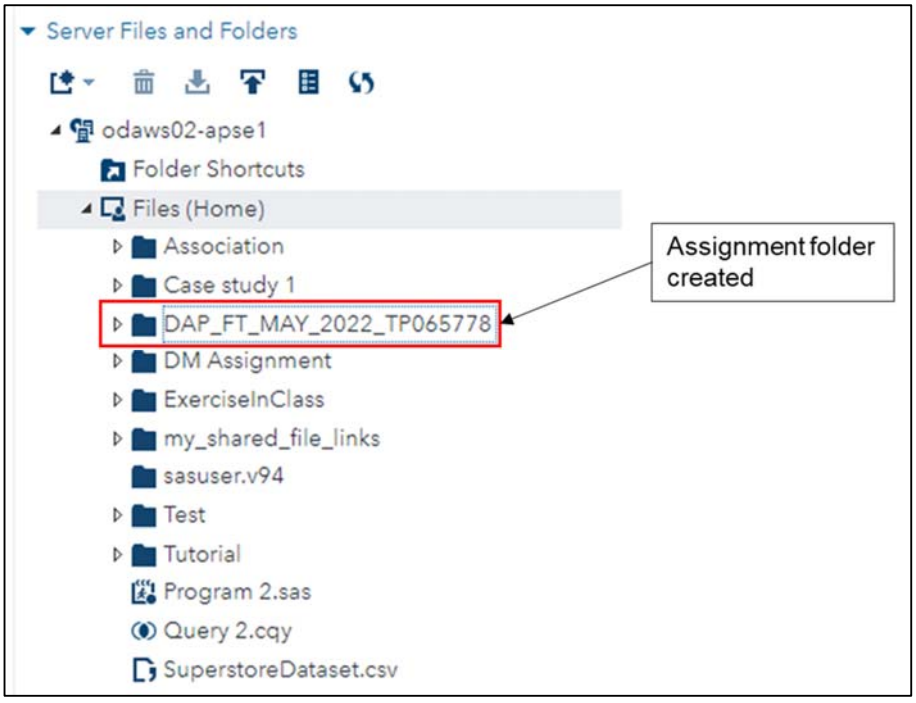


Figure 5.1: Folder creation in SAS

5.1.2 Uploading Datasets

Two datasets were provided by LFI to be analyzed and to produce a prediction model. The datasets were named "TRAINING_DS.csv" and "TESTING_DS.csv". The datasets would need to be uploaded to the SAS platform for performing the analysis and developing the model. Figure 5.2 and Figure 5.3 shows the upload of both the datasets to the project folder. Once the datasets have successfully uploaded to SAS, expanding the project folder would display the two datasets available. Figure 5.4 shows the two datasets have successfully uploaded into the project folder.

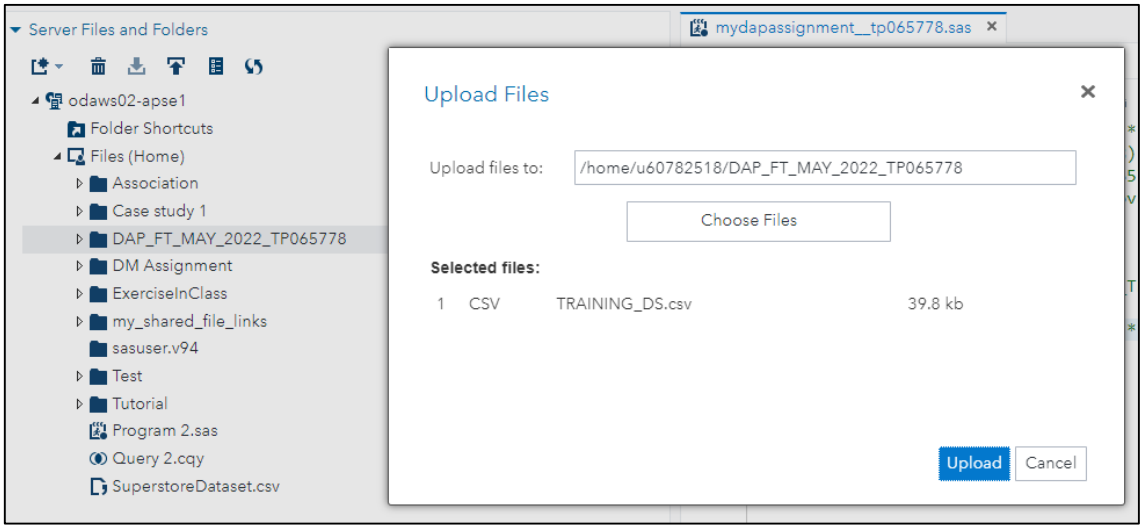


Figure 5.2: Training dataset upload to SAS

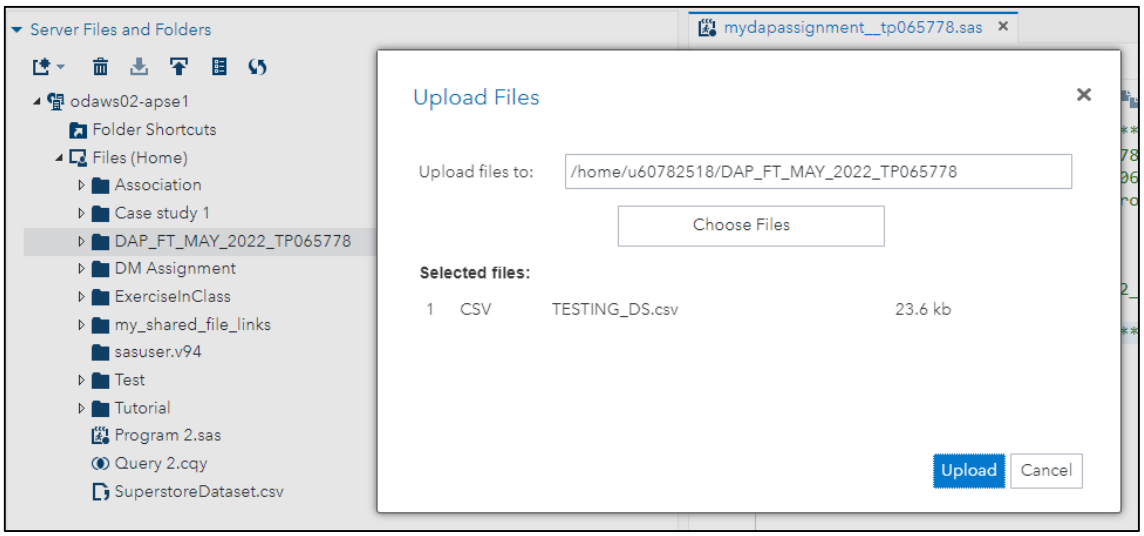


Figure 5.3: Testing dataset upload to SAS

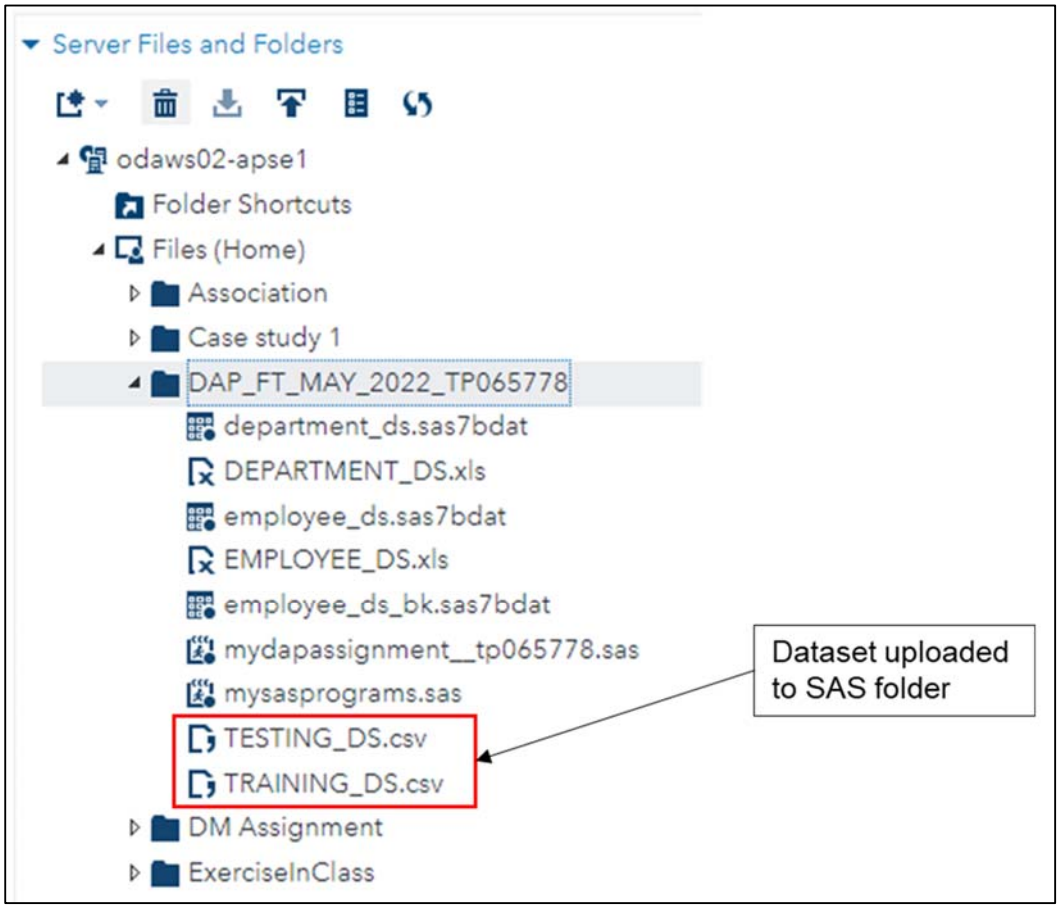


Figure 5.4: Uploaded dataset shown in project folder

5.1.3 Library Creation

There exist two types of user defined libraries in SAS, namely temporary libraries and permanent libraries. Datasets and views stored in the temporary libraries are available until the end of each SAS session and will be deleted automatically. This means that datasets and views that are previously stored in the temporary libraries will not be available at the start of a new SAS session. While datasets and views stored in the permanent libraries will be retained even after the end of each SAS session. This means that datasets and views that are previously stored in the permanent libraries will be available at the start of a new SAS session. It will only be deleted when the user decides to delete the libraries.

The use of permanent libraries is important for saving the progress made onto the datasets as it will be reused at the start of each new SAS session. The default temporary library by SAS is named “WORK”. While the name for the permanent libraries can be defined by the users. A permanent library named “LIB65778” is created for this project as shown in Figure 5.5. By

best practice, the length of the permanent library name should be confined within eight alphanumeric characters.

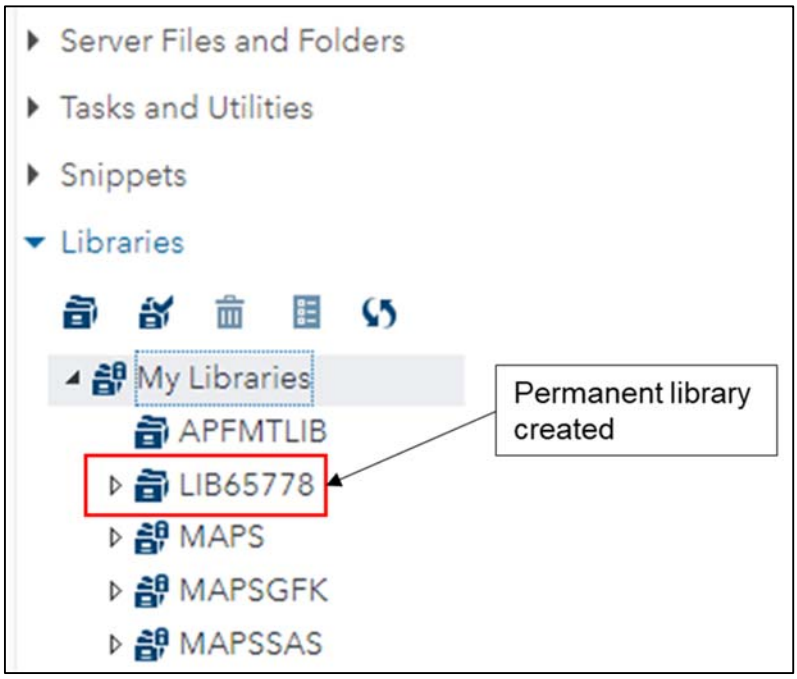


Figure 5.5: Creation of permanent library

5.1.4 Import Dataset to Created Library

It is required to import the specified datasets into the SAS libraries to perform analytics on the SAS platform. Previously, two datasets are uploaded into the project folder. The datasets are now to be imported into the permanent library. The “Import Data” utility can be used to import the uploaded datasets into the permanent library as shown in Figure 5.6. Both the datasets will be imported to the permanent library. Once the datasets are successfully imported, it will be displayed under the permanent library. As shown in Figure 5.7, the datasets “TRAINING_DS” and “TESTING_DS” have successfully imported and shown in the “LIB65778” permanent library.

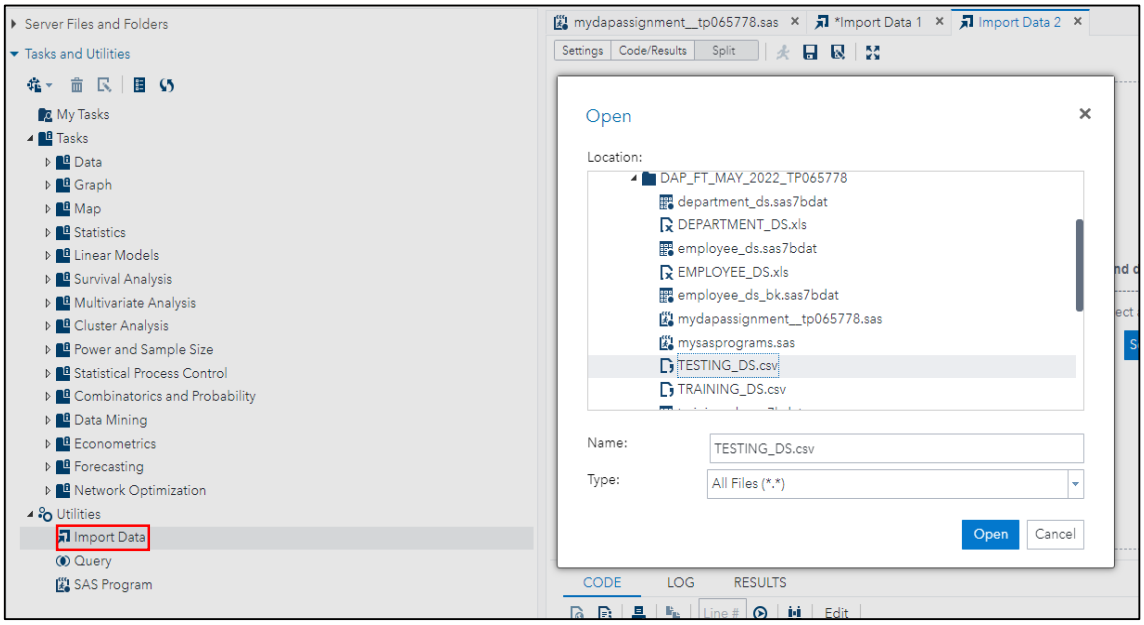


Figure 5.6: Dataset import to permanent library

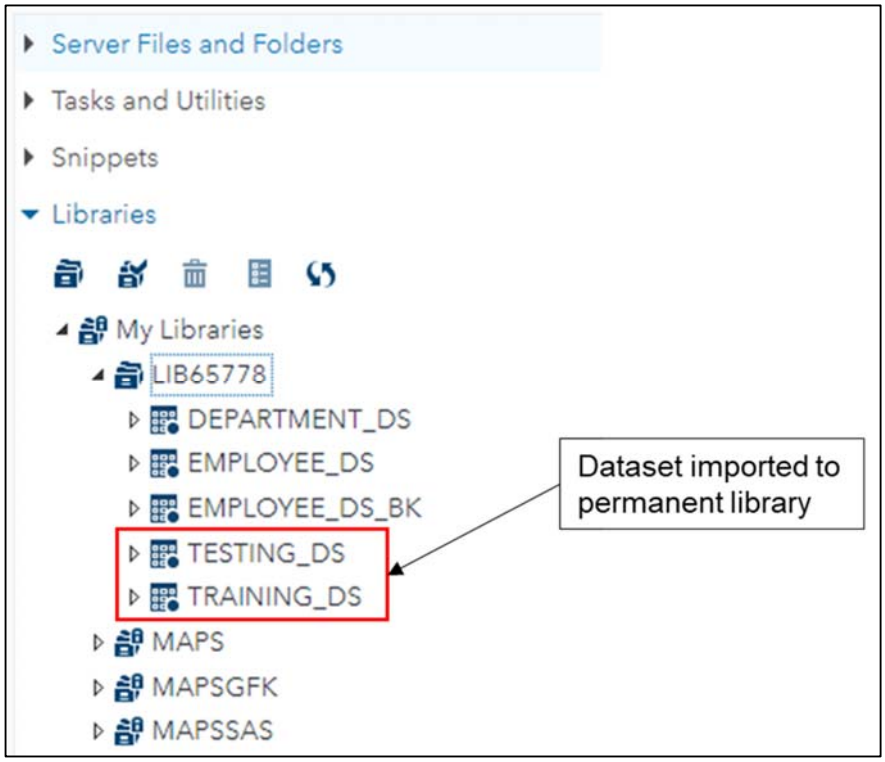


Figure 5.7: Imported dataset in permanent library

5.1.4 Data Dictionary – Structure of the Dataset

Data dictionary or metadata describes the data about data which provides an overview of the information about the dataset. Figure 5.8 shows the SAS code executed to produce the data dictionary for the dataset "TRAINING_DS". Outputs from the code execution is shown in Figure 5.9 to Figure 5.11. Relevant data dictionary for the dataset includes name of dataset, time when the dataset is created and modified, number of observations, number of variables, file size, variable name, variable type, and variable length. The number of observations identified is 614, while the number of variables identified is 13. The variables within the dataset contained a mixed of variable types between character and numeric as shown in Figure 5.11.

```
/*****************************************
Name of DS: Mr.LEE KEAN LIM (TP065778)
Name of program: mydapassignment_tp065778.sas
Description: Performing data analytics on a dataset comprising of SME customer details.
Dataset was provided by LFI with the objective to understand customer behavior and automate
the loan approval process. This program aimed to develop a highly accurate prediction model
to predict the loan approval status.
Date first written: Thu,23-Jun-2022
Date last updated: Thu,7-July-2022

Project Folder name: DAP_FT_MAY_2022_TP065778
Permanent Library name: LIB65778
*****************************************/

/****************** Data Dictionary - LIB65778.TRAINING_DS *****************/
TITLE1 'Structure/Data Dictionary of the dataset - LIB65778.TRAINING_DS';
PROC CONTENTS DATA = LIB65778.TRAINING_DS;
RUN;
```

Figure 5.8: SAS code for generating data dictionary

Structure/Data Dictionary of the dataset - LIB65778.TRAINING_DS

The CONTENTS Procedure

Data Set Name	LIB65778.TRAINING_DS	**Observations**	614
Member Type	DATA	**Variables**	13
Engine	V9	**Indexes**	0
Created	06/23/2022 15:09:56	**Observation Length**	96
Last Modified	06/23/2022 15:09:56	**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)		

Figure 5.9: SAS generated summary information of dataset

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/u60782518/DAP_FT_MAY_2022_TP065778/training_ds.sas7bdat
Release Created	9.0401M6
Host Created	Linux
Inode Number	135268851
Access Permission	rw-r--r--
Owner Name	u60782518
File Size	256KB
File Size (bytes)	262144

Figure 5.10: SAS generated engine dependent information

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.

Structure of data for LIB65778.TRAINING_DS

Figure 5.11: SAS generated data dictionary of dataset

5.2 ANALYSIS OF VARIABLES – LIB65778.TRAINING_DS

This section documents the analysis of categorical and continuous variables found in the "TRAINING_DS" dataset. Univariate and bivariate analysis will be performed on the variables to identify patterns among data, relationship between variables, and missing values in each variable.

5.2.1 Univariate Analysis of the Categorical Variable – MARITAL_STATUS

Figure 5.12 shows the SAS code executed to produce the univariate analysis for the categorical variable "MARITAL_STATUS". The output from the code execution is shown in the table format as shown in Table 5.1 and in the graphical format as shown in Figure 5.13.

```
20 /****************** Univariate Analysis - Categorical - MARITAL_STATUS ******************/
21 TITLE 'Univariate Analysis of the categorical variable: MARITAL_STATUS';
22
23 PROC FREQ DATA = LIB65778.TRAINING_DS;
24
25 TABLE MARITAL_STATUS;
26
27 RUN;
28
29 ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
30
31 PROC SGPLOT DATA = LIB65778.TRAINING_DS;
32
33 VBAR MARITAL_STATUS;
34 TITLE 'Univariate Analysis of the categorical variable: MARITAL_STATUS';
35
36 RUN;
```

Figure 5.12: SAS code for univariate analysis of variable: MARITAL_STATUS

Table 5.1: Table output of univariate analysis for variable: MARITAL_STATUS

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				

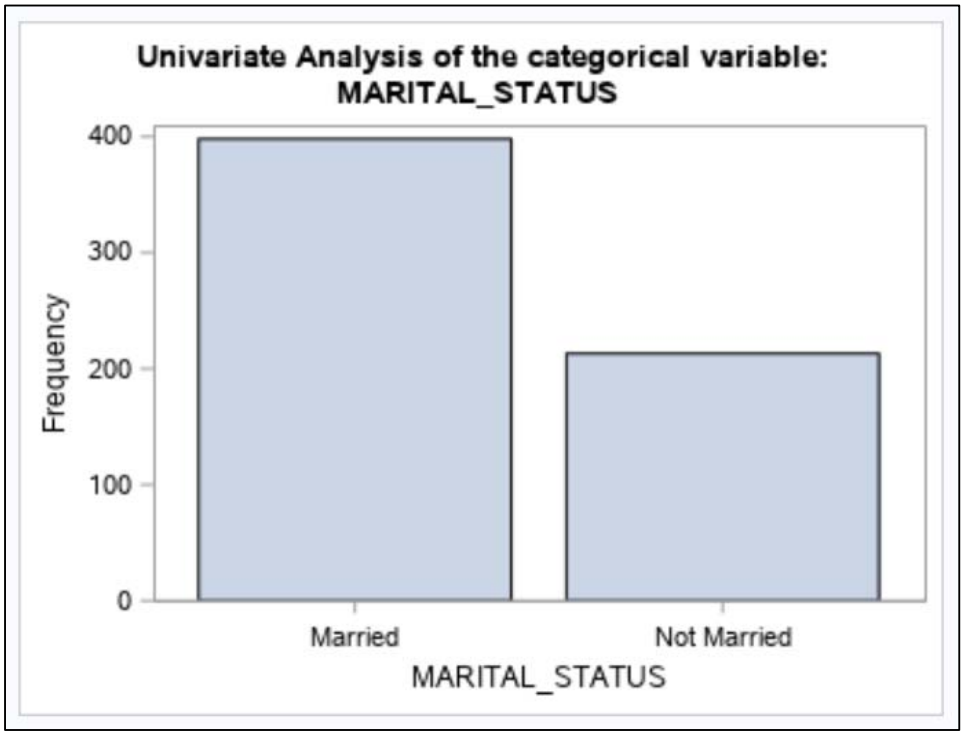


Figure 5.13: Graphical output of univariate analysis for variable: MARITAL_STATUS

Based on the outputs, it can be identified that there are three missing values in this variable, thus would require to undergo imputation at a later stage. Of the 611 applicants, there are 398 applicants who are married (65.14%) while 213 applicants are not married (34.86%). This indicates that a high number of applicants who are married are seeking to acquire loan facilities from the bank. This may be due to the higher commitment faced by married applicants thus requiring additional fundings.

5.2.2 Univariate Analysis of the Categorical Variable – QUALIFICATION

Figure 5.14 shows the SAS code executed to produce the univariate analysis for the categorical variable "QUALIFICATION". The output from the code execution is shown in the table format as shown in Table 5.2 and in the graphical format as shown in Figure 5.15.

```
/****************** Univariate Analysis - Categorical - QUALIFICATION *****************/
TITLE 'Univariate Analysis of the categorical variable: QUALIFICATION';

PROC FREQ DATA = LIB65778.TRAINING_DS;

TABLE QUALIFICATION;

RUN;

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

PROC SGPLOT DATA = LIB65778.TRAINING_DS;

VBAR QUALIFICATION;
TITLE 'Univariate Analysis of the categorical variable: QUALIFICATION';

RUN;
```

Figure 5.14: SAS code for univariate analysis of variable: QUALIFICATION

Table 5.2: Table output of univariate analysis for variable: QUALIFICATION

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

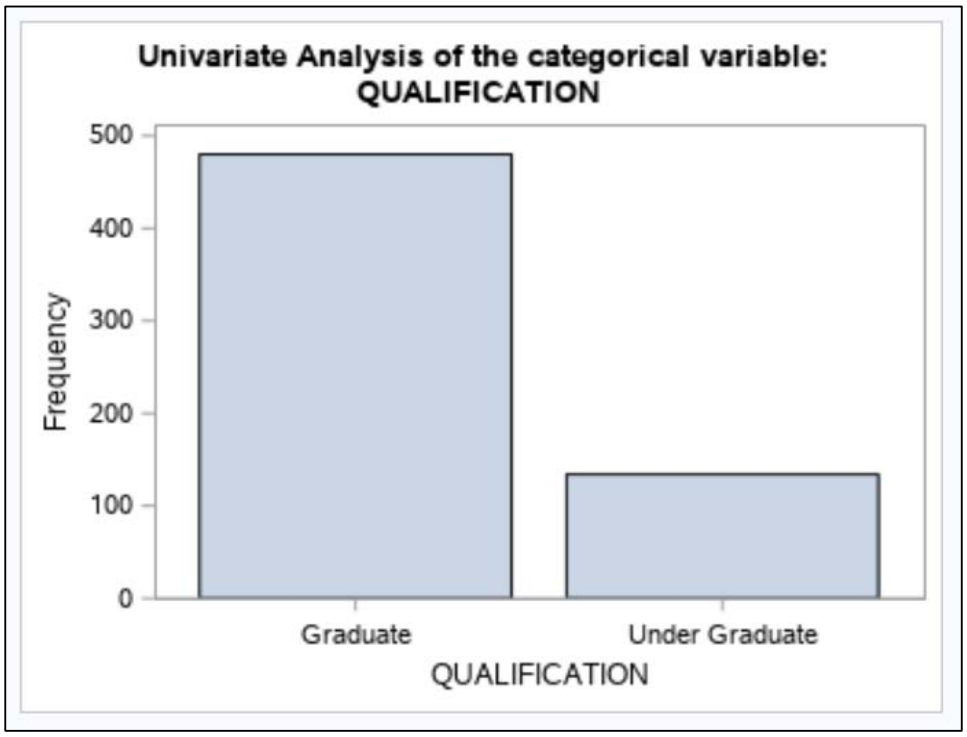


Figure 5.15: Graphical output of univariate analysis for variable: QUALIFICATION

Based on the outputs, it can be identified that there are no missing values in this variable. Of the 614 applicants, there are 480 applicants who are graduate (78.18%) while 134 applicants are undergraduate (21.82%). This indicates that a high number of applicants who are graduate are seeking to acquire loan facilities from the bank. This may be due to applicants who are graduate are more knowledgeable in managing debt and risks, thus are more likely to seek financing from the bank to venture into different businesses or to promote the growth of their companies.

5.2.3 Univariate Analysis of the Categorical Variable – FAMILY_MEMBERS

Figure 5.16 shows the SAS code executed to produce the univariate analysis for the categorical variable "FAMILY_MEMBERS". The output from the code execution is shown in the table format as shown in Table 5.3 and in the graphical format as shown in Figure 5.17.

```
/***************** Univariate Analysis - Categorical - FAMILY_MEMBERS *****************/
TITLE 'Univariate Analysis of the categorical variable: FAMILY_MEMBERS';

PROC FREQ DATA = LIB65778.TRAINING_DS;

TABLE FAMILY_MEMBERS;

RUN;

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

PROC SGPLOT DATA = LIB65778.TRAINING_DS;

VBAR FAMILY_MEMBERS;
TITLE 'Univariate Analysis of the categorical variable: FAMILY_MEMBERS';

RUN;
```

Figure 5.16: SAS code for univariate analysis of variable: FAMILY_MEMBERS

Table 5.3: Table output of univariate analysis for variable: FAMILY_MEMBERS

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				

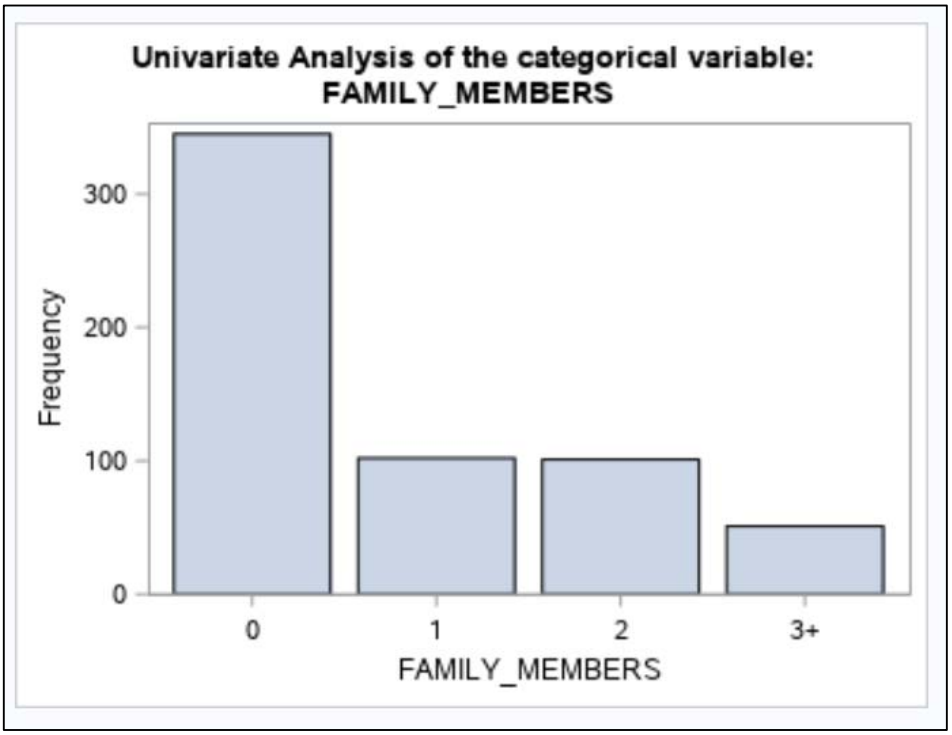


Figure 5.17: Graphical output of univariate analysis for variable: FAMILY_MEMBERS

Based on the outputs, it can be identified that there are 15 missing values in this variable, thus would require to undergo imputation at a later stage. Of the 599 applicants, there are 345 applicants without any dependents (57.60%), 102 applicants with one dependent (17.03%), 101 applicants with two dependents (16.86%), and 51 applicants with three or more dependents (8.51%). This indicates that a higher number of applicants who have lesser dependents are likely to seek for loan facilities from the bank. This may be due to the risks involved when acquiring debt, which applicants with no dependents are willing to take higher risks as they have much lesser commitment and responsibility to be fulfilled.

5.2.4 Univariate Analysis of the Categorical Variable – GENDER

Figure 5.18 shows the SAS code executed to produce the univariate analysis for the categorical variable "GENDER". The output from the code execution is shown in the table format as shown in Table 5.4 and in the graphical format as shown in Figure 5.19.

```
74 /****************** Univariate Analysis - Categorical - GENDER ******************/
75 TITLE 'Univariate Analysis of the categorical variable: GENDER';
76
77 PROC FREQ DATA = LIB65778.TRAINING_DS;
78
79 TABLE GENDER;
80
81 RUN;
82
83 ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
84
85 PROC SGPLOT DATA = LIB65778.TRAINING_DS;
86
87 VBAR GENDER;
88 TITLE 'Univariate Analysis of the categorical variable: GENDER';
89
90 RUN;
```

Figure 5.18: SAS code for univariate analysis of variable: GENDER

Table 5.4: Table output of univariate analysis for variable: GENDER

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				

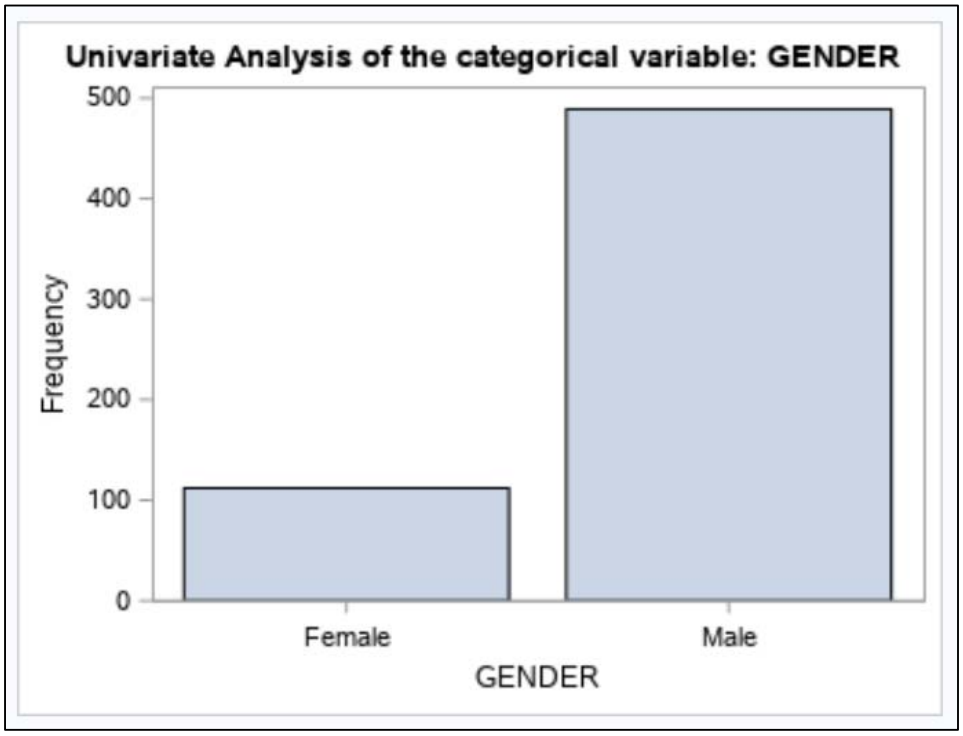


Figure 5.19: Graphical output of univariate analysis for variable: GENDER

Based on the outputs, it can be identified that there are 13 missing values in this variable, thus would require to undergo imputation at a later stage. Of the 601 applicants, there are 112 female applicants (18.64%) and 489 male applicants (81.36%). This indicates that a high number of male applicants are seeking to acquire loan facilities from the bank. This may be due to the risks involved in developing a business which the males are more likely to take risks as compared to females.

5.2.5 Univariate Analysis of the Categorical Variable – EMPLOYMENT

Figure 5.20 shows the SAS code executed to produce the univariate analysis for the categorical variable "EMPLOYMENT". The output from the code execution is shown in the table format as shown in Table 5.5 and in the graphical format as shown in Figure 5.21.

```
92  /***************** Univariate Analysis - Categorical - EMPLOYMENT *****************/
93  TITLE 'Univariate Analysis of the categorical variable: EMPLOYMENT';
94
95  PROC FREQ DATA = LIB65778.TRAINING_DS;
96
97  TABLE EMPLOYMENT;
98
99  RUN;
100
101 ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
102
103 PROC SGPLOT DATA = LIB65778.TRAINING_DS;
104
105 VBAR EMPLOYMENT;
106 TITLE 'Univariate Analysis of the categorical variable: EMPLOYMENT';
107
108 RUN;
```

Figure 5.20: SAS code for univariate analysis of variable: EMPLOYMENT

Table 5.5: Table output of univariate analysis for variable: EMPLOYMENT

Univariate Analysis of the categorical variable: EMPLOYMENT

The FREQ Procedure

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

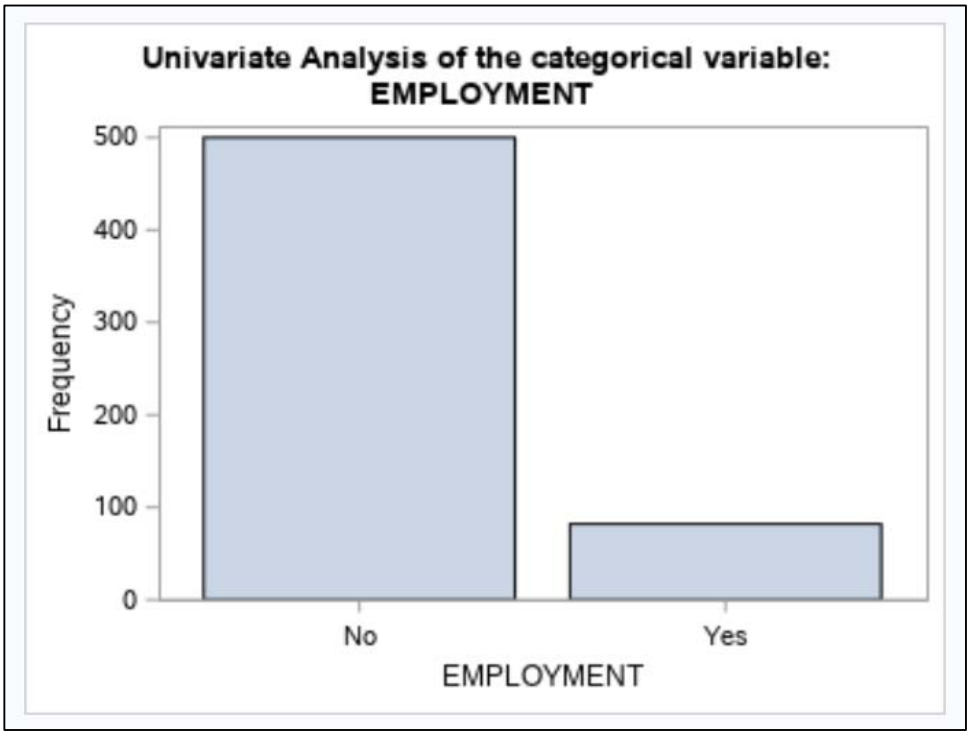


Figure 5.21: Graphical output of univariate analysis for variable: EMPLOYMENT

Based on the outputs, it can be identified that there are 32 missing values in this variable, thus would require to undergo imputation at a later stage. Of the 582 applicants, there are 500 applicants who are unemployed (85.91%) while 82 applicants are employed (14.09%). This indicates that a high number of applicants who are unemployed are seeking to acquire loan facilities from the bank. This may be due to the lack of fundings from the unemployed group to raise sufficient startup capital thus requiring additional fundings from the bank.

5.2.6 Univariate Analysis of the Categorical Variable – LOAN_HISTORY

Figure 5.22 shows the SAS code executed to produce the univariate analysis for the categorical variable "LOAN_HISTORY". The output from the code execution is shown in the table format as shown in Table 5.6 and in the graphical format as shown in Figure 5.23.

```
/****************** Univariate Analysis - Categorical - LOAN_HISTORY ******************/
TITLE 'Univariate Analysis of the categorical variable: LOAN_HISTORY';

PROC FREQ DATA = LIB65778.TRAINING_DS;

TABLE LOAN_HISTORY;

RUN;

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

PROC SGPLOT DATA = LIB65778.TRAINING_DS;

VBAR LOAN_HISTORY;
TITLE 'Univariate Analysis of the categorical variable: LOAN_HISTORY';

RUN;
```

Figure 5.22: SAS code for univariate analysis of variable: LOAN_HISTORY

Table 5.6: Table output of univariate analysis for variable: LOAN_HISTORY

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				

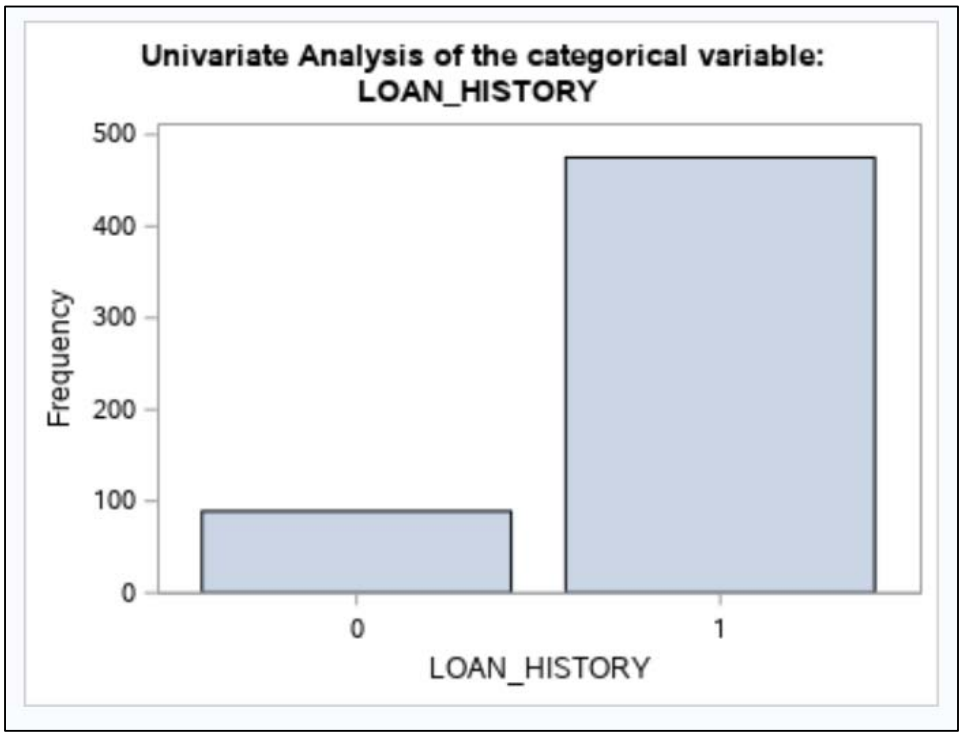


Figure 5.23: Graphical output of univariate analysis for variable: LOAN_HISTORY

Based on the outputs, it can be identified that there are 50 missing values in this variable, thus would require to undergo imputation at a later stage. Of the 564 applicants, there are 89 applicants who have missed repayment in loan history (15.78%) while 475 applicants have no missed repayment in loan history (84.22%). This indicates that a high number of applicants who have good repayment history are seeking to acquire loan facilities from the bank. This may be due to the higher chances of getting the loan approved due to the good credit history from the applicants who have good repayment loan history.

5.2.7 Univariate Analysis of the Categorical Variable – LOAN_LOCATION

Figure 5.24 shows the SAS code executed to produce the univariate analysis for the categorical variable "LOAN_LOCATION". The output from the code execution is shown in the table format as shown in Table 5.7 and in the graphical format as shown in Figure 5.25.

```
/****************** Univariate Analysis - Categorical - LOAN_LOCATION *****************/
TITLE 'Univariate Analysis of the categorical variable: LOAN_LOCATION';

PROC FREQ DATA = LIB65778.TRAINING_DS;

TABLE LOAN_LOCATION;

RUN;

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

PROC SGPLOT DATA = LIB65778.TRAINING_DS;

VBAR LOAN_LOCATION;
TITLE 'Univariate Analysis of the categorical variable: LOAN_LOCATION';

RUN;
```

Figure 5.24: SAS code for univariate analysis of variable: LOAN_LOCATION

Table 5.7: Table output of univariate analysis for variable: LOAN_LOCATION

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

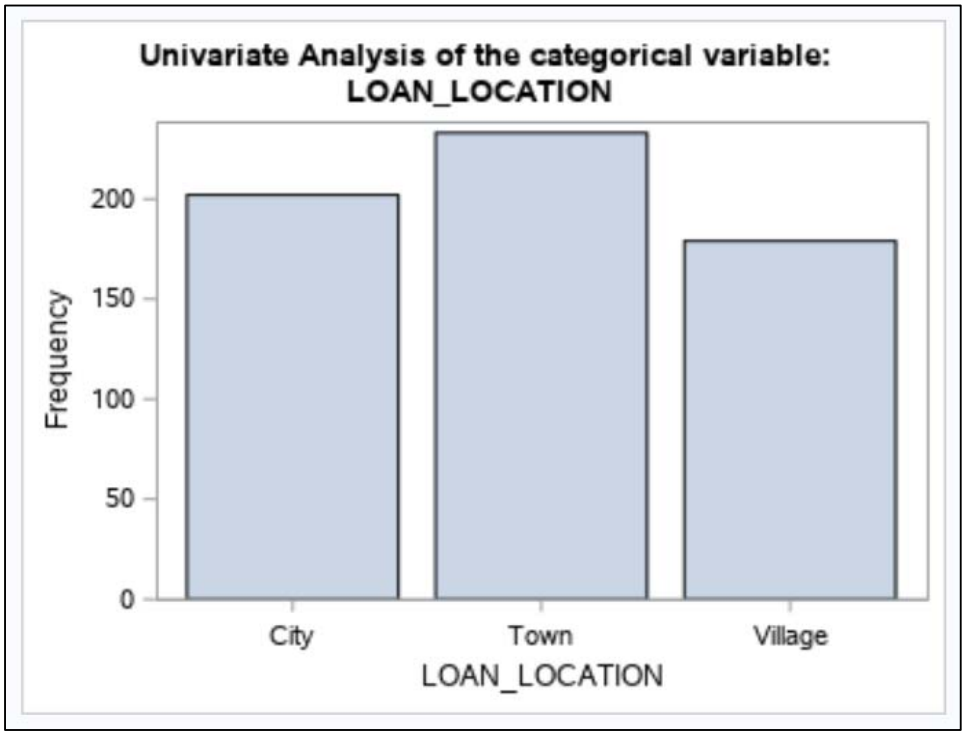


Figure 5.25: Graphical output of univariate analysis for variable: LOAN_LOCATION

Based on the outputs, it can be identified that there are no missing values in this variable. Of the 614 applicants, there are 202 applicants from the city (32.90%), 233 applicants from the town (37.95%), and 179 applicants from the village (29.15%). This indicates that a high number of applicants from the town and city are seeking to acquire loan facilities from the bank. This may be due to higher business opportunities present in the city and town area as compared to the village, thus the higher number of applicants seeking additional fundings for business startups.

5.2.8 Univariate Analysis of the Categorical Variable – LOAN_APPROVAL_STATUS

Figure 5.26 shows the SAS code executed to produce the univariate analysis for the categorical variable "LOAN_APPROVAL_STATUS". The output from the code execution is shown in the table format as shown in Table 5.8 and in the graphical format as shown in Figure 5.27.

```
/***************** Univariate Analysis - Categorical - LOAN_APPROVAL_STATUS *****************/
TITLE 'Univariate Analysis of the categorical variable: LOAN_APPROVAL_STATUS';

PROC FREQ DATA = LIB65778.TRAINING_DS;

TABLE LOAN_APPROVAL_STATUS;

RUN;

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

PROC SGPLOT DATA = LIB65778.TRAINING_DS;

VBAR LOAN_APPROVAL_STATUS;
TITLE 'Univariate Analysis of the categorical variable: LOAN_APPROVAL_STATUS';

RUN;
```

Figure 5.26: SAS code for univariate analysis of variable: LOAN_APPROVAL_STATUS

Table 5.8: Table output of univariate analysis for variable: 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

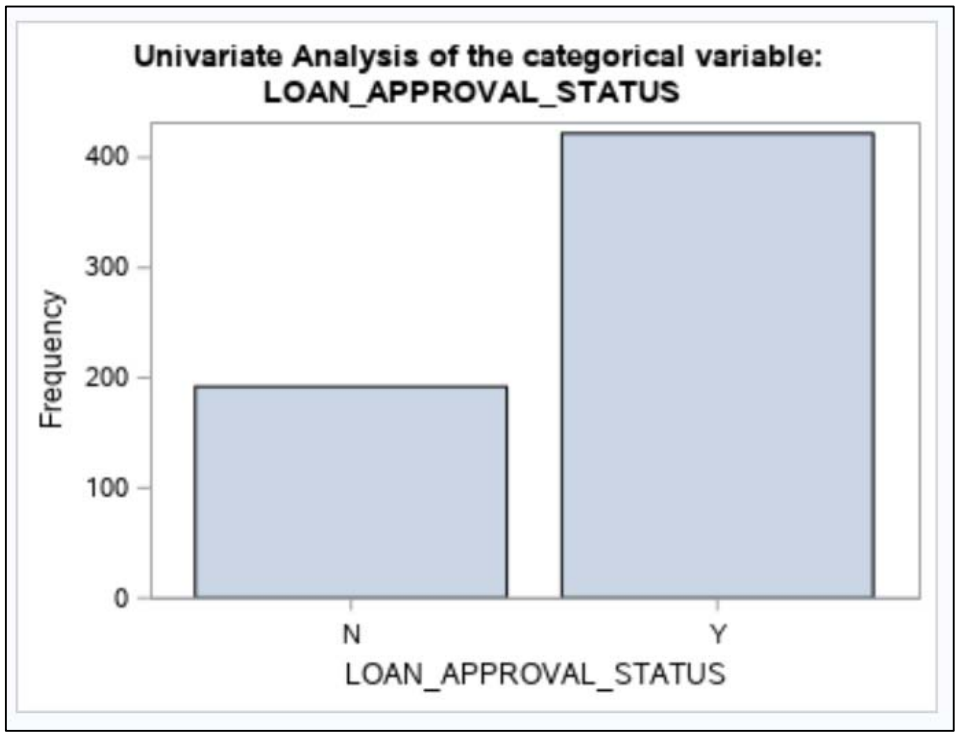


Figure 5.27: Graphical output of univariate analysis for variable: LOAN_APPROVAL_STATUS

Based on the outputs, it can be identified that there are no missing values in this variable. Of the 614 applicants, there are 192 applicants with loan rejected (31.27%) while 422 applicants got their loans approved (68.73%). This indicates that a high number of applicants got their loans approved from this bank. This may be due to the leniency in the policies defined by the bank thus resulting in a high number of loan approval rate.

5.2.9 Univariate Analysis of the Continuous Variable – CANDIDATE_INCOME

Figure 5.28 shows the SAS code executed to produce the univariate analysis for the continuous variable "CANDIDATE_INCOME". The output from the code execution is shown in the table format as shown in Table 5.9 and in the graphical format as shown in Figure 5.29.

```
164 /***************** Univariate Analysis - Continuous - CANDIDATE_INCOME *****************/
165 TITLE1 'Univariate Analysis of the Continuous variable: CANDIDATE_INCOME';
166 FOOTNOTE '-----END-----';
167
168 PROC MEANS DATA = LIB65778.TRAINING_DS N NMISS MIN MAX MEAN MEDIAN STD;
169
170 VAR CANDIDATE_INCOME;
171
172 RUN;
173
174 ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
175
176 PROC SGPLOT DATA = LIB65778.TRAINING_DS;
177
178 HISTOGRAM CANDIDATE_INCOME;
179
180 TITLE 'Univariate Analysis of the Continuous variable: CANDIDATE_INCOME';
181
182 RUN;
```

Figure 5.28: SAS code for univariate analysis of variable: CANDIDATE_INCOME

Table 5.9: Table output of univariate analysis for variable: CANDIDATE_INCOME

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

-----END-----

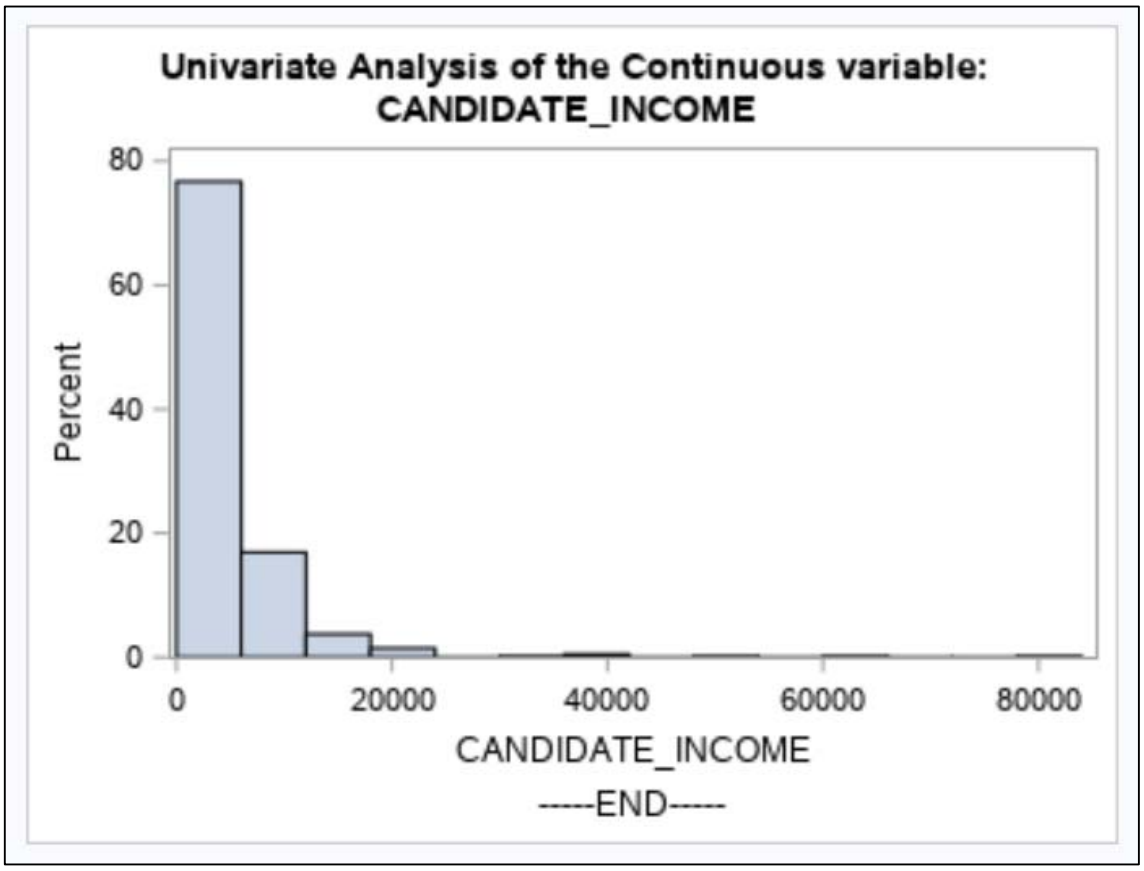


Figure 5.29: Graphical output of univariate analysis for variable: CANDIDATE_INCOME

Based on the outputs, it can be identified that there are no missing values in this variable. The variable has a range between 150 to 81,000 which represents the range of income of the applicants. The mean income of the applicants is 5,403.46 which is greater than the median income of the applicants which is 3812.50. This indicates that the sample distribution has a positive skew, where most of the applicants will have an income on the lower range. It is evident based on Figure 5.29 where about 75% of the applicants have an income around 5400. This information can signify the spending power of the applicants, where a higher income would likely indicate a higher ability of the applicant to repay the loan.

5.2.10 Univariate Analysis of the Continuous Variable – LOAN_DURATION

Figure 5.30 shows the SAS code executed to produce the univariate analysis for the continuous variable "LOAN_DURATION". The output from the code execution is shown in the table format as shown in Table 5.10 and in the graphical format as shown in Figure 5.31.

```
/***************** Univariate Analysis - Continuous - LOAN_DURATION *****************/
TITLE1 'Univariate Analysis of the Continuous variable: LOAN_DURATION';
FOOTNOTE '-----END-----';

PROC MEANS DATA = LIB65778.TRAINING_DS 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 = LIB65778.TRAINING_DS;

HISTOGRAM LOAN_DURATION;

TITLE 'Univariate Analysis of the Continuous variable: LOAN_DURATION';

RUN;
```

Figure 5.30: SAS code for univariate analysis of variable: LOAN_DURATION

Table 5.10: Table output of univariate analysis for variable: LOAN_DURATION

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

-----END-----

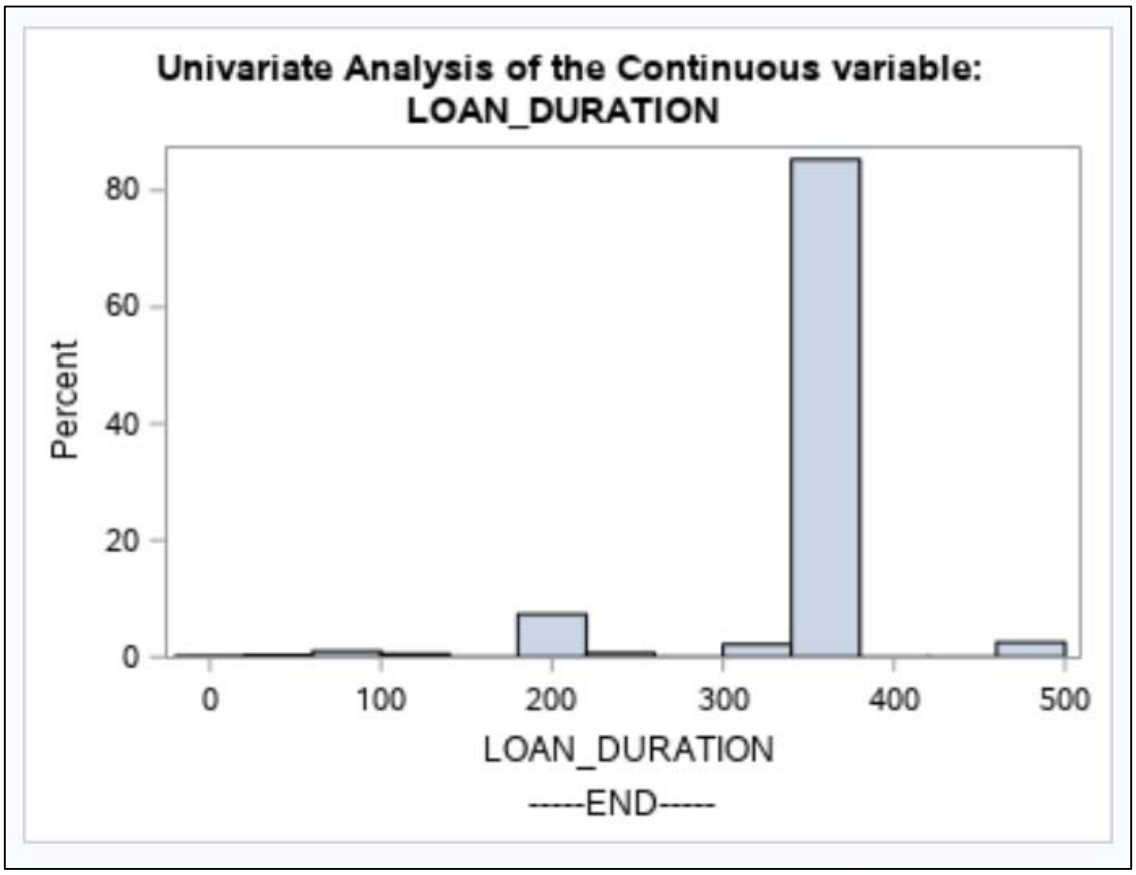


Figure 5.31: Graphical output of univariate analysis for variable: LOAN_DURATION

Based on the outputs, it can be identified that there are 14 missing values in this variable, thus would require to undergo imputation at a later stage. The variable has a range between 12 to 480 which represents the range of loan duration of the applicants in the unit of months. The mean duration of the loan is 342 months which is lower than the median loan duration of 360 months. This indicates that the sample distribution has a negative skew, where most of the applicants will have a loan duration on the higher range. It is evident based on Figure 5.31 where about 85% of the applicants have a loan duration around 360 months. This can be explained due to the 360 months duration is the typical loan duration when an applicant is obtaining a large amount of loan to lower the monthly repayment value but stretched over a longer period of time.

5.2.11 Univariate Analysis of the Continuous Variable – GUARANTEE_INCOME

Figure 5.32 shows the SAS code executed to produce the univariate analysis for the continuous variable "GUARANTEE_INCOME". The output from the code execution is shown in the table format as shown in Table 5.11 and in the graphical format as shown in Figure 5.33.

```
/****************** Univariate Analysis - Continuous - GUARANTEE_INCOME *****************/
TITLE1 'Univariate Analysis of the Continuous variable: GUARANTEE_INCOME';
FOOTNOTE '-----END-----';

PROC MEANS DATA = LIB65778.TRAINING_DS 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 = LIB65778.TRAINING_DS;

HISTOGRAM GUARANTEE_INCOME;

TITLE 'Univariate Analysis of the Continuous variable: GUARANTEE_INCOME';

RUN;
```

Figure 5.32: SAS code for univariate analysis of variable: GUARANTEE_INCOME

Table 5.11: Table output of univariate analysis for variable: GUARANTEE_INCOME

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

-----END-----

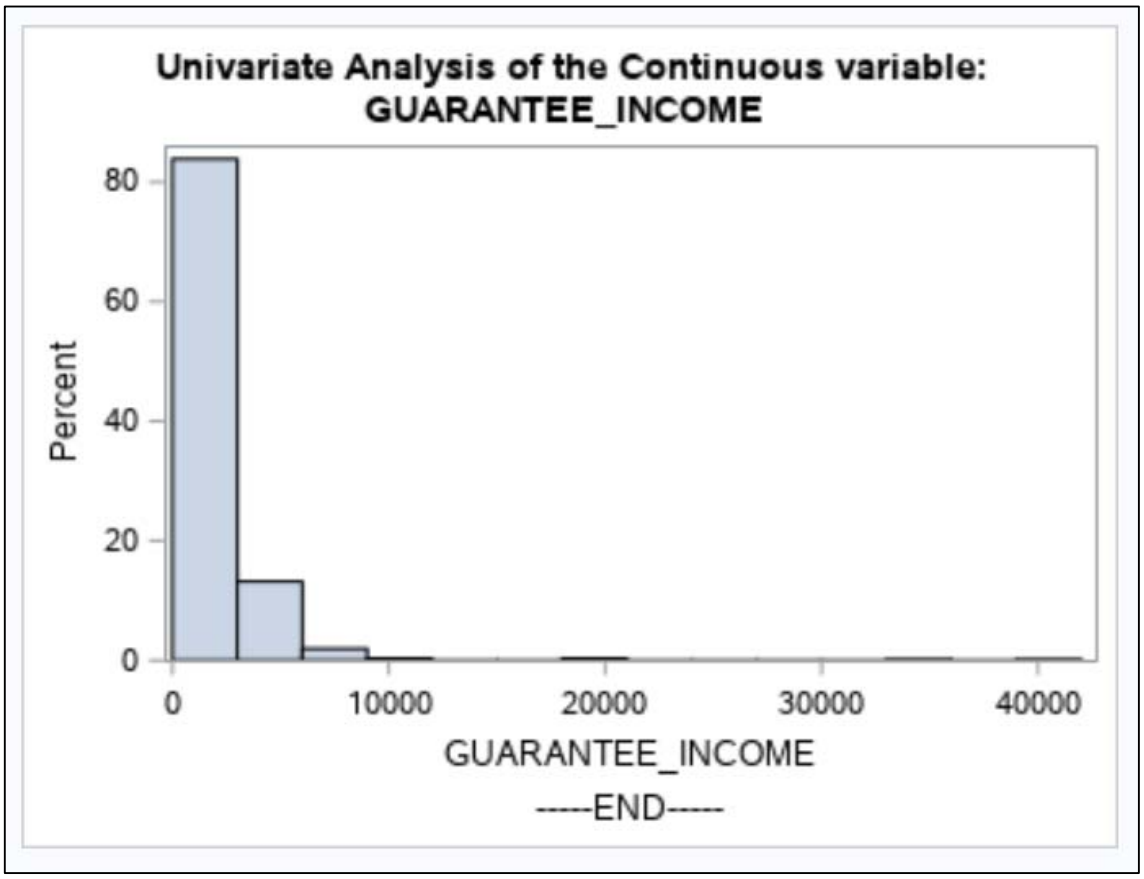


Figure 5.33: Graphical output of univariate analysis for variable: GUARANTEE_INCOME

Based on the outputs, it can be identified that there are no missing values in this variable. The variable has a range between 0 to 41,667 which represents the range of income of the co-applicants. The mean income of the co-applicants is 1621.25 which is greater than the median income of the co-applicants which is 1188.50. This indicates that the sample distribution has a positive skew, where most of the co-applicants will have an income on the lower range. It is evident based on Figure 5.33 where about 85% of the co-applicants have an income around 1621. This can be explained due to the commitment of co-applicant leading to higher unemployment or having to adopt a more flexible employment which pays lesser.

5.2.12 Univariate Analysis of the Continuous Variable – LOAN_AMOUNT

Figure 5.34 shows the SAS code executed to produce the univariate analysis for the continuous variable "LOAN_AMOUNT". The output from the code execution is shown in the table format as shown in Table 5.12 and in the graphical format as shown in Figure 5.35.

```
/****************** Univariate Analysis - Continuous - LOAN_AMOUNT *****************/
TITLE1 'Univariate Analysis of the Continuous variable: LOAN_AMOUNT';
FOOTNOTE '-----END-----';

PROC MEANS DATA = LIB65778.TRAINING_DS 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 = LIB65778.TRAINING_DS;

HISTOGRAM LOAN_AMOUNT;

TITLE 'Univariate Analysis of the Continuous variable: LOAN_AMOUNT';

RUN;
```

Figure 5.34: SAS code for univariate analysis of variable: LOAN_AMOUNT

Table 5.12: Table output of univariate analysis for variable: LOAN_AMOUNT

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

-----END-----

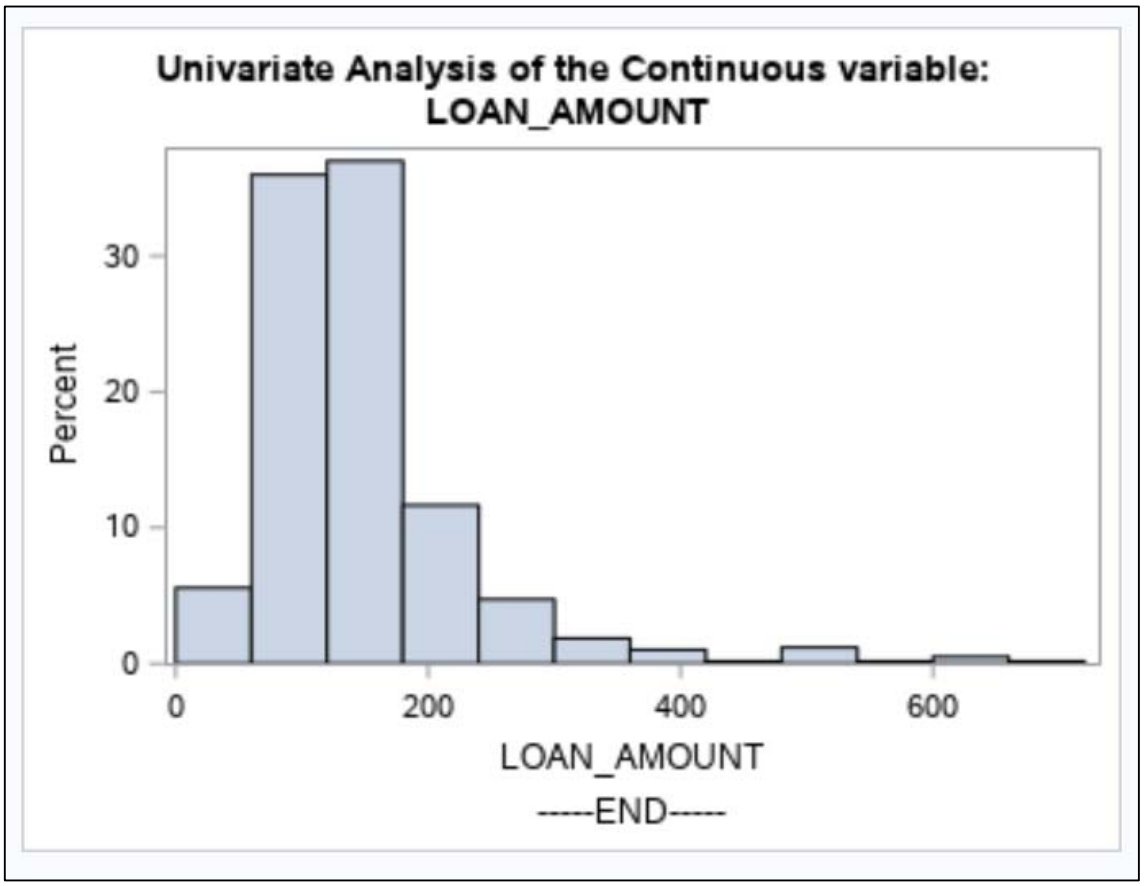


Figure 5.35: Graphical output of univariate analysis for variable: LOAN_AMOUNT

Based on the outputs, it can be identified that there are 22 missing values in this variable, thus would require to undergo imputation at a later stage. The variable has a range between 9 to 700 which represents the range of loan amount applied by the applicants in the units of thousands. The mean loan amount is 146.41 thousand which is greater than the median loan amount which is 128 thousand. This indicates that the sample distribution has a positive skew, where most of the applicants are applying a loan amount closer to the lower loan amount range. It is evident based on Figure 5.35 where about 70% of the applications are borrowing about 130 thousand in loan. This can be explained due to the nature of small businesses where it does not require a huge sum of money for startup or business growth.

5.2.13 Bivariate Analysis of the Variables (MARITAL_STATUS – Categorical variable versus LOAN_APPROVAL_STATUS – Categorical variable)

Figure 5.36 shows the SAS code executed to produce the bivariate analysis for the categorical variable "MARITAL_STATUS" and categorical variable "LOAN_APPROVAL_STATUS". The output from the code execution is shown in the table format as shown in Table 5.13 and in the graphical format as shown in Figure 5.37.

```
/****************** Bivariate Analysis - Categorical vs Categorical - MARITAL_STATUS vs LOAN_APPROVAL_STATUS ******************/
TITLE1 'Bivariate Analysis of variables: ';
TITLE2 '(MARITAL_STATUS - Categorical variables vs LOAN_APPROVAL_STATUS - Categorical variable)';
FOOTNOTE '-----END-----';

PROC FREQ DATA = LIB65778.TRAINING_DS;

TABLE MARITAL_STATUS * LOAN_APPROVAL_STATUS /
PLOTS = FREQPLOT( TWOWAY = STACKED SCALE = GROUPPCT );

RUN;
```

Figure 5.36: SAS code for bivariate analysis of variables: MARITAL_STATUS versus LOAN_APPROVAL_STATUS

Table 5.13: Table output of bivariate analysis for variables: MARITAL_STATUS versus LOAN_APPROVAL_STATUS

Bivariate Analysis of variables:
(MARITAL_STATUS - Categorical variables vs LOAN_APPROVAL_STATUS - Categorical variable)

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

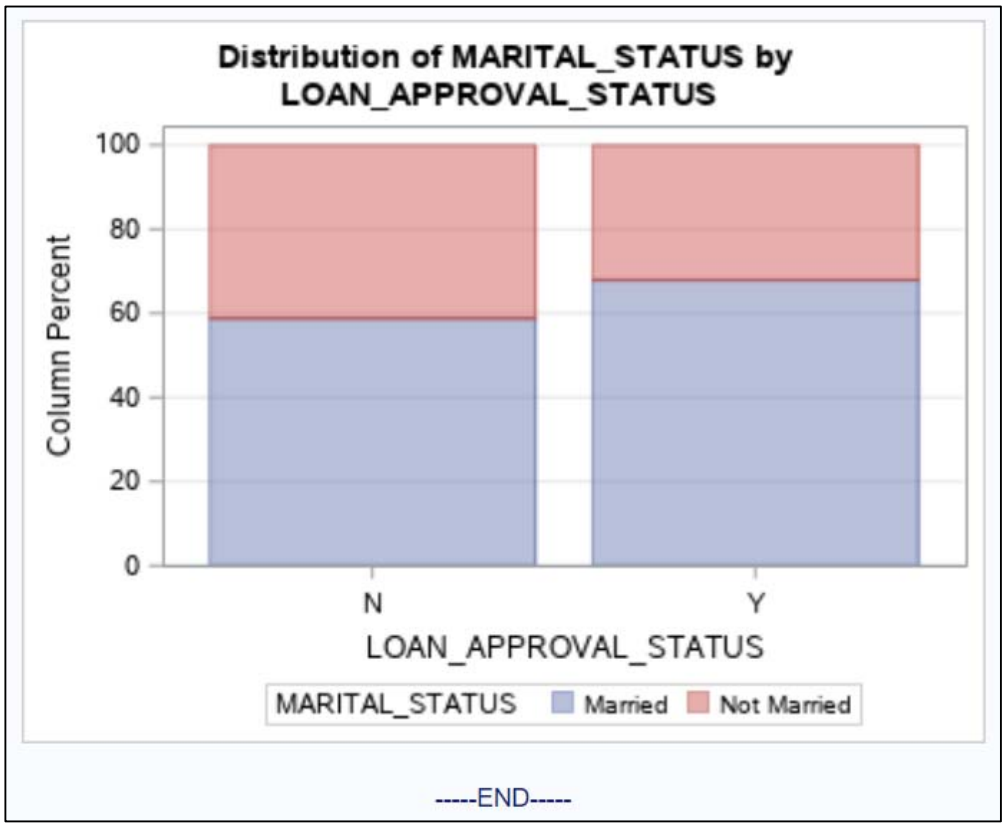


Figure 5.37: Graphical output of bivariate analysis for variables: MARITAL_STATUS versus LOAN_APPROVAL_STATUS

In overall, there are 419 approved loans (68.58%) which is greater than the 192 rejected loans (31.42%). Of the 611 applicants, there are 398 applicants who are married (65.14%) and 213 applicants who are not married (34.86%). For the 398 married applicants, there are 285 applicants with their loans approved (71.61%) while 113 applicants have their loans rejected (28.39%). For the 213 unmarried applicants, there are 134 applicants with their loans approved (62.91%) while 79 applicants have their loans rejected (37.09%). It can be identified that married applicants have a higher rate of loan approval as compared to unmarried applicants. This may be due to the married applicants potentially having a higher combined income resulting in higher credit worthiness as compared to unmarried applicants. Thus, leading to higher rate of loan approval observed in the married applicants.

5.2.14 Bivariate Analysis of the Variables (LOAN_LOCATION – Categorical variable versus LOAN_APPROVAL_STATUS – Categorical variable)

Figure 5.38 shows the SAS code executed to produce the bivariate analysis for the categorical variable "LOAN_LOCATION" and categorical variable "LOAN_APPROVAL_STATUS". The output from the code execution is shown in the table format as shown in Table 5.14 and in the graphical format as shown in Figure 5.39.

```
/****************** Bivariate Analysis - Categorical vs Categorical - LOAN_LOCATION vs LOAN_APPROVAL_STATUS ******************/
TITLE1 'Bivariate Analysis of variables: ';
TITLE2 '(LOAN_LOCATION - Categorical variables vs LOAN_APPROVAL_STATUS - Categorical variable)';
FOOTNOTE '-----END-----';

PROC FREQ DATA = LIB65778.TRAINING_DS;

TABLE LOAN_LOCATION * LOAN_APPROVAL_STATUS /
PLOTS = FREQPLOT( TWOWAY = STACKED SCALE = GROUPPCT );

RUN;
```

Figure 5.38: SAS code for bivariate analysis of variables: LOAN_LOCATION versus LOAN_APPROVAL_STATUS

Table 5.14: Table output of bivariate analysis for variables: LOAN_LOCATION versus LOAN_APPROVAL_STATUS

Bivariate Analysis of variables:
(LOAN_LOCATION - Categorical variables vs LOAN_APPROVAL_STATUS - Categorical variable)

The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of LOAN_LOCATION by LOAN_APPROVAL_STATUS

LOAN_LOCATION	LOAN_APPROVAL_STATUS		
	N	Y	Total
City	69 11.24 34.16 35.94	133 21.66 65.84 31.52	202 32.90
Town	54 8.79 23.18 28.13	179 29.15 76.82 42.42	233 37.95
Village	69 11.24 38.55 35.94	110 17.92 61.45 26.07	179 29.15
Total	192 31.27	422 68.73	614 100.00

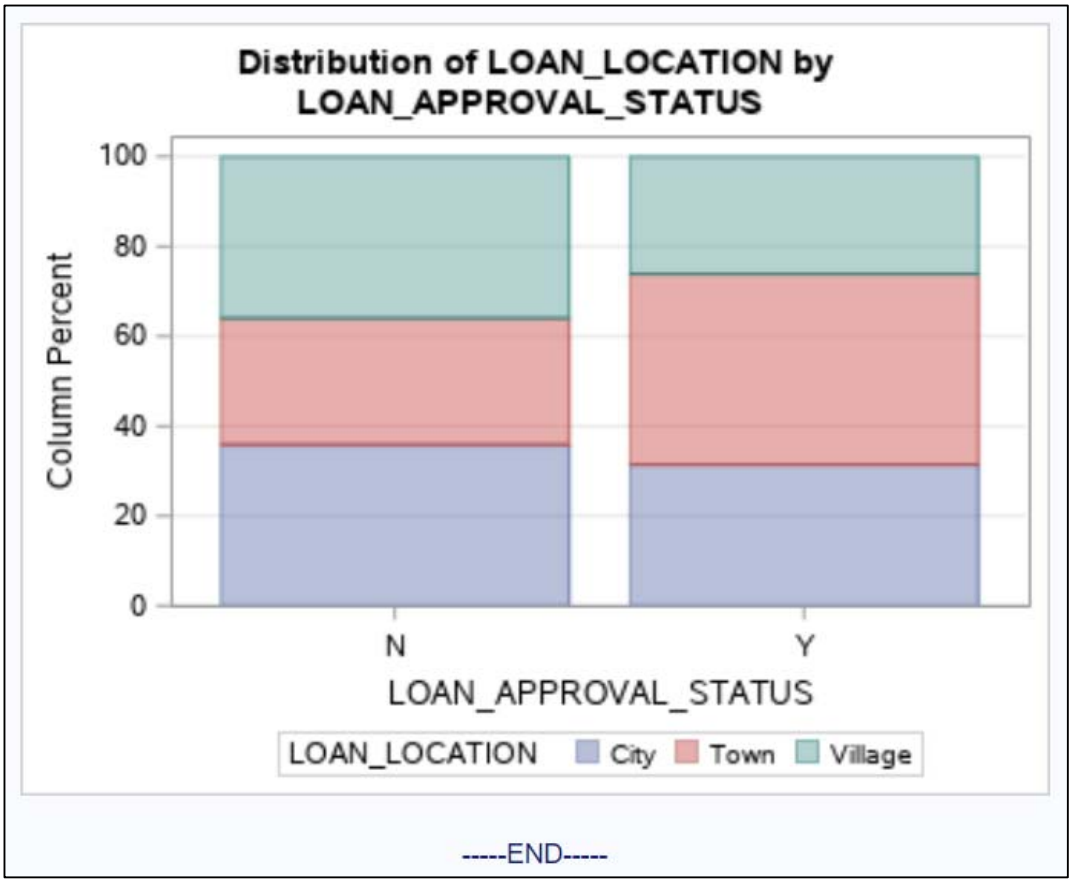


Figure 5.39: Graphical output of bivariate analysis for variables: LOAN_LOCATION versus LOAN_APPROVAL_STATUS

In overall, there are 422 approved loans (68.73%) which is greater than the 192 rejected loans (31.27%). Of the 611 applicants, there are 202 applicants from city (32.90%), 233 applicants from town (37.95%), and 179 applicants from village (29.15%). For the 202 applicants from city, there are 133 applicants with their loans approved (65.84%) while 69 applicants have their loans rejected (34.16%). For the 233 applicants from town, there are 179 applicants with their loans approved (76.82%) while 54 applicants have their loans rejected (23.18%). For the 179 applicants from village, there are 110 applicants with their loans approved (61.45%) while 69 applicants have their loans rejected (38.55%). It can be identified that applicants from town have the highest loan approval rate, followed by applicants from city and lastly applicants from village. This may be due to the applicants from town having a less business competitive environment as compared to the city while having more business opportunities as compared to the village. This provides a healthy environment for business growth leading to higher chance of business success. Thus, leading to higher rate of loan approval observed in applicants from town.

5.2.15 Bivariate Analysis of the Variables (LOAN_LOCATION – Categorical variable versus CANDIDATE_INCOME – Continuous variable)

Figure 5.40 shows the SAS code executed to produce the bivariate analysis for the categorical variable "LOAN_LOCATION" and the numerical variable "CANDIDATE_INCOME". The output from the code execution is shown in the table format as shown in Table 5.15 and in the graphical format as shown in Figure 5.41.

```
/****************** Bivariate Analysis - Categorical vs Continuous - LOAN_LOCATION vs CANDIDATE_INCOME *****************/
TITLE1 'Bivariate Analysis of variables:';
TITLE2 'LOAN_LOCATION - Categorical variables vs CANDIDATE_INCOME - Continuous variable';
FOOTNOTE '-----END-----';

PROC MEANS DATA = LIB65778.TRAINING_DS;

CLASS LOAN_LOCATION;    /* Categorical variable */
VAR CANDIDATE_INCOME;   /* Numeric variable */

RUN;

PROC SGPLOT DATA = LIB65778.TRAINING_DS;

VBOX CANDIDATE_INCOME / CATEGORY=LOAN_LOCATION;
/* LOAN_LOCATION -> X-Axis ; CANDIDATE_INCOME -> Y-Axis */
TITLE 'Bivariate Analysis of the variables: LOAN_LOCATION(Categorical variable) vs CANDIDATE_INCOME(Continuous variable)';

RUN;
```

Figure 5.40: SAS code for bivariate analysis of variables: LOAN_LOCATION versus CANDIDATE_INCOME

Table 5.15: Table output of bivariate analysis for variables: LOAN_LOCATION versus CANDIDATE_INCOME

Bivariate Analysis of variables:
LOAN_LOCATION - Categorical variables vs CANDIDATE_INCOME - Continuous variable

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

-----END-----

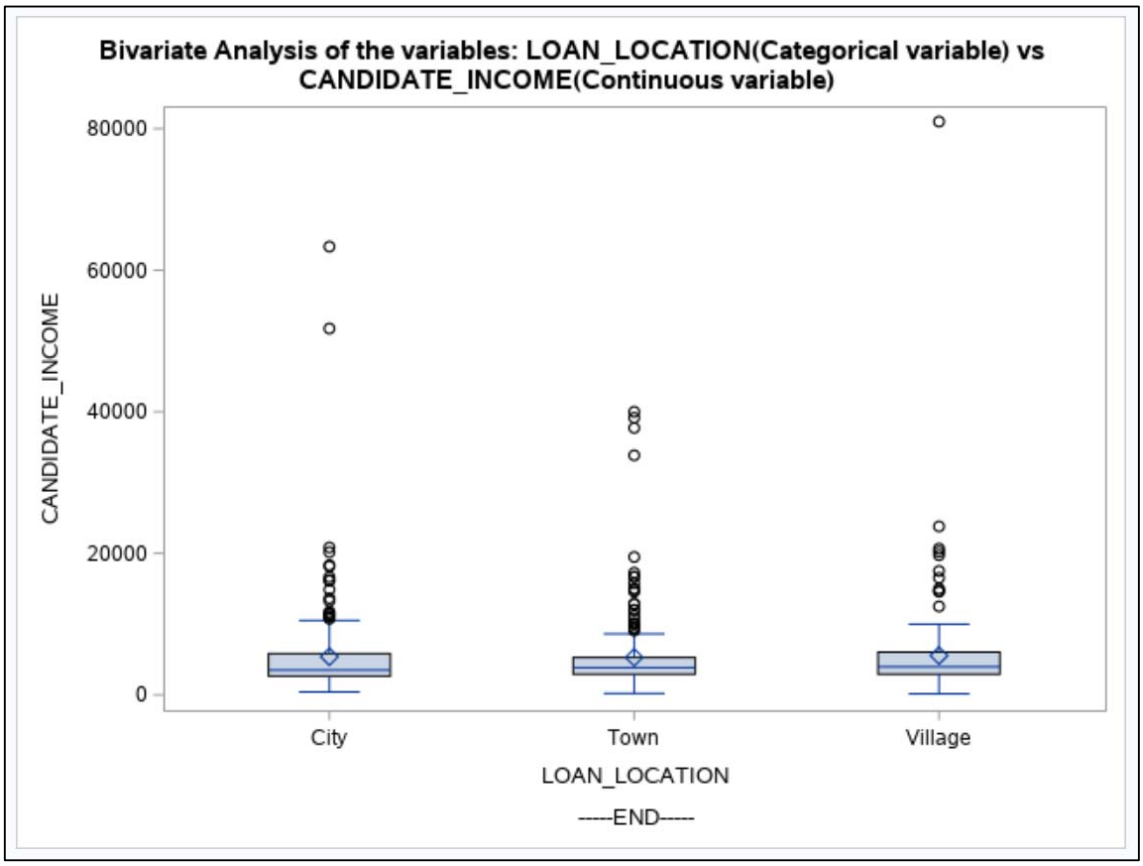


Figure 5.41: Graphical output of bivariate analysis for variables: LOAN_LOCATION versus CANDIDATE_INCOME

There are 202 applicants from city, 233 applicants from town, and 179 applicants from village. The average income of applicants from village is 5554.08 which is the highest, followed by applicants from city with an average income of 5398.25, and lastly applicants from the town have an average income of 5279.63. In general, applicants from the city should have a higher income bracket as compared to applicants from the village. However, based on Figure 5.41, it is observed that applicants from village have the highest standard deviation, followed by applicants from city and town. This may indicate potential outliers are present in the dataset for applicants from the village and city, causing a skew in the observations. It is advisable to investigate further on the extreme values to ensure veracity of data.

5.2.16 Bivariate Analysis of the Variables (LOAN_APPROVAL_STATUS – Categorical variable versus CANDIDATE_INCOME – Continuous variable)

Figure 5.42 shows the SAS code executed to produce the bivariate analysis for the categorical variable "LOAN_APPROVAL_STATUS" and the numerical variable "CANDIDATE_INCOME". The output from the code execution is shown in the table format as shown in Table 5.16 and in the graphical format as shown in Figure 5.43.

```
/***************** Bivariate Analysis - Categorical vs Continuous - LOAN_APPROVAL_STATUS vs CANDIDATE_INCOME *****************/
TITLE1 'Bivariate Analysis of variables:';
TITLE2 'LOAN_APPROVAL_STATUS - Categorical variables vs CANDIDATE_INCOME - Continuous variable';
FOOTNOTE '-----END-----';

PROC MEANS DATA = LIB65778.TRAINING_DS;

CLASS LOAN_APPROVAL_STATUS; /* Categorical variable */
VAR CANDIDATE_INCOME;   /* Numeric variable */

RUN;

PROC SGPLOT DATA = LIB65778.TRAINING_DS;

VBOX CANDIDATE_INCOME / CATEGORY=LOAN_APPROVAL_STATUS;
/* LOAN_LOCATION -> X-Axis ; CANDIDATE_INCOME -> Y-Axis */
TITLE 'Bivariate Analysis of the variables: LOAN_APPROVAL_STATUS(Categorical variable) vs CANDIDATE_INCOME(Continuous variable)';

RUN;
```

Figure 5.42: SAS code for bivariate analysis of variables: LOAN_APPROVAL_STATUS versus CANDIDATE_INCOME

Table 5.16: Table output of bivariate analysis for variables: LOAN_APPROVAL_STATUS versus CANDIDATE_INCOME

Bivariate Analysis of variables:
LOAN_APPROVAL_STATUS - Categorical variables vs CANDIDATE_INCOME - Continuous variable

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
LOAN_APPROVAL_STATUS	N Obs	N	Mean	Std Dev	Minimum	Maximum
N	192	192	5446.08	6819.56	150.0000000	81000.00
Y	422	422	5384.07	5765.44	210.0000000	63337.00

-----END-----

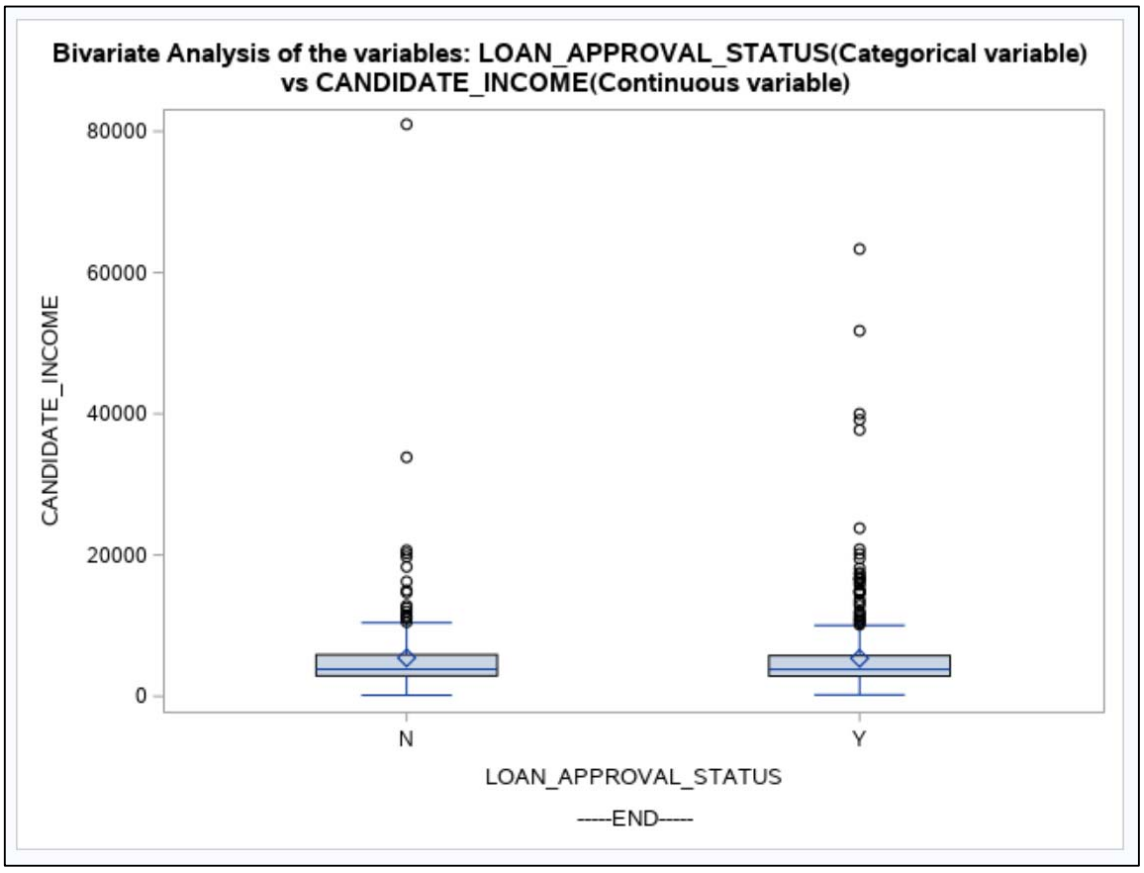


Figure 5.43: Graphical output of bivariate analysis for variables: LOAN_APPROVAL_STATUS versus CANDIDATE_INCOME

There are 422 applicants with loans approved and 192 applicants with loans rejected. The average income of applicants with loans approved is 5384.07 which is lower than the average income of applicants with loans rejected which is 5446.08. In general, applicants with a higher income should have a higher chance of getting their loans approved. However, based on Figure 5.43, it is observed that the applicants with loans rejected have a higher standard deviation than applicants with loans approved. This may indicate potential outliers present in the applicants with loans rejected, causing a skew in the observations. It is advisable to investigate further on the extreme values to ensure veracity of the data.

5.3 MISSING VALUES IMPUTATION – LIB65778.TRAINING_DS

This section documents the missing value imputation process for variables found in "TRAINING_DS" dataset. Missing values imputation will be performed on the categorical and numerical variables.

5.3.1 Missing Values Imputation in the Categorical Variable – MARITAL_STATUS

The following outline each step of the missing value imputation process for the categorical variable "MARITAL_STATUS". There are six steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TRAINING_DS

```
/********** Imputing missing values - Categorical variable: MARITAL_STATUS **********/
/* STEP 1: Make a copy of the dataset - LIB65778.TRAINING_DS */

PROC SQL;

CREATE TABLE LIB65778.TRAINING_DS_BK AS
SELECT *
FROM LIB65778.TRAINING_DS;

QUIT;
```

Figure 5.44: SAS code for making a dataset backup prior to imputation for variable: MARITAL_STATUS

Total rows: 614 Total columns: 13 — Rows 1-100

	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME
1	LP001002	Male	Not Married	0	Graduate	No	5849	0
2	LP001003	Male	Married	1	Graduate	No	4583	1508
3	LP001005	Male	Married	0	Graduate	Yes	3000	0
4	LP001006	Male	Married	0	Under Graduate	No	2583	2358
5	LP001008	Male	Not Married	0	Graduate	No	6000	0
6	LP001011	Male	Married	2	Graduate	Yes	5417	4196
7	LP001013	Male	Married	0	Under Graduate	No	2333	1516
8	LP001014	Male	Married	3+	Graduate	No	3036	2504
9	LP001018	Male	Married	2	Graduate	No	4006	1526
10	LP001020	Male	Married	1	Graduate	No	12841	10968

Figure 5.45: Backup dataset creation output

Figure 5.44 shows the SAS code executed to make a copy of the dataset for backup. The output from the code execution is shown in Figure 5.45 which shows the backup dataset. The dataset backup is an essential procedure before performing any manipulation to the dataset. This ensures the original dataset can be retrieved if the data manipulation process did not provide the outcome as expected.

STEP 2: Find the number of missing values in the variable – MARITAL_STATUS

```
/* STEP 2: Find the number of observations with missing values in the variable - MARITAL_STATUS */

TITLE1 'Find the number of observations with missing values in the variable - MARITAL_STATUS';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of observations'
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.MARITAL_STATUS IS MISSING ) OR
        ( t.MARITAL_STATUS EQ '' ) OR
        ( t.MARITAL_STATUS IS NULL ) );

QUIT;
```

Figure 5.46: SAS code for identifying quantity of missing values for variable: MARITAL_STATUS

Table 5.17: Output of quantity of missing values for variable: MARITAL_STATUS

Find the number of observations with missing values in the variable - MARITAL_STATUS

Number of observations
3

-----END-----

Figure 5.46 shows the SAS code executed to identify the number of missing values in the variable "MARITAL_STATUS". The output from the code execution is shown in Table 5.17. The number of observations with missing value is identified to be three.

STEP 3: Find the details of missing values in the variable – MARITAL_STATUS

```
/* STEP 3: Find the details of observations with missing values in the variable - MARITAL_STATUS */

TITLE1 'Find details';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.MARITAL_STATUS IS MISSING ) OR
        ( t.MARITAL_STATUS EQ '' ) OR
        ( t.MARITAL_STATUS IS NULL ) );

QUIT;
```

Figure 5.47: SAS code for identifying details of missing values for variable: MARITAL_STATUS

Table 5.18: Output of details of missing values for variable: MARITAL_STATUS

Find details of observations with missing values in the variable - MARITAL_STATUS

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—

Figure 5.47 shows the SAS code executed to identify the details of missing values in the variable "MARITAL_STATUS". The output from the code execution is shown in Table 5.18 which shows the three observations with the missing value identified. Based on the output, it is identified that the data in the cells of the "MARITAL_STATUS" variable is missing.

STEP 4: Create a temporary dataset to hold MARITAL_STATUS and number of applicants

```
/* STEP 4: Create a temporary dataset to hold MARITAL_STATUS and number of applicants */

PROC SQL;

CREATE TABLE LIB65778.TRAINING_DS_FI_MARITAL_STATUS AS
SELECT t.MARITAL_STATUS AS MARITAL_STATUS, COUNT(*) AS COUNTS
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.MARITAL_STATUS NE '' ) OR
        ( t.MARITAL_STATUS IS NOT NULL ) )
GROUP BY t.MARITAL_STATUS;

QUIT;
```

Figure 5.48: SAS code for temporary dataset creation for variable: MARITAL_STATUS

Total rows: 2 Total columns: 2

	MARITAL_STAT...	COUNTS
1	Married	398
2	Not Married	213

Figure 5.49: Output of temporary dataset for variable: MARITAL_STATUS

Figure 5.48 shows the SAS code executed to create a temporary dataset to contain the counts from each category of variable "MARITAL_STATUS". The output from the code execution is shown in Figure 5.49. Based on the output, it is identified that there are 398 data with the label "Married" and 213 data with the label "Not Married". Thus, the mode of this variable is identified to be the "Married" label.

A temporary dataset is utilized in this step to achieve optimal time and memory complexity of an algorithm. This is due to the need of repeatedly using the same results from the output in future programs. The use of the temporary dataset would only require the chunk of codes to be run once, and have the results saved for future use. This significantly reduced the time and memory required to run the main program.

STEP 5: Find the mode and impute the missing values found in the variable - MARITAL_STATUS

```
/* STEP 5: Find the MOD and impute the missing values found in the variable - MARITAL_STATUS */

PROC SQL;

UPDATE LIB65778.TRAINING_DS
SET marital_status = ( SELECT to.MARITAL_STATUS Label = 'M_STATUS'
                       FROM LIB65778.TRAINING_DS_FI_MARITAL_STATUS to
                       WHERE to.COUNTS EQ ( SELECT MAX(ti.COUNTS) Label = 'Highest Count'
                                            FROM LIB65778.TRAINING_DS_FI_MARITAL_STATUS ti ) )
                       /* Above is a sub-program to identify the Mode of marital_status */
WHERE ( ( marital_status IS MISSING ) OR
        ( marital_status EQ '') OR
        ( marital_status IS NULL ) );

QUIT;
```

Figure 5.50: SAS code for identifying the mode and perform missing value imputation for variable: MARITAL_STATUS

```
Errors, Warnings, Notes
Errors
Warnings
Notes (2)
71          UPDATE LIB65778.TRAINING_DS
72          SET marital_status = ( SELECT to.MARITAL_STATUS Label = 'M_STATUS'
73             FROM LIB65778.TRAINING_DS_FI_MARITAL_STATUS to
74             WHERE to.COUNTS EQ ( SELECT MAX(ti.COUNTS) Label = 'Highest Count'
75             FROM LIB65778.TRAINING_DS_FI_MARITAL_STATUS ti ) )
76             /* Above is a sub-program to identify the Mode of marital_status */
77          WHERE ( ( marital_status IS MISSING ) OR
78          ( marital_status EQ '') OR
79          ( marital_status IS NULL ) );
NOTE: 3 rows were updated in LIB65778.TRAINING_DS.
```

Figure 5.51: Output from imputation of variable: MARITAL_STATUS

Figure 5.50 shows the SAS code executed to identify the mode of the categorical variable "MARITAL_STATUS" and performing missing value imputation using the identified mode label. Figure 5.51 shows the output after a successful missing value imputation. It is identified that three rows of observations were imputed with the mode label.

STEP 6: (After imputation) find the number of observations with missing values in the variable – MARITAL_STATUS

```
/* STEP 6: (After imputation) Find the number of observations with missing values in the variable - MARITAL_STATUS */

TITLE1 'Find the number of observations with missing values in the variable - MARITAL_STATUS';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of observations'
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.MARITAL_STATUS IS MISSING ) OR
        ( t.MARITAL_STATUS EQ '' ) OR
        ( t.MARITAL_STATUS IS NULL ) );

QUIT;

/* STEP 7: (After imputation) Find the details of observations with missing values in the variable - MARITAL_STATUS */

TITLE1 'Find details';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.MARITAL_STATUS IS MISSING ) OR
        ( t.MARITAL_STATUS EQ '' ) OR
        ( t.MARITAL_STATUS IS NULL ) );

QUIT;
```

Figure 5.52: SAS code to identify missing value after imputation for variable: MARITAL_STATUS

Table 5.19: Output of missing value after imputation for variable: MARITAL_STATUS

Find the number of observations with missing values in the variable - MARITAL_STATUS

Number of observations
0

-----END-----

Find details

-----END-----

Figure 5.52 shows the SAS code executed to identify the quantity and details of missing values in the variable "MARITAL_STATUS" after imputation. The output from the code execution is shown in Table 5.19. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.3.2 Missing Values Imputation in the Categorical Variable – FAMILY_MEMBERS

The following outline each step of the missing value imputation process for the categorical variable "FAMILY_MEMBERS". There are six steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TRAINING_DS

```
/********** Imputing missing values - Categorical variable: FAMILY_MEMBERS **********/
/* STEP 1: Make a copy of the dataset - LIB65778.TRAINING_DS */

PROC SQL;

CREATE TABLE LIB65778.TRAINING_DS_BK AS
SELECT *
FROM LIB65778.TRAINING_DS;

QUIT;
```

Figure 5.53: SAS code for making a dataset backup prior to imputation for variable: FAMILY_MEMBERS

Figure 5.53 shows the SAS code executed to make a copy of the dataset for backup before the imputation procedure.

STEP 2: Find the number of missing values in the variable – FAMILY_MEMBERS

```
/* STEP 2: Find the number of observations with missing values in the variable - FAMILY_MEMBERS */

TITLE1 'Find the number of observations with missing values in the variable - FAMILY_MEMBERS';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of observations'
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.FAMILY_MEMBERS IS MISSING ) OR
        ( t.FAMILY_MEMBERS EQ '' ) OR
        ( t.FAMILY_MEMBERS IS NULL ) );

QUIT;
```

Figure 5.54: SAS code for identifying quantity of missing values for variable: FAMILY_MEMBERS

Table 5.20: Output of quantity of missing values for variable: FAMILY_MEMBERS

Find the number of observations with missing values in the variable - FAMILY_MEMBERS

Number of observations
15

-----END-----

Figure 5.54 shows the SAS code executed to identify the number of missing values in the variable "FAMILY_MEMBERS". The output from the code execution is shown in Table 5.20. The number of observations with missing value is identified to be 15.

STEP 3: Remove the '+' found in the variable - FAMILY_MEMBERS

```
434 /* STEP 3: Remove the '+' found in the variable: FAMILY_MEMBERS */
435
436 PROC SQL;
437
438 UPDATE LIB65778.TRAINING_DS
439 SET FAMILY_MEMBERS = SUBSTR(FAMILY_MEMBERS,1,1)
440 WHERE SUBSTR(FAMILY_MEMBERS,2,1) EQ '+';
441
442 QUIT;
```

Figure 5.55: SAS code for data manipulation for variable: FAMILY_MEMBERS

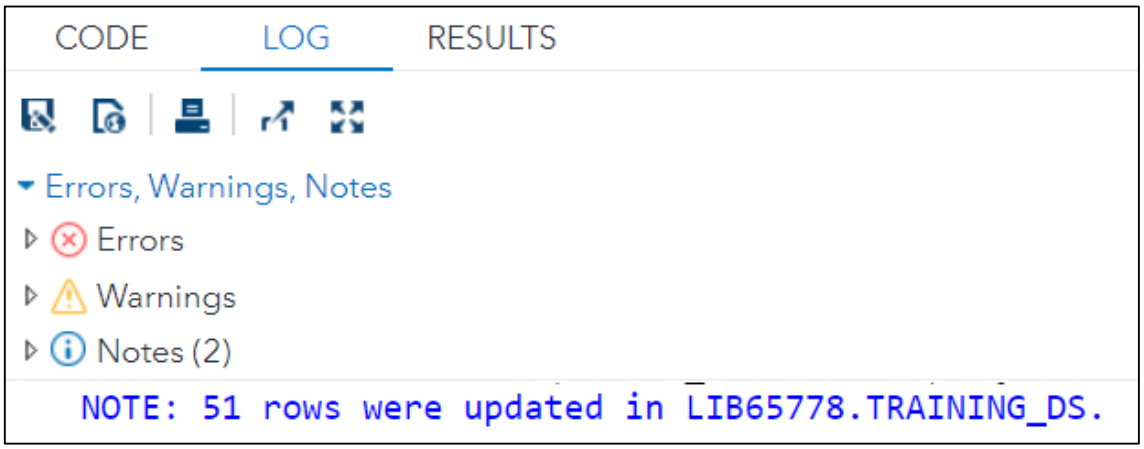


Figure 5.56: Output from data manipulation of variable: FAMILY_MEMBERS

Figure 5.55 shows the SAS code executed to manipulate the data in the variable "FAMILY_MEMBERS" to remove the "+" symbol from the data. The removal of the symbol would facilitate the data analysis process by converting the string into a numerical data type. The manipulation process is done by filtering the data that contain the symbol and retain only the first value of the entire string. The output from the data manipulation process is shown in Figure 5.56. It is identified that 51 observations have been manipulated.

STEP 4: Create a temporary dataset to hold FAMILY_MEMBERS and counts

```
444 /* STEP 4: Create a temporary dataset to hold FAMILY_MEMBERS and counts */
445
446 PROC SQL;
447
448 CREATE TABLE LIB65778.TRAINING_DS_FI_FAMILY_MEMBERS AS
449 SELECT t.FAMILY_MEMBERS AS FAMILY_MEMBERS,
450                COUNT(*) AS COUNTS
451 FROM LIB65778.TRAINING_DS t
452 WHERE ( ( t.FAMILY_MEMBERS NE '' ) OR
453         ( t.FAMILY_MEMBERS IS NOT NULL ) )
454 GROUP BY t.FAMILY_MEMBERS;
455
456 QUIT;
```

Figure 5.57: SAS code for temporary dataset creation for variable: FAMILY_MEMBERS

Total rows: 4 Total columns: 2

	FAMILY_MEMBERS	COUNTS
1	0	345
2	1	102
3	2	101
4	3	51

Figure 5.58: Output of temporary dataset for variable: FAMILY_MEMBERS

Figure 5.57 shows the SAS code executed to create a temporary dataset to contain the counts from each category of variable “FAMILY_MEMBERS”. The output from the code execution is shown in Figure 5.58. Based on the output, it is identified that there are 345 data with the label “0”, 102 data with the label “1”, 101 data with the label “2”, and 51 data with the label “3”. Thus, the mode of this variable is identified to be the “0” label.

STEP 5: Find the mode and imputing the missing values for variable - FAMILY_MEMBERS

```
458 /* STEP 5: Find the MOD and impute the missing values found in the variable: FAMILY_MEMBERS */
459
460 PROC SQL;
461
462 UPDATE LIB65778.TRAINING_DS
463 SET FAMILY_MEMBERS = ( SELECT (to.FAMILY_MEMBERS) Label = 'Family Member Category'
464                        FROM LIB65778.TRAINING_DS_FI_FAMILY_MEMBERS to
465                        WHERE to.counts EQ ( SELECT MAX(ti.counts) Label = 'Highest Counts'
466                                             FROM LIB65778.TRAINING_DS_FI_FAMILY_MEMBERS ti ) )
467                                             /* Above is a sub-program to identify the counts of the MOD */
468                                             /* and to identify the category of the MOD */
469 WHERE ( ( FAMILY_MEMBERS IS MISSING ) OR
470         ( FAMILY_MEMBERS EQ '' ) OR
471         ( FAMILY_MEMBERS IS NULL ) );
472
473 QUIT;
```

Figure 5.59: SAS code for identifying the mode and perform missing value imputation for variable: FAMILY_MEMBERS

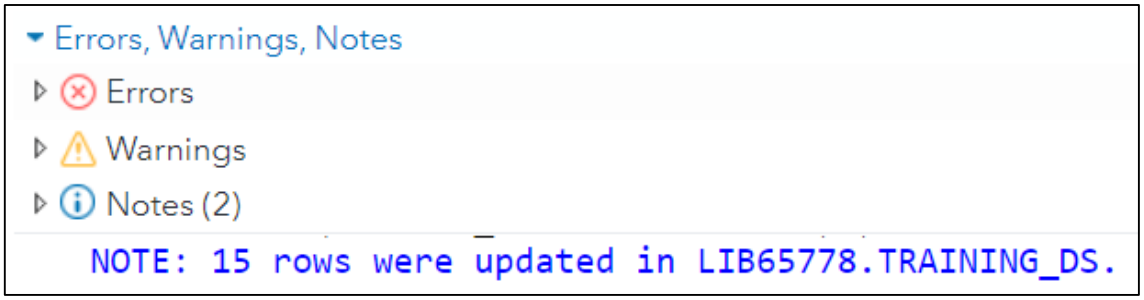


Figure 5.60: Output from imputation of variable: FAMILY_MEMBERS

Figure 5.59 shows the SAS code executed to identify the mode of the categorical variable "FAMILY_MEMBERS" and performing missing value imputation using the identified mode label. Figure 5.60 shows the output after a successful missing value imputation. It is identified that 15 rows of observations were imputed with the mode label.

STEP 6: (After imputation) find the number of observations with missing values in the variable – FAMILY_MEMBERS

```
/* STEP 6: (After imputation) Find the number of observations with missing values in the variable - FAMILY_MEMBERS */

TITLE1 'Find the number of observations with missing values in the variable - FAMILY_MEMBERS';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of observations'
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.FAMILY_MEMBERS IS MISSING ) OR
        ( t.FAMILY_MEMBERS EQ '' ) OR
        ( t.FAMILY_MEMBERS IS NULL ) );

QUIT;

/* STEP 7: (After imputation) Find the details of observations with missing values in the variable - FAMILY_MEMBERS */

TITLE1 'Find details';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.FAMILY_MEMBERS IS MISSING ) OR
        ( t.FAMILY_MEMBERS EQ '' ) OR
        ( t.FAMILY_MEMBERS IS NULL ) );

QUIT;
```

Figure 5.61: SAS code to identify missing value after imputation for variable: FAMILY_MEMBERS

Table 5.21: Output of missing value after imputation for variable: FAMILY_MEMBERS

Find the number of observations with missing values in the variable - FAMILY_MEMBERS

Number of observations
0

-----END-----

Find details

-----END-----

Figure 5.61 shows the SAS code executed to identify the quantity and details of missing values in the variable “FAMILY_MEMBERS” after imputation. The output from the code execution is shown in Table 5.21. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.3.3 Missing Values Imputation in the Categorical Variable – GENDER

The following outline each step of the missing value imputation process for the categorical variable "GENDER". There are six steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TRAINING_DS

```
505 /********** Imputing missing values - Categorical variable: GENDER **********/
506 /* STEP 1: Make a copy of the dataset - LIB65778.TRAINING_DS */
507
508 PROC SQL;
509
510 CREATE TABLE LIB65778.TRAINING_DS_BK AS
511 SELECT *
512 FROM LIB65778.TRAINING_DS;
513
514 QUIT;
```

Figure 5.62: SAS code for making a dataset backup prior to imputation for variable: GENDER

Figure 5.62 shows the SAS code executed to make a copy of the dataset for backup before the imputation procedure.

STEP 2: Find the number of missing values in the variable – GENDER

```
516 /* STEP 2: Find the number of observations with missing values in the variable - GENDER */
517
518 TITLE1 'Find the number of observations with missing values in the variable - GENDER';
519 FOOTNOTE '-----END-----';
520
521 PROC SQL;
522
523 SELECT COUNT(*) Label = 'Number of observations'
524 FROM LIB65778.TRAINING_DS t
525 WHERE ( ( t.GENDER IS MISSING ) OR
526         ( t.GENDER EQ '' ) OR
527         ( t.GENDER IS NULL ) );
528
529 QUIT;
```

Figure 5.63: SAS code for identifying quantity of missing values for variable: GENDER

Table 5.22: Output of quantity of missing values for variable: GENDER

Find the number of observations with missing values in the variable - GENDER

Number of observations
13

-----END-----

Figure 5.63 shows the SAS code executed to identify the number of missing values in the variable "GENDER". The output from the code execution is shown in Table 5.22. The number of observations with missing value is identified to be 13.

STEP 3: Find the details of missing values in the variable – GENDER

```
/* STEP 3: Find the details of observations with missing values in the variable - GENDER */

TITLE1 'Find the details of observations with missing values in the variable - GENDER';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.GENDER IS MISSING ) OR
        ( t.GENDER EQ '' ) OR
        ( t.GENDER IS NULL ) );

QUIT;
```

Figure 5.64: SAS code for identifying details of missing values for variable: GENDER

Table 5.23: Output of details of missing values for variable: GENDER

Find the details of observations with missing values in the variable - GENDER

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-----

Figure 5.64 shows the SAS code executed to identify the details of missing values in the variable "GENDER". The output from the code execution is shown in Table 5.23 which shows the 13 observations with the missing value identified. Based on the output, it is identified that the data in the cells of the "GENDER" variable is missing.

STEP 4: Create a temporary dataset to hold GENDER and number of applicants

```
546 /* STEP 4: Create a temporary dataset to hold GENDER category and counts */
547
548 PROC SQL;
549
550 CREATE TABLE LIB65778.TRAINING_FI_GENDER_DS AS
551 SELECT t.gender AS Gender,
552        COUNT(*) AS COUNTS
553 FROM LIB65778.TRAINING_DS t
554 WHERE ( ( t.GENDER NE '' ) OR
555         ( t.GENDER IS NOT NULL ) )
556 GROUP BY t.gender;
557
558 QUIT;
```

Figure 5.65: SAS code for temporary dataset creation for variable: GENDER

Total rows: 2 Total columns: 2

	Gender	COUNTS
1	Female	112
2	Male	489

Figure 5.66: Output of temporary dataset for variable: GENDER

Figure 5.65 shows the SAS code executed to create a temporary dataset to contain the counts from each category of variable "GENDER". The output from the code execution is shown in Figure 5.66. Based on the output, it is identified that there are 112 data with the label "Female" and 489 data with the label "Male". Thus, the mode of this variable is identified to be the "Male" label.

STEP 5: Find the mode and impute the missing values found in the variable - GENDER

```
560 /* STEP 5: Find the MOD and impute the missing values found in the variable - GENDER */
561
562 PROC SQL;
563
564 UPDATE LIB65778.TRAINING_DS
565 SET GENDER = ( SELECT to.GENDER Label = 'Gender'
566                FROM LIB65778.TRAINING_FI_GENDER_DS to
567                WHERE to.counts EQ ( SELECT MAX(ti.counts) Label = 'Highest Counts'
568                                     FROM LIB65778.TRAINING_FI_GENDER_DS ti ) )
569                                     /* Above is a sub-program to identify the Mode of GENDER */
570                                     /* and to identify the category of the MOD */
571 WHERE ( ( GENDER IS MISSING ) OR
572         ( GENDER EQ '' ) OR
573         ( GENDER IS NULL ) );
574
575 QUIT;
```

Figure 5.67: SAS code for identifying the mode and perform missing value imputation for variable: GENDER

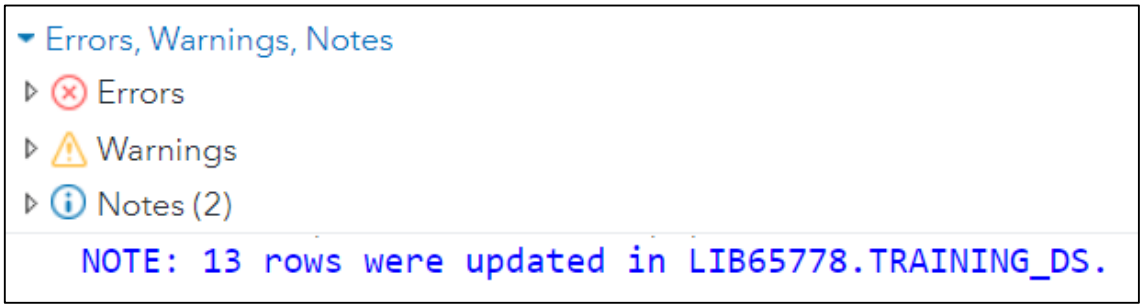


Figure 5.68: Output from imputation of variable: GENDER

Figure 5.67 shows the SAS code executed to identify the mode of the categorical variable "GENDER" and performing missing value imputation using the identified mode label. Figure 5.68 shows the output after a successful missing value imputation. It is identified that 13 rows of observations were imputed with the mode label.

STEP 6: (After imputation) find the number of observations with missing values in the variable – GENDER

```
577 /* STEP 6: (After imputation) Find the number of observations with missing values in the variable - GENDER */
578
579 TITLE1 'Find the number of observations with missing values in the variable - GENDER';
580 FOOTNOTE '-----END-----';
581
582 PROC SQL;
583
584 SELECT COUNT(*) Label = 'Number of observations'
585 FROM LIB65778.TRAINING_DS t
586 WHERE ( ( t.GENDER IS MISSING ) OR
587         ( t.GENDER EQ '' ) OR
588         ( t.GENDER IS NULL ) );
589
590 QUIT;
591
592 /* STEP 7: (After imputation) Find the details of observations with missing values in the variable - GENDER */
593
594 TITLE1 'Find details';
595 FOOTNOTE '-----END-----';
596
597 PROC SQL;
598
599 SELECT *
600 FROM LIB65778.TRAINING_DS t
601 WHERE ( ( t.GENDER IS MISSING ) OR
602         ( t.GENDER EQ '' ) OR
603         ( t.GENDER IS NULL ) );
604
605 QUIT;
```

Figure 5.69: SAS code to identify missing value after imputation for variable: GENDER

Table 5.24: Output of missing value after imputation for variable: GENDER

Find the number of observations with missing values in the variable - GENDER

Number of observations
0

-----END-----

Find details

-----END-----

Figure 5.69 shows the SAS code executed to identify the quantity and details of missing values in the variable "GENDER" after imputation. The output from the code execution is shown in Table 5.24. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.3.4 Missing Values Imputation in the Categorical Variable – EMPLOYMENT

The following outline each step of the missing value imputation process for the categorical variable "EMPLOYMENT". There are six steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TRAINING_DS

```
607 /********** Imputing missing values - Categorical variable: EMPLOYMENT **********/
608 /* STEP 1: Make a copy of the dataset - LIB65778.TRAINING_DS */
609
610 PROC SQL;
611
612 CREATE TABLE LIB65778.TRAINING_DS_BK AS
613 SELECT *
614 FROM LIB65778.TRAINING_DS;
615
616 QUIT;
```

Figure 5.70: SAS code for making a dataset backup prior to imputation for variable: EMPLOYMENT

Figure 5.70 shows the SAS code executed to make a copy of the dataset for backup before the imputation procedure.

STEP 2: Find the number of missing values in the variable – EMPLOYMENT

```
618 /* STEP 2: Find the number of observations with missing values in the variable - EMPLOYMENT */
619
620 TITLE1 'Find the number of observations with missing values in the variable - EMPLOYMENT';
621 FOOTNOTE '-----END-----';
622
623 PROC SQL;
624
625 SELECT COUNT(*) Label = 'Number of observations'
626 FROM LIB65778.TRAINING_DS t
627 WHERE ( ( t.EMPLOYMENT IS MISSING ) OR
628         ( t.EMPLOYMENT EQ '' ) OR
629         ( t.EMPLOYMENT IS NULL ) );
630
631 QUIT;
```

Figure 5.71: SAS code for identifying quantity of missing values for variable: EMPLOYMENT

Table 5.25: Output of quantity of missing values for variable: EMPLOYMENT

Find the number of observations with missing values in the variable - EMPLOYMENT

Number of observations
32

-----END-----

Figure 5.71 shows the SAS code executed to identify the number of missing values in the variable "EMPLOYMENT". The output from the code execution is shown in Table 5.25. The number of observations with missing value is identified to be 32.

STEP 3: Find the details of missing values in the variable – EMPLOYMENT

```
633 /* STEP 3: Find the details of observations with missing values in the variable - EMPLOYMENT */
634
635 TITLE1 'Find the details of observations with missing values in the variable - EMPLOYMENT';
636 FOOTNOTE '-----END-----';
637
638 PROC SQL;
639
640 SELECT *
641 FROM LIB65778.TRAINING_DS t
642 WHERE ( ( t.EMPLOYMENT IS MISSING ) OR
643         ( t.EMPLOYMENT EQ '' ) OR
644         ( t.EMPLOYMENT IS NULL ) );
645
646 QUIT;
```

Figure 5.72: SAS code for identifying details of missing values for variable: EMPLOYMENT

Table 5.26: Output of details of missing values for variable: EMPLOYMENT

Find the details of observations with missing values in the variable - EMPLOYMENT

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
LP002435	Male	Married	0	Graduate		3539	1376	55	360	1	Village	N
LP002489	Female	Not Married	1	Under Graduate		5191	0	132	360	1	Town	Y
LP002502	Female	Married	2	Under Graduate		210	2917	98	360	1	Town	Y
LP002732	Male	Not Married	0	Under Graduate		2550	2042	126	360	1	Village	Y
LP002753	Female	Not Married	1	Graduate		3652	0	95	360	1	Town	Y
LP002888	Male	Not Married	0	Graduate		3182	2917	161	360	1	City	Y
LP002949	Female	Not Married	3	Graduate		416	41667	350	180	.	City	N
LP002950	Male	Married	0	Under Graduate		2894	2792	155	360	1	Village	Y

---END---

Figure 5.72 shows the SAS code executed to identify the details of missing values in the variable “EMPLOYMENT”. The output from the code execution is shown in Table 5.26 which shows 32 observations with the missing value identified. Based on the output, it is identified that the data in the cells of the “EMPLOYMENT” variable is missing.

STEP 4: Create a temporary dataset to hold EMPLOYMENT and number of applicants

```
/* STEP 4: Create a temporary dataset to hold EMPLOYMENT category and counts */

PROC SQL;

CREATE TABLE LIB65778.TRAINING_FI_EMPLOYMENT_DS AS
SELECT t.EMPLOYMENT AS Employment,
          COUNT(*) AS COUNTS
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.EMPLOYMENT NE '' ) OR
        ( t.EMPLOYMENT IS NOT NULL ) )
GROUP BY t.EMPLOYMENT;

QUIT;
```

Figure 5.73: SAS code for temporary dataset creation for variable: EMPLOYMENT

Total rows: 2 Total columns: 2

	Employment	COUNTS
1	No	500
2	Yes	82

Figure 5.74: Output of temporary dataset for variable: EMPLOYMENT

Figure 5.73 shows the SAS code executed to create a temporary dataset to contain the counts from each category of variable "EMPLOYMENT". The output from the code execution is shown in Figure 5.74. Based on the output, it is identified that there are 500 data with the label "No" and 82 data with the label "Yes". Thus, the mode of this variable is identified to be the "No" label.

STEP 5: Find the mode and impute the missing values found in the variable - EMPLOYMENT

```
662 /* STEP 5: Find the MOD and impute the missing values found in the variable - EMPLOYMENT */
663
664 PROC SQL;
665
666 UPDATE LIB65778.TRAINING_DS
667 SET EMPLOYMENT = ( SELECT to.EMPLOYMENT Label = 'Employment'
668                    FROM LIB65778.TRAINING_FI_EMPLOYMENT_DS to
669                    WHERE to.counts EQ ( SELECT MAX(ti.counts) Label = 'Highest Counts'
670                                         FROM LIB65778.TRAINING_FI_EMPLOYMENT_DS ti ) )
671                                         /* Above is a sub-program to identify the Mode of EMPLOYMENT */
672                                         /* and to identify the category of the MOD */
673 WHERE ( ( EMPLOYMENT IS MISSING ) OR
674         ( EMPLOYMENT EQ '' ) OR
675         ( EMPLOYMENT IS NULL ) );
676
677 QUIT;
```

Figure 5.75: SAS code for identifying the mode and perform missing value imputation for variable: EMPLOYMENT

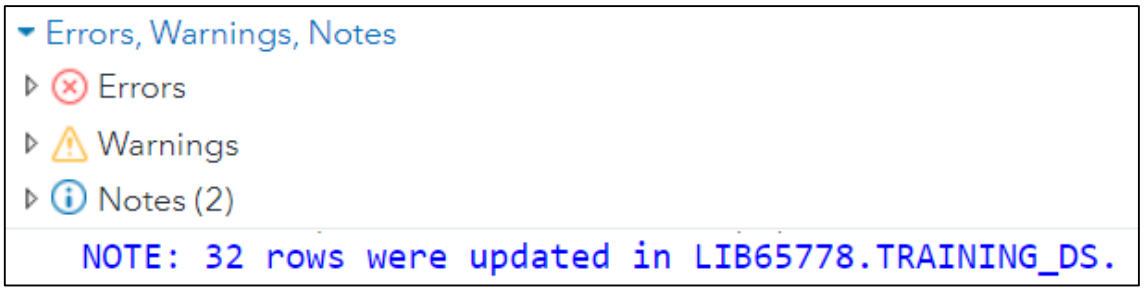


Figure 5.76: Output from imputation of variable: EMPLOYMENT

Figure 5.75 shows the SAS code executed to identify the mode of the categorical variable "EMPLOYMENT" and performing missing value imputation using the identified mode label. Figure 5.76 shows the output after a successful missing value imputation. It is identified that 32 rows of observations were imputed with the mode label.

STEP 6: (After imputation) find the number of observations with missing values in the variable – EMPLOYMENT

```
679 /* STEP 6: (After imputation) Find the number of observations with missing values in the variable - EMPLOYMENT */
680
681 TITLE1 'Find the number of observations with missing values in the variable - EMPLOYMENT';
682 FOOTNOTE '-----END-----';
683
684 PROC SQL;
685
686 SELECT COUNT(*) Label = 'Number of observations'
687 FROM LIB65778.TRAINING_DS t
688 WHERE ( ( t.EMPLOYMENT IS MISSING ) OR
689         ( t.EMPLOYMENT EQ '' ) OR
690         ( t.EMPLOYMENT IS NULL ) );
691
692 QUIT;
693
694 /* STEP 7: (After imputation) Find the details of observations with missing values in the variable - EMPLOYMENT */
695
696 TITLE1 'Find details';
697 FOOTNOTE '-----END-----';
698
699 PROC SQL;
700
701 SELECT *
702 FROM LIB65778.TRAINING_DS t
703 WHERE ( ( t.EMPLOYMENT IS MISSING ) OR
704         ( t.EMPLOYMENT EQ '' ) OR
705         ( t.EMPLOYMENT IS NULL ) );
706
707 QUIT;
```

Figure 5.77: SAS code to identify missing value after imputation for variable: EMPLOYMENT

Table 5.27: Output of missing value after imputation for variable: EMPLOYMENT

Find the number of observations with missing values in the variable - EMPLOYMENT

Number of observations
0

-----END-----

Find details

-----END-----

Figure 5.77 shows the SAS code executed to identify the quantity and details of missing values in the variable "EMPLOYMENT" after imputation. The output from the code execution is shown in Table 5.27. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.3.5 Missing Values Imputation in the Categorical Variable – LOAN_HISTORY

The following outline each step of the missing value imputation process for the categorical variable "LOAN_HISTORY". There are six steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TRAINING_DS

```
/********** Imputing missing values - Categorical variable: LOAN_HISTORY **********/
/* STEP 1: Make a copy of the dataset - LIB65778.TRAINING_DS */

PROC SQL;

CREATE TABLE LIB65778.TRAINING_DS_BK AS
SELECT *
FROM LIB65778.TRAINING_DS;

QUIT;
```

Figure 5.78: SAS code for making a dataset backup prior to imputation for variable: LOAN_HISTORY

Figure 5.78 shows the SAS code executed to make a copy of the dataset for backup before the imputation procedure.

STEP 2: Find the number of missing values in the variable – LOAN_HISTORY

```
/* STEP 2: Find the number of observations with missing values in the variable - LOAN_HISTORY */

TITLE1 'Find the number of observations with missing values in the variable - LOAN_HISTORY';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of observations'
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_HISTORY EQ . ) OR
        ( t.LOAN_HISTORY IS NULL ) );

QUIT;
```

Figure 5.79: SAS code for identifying quantity of missing values for variable: LOAN_HISTORY

Table 5.28: Output of quantity of missing values for variable: LOAN_HISTORY

Find the number of observations with missing values in the variable - LOAN_HISTORY

Number of observations
50

-----END-----

Figure 5.79 shows the SAS code executed to identify the number of missing values in the variable "LOAN_HISTORY". The output from the code execution is shown in Table 5.28. The number of observations with missing value is identified to be 50.

STEP 3: Find the details of missing values in the variable – LOAN_HISTORY

```
734 /* STEP 3: Find the details of observations with missing values in the variable - LOAN_HISTORY */
735
736 TITLE1 'Find the details of observations with missing values in the variable - LOAN_HISTORY';
737 FOOTNOTE '-----END-----';
738
739 PROC SQL;
740
741 SELECT *
742 FROM LIB65778.TRAINING_DS t
743 WHERE ( ( t.LOAN_HISTORY EQ . ) OR
744         ( t.LOAN_HISTORY IS NULL ) );
745
746 QUIT;
```

Figure 5.80: SAS code for identifying details of missing values for variable: LOAN_HISTORY

Table 5.29: Output of details of missing values for variable: LOAN_HISTORY

Find the details of observations with missing values in the variable - LOAN_HISTORY

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	No	3717	2925	151	360	.	Town	N
LP001091	Male	Married	1	Graduate	No	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	No	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
LP001786	Male	Married	0	Graduate	No	5746	0	255	360	.	City	N
LP001788	Female	Not Married	0	Graduate	Yes	3463	0	122	360	.	City	Y
LP001864	Male	Married	3	Under Graduate	No	4931	0	128	360	.	Town	N
LP001865	Male	Married	1	Graduate	No	6083	4250	330	360	.	City	Y
LP001908	Female	Married	0	Under Graduate	No	4100	0	124	360	.	Village	Y
LP001998	Male	Married	2	Under Graduate	No	7667	0	185	360	.	Village	Y
LP002008	Male	Married	2	Graduate	Yes	5746	0	144	84	.	Village	Y
LP002036	Male	Married	0	Graduate	No	2058	2134	88	360	.	City	Y
LP002043	Female	Not Married	1	Graduate	No	3541	0	112	360	.	Town	Y

Figure 5.80 shows the SAS code executed to identify the details of missing values in the variable "LOAN_HISTORY". The output from the code execution is shown in Table 5.29 which shows the partial output observations with the missing value identified due to the length of the output. Based on the output, it is identified that the data in the cells of the "LOAN_HISTORY" variable is missing.

STEP 4: Create a temporary dataset to hold LOAN_HISTORY and number of applicants

```
748 /* STEP 4: Create a temporary dataset to hold LOAN_HISTORY category and counts */
749
750 PROC SQL;
751
752 CREATE TABLE LIB65778.TRAINING_FI_LOAN_HISTORY_DS AS
753 SELECT t.LOAN_HISTORY AS LoanHistory,
754              COUNT(*) AS COUNTS
755 FROM LIB65778.TRAINING_DS t
756 WHERE ( ( t.LOAN_HISTORY NE . ) OR
757         ( t.LOAN_HISTORY IS NOT NULL ) )
758 GROUP BY t.LOAN_HISTORY;
759
760 QUIT;
```

Figure 5.81: SAS code for temporary dataset creation for variable: LOAN_HISTORY

Total rows: 2 Total columns: 2

	LoanHistory	COUNTS
1	0	89
2	1	475

Figure 5.82: Output of temporary dataset for variable: LOAN_HISTORY

Figure 5.81 shows the SAS code executed to create a temporary dataset to contain the counts from each category of variable "LOAN_HISTORY". The output from the code execution is shown in Figure 5.82. Based on the output, it is identified that there are 89 data with the label "0" and 475 data with the label "1". Thus, the mode of this variable is identified to be the "1" label.

STEP 5: Find the mode and impute the missing values found in the variable – LOAN_HISTORY

```
762 /* STEP 5: Find the MOD and impute the missing values found in the variable - LOAN_HISTORY */
763
764 PROC SQL;
765
766 UPDATE LIB65778.TRAINING_DS
767 SET LOAN_HISTORY = ( SELECT to.LoanHistory Label = 'Loan History'
768                      FROM LIB65778.TRAINING_FI_LOAN_HISTORY_DS to
769                      WHERE to.counts EQ ( SELECT MAX(ti.counts) Label = 'Highest Counts'
770                                           FROM LIB65778.TRAINING_FI_LOAN_HISTORY_DS ti ) )
771                                           /* Above is a sub-program to identify the Mode of LOAN_HISTORY */
772                                           /* and to identify the category of the MOD */
773 WHERE ( ( LOAN_HISTORY IS MISSING ) OR
774         ( LOAN_HISTORY EQ . ) OR
775         ( LOAN_HISTORY IS NULL ) );
776
777 QUIT;
```

Figure 5.83: SAS code for identifying the mode and perform missing value imputation for variable: LOAN_HISTORY

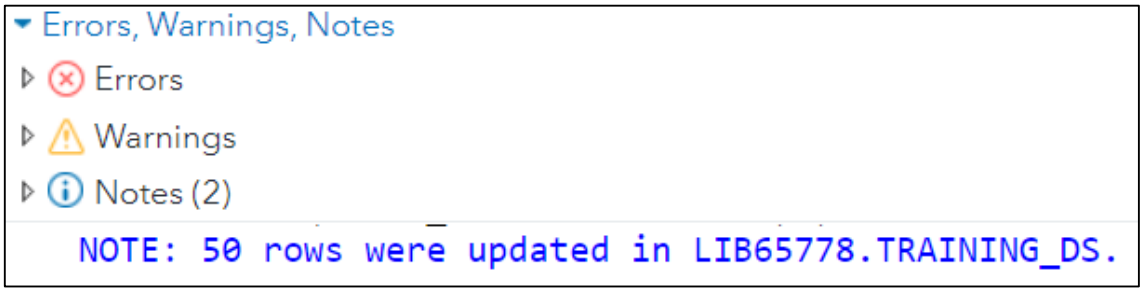


Figure 5.84: Output from imputation of variable: LOAN_HISTORY

Figure 5.83 shows the SAS code executed to identify the mode of the categorical variable "LOAN_HISTORY" and performing missing value imputation using the identified mode label. Figure 5.84 shows the output after a successful missing value imputation. It is identified that 50 rows of observations were imputed with the mode label.

STEP 6: (After imputation) find the number of observations with missing values in the variable – LOAN_HISTORY

```
/* STEP 6: (After imputation) Find the number of observations with missing values in the variable - LOAN_HISTORY */

TITLE1 'Find the number of observations with missing values in the variable - LOAN_HISTORY';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of observations'
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_HISTORY IS MISSING ) OR
        ( t.LOAN_HISTORY EQ . ) OR
        ( t.LOAN_HISTORY IS NULL ) );

QUIT;

/* STEP 7: (After imputation) Find the details of observations with missing values in the variable - LOAN_HISTORY */

TITLE1 'Find details';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_HISTORY IS MISSING ) OR
        ( t.LOAN_HISTORY EQ .) OR
        ( t.LOAN_HISTORY IS NULL ) );

QUIT;
```

Figure 5.85: SAS code to identify missing value after imputation for variable: LOAN_HISTORY

Table 5.30: Output of missing value after imputation for variable: LOAN_HISTORY

Find the number of observations with missing values in the variable - LOAN_HISTORY

Number of observations
0

-----END-----

Find details

-----END-----

Figure 5.85 shows the SAS code executed to identify the quantity and details of missing values in the variable "LOAN_HISTORY" after imputation. The output from the code execution is shown in Table 5.30. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.3.6 Missing Values Imputation in the Numerical Variable – LOAN_AMOUNT

The following outline each step of the missing value imputation process for the numerical variable "LOAN_AMOUNT". There are five steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TRAINING_DS

```
/********** Imputing missing values - Numerical variable: LOAN_AMOUNT **********/
/* STEP 1: Make a copy of the dataset - LIB65778.TRAINING_DS */

PROC SQL;

CREATE TABLE LIB65778.TRAINING_DS_BK AS
SELECT *
FROM LIB65778.TRAINING_DS;

QUIT;
```

Figure 5.86: SAS code for making a dataset backup prior to imputation for variable: LOAN_AMOUNT

Figure 5.86 shows the SAS code executed to make a copy of the dataset for backup before the imputation procedure.

STEP 2: Find the number of missing values in the variable – LOAN_AMOUNT

```
/* STEP 2: Find the number of observations with missing values in the variable - LOAN_AMOUNT */

TITLE1 'Find the number of observations with missing values in the variable - LOAN_AMOUNT';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of Observations'
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_AMOUNT EQ . ) OR
        ( t.LOAN_AMOUNT IS NULL ) OR
        ( t.LOAN_AMOUNT IS MISSING ) );

QUIT;
```

Figure 5.87: SAS code for identifying quantity of missing values for variable: LOAN_AMOUNT

Table 5.31: Output of quantity of missing values for variable: LOAN_AMOUNT

Find the number of observations with missing values in the variable - LOAN_AMOUNT

Number of Observations
22

-----END-----

Figure 5.87 shows the SAS code executed to identify the number of missing values in the variable "LOAN_AMOUNT". The output from the code execution is shown in Table 5.31. The number of observations with missing value is identified to be 22.

STEP 3: Find the details of missing values in the variable – LOAN_AMOUNT

```
/* STEP 3: Find the details of observations with missing values in the variable - LOAN_AMOUNT */

TITLE1 'Find the details of observations with missing values in the variable - LOAN_AMOUNT';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_AMOUNT EQ . ) OR
        ( t.LOAN_AMOUNT IS NULL ) OR
        ( t.LOAN_AMOUNT IS MISSING ) );

QUIT;
```

Figure 5.88: SAS code for identifying details of missing values for variable: LOAN_AMOUNT

Table 5.32: Output of details of missing values for variable: LOAN_AMOUNT

Find the details of observations with missing values in the variable - LOAN_AMOUNT

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	No	6782	0		360	1	City	N
LP001350	Male	Married	0	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	Married	0	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-----

Figure 5.88 shows the SAS code executed to identify the details of missing values in the variable "LOAN_AMOUNT". The output from the code execution is shown in Table 5.32 which shows the 22 observations with the missing value identified. Based on the output, it is identified that the data in the cells of the "LOAN_AMOUNT" variable is missing.

STEP 4: Imputing missing values found in the variable – LOAN_AMOUNT

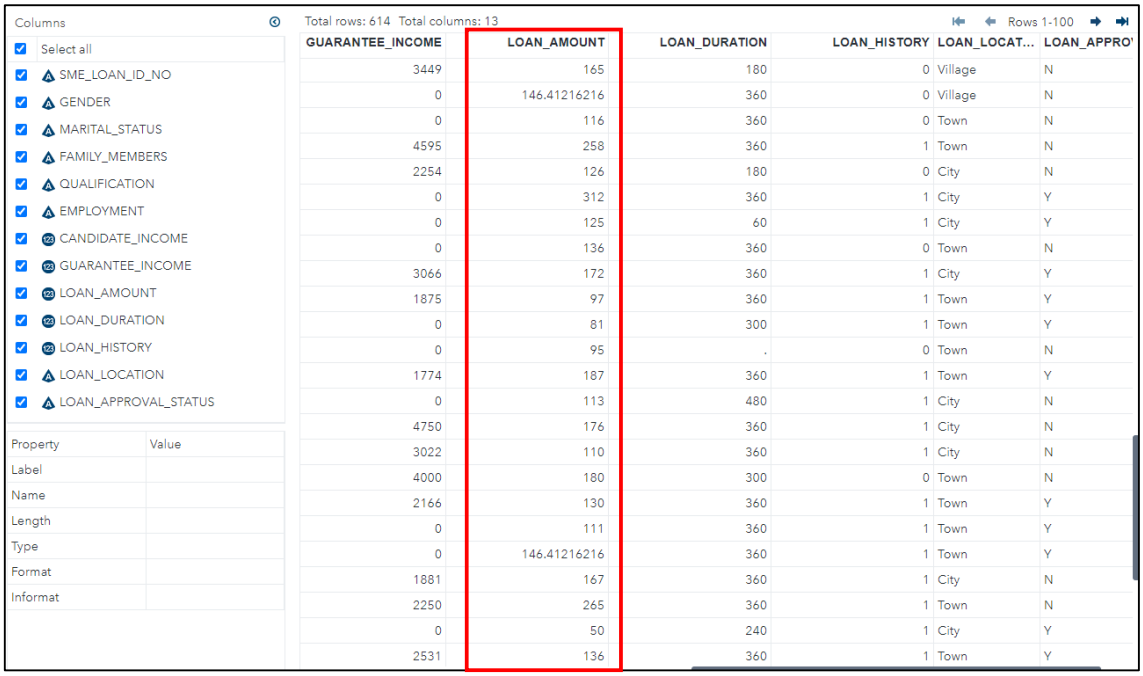
```
/* STEP 4: Imputing missing values found in the variable - LOAN_AMOUNT */

PROC STDIZE DATA=LIB65778.TRAINING_DS REPONLY

METHOD=MEAN OUT=LIB65778.TRAINING_DS;
VAR LOAN_AMOUNT;

QUIT;
```

Figure 5.89: SAS code for performing missing value imputation for variable: LOAN_AMOUNT



GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCAT...	LOAN_APPRO
3449	165	180	0	Village	N
0	146.41216216	360	0	Village	N
0	116	360	0	Town	N
4595	258	360	1	Town	N
2254	126	180	0	City	N
0	312	360	1	City	Y
0	125	60	1	City	Y
0	136	360	0	Town	N
3066	172	360	1	City	Y
1875	97	360	1	Town	Y
0	81	300	1	Town	Y
0	95	.	0	Town	N
1774	187	360	1	Town	Y
0	113	480	1	City	N
4750	176	360	1	City	N
3022	110	360	1	City	N
4000	180	300	0	Town	N
2166	130	360	1	Town	Y
0	111	360	1	Town	Y
0	146.41216216	360	1	Town	Y
1881	167	360	1	City	N
2250	265	360	1	Town	N
0	50	240	1	City	Y
2531	136	360	1	Town	Y

Figure 5.90: Output from imputation of variable: LOAN_AMOUNT

Figure 5.89 shows the SAS code executed to perform missing value imputation for the variable "LOAN_AMOUNT". The code execution replaces missing value with the mean value of the variable. Figure 5.90 shows the output after a successful missing value imputation procedure. It is identified that the missing value cells are filled with the mean value of the variable.

STEP 5: (After imputation) find the number of observations with missing values in the variable – LOAN_AMOUNT

```
/* STEP 5: (After imputation) Find the number of observations with missing values in the variable - LOAN_AMOUNT */

TITLE1 'Find the number of observations with missing values in the variable - LOAN_AMOUNT';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of Observations'
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_AMOUNT EQ . ) OR
        ( t.LOAN_AMOUNT IS NULL ) OR
        ( t.LOAN_AMOUNT IS MISSING ) );

QUIT;

/* STEP 6: (After imputation) Find the details of observations with missing values in the variable - LOAN_AMOUNT */

TITLE1 'Find the details of observations with missing values in the variable - LOAN_AMOUNT';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_AMOUNT EQ . ) OR
        ( t.LOAN_AMOUNT IS NULL ) OR
        ( t.LOAN_AMOUNT IS MISSING ) );

QUIT;
```

Figure 5.91: SAS code to identify missing value after imputation for variable: LOAN_AMOUNT

Table 5.33: Output of missing value after imputation for variable: LOAN_AMOUNT

Find the number of observations with missing values in the variable - LOAN_AMOUNT

Number of Observations
0

-----END-----

Find the details of observations with missing values in the variable - LOAN_AMOUNT

-----END-----

Figure 5.91 shows the SAS code executed to identify the quantity and details of missing values in the variable “LOAN_AMOUNT” after imputation. The output from the code execution is shown in Table 5.33. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.3.7 Missing Values Imputation in the Numerical Variable – LOAN_DURATION

The following outline each step of the missing value imputation process for the numerical variable "LOAN_DURATION". There are five steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TRAINING_DS

```
889 /********** Imputing missing values - Numerical variable: LOAN_DURATION **********/
890 /* STEP 1: Make a copy of the dataset - LIB65778.TRAINING_DS */
891
892 PROC SQL;
893
894 CREATE TABLE LIB65778.TRAINING_DS_BK AS
895 SELECT *
896 FROM LIB65778.TRAINING_DS;
897
898 QUIT;
```

Figure 5.92: SAS code for making a dataset backup prior to imputation for variable: LOAN_DURATION

Figure 5.92 shows the SAS code executed to make a copy of the dataset for backup before the imputation procedure.

STEP 2: Find the number of missing values in the variable – LOAN_DURATION

```
900 /* STEP 2: Find the number of observations with missing values in the variable - LOAN_DURATION */
901
902 TITLE1 'Find the number of observations with missing values in the variable - LOAN_DURATION';
903 FOOTNOTE '-----END-----';
904
905 PROC SQL;
906
907 SELECT COUNT(*) Label = 'Number of observations'
908 FROM LIB65778.TRAINING_DS t
909 WHERE ( ( t.LOAN_DURATION EQ . ) OR
910         ( t.LOAN_DURATION IS NULL ) OR
911         ( t.LOAN_DURATION IS MISSING ) );
912
913 QUIT;
```

Figure 5.93: SAS code for identifying quantity of missing values for variable: LOAN_DURATION

Table 5.34: Output of quantity of missing values for variable: LOAN_DURATION

Find the number of observations with missing values in the variable - LOAN_DURATION

Number of observations
14

-----END-----

Figure 5.93 shows the SAS code executed to identify the number of missing values in the variable “LOAN_DURATION”. The output from the code execution is shown in Table 5.34. The number of observations with missing value is identified to be 14.

STEP 3: Find the details of missing values in the variable – LOAN_DURATION

```
/* STEP 3: Find the details of observations with missing values in the variable - LOAN_DURATION */

TITLE1 'Find the details of observations with missing values in the variable - LOAN_DURATION';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_DURATION EQ . ) OR
        ( t.LOAN_DURATION IS NULL ) OR
        ( t.LOAN_DURATION IS MISSING ) );

QUIT;
```

Figure 5.94: SAS code for identifying details of missing values for variable: LOAN_DURATION

Table 5.35: Output of details of missing values for variable: LOAN_DURATION

Find the details of observations with missing values in the variable - LOAN_DURATION

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	No	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	0	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—

Figure 5.94 shows the SAS code executed to identify the details of missing values in the variable “LOAN_DURATION”. The output from the code execution is shown in Table 5.35 which shows the 14 observations with the missing value identified. Based on the output, it is identified that the data in the cells of the “LOAN_DURATION” variable is missing.

STEP 4: Imputing missing values found in the variable – LOAN_DURATION

```
930 /* STEP 4: Imputing missing values found in the variable - LOAN_DURATION */
931
932 PROC STDIZE DATA=LIB65778.TRAINING_DS REPONLY
933
934 METHOD=MEAN OUT=LIB65778.TRAINING_DS;
935 VAR LOAN_DURATION;
936
937 QUIT;
```

Figure 5.95: SAS code for performing missing value imputation for variable: LOAN_DURATION

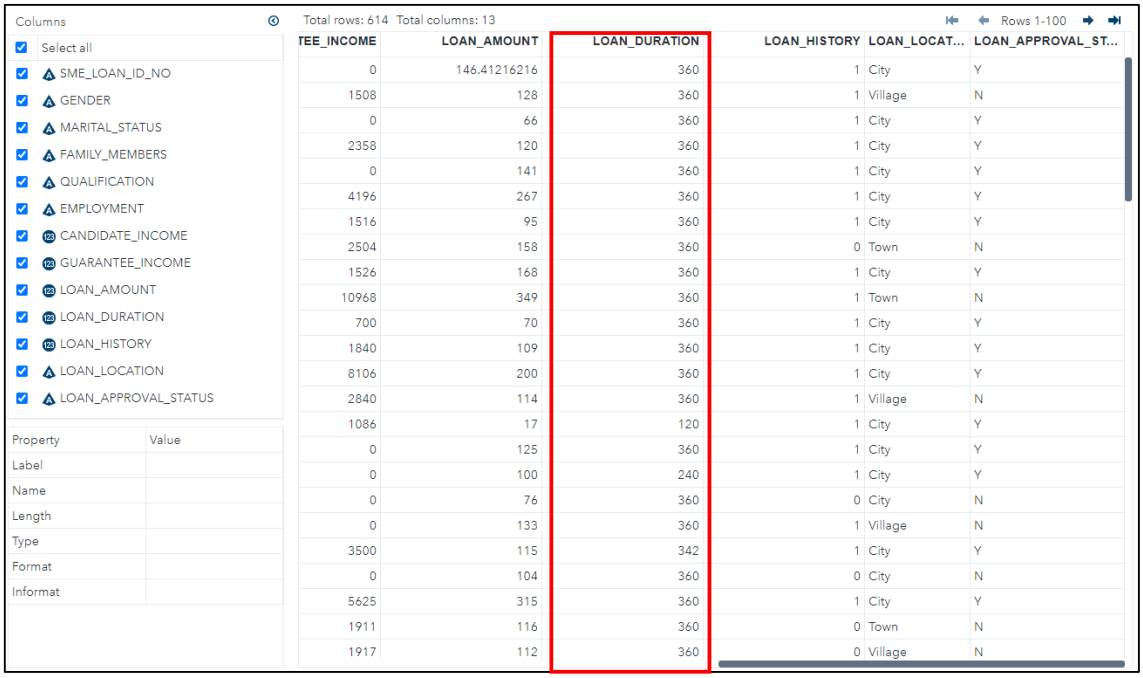


Figure 5.96: Output from imputation of variable: LOAN_DURATION

Figure 5.95 shows the SAS code executed to perform missing value imputation for the variable "LOAN_DURATION". The code execution replaces missing value with the mean value of the variable. Figure 5.96 shows the output after a successful missing value imputation procedure. It is identified that the missing value cells are filled with the mean value of the variable.

STEP 5: (After imputation) find the number of observations with missing values in the variable – LOAN_DURATION

```
/* STEP 5: (After imputation) Find the number of observations with missing values in the variable - LOAN_DURATION */

TITLE1 'Find the number of observations with missing values in the variable - LOAN_DURATION';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of Observations'
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_DURATION EQ . ) OR
        ( t.LOAN_DURATION IS NULL ) OR
        ( t.LOAN_DURATION IS MISSING ) );

QUIT;

/* STEP 6: (After imputation) Find the details of observations with missing values in the variable - LOAN_DURATION */

TITLE1 'Find the details of observations with missing values in the variable - LOAN_DURATION';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TRAINING_DS t
WHERE ( ( t.LOAN_DURATION EQ . ) OR
        ( t.LOAN_DURATION IS NULL ) OR
        ( t.LOAN_DURATION IS MISSING ) );

QUIT;
```

Figure 5.97: SAS code to identify missing value after imputation for variable: LOAN_DURATION

Table 5.36: Output of missing value after imputation for variable: LOAN_DURATION

Find the number of observations with missing values in the variable - LOAN_DURATION

Number of Observations
0

-----END-----

Find the details of observations with missing values in the variable - LOAN_DURATION

-----END-----

Figure 5.97 shows the SAS code executed to identify the quantity and details of missing values in the variable “LOAN_DURATION” after imputation. The output from the code execution is shown in Table 5.36. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.4 ANALYSIS OF VARIABLES – LIB65778.TRAINING_DS

This section documents the analysis of categorical and continuous variables found in the "TESTING_DS" dataset. Univariate and bivariate analysis will be performed on the variables.

5.4.1 Univariate Analysis of the Categorical Using SAS MACRO

Introduction to SAS MACRO

In SAS programming, the macro function is a powerful feature which allows the repeated execution of a set of predefined SAS statements when called. The predefined functions are saved under the name of the macro, which can be called with just one statement in future use. The macro function enables the reduction of programming of repetitive sections of codes that may require repeated usage in the future. This reduces the amount of regular coding required, which saves time and improves efficiency. In addition, macro is able to define parameters which allows the creation of dynamic variables for different input values. This allows different input values to be used on the same set of predefined statements to achieve different outputs.

Figure 5.98 shows the SAS code of the macro for performing univariate analysis of the categorical variables. While Figure 5.99 shows the calling of the macro to perform univariate analysis on each categorical variable.

```
969 /* Macro for Univariate analysis of categorical variables found in LIB65778.TESTING_DS */
970 /* Macro begins here */
971
972 %MACRO MACRO_UVA_CAT_VAR(PDS_NAME, PVAR_NAME, PTITLE_NAME);
973
974 PROC FREQ DATA = &PDS_NAME;
975 TABLE &PVAR_NAME;
976 TITLE &PTITLE_NAME;
977 QUIT;
978
979 %MEND MACRO_UVA_CAT_VAR;
980
981 /* Macro ends here */
```

Figure 5.98: SAS code to program a macro for univariate analysis of categorical variables

```
983 /* Calling the Macro for Univariate analysis of categorical variables */
984
985 %MACRO_UVA_CAT_VAR(LIB65778.TESTING_DS, MARITAL_STATUS, "Univariate Analysis of the Categorical variable: MARITAL_STATUS");
986 %MACRO_UVA_CAT_VAR(LIB65778.TESTING_DS, QUALIFICATION, "Univariate Analysis of the Categorical variable: QUALIFICATION");
987 %MACRO_UVA_CAT_VAR(LIB65778.TESTING_DS, FAMILY_MEMBERS, "Univariate Analysis of the Categorical variable: FAMILY_MEMBERS");
988 %MACRO_UVA_CAT_VAR(LIB65778.TESTING_DS, GENDER, "Univariate Analysis of the Categorical variable: GENDER");
989 %MACRO_UVA_CAT_VAR(LIB65778.TESTING_DS, EMPLOYMENT, "Univariate Analysis of the Categorical variable: EMPLOYMENT");
990 %MACRO_UVA_CAT_VAR(LIB65778.TESTING_DS, LOAN_HISTORY, "Univariate Analysis of the Categorical variable: LOAN_HISTORY");
991 %MACRO_UVA_CAT_VAR(LIB65778.TESTING_DS, LOAN_LOCATION, "Univariate Analysis of the Categorical variable: LOAN_LOCATION");
```

Figure 5.99: SAS code for calling the macro of univariate analysis of categorical variables

Table 5.37: Table output of univariate analysis for variable: MARITAL_STATUS

Univariate 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

Table 5.38: Table output of univariate analysis for variable: QUALIFICATION

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

Table 5.39: Table output of univariate analysis for variable: FAMILY_MEMBERS

Univariate Analysis of the Categorical variable: FAMILY_MEMBERS

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

Missing value present

Table 5.40: Table output of univariate analysis for variable: GENDER

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				

Missing value present

Table 5.41: Table output of univariate analysis for variable: EMPLOYMENT

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				

Missing value present

Table 5.42: Table output of univariate analysis for variable: LOAN_HISTORY

Univariate Analysis of the Categorical variable: LOAN_HISTORY

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				

Missing value present

Table 5.43: Table output of univariate analysis for variable: LOAN_LOCATION

Univariate Analysis of the Categorical variable: LOAN_LOCATION

The FREQ Procedure

LOAN_LOCATION	Frequency	Percent	Cumulative Frequency	Cumulative Percent
City	140	38.15	140	38.15
Town	116	31.61	256	69.75
Village	111	30.25	367	100.00

From Table 5.37 up to Table 5.43 shows the output of univariate analysis for the categorical variables in the "TESTING_DS" dataset. Seven outputs were shown, and each output was generated by calling the previously written macro. It is identified that four variables are experiencing missing values. These variables are "FAMILY_MEMBERS", "GENDER", "EMPLOYMENT", and "LOAN_HISTORY". Missing value imputation will be performed at a later stage for the affected variables.

In overall, each variable exhibits a similar proportion in each category label as the "TRAINING_DS" dataset. However, two variables were identified to have slight variation in the category label proportions. The "EMPLOYMENT" variable in "TESTING_DS" shows a slightly higher proportion of "NO" (89.24%) and slightly lower proportion of "YES" (10.76%) as compared to "TRAINING_DS" which proportion of "NO" (85.91%) and proportion of "YES" (14.09%). While the "LOAN_LOCATION" variable in "TESTING_DS" shows a slightly higher proportion of "CITY" (38.15%), slightly lower proportion of "TOWN" (31.61%), but similar proportion of "VILLAGE" (30.25%) as compared to "TRAINING_DS" which proportion of "CITY" (32.90%), proportion of "TOWN" (37.95%), and proportion of "VILLAGE" (29.15%).

5.4.2 Univariate Analysis of the Continuous Variable – CANDIDATE_INCOME

Figure 5.100 shows the SAS code executed to produce the univariate analysis for the continuous variable "CANDIDATE_INCOME". The output from the code execution is shown in the table format as shown in Table 5.44 and in the graphical format as shown in Figure 5.101.

```
/****************** Univariate Analysis - Continuous - CANDIDATE_INCOME ******************/
TITLE1 'Univariate Analysis of the Continuous variable: CANDIDATE_INCOME';
FOOTNOTE '-----END-----';

PROC MEANS DATA = LIB65778.TESTING_DS 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 = LIB65778.TESTING_DS;

HISTOGRAM CANDIDATE_INCOME;

TITLE 'Univariate Analysis of the Continuous variable: CANDIATE_INCOME';

RUN;
```

Figure 5.100: SAS code for univariate analysis of variable: CANDIDATE_INCOME

Table 5.44: Table output of univariate analysis for variable: CANDIDATE_INCOME

Univariate Analysis of the Continuous variable: CANDIDATE_INCOME

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
367	0	0	72529.00	4805.60	3786.00	4910.69

-----END-----

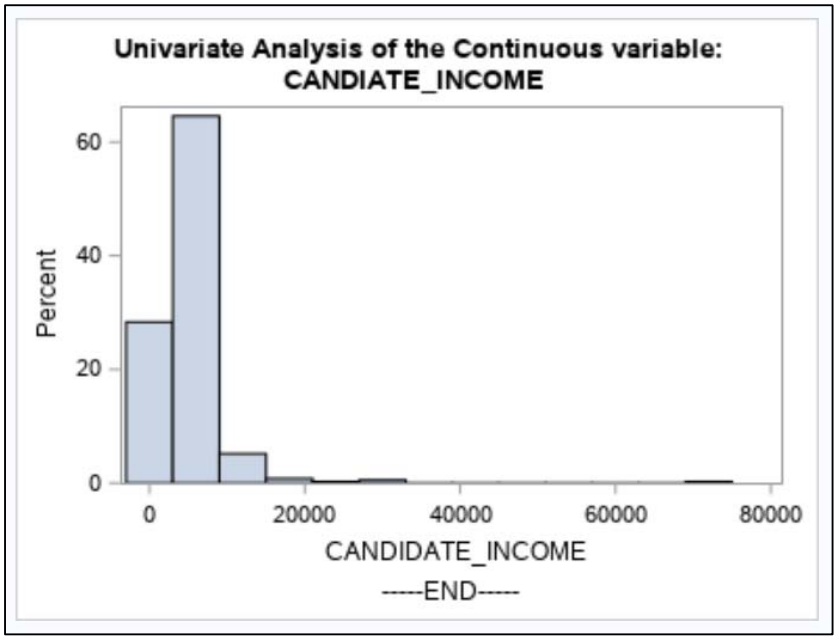


Figure 5.101: Graphical output of univariate analysis for variable: CANDIDATE_INCOME

Based on the outputs, it can be identified that there are no missing values in this variable. The variable has a range between 0 to 72529 which represents the range of income of the applicants. The mean income of the applicants is 4805.60 which is lower than the median income of the applicants which is 3786. This indicates that the sample distribution has a negative skew, where most of the applicants will have an income on the lower range. It is evident based on Figure 5.101 where about 65% of the applicants have an income around the 4800. This information can signify the spending power of the applicants, where a higher income would likely indicate a higher ability of the applicant to repay the loan.

5.4.3 Univariate Analysis of the Continuous Variable – LOAN_DURATION

Figure 5.102 shows the SAS code executed to produce the univariate analysis for the continuous variable "LOAN_DURATION". The output from the code execution is shown in the table format as shown in Table 5.45 and in the graphical format as shown in Figure 5.103.

```
1013 /****************** Univariate Analysis - Continuous - LOAN_DURATION ******************/
1014 TITLE1 'Univariate Analysis of the Continuous variable: LOAN_DURATION';
1015 FOOTNOTE '-----END-----';
1016
1017 PROC MEANS DATA = LIB65778.TESTING_DS N NMISS MIN MAX MEAN MEDIAN STD;
1018
1019 VAR LOAN_DURATION;
1020
1021 RUN;
1022
1023 ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
1024
1025 PROC SGPLOT DATA = LIB65778.TESTING_DS;
1026
1027 HISTOGRAM LOAN_DURATION;
1028
1029 TITLE 'Univariate Analysis of the Continuous variable: LOAN_DURATION';
1030
1031 RUN;
```

Figure 5.102: SAS code for univariate analysis of variable: LOAN_DURATION

Table 5.45: Table output of univariate analysis for variable: LOAN_DURATION

Univariate Analysis of the Continuous variable: LOAN_DURATION

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

-----END-----

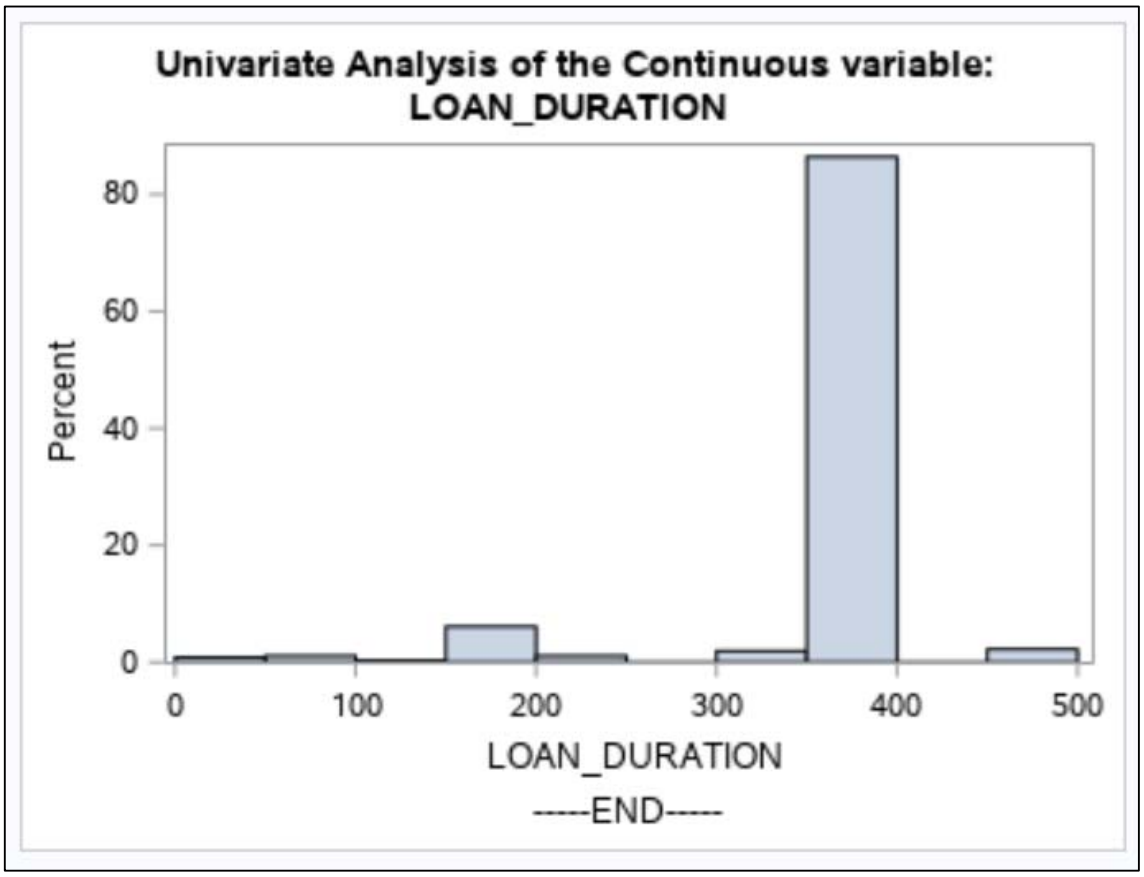


Figure 5.103: Graphical output of univariate analysis for variable: LOAN_DURATION

Based on the outputs, it can be identified that there are 6 missing values in this variable, thus would require to undergo imputation at a later stage. The variable has a range between 6 to 480 which represents the range of loan duration of the applicants in the unit of months. The mean duration of the loan is about 342 months which is lower than the median loan duration of 360 months. This indicates that the sample distribution has a negative skew, where most of the applicants will have a loan duration on the higher range. It is evident based on Figure 5.103 where about 85% of the applicants have a loan duration around 360 months. This can be explained due to the 360 months duration is the typical loan duration when an applicant is obtaining a large amount of loan to lower the monthly repayment value but stretched over a longer period of time.

5.4.4 Univariate Analysis of the Continuous Variable – GUARANTEE_INCOME

Figure 5.104 shows the SAS code executed to produce the univariate analysis for the continuous variable "GUARANTEE_INCOME". The output from the code execution is shown in the table format as shown in Table 5.46 and in the graphical format as shown in Figure 5.105.

```
1033 /****************** Univariate Analysis - Continuous - GUARANTEE_INCOME *****************/
1034 TITLE1 'Univariate Analysis of the Continuous variable: GUARANTEE_INCOME';
1035 FOOTNOTE '-----END-----';
1036
1037 PROC MEANS DATA = LIB65778.TESTING_DS N NMISS MIN MAX MEAN MEDIAN STD;
1038
1039 VAR GUARANTEE_INCOME;
1040
1041 RUN;
1042
1043 ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
1044
1045 PROC SGPLOT DATA = LIB65778.TESTING_DS;
1046
1047 HISTOGRAM GUARANTEE_INCOME;
1048
1049 TITLE 'Univariate Analysis of the Continuous variable: GUARANTEE_INCOME';
1050
1051 RUN;
```

Figure 5.104: SAS code for univariate analysis of variable: GUARANTEE_INCOME

Table 5.46: Table output of univariate analysis for variable: GUARANTEE_INCOME

Univariate Analysis of the Continuous variable: GUARANTEE_INCOME

The MEANS Procedure

Analysis Variable : GUARANTEE_INCOME						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
367	0	0	24000.00	1569.58	1025.00	2334.23

-----END-----

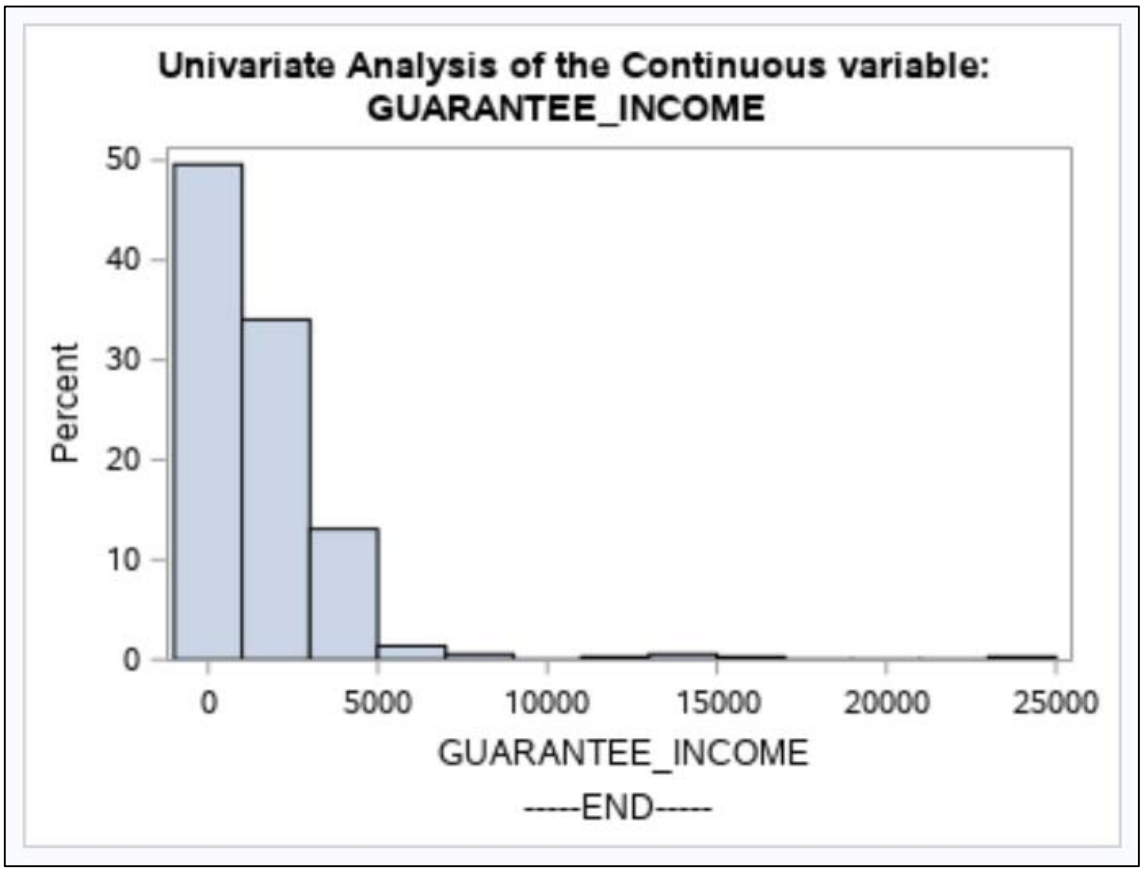


Figure 5.105: Graphical output of univariate analysis for variable: GUARANTEE_INCOME

Based on the outputs, it can be identified that there are no missing values in this variable. The variable has a range between 0 to 24,000 which represents the range of income of the co-applicants. The mean income of the co-applicants is 1569.58 which is greater than the median income of the co-applicants which is 1025.00. This indicates that the sample distribution has a positive skew, where most of the co-applicants will have an income on the lower range. It is evident based on Figure 5.105 where about 85% of the co-applicants have an income around 1600. This can be explained due to the commitment of co-applicant leading to higher unemployment or having to adopt a more flexible employment which pays lesser.

5.4.5 Univariate Analysis of the Continuous Variable – LOAN_AMOUNT

Figure 5.106 shows the SAS code executed to produce the univariate analysis for the continuous variable "LOAN_AMOUNT". The output from the code execution is shown in the table format as shown in Table 5.47 and in the graphical format as shown in Figure 5.107.

```
1053 /****************** Univariate Analysis - Continuous - LOAN_AMOUNT *****************/
1054 TITLE1 'Univariate Analysis of the Continuous variable: LOAN_AMOUNT';
1055 FOOTNOTE '-----END-----';
1056
1057 PROC MEANS DATA = LIB65778.TESTING_DS N NMISS MIN MAX MEAN MEDIAN STD;
1058
1059 VAR LOAN_AMOUNT;
1060
1061 RUN;
1062
1063 ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
1064
1065 PROC SGPLOT DATA = LIB65778.TESTING_DS;
1066
1067 HISTOGRAM LOAN_AMOUNT;
1068
1069 TITLE 'Univariate Analysis of the Continuous variable: LOAN_AMOUNT';
1070
1071 RUN;
```

Figure 5.106: SAS code for univariate analysis of variable: LOAN_AMOUNT

Table 5.47: Table output of univariate analysis for variable: LOAN_AMOUNT

Univariate Analysis of the Continuous variable: LOAN_AMOUNT

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

-----END-----

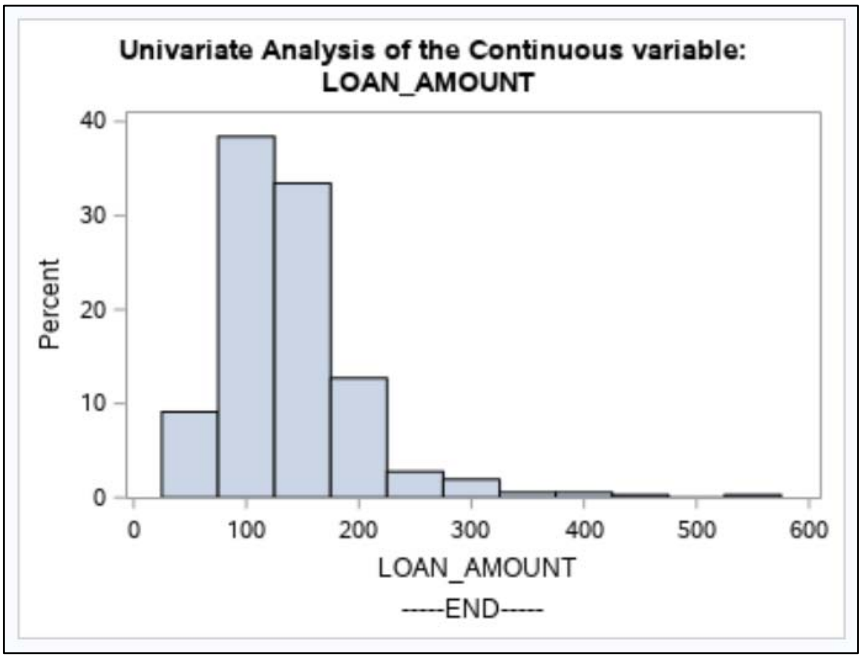


Figure 5.107: Graphical output of univariate analysis for variable: LOAN_AMOUNT

Based on the outputs, it can be identified that there are 5 missing values in this variable, thus would require to undergo imputation at a later stage. The variable has a range between 28 to 550 which represents the range of loan amount applied by the applicants in the units of thousands. The mean loan amount is 136.13 thousand which is greater than the median loan amount which is 125 thousand. This indicates that the sample distribution has a positive skew, where most of the applicants are applying a loan amount closer to the lower loan amount range. It is evident based on Figure 5.107 where about 70% of the applications are borrowing about 135 thousand in loan. This can be explained due to the nature of small businesses where it does not require a huge sum of money for startup or business growth.

5.4.6 Bivariate Analysis of the Categorical Variables Using SAS MACRO

Figure 5.108 shows the SAS code of the macro for performing bivariate analysis of the categorical variables. While Figure 5.109 shows the calling of the macro to perform bivariate analysis on the specified categorical variables.

```
1073 /* Macro for Bivariate analysis of categorical variables found in LIB65778.TESTING_DS */
1074 /* Macro begins here */
1075
1076 %MACRO MACRO_BVA_CAT_VAR(PDS_NAME, PVAR_1, PVAR_2, PTITLE_NAME);
1077
1078 PROC FREQ DATA = &PDS_NAME;
1079
1080 TABLE &PVAR_1 * &PVAR_2 /
1081 PLOTS=FREQPLOT(TWOWAY=STACKED SCALE=GROUPPCT);
1082 TITLE1 &PTITLE_NAME;
1083
1084 RUN;
1085
1086 %MEND MACRO_BVA_CAT_VAR;
1087
1088 /* Macro ends here */
```

Figure 5.108: SAS code to program a macro for bivariate analysis of categorical variables

```
1090 /* Calling the Macro for Bivariate analysis of categorical variables */
1091
1092 %MACRO_BVA_CAT_VAR(LIB65778.TESTING_DS, MARITAL_STATUS, LOAN_LOCATION, "Bivariate Analysis of variables: MARITAL_STATUS vs LOAN_LOCATION");
1093 %MACRO_BVA_CAT_VAR(LIB65778.TESTING_DS, GENDER, FAMILY_MEMBERS, "Bivariate Analysis of variables: GENDER vs FAMILY_MEMBERS");
```

Figure 5.109: SAS code for calling the macro of bivariate analysis of categorical variables

Table 5.48: Table output of bivariate analysis for variables: MARITAL_STATUS versus LOAN_LOCATION

Bivariate Analysis of variables: MARITAL_STATUS vs LOAN_LOCATION

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of MARITAL_STATUS by LOAN_LOCATION			
	LOAN_LOCATION			
MARITAL_STATUS	City	Town	Village	Total
Married	91 24.80 39.06 65.00	71 19.35 30.47 61.21	71 19.35 30.47 63.96	233 63.49
Not Married	49 13.35 36.57 35.00	45 12.26 33.58 38.79	40 10.90 29.85 36.04	134 36.51
Total	140 38.15	116 31.61	111 30.25	367 100.00

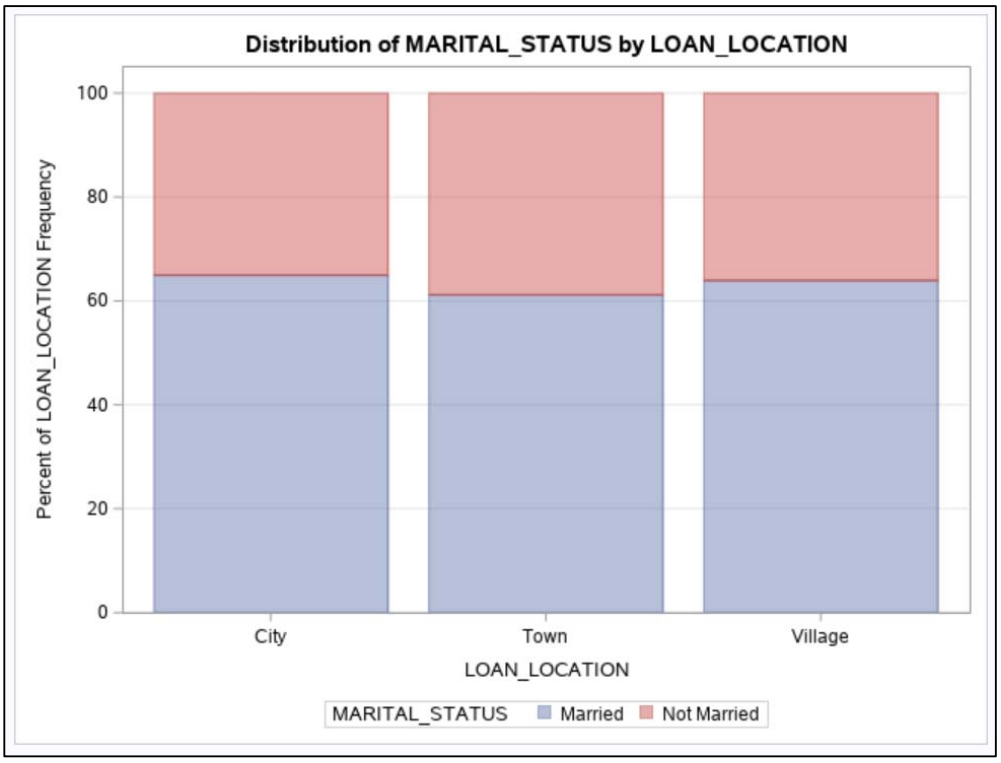


Figure 5.110: Graphical output of bivariate analysis for variables: MARITAL_STATUS versus LOAN_LOCATION

The macro was executed to produce the bivariate analysis for the categorical variable "MARITAL_STATUS" versus categorical variable "LOAN_LOCATION". The output from the macro execution is shown in the table format as shown in Table 5.48 and in the graphical format as shown in Figure 5.110.

In overall, there are 233 married applicants (63.49%) and 134 unmarried applicants (36.51%). Of the 367 applicants, there are 140 applicants from city (38.15%), 116 applicants from town (31.61%), and 111 applicants from village (30.25%). For the 233 married applicants, there are 91 applicants from city (39.06%), 71 applicants from town (30.47%), and 71 applicants from village (30.47%). For the 134 unmarried applicants, there are 49 applicants from city (36.57%), 45 applicants from town (33.58%), and 40 applicants from village (29.85%). It can be identified that in the city, there is a slightly higher proportion of married applicants, when comparing between city and town. While in the town, there is a slightly higher proportion of unmarried applicants, when comparing between city and town. However, for applicants from the village, the proportion is similar for both married and unmarried applicants.

Table 5.49: Table output of bivariate analysis for variables: GENDER versus FAMILY_MEMBERS

Bivariate Analysis of variables: GENDER vs FAMILY_MEMBERS

The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of GENDER by FAMILY_MEMBERS

GENDER	FAMILY_MEMBERS 0	1	2	3+	Total
Female	43 12.39 63.24 22.28	13 3.75 19.12 23.21	6 1.73 8.82 10.17	6 1.73 8.82 15.38	68 19.60
Male	150 43.23 53.76 77.72	43 12.39 15.41 76.79	53 15.27 19.00 89.83	33 9.51 11.83 84.62	279 80.40
Total	193 55.62	56 16.14	59 17.00	39 11.24	347 100.00

Frequency Missing = 20

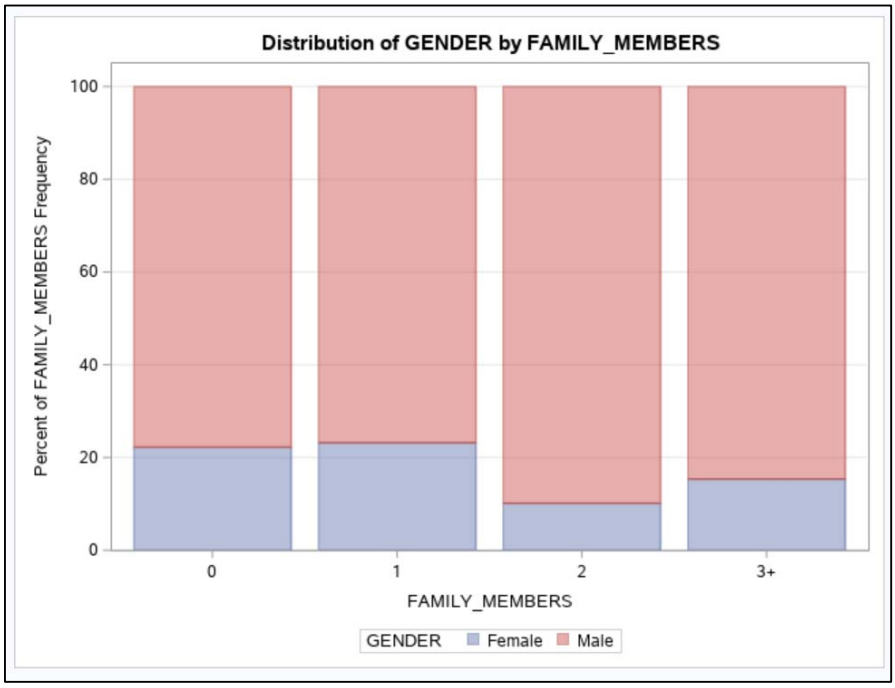


Figure 5.111: Graphical output of bivariate analysis for variables: GENDER versus FAMILY_MEMBERS

The macro was executed to produce the bivariate analysis for the categorical variable "GENDER" versus categorical variable "FAMILY_MEMBERS". The output from the macro execution is shown in the table format as shown in Table 5.49 and in the graphical format as shown in Figure 5.111.

In overall, there are 68 female applicants (19.60%) and 279 male applicants (80.40%). Of the 347 applicants, there are 193 applicants with zero dependents (55.62%), 56 applicants with one dependent (16.14%), 59 applicants with two dependents (17%), and 39 applicants with three or more dependents (11.24%). For the 68 female applicants, there are 43 female applicants with zero dependent (63.24%), 13 female applicants with one dependent (19.12%), 6 female applicants with two dependents (8.82%), and 6 female applicants with three or more dependents (8.82%). While for the 279 male applicants, there are 150 male applicants with zero dependent (53.76%), 43 male applicants with one dependent (15.41%), 53 male applicants with two dependents (19%), and 33 male applicants with three or more dependents (11.83%). It can be identified that the male applicants with higher number of dependents exhibit a greater proportion as compared to the female applicants. This can indicate the risk appetite between male and female applicants. Where in general, females with more dependents are less likely to take risks.

5.4.7 Bivariate Analysis of the Categorical Variable and Numerical Variable Using SAS MACRO

Figure 5.112 shows the SAS code for the macro for performing bivariate analysis of the categorical variable versus numerical variable. While Figure 5.113 shows the calling of the macro to perform bivariate analysis on the specified variables.

```
1095 /* Macro for Bivariate analysis of variable (categorical vs numerical) found in LIB65778.TESTING_DS */
1096
1097 %MACRO MACRO_BVA_CAT_NUM(PDS_NAME, CAT_VAR, NUM_VAR, PTITLE1, PTITLE2); /* Macro begins here */
1098 TITLE1 &PTITLE1;
1099 PROC MEANS DATA = &PDS_NAME;
1100 CLASS &CAT_VAR;
1101 VAR &NUM_VAR;
1102 RUN;
1103 PROC SGPLOT DATA = &PDS_NAME;
1104 VBOX &NUM_VAR / CATEGORY=&CAT_VAR;
1105 /* LOAN_LOCATION --> X-Axis ; CANDIDATE_INCOME --> Y-Axis */
1106 TITLE &PTITLE2;
1107 RUN;
1108
1109 %MEND MACRO_BVA_CAT_NUM; /* Macro ends here */
```

Figure 5.112: SAS code to program a macro for bivariate analysis of categorical variable versus numerical variable

```
1111 /* Calling the Macro for Bivariate analysis of (categorical vs numerical) variables */
1112
1113 %MACRO_BVA_CAT_NUM(LIB65778.TESTING_DS, GENDER, CANDIDATE_INCOME, "Bivariate Analysis of the variable", "GENDER vs CANDIDATE_INCOME");
1114 %MACRO_BVA_CAT_NUM(LIB65778.TESTING_DS, GENDER, LOAN_AMOUNT, "Bivariate Analysis of the variable", "GENDER vs LOAN_AMOUNT");
```

Figure 5.113: SAS code for calling the macro of bivariate analysis of categorical variable versus numerical variable

Table 5.50: Table output of bivariate analysis for variables: GENDER versus CANDIDATE_INCOME

Bivariate Analysis of the variable

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
GENDER	N Obs	N	Mean	Std Dev	Minimum	Maximum
Female	70	70	4163.60	2644.87	0	14987.00
Male	286	286	4932.86	5186.56	0	72529.00

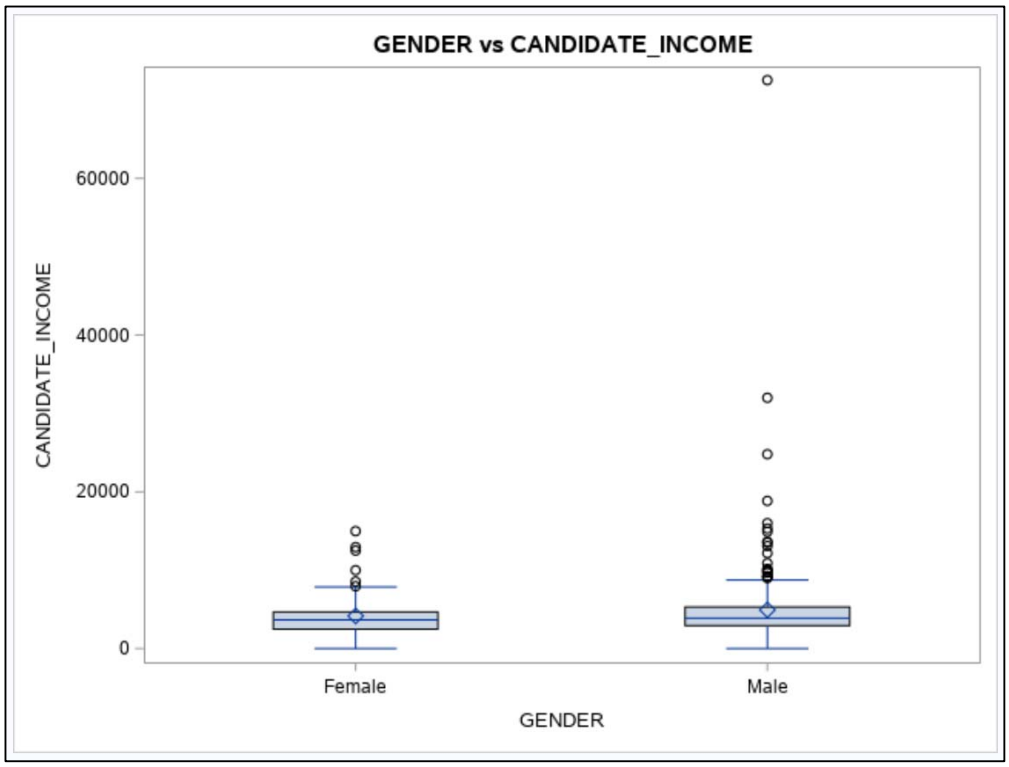


Figure 5.114: Graphical output of bivariate analysis for variables: GENDER versus CANDIDATE_INCOME

There are 70 female applicants and 286 male applicants. The average income of the female applicants is 4163.60 which is lower than the male applicants which has an average income of 4932.86. However, based on Figure 5.114, it is observed that the male applicants have a much higher standard deviation in income as compared to the female applicants. This may indicate potential outliers present in the income data of male applicants, causing a skew in the observation. It is advisable to investigate further on the extreme values to ensure veracity of the data.

Table 5.51: Table output of bivariate analysis for variables: GENDER versus LOAN_AMOUNT

Bivariate Analysis of the variable

The MEANS Procedure

Analysis Variable : LOAN_AMOUNT						
GENDER	N Obs	N	Mean	Std Dev	Minimum	Maximum
Female	70	69	126.7971014	61.1698647	28.0000000	460.0000000
Male	286	282	139.0319149	62.1310953	28.0000000	550.0000000

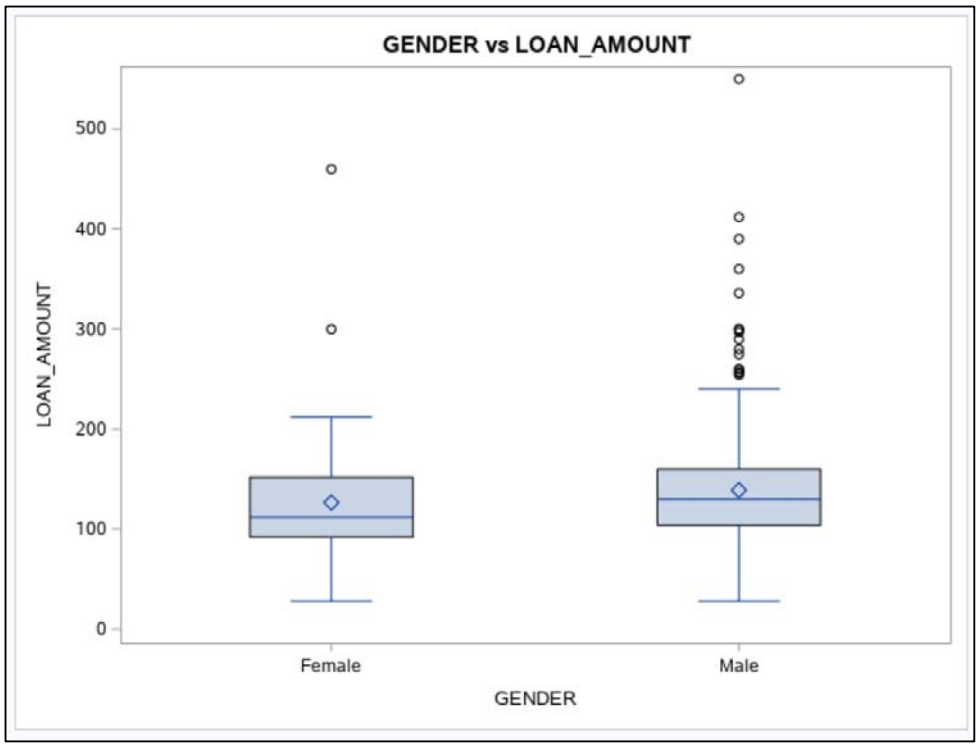


Figure 5.115: Graphical output of bivariate analysis for variables: GENDER versus LOAN_AMOUNT

There are 70 female applicants and 286 male applicants. The average loan amount of female applicants is about 126.80 thousands which is lower than the average loan amount of the male applicants which is about 139.03 thousands. However, based on Figure 5.115, it is observed that both the male and female applicants have high standard deviation. This may indicate potential outliers present in both the loan amount of male and female applicants, causing a skew in the observation. It is advisable to investigate further on the extreme values to ensure veracity of the data.

5.5 MISSING VALUES IMPUTATION – LIB65778.TESTING_DS

This section documents the missing value imputation process for variables found in "TESTING_DS" dataset. Missing values imputation will be performed on the categorical and numerical variables.

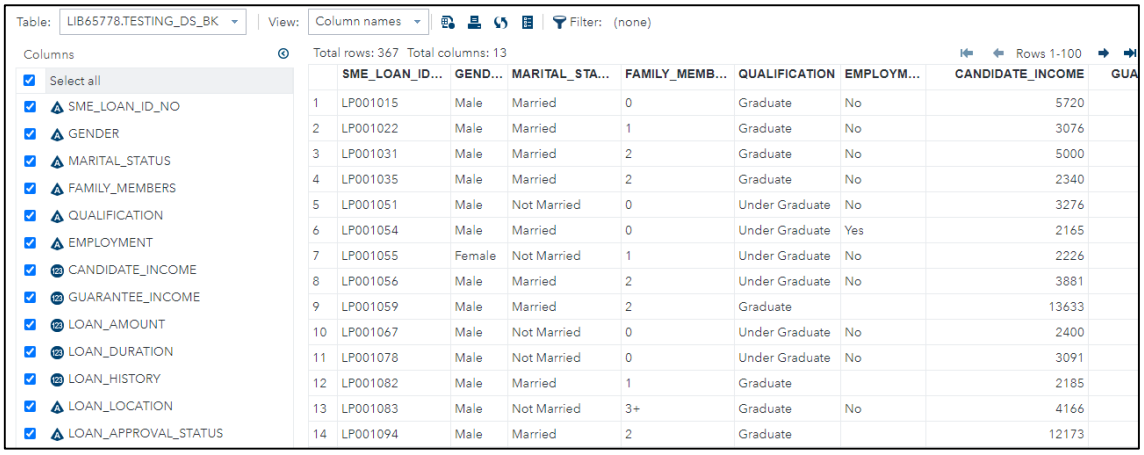
5.5.1 Missing Values Imputation in the Categorical Variable – GENDER

The following outline each step of the missing value imputation process for the categorical variable "GENDER". There are six steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TESTING_DS

```
1116 /********** Imputing missing values - Categorical variable: GENDER **********/
1117 /* STEP 1: Make a copy of the dataset - LIB65778.TESTING_DS */
1118
1119 PROC SQL;
1120
1121 CREATE TABLE LIB65778.TESTING_DS_BK AS
1122 SELECT *
1123 FROM LIB65778.TESTING_DS;
1124
1125 QUIT;
```

Figure 5.116: SAS code for making a dataset backup prior to imputation for variable: GENDER



	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUA
1	LP001015	Male	Married	0	Graduate	No	5720	
2	LP001022	Male	Married	1	Graduate	No	3076	
3	LP001031	Male	Married	2	Graduate	No	5000	
4	LP001035	Male	Married	2	Graduate	No	2340	
5	LP001051	Male	Not Married	0	Under Graduate	No	3276	
6	LP001054	Male	Married	0	Under Graduate	Yes	2165	
7	LP001055	Female	Not Married	1	Under Graduate	No	2226	
8	LP001056	Male	Married	2	Under Graduate	No	3881	
9	LP001059	Male	Married	2	Graduate		13633	
10	LP001067	Male	Not Married	0	Under Graduate	No	2400	
11	LP001078	Male	Not Married	0	Under Graduate	No	3091	
12	LP001082	Male	Married	1	Graduate		2185	
13	LP001083	Male	Not Married	3+	Graduate	No	4166	
14	LP001094	Male	Married	2	Graduate		12173	

Figure 5.117: Backup dataset creation output

Figure 5.116 shows the SAS code executed to make a copy of the dataset for backup. The output from the code execution is shown in Figure 5.117 which shows the backup dataset. The dataset backup is an essential procedure before performing any manipulation to the dataset. This ensures the original dataset can be retrieved if the data manipulation process did not provide the outcome as expected.

STEP 2: Find the number of missing values in the variable – GENDER

```
1127 /* STEP 2: Find the number of observations with missing values in the variable - GENDER */
1128
1129 TITLE1 'Find the number of observations with missing values in the variable - GENDER';
1130 FOOTNOTE '-----END-----';
1131
1132 PROC SQL;
1133
1134 SELECT COUNT(*) Label = 'Number of observations'
1135 FROM LIB65778.TESTING_DS t
1136 WHERE ( ( t.GENDER IS MISSING ) OR
1137         ( t.GENDER EQ '' ) OR
1138         ( t.GENDER IS NULL ) );
1139
1140 QUIT;
```

Figure 5.118: SAS code for identifying quantity of missing values for variable: GENDER

Table 5.52: Output of quantity of missing values for variable: GENDER

Find the number of observations with missing values in the variable - GENDER

Number of observations
11

-----END-----

Figure 5.118 shows the SAS code executed to identify the number of missing values in the variable “GENDER”. The output from the code execution is shown in Table 5.52. The number of observations with missing value is identified to be 11.

STEP 3: Find the details of missing values in the variable – GENDER

```
1142 /* STEP 3: Find the details of observations with missing values in the variable - GENDER */
1143
1144 TITLE1 'Find the details of observations with missing values in the variable - GENDER';
1145 FOOTNOTE '-----END-----';
1146
1147 PROC SQL;
1148
1149 SELECT *
1150 FROM LIB65778.TESTING_DS t
1151 WHERE ( ( t.GENDER IS MISSING ) OR
1152         ( t.GENDER EQ '' ) OR
1153         ( t.GENDER IS NULL ) );
1154
1155 QUIT;
```

Figure 5.119: SAS code for identifying details of missing values for variable: GENDER

Table 5.53: Output of details of missing values for variable: GENDER

Find the details of observations with missing values in the variable - GENDER

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
LP001128		Not Married	0	Graduate	No	3909	0	101	360	1	City	
LP001287		Married	3+	Under Graduate	No	3500	833	120	360	1	Town	
LP001563		Not Married	0	Graduate	No	1596	1760	119	360	0	City	
LP001769		Not Married		Graduate	No	3333	1250	110	360	1	Town	
LP002165		Not Married	1	Under Graduate	No	2038	4027	100	360	1	Village	
LP002296		Not Married	0	Graduate	Yes	2860	2968	138	360	1	City	
LP002355		Married	0	Graduate	No	3186	3145	150	180	0	Town	
LP002553		Not Married	0	Graduate	No	29167	0	185	360	1	Town	
LP002614		Not Married	0	Graduate	No	6478	0	108	360	1	Town	
LP002657		Married	1	Under Graduate	Yes	570	2125	68	360	1	Village	
LP002775		Not Married	0	Under Graduate	No	4768	0	125	360	1	Village	

---END---

Figure 5.119 shows the SAS code executed to identify the details of missing values in the variable “GENDER”. The output from the code execution is shown in Table 5.53 which shows the 11 observations with the missing value identified. Based on the output, it is identified that the data in the cells of the “GENDER” variable is missing.

STEP 4: Create a temporary dataset to hold GENDER and number of applicants

```
1157 /* STEP 4: Create a temporary dataset to hold GENDER category and counts */
1158
1159 PROC SQL;
1160
1161 CREATE TABLE LIB65778.TESTING_FI_GENDER_DS AS
1162 SELECT t.gender AS Gender,
1163        COUNT(*) AS COUNTS
1164 FROM LIB65778.TESTING_DS t
1165 WHERE ( ( t.GENDER NE '' ) OR
1166         ( t.GENDER IS NOT NULL ) )
1167 GROUP BY t.gender;
1168
1169 QUIT;
```

Figure 5.120: SAS code for temporary dataset creation for variable: GENDER

Total rows: 2 Total columns: 2

	Gender	COUNTS
1	Female	70
2	Male	286

Figure 5.121: Output of temporary dataset for variable: GENDER

Figure 5.120 shows the SAS code executed to create a temporary dataset to contain the counts from each category of variable “GENDER”. The output from the code execution is shown in Figure 5.121. Based on the output, it is identified that there are 70 data with the label “Female” and 286 data with the label “Male”. Thus, the mode of this variable is identified to be the “Male” label.

STEP 5: Find the mode and impute the missing values found in the variable - GENDER

```
1171 /* STEP 5: Find the MOD and impute the missing values found in the variable - GENDER */
1172
1173 PROC SQL;
1174
1175 UPDATE LIB65778.TESTING_DS
1176 SET GENDER = ( SELECT to.GENDER Label = 'Gender'
1177                FROM LIB65778.TESTING_FI_GENDER_DS to
1178                WHERE to.counts EQ ( SELECT MAX(ti.counts) Label = 'Highest Counts'
1179                                     FROM LIB65778.TESTING_FI_GENDER_DS ti ) )
1180                                     /* Above is a sub-program to identify the Mode of GENDER */
1181                                     /* and to identify the category of the MOD */
1182 WHERE ( ( GENDER IS MISSING ) OR
1183         ( GENDER EQ '' ) OR
1184         ( GENDER IS NULL ) );
1185
1186 QUIT;
```

Figure 5.122: SAS code for identifying the mode and perform missing value imputation for variable: GENDER

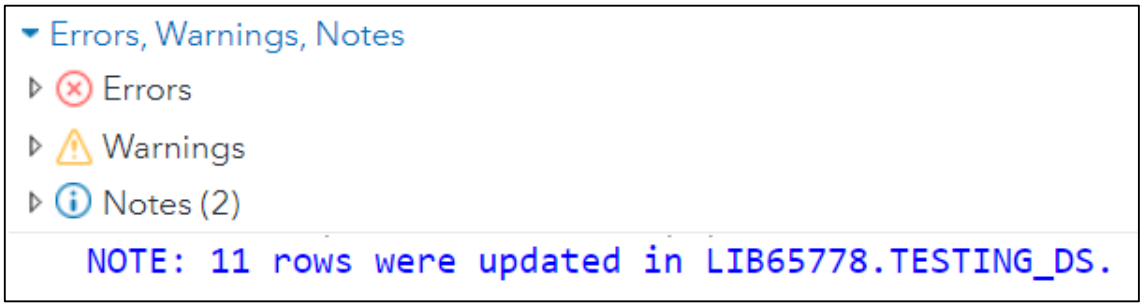


Figure 5.123: Output from imputation of variable: GENDER

Figure 5.122 shows the SAS code executed to identify the mode of the categorical variable "GENDER" and performing missing value imputation using the identified mode label. Figure 5.123 shows the output after a successful missing value imputation. It is identified that 11 rows of observations were imputed with the mode label.

STEP 6: (After imputation) find the number of observations with missing values in the variable – GENDER

```
1188 /* STEP 6: (After imputation) Find the number of observations with missing values in the variable - GENDER */
1189
1190 TITLE1 'Find the number of observations with missing values in the variable - GENDER';
1191 FOOTNOTE '-----END-----';
1192
1193 PROC SQL;
1194
1195 SELECT COUNT(*) Label = 'Number of observations'
1196 FROM LIB65778.TESTING_DS t
1197 WHERE ( ( t.GENDER IS MISSING ) OR
1198         ( t.GENDER EQ '' ) OR
1199         ( t.GENDER IS NULL ) );
1200
1201 QUIT;
1202
1203 /* STEP 7: (After imputation) Find the details of observations with missing values in the variable - GENDER */
1204
1205 TITLE1 'Find details';
1206 FOOTNOTE '-----END-----';
1207
1208 PROC SQL;
1209
1210 SELECT *
1211 FROM LIB65778.TESTING_DS t
1212 WHERE ( ( t.GENDER IS MISSING ) OR
1213         ( t.GENDER EQ '' ) OR
1214         ( t.GENDER IS NULL ) );
1215
1216 QUIT;
```

Figure 5.124: SAS code to identify missing value after imputation for variable: GENDER

Table 5.54: Output of missing value after imputation for variable: GENDER

Find the number of observations with missing values in the variable - GENDER

Number of observations
0

-----END-----

Find details

-----END-----

Figure 5.124 shows the SAS code executed to identify the quantity and details of missing values in the variable "GENDER" after imputation. The output from the code execution is shown in Table 5.54. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.5.2 Missing Values Imputation in the Categorical Variable – FAMILY_MEMBERS

The following outline each step of the missing value imputation process for the categorical variable "FAMILY_MEMBERS". There are six steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TESTING_DS

```
1218 /********** Imputing missing values - Categorical variable: FAMILY_MEMBERS **********/
1219 /* STEP 1: Make a copy of the dataset - LIB65778.TESTING_DS */
1220
1221 PROC SQL;
1222
1223 CREATE TABLE LIB65778.TESTING_DS_BK AS
1224 SELECT *
1225 FROM LIB65778.TESTING_DS;
1226
1227 QUIT;
```

Figure 5.125: SAS code for making a dataset backup prior to imputation for variable: FAMILY_MEMBERS

Figure 5.125 shows the SAS code executed to make a copy of the dataset for backup before the imputation procedure.

STEP 2: Find the number of missing values in the variable – FAMILY_MEMBERS

```
1229 /* STEP 2: Find the number of observations with missing values in the variable - FAMILY_MEMBERS */
1230
1231 TITLE1 'Find the number of observations with missing values in the variable - FAMILY_MEMBERS';
1232 FOOTNOTE '-----END-----';
1233
1234 PROC SQL;
1235
1236 SELECT COUNT(*) Label = 'Number of observations'
1237 FROM LIB65778.TESTING_DS t
1238 WHERE ( ( t.FAMILY_MEMBERS IS MISSING ) OR
1239         ( t.FAMILY_MEMBERS EQ '' ) OR
1240         ( t.FAMILY_MEMBERS IS NULL ) );
1241
1242 QUIT;
```

Figure 5.126: SAS code for identifying quantity of missing values for variable: FAMILY_MEMBERS

Table 5.55: Output of quantity of missing values for variable: FAMILY_MEMBERS

Find the number of observations with missing values in the variable - FAMILY_MEMBERS

Number of observations
10

-----END-----

Figure 5.126 shows the SAS code executed to identify the number of missing values in the variable "FAMILY_MEMBERS". The output from the code execution is shown in Table 5.55. The number of observations with missing value is identified to be 10.

STEP 3: Remove the '+' found in the variable - FAMILY_MEMBERS

```
1244 /* STEP 3: Remove the '+' found in the variable: FAMILY_MEMBERS */
1245
1246 PROC SQL;
1247
1248 UPDATE LIB65778.TESTING_DS
1249 SET FAMILY_MEMBERS = SUBSTR(FAMILY_MEMBERS,1,1)
1250 WHERE SUBSTR(FAMILY_MEMBERS,2,1) EQ '+';
1251
1252 QUIT;
```

Figure 5.127: SAS code for data manipulation for variable: FAMILY_MEMBERS

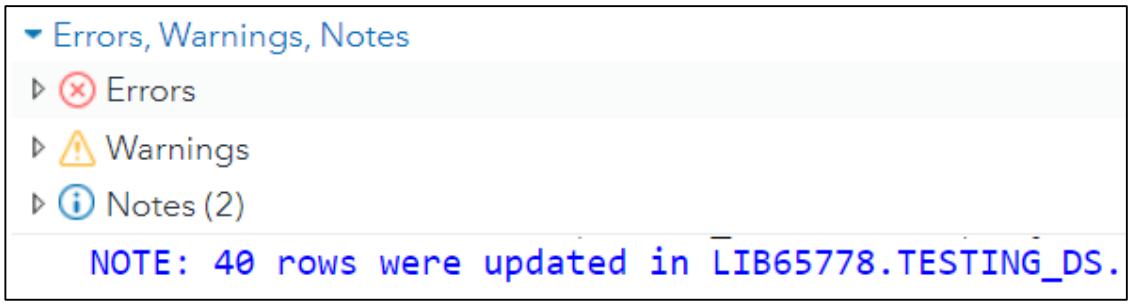


Figure 5.128: Output from data manipulation of variable: FAMILY_MEMBERS

Figure 5.127 shows the SAS code executed to manipulate the data in the variable "FAMILY_MEMBERS" to remove the "+" symbol from the data. The removal of the symbol would facilitate the data analysis process by converting the string into a numerical data type. The manipulation process is done by filtering the data that contain the symbol and retain only the first value of the entire string. The output from the data manipulation process is shown in Figure 5.128. It is identified that 40 observations have been manipulated.

STEP 4: Create a temporary dataset to hold FAMILY_MEMBERS and counts

```
1254 /* STEP 4: Create a temporary dataset to hold FAMILY_MEMBERS and counts */
1255
1256 PROC SQL;
1257
1258 CREATE TABLE LIB65778.TESTING_DS_FI_FAMILY_MEMBERS AS
1259 SELECT t.FAMILY_MEMBERS AS FAMILY_MEMBERS,
1260                COUNT(*) AS COUNTS
1261 FROM LIB65778.TESTING_DS t
1262 WHERE ( ( t.FAMILY_MEMBERS NE '' ) OR
1263         ( t.FAMILY_MEMBERS IS NOT NULL ) )
1264 GROUP BY t.FAMILY_MEMBERS;
1265
1266 QUIT;
```

Figure 5.129: SAS code for temporary dataset creation for variable: FAMILY_MEMBERS

Total rows: 4 Total columns: 2

	FAMILY_MEMBERS	COUNTS
1	0	200
2	1	58
3	2	59
4	3	40

Figure 5.130: Output of temporary dataset for variable: FAMILY_MEMBERS

Figure 5.129 shows the SAS code executed to create a temporary dataset to contain the counts from each category of variable “FAMILY_MEMBERS”. The output from the code execution is shown in Figure 5.130. Based on the output, it is identified that there are 200 data with the label “0”, 58 data with the label “1”, 59 data with the label “2”, and 40 data with the label “3”. Thus, the mode of this variable is identified to be the “0” label.

STEP 5: Find the mode and imputing the missing values for variable - FAMILY_MEMBERS

```
1268 /* STEP 5: Find the MOD and impute the missing values found in the variable: FAMILY_MEMBERS */
1269
1270 PROC SQL;
1271
1272 UPDATE LIB65778.TESTING_DS
1273 SET FAMILY_MEMBERS = ( SELECT (to.FAMILY_MEMBERS) Label = 'Family Member Category'
1274                        FROM LIB65778.TESTING_DS_FI_FAMILY_MEMBERS to
1275                        WHERE to.counts EQ ( SELECT MAX(ti.counts) Label = 'Highest Counts'
1276                                             FROM LIB65778.TESTING_DS_FI_FAMILY_MEMBERS ti ) )
1277                                             /* Above is a sub-program to identify the counts of the MOD */
1278                                             /* and to identify the category of the MOD */
1279 WHERE ( ( FAMILY_MEMBERS IS MISSING ) OR
1280         ( FAMILY_MEMBERS EQ '' ) OR
1281         ( FAMILY_MEMBERS IS NULL ) );
1282
1283 QUIT;
```

Figure 5.131: SAS code for identifying the mode and perform missing value imputation for variable: FAMILY_MEMBERS

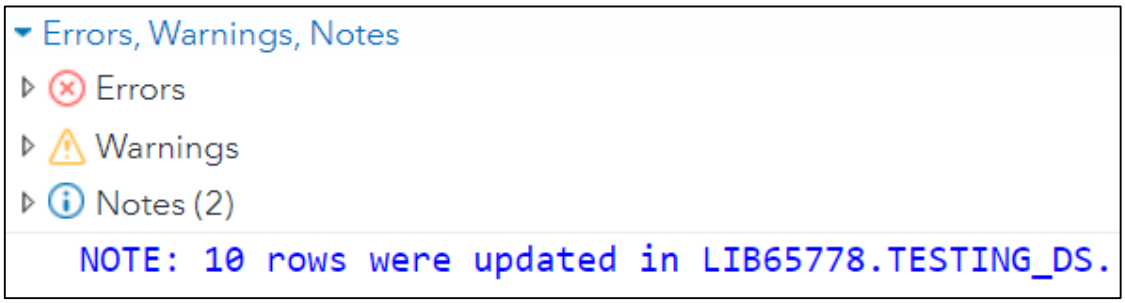


Figure 5.132: Output from imputation of variable: FAMILY_MEMBERS

Figure 5.131 shows the SAS code executed to identify the mode of the categorical variable "FAMILY_MEMBERS" and performing missing value imputation using the identified mode label. Figure 5.132 shows the output after a successful missing value imputation. It is identified that 10 rows of observations were imputed with the mode label.

STEP 6: (After imputation) find the number of observations with missing values in the variable – FAMILY_MEMBERS

```
1285 /* STEP 6: (After imputation) Find the number of observations with missing values in the variable - FAMILY_MEMBERS */
1286
1287 TITLE1 'Find the number of observations with missing values in the variable - FAMILY_MEMBERS';
1288 FOOTNOTE '-----END-----';
1289
1290 PROC SQL;
1291
1292 SELECT COUNT(*) Label = 'Number of observations'
1293 FROM LIB65778.TESTING_DS t
1294 WHERE ( ( t.FAMILY_MEMBERS IS MISSING ) OR
1295         ( t.FAMILY_MEMBERS EQ '' ) OR
1296         ( t.FAMILY_MEMBERS IS NULL ) );
1297
1298 QUIT;
1299
1300 /* STEP 7: (After imputation) Find the details of observations with missing values in the variable - FAMILY_MEMBERS */
1301
1302 TITLE1 'Find details';
1303 FOOTNOTE '-----END-----';
1304
1305 PROC SQL;
1306
1307 SELECT *
1308 FROM LIB65778.TESTING_DS t
1309 WHERE ( ( t.FAMILY_MEMBERS IS MISSING ) OR
1310         ( t.FAMILY_MEMBERS EQ '' ) OR
1311         ( t.FAMILY_MEMBERS IS NULL ) );
1312
1313 QUIT;
```

Figure 5.133: SAS code to identify missing value after imputation for variable: FAMILY_MEMBERS

Table 5.56: Output of missing value after imputation for variable: FAMILY_MEMBERS

Find the number of observations with missing values in the variable - FAMILY_MEMBERS

Number of observations
0

-----END-----

Find details

-----END-----

Figure 5.133 shows the SAS code executed to identify the quantity and details of missing values in the variable "FAMILY_MEMBERS" after imputation. The output from the code execution is shown in Table 5.56. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.5.3 Missing Values Imputation in the Continuous Variable – LOAN_AMOUNT

The following outline each step of the missing value imputation process for the numerical variable "LOAN_AMOUNT". There are five steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TESTING_DS

```
1517 /********** Imputing missing values - Numerical variable: LOAN_AMOUNT **********/
1518 /* STEP 1: Make a copy of the dataset - LIB65778.TESTING_DS */
1519
1520 PROC SQL;
1521
1522 CREATE TABLE LIB65778.TESTING_DS_BK AS
1523 SELECT *
1524 FROM LIB65778.TESTING_DS;
1525
1526 QUIT;
```

Figure 5.134: SAS code for making a dataset backup prior to imputation for variable: LOAN_AMOUNT

Figure 5.134 shows the SAS code executed to make a copy of the dataset for backup before the imputation procedure.

STEP 2: Find the number of missing values in the variable – LOAN_AMOUNT

```
1528 /* STEP 2: Find the number of observations with missing values in the variable - LOAN_AMOUNT */
1529
1530 TITLE1 'Find the number of observations with missing values in the variable - LOAN_AMOUNT';
1531 FOOTNOTE '-----END-----';
1532
1533 PROC SQL;
1534
1535 SELECT COUNT(*) Label = 'Number of Observations'
1536 FROM LIB65778.TESTING_DS t
1537 WHERE ( ( t.LOAN_AMOUNT EQ . ) OR
1538         ( t.LOAN_AMOUNT IS NULL ) OR
1539         ( t.LOAN_AMOUNT IS MISSING ) );
1540
1541 QUIT;
```

Figure 5.135: SAS code for identifying quantity of missing values for variable: LOAN_AMOUNT

Table 5.57: Output of quantity of missing values for variable: LOAN_AMOUNT

Find the number of observations with missing values in the variable - LOAN_AMOUNT

Number of Observations
5

-----END-----

Figure 5.135 shows the SAS code executed to identify the number of missing values in the variable "LOAN_AMOUNT". The output from the code execution is shown in Table 5.57. The number of observations with missing value is identified to be 5.

STEP 3: Find the details of missing values in the variable – LOAN_AMOUNT

```
1543 /* STEP 3: Find the details of observations with missing values in the variable - LOAN_AMOUNT */
1544
1545 TITLE1 'Find the details of observations with missing values in the variable - LOAN_AMOUNT';
1546 FOOTNOTE '-----END-----';
1547
1548 PROC SQL;
1549
1550 SELECT *
1551 FROM LIB65778.TESTING_DS t
1552 WHERE ( ( t.LOAN_AMOUNT EQ . ) OR
1553         ( t.LOAN_AMOUNT IS NULL ) OR
1554         ( t.LOAN_AMOUNT IS MISSING ) );
1555
1556 QUIT;
```

Figure 5.136: SAS code for identifying details of missing values for variable: LOAN_AMOUNT

Table 5.58: Output of details of missing values for variable: LOAN_AMOUNT

Find the details of observations with missing values in the variable - LOAN_AMOUNT

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
LP001415	Male	Married	1	Graduate	No	3413	4053		360	1	Town	
LP001542	Female	Married	0	Graduate	No	2262	0		480	0	Town	
LP002057	Male	Married	0	Under Graduate	No	13083	0		360	1	Village	
LP002360	Male	Married	0	Graduate	No	10000	0		360	1	City	
LP002593	Male	Married	1	Graduate	No	8333	4000		360	1	City	

—END—

Figure 5.136 shows the SAS code executed to identify the details of missing values in the variable "LOAN_AMOUNT". The output from the code execution is shown in Table 5.58 which shows the 5 observations with the missing value identified. Based on the output, it is identified that the data in the cells of the "LOAN_AMOUNT" variable is missing.

STEP 4: Imputing missing values found in the variable – LOAN_AMOUNT

```
1558 /* STEP 4: Imputing missing values found in the variable - LOAN_AMOUNT */
1559
1560 PROC STDIZE DATA=LIB65778.TESTING_DS REPONLY
1561
1562 METHOD=MEAN OUT=LIB65778.TESTING_DS;
1563 VAR LOAN_AMOUNT;
1564
1565 QUIT;
```

Figure 5.137: SAS code for performing missing value imputation for variable: LOAN_AMOUNT

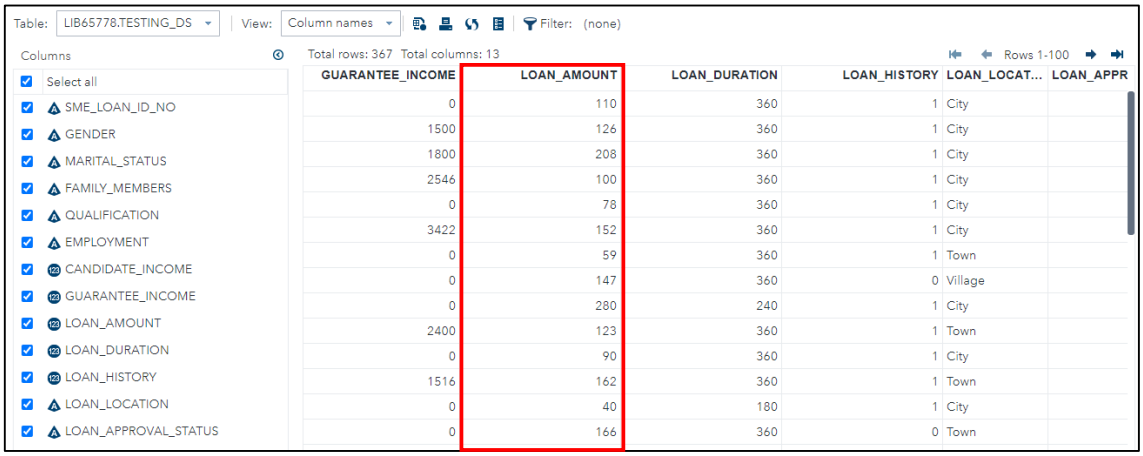


Table: LIB65778.TESTING_DS | View: Column names | Filter: (none)

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: 367 Total columns: 13 | Rows 1-100

GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCAT...	LOAN_APPR
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	59	360	1	Town	
0	147	360	0	Village	
0	280	240	1	City	
2400	123	360	1	Town	
0	90	360	1	City	
1516	162	360	1	Town	
0	40	180	1	City	
0	166	360	0	Town	

Figure 5.138: Output from imputation of variable: LOAN_AMOUNT

Figure 5.137 shows the SAS code executed to perform missing value imputation for the variable "LOAN_AMOUNT". The code execution replaces missing value with the mean value of the variable. Figure 5.138 shows the output after a successful missing value imputation procedure. It is identified that the missing value cells are filled with the mean value of the variable.

STEP 5: (After imputation) find the number of observations with missing values in the variable – LOAN_AMOUNT

```
1567 /* STEP 5: (After imputation) Find the number of observations with missing values in the variable - LOAN_AMOUNT */
1568
1569 TITLE1 'Find the number of observations with missing values in the variable - LOAN_AMOUNT';
1570 FOOTNOTE '-----END-----';
1571
1572 PROC SQL;
1573
1574 SELECT COUNT(*) Label = 'Number of Observations'
1575 FROM LIB65778.TESTING_DS t
1576 WHERE ( ( t.LOAN_AMOUNT EQ . ) OR
1577         ( t.LOAN_AMOUNT IS NULL ) OR
1578         ( t.LOAN_AMOUNT IS MISSING ) );
1579
1580 QUIT;
1581
1582 /* STEP 6: (After imputation) Find the details of observations with missing values in the variable - LOAN_AMOUNT */
1583
1584 TITLE1 'Find the details of observations with missing values in the variable - LOAN_AMOUNT';
1585 FOOTNOTE '-----END-----';
1586
1587 PROC SQL;
1588
1589 SELECT *
1590 FROM LIB65778.TESTING_DS t
1591 WHERE ( ( t.LOAN_AMOUNT EQ . ) OR
1592         ( t.LOAN_AMOUNT IS NULL ) OR
1593         ( t.LOAN_AMOUNT IS MISSING ) );
1594
1595 QUIT;
```

Figure 5.139: SAS code to identify missing value after imputation for variable: LOAN_AMOUNT

Table 5.59: Output of missing value after imputation for variable: LOAN_AMOUNT

Find the number of observations with missing values in the variable - LOAN_AMOUNT

Number of Observations
0

-----END-----

Find the details of observations with missing values in the variable - LOAN_AMOUNT

-----END-----

Figure 5.139 shows the SAS code executed to identify the quantity and details of missing values in the variable "LOAN_AMOUNT" after imputation. The output from the code execution is shown in Table 5.59. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

5.5.4 Missing Values Imputation in the Continuous Variable – LOAN_DURATION

The following outline each step of the missing value imputation process for the numerical variable "LOAN_DURATION". There are five steps in this process.

STEP 1: Make a copy of the dataset – LIB65778.TRAINING_DS

```
1597 /********** Imputing missing values - Numerical variable: LOAN_DURATION **********/
1598 /* STEP 1: Make a copy of the dataset - LIB65778.TESTING_DS */
1599
1600 PROC SQL;
1601
1602 CREATE TABLE LIB65778.TESTING_DS_BK AS
1603 SELECT *
1604 FROM LIB65778.TESTING_DS;
1605
1606 QUIT;
```

Figure 5.140: SAS code for making a dataset backup prior to imputation for variable: LOAN_DURATION

Figure 5.140 shows the SAS code executed to make a copy of the dataset for backup before the imputation procedure.

STEP 2: Find the number of missing values in the variable – LOAN_DURATION

```
1608 /* STEP 2: Find the number of observations with missing values in the variable - LOAN_DURATION */
1609
1610 TITLE1 'Find the number of observations with missing values in the variable - LOAN_DURATION';
1611 FOOTNOTE '-----END-----';
1612
1613 PROC SQL;
1614
1615 SELECT COUNT(*) Label = 'Number of observations'
1616 FROM LIB65778.TESTING_DS t
1617 WHERE ( ( t.LOAN_DURATION EQ . ) OR
1618         ( t.LOAN_DURATION IS NULL ) OR
1619         ( t.LOAN_DURATION IS MISSING ) );
1620
1621 QUIT;
```

Figure 5.141: SAS code for identifying quantity of missing values for variable: LOAN_DURATION

Table 5.60: Output of quantity of missing values for variable: LOAN_DURATION

Find the number of observations with missing values in the variable - LOAN_DURATION

Number of observations
6

-----END-----

Figure 5.141 shows the SAS code executed to identify the number of missing values in the variable "LOAN_DURATION". The output from the code execution is shown in Table 5.60. The number of observations with missing value is identified to be 6.

STEP 3: Find the details of missing values in the variable – LOAN_DURATION

```
1623 /* STEP 3: Find the details of observations with missing values in the variable - LOAN_DURATION */
1624
1625 TITLE1 'Find the details of observations with missing values in the variable - LOAN_DURATION';
1626 FOOTNOTE '-----END-----';
1627
1628 PROC SQL;
1629
1630 SELECT *
1631 FROM LIB65778.TESTING_DS t
1632 WHERE ( ( t.LOAN_DURATION EQ . ) OR
1633         ( t.LOAN_DURATION IS NULL ) OR
1634         ( t.LOAN_DURATION IS MISSING ) );
1635
1636 QUIT;
```

Figure 5.142: SAS code for identifying details of missing values for variable: LOAN_DURATION

Table 5.61: Output of details of missing values for variable: LOAN_DURATION

Find the details of observations with missing values in the variable - LOAN_DURATION

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
LP001232	Male	Married	0	Graduate	No	4260	3900	105	.	1	City	
LP001268	Male	Not Married	0	Graduate	No	6792	3338	187	.	1	City	
LP001611	Male	Married	1	Graduate	No	1516	2900	80	.	0	Village	
LP001695	Male	Married	1	Under Graduate	No	3321	2088	70	.	1	Town	
LP002045	Male	Married	3	Graduate	No	10166	750	150	.	1	City	
LP002183	Male	Married	0	Under Graduate	No	3754	3719	110	.	1	Village	

—END—

Figure 5.142 shows the SAS code executed to identify the details of missing values in the variable "LOAN_DURATION". The output from the code execution is shown in Table 5.61 which shows the 6 observations with the missing value identified. Based on the output, it is identified that the data in the cells of the "LOAN_DURATION" variable is missing.

STEP 4: Imputing missing values found in the variable – LOAN_DURATION

```
1638 /* STEP 4: Imputing missing values found in the variable - LOAN_DURATION */
1639
1640 PROC STDIZE DATA=LIB65778.TESTING_DS REPONLY
1641
1642 METHOD=MEAN OUT=LIB65778.TESTING_DS;
1643 VAR LOAN_DURATION;
1644
1645 QUIT;
```

Figure 5.143: SAS code for performing missing value imputation for variable: LOAN_DURATION

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: 367 Total columns: 13

_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCAT...
5720	0	110	360	1	City
3076	1500	126	360	1	City
5000	1800	208	360	1	City
2340	2546	100	360	1	City
3276	0	78	360	1	City
2165	3422	152	360	1	City
2226	0	59	360	1	Town
3881	0	147	360	0	Village
13633	0	280	240	1	City
2400	2400	123	360	1	Town
3091	0	90	360	1	City
2185	1516	162	360	1	Town
4166	0	40	180	1	City
12173	0	166	360	0	Town

Figure 5.144: Output from imputation of variable: LOAN_DURATION

Figure 5.143 shows the SAS code executed to perform missing value imputation for the variable "LOAN_DURATION". The code execution replaces missing value with the mean value of the variable. Figure 5.144 shows the output after a successful missing value imputation

procedure. It is identified that the missing value cells are filled with the mean value of the variable.

STEP 5: (After imputation) find the number of observations with missing values in the variable – LOAN_DURATION

```
/* STEP 5: (After imputation) Find the number of observations with missing values in the variable - LOAN_DURATION */

TITLE1 'Find the number of observations with missing values in the variable - LOAN_DURATION';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT COUNT(*) Label = 'Number of Observations'
FROM LIB65778.TESTING_DS t
WHERE ( ( t.LOAN_DURATION EQ . ) OR
        ( t.LOAN_DURATION IS NULL ) OR
        ( t.LOAN_DURATION IS MISSING ) );

QUIT;

/* STEP 6: (After imputation) Find the details of observations with missing values in the variable - LOAN_DURATION */

TITLE1 'Find the details of observations with missing values in the variable - LOAN_DURATION';
FOOTNOTE '-----END-----';

PROC SQL;

SELECT *
FROM LIB65778.TESTING_DS t
WHERE ( ( t.LOAN_DURATION EQ . ) OR
        ( t.LOAN_DURATION IS NULL ) OR
        ( t.LOAN_DURATION IS MISSING ) );

QUIT;
```

Figure 5.145: SAS code to identify missing value after imputation for variable: LOAN_DURATION

Table 5.62: Output of missing value after imputation for variable: LOAN_DURATION

Find the number of observations with missing values in the variable - LOAN_DURATION

Number of Observations
0

-----END-----

Find the details of observations with missing values in the variable - LOAN_DURATION

-----END-----

Figure 5.145 shows the SAS code executed to identify the quantity and details of missing values in the variable “LOAN_DURATION” after imputation. The output from the code execution is shown in Table 5.62. Based on the output, there is no more missing value to be identified. This signify the success of missing value imputation procedure has been achieved.

SECTION 6

PREDICTION MODEL AND ODS

6.1 PREDICTION MODEL

This section documents the prediction model development process. The prediction model utilizes the logistic regression as the prediction algorithm. Following the successful development of the prediction model, it will be used to predict the loan approval outcome for the "TESTING_DS" dataset.

```
1677 /********** Prediction model using logistic regression **********/
1678 /********** Model training **********/
1679
1680 PROC LOGISTIC DATA=LIB65778.TRAINING_DS OUTMODEL=LIB65778.TRAINING_DS_MODEL;
1681 CLASS
1682 GENDER
1683 FAMILY_MEMBERS
1684 LOAN_LOCATION
1685 LOAN_HISTORY
1686 MARITAL_STATUS
1687 QUALIFICATION
1688 EMPLOYMENT;
1689 /* Above lists the categorical variables */
1690 /* LOAN_APPROVAL_STATUS is the dependent variable */
1691 MODEL LOAN_APPROVAL_STATUS =
1692 /* Following are the independent variables */
1693 GENDER
1694 FAMILY_MEMBERS
1695 LOAN_LOCATION
1696 LOAN_HISTORY
1697 MARITAL_STATUS
1698 QUALIFICATION
1699 EMPLOYMENT
1700 LOAN_AMOUNT
1701 LOAN_DURATION;
1702 OUTPUT OUT = LIB65778.TRAINING_OUT_DS P = PRED_PROB;
1703 /* PRED_PROB is a variable holding the predicted probability */
1704 /* OUT is the dataset holding the predicted probability */
1705 /* Akaike Information criterion must (AIC) < SC (Schwarz Criterion) */
1706 RUN;
```

Figure 6.1: SAS code model fitting using logistic regression

The logistic regression model is a supervised machine learning model typically used for classification task. The model outputs the probability of a binary dependent variable. The dependent variable in this study is the "LOAN_APPROVAL_STATUS" variable. Figure 6.1 shows the SAS code for developing the logistic regression model that predicts the approval

status. The model performs fitting based on the selected independent variables as shown in the code on the "TRAINING_DS" dataset. Specifically, the "CANDIDATE_INCOME" and "GUARANTEE_INCOME" was left out from the model fitting phase. This is due to the insignificance of the model in contributing to the prediction outcome.

Figure 6.2: Observations used in model

Figure 6.2 shows the number of observations used in developing the prediction model. The number of observations identified in the original dataset is 614, while the number of observations used in developing the model is the same 614. This indicates that all observations have been utilized in developing the model which would result in a better prediction model due to all the instances have been utilized and learnt by the model.

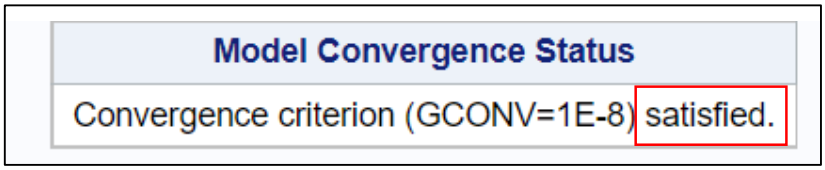


Figure 6.3: Model convergence status

Figure 6.3 shows the model convergence status of the fitted logistic regression model. The status shows that the model convergence criterion has been satisfied. This indicates that the iterative process has completed, and the final estimates are computed.

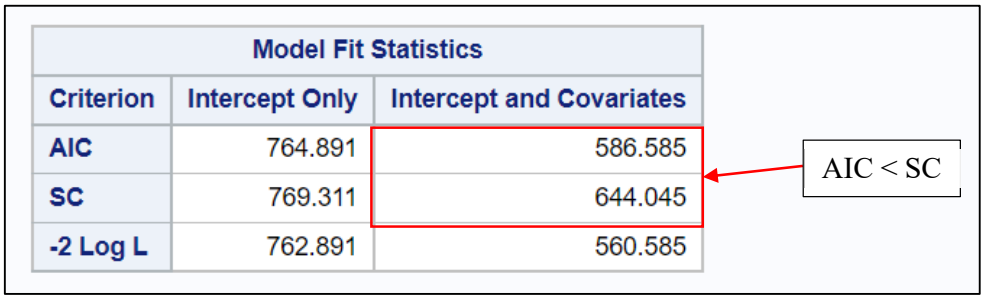
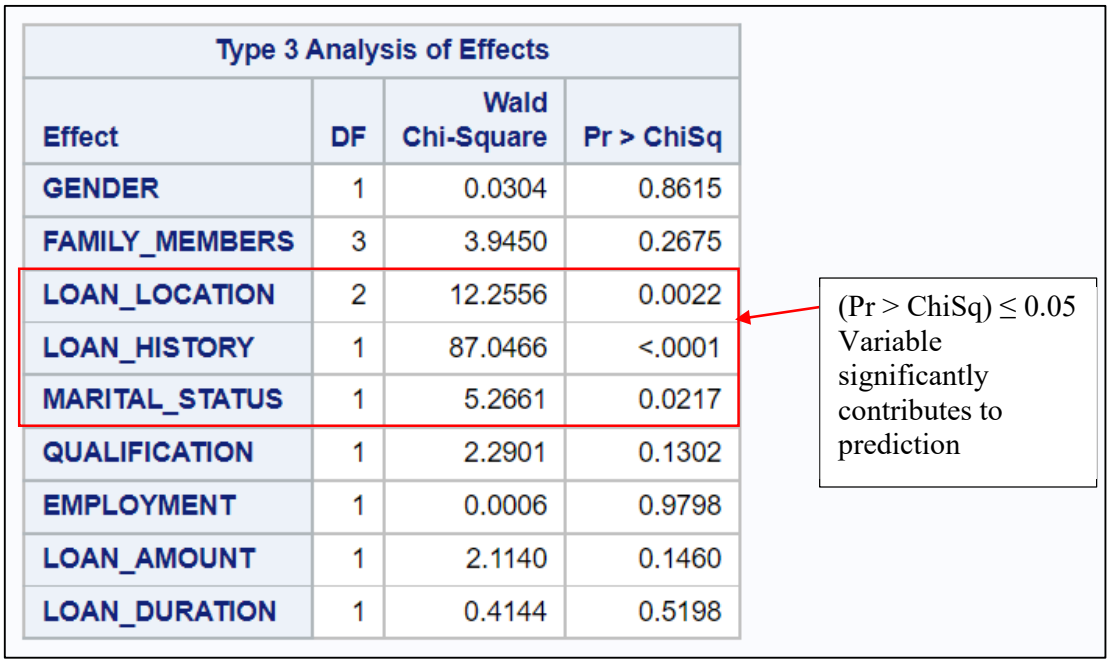


Figure 6.4: Model fit statistics

Figure 6.4 shows the model fit statistics computed. The Akaike's information criterion (AIC) and Schwarz criterion (SC) is typically used to compare different prediction models by measuring the model fitness. A lower AIC value is typically preferred as it indicates better model fitness. Similarly, a lower SC value is typically preferred. Based on the provided criteria, the model is deemed sufficiently fitted by having an AIC score less than the SC score. Based on Figure 6.4, it is evident that the computed AIC is less than the SC. Thus, the model fitness is acceptable.



Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
GENDER	1	0.0304	0.8615
FAMILY_MEMBERS	3	3.9450	0.2675
LOAN_LOCATION	2	12.2556	0.0022
LOAN_HISTORY	1	87.0466	<.0001
MARITAL_STATUS	1	5.2661	0.0217
QUALIFICATION	1	2.2901	0.1302
EMPLOYMENT	1	0.0006	0.9798
LOAN_AMOUNT	1	2.1140	0.1460
LOAN_DURATION	1	0.4144	0.5198

Figure 6.5: Type 3 analysis of effects

Figure 6.5 shows the independent variables used in the building the prediction model along with the computed Wald Chi-Square statistics and probability value. This test is used to determine significance of variables in affecting the prediction outcome. For independent variable to deem significant, the criteria of (Pr > ChiSq) is to be less than or equal to 0.05. Based on the outputs, it is identified that three variables have satisfied this criterion, namely "LOAN_LOCATION", "LOAN_HISTORY", and "MARITAL_STATUS". This indicates only three variables provide significant impact to the prediction outcome, and other independent variables do not provide significant impact to the prediction outcome.

Figure 6.6 shows the SAS code executed to utilize the fitted model to predict the outcomes on the "TESTING_DS" dataset while Figure 6.7 shows the predicted outcome table.

```
1711 /*********** Model predicting LIB65778.TESTING_DS ***********/
1712
1713 PROC LOGISTIC INMODEL=LIB65778.TRAINING_DS_MODEL;
1714
1715 SCORE DATA=LIB65778.TESTING_DS
1716 OUT=LIB65778.TESTING_DS_LAS_PREDICTED; /* Location of output */
1717
1718 QUIT;
1719
1720 TITLE1 'Loan Approval Status';
1721 TITLE2 'Lasiandra Finance Inc.';
1722 FOOTNOTE '-----END-----';
1723
1724 PROC SQL;
1725
1726 SELECT *
1727 FROM LIB65778.TESTING_DS_LAS_PREDICTED;
1728
1729 QUIT;
```

Figure 6.6: SAS code for using fitted model to predict outcome

GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS	Into: LOAN_APPROVAL_STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
0	110	360	1	City			Y	0.175129	0.824871
1500	126	360	1	City			Y	0.25402	0.74598
1800	208	360	1	City			Y	0.160854	0.839146
2546	100	360	1	City			Y	0.136607	0.863393
0	78	360	1	City			Y	0.345476	0.654524
3422	152	360	1	City			Y	0.251025	0.748975
0	59	360	1	Town			Y	0.273336	0.726664
0	147	360	0	Village			N	0.94034	0.05966
0	280	240	1	City			Y	0.159444	0.840556
2400	123	360	1	Town			Y	0.221588	0.778412
0	90	360	1	City			Y	0.350313	0.649687
1516	162	360	1	Town			Y	0.153069	0.846931
0	40	180	1	City			Y	0.210135	0.789865
0	186	360	0	Town			N	0.816348	0.183652
0	124	360	1	Town			Y	0.15496	0.84504
0	131	360	1	City			Y	0.379712	0.620288
2916	200	360	1	City			Y	0.158944	0.841056
333	126	360	1	Town			Y	0.096323	0.903677
7916	300	360	1	City			Y	0.22932	0.77068
3179	100	180	1	Town			Y	0.130696	0.869304
1620	48	360	1	City			Y	0.304508	0.695492
0	28	180	1	City			Y	0.267545	0.732455
0	101	360	1	City			Y	0.271295	0.728705
0	125	360	1	City			Y	0.35272	0.64728
4380	290	360	1	City			Y	0.226195	0.773805
24000	148	360	0	Village			N	0.961704	0.038296
1250	140	360	1	City			Y	0.145207	0.854793
3750	275	360	1	City			Y	0.221565	0.778435
833	57	360	1	Town			Y	0.087763	0.912237
2302	125	180	1	City			Y	0.239939	0.760061
0	75	360	1	Town			Y	0.198877	0.801123
820	192	360	1	City			Y	0.157054	0.842946
1653	152	360	1	Town			Y	0.168702	0.831298
2708	158	360	1	City			Y	0.187791	0.812209
1541	101	360	1	City			Y	0.172831	0.827169
0	176	360	0	Town			N	0.914	0.086
4029	185	180	1	City			Y	0.13007	0.86993
2792	90	360	1	City			Y	0.168929	0.831071
0	116	360	1	City			Y	0.274997	0.725003
1963	138	360	1	City			Y	0.284484	0.715516
818	100	360	1	City			Y	0.172577	0.827423

Predicted outcome

Figure 6.7: Prediction outcome

As shown in Figure 6.7, it is evident that all 367 observations in the “TESTING_DS” dataset has the “LOAN_APPROVAL_STATUS” dependent variable predicted. The dependent variable previously does not have any value, but after prediction the dependent variable now has the approval status with “Y” indicating yes and “N” indicating no.

6.2 SAS ODS – OUTPUT DELIVERY SYSTEM

SAS Output Delivery System (ODS) is an advance feature in SAS that provides specified formatting to the output data derived from a particular SAS program. This feature allows the automation of producing tailored and detailed presentation of reports in bulks for easier comprehension and improves productivity. In addition, the reports generated can be delivered in any other file formats such as portable document format instead of displaying only on the screen. The following section outlines the development of the ODS and producing the loan approval status output of each individual loan applicant.

6.2.1 Output Delivery System Output

Figure 6.8 shows the SAS code executed for utilizing ODS to produce individual report based on each loan applicant. While Figure 6.9 shows a few examples of the output where line by line each report shows the details of a single loan applicant. LFI would be able to utilize the ODS to compile and send such reports to each applicant notifying their loan approval status. This significantly saves time and resources for LFI in producing bulk amounts of reports. In addition, LFI may customize the report presentation format if required to enhance the report readability.

```
1736 /*********** ODS - Output Delivery System ***********/
1737 ODS HTML CLOSE;
1738 ODS PDF CLOSE;
1739
1740 /* Determine the physical location of pdf */
1741 ODS PDF FILE="/home/u60782518/DAP_FT_MAY_2022_TP065778/Report.pdf";
1742 OPTIONS NOBYLINE NODATE;
1743 TITLE1 "Bank Loan Approval Status Predicted";
1744 TITLE2 "Lasiandra Finance Inc.";
1745
1746 PROC REPORT DATA=LIB65778.TESTING_DS_LAS_PREDICTED NOWINDOWS;
1747
1748 BY SME_LOAN_ID_NO; /* To separate each by SME_LOAN_ID_NO */
1749 DEFINE SME_LOAN_ID_NO / GROUP 'Loan Application ID';
1750 DEFINE I_LOAN_APPROVAL_STATUS / GROUP 'Loan Approval Status';
1751 FOOTNOTE '-----End of Report-----';
1752
1753 RUN;
1754 OPTIONS BYLINE;
```

Figure 6.8: SAS code for developing ODS

Bank Loan Approval Status Predicted
Lasiandra Finance Inc.

Loan Application 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	0.1751289	0.8248711

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

Bank Loan Approval Status Predicted
Lasiandra Finance Inc.

Loan Application 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	0.2540196	0.7459804

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

Bank Loan Approval Status Predicted
Lasiandra Finance Inc.

Loan Application 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
LP001031	Male	Married	2	Graduate	No	5000	1800	208	360	1	City			Y	0.1608536	0.8391464

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

Bank Loan Approval Status Predicted
Lasiandra Finance Inc.

Loan Application 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
LP001035	Male	Married	2	Graduate	No	2340	2546	100	360	1	City			Y	0.136607	0.863393

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

Bank Loan Approval Status Predicted
Lasiandra Finance Inc.

Loan Application 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
LP001051	Male	Not Married	0	Under Graduate	No	3276	0	78	360	1	City			Y	0.3454764	0.6545236

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

Figure 6.8: ODS output

Example looking at the first report, for applicant with loan ID LP001015. Based on the provided details which are the independent variables, the model has predicted that the applicant is eligible for the loan with a confidence of 82.49%. This indicates that the model is highly confident that the applicant is eligible and very likely to pay back future repayments.

SECTION 7

CONCLUSION

This study has documented the development of an automated loan approval program for LFI using SAS programming language. The program includes descriptive analytics that produce tables and graphics of the variables found in the dataset which allowed the identification of data trends and data quality issues. The prediction model was developed using the logistic regression algorithm which provides the output in terms of probability. The probability output allows the data scientist to gauge the confidence of the model in the predicted outcome which also enhances interpretability. In addition, the predicted outcome is shown in a report format which include the details of the applicant along with the confidence of the predicted outcome. This report is generated automatically for each applicant using the output delivery system. The entire program automates the descriptive and predictive analytics along with producing the report for each individual applicant. Therefore, application of the automated program would significantly accelerate the loan approval process for LFI. Thus, enhances the overall efficiency and productivity, allowing LFI to focus on more critical tasks. However, further improvement can be done to enhance the accuracy of the prediction model. Such as the use of a bigger dataset which contains more observations allowing the model to learn and capture more hidden patterns from the data. In addition, validation of the fitted model should be performed to ensure the model fitness is optimal which would lead to a better prediction model.

REFLECTION

Learning SAS programming was not a difficult task as I have previous experience in SQL and Python language. However, what I enjoyed and learnt from this module is the best practices introduced, which significantly enhances the readability of my codes. In overall, this module has provided a platform to allow the practice and learning of SQL. It has improved my SQL programming level.

The assignment was manageable with the assistance and encouragement from the lecturer. However, there was no major problem faced during the working of the assignment. The major factor on managing the assignment is time management. How I manage, is to typically work purely on a single assignment in several consecutive days. This ensures the idea and workflow does not mix with other assignments causing confusion.

REFERENCES

Corporate Finance Institute. (2022). Small and Medium-sized Enterprises (SMEs). Retrieved from https://corporatefinanceinstitute.com/resources/knowledge/other/small-and-medium-sized-enterprises-smes/

Cowling, M., Liu, W., & Calabrese, R. (2021). Has previous loan rejection scarred firms from applying for loans during Covid-19? *Small Business Economics*. doi:10.1007/s11187-021-00586-2

Cusmano, L., Koreen, M., & Pissareva, L. (2018). Enhancing SME access to diversified financing instruments. *OECD SME and Entrepreneurship Papers, 7*. doi:doi:https://doi.org/10.1787/90c8823c-en

Dastile, X., & Celik, T. (2021). Making Deep Learning-Based Predictions for Credit Scoring Explainable. *IEEE Access, 9*, 50426-50440. doi:10.1109/ACCESS.2021.3068854

Downing, D. (2021). *Demographic Makeup of SMEs in the United States and United Kingdom*. Paper presented at the United States International Trade Commission Executive Briefings on Trade. https://ssrn.com/abstract=3981832

Fan, Z. (2021). *The evaluation of bank credit based on the improved decision tree model*. Paper presented at the 2021 4th International Conference on Information Systems and Computer Aided Education, Dalian, China. https://doi-org.ezproxy.apu.edu.my/10.1145/3482632.3487479

Haoran, S., & Boyang, W. (2020). *Research on Credit Risk Assessment of Online Network Credit Based on GBDT*. Paper presented at the Proceedings of the 2020 International Conference on Big Data in Management, Manchester, United Kingdom. https://doi-org.ezproxy.apu.edu.my/10.1145/3437075.3437081

Jiang, L. (2021). *Credit Risk Assessment Method for Small and Medium-Sized Enterprises Based On Artificial Intelligence*. Paper presented at the 2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture, Manchester, United Kingdom. https://doi-org.ezproxy.apu.edu.my/10.1145/3495018.3495441

Knutson, & L, M. (2020). Credit Scoring Approaches Guidelines. *World Development Indicators, The World Bank Group*. doi:https://thedocs.worldbank.org/en/doc/935891585869698451-0130022020/original/CREDITSCORINGAPPROACHESGUIDELINESFINALWEB.pdf

Liberto, D. (2021). Small and Mid-size Enterprise (SME). Retrieved from https://www.investopedia.com/terms/s/smallandmidsizeenterprises.asp

Lusinga, M., Mokoena, T., Modupe, A., & Mariate, V. (2021, 13-15 Sept). *Investigating Statistical and Machine Learning Techniques to Improve the Credit Approval Process in Developing Countries.* Paper presented at the 2021 IEEE AFRICON.

Misera, L., Wiersch, A. M., Marre, A., & Corcoran, E. W. (2022). *Small Business Credit Survey: 2022 Report on Employer Firms*. Fed Small Business Retrieved from https://www.fedsmallbusiness.org/medialibrary/FedSmallBusiness/files/2021/2022-sbcs-employer-firms-report

Parungao, J. (2020). The Role of Artificial Intelligence in Transforming Loan Origination. Retrieved from https://www.spacequant.com/news/2020/07/14/the-role-of-artificial-intelligence-in-transforming-loan-origination/

Shoumo, S. Z. H., Dhruba, M. I. M., Hossain, S., Ghani, N. H., Arif, H., & Islam, S. (2019, 17-20 Oct. 2019). *Application of Machine Learning in Credit Risk Assessment: A Prelude to Smart Banking*. Paper presented at the TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON).

Wang, M., Yu, J., & Ji, Z. (2018). *Credit risk assessment of high-tech enterprises based on RSNCL-ANN ensemble model*. Paper presented at the Proceedings of 2018 International Conference on Mathematics and Artificial Intelligence, Chengdu, China. https://doi-org.ezproxy.apu.edu.my/10.1145/3208788.3208801

Win, S. (2018). What are the possible future research directions for bank's credit risk assessment research? A systematic review of literature. *International Economics and Economic Policy, 15*(4), 743-759. doi:10.1007/s10368-018-0412-z

Xiao, Z., & Jiao, J. (2021, 1-3 Nov. 2021). *Interpretable Credit Risk Assessment Based on Heuristic Knowledge Extraction Method*. Paper presented at the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI).

Yu, X., Yang, Q., Wang, R., Fang, R., & Deng, M. (2020). Data Cleaning for Personal Credit Scoring by Utilizing Social Media Data: An Empirical Study. *IEEE Intelligent Systems, 35*(2), 7-15. doi:10.1109/MIS.2020.2972214