

```

/*IMPORT TRAINING DATA */
PROC IMPORT DATAFILE="/home/u64255585/EPG1V2/loan prediction/TRAINING_DS.csv"
DBMS=CSV
OUT=TRAINING_DS
REPLACE;
GETNAMES=YES;
RUN;

/* IMPORT TESTING DATA */

PROC IMPORT DATAFILE="/home/u64255585/EPG1V2/loan prediction/TESTING_DS.csv"
DBMS=CSV
OUT=TESTING_DS
REPLACE;
GETNAMES=YES;
RUN;

PROC SQL;
CREATE TABLE TRAINING_DS_TP061966_BK AS
SELECT * FROM TRAINING_DS;
QUIT;

PROC SQL;
DESCRIBE TABLE TRAINING_DS_TP061966_BK;
QUIT;

DATA TRAINING_DS_TP061966_BK;
SET TRAINING_DS_TP061966_BK;
LABEL SME_LOAN_ID_NO = 'Loan Application Number'
GENDER = 'Gender Name'
MARITAL_STATUS = 'Marital Status'
FAMILY_MEMBERS = 'Family Members'
QUALIFICATION = 'Qualification'
EMPLOYMENT = 'Employment'
CANDIDATE_INCOME = 'Candidate Income'
GUARANTEE_INCOME = 'Guarantee Income'
LOAN_AMOUNT = 'Loan Amount'
LOAN_DURATION = 'Loan Duration'
LOAN_HISTORY = 'Loan History'
LOAN_LOCATION = 'Loan Location'
LOAN_APPROVAL_STATUS = 'Loan Approval Status';
RUN;

PROC SQL;
SELECT * FROM TRAINING_DS_TP061966_BK;
QUIT;

PROC FREQ DATA = TRAINING_DS_TP061966_BK;
TABLE GENDER;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPlot DATA = TRAINING_DS_TP061966_BK;
VBAR GENDER;
RUN;

PROC FREQ DATA = TRAINING_DS_TP061966_BK;
TABLE MARITAL_STATUS;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPlot DATA = TRAINING_DS_TP061966_BK;
VBAR MARITAL_STATUS;
RUN;

PROC FREQ DATA = TRAINING_DS_TP061966_BK;
TABLE FAMILY_MEMBERS;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPlot DATA = TRAINING_DS_TP061966_BK;
VBAR FAMILY_MEMBERS;
RUN;

PROC FREQ DATA = TRAINING_DS_TP061966_BK;
TABLE QUALIFICATION;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;

```

```

PROC SGPLT DATA = TRAINING_DS_TP061966_BK;
VBAR QUALIFICATION;
RUN;

PROC FREQ DATA = TRAINING_DS_TP061966_BK;
TABLE EMPLOYMENT;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLT DATA = TRAINING_DS_TP061966_BK;
VBAR EMPLOYMENT;
RUN;

PROC FREQ DATA = TRAINING_DS_TP061966_BK;
TABLE LOAN_HISTORY;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLT DATA = TRAINING_DS_TP061966_BK;
VBAR LOAN_HISTORY;
RUN;

PROC FREQ DATA = TRAINING_DS_TP061966_BK;
TABLE LOAN_LOCATION;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLT DATA = TRAINING_DS_TP061966_BK;
VBAR LOAN_LOCATION;
RUN;

PROC FREQ DATA = TRAINING_DS_TP061966_BK;
TABLE LOAN_APPROVAL_STATUS;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLT DATA = TRAINING_DS_TP061966_BK;
VBAR LOAN_APPROVAL_STATUS;
RUN;

PROC MEANS DATA = TRAINING_DS_TP061966_BK N NMISS MIN MAX MEAN MEDIAN STD;
VAR CANDIDATE_INCOME;
RUN;
ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLT DATA = TRAINING_DS_TP061966_BK;
HISTOGRAM CANDIDATE_INCOME;
RUN;

PROC MEANS DATA = TRAINING_DS_TP061966_BK N NMISS MIN MAX MEAN MEDIAN STD;
VAR GUARANTEE_INCOME;
RUN;
ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLT DATA = TRAINING_DS_TP061966_BK;
HISTOGRAM GUARANTEE_INCOME;
RUN;

PROC MEANS DATA = TRAINING_DS_TP061966_BK N NMISS MIN MAX MEAN MEDIAN STD;
VAR LOAN_AMOUNT;
RUN;
ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLT DATA = TRAINING_DS_TP061966_BK;
HISTOGRAM LOAN_AMOUNT;
RUN;

PROC MEANS DATA = TRAINING_DS_TP061966_BK N NMISS MIN MAX MEAN MEDIAN STD;
VAR LOAN_DURATION;
RUN;
ODS GRAPHICS / RESET WIDTH =4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLT DATA = TRAINING_DS_TP061966_BK;
HISTOGRAM LOAN_DURATION;
RUN;

%MACRO MACRO_BIVA_CV_TP061966( DATASET_NAME, VARIABLE_1, VARIABLE_2, TITLE_1, TITLE_2);
PROC FREQ DATA = &DATASET_NAME;
TABLE &VARIABLE_1 * &VARIABLE_2 /
PLOTS = FREQPLOT( TWOWAY = STACKED SCALE = GROUPPCT );
TITLE &TITLE_1;
TITLE2 &TITLE_2;
RUN;
%MEND MACRO_BIVA_CV_TP061966;

```

```
%MACRO_BIVA_CV_TP061966(TRAINING_DS_TP061966_BK, GENDER, MARITAL_STATUS, 'Bivariate analysis', 'on GENDER (Categorical) Vs. M
%MACRO_BIVA_CV_TP061966(TRAINING_DS_TP061966_BK, FAMILY_MEMBERS, QUALIFICATION, 'Bivariate analysis', 'on FAMILY_MEMBERS (Cat
%MACRO_BIVA_CV_TP061966(TRAINING_DS_TP061966_BK, EMPLOYMENT, LOAN_HISTORY, 'Bivariate analysis', 'on EMPLOYMENT (Categorical)
%MACRO_BIVA_CV_TP061966(TRAINING_DS_TP061966_BK, LOAN_LOCATION, LOAN_APPROVAL_STATUS, 'Bivariate analysis', 'on LOAN_LOCATION
%MACRO_BIVA_CV_TP061966(TRAINING_DS_TP061966_BK, GENDER, LOAN_APPROVAL_STATUS, 'Bivariate analysis', 'on GENDER (Categorical)

PROC MEANS DATA = TRAINING_DS_TP061966_BK;
CLASS LOAN_APPROVAL_STATUS ;
VAR CANDIDATE_INCOME;
TITLE "Bivariate analysis on LOAN_APPROVAL_STATUS ( CHAR ) Versus CANDIDATE_INCOME ( NUMERIC )";
RUN;

PROC SGPLOT DATA = TRAINING_DS_TP061966_BK;
VBOX CANDIDATE_INCOME / CATEGORY = LOAN_APPROVAL_STATUS ;
TITLE "Bivariate analysis on LOAN_APPROVAL_STATUS ( CHAR ) Versus CANDIDATE_INCOME ( NUMERIC )";
RUN;

PROC MEANS DATA = TRAINING_DS_TP061966_BK;
CLASS LOAN_APPROVAL_STATUS ;
VAR GUARANTEE_INCOME;
TITLE "Bivariate analysis on LOAN_APPROVAL_STATUS ( CHAR ) Versus GUARANTEE_INCOME ( NUMERIC )";
RUN;

PROC SGPLOT DATA = TRAINING_DS_TP061966_BK;
VBOX GUARANTEE_INCOME / CATEGORY = LOAN_APPROVAL_STATUS ;
TITLE "Bivariate analysis on LOAN_APPROVAL_STATUS ( CHAR ) Versus GUARANTEE_INCOME ( NUMERIC )";
RUN;

PROC MEANS DATA = TRAINING_DS_TP061966_BK;
CLASS LOAN_APPROVAL_STATUS ;
VAR LOAN_AMOUNT;
TITLE "Bivariate analysis on LOAN_APPROVAL_STATUS ( CHAR ) Versus LOAN_AMOUNT ( NUMERIC )";
RUN;

PROC SGPLOT DATA = TRAINING_DS_TP061966_BK;
VBOX LOAN_AMOUNT / CATEGORY = LOAN_APPROVAL_STATUS ;
TITLE "Bivariate analysis on LOAN_APPROVAL_STATUS ( CHAR ) Versus LOAN_AMOUNT ( NUMERIC )";
RUN;

PROC CORR DATA = TRAINING_DS_TP061966_BK PLOTS = SCATTER;
VAR CANDIDATE_INCOME GUARANTEE_INCOME;
ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;
TITLE "Bivariate analysis on CANDIDATE_INCOME ( NUMERIC ) Versus GUARANTEE_INCOME ( NUMERIC )";
QUIT;

PROC CORR DATA = TRAINING_DS_TP061966_BK PLOTS = SCATTER;
VAR LOAN_AMOUNT LOAN_DURATION;
ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;
TITLE "Bivariate analysis on LOAN_AMOUNT ( NUMERIC ) Versus LOAN_DURATION ( NUMERIC )";
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_GENDER_DS_BK AS
SELECT * FROM TRAINING_DS_TP061966_BK;
QUIT;

PROC SQL;
SELECT *
FROM TRAINING_FI_GENDER_DS_BK g
WHERE ( ( g.gender EQ '' ) OR
       ( g.gender IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_GENDER_DS_BK g
WHERE ( ( g.gender EQ '' ) OR
       ( g.gender IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_GENDER_DS_BK_NEW AS
SELECT g.GENDER AS GENDER_NAME, COUNT(*) AS TOTAL_COUNTS
FROM TRAINING_FI_GENDER_DS_BK g
WHERE ( ( g.GENDER IS NOT NULL ) OR
       ( g.GENDER NE '' ) )
GROUP BY g.GENDER;
```

```
QUIT;
```

```
PROC SQL;
SELECT * FROM TRAINING_FI_GENDER_DS_BK_NEW;
QUIT;
```

```
PROC SQL;
SELECT MAX(g.TOTAL_COUNTS) AS MAX_COUNTS
FROM TRAINING_FI_GENDER_DS_BK_NEW g;
QUIT;
```

```
PROC SQL;
SELECT go.GENDER_NAME
FROM TRAINING_FI_GENDER_DS_BK_NEW go
WHERE go.TOTAL_COUNTS EQ ( SELECT MAX(gi.TOTAL_COUNTS)
                           FROM TRAINING_FI_GENDER_DS_BK_NEW gi );
QUIT;
```

```
PROC SQL;
UPDATE TRAINING_FI_GENDER_DS_BK
SET GENDER = ( SELECT go.GENDER_NAME
               FROM TRAINING_FI_GENDER_DS_BK_NEW go
               WHERE go.TOTAL_COUNTS EQ ( SELECT MAX(gi.TOTAL_COUNTS)
                                         FROM TRAINING_FI_GENDER_DS_BK_NEW gi ) )
WHERE ( ( GENDER IS NULL) OR
       ( GENDER EQ '' ) );
QUIT;
```

```
PROC SQL;
SELECT *
FROM TRAINING_FI_GENDER_DS_BK g
WHERE ( ( g.gender EQ '' ) OR
       ( g.gender IS NULL ) );
QUIT;
```

```
PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_GENDER_DS_BK g
WHERE ( ( g.gender EQ '' ) OR
       ( g.gender IS NULL ) );
QUIT;
```

```
PROC SQL;
CREATE TABLE TRAINING_FI_FAMILY_MEMBERS_DS_BK AS
SELECT * FROM TRAINING_FI_GENDER_DS_BK;
QUIT;
```

```
PROC SQL;
SELECT *
FROM TRAINING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( ( fm.FAMILY_MEMBERS EQ '' ) OR
       ( fm.FAMILY_MEMBERS IS NULL ) );
QUIT;
```

```
PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_FAMILY_MEMBERS_DS_BK FM
WHERE ( ( fm.FAMILY_MEMBERS EQ '' ) OR
       ( fm.FAMILY_MEMBERS IS NULL ) );
QUIT;
```

```
PROC SQL;
CREATE TABLE TRAINING_FI_FM_DS_BK_N AS
SELECT fm.FAMILY_MEMBERS AS FAMILY_MEMBERS, COUNT(*) AS TOTAL_COUNTS
FROM TRAINING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( ( fm.FAMILY_MEMBERS IS NOT NULL ) OR
       ( fm.FAMILY_MEMBERS NE '' ) )
GROUP BY fm.FAMILY_MEMBERS;
QUIT;
```

```
PROC SQL;
SELECT * FROM TRAINING_FI_FM_DS_BK_N;
QUIT;
```

```
PROC SQL;
SELECT MAX(fm.TOTAL_COUNTS) AS MAX_COUNTS
```

```

FROM TRAINING_FI_FM_DS_BK_N fm;
QUIT;

PROC SQL;
SELECT fmo.FAMILY_MEMBERS
FROM TRAINING_FI_FM_DS_BK_N fmo
WHERE fmo.TOTAL_COUNTS EQ ( SELECT MAX(fmi.TOTAL_COUNTS)
                           FROM TRAINING_FI_FM_DS_BK_N fmi);
QUIT;

PROC SQL;
UPDATE TRAINING_FI_FAMILY_MEMBERS_DS_BK
SET FAMILY_MEMBERS = ( SELECT fmo.FAMILY_MEMBERS
                       FROM TRAINING_FI_FM_DS_BK_N fmo
                       WHERE fmo.TOTAL_COUNTS EQ ( SELECT MAX(fmi.TOTAL_COUNTS)
                           FROM TRAINING_FI_FM_DS_BK_N fmi ) )
WHERE ( ( FAMILY_MEMBERS IS NULL) OR
       ( FAMILY_MEMBERS EQ '' ) );
QUIT;

PROC SQL;
SELECT *
FROM TRAINING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( ( fm.FAMILY_MEMBERS EQ '' ) OR
       ( fm.FAMILY_MEMBERS IS NULL ) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( ( fm.FAMILY_MEMBERS EQ '' ) OR
       ( fm.FAMILY_MEMBERS IS NULL ) );
QUIT;

PROC SQL;
SELECT fm.FAMILY_MEMBERS,
       SUBSTR(fm.FAMILY_MEMBERS,1,1),
       SUBSTR(fm.FAMILY_MEMBERS,2,1)
FROM TRAINING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( SUBSTR(fm.FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'Total Observation'
FROM TRAINING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( SUBSTR(fm.FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;

PROC SQL;
UPDATE TRAINING_FI_FAMILY_MEMBERS_DS_BK
SET FAMILY_MEMBERS = SUBSTR(FAMILY_MEMBERS,1,1)
WHERE ( SUBSTR(FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;

PROC SQL;
SELECT fm.FAMILY_MEMBERS LABEL = 'FAMILY_MEMBERS',
       COUNT(*) LABEL = 'Number of Applicants'
FROM TRAINING_FI_FAMILY_MEMBERS_DS_BK fm
GROUP BY fm.FAMILY_MEMBERS;
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_MARITAL_STATUS_DS_BK AS
SELECT * FROM TRAINING_FI_FAMILY_MEMBERS_DS_BK;
QUIT;

PROC SQL;
SELECT *
FROM TRAINING_FI_MARITAL_STATUS_DS_BK ms
WHERE ( ( ms.MARITAL_STATUS EQ '' ) OR
       ( ms.MARITAL_STATUS IS NULL ) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_MARITAL_STATUS_DS_BK ms
WHERE ( ( ms.MARITAL_STATUS EQ '' ) OR
       ( ms.MARITAL_STATUS IS NULL ) );

```

```

( ms.MARITAL_STATUS IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_MS_DS_BK_NEW AS
SELECT ms.MARITAL_STATUS AS MARITAL_STATUS, COUNT(*) AS TOTAL_COUNTS
FROM TRAINING_FI_MARITAL_STATUS_DS_BK ms
WHERE ( ( ms.MARITAL_STATUS IS NOT NULL ) OR
        ( ms.MARITAL_STATUS NE '' ) )
GROUP BY ms.MARITAL_STATUS;
QUIT;

PROC SQL;
SELECT * FROM TRAINING_FI_MS_DS_BK_NEW;
QUIT;

PROC SQL;
SELECT MAX(ms.TOTAL_COUNTS) AS MAX_COUNTS
FROM TRAINING_FI_MS_DS_BK_NEW ms;
QUIT;

PROC SQL;
SELECT mso.MARITAL_STATUS
FROM TRAINING_FI_MS_DS_BK_NEW mso
WHERE mso.TOTAL_COUNTS EQ ( SELECT MAX(msi.TOTAL_COUNTS)
                           FROM TRAINING_FI_MS_DS_BK_NEW msi );
QUIT;

PROC SQL;
UPDATE TRAINING_FI_MARITAL_STATUS_DS_BK
SET MARITAL_STATUS = ( SELECT mso.MARITAL_STATUS
                       FROM TRAINING_FI_MS_DS_BK_NEW mso
                       WHERE mso.TOTAL_COUNTS EQ ( SELECT MAX(msi.TOTAL_COUNTS)
                           FROM TRAINING_FI_MS_DS_BK_NEW msi ) )
WHERE ( ( MARITAL_STATUS IS NULL) OR
        ( MARITAL_STATUS EQ '' ) );
QUIT;

PROC SQL;
SELECT *
FROM TRAINING_FI_MARITAL_STATUS_DS_BK ms
WHERE ( ( ms.MARITAL_STATUS EQ '' ) OR
        ( ms.MARITAL_STATUS IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT (*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_MARITAL_STATUS_DS_BK ms
WHERE ( ( ms.MARITAL_STATUS EQ '' ) OR
        ( ms.MARITAL_STATUS IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_EMPLOYMENT_DS_BK AS
SELECT * FROM TRAINING_FI_MARITAL_STATUS_DS_BK;
QUIT;

PROC SQL;
SELECT *
FROM TRAINING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT EQ '' ) OR
        ( e.EMPLOYMENT IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT (*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT EQ '' ) OR
        ( e.EMPLOYMENT IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_EMPLOYMENT_DS_BK_NEW AS
SELECT e.EMPLOYMENT AS EMPLOYMENT, COUNT(*) AS TOTAL_COUNTS
FROM TRAINING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT IS NOT NULL ) OR
        ( e.EMPLOYMENT NE '' ) )

```

```

GROUP BY e.EMPLOYMENT;
QUIT;

PROC SQL;
SELECT * FROM TRAINING_FI_EMPLOYMENT_DS_BK_NEW;
QUIT;

PROC SQL;
SELECT MAX(e.TOTAL_COUNTS) AS MAX_COUNTS
FROM TRAINING_FI_EMPLOYMENT_DS_BK_NEW e;
QUIT;

PROC SQL;
SELECT eo.EMPLOYMENT
FROM TRAINING_FI_EMPLOYMENT_DS_BK_NEW eo
WHERE eo.TOTAL_COUNTS EQ ( SELECT MAX(ei.TOTAL_COUNTS)
                           FROM TRAINING_FI_EMPLOYMENT_DS_BK_NEW ei );
QUIT;

PROC SQL;
UPDATE TRAINING_FI_EMPLOYMENT_DS_BK
SET EMPLOYMENT = ( SELECT eo.EMPLOYMENT
                    FROM TRAINING_FI_EMPLOYMENT_DS_BK_NEW eo
                    WHERE eo.TOTAL_COUNTS EQ ( SELECT MAX(ei.TOTAL_COUNTS)
                                               FROM TRAINING_FI_EMPLOYMENT_DS_BK_NEW ei ) )
WHERE ( ( EMPLOYMENT IS NULL) OR
       ( EMPLOYMENT EQ '' ) );
QUIT;

PROC SQL;
SELECT *
FROM TRAINING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT EQ '' ) OR
       ( e.EMPLOYMENT IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT EQ '' ) OR
       ( e.EMPLOYMENT IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_LOAN_HISTORY_DS_BK AS
SELECT * FROM TRAINING_FI_EMPLOYMENT_DS_BK;
QUIT;

PROC SQL;
SELECT *
FROM TRAINING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.LOAN_HISTORY EQ . ) OR
       ( lh.LOAN_HISTORY IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.LOAN_HISTORY EQ . ) OR
       ( lh.LOAN_HISTORY IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_LH_DS_BK_NEW AS
SELECT lh.LOAN_HISTORY AS LOAN_HISTORY, COUNT(*) AS TOTAL_COUNTS
FROM TRAINING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.LOAN_HISTORY IS NOT NULL ) OR
       ( lh.LOAN_HISTORY NE . ) )
GROUP BY lh.LOAN_HISTORY;
QUIT;

PROC SQL;
SELECT * FROM TRAINING_FI_LH_DS_BK_NEW;
QUIT;

PROC SQL;
SELECT MAX(lh.TOTAL_COUNTS) AS MAX_COUNTS

```

```

FROM TRAINING_FI_LH_DS_BK_NEW lh;
QUIT;

PROC SQL;
SELECT lho.POAN_HISTORY
FROM TRAINING_FI_LH_DS_BK_NEW lho
WHERE lho.TOTAL_COUNTS EQ ( SELECT MAX(lhi.TOTAL_COUNTS)
                           FROM TRAINING_FI_LH_DS_BK_NEW lhi );
QUIT;

PROC SQL;
UPDATE TRAINING_FI_LOAN_HISTORY_DS_BK
SET LOAN_HISTORY = ( SELECT lho.POAN_HISTORY
                      FROM TRAINING_FI_LH_DS_BK_NEW lho
                      WHERE lho.TOTAL_COUNTS EQ ( SELECT MAX(lhi.TOTAL_COUNTS)
                           FROM TRAINING_FI_LH_DS_BK_NEW lhi ) )
WHERE ( ( LOAN_HISTORY IS NULL) OR
       ( LOAN_HISTORY EQ . ) );
QUIT;

PROC SQL;
SELECT *
FROM TRAINING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.POAN_HISTORY EQ . ) OR
       ( lh.POAN_HISTORY IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TRAINING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.POAN_HISTORY EQ . ) OR
       ( lh.POAN_HISTORY IS NULL) );
QUIT;

PROC SQL;
SELECT * FROM TRAINING_FI_LOAN_HISTORY_DS_BK la
WHERE ( la.POAN_AMOUNT EQ . );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'Total Missing Values'
FROM TRAINING_FI_LOAN_HISTORY_DS_BK la
WHERE ( la.POAN_AMOUNT EQ . );
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_LOAN_AMOUNT_DS_BK AS
SELECT * FROM TRAINING_FI_LOAN_HISTORY_DS_BK;
QUIT;

PROC STDIZE DATA = TRAINING_FI_LOAN_AMOUNT_DS_BK REONLY
METHOD = MEAN OUT=TRAINING_FI_LOAN_AMOUNT_DS_BK;
VAR POAN_AMOUNT;
QUIT;

PROC SQL;
SELECT * FROM TRAINING_FI_LOAN_AMOUNT_DS_BK ld
WHERE ( ld.POAN_DURATION EQ . );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'Total Missing Values'
FROM TRAINING_FI_LOAN_AMOUNT_DS_BK ld
WHERE ( ld.POAN_DURATION EQ . );
QUIT;

PROC SQL;
CREATE TABLE TRAINING_FI_LOAN_DURATION_DS_BK AS
SELECT * FROM TRAINING_FI_LOAN_AMOUNT_DS_BK;
QUIT;

PROC STDIZE DATA = TRAINING_FI_LOAN_DURATION_DS_BK REONLY
METHOD = MEAN OUT=TRAINING_FI_LOAN_DURATION_DS_BK;
VAR POAN_DURATION;
QUIT;

PROC SQL;

```

```

CREATE TABLE TESTING_DS_TP061966_BK AS
SELECT * FROM TESTING_DS;
QUIT;

%MACRO MACRO_UVA_CatV_TP061966_TESTING( DATASET_NAME, VARIABLE_1, TITLE_1, TITLE_2);
TITLE &TITLE_1;
TITLE2 &TITLE_2;
PROC FREQ DATA = &DATASET_NAME;
TABLE &VARIABLE_1;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = &DATASET_NAME;
VBAR &VARIABLE_1;
RUN;
%MEND MACRO_UVA_CatV_TP061966_TESTING;

%MACRO_UVA_CatV_TP061966_TESTING(TESTING_DS_TP061966_BK, GENDER, 'Univariate analysis', 'on GENDER (Categorical Variable');
%MACRO_UVA_CatV_TP061966_TESTING(TESTING_DS_TP061966_BK, MARITAL_STATUS, 'Univariate analysis', 'on MARITAL_STATUS (Categorical Variable');
%MACRO_UVA_CatV_TP061966_TESTING(TESTING_DS_TP061966_BK, FAMILY_MEMBERS, 'Univariate analysis', 'on FAMILY_MEMBERS (Categorical Variable');
%MACRO_UVA_CatV_TP061966_TESTING(TESTING_DS_TP061966_BK, QUALIFICATION, 'Univariate analysis', 'on QUALIFICATION (Categorical Variable');
%MACRO_UVA_CatV_TP061966_TESTING(TESTING_DS_TP061966_BK, EMPLOYMENT, 'Univariate analysis', 'on EMPLOYMENT (Categorical Variable');
%MACRO_UVA_CatV_TP061966_TESTING(TESTING_DS_TP061966_BK, LOAN_HISTORY, 'Univariate analysis', 'on LOAN_HISTORY (Categorical Variable');
%MACRO_UVA_CatV_TP061966_TESTING(TESTING_DS_TP061966_BK, LOAN_LOCATION, 'Univariate analysis', 'on LOAN_LOCATION (Categorical Variable');

%MACRO MACRO_UVA_ConV_TP061966_TESTING( DATASET_NAME, VARIABLE_1, TITLE_1, TITLE_2);
PROC MEANS DATA = &DATASET_NAME N NMISS MIN MAX MEAN MEDIAN STD;
VAR &VARIABLE_1;
TITLE &TITLE_1;
TITLE2 &TITLE_2;
RUN;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;
PROC SGPLOT DATA = &DATASET_NAME;
HISTOGRAM &VARIABLE_1;
RUN;
%MEND MACRO_UVA_ConV_TP061966_TESTING;

%MACRO_UVA_ConV_TP061966_TESTING(TESTING_DS_TP061966_BK, CANDIDATE_INCOME, 'Univariate analysis', 'on CANDIDATE_INCOME (Continuous Variable');
%MACRO_UVA_ConV_TP061966_TESTING(TESTING_DS_TP061966_BK, GUARANTEE_INCOME, 'Univariate analysis', 'on GUARANTEE_INCOME (Continuous Variable');
%MACRO_UVA_ConV_TP061966_TESTING(TESTING_DS_TP061966_BK, LOAN_AMOUNT, 'Univariate analysis', 'on LOAN_AMOUNT (Continuous Variable');
%MACRO_UVA_ConV_TP061966_TESTING(TESTING_DS_TP061966_BK, LOAN_DURATION, 'Univariate analysis', 'on LOAN_DURATION (Continuous Variable');

%MACRO MACRO_BIVA_Cat_Testing_TP061966( DATASET_NAME, VARIABLE_1, VARIABLE_2, TITLE_1, TITLE_2);
PROC FREQ DATA = &DATASET_NAME;
TABLE &VARIABLE_1 * &VARIABLE_2 /
PLOTS = FREQPLOT( TWOWAY = STACKED SCALE = GROUPPCT );
TITLE &TITLE_1;
TITLE2 &TITLE_2;
RUN;
%MEND MACRO_BIVA_Cat_Testing_TP061966;

%MACRO_BIVA_Cat_Testing_TP061966(TESTING_DS_TP061966_BK, GENDER, MARITAL_STATUS, 'Bivariate analysis', 'on GENDER (Categorical Variable');
%MACRO_BIVA_Cat_Testing_TP061966(TESTING_DS_TP061966_BK, FAMILY_MEMBERS, QUALIFICATION, 'Bivariate analysis', 'on FAMILY_MEMBERS (Categorical Variable');
%MACRO_BIVA_Cat_Testing_TP061966(TESTING_DS_TP061966_BK, EMPLOYMENT, LOAN_HISTORY, 'Bivariate analysis', 'on EMPLOYMENT (Categorical Variable');
%MACRO_BIVA_Cat_Testing_TP061966(TESTING_DS_TP061966_BK, GENDER, QUALIFICATION, 'Bivariate analysis', 'on GENDER (Categorical Variable');
%MACRO_BIVA_Cat_Testing_TP061966(TESTING_DS_TP061966_BK, GENDER, LOAN_LOCATION, 'Bivariate analysis', 'on GENDER (Categorical Variable');
%MACRO_BIVA_Cat_Testing_TP061966(TESTING_DS_TP061966_BK, FAMILY_MEMBERS, LOAN_LOCATION, 'Bivariate analysis', 'on FAMILY_MEMBERS (Categorical Variable');

%MACRO MACRO_BIVA_CatVCon_Test_TP061966( DATASET_NAME, VARIABLE_1, VARIABLE_2, TITLE_1);
PROC MEANS DATA = &DATASET_NAME;
CLASS &VARIABLE_1;
VAR &VARIABLE_2;
TITLE &TITLE_1;
RUN;
PROC SGPLOT DATA = &DATASET_NAME;
VBOX &VARIABLE_2 / CATEGORY = &VARIABLE_1;
TITLE &TITLE_1;
RUN;
%MEND MACRO_BIVA_CatVCon_Test_TP061966;

%MACRO_BIVA_CatVCon_Test_TP061966(TESTING_DS_TP061966_BK, GENDER, CANDIDATE_INCOME, 'Bivariate analysis on GENDER ( CHAR ) Variable');
%MACRO_BIVA_CatVCon_Test_TP061966(TESTING_DS_TP061966_BK, MARITAL_STATUS, GUARANTEE_INCOME, 'Bivariate analysis on MARITAL_STATUS ( CHAR ) Variable');
%MACRO_BIVA_CatVCon_Test_TP061966(TESTING_DS_TP061966_BK, EMPLOYMENT, LOAN_AMOUNT, 'Bivariate analysis on EMPLOYMENT ( CHAR ) Variable');

%MACRO MACRO_BIVA_ConVCon_Test_TP061966( DATASET_NAME, VARIABLE_1, VARIABLE_2, TITLE_1);
PROC CORR DATA = &DATASET_NAME PLOTS = SCATTER;
VAR &VARIABLE_1 &VARIABLE_2;
ODS GRAPHICS / RESET WIDTH=4.0 IN HEIGHT=3.0 IN IMAGEMAP;

```

```

TITLE &TITLE_1;
QUIT;
%MEND MACRO_BIVA_ConVCon_Test_TP061966;

%MACRO_BIVA_ConVCon_Test_TP061966(TESTING_DS_TP061966_BK, CANDIDATE_INCOME, GUARANTEE_INCOME, 'Bivariate analysis on CANDIDATE_INCOME');
%MACRO_BIVA_ConVCon_Test_TP061966(TESTING_DS_TP061966_BK, LOAN_AMOUNT, LOAN_DURATION, 'Bivariate analysis on LOAN_AMOUNT ( NUI)');
%MACRO_BIVA_ConVCon_Test_TP061966(TESTING_DS_TP061966_BK, CANDIDATE_INCOME, LOAN_DURATION, 'Bivariate analysis on CANDIDATE_INCOME');

PROC SQL;
CREATE TABLE TESTING_FI_GENDER_DS_BK AS
SELECT * FROM TESTING_DS_TP061966_BK;
QUIT;

PROC SQL;
SELECT *
FROM TESTING_FI_GENDER_DS_BK g
WHERE ( ( g.gender EQ '' ) OR
        ( g.gender IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TESTING_FI_GENDER_DS_BK g
WHERE ( ( g.gender EQ '' ) OR
        ( g.gender IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TESTING_FI_GENDER_DS_BK_NEW AS
SELECT g.GENDER AS GENDER_NAME, COUNT(*) AS TOTAL_COUNTS
FROM TESTING_FI_GENDER_DS_BK g
WHERE ( ( g.GENDER IS NOT NULL ) OR
        ( g.GENDER NE '' ) )
GROUP BY g.GENDER;
QUIT;

PROC SQL;
SELECT * FROM TESTING_FI_GENDER_DS_BK_NEW;
QUIT;

PROC SQL;
SELECT MAX(g.TOTAL_COUNTS) AS MAX_COUNTS
FROM TESTING_FI_GENDER_DS_BK_NEW g;
QUIT;

PROC SQL;
SELECT go.GENDER_NAME
FROM TESTING_FI_GENDER_DS_BK_NEW go
WHERE go.TOTAL_COUNTS EQ ( SELECT MAX(gi.TOTAL_COUNTS)
                           FROM TESTING_FI_GENDER_DS_BK_NEW gi );
QUIT;

PROC SQL;
UPDATE TESTING_FI_GENDER_DS_BK
SET GENDER = ( SELECT go.GENDER_NAME
               FROM TESTING_FI_GENDER_DS_BK_NEW go
               WHERE go.TOTAL_COUNTS EQ ( SELECT MAX(gi.TOTAL_COUNTS)
                                         FROM TESTING_FI_GENDER_DS_BK_NEW gi ) )
WHERE ( ( GENDER IS NULL) OR
        ( GENDER EQ '' ) );
QUIT;

PROC SQL;
SELECT *
FROM TESTING_FI_GENDER_DS_BK g
WHERE ( ( g.gender EQ '' ) OR
        ( g.gender IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TESTING_FI_GENDER_DS_BK g
WHERE ( ( g.gender EQ '' ) OR
        ( g.gender IS NULL) );
QUIT;

PROC SQL;

```

```

CREATE TABLE TESTING_FI_FAMILY_MEMBERS_DS_BK AS
SELECT * FROM TESTING_FI_GENDER_DS_BK;
QUIT;

PROC SQL;
SELECT *
FROM TESTING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( ( fm.FAMILY_MEMBERS EQ '' ) OR
        ( fm.FAMILY_MEMBERS IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TESTING_FI_FAMILY_MEMBERS_DS_BK FM
WHERE ( ( fm.FAMILY_MEMBERS EQ '' ) OR
        ( fm.FAMILY_MEMBERS IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TESTING_FI_FM_DS_BK_N AS
SELECT fm.FAMILY_MEMBERS AS FAMILY_MEMBERS, COUNT(*) AS TOTAL_COUNTS
FROM TESTING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( ( fm.FAMILY_MEMBERS IS NOT NULL ) OR
        ( fm.FAMILY_MEMBERS NE '' ) )
GROUP BY fm.FAMILY_MEMBERS;
QUIT;

PROC SQL;
SELECT * FROM TESTING_FI_FM_DS_BK_N;
QUIT;

PROC SQL;
SELECT MAX(fm.TOTAL_COUNTS) AS MAX_COUNTS
FROM TESTING_FI_FM_DS_BK_N fm;
QUIT;

PROC SQL;
SELECT fmo.FAMILY_MEMBERS
FROM TESTING_FI_FM_DS_BK_N fmo
WHERE fmo.TOTAL_COUNTS EQ ( SELECT MAX(fmi.TOTAL_COUNTS)
                            FROM TESTING_FI_FM_DS_BK_N fmi);
QUIT;

PROC SQL;
UPDATE TESTING_FI_FAMILY_MEMBERS_DS_BK
SET FAMILY_MEMBERS = ( SELECT fmo.FAMILY_MEMBERS
                       FROM TESTING_FI_FM_DS_BK_N fmo
                       WHERE fmo.TOTAL_COUNTS EQ ( SELECT MAX(fmi.TOTAL_COUNTS)
                                         FROM TESTING_FI_FM_DS_BK_N fmi ) )
WHERE ( ( FAMILY_MEMBERS IS NULL) OR
        ( FAMILY_MEMBERS EQ '' ) );
QUIT;

PROC SQL;
SELECT *
FROM TESTING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( ( fm.FAMILY_MEMBERS EQ '' ) OR
        ( fm.FAMILY_MEMBERS IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TESTING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( ( fm.FAMILY_MEMBERS EQ '' ) OR
        ( fm.FAMILY_MEMBERS IS NULL) );
QUIT;

PROC SQL;
SELECT fm.FAMILY_MEMBERS,
       SUBSTR(fm.FAMILY_MEMBERS,1,1),
       SUBSTR(fm.FAMILY_MEMBERS,2,1)
FROM TESTING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( SUBSTR(fm.FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'Total Observation'

```

```

FROM TESTING_FI_FAMILY_MEMBERS_DS_BK fm
WHERE ( SUBSTR(fm.FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;

PROC SQL;
UPDATE TESTING_FI_FAMILY_MEMBERS_DS_BK
SET FAMILY_MEMBERS = SUBSTR(FAMILY_MEMBERS,1,1)
WHERE ( SUBSTR(FAMILY_MEMBERS,2,1) EQ '+' );
QUIT;

PROC SQL;
SELECT fm.FAMILY_MEMBERS LABEL = 'FAMILY_MEMBERS',
COUNT(*) LABEL = 'Number of Applicants'
FROM TESTING_FI_FAMILY_MEMBERS_DS_BK fm
GROUP BY fm.FAMILY_MEMBERS;
QUIT;

PROC SQL;
CREATE TABLE TESTING_FI_EMPLOYMENT_DS_BK AS
SELECT * FROM TESTING_FI_FAMILY_MEMBERS_DS_BK;
QUIT;

PROC SQL;
SELECT *
FROM TESTING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT EQ '' ) OR
( e.EMPLOYMENT IS NULL ) );
QUIT;

PROC SQL;
SELECT COUNT(*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TESTING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT EQ '' ) OR
( e.EMPLOYMENT IS NULL ) );
QUIT;

PROC SQL;
CREATE TABLE TESTING_FI_EMPLOYMENT_DS_BK_NEW AS
SELECT e.EMPLOYMENT AS EMPLOYMENT, COUNT(*) AS TOTAL_COUNTS
FROM TESTING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT IS NOT NULL ) OR
( e.EMPLOYMENT NE '' ) )
GROUP BY e.EMPLOYMENT;
QUIT;

PROC SQL;
SELECT * FROM TESTING_FI_EMPLOYMENT_DS_BK_NEW;
QUIT;

PROC SQL;
SELECT MAX(e.TOTAL_COUNTS) AS MAX_COUNTS
FROM TESTING_FI_EMPLOYMENT_DS_BK_NEW e;
QUIT;

PROC SQL;
SELECT eo.EMPLOYMENT
FROM TESTING_FI_EMPLOYMENT_DS_BK_NEW eo
WHERE eo.TOTAL_COUNTS EQ ( SELECT MAX(ei.TOTAL_COUNTS)
                           FROM TESTING_FI_EMPLOYMENT_DS_BK_NEW ei );
QUIT;

PROC SQL;
UPDATE TESTING_FI_EMPLOYMENT_DS_BK
SET EMPLOYMENT = ( SELECT eo.EMPLOYMENT
                   FROM TESTING_FI_EMPLOYMENT_DS_BK_NEW eo
                   WHERE eo.TOTAL_COUNTS EQ ( SELECT MAX(ei.TOTAL_COUNTS)
                                              FROM TESTING_FI_EMPLOYMENT_DS_BK_NEW ei ) )
WHERE ( ( EMPLOYMENT IS NULL ) OR
( EMPLOYMENT EQ '' ) );
QUIT;

PROC SQL;
SELECT *
FROM TESTING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT EQ '' ) OR
( e.EMPLOYMENT IS NULL ) );
QUIT;

```

```

PROC SQL;
SELECT COUNT (*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TESTING_FI_EMPLOYMENT_DS_BK e
WHERE ( ( e.EMPLOYMENT EQ '' ) OR
       ( e.EMPLOYMENT IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TESTING_FI_LOAN_HISTORY_DS_BK AS
SELECT * FROM TESTING_FI_EMPLOYMENT_DS_BK;
QUIT;

PROC SQL;
SELECT *
FROM TESTING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.LOAN_HISTORY EQ . ) OR
       ( lh.LOAN_HISTORY IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT (*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TESTING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.LOAN_HISTORY EQ . ) OR
       ( lh.LOAN_HISTORY IS NULL) );
QUIT;

PROC SQL;
CREATE TABLE TESTING_FI_LH_DS_BK_NEW AS
SELECT lh.LOAN_HISTORY AS LOAN_HISTORY, COUNT(*) AS TOTAL_COUNTS
FROM TESTING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.LOAN_HISTORY IS NOT NULL ) OR
       ( lh.LOAN_HISTORY NE . ) )
GROUP BY lh.LOAN_HISTORY;
QUIT;

PROC SQL;
SELECT * FROM TESTING_FI_LH_DS_BK_NEW;
QUIT;

PROC SQL;
SELECT MAX(lh.TOTAL_COUNTS) AS MAX_COUNTS
FROM TESTING_FI_LH_DS_BK_NEW lh;
QUIT;

PROC SQL;
SELECT lho.LOAN_HISTORY
FROM TESTING_FI_LH_DS_BK_NEW lho
WHERE lho.TOTAL_COUNTS EQ ( SELECT MAX(lhi.TOTAL_COUNTS)
                           FROM TESTING_FI_LH_DS_BK_NEW lhi );
QUIT;

PROC SQL;
UPDATE TESTING_FI_LOAN_HISTORY_DS_BK
SET LOAN_HISTORY = ( SELECT lho.LOAN_HISTORY
                     FROM TESTING_FI_LH_DS_BK_NEW lho
                     WHERE lho.TOTAL_COUNTS EQ ( SELECT MAX(lhi.TOTAL_COUNTS)
                           FROM TESTING_FI_LH_DS_BK_NEW lhi ) )
WHERE ( ( LOAN_HISTORY IS NULL) OR
       ( LOAN_HISTORY EQ . ) );
QUIT;

PROC SQL;
SELECT *
FROM TESTING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.LOAN_HISTORY EQ . ) OR
       ( lh.LOAN_HISTORY IS NULL) );
QUIT;

PROC SQL;
SELECT COUNT (*) LABEL = 'NUMBER OF MISSING VALUES'
FROM TESTING_FI_LOAN_HISTORY_DS_BK lh
WHERE ( ( lh.LOAN_HISTORY EQ . ) OR
       ( lh.LOAN_HISTORY IS NULL) );
QUIT;

PROC SQL;

```

```
SELECT * FROM TESTING_FI_LOAN_HISTORY_DS_BK la
WHERE ( la.LOAN_AMOUNT EQ . );
QUIT;
```

---

```
PROC SQL;
SELECT COUNT(*) LABEL = 'Total Missing Values'
FROM TESTING_FI_LOAN_HISTORY_DS_BK la
WHERE ( la.LOAN_AMOUNT EQ . );
QUIT;
```

```
PROC SQL;
CREATE TABLE TESTING_FI_LOAN_AMOUNT_DS_BK AS
SELECT * FROM TESTING_FI_LOAN_HISTORY_DS_BK;
QUIT;
```

---

```
PROC STDIZE DATA = TESTING_FI_LOAN_AMOUNT_DS_BK REONLY
METHOD = MEAN OUT=TESTING_FI_LOAN_AMOUNT_DS_BK;
VAR LOAN_AMOUNT;
QUIT;
```

```
PROC SQL;
SELECT * FROM TESTING_FI_LOAN_AMOUNT_DS_BK ld
WHERE ( ld.LOAN_DURATION EQ . );
QUIT;
```

---

```
PROC SQL;
SELECT COUNT(*) LABEL = 'Total Missing Values'
FROM TESTING_FI_LOAN_AMOUNT_DS_BK ld
WHERE ( ld.LOAN_DURATION EQ . );
QUIT;
```

```
PROC SQL;
CREATE TABLE TESTING_FI_LOAN_DURATION_DS_BK AS
SELECT * FROM TESTING_FI_LOAN_AMOUNT_DS_BK;
QUIT;
```

---

```
PROC STDIZE DATA = TESTING_FI_LOAN_DURATION_DS_BK REONLY
METHOD = MEAN OUT=TESTING_FI_LOAN_DURATION_DS_BK;
VAR LOAN_DURATION;
QUIT;
```

---

```
PROC LOGISTIC DATA = TRAINING_FI_LOAN_DURATION_DS_BK OUTMODEL = TRAINING_DS_LRMODEL;
CLASS
GENDER
MARITAL_STATUS
FAMILY_MEMBERS
QUALIFICATION
EMPLOYMENT
LOAN_LOCATION;
MODEL LOAN_APPROVAL_STATUS =
GENDER
MARITAL_STATUS
FAMILY_MEMBERS
QUALIFICATION
EMPLOYMENT
CANDIDATE_INCOME
GUARANTEE_INCOME
LOAN_AMOUNT
LOAN_DURATION
LOAN_HISTORY
LOAN_LOCATION;
OUTPUT OUT = TRAINING_DS_LRMODEL_OUTPUT P = PRED_PROB;
RUN;
```

---

```
PROC SQL;
SELECT * FROM TRAINING_DS_LRMODEL_OUTPUT;
QUIT;
```

---

```
PROC SQL;
SELECT * FROM TRAINING_DS_LRMODEL;
QUIT;
```

---

```
PROC LOGISTIC INMODEL = TRAINING_DS_LRMODEL;
SCORE DATA = TESTING_FI_LOAN_DURATION_DS_BK
OUT = TESTING_DS_PREDICTIONS;
----
```