%include 'C:\Study\SAS\included.sas';

LIBNAME P4 "C:\Users\Shu\Dropbox\SAS project\Project data\project4";

\*read the data;

**DATA** P4;

SET "C:\Users\Shu\Dropbox\SAS project\Project data\project4\project4";

**RUN**;

\* Print the first 8 observations of the data set;

%***head***(P4)

%***tail***(P4, n = **5**)

\*List the highest and lowest 10 values of checking balance, together with account ages;

%***HighLowN***(Dat = P4, Var = DDABal, Idvar = AcctAge, n = **10**)

\*List the highest and lowest 5% values of checking balance, together with account ages;

%***HighLow\_nPercent***(Dat=P4, Var=DDABal, Percent=**5**, Idvar=AcctAge)

\*Calculate cell means;

%***CellMeans***(P4,Tmp,Res Branch,Age SavBal)

\* The summary statistics of the data set;

**PROC** **MEANS** DATA = P4 N SUM NMISS MIN MAX MEAN MEDIAN STD MAXDEC = **2**;

TITLE "Statistics summary for numeric variables";

VAR \_NUMERIC\_;

**RUN**;

**PROC** **SORT** DATA = P4;

BY Branch;

**RUN**;

**PROC** **MEANS** DATA = P4 N SUM NMISS MIN MAX MEDIAN STD MAXDEC = **2** NOPRINT;

CLASS Branch;

VAR DDA DDABal DirDep Sav SavBal CD CDBal IRA IRABal LOC LOCBal ILS ILSBal

MM MMBal MTG MTGBal;

OUTPUT OUT = ByBranchSum

N =

SUM =

NMISS =

MIN =

MAX =

MEDIAN =

STD = /AUTONAME;

**RUN**;

**PROC** **PRINT** DATA = ByBranchSum;

VAR Branch DDA\_N DDA\_Sum DDABal\_Sum;

**RUN**;

\* Calculate total number and percentages of total for checking account and checking balance;

**DATA** DDABranch;

SET ByBranchSum(KEEP = Branch DDA\_N DDA\_Sum DDABal\_Sum);

RETAIN DDATotal DDASumTotal DDABalTotal;

IF \_N\_ = **1** THEN DO;

DDATotal = DDA\_N;

DDASumTotal = DDA\_Sum;

DDABalTotal = DDABal\_Sum;

END;

Pct\_DDAN = DDA\_N/DDATotal;

Pct\_DDAS = DDA\_Sum/DDASumTotal;

Pct\_DDAB = DDABal\_Sum/DDABalTotal;

DROP DDATotal DDASumTotal DDABalTotal;

**RUN**;

**PROC** **PRINT** DATA = DDABranch;

VAR Branch Pct\_DDAN Pct\_DDAS Pct\_DDAB;

SUM;

**RUN**;

\* Calculate total number and percentages of total for saving account and saving balance;

**DATA** DDABranch2;

SET ByBranchSum(KEEP = Branch Sav\_N Sav\_Sum SavBal\_Sum);

RETAIN SavTotal SavSumTotal SavBalTotal;

IF \_N\_ = **1** THEN DO;

SavTotal = Sav\_N;

SavSumTotal = Sav\_Sum;

SavBalTotal = SavBal\_Sum;

END;

Pct\_SavN = Sav\_N/SavTotal;

Pct\_SavS = Sav\_Sum/SavSumTotal;

Pct\_SavB = SavBal\_Sum/SavBalTotal;

DROP SavTotal SavSumTotal SavBalTotal;

**RUN**;

**PROC** **EXPORT** DATA= WORK.DDABRANCH2

OUTFILE= "C:\Users\Shu\Dropbox\SAS project\Project data\project4\sav.csv"

DBMS=CSV REPLACE;

PUTNAMES=YES;

**RUN**;

**PROC** **PRINT** DATA = DDABranch2;

VAR Branch Pct\_SavN Pct\_SavS Pct\_SavB;

SUM;

**RUN**;

\* Calculate total number and percentages of total for retirement account and retirement account balance;

**DATA** DDABranch3;

SET ByBranchSum(KEEP = Branch IRA\_N IRA\_Sum IRABal\_Sum);

RETAIN IRATotal IRASumTotal IRABalTotal;

IF \_N\_ = **1** THEN DO;

IRATotal = IRA\_N;

IRASumTotal = IRA\_Sum;

IRABalTotal = IRABal\_Sum;

END;

Pct\_IRAN = IRA\_N/IRATotal;

Pct\_IRAS = IRA\_Sum/IRASumTotal;

Pct\_IRAB = IRABal\_Sum/IRABalTotal;

DROP IRATotal IRASumTotal IRABalTotal;

**RUN**;

**PROC** **EXPORT** DATA= WORK.DDABRANCH3

OUTFILE= "C:\Users\Shu\Dropbox\SAS project\Project data\project4\ira.csv"

DBMS=CSV REPLACE;

PUTNAMES=YES;

**RUN**;

**PROC** **PRINT** DATA = DDABranch3;

VAR Branch Pct\_IRAN Pct\_IRAS Pct\_IRAB;

SUM;

**RUN**;

\*Create a table for percentages;

**PROC** **SQL**;

CREATE TABLE P4.TotalAcct AS

SELECT c.Branch, c.Pct\_DDAS, c.Pct\_DDAB,

s.Pct\_SavS, s.Pct\_SavB,

r.Pct\_IRAS, r.Pct\_IRAB

FROM DDABranch c, DDABranch2 s, DDABranch3 r

WHERE c.Branch = s.Branch = r.Branch;

**QUIT**;

**DATA** TotalAcct;

SET P4.TotalAcct(RENAME = (Pct\_DDAS = Checking Pct\_DDAB = CheckingBalance Pct\_SavS = Saving Pct\_SavB = SavingBalance

Pct\_IRAS = Retirement Pct\_IRAB = RetireBalance));

IF \_N\_ = **1** THEN DELETE;

**RUN**;

\*Transpose data for making graphs;

**PROC** **TRANSPOSE** DATA = TotalAcct out = TotalBranch(RENAME = (\_NAME\_ = Account COL1 = Percentage));

by Branch;

**RUN**;

**DATA** TotalBranch;

SET TotalBranch;

BranchNum = INPUT(COMPRESS(Branch, "B"), **8.**);

LABEL Account = ' ';

FORMAT Percentage percent8.2;

**RUN**;

**PROC** **SORT** DATA = TotalBranch;

BY BranchNum;

**RUN**;

\*Produce the first graph;

**PROC** **SGPLOT** DATA = TotalBranch;

TITLE "Pecentge of Four Types of Accounts By Branch";

VBAR Branch /RESPONSE = Percentage group=Account groupdisplay=cluster

DATASKIN = crisp GROUPORDER = data;

XAXIS DISCRETEORDER = data;

YAXIS LABEL = "Percentage of Total" GRID;

**RUN**;

**DATA** BranchPanel;

SET TotalBranch;

**RUN**;

**PROC** **SORT** DATA = BranchPanel;

BY Account BranchNum;

**RUN**;

**PROC** **SGPANEL** DATA = BranchPanel; \*Produce the second graph;

TITLE "Percentage of Total by Account Types";

PANELBY Account;

VBAR Branch / RESPONSE = Percentage group = branch;

COLAXIS DISCRETEORDER = data;

ROWAXIS LABEL = "Percentage of Total" GRID;

**RUN**;

**PROC** **SORT** DATA = P4 OUT = ByBranch2;

BY Branch;

**RUN**;

**PROC** **MEANS** DATA = ByBranch2 MAXDEC =**2** NOPRINT;

VAR SavBal;

BY Branch;

OUTPUT OUT = SavBal (DROP = \_FREQ\_ \_TYPE\_)

MEAN = M\_Bal;

format M\_Bal dollar10.2;

**RUN**;

**PROC** **SORT** DATA = SavBal;

BY DESCENDING M\_Bal;

**RUN**;

**PROC** **SGPLOT** DATA = SavBal; \*Produce the third graph;

TITLE "Average Saving Balance By Branches";

HBAR Branch / RESPONSE = M\_Bal DATASKIN = gloss DATALABEL

CATEGORYORDER = respdesc;

XAXIS grid;

YAXIS DISCRETEORDER = data;

**RUN**;

**PROC** **MEANS** DATA = ByBranch2 MAXDEC =**2** NOPRINT;

VAR DDABal;

BY Branch;

OUTPUT OUT = DDABal (DROP = \_FREQ\_ \_TYPE\_)

MEAN = M\_DDABal;

format M\_DDABal dollar10.2;

**RUN**;

**PROC** **SORT** DATA = DDABal;

BY DESCENDING M\_DDABal;

**RUN**;

**PROC** **SGPLOT** DATA = DDABal; \*Produce the fourth graph;

TITLE "Average Checking Balance By Branches";

HBAR Branch / RESPONSE = M\_DDABal DATASKIN = gloss DATALABEL

CATEGORYORDER = respdesc;

XAXIS grid;

YAXIS DISCRETEORDER = data;

**RUN**;

\*Cluster analysis, not on the slides;

**data** P4clust;

set p4;

BranchNum = INPUT(COMPRESS(Branch, "B"), **8.**);

**RUN**;

**PROC** **FASTCLUS** DATA = P4clust OUT=aaa MAXC=**3** CLUSTER=c;

VAR DDA DDABal Sav SavBal BranchNum;

**RUN**;

**DATA** b1 b2 b3;

SET aaa;

IF c=**1** THEN OUTPUT b1;

IF c=**2** THEN OUTPUT b2;

IF c=**3** THEN OUTPUT b3;

**run**;

**PROC** **PRINT** DATA=b1;

**run**;

**PROC** **PRINT** DATA=b2;

**run**;

**PROC** **PRINT** DATA=b3;

**RUN**;

/\*delete observations with missing values\*/

**DATA** P4;

SET P4.Project4;

if nmiss(of \_NUMERIC\_)=**0** AND cmiss(of \_ALL\_)=**0**;

**RUN**;

/\*data partition\*/

**DATA** Train valid test;

SET P4;

RanNum = RANUNI(**12345**);

IF RanNum < **0.25** THEN OUTPUT test;

ELSE IF RanNum < **0.5** THEN OUTPUT valid;

ELSE OUTPUT Train;

DROP RanNum;

**RUN**;

\*First full model;

**PROC** **LOGISTIC** DATA = Train;

MODEL INS(EVENT = '1') = SavBal DDABal CDBal CD DepAmt DDA Dep ATMAmt CC Phone MMBal CCBal MM Sav;

/\* OUTPUT OUT = a PRED= phat;\*/

**RUN**;

**DATA** b;

SET a;

w = phat\*(**1**-phat);

**RUN**;

\*test for multicollinearity, MMBal and MM are correlated remove MMBal;

**PROC** **REG** DATA=b;

WEIGHT w;

MODEL INS = SavBal DDABal CDBal CD DepAmt DDA Dep ATMAmt CC Phone MMBal CCBal MM Sav / TOL VIF;

**RUN**;

\*second full model without MMBal(model 1);

ODS GRAPHICS ON;

**PROC** **LOGISTIC** DATA = Train PLOTS(ONLY)=ROC;

MODEL INS(EVENT = '1') = SavBal DDABal CDBal CD DepAmt DDA Dep ATMAmt CC Phone CCBal MM Sav;

**RUN**;

ODS GRAPHICS ON;

\*Model 1 on valid set;

ODS GRAPHICS ON;

**PROC** **LOGISTIC** DATA = Valid PLOTS(ONLY)=ROC;

MODEL INS(EVENT = '1') = SavBal DDABal CDBal CD DepAmt DDA Dep ATMAmt CC Phone CCBal MM Sav;

**RUN**;

ODS GRAPHICS ON;

\*use stepwise to select variables;

**PROC** **LOGISTIC** DATA = Train;

MODEL INS(EVENT = '1') = SavBal DDABal CDBal CD DepAmt DDA Dep ATMAmt CC Phone CCBal MM Sav /selection = stepwise;

**RUN**;

\*Final model by using stepwise selection(model 2);

ODS GRAPHICS ON;

**PROC** **LOGISTIC** DATA = Train PLOTS(ONLY)=ROC;

MODEL INS(EVENT = '1') = CD SavBal DDA DDABal MM Sav ATMAmt CC Phone CCBal Dep DepAmt;

**RUN**;

ODS GRAPHICS OFF;

\*Model 2 on valid set;

ODS GRAPHICS ON;

**PROC** **LOGISTIC** DATA = Valid PLOTS(ONLY)=ROC;

MODEL INS(EVENT = '1') = CD SavBal DDA DDABal MM Sav ATMAmt CC Phone CCBal Dep DepAmt;

**RUN**;

ODS GRAPHICS OFF;

\*remove variables with small coefficients (model3);

ODS GRAPHICS ON;

**PROC** **LOGISTIC** DATA = Train PLOTS(ONLY)=ROC;

MODEL INS(EVENT = '1') = CD DDA MM Sav CC Phone;

**RUN**;

ODS GRAPHICS OFF;

\*model3 on valid set;

ODS GRAPHICS ON;

**PROC** **LOGISTIC** DATA = Valid PLOTS(ONLY)=ROC;

MODEL INS(EVENT = '1') = CD DDA MM Sav CC Phone;

**RUN**;

ODS GRAPHICS OFF;

\*Model 2 on test set;

ODS GRAPHICS ON;

**PROC** **LOGISTIC** DATA = Test PLOTS(ONLY)=ROC;

MODEL INS(EVENT = '1') = CD SavBal DDA DDABal MM Sav ATMAmt CC Phone CCBal Dep DepAmt;

**RUN**;

ODS GRAPHICS OFF;

\*Use selected model to calculate predict values for all observations;

**PROC** **LOGISTIC** DATA = P4;

MODEL INS(EVENT = '1') = CD SavBal DDA DDABal MM Sav ATMAmt CC Phone CCBal Dep DepAmt;

OUTPUT OUT = a PRED= phat;

**RUN**;

\*select observations with ins = 0;

**DATA** P4ins;

SET a;

IF Ins = **0** AND NOT MISSING(phat);

**RUN**;

\*segment customers to 10 groups;

**PROC** **RANK** DATA = P4ins GROUPS = **10** OUT = P4.RankIns;

VAR phat;

RANKS PScore;

**RUN**;

**DATA** Rankdata;

SET P4.RankIns;

PScore + **1**;

**RUN**;

\*Find the group with highest score;

**DATA** P4.Ins10;

SET Rankdata;

IF PScore = **10**;

**RUN**;

**DATA** Ins10;

SET P4.Ins10;

**RUN**;

**PROC** **MEANS** DATA = Ins10 MEAN N;

VAR Age Income HMOwn;

**RUN**;

\*The group with lowest score;

**DATA** P4.Ins1;

SET Rankdata;

IF PScore = **1**;

**RUN**;

**DATA** Ins1;

SET P4.Ins1;

**RUN**;

**PROC** **MEANS** DATA = Ins1 MEAN N;

VAR Age Income HMOwn;

**RUN**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*SAS code for calculating Weight of Evidence and Information Value;

**DATA** Project4.P4WOEIV;

SET Project4.Project4(RENAME = (Branch = oldBranch));

Branch = INPUT(COMPRESS(oldBranch, "B"), **8.**);

IF Res = "R" THEN AreaNum = **1**;

ELSE IF Res = "S" THEN AreaNum = **2**;

ELSE IF Res = "U" THEN AreaNum = **3**;

DROP Res oldBranch;

**RUN**;

%let inset=P4woeiv; /\* data set name \*/

%let target=Ins; /\* target variable (y) \*/

%let libout=C:\Temp\SAS\WOEIV; /\* folder for export outputs \*/

%let varall= AcctAge DDA DDABal CashBk Checks DirDep NSF NSFAmt Phone Teller Sav SavBal ATM ATMAmt POS POSAmt CD CDBal IRA IRABal LOC LOCBal ILS ILSBal MM MMBal MMCred MTG MTGBal CC CCBal CCPurc SDB Income HMOwn LORes HMVal Age CRScore Moved InArea Branch Dep DepAmt Inv InvBal AreaNum; /\* list of variables to be assessed \*/

%let tiermax=10; /\* max number of bins to assign to variables \*/

%let ivthresh=0.1; /\* set to 0 if you want to see output graphs for all variables \*/

%let outgraph=iv\_woe\_graph; /\* pdf graph for top predictors \*/

%let ivout=iv\_ranked; /\* output file in txt for Information Value \*/

%let woeout=woe\_ranked; /\* output file in txt for Weight of Evidence \*/

%let libdata=Project4; /\* name of library where the data set is stored. \*/

%let outcome=pct\_cust\_insured; /\* name of target for summary tables \*/

%let outname=% Customers have insurance;/\*label of target for summary tables \*/

ODS OUTPUT nlevels=checkfreq;\*create a sas data set from the nlevels of proc freq;

**PROC** **FREQ** DATA = &libdata.**.**&inset nlevels;

TABLES &varall/NOPRINT;

**RUN**;

ODS OUTPUT CLOSE;

**DATA** varcnt; \*count the number of variables;

SET checkfreq;

varcnt + **1**;

**RUN**;

**PROC** **UNIVARIATE** DATA = varcnt;

VAR varcnt;

OUTPUT OUT = pctscore pctlpts = **0** **10** **20** **30** **40** **50** **60** **70** **80** **90** **100**

pctlpre = pct\_;

**RUN**;

**DATA** \_NULL\_;

SET pctscore;

CALL SYMPUT('start1', **1**);

CALL SYMPUT('end1', int(pct\_10) - **1**);

CALL SYMPUT('start2', int(pct\_10));

CALL SYMPUT('end2', int(pct\_20) - **1**);

CALL SYMPUT('start3', int(pct\_20));

CALL SYMPUT('end3', int(pct\_30) - **1**);

CALL SYMPUT('start4', int(pct\_30));

CALL SYMPUT('end4', int(pct\_40) - **1**);

CALL SYMPUT('start5', int(pct\_40));

CALL SYMPUT('end5', int(pct\_50) - **1**);

CALL SYMPUT('start6', int(pct\_50));

CALL SYMPUT('end6', int(pct\_60) - **1**);

CALL SYMPUT('start7', int(pct\_60));

CALL SYMPUT('end7', int(pct\_70) - **1**);

CALL SYMPUT('start8', int(pct\_70));

CALL SYMPUT('end8', int(pct\_80) - **1**);

CALL SYMPUT('start9', int(pct\_80));

CALL SYMPUT('end9', int(pct\_90) - **1**);

CALL SYMPUT('start10', int(pct\_90));

CALL SYMPUT('end10', pct\_100);

**RUN**;

\*\* select variables with less than needed number of tiers such as 10 in this example;

**PROC** **SQL** NOPRINT;

SELECT tablevar INTO : varmore SEPARATED BY ' '

FROM varcnt

WHERE nlevels > &tiermax;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE vcnt AS

SELECT COUNT(\*) AS vcnt

FROM varcnt

WHERE nlevels > &tiermax;

**QUIT**;

**DATA** \_NULL\_;

SET vcnt;

CALL SYMPUT('vmcnt', vcnt);

**RUN**;

**PROC** **SQL** NOPRINT;

SELECT TableVar

INTO :v1 - : v%***LEFT***(&vmcnt) /\*%let statement trim the leading and trailing blanks.select into does not\*/

FROM varcnt

WHERE nlevels > &tiermax;

**QUIT**;

**PROC** **SQL** NOPRINT;

SELECT TableVar

INTO : x1 - :x%***LEFT***(&end10)

FROM varcnt;

**QUIT**;

**PROC** **SQL** NOPRINT;

SELECT MAX(varcnt), COMPRESS('&x'||PUT(varcnt, **10.**))

INTO : varcount, : tempvar SEPARATED BY ' '

FROM varcnt

ORDER BY varcnt;

**QUIT**;

**PROC** **SQL** NOPRINT;

SELECT COUNT(\*)

INTO : obscnt /\*hold total number of observations in origianl data set\*/

FROM &libdata.**.**&inset;

**QUIT**;

\*create 10 categories for each variable and calculate statistics;

**%macro** ***stkorig***;

%DO i = **1** %TO &vmcnt;

DATA v&i;

LENGTH TableVar $32.;

SET &libdata.**.**&inset(KEEP = &&v&i RENAME = (&&v&i=origvalue));

TableVar = "&&v&i";

FORMAT TableVar $32.;

ATTRIB \_ALL\_ LABEL = ' ';

RUN;

PROC RANK DATA = v&i GROUPS = &tiermax OUT = v&i;

BY TableVar;

VAR origvalue;

RANKS rankvmore;

RUN;

PROC MEANS DATA = v&i MEDIAN MEAN MIN MAX NWAY NOPRINT;

CLASS TableVar rankvmore;

VAR origvalue;

OUTPUT OUT = vmoreranked&i(DROP = \_FREQ\_ \_TYPE\_)

MEDIAN = Med\_Origv

MEAN = Mean\_Origv

MIN = Min\_Origv

MAX = Max\_Origv;

RUN;

%END;

**%mend** stkorig;

%***stkorig***

**DATA** stackorig;

SET Vmoreranked1-Vmoreranked%***LEFT***(&vmcnt);

**RUN**;

\*\* make a permanent dataset;

**DATA** &libdata.**.s**tackorig;

SET stackorig;

**RUN**;

\*\* only rank these variables with more than 10 values;

\*\* the following dataset is for later aggregation in a sas macro;

**PROC** **RANK** DATA = &libdata.**.**&inset GROUPS = &tiermax OUT = try\_model(KEEP = &tempvar &target);

VAR &varmore;

RANKS &varmore;

**RUN**;

/\*\*\* generate Information Value and Weight of Evidence;\*/

**%macro** ***outshell***;

%DO i = **1** %TO &varcount;

DATA Try\_Model&i(KEEP = &&x&i &Target);

SET Try\_model;

IF &&x&i = **.** THEN &&x&i = -**1000000000**;

RUN;

PROC SQL NOPRINT;

SELECT SUM(CASE WHEN &target = **1** THEN **1** ELSE **0** END),

SUM(CASE WHEN &target = **0** THEN **1** ELSE **0** END),

COUNT(\*) INTO : tot\_bad, : tot\_good, : tot\_both

FROM Try\_Model&i

WHERE &&x&i NE -**1000000000**;

QUIT;

PROC SQL NOPRINT;

SELECT COUNT(\*) INTO : nonmiss

FROM Try\_Model&i

WHERE &&x&i NE -**1000000000**;

QUIT;

PROC SQL;

CREATE TABLE woe&i AS

SELECT "&&x&i" AS TableVar,

&&x&i AS tier,

COUNT(\*) AS cnt,

COUNT(\*) / &tot\_both AS cnt\_pct,

SUM(CASE WHEN &target = **0** THEN **1** ELSE **0** END) AS sum\_good,

SUM(CASE WHEN &target = **0** THEN **1** ELSE **0** END)/&tot\_good AS dist\_good,

SUM(CASE WHEN &target = **1** THEN **1** ELSE **0** END) AS sum\_bad,

SUM(CASE WHEN &target = **1** THEN **1** ELSE **0** END)/&tot\_bad as dist\_bad,

LOG(SUM(CASE WHEN &target = **0** THEN **1** ELSE **0** END)/(SUM(CASE WHEN

&target = **1** THEN **1** ELSE **0** END)/&tot\_bad))\***100** AS woe,

((SUM(CASE WHEN &target = **0** THEN **1** ELSE **0** END)/&tot\_good)-(SUM(CASE WHEN

&target= **1** THEN **1** ELSE **0** END)/&tot\_bad))\*LOG((SUM(CASE WHEN &target = **0** THEN **1**

ELSE **0** END)/&tot\_good)/(SUM(CASE WHEN &target = **1** THEN **1** ELSE **0** END)/&tot\_bad))

AS pre\_iv,

SUM(CASE WHEN &target = **1** THEN **1** ELSE **0** END)/count(\*) AS &outcome

FROM Try\_Model&i

GROUP BY &&x&i;

QUIT;

PROC SQL;

CREATE TABLE iv&i AS

SELECT "&&x&i" AS TableVar,

SUM(pre\_iv) AS iv,

(**1** - %SYSEVALF(&nonmiss/&obscnt)) AS pct\_missing

FROM woe&i;

QUIT;

%END;

**%mend** outshell;

%***outshell***;

**%macro** ***stackset***;

%do j=**1** %to **10**;

data tempiv&j;

length tablevar $32.;

set iv&&start&j-iv&&end&j;

format tablevar $32.;

run;

data tempwoe&j;

length tablevar $32.;

set woe&&start&j-woe&&end&j;

format tablevar $32.;

run;

%end;

**%mend**;

%***stackset***;

**%macro** ***stackset***;

%DO j = **1** %TO **10**;

DATA tempiv&j;

LENGTH TableVar $32.;

SET iv%***left***(&&start&j) - iv%***left***(&&end&j);

FORMAT Tablevar $32.;

RUN;

DATA tempwoe&j;

LENGTH TableVar $32.;

SET woe%***left***(&&start&j) - woe%***left***(&&end&j);

FORMAT TableVar $32.;

RUN;

%END;

**%mend**;

%***stackset***;

**DATA** &libdata.**.i**vall;

SET tempiv1 - tempiv10;

**RUN**;

**DATA** &libdata.**.w**oeall;

SET tempwoe1-tempwoe10;

**RUN**;

**PROC** **SORT** DATA = &libdata.**.i**vall;

BY DESCENDING iv;

**RUN**;

**DATA** &libdata.**.i**vall;

SET &libdata.**.i**vall;

ivrank + **1**;

**RUN**;

**PROC** **SORT** DATA = &libdata.**.i**vall NODUPKEY OUT = ivtemp(KEEP = iv);

BY DESCENDING iv;

**RUN**;

**DATA** ivtemp;

SET ivtemp;

ivtier + **1**;

**RUN**;

**PROC** **SORT** DATA = ivtemp;

BY iv;

**RUN**;

**PROC** **SORT** DATA = &libdata.**.i**vall;

BY iv;

**RUN**;

**DATA** &ivout;

MERGE &libdata.**.i**vall ivtemp;

BY iv;

**RUN**;

**PROC** **SORT** DATA = &ivout;

BY TableVar;

**RUN**;

**PROC** **SORT** DATA = &libdata.**.w**oeall;

BY TableVar;

**RUN**;

**DATA** &libdata.**.i**v\_woe\_all;

MERGE &ivout &libdata.**.w**oeall;

BY TableVar;

**RUN**;

**PROC** **SORT** DATA = &libdata.**.i**v\_woe\_all;

BY TableVar;

**RUN**;

**PROC** **SORT** DATA = &libdata.**.s**tackorig;

BY TableVar rankvmore;

**RUN**;

**DATA** &libdata.**.i**v\_woe\_all2;

MERGE &libdata.**.i**v\_woe\_all(in = t)

&libdata.**.s**tackorig(in = s RENAME = (rankvmore = tier));

BY TableVar Tier;

IF t;

IF s THEN tier = med\_origv;

**RUN**;

**PROC** **SORT** DATA = &libdata.**.i**v\_woe\_all2;

BY ivrank tier;

**RUN**;

%LET retvar=tablevar iv ivrank ivtier tier cnt cnt\_pct dist\_good dist\_bad woe &outcome pct\_missing;

**DATA** &libdata.**.**&woeout(KEEP = &retvar);

RETAIN &retvar;

SET &libdata.**.i**v\_woe\_all2;

LABEL TableVar = "Variable";

LABEL iv = "Information Value";

LABEL ivrank = "IV Rank";

LABEL tier = "Tier/Bin";

LABEL cnt = "# Customers";

LABEL cnt\_pct = "% Customers";

LABEL dist\_good = "% Good";

LABEL dist\_bad = "% Bad";

LABEL woe = "Weight of Evidence";

LABEL &outcome = "&outname";

LABEL pct\_missing = "% Missing Values";

**RUN**;

\*\* examine KS;

**PROC** **NPAR1WAY** DATA = &libdata.**.**&inset EDF NOPRINT;/\* specify the input dataset \*/

VAR &varall;/\* type your list of predictors(x) here \*/

CLASS &target;/\* target variable such as BAD \*/

OUTPUT OUT = ks101(KEEP = \_var\_ \_D\_ RENAME = (\_VAR\_ = TableVar \_D\_ = ks));

**RUN**;

**PROC** **SORT** DATA = ks101;

BY TableVar;

**RUN**;

**PROC** **SORT** DATA = &ivout;

BY TableVar;

**RUN**;

**DATA** &libdata.**.**&ivout;

RETAIN TableVar iv ivrank ivtier ks pct\_missing;

MERGE ks101 &ivout;

BY TableVar;

KEEP TableVar iv ivrank ivtier ks pct\_missing;

**RUN**;

**PROC** **CONTENTS** DATA = &libdata.**.**&woeout VARNUM;

**RUN**;

**PROC** **CONTENTS** DATA = &libdata.**.**&ivout VARNUM;

**RUN**;

**PROC** **SORT** DATA = &libdata.**.**&woeout OUT = &woeout(DROP = ivrank RENAME = (ivtier = iv\_rank));

BY ivtier TableVar;

**RUN**;

**PROC** **SORT** DATA = &libdata.**.**&ivout OUT = &ivout(DROP = ivrank RENAME = (ivtier = iv\_rank));

BY ivtier;

**RUN**;

**%macro** to\_excel(data\_sum);

PROC EXPORT DATA = &data\_sum

OUTFILE = "&libout/&data\_sum..xls"

DBMS = tab

REPLACE;

RUN;

**%mend**;

%***to\_excel***(&ivout);

%***to\_excel***(&woeout);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*SAS macros in the inldued sas file;

\*---------------------------------------------------------------\*

|Program Name: Head.sas |

|Purpose: Macro which prints out the first 8 observations or |

| the first n observations. |

|Argument: %Head(Dat, N = ) |

|Examples: %Head(Test) |

| %Head(Test, N = 10) |

\*---------------------------------------------------------------\*;

**%macro** Head(Dat, N = **8**);

PROC PRINT DATA = &Dat(OBS = &N) NOOBS;

TITLE "The First &N Observations of Data Set &Dat";

RUN;

**%mend** Head;

\*---------------------------------------------------------------\*

|Program Name: Tail.sas |

|Purpose: Macro which prints out the Last 8 observations or |

| the Last n observations. |

|Argument: %Tail(Dat, N = ) |

|Examples: %Tail(Test) |

| %Tail(Test, N = 10) |

\*---------------------------------------------------------------\*;

**%macro** Tail(Dat, N = **8**);

Data New;

SET &Dat NOBS = TNumbs;

TotalNum = TNumbs;

RUN;

PROC SQL NOPRINT;

SELECT TotalNum INTO : NRows

FROM New;

QUIT;

PROC PRINT DATA = New (FIRSTOBS = %eval(&NRows - &N + **1**) OBS = &NRows) NOOBS;

TITLE " The Last &N Observations of Data set &Dat";

RUN;

**%mend** Tail;

\*---------------------------------------------------------------\*

|Program Name: HighLowN |

|Purpose: To list the "n" highest and lowest values |

|Arguments: Dat - Data set name (one- or two-level |

| Var - Variable to list |

| Idvar - ID variable |

| n - Number of variables to list |

|Example: %HighLowN(Dat = P4, |

| Var = DDABal, |

| Idvar = AcctAge |

| n = 7) |

\*---------------------------------------------------------------\*;

**%macro** HighLowN(Dat =, Var =, Idvar =, n =);

PROC SORT DATA = &Dat(KEEP=&Idvar &Var

WHERE=(&Var is not missing)) out=tmp;

by &Var;

run;

DATA \_NULL\_;

SET tmp NOBS = Num\_Obs;

CALL SYMPUT('Num', Num\_Obs);

STOP;

RUN;

%LET High = %EVAL(&Num - &n + 1);

TITLE "&n Highest and Lowest Values for &Var";

DATA \_NULL\_;

SET tmp(obs = &n) /\* lowest values \*/

tmp(firstobs = &high) /\* highest values \*/;

FILE PRINT;

IF \_N\_ LE &n THEN DO;

IF \_N\_ = **1** THEN PUT / "&n Lowest Values";

PUT "&Idvar = " &Idvar @**15** "Value = " &Var;

END;

ELSE IF \_N\_ GE %EVAL(&n + **1**) THEN DO;

IF \_N\_ = %EVAL(&n + **1**) THEN PUT / "&n Highest Values";

PUT "&Idvar = " &Idvar @**15** "Value = " &Var;

END;

RUN;

PROC DATASETS LIBRARY = work NOLIST;

DELETE tmp;

RUN;

QUIT;

**%mend** HighLowN;

\*---------------------------------------------------------------\*

|Macro Name: HighLowNPercent |

|Purpose: To list the upper and lower n percent of values |

|Arguments: Dat - Data set name |

| Var - Variable to test |

| Percent - Upper and lower n percent |

| Idvar - ID variable |

|Example: %HighLow\_nPercent(Dat=P4, |

| Var=DDABal, |

| Percent=2, |

| Idvar=AcctAge) |

----------------------------------------------------------------\*;

**%macro** HighLow\_nPercent(Dat=, Var=, Percent=, Idvar=);

%LET Low = %EVAL(&Percent - 1);

%LET High = %EVAL(100 - &Percent);

PROC FORMAT;

VALUE rnk **0** - &Low = 'Low'

&High - **99** = 'High';

RUN;

PROC RANK DATA = &Dat(KEEP=&Var &Idvar)

OUT = New(WHERE = (&Var IS NOT MISSING))

GROUPS = **100**;

VAR &Var;

RANKS Range;

RUN;

\*\*\*Sort and keep top and bottom n%;

PROC SORT DATA=New(WHERE=(Range le &Low or Range GE &High));

BY &Var;

RUN;

\*\*\*Produce the report;

PROC PRINT DATA = New;

TITLE "Upper and Lower &Percent.% Values for %UPCASE(&Var)";

ID &Idvar;

VAR Range &Var;

FORMAT Range rnk.;

RUN;

PROC DATASETS LIBRARY = work NOLIST;

DELETE New;

RUN;

QUIT;

**%mend** HighLow\_nPercent;

\*---------------------------------------------------------------\*

| Macro Name: CellMeans |

| Purpose: Computes means (and n's) for a list of variables, |

| given a list of CLASS variables |

| Arguments: IN\_Dat = Data set name |

| OUT\_Dat = Output data set name |

| CLASS = List of CLASS variables |

| VARLIST = List of variables on which to compute |

| means and counts |

| Example: %CELLMEANS(TESTIT,TEMP,GENDER RACE,X Y) |

\*---------------------------------------------------------------\*;

**%macro** CellMeans(In\_Dat, Out\_Dat, Class, Varlist);

PROC MEANS DATA = &In\_Dat NOPRINT NWAY;

CLASS &Class;

VAR &Varlist;

OUTPUT OUT = &Out\_Dat(DROP = \_FREQ\_ \_TYPE\_)

MEAN =

N = /AUTONAME;

RUN;

**%mend** Cellmeans;