```
/*
Name of DS : Abdalla Elbedwihi (TP041553)
Name of SAS program: mydapproject_tp041553.sas
Description: DAP assignment for bank loans
Date first written: Fri, 26-March-2021
Date last updated: Wednesday, 7-April-2021
Folder name : MYDAPFTTP041553
Library name: LIBFT553
*/

/*******************************************************
To create a copy of the training dataset
*******************************************************/

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_TP041553_COPY_DS AS
SELECT * FROM LIBFT553.TRAINING_TP041553_DS;

QUIT;


/*******************************************************
To display the structure of the training dataset
LIBFT553.TRAINING_TP041553_COPY_DS
*******************************************************/

PROC CONTENTS DATA = LIBFT553.TRAINING_TP041553_COPY_DS;
QUIT;


/*****************************************************************************************************************************
/*****************************************Univariate analysis of categorical variables*****************************************
/*****************************************************************************************************************************


/*******************************************************
Univariate analysis on EMPLOYEMENT - Categorical Variable
*******************************************************/


PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE EMPLOYMENT;
TITLE 'Frequency table of the variable : EMPLOYMENT - Categorical variable';

RUN;

ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;    /* To adjust size of results*/

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBAR EMPLOYMENT;
TITLE ' Univariate analysis of the variable : EMPLOYMENT - Categorical variable';

RUN;


/*******************************************************
Univariate analysis on FAMILY_MEMBERS - Categorical Variable
*******************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE FAMILY_MEMBERS;
TITLE 'Frequency table of the variable : FAMILY_MEMBERS - Categorical variable';

RUN;

ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;    /* To adjust size of results*/

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBAR FAMILY_MEMBERS;
TITLE ' Univariate analysis of the variable : FAMILY_MEMBERS - Categorical variable';
```

```
RUN;


/*****************************************************
Univariate analysis on GENDER - Categorical Variable
*****************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE GENDER;
TITLE 'Frequency table of the variable : GENDER - Categorical variable';

RUN;

ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;    /* To adjust size of results*/

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBAR GENDER;
TITLE ' Univariate analysis of the variable : GENDER - Categorical variable';

RUN;


/*****************************************************
Univariate analysis on LOAN_LOCATION - Categorical Variable
*****************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE LOAN_LOCATION;
TITLE 'Frequency table of the variable : LOAN_LOCATION - Categorical variable';

RUN;

ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;    /* To adjust size of results*/

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBAR LOAN_LOCATION;
TITLE ' Univariate analysis of the variable : LOAN_LOCATION - Categorical variable';

RUN;


/*****************************************************
Univariate analysis on MARITAL_STATUS - Categorical Variable
*****************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE MARITAL_STATUS;
TITLE 'Frequency table of the variable : MARITAL_STATUS - Categorical variable';

RUN;

ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;    /* To adjust size of results*/

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBAR MARITAL_STATUS;
TITLE ' Univariate analysis of the variable : MARITAL_STATUS - Categorical variable';

RUN;


/*****************************************************
Univariate analysis on QUALIFICATION - Categorical Variable
*****************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE QUALIFICATION;
TITLE 'Frequency table of the variable : QUALIFICATION - Categorical variable';

RUN;
```

```
ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;    /* To adjust size of results*/

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBAR QUALIFICATION;
TITLE ' Univariate analysis of the variable : QUALIFICATION - Categorical variable';

RUN;


/*****************************************************
Univariate analysis on LOAN_HISTORY - Categorical Variable
*****************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE LOAN_HISTORY;
TITLE 'Frequency table of the variable : LOAN_HISTORY - Categorical variable';

RUN;

ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;    /* To adjust size of results*/

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBAR LOAN_HISTORY;
TITLE ' Univariate analysis of the variable : LOAN_HISTORY - Categorical variable';

RUN;


/*****************************************************************************************************************************
/*****************************************Univariate analysis of continuous variables*************************************
/*****************************************************************************************************************************


/*****************************************************
Univariate analysis on LOAN_AMOUNT - Continuous Variable
*****************************************************/

PROC MEANS DATA = LIBFT553.TRAINING_TP041553_COPY_DS N NMISS MIN MAX MEAN MEDIAN STD;
VAR LOAN_AMOUNT;
TITLE "Univariate Analysis on LOAN_AMOUNT - Continuous Variable";

RUN;

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

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

HISTOGRAM LOAN_AMOUNT;
TITLE ' Distribution of LOAN_AMOUNT - Continuous Variable';

RUN;


/*****************************************************
Univariate analysis on CANDIDATE_INCOME - Continuous Variable
*****************************************************/

PROC MEANS DATA = LIBFT553.TRAINING_TP041553_COPY_DS N NMISS MIN MAX MEAN MEDIAN STD;
VAR CANDIDATE_INCOME;
TITLE "Univariate Analysis on CANDIDATE_INCOME- Continuous Variable";

RUN;

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

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

HISTOGRAM CANDIDATE_INCOME;
TITLE 'Distribution of CANDIDATE_INCOME - Continuous Variable';

RUN;


/*****************************************************
```

```
Univariate analysis on GUARANTEE_INCOME - Continuous Variable
*****************************************************/

PROC MEANS DATA = LIBFT553.TRAINING_TP041553_COPY_DS N NMISS MIN MAX MEAN MEDIAN STD;
VAR GUARANTEE_INCOME;
TITLE "Univariate Analysis on GUARANTEE_INCOME- Continuous Variable";

RUN;

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

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

HISTOGRAM GUARANTEE_INCOME;
TITLE 'Distribution of GUARANTEE_INCOME - Continuous Variable';

RUN;


/*****************************************************
Univariate analysis on LOAN_DURATION - Continuous Variable
*****************************************************/

PROC MEANS DATA = LIBFT553.TRAINING_TP041553_COPY_DS N NMISS MIN MAX MEAN MEDIAN STD;
VAR LOAN_DURATION;
TITLE "Univariate Analysis on LOAN_DURATION - Continuous Variable";

RUN;

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

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

HISTOGRAM LOAN_DURATION;
TITLE 'Distribution of LOAN_DURATION - Continuous Variable';

RUN;



/*****************************************************************************************************************
/************************************************** BIVARIATE ANALYSIS ***********************************************
/*************************************************CATEGORICAL VS CATEGORICAL**********************************************
/*****************************************************************************************************************




/*****************************************************************************************
Bivariate analysis on LOAN_HISTORY VS LOAN_APPROVAL_STATUS : Categorical VS Categorical variables
*****************************************************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE LOAN_HISTORY * LOAN_APPROVAL_STATUS/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on LOAN_HISTORY VS LOAN_APPROVAL_STATUS : Categorical VS Categorical variables";

RUN;


/*****************************************************************************************
Bivariate analysis on FAMILY_MEMBERS VS LOAN_APPROVAL_STATUS : Categorical VS Categorical variables
*****************************************************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE FAMILY_MEMBERS * LOAN_APPROVAL_STATUS/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on FAMILY_MEMBERS VS LOAN_APPROVAL_STATUS : Categorical VS Categorical variables";

RUN;


/*****************************************************************************************
Bivariate analysis on FAMILY_MEMBERS VS GENDER : Categorical VS Categorical variables
*****************************************************************************************/
```

```
PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE FAMILY_MEMBERS * GENDER/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on FAMILY_MEMBERS VS GENDER : Categorical VS Categorical variables";

RUN;


/*********************************************************************************************
Bivariate analysis on LOAN_LOCATION VS FAMILY_MEMBERS : Categorical VS Categorical variables
*********************************************************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE FAMILY_MEMBERS * LOAN_LOCATION/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on FAMILY_MEMBERS VS LOAN_LOCATION : Categorical VS Categorical variables";

RUN;


/*********************************************************************************************
Bivariate analysis on LOAN_APPROVAL_STATUS VS EMPLOYMENT : Categorical VS Categorical variables
*********************************************************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE EMPLOYMENT * LOAN_APPROVAL_STATUS/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on EMPLOYMENT VS LOAN_APPROVAL_STATUS : Categorical VS Categorical variables";

RUN;


/*********************************************************************************************
Bivariate analysis on LOAN_APPROVAL_STATUS VS QUALIFICATION : Categorical VS Categorical variables
*********************************************************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE QUALIFICATION * LOAN_APPROVAL_STATUS/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on QUALIIFICATION VS LOAN_APPROVAL_STATUS : Categorical VS Categorical variables";

RUN;


/*********************************************************************************************
Bivariate analysis on EMPLOYMENT VS QUALIFICATION : Categorical VS Categorical variables
*********************************************************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE QUALIFICATION * EMPLOYMENT/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on QUALIIFICATION VS EMPLOYMENT: Categorical VS Categorical variables";

RUN;


/*********************************************************************************************
Bivariate analysis on LOAN_HISTORY VS EMPLOYMENT : Categorical VS Categorical variables
*********************************************************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE LOAN_HISTORY * EMPLOYMENT/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on LOAN_HISTORY VS EMPLOYMENT: Categorical VS Categorical variables";

RUN;


/*********************************************************************************************
Bivariate analysis on GENDER VS LOAN_APPROVAL_STATUS : Categorical VS Categorical variables
```

```
**********************************************************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE GENDER * LOAN_APPROVAL_STATUS/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on LOAN_APPROVAL_STATUS VS GENDER: Categorical VS Categorical variables";

RUN;


/**********************************************************************************************
Bivariate analysis on GENDER VS MARITAL_STATUS : Categorical VS Categorical variables
**********************************************************************************************/

PROC FREQ DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

TABLE GENDER * MARITAL_STATUS/
PLOTS= FREQPLOT(TWOWAY = STACKED SCALE = GROUPPCT);
TITLE " Bivariate analysis on MARITAL_STATUS VS GENDER: Categorical VS Categorical variables";

RUN;


/*****************************************************************************************************************************
/************************************************************ BIVARIATE ANALYSIS ***********************************************
/************************************************************CATEGORICAL VS CONTINUOUS**********************************************
/*****************************************************************************************************************************



/**********************************************************************************************
Bivariate analysis on LOAN_AMOUNT VS LOAN_APPROVAL_STATUS: Categorical VS Continuous variables
**********************************************************************************************/

PROC MEANS DATA = LIBFT553.TRAINING_TP041553_COPY_DS N NMISS MIN MAX MEAN MEDIAN STD;

CLASS LOAN_APPROVAL_STATUS;
VAR LOAN_AMOUNT;
TITLE 'Bivariate Analysis on Loan_AMOUNT VS LOAN_APPROVAL_STATUS';

RUN;

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBOX LOAN_AMOUNT / CATEGORY = LOAN_APPROVAL_STATUS;
TITLE 'Bivariate Analysis on Loan_AMOUNT VS LOAN_APPROVAL_STATUS';

RUN;


/**********************************************************************************************
Bivariate analysis on GUARANTEE_INCOME VS LOAN_APPROVAL_STATUS: Categorical VS Continuous variables
**********************************************************************************************/

PROC MEANS DATA = LIBFT553.TRAINING_TP041553_COPY_DS N NMISS MIN MAX MEAN MEDIAN STD;

CLASS LOAN_APPROVAL_STATUS;
VAR GUARANTEE_INCOME;
TITLE 'Bivariate Analysis on GUARANTEE_INCOME VS LOAN_APPROVAL_STATUS';

RUN;

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBOX GUARANTEE_INCOME / CATEGORY = LOAN_APPROVAL_STATUS;
TITLE 'Bivariate Analysis on GUARANTEE_INCOME VS LOAN_APPROVAL_STATUS';

RUN;


/**********************************************************************************************
Bivariate analysis on LOAN_AMOUNT VS FAMILY_MEMBERS: Continuous VS Categorical variables
**********************************************************************************************/

PROC MEANS DATA = LIBFT553.TRAINING_TP041553_COPY_DS N NMISS MIN MAX MEAN MEDIAN STD;
```

```
CLASS FAMILY_MEMBERS;
VAR LOAN_AMOUNT;
TITLE 'Bivariate Analysis on LOAN_AMOUNT VS FAMILY_MEMBERS';

RUN;

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBOX LOAN_AMOUNT / CATEGORY = FAMILY_MEMBERS;
TITLE 'Bivariate Analysis on LOAN_AMOUNT VS FAMILY_MEMBERS';

RUN;


/*********************************************************************************************
Bivariate analysis on CANDIDATE_INCOME VS LOAN_APPROVAL_STATUS: Categorical VS Continuous variables
*********************************************************************************************/

PROC MEANS DATA = LIBFT553.TRAINING_TP041553_COPY_DS N NMISS MIN MAX MEAN MEDIAN STD;

CLASS LOAN_APPROVAL_STATUS;
VAR CANDIDATE_INCOME;
TITLE 'Bivariate Analysis on CANDIDATE_INCOME VS LOAN_APPROVAL_STATUS';

RUN;

PROC SGPLOT DATA = LIBFT553.TRAINING_TP041553_COPY_DS;

VBOX CANDIDATE_INCOME / CATEGORY = LOAN_APPROVAL_STATUS;
TITLE 'Bivariate Analysis on CANDIDATE_INCOME VS LOAN_APPROVAL_STATUS';

RUN;




/*****************************************************************************************************************************
/*********************************************************SAS MACRO using TESTING data*********************************
/*****************************************************************************************************************************



/*****************************************************************************************

Univariate analysis for the categorical variables found in the TESTING_DS

*****************************************************************************************/

/* MACRO CREATION */

%MACRO MACRO_UVA_CV_553(DS_NAME,VARI_NAME,FIRST_TITLE,SECOND_TITLE);                    /*  MACRO_UV_CV     Macro,  U

PROC FREQ DATA = &DS_NAME;
TABLE &VARI_NAME;

RUN;

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

PROC SGPLOT DATA = &DS_NAME;

VBAR &VARI_NAME;
TITLE &FIRST_TITLE;
TITLE2 &SECOND_TITLE;

RUN;

%MEND MACRO_UVA_CV_553;                                                                 /* MACRO begins with %MAC


/*****************************************************************************************************************************
                                                Call the SAS MACRO to do UV analysis for the Cateogrical variables Found in T
*****************************************************************************************************************************



/*********************************************************
Call the SAS MACRO to do UV analysis for EMPLOYMENT variable
```

```
************************************************************/

%MACRO_UVA_CV_553(LIBFT553.TESTING_TP041553_COPY_DS,EMPLOYMENT,'Univariate analysis of','EMPLOYMENT variable');


/************************************************************
Call the SAS MACRO to do UV analysis for FAMILY_MEMBERS variable
************************************************************/

%MACRO_UVA_CV_553(LIBFT553.TESTING_TP041553_COPY_DS,FAMILY_MEMBERS,'Univariate analysis of','FAMILY_MEMBERS variable');

/************************************************************
Call the SAS MACRO to do UV analysis for GENDER variable
************************************************************/
%MACRO_UVA_CV_553(LIBFT553.TESTING_TP041553_COPY_DS,GENDER,'Univariate analysis of','GENDER variable');

/************************************************************
Call the SAS MACRO to do UV analysis for LOAN_LOCATION variable
************************************************************/
%MACRO_UVA_CV_553(LIBFT553.TESTING_TP041553_COPY_DS,LOAN_LOCATION,'Univariate analysis of','LOAN_LOCATION variable');

/************************************************************
Call the SAS MACRO to do UV analysis for MARITAL_STATUS variable
************************************************************/

%MACRO_UVA_CV_553(LIBFT553.TESTING_TP041553_COPY_DS,MARITAL_STATUS,'Univariate analysis of','MARITAL_STATUS variable');

/************************************************************
Call the SAS MACRO to do UV analysis for QUALIFICATION variable
************************************************************/
%MACRO_UVA_CV_553(LIBFT553.TESTING_TP041553_COPY_DS,QUALIFICATION,'Univariate analysis of','QUALIFICATION variable');

/************************************************************
Call the SAS MACRO to do UV analysis for LOAN_HISTORY variable
************************************************************/
%MACRO_UVA_CV_553(LIBFT553.TESTING_TP041553_COPY_DS,LOAN_HISTORY,'Univariate analysis of','LOAN_HISTORY variable');




/*************************************************************************************************

Univariate analysis for the continuous variables found in the TESTING_DS

*************************************************************************************************/

/* MACRO CREATION */

%MACRO MACRO_UVA_CONTVAR_553(DS_NAME,VARI_NAME,TITLE1,TITLE2);

PROC MEANS DATA = &DS_NAME N NMISS MIN MAX MEAN MEDIAN STD;
VAR &VARI_NAME;
TITLE &TITLE1;

RUN;

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

PROC SGPLOT DATA = &DS_NAME;

HISTOGRAM &VARI_NAME;
TITLE &TITLE2;

RUN;

%MEND MACRO_UVA_CV_553;


/*************************************************************************************************************************************
                                                Call the SAS MACRO to do UV analysis for the Continuous variables Found in Te
*************************************************************************************************************************************/


/************************************************************
Call the SAS MACRO to do UV analysis for LOAN_AMOUNT variable
************************************************************/
```

```
%MACRO_UVA_CONTVAR_553(LIBFT553.TESTING_TP041553_COPY_DS,LOAN_AMOUNT,'Univariate analysis of LOAN_AMOUNT variable','Distrbuti


/*********************************************************
Call the SAS MACRO to do UV analysis for CANDIDATE_INCOME variable
*********************************************************/

%MACRO_UVA_CONTVAR_553(LIBFT553.TESTING_TP041553_COPY_DS,CANDIDATE_INCOME,'Univariate analysis of CANDIDATE_INCOME variable',


/*********************************************************
Call the SAS MACRO to do UV analysis for GUARANTEE_INCOME variable
*********************************************************/
%MACRO_UVA_CONTVAR_553(LIBFT553.TESTING_TP041553_COPY_DS,GUARANTEE_INCOME,'Univariate analysis of GUARANTEE_INCOME variable',


/*********************************************************
Call the SAS MACRO to do UV analysis for LOAN_DURATION variable
*********************************************************/

%MACRO_UVA_CONTVAR_553(LIBFT553.TESTING_TP041553_COPY_DS,LOAN_DURATION,'Univariate analysis of LOAN_DURATION variable','Distr




/*****************************************************************************************

Bivariate analysis for the categorical vs categorical variables found in the TESTING dataset

*****************************************************************************************/

%MACRO MACRO_BIVA_CATCAT(DS_NAME,CAT_VAR1,CAT_VAR2,TITLE_1,TITLE_2);

PROC FREQ DATA = &DS_NAME;

TABLE &CAT_VAR1 * &CAT_VAR2 /
PLOTS = FREQPLOT( TWOWAY = STACKED SCALE = GROUPPCT );
TITLE &TITLE_1;
TITLE2 &TITLE_2;

RUN;

%MEND MACRO_BIVA_CATCAT;


/*****************************************************************************************************************
                                Call the SAS MACRO to do BIVARIATE analysis for the Cateogrical VS Categorical variables in T
*****************************************************************************************************************/

/*****************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for LOAN_HISTORY VS FAMILY_MEMBERS in Testing dataset
*****************************************************************************************/

%MACRO_BIVA_CATCAT(LIBFT553.TESTING_TP041553_COPY_DS,LOAN_HISTORY,FAMILY_MEMBERS,'Bivariate analysis of', 'LOAN_HISTORY VS FA

/*****************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for FAMILY_MEMBERS VS EMPLOYMENT in Testing dataset
*****************************************************************************************/
%MACRO_BIVA_CATCAT(LIBFT553.TESTING_TP041553_COPY_DS,FAMILY_MEMBERS,EMPLOYMENT,'Bivariate analysis of FAMILY_MEMBERS VS EMPLO


/*****************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for FAMILY_MEMBERS VS GENDER in Testing dataset
*****************************************************************************************/
%MACRO_BIVA_CATCAT(LIBFT553.TESTING_TP041553_COPY_DS,FAMILY_MEMBERS,GENDER,'Bivariate analysis of FAMILY_MEMBERS VS GENDER');


/*****************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for LOAN_LOCATION VS FAMILY_MEMBERS in Testing dataset
*****************************************************************************************/
%MACRO_BIVA_CATCAT(LIBFT553.TESTING_TP041553_COPY_DS,FAMILY_MEMBERS,LOAN_LOCATION,'Bivariate analysis of LOAN_LOCATION VS FAM


/*****************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for EMPLOYMENT VS GENDER in Testing dataset
*****************************************************************************************/
```

```
%MACRO_BIVA_CATCAT(LIBFT553.TESTING_TP041553_COPY_DS,EMPLOYMENT,GENDER,'Bivariate analysis of EMPLOYMENT VS GENDER');


/*************************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for MARITAL_STATUS VS FAMILY_MEMBERS in Testing dataset
*************************************************************************************************/
%MACRO_BIVA_CATCAT(LIBFT553.TESTING_TP041553_COPY_DS,MARITAL_STATUS,FAMILY_MEMBERS,'Bivariate analysis of MARITAL STATUS VS F


/*************************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for MARITAL_STATUS VS EMPLOYMENT in Testing dataset
*************************************************************************************************/
%MACRO_BIVA_CATCAT(LIBFT553.TESTING_TP041553_COPY_DS,MARITAL_STATUS,EMPLOYMENT,'Bivariate analysis of MARITAL STATUS VS EMPLO


/*************************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for QUALIFICATION VS EMPLOYMENTS in Testing dataset
*************************************************************************************************/

%MACRO_BIVA_CATCAT(LIBFT553.TESTING_TP041553_COPY_DS,QUALIFICATION,EMPLOYMENT,'Bivariate analysis of QUALIFICATION VS EMPLOYM



/*************************************************************************************

Bivariate analysuis for the Categorical VS Continuous variables found in the TESTING_DS

*************************************************************************************/

%MACRO MACRO_BIVARIATE_CATCONT(DS_NAME,CAT_VAR1,CONT_VAR2,TITLE1);

PROC MEANS DATA = &DS_NAME N NMISS MIN MAX MEAN MEDIAN STD;

CLASS &CAT_VAR1;
VAR &CONT_VAR2;
TITLE &TITLE1;

RUN;

PROC SGPLOT DATA = &DS_NAME;

VBOX &CONT_VAR2 / CATEGORY = &CAT_VAR1;

RUN;

%MEND MACRO_BIVARIATE_CATCONT;



/*************************************************************************************************************************************
                                Call the SAS MACRO to do BIVARIATE analysis for the Cateogrical VS Continuous variables in Te
*************************************************************************************************************************************/


/*************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for LOAN_AMOUNT VS GENDER in Testing dataset
*************************************************************************************/
%MACRO_BIVARIATE_CATCONT(LIBFT553.TESTING_TP041553_COPY_DS,GENDER,LOAN_AMOUNT,'Bivariate analysis of LOAN_AMOUNT VS GENDER');


/*************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for GUARANTEE_INCOME VS GENDER in Testing dataset
*************************************************************************************/
%MACRO_BIVARIATE_CATCONT(LIBFT553.TESTING_TP041553_COPY_DS,GENDER,GUARANTEE_INCOME,'Bivariate analysis of GUARANTEE_INCOME VS


/*************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for LOAN_AMOUNT VS FAMILY_MEMBERS in Testing dataset
*************************************************************************************/
%MACRO_BIVARIATE_CATCONT(LIBFT553.TESTING_TP041553_COPY_DS,FAMILY_MEMBERS,LOAN_AMOUNT,'Bivariate analysis of FAMILY_MEMBERS V


/*************************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for CANDIDATE_INCOME VS EMPLOYMENT in Testing dataset
*************************************************************************************************/
%MACRO_BIVARIATE_CATCONT(LIBFT553.TESTING_TP041553_COPY_DS,EMPLOYMENT,CANDIDATE_INCOME,'Bivariate analysis of CANDIDATE_INCOM
```

```
/*********************************************************************************************

Bivariate analysis for the Continuous VS Continuous variables found in the TESTING_DS

*********************************************************************************************/

%MACRO MACRO_BIVARIATE_CONTCONT(DS_NAME,CONT_VAR1,CONT_VAR2,TITLE1);

PROC CORR DATA = &DS_NAME PLOTS = SCATTER;

VAR &CONT_VAR1 &CONT_VAR2;

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

TITLE &TITLE1;

RUN;

%MEND MACRO_BIVARIATE_CONTCONT;



/*****************************************************************************************************************
                                Call the SAS MACRO to do BIVARIATE analysis for the Continuous VS Continuous variables in Tes
*****************************************************************************************************************/


/*********************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for LOAN_AMOUNT VS CANDIDATE_INCOME in Testing dataset
*********************************************************************************************/
%MACRO_BIVARIATE_CONTCONT(LIBFT553.TESTING_TP041553_COPY_DS,CANDIDATE_INCOME,LOAN_AMOUNT,'Bivariate analysis of LOAN_AMOUNT V


/*********************************************************************************************
Call the SAS MACRO to do BIVARIATE analysis for GUARANTEE_INCOME VS LOAN_AMOUNT in Testing dataset
*********************************************************************************************/
%MACRO_BIVARIATE_CONTCONT(LIBFT553.TESTING_TP041553_COPY_DS,LOAN_AMOUNT,GUARANTEE_INCOME,'Bivariate analysis of GUARANTEE_INC




/*****************************************************************************************************************
**************************************************        Missing data imputation - TRAINING DATASET      **********
*****************************************************************************************************************/


/*****************************************************
*************  GENDER      **************************
*****************************************************/


/* STEP 1 : Make a copy of the training dataset */

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_IMPUT_GENDER AS
SELECT * FROM LIBFT553.TRAINING_TP041553_COPY_DS;

QUIT;


/*  STEP 2 : Univariate analysis done for GENDER variable before imputation */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_GENDER;

TABLE GENDER;
TITLE 'Before Imputation';

RUN;
```

```
/*  STEP 3 : Details of the missing values found in Gender variable */

PROC SQL;

SELECT *
FROM LIBFT553.TRAINING_COPY_DS_IMPUT_GENDER e
WHERE ( ( e.GENDER IS NULL) OR
        ( e.GENDER EQ '' ) );

QUIT;


/*  STEP 4 : Create another copy of the dataset as MOD*/

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_IMPUT_MOD AS
SELECT e.GENDER, COUNT(*) AS COUNTS FROM LIBFT553.TRAINING_COPY_DS_IMPUT_GENDER e
WHERE ( ( e.GENDER IS NOT NULL ) OR
        ( e.GENDER NE '' ) )
GROUP BY e.GENDER;

QUIT;


/*  STEP 5 : Display the details found in the dataset LIBFT553.TRAINING_COPY_DS_IMPUT_MOD   */

PROC SQL;

TITLE 'Details of Gender variable without missing';
FOOTNOTE 'END OF REPORT';

SELECT *
FROM LIBFT553.TRAINING_COPY_DS_IMPUT_MOD e;

QUIT;



/*  STEP 6 : Find the mode of the gender variable   */

PROC SQL;

TITLE 'Gender with highest frequency';
FOOTNOTE 'END OF REPORT';

SELECT eo.gender label = 'Mode of gender'
FROM LIBFT553.TRAINING_COPY_DS_IMPUT_MOD eo
WHERE ( eo.counts EQ ( SELECT MAX(ei.counts)
                       FROM LIBFT553.TRAINING_COPY_DS_IMPUT_MOD ei ) );

QUIT;


/*  STEP 7 : Make a copy of the dataset for BEFORE IMPUTATION   */

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_IMPUT_GENDER_BI AS
SELECT * FROM LIBFT553.TRAINING_COPY_DS_IMPUT_GENDER;

QUIT;


/*  STEP 8 : IMPUTE MISSING VALUES  */

PROC SQL;

UPDATE LIBFT553.TRAINING_COPY_DS_IMPUT_GENDER_BI
SET GENDER = ( SELECT eo.gender label = 'mode of gender'
               FROM LIBFT553.TRAINING_COPY_DS_IMPUT_MOD eo
               WHERE eo.counts EQ ( SELECT MAX(e.counts)
                                    FROM LIBFT553.TRAINING_COPY_DS_IMPUT_MOD e ) )
               WHERE ( ( gender IS NULL ) OR
                       ( gender EQ '') );
```

```
QUIT;


/*  STEP 9 : AFTER IMPUTATION   */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_GENDER_BI;

TABLE GENDER;
TITLE 'AFTER IMPUTATION';

QUIT;



/*********************************************
********        MARITAL_STATUS  ***************
*********************************************/


/* STEP 1 : Make a copy of the training dataset using final dataset after imputing Gender */

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_IMPUT_MARITAL AS
SELECT * FROM LIBFT553.TRAINING_COPY_DS_IMPUT_GENDER_BI;

QUIT;


/*  STEP 2 : Univariate analysis done for MARITAL_STATUS variable before imputation */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_MARITAL;

TABLE MARITAL_STATUS;
TITLE 'Before Imputation';

RUN;

/*  STEP 3 : Create another copy of the dataset as MOD*/

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_MARITALIMP_MODE AS
SELECT e.MARITAL_STATUS, COUNT(*) AS COUNTS FROM LIBFT553.TRAINING_COPY_DS_IMPUT_MARITAL e
WHERE ( ( e.MARITAL_STATUS IS NOT NULL ) OR
        ( e.MARITAL_STATUS NE '' ) )
GROUP BY e.MARITAL_STATUS;

QUIT;


/*  STEP 4 : IMPUTE MISSING VALUES  */

PROC SQL;

UPDATE LIBFT553.TRAINING_COPY_DS_IMPUT_MARITAL
SET MARITAL_STATUS = ( SELECT eo.MARITAL_STATUS
                       FROM LIBFT553.TRAINING_COPY_DS_MARITALIMP_MODE eo
                       WHERE eo.COUNTS EQ ( SELECT MAX(e.COUNTS)
                                            FROM LIBFT553.TRAINING_COPY_DS_MARITALIMP_MODE e ) )
WHERE ( ( MARITAL_STATUS IS NULL ) OR
        ( MARITAL_STATUS EQ '' ) );

QUIT;


/*  STEP 5 : AFTER IMPUTATION   */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_MARITAL;

TABLE MARITAL_STATUS;
TITLE 'AFTER IMPUTATION';

QUIT;
```

```
/*********************************************
********        FAMILY_MEMBERS  ***************
*********************************************/


/* STEP 1 : Make a copy of the training dataset using final dataset after imputing MARITAL STATUS */

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_IMPUT_FAM_MEM AS
SELECT * FROM LIBFT553.TRAINING_COPY_DS_IMPUT_MARITAL;

QUIT;


/*  STEP 2 : Univariate analysis done for FAMILY_MEMBERS variable before imputation */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_FAM_MEM;

TABLE FAMILY_MEMBERS;
TITLE 'Before Imputation';

RUN;

/*  STEP 3 : Create another copy of the dataset as MODE table*/

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_FAM_MEMIMP_MODE AS
SELECT e.FAMILY_MEMBERS, COUNT(*) AS COUNTS FROM LIBFT553.TRAINING_COPY_DS_IMPUT_FAM_MEM e
WHERE ( ( e.FAMILY_MEMBERS IS NOT NULL ) OR
        ( e.FAMILY_MEMBERS NE '' ) )
GROUP BY e.FAMILY_MEMBERS;

QUIT;


/*  STEP 4 : IMPUTE MISSING VALUES  */

PROC SQL;

UPDATE LIBFT553.TRAINING_COPY_DS_IMPUT_FAM_MEM
SET FAMILY_MEMBERS = ( SELECT eo.FAMILY_MEMBERS
                       FROM LIBFT553.TRAINING_COPY_DS_FAM_MEMIMP_MODE eo
                       WHERE eo.COUNTS EQ ( SELECT MAX(e.COUNTS)
                                            FROM LIBFT553.TRAINING_COPY_DS_FAM_MEMIMP_MODE e ) )
WHERE ( ( FAMILY_MEMBERS IS NULL ) OR
        ( FAMILY_MEMBERS EQ '' ) );

QUIT;


/*  STEP 5 : AFTER IMPUTATION   */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_FAM_MEM;

TABLE FAMILY_MEMBERS;
TITLE 'AFTER IMPUTATION';

QUIT;


/*********************************************
********        EMPLOYMENT  *******************
*********************************************/


/* STEP 1 : Make a copy of the training dataset using final dataset after imputing FAMILY_MEMBERS */

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_IMPUT_EMPLOY AS
SELECT * FROM LIBFT553.TRAINING_COPY_DS_IMPUT_FAM_MEM;

QUIT;
```

```
/*  STEP 2 : Univariate analysis done for EMPLOYMENT variable before imputation */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_EMPLOY;

TABLE EMPLOYMENT;
TITLE 'Before Imputation';

RUN;

/*  STEP 3 : Create another copy of the dataset as MODE table*/

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_EMPIMP_MODE AS
SELECT e.EMPLOYMENT, COUNT(*) AS COUNTS FROM LIBFT553.TRAINING_COPY_DS_IMPUT_EMPLOY e
WHERE ( ( e.EMPLOYMENT IS NOT NULL ) OR
        ( e.EMPLOYMENT NE '' ) )
GROUP BY e.EMPLOYMENT;

QUIT;


/*  STEP 4 : IMPUTE MISSING VALUES  */

PROC SQL;

UPDATE LIBFT553.TRAINING_COPY_DS_IMPUT_EMPLOY
SET EMPLOYMENT = ( SELECT eo.EMPLOYMENT
                        FROM LIBFT553.TRAINING_COPY_DS_EMPIMP_MODE eo
                        WHERE eo.COUNTS EQ ( SELECT MAX(e.COUNTS)
                                    FROM LIBFT553.TRAINING_COPY_DS_EMPIMP_MODE e ) )
WHERE ( ( EMPLOYMENT IS NULL ) OR
        ( EMPLOYMENT EQ '' ) );

QUIT;


/*  STEP 5 : AFTER IMPUTATION   */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_EMPLOY;

TABLE EMPLOYMENT;
TITLE 'AFTER IMPUTATION';

QUIT;



/*********************************************
********        LOAN_HISTORY    ***************
*********************************************/


/* STEP 1 : Make a copy of the training dataset using final dataset after imputing EMPLOYMENT */

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS AS
SELECT * FROM LIBFT553.TRAINING_COPY_DS_IMPUT_EMPLOY;

QUIT;


/*  STEP 2 : Univariate analysis done for LOAN_HISTORY variable before imputation */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS;

TABLE LOAN_HISTORY;
TITLE 'Before Imputation';

RUN;

/*  STEP 3 : Create another copy of the dataset as MODE table*/

PROC SQL;

CREATE TABLE LIBFT553.TRAINING_COPY_DS_LOANHIS_MODE AS
```

```
SELECT e.LOAN_HISTORY, COUNT(*) AS COUNTS FROM LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS e
WHERE ( ( e.LOAN_HISTORY IS NOT NULL ) )
GROUP BY e.LOAN_HISTORY;

QUIT;


/*  STEP 4 : IMPUTE MISSING VALUES  */

PROC SQL;

UPDATE LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS
SET LOAN_HISTORY = ( SELECT eo.LOAN_HISTORY
                        FROM LIBFT553.TRAINING_COPY_DS_LOANHIS_MODE eo
                        WHERE eo.COUNTS EQ ( SELECT MAX(e.COUNTS)
                                    FROM LIBFT553.TRAINING_COPY_DS_LOANHIS_MODE e ) )
WHERE ( ( LOAN_HISTORY IS NULL ) );

QUIT;


/*  STEP 5 : AFTER IMPUTATION   */

PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS;

TABLE LOAN_HISTORY;
TITLE 'AFTER IMPUTATION';

QUIT;



/*************************************************************************************************************************
*************************************************  IMPUTE CONTINUOUS VARIABLES - TRAINING DATA     *********
*************************************************************************************************************************/


/*********************************************
********        LOAN_DURATION    ***************
*********************************************/

/*  STEP 1: Univariate analysis of LOAN_DURATION variable before imputation     */

PROC MEANS DATA =  LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS N NMISS MIN MAX MEAN MEDIAN STD;

VAR LOAN_DURATION;
TITLE 'Univariate analysis of LOAN_DURATION variable - Before imputation';

RUN;


/* STEP 2 : Cleanse/impute missing valeus found in the LOAN_DURATION variable   */

PROC STDIZE DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS REPONLY

METHOD = MEAN OUT = LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS;
VAR LOAN_DURATION;

QUIT;


/*  STEP 3: Univariate analysis of LOAN_DURATION variable after imputation  */

PROC MEANS DATA =  LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS N NMISS MIN MAX MEAN MEDIAN STD;

VAR LOAN_DURATION;
TITLE 'Univariate analysis of LOAN_DURATION variable - After imputation';

RUN;



/*********************************************
********        LOAN_AMOUNT ********************
*********************************************/

/*  STEP 1: Univariate analysis of LOAN_AMOUNT variable before imputation   */
```

```
PROC MEANS DATA =  LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS N NMISS MIN MAX MEAN MEDIAN STD;

VAR LOAN_AMOUNT;
TITLE 'Univariate analysis of LOAN_AMOUNT variable - Before imputation';

RUN;


/* STEP 2 : Cleanse/impute missing valeus found in the LOAN_AMOUNT variable     */

PROC STDIZE DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS REPONLY

METHOD = MEAN OUT = LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS;
VAR LOAN_AMOUNT;

QUIT;


/*  STEP 3: Univariate analysis of LOAN_AMOUNT variable after imputation    */

PROC MEANS DATA =  LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS N NMISS MIN MAX MEAN MEDIAN STD;

VAR LOAN_AMOUNT;
TITLE 'Univariate analysis of LOAN_AMOUNT variable - After imputation';

RUN;




/*************************************************************************************************************************
************************************************  IMPUTE CATEGROCIAL VARIABLES - TESTING DATA     *********
*************************************************************************************************************************/

/*********************************************
***************        GENDER  ***************
*********************************************/


/* STEP 1 : Make a copy of the testing dataset */

PROC SQL;

CREATE TABLE LIBFT553.TESTING_COPY_DS_GENDERIMP AS
SELECT * FROM LIBFT553.TESTING_TP041553_COPY_DS;

QUIT;


/*  STEP 2 : Univariate analysis done for GENDER variable before imputation */

PROC FREQ DATA = LIBFT553.TESTING_COPY_DS_GENDERIMP;

TABLE GENDER;
TITLE 'Before Imputation';

RUN;

/*  STEP 3 : Create another copy of the dataset as MODE */

PROC SQL;

CREATE TABLE LIBFT553.TESTING_COPY_DS_GENDER_MODE AS
SELECT e.GENDER, COUNT(*) AS COUNTS FROM LIBFT553.TESTING_COPY_DS_GENDERIMP e
WHERE ( ( e.GENDER IS NOT NULL ) OR
        ( e.GENDER NE '' ) )
GROUP BY e.GENDER;

QUIT;


/*  STEP 4 : IMPUTE MISSING VALUES  */

PROC SQL;
```

```
UPDATE LIBFT553.TESTING_COPY_DS_GENDERIMP
SET GENDER = ( SELECT eo.GENDER
                      FROM LIBFT553.TESTING_COPY_DS_GENDER_MODE eo
                      WHERE eo.COUNTS EQ ( SELECT MAX(e.COUNTS)
                                  FROM LIBFT553.TESTING_COPY_DS_GENDER_MODE e ) )
WHERE ( ( GENDER IS NULL ) OR
        ( GENDER EQ '' ) );

QUIT;


/*  STEP 5 : AFTER IMPUTATION   */

PROC FREQ DATA = LIBFT553.TESTING_COPY_DS_GENDERIMP;

TABLE GENDER;
TITLE 'AFTER IMPUTATION';

QUIT;


/*********************************************
********        FAMILY_MEMBERS  ***************
*********************************************/


/* STEP 1 : Make a copy of the training dataset using final dataset after imputing GENDER */

PROC SQL;

CREATE TABLE LIBFT553.TESTING_COPY_DS_IMPUT_FAM_MEM AS
SELECT * FROM LIBFT553.TESTING_COPY_DS_GENDERIMP;

QUIT;


/*  STEP 2 : Univariate analysis done for FAMILY_MEMBERS variable before imputation */

PROC FREQ DATA = LIBFT553.TESTING_COPY_DS_IMPUT_FAM_MEM;

TABLE FAMILY_MEMBERS;
TITLE 'Before Imputation';

RUN;

/*  STEP 3 : Create another copy of the dataset as MODE table*/

PROC SQL;

CREATE TABLE LIBFT553.TESTING_COPY_DS_FAM_MEMIMP_MODE AS
SELECT e.FAMILY_MEMBERS, COUNT(*) AS COUNTS FROM LIBFT553.TESTING_COPY_DS_IMPUT_FAM_MEM e
WHERE ( ( e.FAMILY_MEMBERS IS NOT NULL ) OR
        ( e.FAMILY_MEMBERS NE '' ) )
GROUP BY e.FAMILY_MEMBERS;

QUIT;


/*  STEP 4 : IMPUTE MISSING VALUES  */

PROC SQL;

UPDATE LIBFT553.TESTING_COPY_DS_IMPUT_FAM_MEM
SET FAMILY_MEMBERS = ( SELECT eo.FAMILY_MEMBERS
                      FROM LIBFT553.TESTING_COPY_DS_FAM_MEMIMP_MODE eo
                      WHERE eo.COUNTS EQ ( SELECT MAX(e.COUNTS)
                                  FROM LIBFT553.TESTING_COPY_DS_FAM_MEMIMP_MODE e ) )
WHERE ( ( FAMILY_MEMBERS IS NULL ) OR
        ( FAMILY_MEMBERS EQ '' ) );

QUIT;


/*  STEP 5 : AFTER IMPUTATION   */

PROC FREQ DATA = LIBFT553.TESTING_COPY_DS_IMPUT_FAM_MEM;
```

```
TABLE FAMILY_MEMBERS;
TITLE 'AFTER IMPUTATION';

QUIT;


/*********************************************
********        EMPLOYMENT  ********************
*********************************************/


/* STEP 1 : Make a copy of the testing dataset using final dataset after imputing FAMILY_MEMBERS */

PROC SQL;

CREATE TABLE LIBFT553.TESTING_COPY_DS_IMPUT_EMPLOY AS
SELECT * FROM LIBFT553.TESTING_COPY_DS_IMPUT_FAM_MEM;

QUIT;


/*  STEP 2 : Univariate analysis done for EMPLOYMENT variable before imputation */

PROC FREQ DATA = LIBFT553.TESTING_COPY_DS_IMPUT_EMPLOY;

TABLE EMPLOYMENT;
TITLE 'Before Imputation';

RUN;

/*  STEP 3 : Create another copy of the dataset as MODE table*/

PROC SQL;

CREATE TABLE LIBFT553.TESTING_COPY_DS_EMPIMP_MODE AS
SELECT e.EMPLOYMENT, COUNT(*) AS COUNTS FROM LIBFT553.TESTING_COPY_DS_IMPUT_EMPLOY e
WHERE ( ( e.EMPLOYMENT IS NOT NULL ) OR
        ( e.EMPLOYMENT NE '' ) )
GROUP BY e.EMPLOYMENT;

QUIT;


/*  STEP 4 : IMPUTE MISSING VALUES  */

PROC SQL;

UPDATE LIBFT553.TESTING_COPY_DS_IMPUT_EMPLOY
SET EMPLOYMENT = ( SELECT eo.EMPLOYMENT
                        FROM LIBFT553.TESTING_COPY_DS_EMPIMP_MODE eo
                        WHERE eo.COUNTS EQ ( SELECT MAX(e.COUNTS)
                                    FROM LIBFT553.TESTING_COPY_DS_EMPIMP_MODE e ) )
WHERE ( ( EMPLOYMENT IS NULL ) OR
        ( EMPLOYMENT EQ '' ) );

QUIT;


/*  STEP 5 : AFTER IMPUTATION   */

PROC FREQ DATA = LIBFT553.TESTING_COPY_DS_IMPUT_EMPLOY;

TABLE EMPLOYMENT;
TITLE 'AFTER IMPUTATION';

QUIT;



/*********************************************
********        LOAN_HISTORY    ***************
*********************************************/


/* STEP 1 : Make a copy of the testing dataset using final dataset after imputing EMPLOYMENT */
```

```
PROC SQL;

CREATE TABLE LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS AS
SELECT * FROM LIBFT553.TESTING_COPY_DS_IMPUT_EMPLOY;

QUIT;


/*  STEP 2 : Univariate analysis done for LOAN_HISTORY variable before imputation */

PROC FREQ DATA = LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS;

TABLE LOAN_HISTORY;
TITLE 'Before Imputation';

RUN;

/*  STEP 3 : Create another copy of the dataset as MODE table*/

PROC SQL;

CREATE TABLE LIBFT553.TESTING_COPY_DS_LOANHIS_MODE AS
SELECT e.LOAN_HISTORY, COUNT(*) AS COUNTS FROM LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS e
WHERE ( ( e.LOAN_HISTORY IS NOT NULL ) )
GROUP BY e.LOAN_HISTORY;

QUIT;


/*  STEP 4 : IMPUTE MISSING VALUES  */

PROC SQL;

UPDATE LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS
SET LOAN_HISTORY = ( SELECT eo.LOAN_HISTORY
                        FROM LIBFT553.TESTING_COPY_DS_LOANHIS_MODE eo
                        WHERE eo.COUNTS EQ ( SELECT MAX(e.COUNTS)
                                    FROM LIBFT553.TESTING_COPY_DS_LOANHIS_MODE e ) )
WHERE ( ( LOAN_HISTORY IS NULL ) );

QUIT;


/*  STEP 5 : AFTER IMPUTATION   */

PROC FREQ DATA = LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS;

TABLE LOAN_HISTORY;
TITLE 'AFTER IMPUTATION';

QUIT;




/*************************************************************************************************************************
*************************************************  IMPUTE CONTINUOUS VARIABLES - TESTING DATA      *********
*************************************************************************************************************************


/*********************************************
********        LOAN_DURATION    ***************
*********************************************/

/*  STEP 1: Univariate analysis of LOAN_DURATION variable before imputation     */

PROC MEANS DATA =  LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS N NMISS MIN MAX MEAN MEDIAN STD;

VAR LOAN_DURATION;
TITLE 'Univariate analysis of LOAN_DURATION variable - Before imputation';

RUN;


/* STEP 2 : Cleanse/impute missing valeus found in the LOAN_DURATION variable   */

PROC STDIZE DATA = LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS REPONLY
```

```
METHOD = MEAN OUT = LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS;
VAR LOAN_DURATION;

QUIT;


/*  STEP 3: Univariate analysis of LOAN_DURATION variable after imputation  */

PROC MEANS DATA =  LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS N NMISS MIN MAX MEAN MEDIAN STD;

VAR LOAN_DURATION;
TITLE 'Univariate analysis of LOAN_DURATION variable - After imputation';

RUN;



/*********************************************
********        LOAN_AMOUNT ********************
*********************************************/

/*  STEP 1: Univariate analysis of LOAN_AMOUNT variable before imputation   */

PROC MEANS DATA =  LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS N NMISS MIN MAX MEAN MEDIAN STD;

VAR LOAN_AMOUNT;
TITLE 'Univariate analysis of LOAN_AMOUNT variable - Before imputation';

RUN;


/* STEP 2 : Cleanse/impute missing valeus found in the LOAN_AMOUNT variable     */

PROC STDIZE DATA = LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS REPONLY

METHOD = MEAN OUT = LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS;
VAR LOAN_AMOUNT;

QUIT;


/*  STEP 3: Univariate analysis of LOAN_AMOUNT variable after imputation    */

PROC MEANS DATA =  LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS N NMISS MIN MAX MEAN MEDIAN STD;

VAR LOAN_AMOUNT;
TITLE 'Univariate analysis of LOAN_AMOUNT variable - After imputation';

RUN;




/*****************************************************************************************************************
*************************************************                    BUILDING LOGISTIC REGRESSION MODEL      *************
*****************************************************************************************************************/


/*      Checking if training dataset still has missing data         */

/*      CATEGORICAL VARIABLES       */
PROC FREQ DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS;

TABLE

LOAN_HISTORY
LOAN_LOCATION
GENDER
FAMILY_MEMBERS
MARITAL_STATUS
EMPLOYMENT
QUALIFICATION;

RUN;
```

```
/*  CONTINUOUS VARIABLES    */
PROC MEANS DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS N NMISS;

VAR LOAN_AMOUNT
    CANDIDATE_INCOME
    GUARANTEE_INCOME
    LOAN_DURATION;

RUN;


/*      Checking if testing dataset still has missing data          */

PROC FREQ DATA = LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS;

TABLE

LOAN_HISTORY
LOAN_LOCATION
GENDER
FAMILY_MEMBERS
MARITAL_STATUS
EMPLOYMENT
QUALIFICATION;

RUN;


/*  CONTINUOUS VARIABLES    */
PROC MEANS DATA = LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS N NMISS;

VAR LOAN_AMOUNT
    CANDIDATE_INCOME
    GUARANTEE_INCOME
    LOAN_DURATION;

RUN;


/*  BUILDING LOGISTIC REGRESSION MODEL  */

PROC LOGISTIC DATA = LIBFT553.TRAINING_COPY_DS_IMPUT_LOANHIS OUTMODEL = LIBFT553.TP041553_LRMODEL_ASSIGNMENT;
CLASS

LOAN_HISTORY
LOAN_LOCATION
GENDER
FAMILY_MEMBERS
MARITAL_STATUS
EMPLOYMENT
QUALIFICATION;
/*  These are categorical variables */
MODEL LOAN_APPROVAL_STATUS =                /*      List all independant variables below this   */

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

OUTPUT OUT = LIBFT553.OUTPUT_TRAININGDS P = PREDICTION_PROBABILITY;
        /*  OUTPUT OUT = Dataset to hold the predicted values for Dependent variable
            P is predicted probability which s a variable that holds predicted probability.
            Akaike Inforation criterion (AIC) must be less than S (Schwarz criterion)

            Pr > chiSq should be <= 0.05, this means independant variable is important*/

RUN ;
```

```
/*  To display the output dataset from logistic regression model including prediction probabilities */

PROC SQL;

SELECT * FROM LIBFT553.OUTPUT_TRAININGDS;

QUIT;


/*  Display contents of LR model created    */

PROC SQL;

SELECT * FROM LIBFT553.TP041553_LRMODEL_ASSIGNMENT;

QUIT;



/*  USE crated model to evaluate loan approval status in testing dataset    */
PROC LOGISTIC INMODEL = LIBFT553.TP041553_LRMODEL_ASSIGNMENT;   /* Created Model */

SCORE DATA = LIBFT553.TESTING_COPY_DS_IMPUT_LOANHIS             /* Testing dataset after imputation */
OUT = LIBFT553.TESTING_DATA_PREDICTIONS ;              /* Output / results location    */

QUIT;


/*  To display the predicted approval status of loans in testing dataset    */
PROC SQL;

SELECT  SME_LOAN_ID_NO LABEL = 'LOAN ID',
        I_LOAN_APPROVAL_STATUS LABEL = 'LOAN APPROVAL STATUS',
        P_N LABEL = 'Probability rejected',
        P_Y LABEL = 'Probability approved'

FROM LIBFT553.TESTING_DATA_PREDICTIONS;
RUN;


/*      To identify physical location of the LIBFT553 library   */

PROC DATASETS LIBRARY = LIBFT553 memtype = data;

RUN;



/*  ODS - Output Delivery System    */

ODS HTML CLOSE;
ODS PDF CLOSE;

/*  Identify the physical location of pdf file to be downladed  */
ODS PDF FILE = '/home/u49980108/LOAN_APPLICATIONS_ASSIGNMENT_REPORT.pdf';
OPTIONS NOBYLINE NODATE;
TITLE ' Loan approval status predictions on testing dataset ';


PROC REPORT DATA = LIBFT553.TESTING_DATA_PREDICTIONS NOWINDOWS;
```