libname class "/gpfs/user\_home/os\_home\_dirs/lcohen7/STAT4330";

\*loading data;

data testcpr; \*339 variables;

set cd4330.cpr;

data testperf; \*18 variables;

set cd4330.perf;

\*sorting data;

proc sort data=cd4330.perf out=test;

by Matchkey delqid;

proc sort data=cd4330.cpr out= testcpr;

by Matchkey;

\*merging the data and checking content;

data merged; \*356 variables;

merge testcpr testperf;

by Matchkey;

if last.Matchkey; \* puts all matchkeys in order and sort by delqid, use the the last number;

if delqid ne . ; \* if delqid is missing get rid of it;

if age ne .; \* every person has age so taking out any obs they don't have age;

proc print data=merged (obs=20);

proc contents data= merged;

\* dropping variables in perf dataset;

data class.merged;

set merged;

data class.merged; \*341 variables;

set class.merged;

drop beacon numpur payamt puramt chgoffid numcash cashamt numpay cashbal paydue tba latefee

purbal ovlimfee cycledt opendate;

\* run macro \*;

\* 165 variables\* ;

%IMPV3 (DSN=class.merged, VARS=\_ALL\_, EXCLUDE= matchkey crelim delqid goodbad age,

PCTREM=0.40, MSTD=4);

proc means data=class.mergedout max nmiss maxdec=1;

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

%Let pc = 0.1; \*The PCTREM value you want to start on, it will currently run 6 additional times adding 0.10 each time;

%Let ds = class.merged; \*The file you want to run the imputation macro on;

%Let ln = 'CLASS'; \*Must be in all caps, the name of the library your file is in;

%Let mn = 'MERGEDOUT'; \*Must be in all caps, the name of the output file the imputation macro creates;

\*This line runs the macro and imputes \_ALL\_ variables except age, matchkey, crelim, delqid, and goodbad;

%IMPV3 (DSN=&ds, VARS=\_ALL\_, EXCLUDE= age matchkey crelim delqid goodbad, PCTREM= &pc, MSTD=4);

proc sql;

create table class.vars\_rem as

select libname, memname, &pc as PCTREM, nvar as totalcolumns from sashelp.vtable

where libname in (&ln) and memname in (&mn);

quit;

%Let pc = &pc + 0.1;

%IMPV3 (DSN=&ds, VARS=\_ALL\_, EXCLUDE= age matchkey crelim delqid goodbad, PCTREM= &pc, MSTD=4);

proc sql;

insert into class.vars\_rem

select libname, memname, &pc as PCTREM, nvar as totalcolumns from sashelp.vtable

where libname in (&ln) and memname in (&mn);

quit;

%Let pc = &pc + 0.1;

%IMPV3 (DSN=&ds, VARS=\_ALL\_, EXCLUDE= age matchkey crelim delqid goodbad, PCTREM= &pc, MSTD=4);

proc sql;

insert into class.vars\_rem

select libname, memname, &pc as PCTREM, nvar as totalcolumns from sashelp.vtable

where libname in (&ln) and memname in (&mn);

quit;

%Let pc = &pc + 0.1;

%IMPV3 (DSN=&ds, VARS=\_ALL\_, EXCLUDE= age matchkey crelim delqid goodbad, PCTREM= &pc, MSTD=4);

proc sql;

insert into class.vars\_rem

select libname, memname, &pc as PCTREM, nvar as totalcolumns from sashelp.vtable

where libname in (&ln) and memname in (&mn);

quit;

%Let pc = &pc + 0.1;

%IMPV3 (DSN=&ds, VARS=\_ALL\_, EXCLUDE= age matchkey crelim delqid goodbad, PCTREM= &pc, MSTD=4);

proc sql;

insert into class.vars\_rem

select libname, memname, &pc as PCTREM, nvar as totalcolumns from sashelp.vtable

where libname in (&ln) and memname in (&mn);

quit;

%Let pc = &pc + 0.1;

%IMPV3 (DSN=&ds, VARS=\_ALL\_, EXCLUDE= age matchkey crelim delqid goodbad, PCTREM= &pc, MSTD=4);

proc sql;

insert into class.vars\_rem

select libname, memname, &pc as PCTREM, nvar as totalcolumns from sashelp.vtable

where libname in (&ln) and memname in (&mn);

quit;

%Let pc = &pc + 0.1;

%IMPV3 (DSN=&ds, VARS=\_ALL\_, EXCLUDE= age matchkey crelim delqid goodbad, PCTREM= &pc, MSTD=4);

proc sql;

insert into class.vars\_rem

select libname, memname, &pc as PCTREM, nvar as totalcolumns from sashelp.vtable

where libname in (&ln) and memname in (&mn);

quit;

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

\* creating good bad;

data merged1;

set class.merged;

if delqid < 3 then goodbad=0;

else goodbad =1;

\* freq table of the people who were accepted and not accepted;

proc freq data=merged1;

tables goodbad;

data class.merged;

set class.merged;

if delqid < 3 then goodbad=0;

else goodbad =1;

\* creating good bad;

data class.mergedout;

set class.merged;

if delqid < 3 then goodbad=0;

else goodbad =1;

\* freq table of the people who were accepted and not accepted;

proc freq data=class.mergedout;

tables goodbad;

\*///Predictor - PRDEROG///;

%let d1 = class.merged;

proc univariate data=&d1 noprint;

Var PRDEROG;

histogram;

\* to get the number of coded values;

proc freq data=cd4330.cpr;

tables PRDEROG;

where PRDEROG ge 93;

proc means data = &d1 mean median;

var PRDEROG;

where PRDEROG<50;

proc means data=class.merged min max median mean;

var prderog;

data &d1;

set &d1;

if PRDEROG ge 93 then PRDEROG = 1;

else PRDEROG = PRDEROG;

proc univariate data=&d1 noprint;

Var PRDEROG;

histogram;

proc means data=&d1 min max median mean;

var prderog;

\* histograms;

%IMPV3 (DSN=class.merged, VARS=\_ALL\_, EXCLUDE= matchkey crelim delqid goodbad age,

PCTREM=0.40, MSTD=4);

proc univariate data=class.mergedout noprint;

var DCAGE;

histogram;

%IMPV3 (DSN=class.merged, VARS=\_ALL\_, EXCLUDE= matchkey crelim delqid goodbad age,

PCTREM=0.40, MSTD=5);

proc univariate data=class.mergedout noprint;

var DCAGE;

histogram;

%IMPV3 (DSN=class.merged, VARS=\_ALL\_, EXCLUDE= matchkey crelim delqid goodbad age,

PCTREM=0.40, MSTD=6);

proc univariate data=class.mergedout noprint;

var DCAGE;

histogram;

\* histograms;

%IMPV3 (DSN=class.merged, VARS=\_ALL\_, EXCLUDE= matchkey crelim delqid goodbad age,

PCTREM=0.40, MSTD=4);

proc univariate data=class.mergedout noprint;

var CRATE79;

histogram;

%IMPV3 (DSN=class.merged, VARS=\_ALL\_, EXCLUDE= matchkey crelim delqid goodbad age,

PCTREM=0.40, MSTD=5);

proc univariate data=class.mergedout noprint;

var CRATE79;

histogram;

%IMPV3 (DSN=class.merged, VARS=\_ALL\_, EXCLUDE= matchkey crelim delqid goodbad age,

PCTREM=0.40, MSTD=6);

proc univariate data=class.mergedout noprint;

var CRATE79;

histogram;

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

\*variable clusters;

%let inputs = COLLS BKP BKPOP FININQS PRMINQS BNKINQS TOPEN24 LAAGE WCRATE AVGMOS TOPENB50 BRWCRATE

BRBAL50 DCWCRATE FFWCRATE ORRATE3 TADB BRADB TPCTSAT TPOPEN CPAF29 TR29P24 PRMINQ2 BNKINQ2 TCR1BAL MOSOPEN

TROPENEX BROPENEX TRR23 BRR23 DCR23 TRCR49 FFCR49 OBRPTAT OT24PTOT OT12PTOT OT3PTOT BRNEW INQ12 ROPEN

RADB6 TADB25 BRADBM TBAL

;

\*insert macro variables, THE CODE ABOVE IS RAN AFTER MACRO IS FIXED AND YOU FIND A GOOD MSTD;

proc reg data=class.merged;

model goodbad = &inputs/ vif;

\*dont run!! add variables;

\*get rid of 1 at a time!! if 2 of similar VIF are there, get rid of the higher one first

anything greater than 10 = delete;

\*after running vif, i have this many variables, and now ill cluster them;

ods listing close;

ods output clusterquality=summary rsquare=clusters;

proc varclus data=class.merged outtree = tree maxclusters=44;

var &inputs;

ods listing;

\*proportion: x amount of all variance can be explained by one variable, highest number in "own cluster"

(related to own cluster) and lowest next closest, ultimately the LOWEST VIF would be the one to speak for the

group;

\*correlation table of lowest 1-R^2 value;

proc corr data = class.merged;

var OT12PTOT FFCR49 BRADB TRCR49 BRWCRATE DCWCRATE BRNEW TROPENEX FININQS TADB25 BRBAL50 MOSOPEN TPCTSAT

BKPOP PRMINQ2 BNKINQ2 BNKINQS TBAL LAAGE BKP ORRATE3 ROPEN PRMINQS TOPENB50 INQ12 TADB; \*add all variables;

\*look for correlation VARIABLES, top number;

\*shows cluster graph;

data \_null\_;

set summary;

call symput ('nvar',compress(NumberofClusters));

proc print data = clusters noobs;

where NumberofClusters = &nvar;

var cluster variable rsquareratio;

proc contents data=summary varnum;

symbol v=square color=black i=join;

proc gplot data=summary;

plot propvarexplained\*numberofclusters;

quit;

\*Let's say you select 5 clusters;

data somename;

set clusters;

if numberofclusters = 26;

\*Correlation table output;

proc corr data=class.merged;

var OT12PTOT FFCR49 BRADB TRCR49 BRWCRATE DCWCRATE BRNEW TROPENEX FININQS TADB25 BRBAL50 MOSOPEN TPCTSAT

BKPOP PRMINQ2 BNKINQ2 BNKINQS TBAL LAAGE BKP ORRATE3 ROPEN PRMINQS TOPENB50 INQ12 TADB;

data newdata;

set class.mergedout;

keep age matchkey delqid goodbad crelim

OT12PTOT FFCR49 BRADB TRCR49 BRWCRATE DCWCRATE BRNEW TROPENEX FININQS TADB25 BRBAL50 MOSOPEN TPCTSAT

BKPOP PRMINQ2 BNKINQ2 BNKINQS TBAL LAAGE BKP ORRATE3 ROPEN PRMINQS TOPENB50 INQ12 TADB;

\*\* Discredization 1;

\*\*\*\*\*\*\*\*\*\*\*OT12PTOT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var OT12PTOT;

histogram;

Data disc1;

set newdata;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var OT12PTOT;

class ORDOT12PTOT;

Proc means data=disc1;

Var goodbad;

class ORDOT12PTOT;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDOT12PTOT;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDOT12PTOT;

Var OT12PTOT GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDOT12PTOT;

Proc sort data=&d1;

by ORDOT12PTOT;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDOT12PTOT;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDOT12PTOT;

histogram;

\*\*\*\*\*\*\*\*\*\*\*BRADB\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var BRADB;

histogram;

Data disc1;

set newdata;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var BRADB;

class ORDBRADB;

Proc means data=disc1;

Var goodbad;

class ORDBRADB;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDBRADB;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDBRADB;

Var BRADB GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDBRADB;

Proc sort data=&d1;

by ORDBRADB;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDBRADB;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDBRADB;

histogram;

\*\*\*\*\*\*\*\*\*\*\*TRCR49\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var TRCR49;

histogram;

Data disc1;

set newdata;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var TRCR49;

class ORDTRCR49;

Proc means data=disc1;

Var goodbad;

class ORDTRCR49;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDTRCR49;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDTRCR49;

Var TRCR49 GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDTRCR49;

Proc sort data=&d1;

by ORDTRCR49;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDTRCR49;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDTRCR49;

histogram;

\*\*\*\*\*\*\*\*\*\*\*BRWCRATE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var BRWCRATE;

histogram;

Data disc1;

set newdata;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var BRWCRATE;

class ORDBRWCRATE;

Proc means data=disc1;

Var goodbad;

class ORDBRWCRATE;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDBRWCRATE;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDBRWCRATE;

Var BRWCRATE GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDBRWCRATE;

Proc sort data=&d1;

by ORDBRWCRATE;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDBRWCRATE;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDBRWCRATE;

histogram;

\*\*\*\*\*\*\*\*\*\*\*DCWCRATE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var DCWCRATE;

histogram;

Data disc1;

set newdata;

if DCWCRATE = 1 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var DCWCRATE;

class ORDDCWCRATE;

Proc means data=disc1;

Var goodbad;

class ORDDCWCRATE;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDDCWCRATE;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDDCWCRATE;

Var DCWCRATE GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDDCWCRATE;

Proc sort data=&d1;

by ORDDCWCRATE;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDDCWCRATE;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDDCWCRATE;

histogram;

\*\*\*\*\*\*\*\*\*\*\*BRNEW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var BRNEW;

histogram;

Data disc1;

set newdata;

if BRNEW < 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var BRNEW;

class ORDBRNEW;

Proc means data=disc1;

Var goodbad;

class ORDBRNEW;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDBRNEW;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDBRNEW;

Var BRNEW GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDBRNEW;

Proc sort data=&d1;

by ORDBRNEW;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDBRNEW;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDBRNEW;

histogram;

\*\*\*\*\*\*\*\*\*\*\*TROPENEX\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var TROPENEX;

histogram;

Data disc1;

set newdata;

if TROPENEX < 10 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var TROPENEX;

class ORDTROPENEX;

Proc means data=disc1;

Var goodbad;

class ORDTROPENEX;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDTROPENEX;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDTROPENEX;

Var TROPENEX GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDTROPENEX;

Proc sort data=&d1;

by ORDTROPENEX;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDTROPENEX;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDTROPENEX;

histogram;

\*\*\*\*\*\*\*\*\*\*\*FININQS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var FININQS;

histogram;

Data disc1;

set newdata;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var FININQS;

class ORDFININQS;

Proc means data=disc1;

Var goodbad;

class ORDFININQS;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDFININQS;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDFININQS;

Var FININQS GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDFININQS;

Proc sort data=&d1;

by ORDFININQS;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDFININQS;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDFININQS;

histogram;

\*\*\*\*\*\*\*\*\*\*\*TADB25\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var TADB25;

histogram;

Data disc1;

set newdata;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var TADB25;

class ORDTADB25;

Proc means data=disc1;

Var goodbad;

class ORDTADB25;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDTADB25;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDTADB25;

Var TADB25 GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDTADB25;

Proc sort data=&d1;

by ORDTADB25;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDTADB25;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDTADB25;

histogram;

\*\*\*\*\*\*\*\*\*\*\*TADB\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var TADB;

histogram;

Data disc1;

set newdata;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var TADB;

class ORDTADB;

Proc means data=disc1;

Var goodbad;

class ORDTADB;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDTADB;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDTADB;

Var TADB GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDTADB;

Proc sort data=&d1;

by ORDTADB;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDTADB;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDTADB;

histogram;

\*\*\*\*\*\*\*\*\*\*\*BRBAL50\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var BRBAL50;

histogram;

Data disc1;

set newdata;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var BRBAL50;

class ORDBRBAL50;

Proc means data=disc1;

Var goodbad;

class ORDBRBAL50;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDBRBAL50;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDBRBAL50;

Var BRBAL50 GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDBRBAL50;

Proc sort data=&d1;

by ORDBRBAL50;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDBRBAL50;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDBRBAL50;

histogram;

\*\*\*\*\*\*\*\*\*\*\*MOSOPEN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var MOSOPEN;

histogram;

Data disc1;

set newdata;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var MOSOPEN;

class ORDMOSOPEN;

Proc means data=disc1;

Var goodbad;

class ORDMOSOPEN;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDMOSOPEN;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDMOSOPEN;

Var MOSOPEN GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDMOSOPEN;

Proc sort data=&d1;

by ORDMOSOPEN;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDMOSOPEN;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDMOSOPEN;

histogram;

\*\*\*\*\*\*\*\*\*\*\*TPCTSAT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var TPCTSAT;

histogram;

Data disc1;

set newdata;

if TPCTSAT < 0.2 then ORDTPCTSAT= 1;

else if TPCTSAT < 0.4 then ORDTPCTSAT= 2;

else if TPCTSAT < 0.6 then ORDTPCTSAT = 3;

else if TPCTSAT < 0.8 then ORDTPCTSAT = 4;

else ORDTPCTSAT = 5;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var TPCTSAT;

class ORDTPCTSAT;

Proc means data=disc1;

Var goodbad;

class ORDTPCTSAT;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDTPCTSAT;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDTPCTSAT;

Var TPCTSAT GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

oddsORDTPCTSAT = (avg\_dep/(1-avg\_dep));

loddsORDTPCTSAT = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDTPCTSAT;

Proc sort data=&d1;

by ORDTPCTSAT;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDTPCTSAT;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDTPCTSAT;

histogram;

\*\*\*\*\*\*\*\*\*\*\*PRMINQ2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var PRMINQ2;

histogram;

Data disc1;

set newdata;

if PRMINQ2 < 5 then ORDPRMINQ2= 1;

else if PRMINQ2 < 10 then ORDPRMINQ2 = 2;

else if PRMINQ2 < 15 then ORDPRMINQ2 = 3;

else if PRMINQ2 < 20 then ORDPRMINQ2 = 4;

else ORDPRMINQ2 = 5;

if TPCTSAT < 0.2 then ORDTPCTSAT= 1;

else if TPCTSAT < 0.4 then ORDTPCTSAT= 2;

else if TPCTSAT < 0.6 then ORDTPCTSAT = 3;

else if TPCTSAT < 0.8 then ORDTPCTSAT = 4;

else ORDTPCTSAT = 5;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var PRMINQ2;

class ORDPRMINQ2;

Proc means data=disc1;

Var goodbad;

class ORDPRMINQ2;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDPRMINQ2;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDPRMINQ2;

Var PRMINQ2 GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

oddsORDTPCTSAT = (avg\_dep/(1-avg\_dep));

loddsORDTPCTSAT = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQ2 = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDPRMINQ2;

Proc sort data=&d1;

by ORDPRMINQ2;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDPRMINQ2;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDPRMINQ2;

histogram;

\*\*\*\*\*\*\*\*\*\*\*BNKINQ2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var BNKINQ2;

histogram;

Data disc1;

set newdata;

if BNKINQ2 < 2 then ORDBNKINQ2= 1;

else if BNKINQ2 < 4 then ORDBNKINQ2 = 2;

else if BNKINQ2 < 6 then ORDBNKINQ2 = 3;

else if BNKINQ2 < 8 then ORDBNKINQ2 = 4;

else if BNKINQ2 < 10 then ORDBNKINQ2 = 5;

else ORDBNKINQ2 = 6;

if PRMINQ2 < 5 then ORDPRMINQ2= 1;

else if PRMINQ2 < 10 then ORDPRMINQ2 = 2;

else if PRMINQ2 < 15 then ORDPRMINQ2 = 3;

else if PRMINQ2 < 20 then ORDPRMINQ2 = 4;

else ORDPRMINQ2 = 5;

if TPCTSAT < 0.2 then ORDTPCTSAT= 1;

else if TPCTSAT < 0.4 then ORDTPCTSAT= 2;

else if TPCTSAT < 0.6 then ORDTPCTSAT = 3;

else if TPCTSAT < 0.8 then ORDTPCTSAT = 4;

else ORDTPCTSAT = 5;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var BNKINQ2;

class ORDBNKINQ2;

Proc means data=disc1;

Var goodbad;

class ORDBNKINQ2;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDBNKINQ2;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDBNKINQ2;

Var BNKINQ2 GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

oddsORDTPCTSAT = (avg\_dep/(1-avg\_dep));

loddsORDTPCTSAT = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQ2 = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDBNKINQ2;

Proc sort data=&d1;

by ORDBNKINQ2;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDBNKINQ2;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDBNKINQ2;

histogram;

\*\*\*\*\*\*\*\*\*\*\*BNKINQS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var BNKINQS;

histogram;

Data disc1;

set newdata;

if BNKINQS < 2.5 then ORDBNKINQS= 1;

else if BNKINQS < 5.0 then ORDBNKINQS = 2;

else if BNKINQS < 7.5 then ORDBNKINQS = 3;

else if BNKINQS < 10 then ORDBNKINQS = 4;

else if BNKINQS < 12.5 then ORDBNKINQS = 5;

else ORDBNKINQS = 6;

if BNKINQ2 < 2 then ORDBNKINQ2= 1;

else if BNKINQ2 < 4 then ORDBNKINQ2 = 2;

else if BNKINQ2 < 6 then ORDBNKINQ2 = 3;

else if BNKINQ2 < 8 then ORDBNKINQ2 = 4;

else if BNKINQ2 < 10 then ORDBNKINQ2 = 5;

else ORDBNKINQ2 = 6;

if PRMINQ2 < 5 then ORDPRMINQ2= 1;

else if PRMINQ2 < 10 then ORDPRMINQ2 = 2;

else if PRMINQ2 < 15 then ORDPRMINQ2 = 3;

else if PRMINQ2 < 20 then ORDPRMINQ2 = 4;

else ORDPRMINQ2 = 5;

if TPCTSAT < 0.2 then ORDTPCTSAT= 1;

else if TPCTSAT < 0.4 then ORDTPCTSAT= 2;

else if TPCTSAT < 0.6 then ORDTPCTSAT = 3;

else if TPCTSAT < 0.8 then ORDTPCTSAT = 4;

else ORDTPCTSAT = 5;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var BNKINQS;

class ORDBNKINQS;

Proc means data=disc1;

Var goodbad;

class ORDBNKINQS;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDBNKINQS;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDBNKINQS;

Var BNKINQS GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

oddsORDTPCTSAT = (avg\_dep/(1-avg\_dep));

loddsORDTPCTSAT = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQS = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQS = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDBNKINQS;

Proc sort data=&d1;

by ORDBNKINQS;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDBNKINQS;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDBNKINQS;

histogram;

\*\*\*\*\*\*\*\*\*\*\*TBAL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var TBAL;

histogram;

Data disc1;

set newdata;

if TBAL < 1 then ORDTBAL= 1;

else if TBAL < 1000 then ORDTBAL = 2;

else if TBAL < 2000 then ORDTBAL = 3;

else if TBAL < 3000 then ORDTBAL = 4;

else ORDTBAL = 5;

if BNKINQS < 2.5 then ORDBNKINQS= 1;

else if BNKINQS < 5.0 then ORDBNKINQS = 2;

else if BNKINQS < 7.5 then ORDBNKINQS = 3;

else if BNKINQS < 10 then ORDBNKINQS = 4;

else if BNKINQS < 12.5 then ORDBNKINQS = 5;

else ORDBNKINQS = 6;

if BNKINQ2 < 2 then ORDBNKINQ2= 1;

else if BNKINQ2 < 4 then ORDBNKINQ2 = 2;

else if BNKINQ2 < 6 then ORDBNKINQ2 = 3;

else if BNKINQ2 < 8 then ORDBNKINQ2 = 4;

else if BNKINQ2 < 10 then ORDBNKINQ2 = 5;

else ORDBNKINQ2 = 6;

if PRMINQ2 < 5 then ORDPRMINQ2= 1;

else if PRMINQ2 < 10 then ORDPRMINQ2 = 2;

else if PRMINQ2 < 15 then ORDPRMINQ2 = 3;

else if PRMINQ2 < 20 then ORDPRMINQ2 = 4;

else ORDPRMINQ2 = 5;

if TPCTSAT < 0.2 then ORDTPCTSAT= 1;

else if TPCTSAT < 0.4 then ORDTPCTSAT= 2;

else if TPCTSAT < 0.6 then ORDTPCTSAT = 3;

else if TPCTSAT < 0.8 then ORDTPCTSAT = 4;

else ORDTPCTSAT = 5;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var TBAL;

class ORDTBAL;

Proc means data=disc1;

Var goodbad;

class ORDTBAL;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDTBAL;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDTBAL;

Var TBAL GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

oddsORDTPCTSAT = (avg\_dep/(1-avg\_dep));

loddsORDTPCTSAT = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQS = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQS = log(avg\_dep/(1-avg\_dep));

oddsORDTBAL = (avg\_dep/(1-avg\_dep));

loddsORDTBAL = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDTBAL;

Proc sort data=&d1;

by ORDTBAL;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDTBAL;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDTBAL;

histogram;

\*\*\*\*\*\*\*\*\*\*\*ROPEN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var ROPEN;

histogram;

Data disc1;

set newdata;

if ROPEN < 2.5 then ORDROPEN = 1;

else if ROPEN < 5.0 then ORDROPEN = 2;

else if ROPEN < 7.5 then ORDROPEN = 3;

else if ROPEN < 10 then ORDROPEN = 4;

else if ROPEN < 12.5 then ORDROPEN = 5;

else if ROPEN < 15.0 then ORDROPEN = 6;

else ORDROPEN = 7;

if TBAL < 1 then ORDTBAL= 1;

else if TBAL < 1000 then ORDTBAL = 2;

else if TBAL < 2000 then ORDTBAL = 3;

else if TBAL < 3000 then ORDTBAL = 4;

else ORDTBAL = 5;

if BNKINQS < 2.5 then ORDBNKINQS= 1;

else if BNKINQS < 5.0 then ORDBNKINQS = 2;

else if BNKINQS < 7.5 then ORDBNKINQS = 3;

else if BNKINQS < 10 then ORDBNKINQS = 4;

else if BNKINQS < 12.5 then ORDBNKINQS = 5;

else ORDBNKINQS = 6;

if BNKINQ2 < 2 then ORDBNKINQ2= 1;

else if BNKINQ2 < 4 then ORDBNKINQ2 = 2;

else if BNKINQ2 < 6 then ORDBNKINQ2 = 3;

else if BNKINQ2 < 8 then ORDBNKINQ2 = 4;

else if BNKINQ2 < 10 then ORDBNKINQ2 = 5;

else ORDBNKINQ2 = 6;

if PRMINQ2 < 5 then ORDPRMINQ2= 1;

else if PRMINQ2 < 10 then ORDPRMINQ2 = 2;

else if PRMINQ2 < 15 then ORDPRMINQ2 = 3;

else if PRMINQ2 < 20 then ORDPRMINQ2 = 4;

else ORDPRMINQ2 = 5;

if TPCTSAT < 0.2 then ORDTPCTSAT= 1;

else if TPCTSAT < 0.4 then ORDTPCTSAT= 2;

else if TPCTSAT < 0.6 then ORDTPCTSAT = 3;

else if TPCTSAT < 0.8 then ORDTPCTSAT = 4;

else ORDTPCTSAT = 5;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var ROPEN;

class ORDROPEN;

Proc means data=disc1;

Var goodbad;

class ORDROPEN;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDROPEN;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDROPEN;

Var ROPEN GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

oddsORDTPCTSAT = (avg\_dep/(1-avg\_dep));

loddsORDTPCTSAT = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQS = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQS = log(avg\_dep/(1-avg\_dep));

oddsORDTBAL = (avg\_dep/(1-avg\_dep));

loddsORDTBAL = log(avg\_dep/(1-avg\_dep));

oddsORDROPEN = (avg\_dep/(1-avg\_dep));

loddsORDROPEN = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDROPEN;

Proc sort data=&d1;

by ORDROPEN;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDROPEN;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDROPEN;

histogram;

proc univariate data=class.mergedout;

var ROPEN;

histogram;

\*\*\*\*\*\*\*\*\*\*\*PRMINQS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var PRMINQS;

histogram;

Data disc1;

set newdata;

if PRMINQS < 5 then ORDPRMINQS = 1;

else if PRMINQS < 10 then ORDPRMINQS = 2;

else if PRMINQS < 15 then ORDPRMINQS = 3;

else if PRMINQS < 20 then ORDPRMINQS = 4;

else if PRMINQS < 25 then ORDPRMINQS = 5;

else ORDPRMINQS = 6;

if ROPEN < 2.5 then ORDROPEN = 1;

else if ROPEN < 5.0 then ORDROPEN = 2;

else if ROPEN < 7.5 then ORDROPEN = 3;

else if ROPEN < 10 then ORDROPEN = 4;

else if ROPEN < 12.5 then ORDROPEN = 5;

else if ROPEN < 15.0 then ORDROPEN = 6;

else ORDROPEN = 7;

if TBAL < 1 then ORDTBAL= 1;

else if TBAL < 1000 then ORDTBAL = 2;

else if TBAL < 2000 then ORDTBAL = 3;

else if TBAL < 3000 then ORDTBAL = 4;

else ORDTBAL = 5;

if BNKINQS < 2.5 then ORDBNKINQS= 1;

else if BNKINQS < 5.0 then ORDBNKINQS = 2;

else if BNKINQS < 7.5 then ORDBNKINQS = 3;

else if BNKINQS < 10 then ORDBNKINQS = 4;

else if BNKINQS < 12.5 then ORDBNKINQS = 5;

else ORDBNKINQS = 6;

if BNKINQ2 < 2 then ORDBNKINQ2= 1;

else if BNKINQ2 < 4 then ORDBNKINQ2 = 2;

else if BNKINQ2 < 6 then ORDBNKINQ2 = 3;

else if BNKINQ2 < 8 then ORDBNKINQ2 = 4;

else if BNKINQ2 < 10 then ORDBNKINQ2 = 5;

else ORDBNKINQ2 = 6;

if PRMINQ2 < 5 then ORDPRMINQ2= 1;

else if PRMINQ2 < 10 then ORDPRMINQ2 = 2;

else if PRMINQ2 < 15 then ORDPRMINQ2 = 3;

else if PRMINQ2 < 20 then ORDPRMINQ2 = 4;

else ORDPRMINQ2 = 5;

if TPCTSAT < 0.2 then ORDTPCTSAT= 1;

else if TPCTSAT < 0.4 then ORDTPCTSAT= 2;

else if TPCTSAT < 0.6 then ORDTPCTSAT = 3;

else if TPCTSAT < 0.8 then ORDTPCTSAT = 4;

else ORDTPCTSAT = 5;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var PRMINQS;

class ORDPRMINQS;

Proc means data=disc1;

Var goodbad;

class ORDPRMINQS;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDPRMINQS;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDPRMINQS;

Var PRMINQS GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

oddsORDTPCTSAT = (avg\_dep/(1-avg\_dep));

loddsORDTPCTSAT = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQS = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQS = log(avg\_dep/(1-avg\_dep));

oddsORDTBAL = (avg\_dep/(1-avg\_dep));

loddsORDTBAL = log(avg\_dep/(1-avg\_dep));

oddsORDROPEN = (avg\_dep/(1-avg\_dep));

loddsORDROPEN = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQS = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQS = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDPRMINQS;

Proc sort data=&d1;

by ORDPRMINQS;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDPRMINQS;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDPRMINQS;

histogram;

proc univariate data=class.mergedout;

var PRMINQS;

histogram;

\*\*\*\*\*\*\*\*\*\*\*TOPENB50\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var TOPENB50;

histogram;

Data disc1;

set newdata;

if TOPENB50 = 0 then ORDTOPENB50= 1;

else if TOPENB50 < 2 then ORDTOPENB50 = 2;

else if TOPENB50 < 4 then ORDTOPENB50 = 3;

else if TOPENB50 < 6 then ORDTOPENB50 = 4;

else if TOPENB50 < 8 then ORDTOPENB50 = 5;

else ORDTOPENB50 = 6;

if PRMINQS < 5 then ORDPRMINQS = 1;

else if PRMINQS < 10 then ORDPRMINQS = 2;

else if PRMINQS < 15 then ORDPRMINQS = 3;

else if PRMINQS < 20 then ORDPRMINQS = 4;

else if PRMINQS < 25 then ORDPRMINQS = 5;

else ORDPRMINQS = 6;

if ROPEN < 2.5 then ORDROPEN = 1;

else if ROPEN < 5.0 then ORDROPEN = 2;

else if ROPEN < 7.5 then ORDROPEN = 3;

else if ROPEN < 10 then ORDROPEN = 4;

else if ROPEN < 12.5 then ORDROPEN = 5;

else if ROPEN < 15.0 then ORDROPEN = 6;

else ORDROPEN = 7;

if TBAL < 1 then ORDTBAL= 1;

else if TBAL < 1000 then ORDTBAL = 2;

else if TBAL < 2000 then ORDTBAL = 3;

else if TBAL < 3000 then ORDTBAL = 4;

else ORDTBAL = 5;

if BNKINQS < 2.5 then ORDBNKINQS= 1;

else if BNKINQS < 5.0 then ORDBNKINQS = 2;

else if BNKINQS < 7.5 then ORDBNKINQS = 3;

else if BNKINQS < 10 then ORDBNKINQS = 4;

else if BNKINQS < 12.5 then ORDBNKINQS = 5;

else ORDBNKINQS = 6;

if BNKINQ2 < 2 then ORDBNKINQ2= 1;

else if BNKINQ2 < 4 then ORDBNKINQ2 = 2;

else if BNKINQ2 < 6 then ORDBNKINQ2 = 3;

else if BNKINQ2 < 8 then ORDBNKINQ2 = 4;

else if BNKINQ2 < 10 then ORDBNKINQ2 = 5;

else ORDBNKINQ2 = 6;

if PRMINQ2 < 5 then ORDPRMINQ2= 1;

else if PRMINQ2 < 10 then ORDPRMINQ2 = 2;

else if PRMINQ2 < 15 then ORDPRMINQ2 = 3;

else if PRMINQ2 < 20 then ORDPRMINQ2 = 4;

else ORDPRMINQ2 = 5;

if TPCTSAT < 0.2 then ORDTPCTSAT= 1;

else if TPCTSAT < 0.4 then ORDTPCTSAT= 2;

else if TPCTSAT < 0.6 then ORDTPCTSAT = 3;

else if TPCTSAT < 0.8 then ORDTPCTSAT = 4;

else ORDTPCTSAT = 5;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var TOPENB50;

class ORDTOPENB50;

Proc means data=disc1;

Var goodbad;

class ORDTOPENB50;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDTOPENB50;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDTOPENB50;

Var TOPENB50 GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

oddsORDTPCTSAT = (avg\_dep/(1-avg\_dep));

loddsORDTPCTSAT = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQS = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQS = log(avg\_dep/(1-avg\_dep));

oddsORDTBAL = (avg\_dep/(1-avg\_dep));

loddsORDTBAL = log(avg\_dep/(1-avg\_dep));

oddsORDROPEN = (avg\_dep/(1-avg\_dep));

loddsORDROPEN = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQS = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQS = log(avg\_dep/(1-avg\_dep));

oddsORDTOPENB50 = (avg\_dep/(1-avg\_dep));

loddsORDTOPENB50 = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDTOPENB50;

Proc sort data=&d1;

by ORDTOPENB50;

Data modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDTOPENB50;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDTOPENB50;

histogram;

\*\*\*\*\*\*\*\*\*\*\*INQ12\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

proc univariate data=newdata;

var INQ12;

histogram;

Data disc1;

set newdata;

if INQ12 = 0 then ORDINQ12 = 1;

else if INQ12 = 1 then ORDINQ12 = 2;

else if INQ12 = 2 then ORDINQ12 = 3;

else if INQ12 = 3 then ORDINQ12 = 4;

else ORDINQ12 = 5;

if TOPENB50 = 0 then ORDTOPENB50= 1;

else if TOPENB50 < 2 then ORDTOPENB50 = 2;

else if TOPENB50 < 4 then ORDTOPENB50 = 3;

else if TOPENB50 < 6 then ORDTOPENB50 = 4;

else if TOPENB50 < 8 then ORDTOPENB50 = 5;

else ORDTOPENB50 = 6;

if PRMINQS < 5 then ORDPRMINQS = 1;

else if PRMINQS < 10 then ORDPRMINQS = 2;

else if PRMINQS < 15 then ORDPRMINQS = 3;

else if PRMINQS < 20 then ORDPRMINQS = 4;

else if PRMINQS < 25 then ORDPRMINQS = 5;

else ORDPRMINQS = 6;

if ROPEN < 2.5 then ORDROPEN = 1;

else if ROPEN < 5.0 then ORDROPEN = 2;

else if ROPEN < 7.5 then ORDROPEN = 3;

else if ROPEN < 10 then ORDROPEN = 4;

else if ROPEN < 12.5 then ORDROPEN = 5;

else if ROPEN < 15.0 then ORDROPEN = 6;

else ORDROPEN = 7;

if TBAL < 1 then ORDTBAL= 1;

else if TBAL < 1000 then ORDTBAL = 2;

else if TBAL < 2000 then ORDTBAL = 3;

else if TBAL < 3000 then ORDTBAL = 4;

else ORDTBAL = 5;

if BNKINQS < 2.5 then ORDBNKINQS= 1;

else if BNKINQS < 5.0 then ORDBNKINQS = 2;

else if BNKINQS < 7.5 then ORDBNKINQS = 3;

else if BNKINQS < 10 then ORDBNKINQS = 4;

else if BNKINQS < 12.5 then ORDBNKINQS = 5;

else ORDBNKINQS = 6;

if BNKINQ2 < 2 then ORDBNKINQ2= 1;

else if BNKINQ2 < 4 then ORDBNKINQ2 = 2;

else if BNKINQ2 < 6 then ORDBNKINQ2 = 3;

else if BNKINQ2 < 8 then ORDBNKINQ2 = 4;

else if BNKINQ2 < 10 then ORDBNKINQ2 = 5;

else ORDBNKINQ2 = 6;

if PRMINQ2 < 5 then ORDPRMINQ2= 1;

else if PRMINQ2 < 10 then ORDPRMINQ2 = 2;

else if PRMINQ2 < 15 then ORDPRMINQ2 = 3;

else if PRMINQ2 < 20 then ORDPRMINQ2 = 4;

else ORDPRMINQ2 = 5;

if TPCTSAT < 0.2 then ORDTPCTSAT= 1;

else if TPCTSAT < 0.4 then ORDTPCTSAT= 2;

else if TPCTSAT < 0.6 then ORDTPCTSAT = 3;

else if TPCTSAT < 0.8 then ORDTPCTSAT = 4;

else ORDTPCTSAT = 5;

if MOSOPEN < 250 then ORDMOSOPEN= 1;

else if MOSOPEN < 500 then ORDMOSOPEN = 2;

else if MOSOPEN < 1000 then ORDMOSOPEN = 3;

else if MOSOPEN < 1500 then ORDMOSOPEN = 4;

else if MOSOPEN < 2000 then ORDMOSOPEN = 5;

else if MOSOPEN < 2500 then ORDMOSOPEN = 6;

else ORDMOSOPEN = 7;

if BRBAL50 < 1 then ORDBRBAL50= 1;

else if BRBAL50 < 3 then ORDBRBAL50 = 2;

else if BRBAL50 < 5 then ORDBRBAL50 = 3;

else ORDBRBAL50 = 4;

if TADB < 0.2 then ORDTADB= 1;

else if TADB < 0.4 then ORDTADB = 2;

else if TADB < 0.6 then ORDTADB = 3;

else if TADB < 0.8 then ORDTADB = 4;

else ORDTADB = 5;

if TADB25 < 2 then ORDTADB25= 1;

else if TADB25 < 4 then ORDTADB25 = 2;

else if TADB25 < 6 then ORDTADB25 = 3;

else if TADB25 < 8 then ORDTADB25 = 4;

else if TADB25 < 10 then ORDTADB25 = 5;

else ORDTADB25 = 6;

if FININQS = 1 then ORDFININQS= 1;

else if FININQS = 2 then ORDFININQS = 2;

else ORDFININQS = 3;

if TROPENEX < 1 then ORDTROPENEX= 1;

else if TROPENEX < 20 then ORDTROPENEX = 2;

else if TROPENEX < 30 then ORDTROPENEX = 3;

else ORDTROPENEX = 4;

if BRNEW = 1 then ORDBRNEW= 1;

else if BRNEW < 10 then ORDBRNEW = 2;

else if BRNEW < 20 then ORDBRNEW = 3;

else if BRNEW < 30 then ORDBRNEW = 4;

else ORDBRNEW = 5;

if DCWCRATE= 0 then ORDDCWCRATE= 1;

else ORDDCWCRATE= 2;

if BRWCRATE = 1 then ORDBRWCRATE= 1;

else ORDBRWCRATE= 2;

if TRCR49 = 0 then ORDTRCR49= 1;

else if TRCR49 < 2 then ORDTRCR49 = 2;

else if TRCR49 < 4 then ORDTRCR49 = 3;

else if TRCR49 < 6 then ORDTRCR49 = 4;

else if TRCR49 < 8 then ORDTRCR49 = 5;

else if TRCR49 < 10 then ORDTRCR49 = 6;

else ORDTRCR49 = 7;

if BRADB = 0 then ORDBRADB= 1;

else if BRADB < 0.2 then ORDBRADB = 2;

else if BRADB < 0.4 then ORDBRADB = 3;

else if BRADB < 0.6 then ORDBRADB = 4;

else if BRADB < 0.8 then ORDBRADB = 5;

else if BRADB < 1.0 then ORDBRADB = 6;

else ORDBRADB = 7;

if OT12PTOT = 0 then ORDOT12PTOT= 1;

else if OT12PTOT < 0.2 then ORDOT12PTOT = 2;

else if OT12PTOT < 0.4 then ORDOT12PTOT = 3;

else if OT12PTOT < 0.6 then ORDOT12PTOT = 4;

else if OT12PTOT < 0.8 then ORDOT12PTOT = 5;

else ORDOT12PTOT = 6;

if delqid < 3 then goodbad=0;

else goodbad =1;

drop OROT12PTOT;

Proc means data = disc1;

var INQ12;

class ORDINQ12;

Proc means data=disc1;

Var goodbad;

class ORDINQ12;

output out=test mean=meangoodbad;

symbol v=square color = black i=join;

Proc gplot data = test;

plot meangoodbad\*ORDINQ12;

quit;

%let d1 = disc1;

Proc means data = &d1 nmiss;

Proc Summary data=&d1 mean missing std;

Class ORDINQ12;

Var INQ12 GOODBAD;

Output out=summary mean=avg\_ind avg\_dep;

Proc print data = summary;

Data summary;

Set summary;

oddsORDOT12PTOT = (avg\_dep/(1-avg\_dep));

loddsORDOT12PTOT = log(avg\_dep/(1-avg\_dep));

oddsORDBRADB = (avg\_dep/(1-avg\_dep));

loddsORDBRADB = log(avg\_dep/(1-avg\_dep));

oddsORDTRCR49 = (avg\_dep/(1-avg\_dep));

loddsORDTRCR49 = log(avg\_dep/(1-avg\_dep));

oddsORDBRWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDBRWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDDCWCRATE = (avg\_dep/(1-avg\_dep));

loddsORDDCWCRATE = log(avg\_dep/(1-avg\_dep));

oddsORDBRNEW = (avg\_dep/(1-avg\_dep));

loddsORDBRNEW = log(avg\_dep/(1-avg\_dep));

oddsORDTROPENEX = (avg\_dep/(1-avg\_dep));

loddsORDTROPENEX = log(avg\_dep/(1-avg\_dep));

oddsORDFININQS = (avg\_dep/(1-avg\_dep));

loddsORDFININQS = log(avg\_dep/(1-avg\_dep));

oddsORDTADB25 = (avg\_dep/(1-avg\_dep));

loddsORDTADB25 = log(avg\_dep/(1-avg\_dep));

oddsORDTADB = (avg\_dep/(1-avg\_dep));

loddsORDTADB = log(avg\_dep/(1-avg\_dep));

oddsORDBRBAL50 = (avg\_dep/(1-avg\_dep));

loddsORDBRBAL50 = log(avg\_dep/(1-avg\_dep));

oddsORDMOSOPEN = (avg\_dep/(1-avg\_dep));

loddsORDMOSOPEN = log(avg\_dep/(1-avg\_dep));

oddsORDTPCTSAT = (avg\_dep/(1-avg\_dep));

loddsORDTPCTSAT = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQ2 = log(avg\_dep/(1-avg\_dep));

oddsORDBNKINQ2 = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQ2 = log(avg\_dep/(1-avg\_dep));

testORDBNKINQ2 = ((1-avg\_dep)/(avg\_dep));

oddsORDBNKINQS = (avg\_dep/(1-avg\_dep));

loddsORDBNKINQS = log(avg\_dep/(1-avg\_dep));

oddsORDTBAL = (avg\_dep/(1-avg\_dep));

loddsORDTBAL = log(avg\_dep/(1-avg\_dep));

oddsORDROPEN = (avg\_dep/(1-avg\_dep));

loddsORDROPEN = log(avg\_dep/(1-avg\_dep));

oddsORDPRMINQS = (avg\_dep/(1-avg\_dep));

loddsORDPRMINQS = log(avg\_dep/(1-avg\_dep));

oddsORDTOPENB50 = (avg\_dep/(1-avg\_dep));

loddsORDTOPENB50 = log(avg\_dep/(1-avg\_dep));

oddsORDINQ12 = (avg\_dep/(1-avg\_dep));

loddsORDINQ12 = log(avg\_dep/(1-avg\_dep));

Proc Print data=summary;

\*Code to merge the summary file values back into the dataset;

Proc sort data=summary;

by ORDINQ12;

Proc sort data=&d1;

by ORDINQ12;

Data class.modelingfile (drop = \_TYPE\_ \_FREQ\_ avg\_ind avg\_dep);

Merge summary &D1;

by ORDINQ12;

if matchkey = . then delete;

proc univariate data=disc1;

var ORDINQ12;

histogram;

proc means data=class.modelingfile N nmiss;

\*DISCREDITIZATION 2\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*ot12ptot;

%disc2(pval\_col = 0.15,var1=OT12PTOT,d1=class.modelingfile);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 2;

else if rank = 1 then rank = 2;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 5;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data class.modelingfile;

Merge ranked1 temp;

by rank;

Run;

proc means data=class.modelingfile N nmiss;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

Data disc2ot12;

set class.modelingfile;

if rank = 2 then ordeqot12ptot = 1;

else if rank = 3 then ordeqot12ptot = 2;

else if rank = 5 then ordeqot12ptot = 3;

else if rank = 6 then ordeqot12ptot = 4;

else if rank = 7 then ordeqot12ptot = 5;

else if rank = 8 then ordeqot12ptot = 6;

else if rank = 9 then ordeqot12ptot = 7;

else rank = rank;

odseqot12ptot = (avg\_dep)/(1-avg\_dep);

lodseqot12ptot = log ((avg\_dep)/(1-avg\_dep));

\*testodseqot12ptot = (1-avg\_dep)/(avg\_dep);

run;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2ot12 (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2ot12;

Run;

Proc freq data=disc2ot12;

table ordeqot12ptot ODSEQot12ptot LODSEQot12ptot;

Run;

Proc Means data = disc2ot12;

class ordeqot12ptot;

var ot12ptot;

run;

proc means data=disc2ot12 N nmiss;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*brnew;

%disc2(pval\_col = 0.15,var1=brnew,d1=disc2ot12);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2ot12;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

Data disc2brnew;

set disc2ot12;

if rank = 0 then ordeqbrnew = 1;

if rank = 1 then ordeqbrnew = 2;

if rank = 2 then ordeqbrnew = 3;

else if rank = 3 then ordeqbrnew = 4;

else if rank = 4 then ordeqbrnew = 5;

else if rank = 5 then ordeqbrnew = 6;

else if rank = 6 then ordeqbrnew = 7;

else if rank = 7 then ordeqbrnew = 8;

else if rank = 8 then ordeqbrnew = 9;

else if rank = 9 then ordeqbrnew = 10;

else rank = rank;

odseqbrnew = (avg\_dep)/(1-avg\_dep);

lodseqbrnew = log ((avg\_dep)/(1-avg\_dep));

\*testodseqbrnew = (1-avg\_dep)/(avg\_dep);

run;

proc freq data=disc2brnew;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2brnew (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2brnew;

Run;

Proc freq data=disc2brnew;

table ordeqbrnew ODSEQbrnew LODSEQbrnew;

Run;

Proc Means data = disc2brnew;

class ordeqbrnew;

var brnew;

run;

proc means data=disc2brnew N nmiss;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*BRADB;

%disc2(pval\_col = 0.15,var1=bradb,d1=disc2brnew);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2brnew;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2brad;

set disc2brnew;

if rank = 0 then ordeqbradb = 1;

if rank = 1 then ordeqbradb = 2;

if rank = 2 then ordeqbradb = 3;

else if rank = 3 then ordeqbradb = 4;

else if rank = 4 then ordeqbradb = 5;

else if rank = 5 then ordeqbradb = 6;

else if rank = 6 then ordeqbradb = 7;

else if rank = 7 then ordeqbradb = 8;

else if rank = 8 then ordeqbradb = 9;

else if rank = 9 then ordeqbradb = 10;

else rank = rank;

odseqbradb = (avg\_dep)/(1-avg\_dep);

lodseqbradb = log ((avg\_dep)/(1-avg\_dep));

\*testodseqbradb = (1-avg\_dep)/(avg\_dep);

run;

proc freq data=disc2brad;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2brad (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2brad;

Run;

Proc freq data=disc2brad;

table ordeqbradb ODSEQbradb LODSEQbradb;

Run;

Proc Means data = disc2brad;

class ordeqbradb;

var bradb;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*TRCR49;

%disc2(pval\_col = 0.15,var1=TRCR49,d1=disc2brad);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 4;

else if rank = 1 then rank = 4;

else if rank = 2 then rank = 4;

else if rank = 3 then rank = 4;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 6;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2brad;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

\*second rank should be 1, 2, 3, 4, 5... etc;

Data disc2trcr49;

set disc2brad;

if rank = 4 then ordeqtrcr49 = 1;

if rank = 6 then ordeqtrcr49 = 2;

else if rank = 7 then ordeqtrcr49 = 3;

else if rank = 8 then ordeqtrcr49 = 4;

else if rank = 9 then ordeqtrcr49 = 5;

else rank = rank;

odseqtrcr49 = (avg\_dep)/(1-avg\_dep);

lodseqtrcr49 = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2trcr49;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2trcr49 (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2trcr49;

Run;

Proc freq data=disc2trcr49;

table ordeqtrcr49 ODSEQtrcr49 LODSEQtrcr49;

Run;

Proc Means data = disc2trcr49;

class ordeqtrcr49;

var trcr49;

run;

\*\*\*\*\*\*\*\*\*\*\*BRWCRATE;

%disc2(pval\_col = 0.15,var1=BRWCRATE,d1=disc2trcr49);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

Proc univariate data=disc2trcr49 noprint;

Var BRWCRATE;

histogram;

Run;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 7;

else if rank = 1 then rank = 7;

else if rank = 2 then rank = 7;

else if rank = 3 then rank = 7;

else if rank = 4 then rank = 7;

else if rank = 5 then rank = 7;

else if rank = 6 then rank = 7;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2trcr49;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2crate;

set disc2trcr49;

if rank = 7 then ordeqbrwcrate = 1;

if rank = 8 then ordeqbrwcrate = 2;

if rank = 9 then ordeqbrwcrate = 3;

else rank = rank;

odseqbrwcrate = (avg\_dep)/(1-avg\_dep);

lodseqbrwcrate = log ((avg\_dep)/(1-avg\_dep));

run;

\*WTFFFFFFF WHERE ARE THE VARIABLES;

proc freq data=disc2crate;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2crate (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2crate;

Run;

Proc freq data=disc2crate;

table ordeqbrwcrate ODSEQbrwcrate LODSEQbrwcrate;

Run;

Proc Means data = disc2crate;

class ordeqbrwcrate;

var brwcrate;

run;

Proc univariate data=disc2crate;

Var BRWCRATE;

histogram;

Run;

\*\*\*\*\*\*\*\*\*\*\*DCWCRATE;

%disc2(pval\_col = 0.15,var1=dcwcrate,d1=disc2crate);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 5;

else if rank = 2 then rank = 5;

else if rank = 3 then rank = 5;

else if rank = 4 then rank = 5;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 6;

else if rank = 8 then rank = 6;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2crate;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2dcrate;

set disc2crate;

if rank = 0 then ordeqdcwcrate = 1;

else if rank = 5 then ordeqdcwcrate = 2;

else if rank = 6 then ordeqdcwcrate = 3;

else if rank = 9 then ordeqdcwcrate = 4;

else rank = rank;

odseqdcwcrate = (avg\_dep)/(1-avg\_dep);

lodseqdcwcrate = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2dcrate;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2dcrate (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2dcrate;

Run;

Proc freq data=disc2dcrate;

table ordeqdcwcrate ODSEQdcwcrate LODSEQdcwcrate;

Run;

Proc Means data = disc2dcrate;

class ordeqdcwcrate;

var dcwcrate;

run;

\*\*\*\*\*\*\*\*\*\*\*TROPENEX;

%disc2(pval\_col = 0.15,var1=tropenex,d1=disc2dcrate);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 2;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 7;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2dcrate;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2tropen;

set disc2dcrate;

if rank = 0 then ordeqtropenex = 1;

if rank = 2 then ordeqtropenex = 2;

if rank = 3 then ordeqtropenex = 3;

else if rank = 4 then ordeqtropenex = 4;

else if rank = 5 then ordeqtropenex = 5;

else if rank = 7 then ordeqtropenex = 6;

else if rank = 8 then ordeqtropenex = 7;

else if rank = 9 then ordeqtropenex = 8;

else rank = rank;

odseqtropenex = (avg\_dep)/(1-avg\_dep);

lodseqtropenex = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2tropen;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2tropen (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2tropen;

Run;

Proc freq data=disc2tropen;

table ordeqtropenex ODSEQtropenex LODSEQtropenex;

Run;

Proc Means data = disc2tropen;

class ordeqtropenex;

var tropenex;

run;

\*\*\*\*\*\*\*\*\*\*\*\*FININQS;

%disc2(pval\_col = 0.15,var1=fininqs,d1=disc2tropen);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 2;

else if rank = 1 then rank = 2;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 9;

else if rank = 8 then rank = 9;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2tropen;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2fin;

set disc2tropen;

if rank = 2 then ordeqfininqs = 1;

if rank = 3 then ordeqfininqs = 2;

if rank = 4 then ordeqfininqs = 3;

else if rank = 5 then ordeqfininqs = 4;

else if rank = 6 then ordeqfininqs = 5;

else if rank = 9 then ordeqfininqs = 6;

else rank = rank;

odseqfininqs = (avg\_dep)/(1-avg\_dep);

lodseqfininqs = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2fin;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2fin (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2fin;

Run;

Proc freq data=disc2fin;

table ordeqfininqs ODSEQfininqs LODSEQfininqs;

Run;

Proc Means data = disc2fin;

class ordeqfininqs;

var fininqs;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*TADB25;

%disc2(pval\_col = 0.15,var1=tadb25,d1=disc2fin);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 1;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 4;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2fin;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2tad25;

set disc2fin;

if rank = 1 then ordeqtadb25 = 1;

if rank = 2 then ordeqtadb25 = 2;

if rank = 4 then ordeqtadb25 = 3;

else if rank = 5 then ordeqtadb25 = 4;

else if rank = 6 then ordeqtadb25 = 5;

else if rank = 7 then ordeqtadb25 = 6;

else if rank = 8 then ordeqtadb25 = 7;

else if rank = 9 then ordeqtadb25 = 8;

else rank = rank;

odseqtadb25 = (avg\_dep)/(1-avg\_dep);

lodseqtadb25 = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2tad25;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2tad25 (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2tad25;

Run;

Proc freq data=disc2tad25;

table ordeqtadb25 ODSEQtadb25 LODSEQtadb25;

Run;

Proc Means data = disc2tad25;

class ordeqtadb25;

var tadb25;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TADB;

%disc2(pval\_col = 0.15,var1=tadb,d1=disc2tad25);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2tad25;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2tad;

set disc2tad25;

if rank = 0 then ordeqtadb = 1;

if rank = 1 then ordeqtadb = 2;

if rank = 2 then ordeqtadb = 3;

else if rank = 3 then ordeqtadb = 4;

else if rank = 4 then ordeqtadb = 5;

else if rank = 5 then ordeqtadb = 6;

else if rank = 6 then ordeqtadb = 7;

else if rank = 7 then ordeqtadb = 8;

else if rank = 8 then ordeqtadb = 9;

else if rank = 9 then ordeqtadb = 10;

else rank = rank;

odseqtadb = (avg\_dep)/(1-avg\_dep);

lodseqtadb = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2tad;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2tad (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2tad;

Run;

Proc freq data=disc2tad;

table ordeqtadb ODSEQtadb LODSEQtadb;

Run;

Proc Means data = disc2tad;

class ordeqtadb;

var tadb;

run;

\*\*\*\*\*\*\*\*\*\*\*\*BRBAL50;

%disc2(pval\_col = 0.15,var1=brbal50,d1=disc2tad);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 1;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 3;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 5;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2tad;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2brb;

set disc2tad;

if rank = 1 then ordeqbrbal50 = 1;

if rank = 3 then ordeqbrbal50 = 2;

if rank = 5 then ordeqbrbal50 = 3;

if rank = 6 then ordeqbrbal50 = 4;

else if rank = 7 then ordeqbrbal50 = 5;

else if rank = 8 then ordeqbrbal50 = 6;

else if rank = 9 then ordeqbrbal50 = 7;

else rank = rank;

odseqbrbal50 = (avg\_dep)/(1-avg\_dep);

lodseqbrbal50 = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2brb;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2brb (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2brb;

Run;

Proc freq data=disc2brb;

table ordeqbrbal50 ODSEQbrbal50 LODSEQbrbal50;

Run;

Proc Means data = disc2brb;

class ordeqbrbal50;

var brbal50;

run;

\*\*\*\*\*\*\*\*\*\*\*\*MOSOPEN;

%disc2(pval\_col = 0.15,var1=mosopen,d1=disc2brb);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 3;

else if rank = 2 then rank = 3;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2brb;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2mos;

set disc2brb;

if rank = 0 then ordeqmosopen = 1;

else if rank = 3 then ordeqmosopen = 2;

else if rank = 4 then ordeqmosopen = 3;

else if rank = 5 then ordeqmosopen = 4;

else if rank = 6 then ordeqmosopen = 5;

else if rank = 7 then ordeqmosopen = 6;

else if rank = 8 then ordeqmosopen = 7;

else if rank = 9 then ordeqmosopen = 8;

else rank = rank;

odseqmosopen = (avg\_dep)/(1-avg\_dep);

lodseqmosopen = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2mos;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2mos (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2mos;

Run;

Proc freq data=disc2mos;

table ordeqmosopen ODSEQmosopen LODSEQmosopen;

Run;

Proc Means data = disc2mos;

class ordeqmosopen;

var mosopen;

run;

\*\*\*\*\*\*\*\*\*\*\*\*TPCTSAT;

%disc2(pval\_col = 0.15,var1=tpctsat,d1=disc2mos);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2mos;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2tpc;

set disc2mos;

if rank = 0 then ordeqtpctsat = 1;

if rank = 1 then ordeqtpctsat = 2;

if rank = 2 then ordeqtpctsat = 3;

else if rank = 3 then ordeqtpctsat = 4;

else if rank = 4 then ordeqtpctsat = 5;

else if rank = 5 then ordeqtpctsat = 6;

else if rank = 6 then ordeqtpctsat = 7;

else if rank = 7 then ordeqtpctsat = 8;

else if rank = 8 then ordeqtpctsat = 9;

else if rank = 9 then ordeqtpctsat = 10;

else rank = rank;

odseqtpctsat = (avg\_dep)/(1-avg\_dep);

lodseqtpctsat = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2tpc;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2tpc (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2tpc;

Run;

Proc freq data=disc2tpc;

table ordeqtpctsat ODSEQtpctsat LODSEQtpctsat;

Run;

Proc Means data = disc2tpc;

class ordeqtpctsat;

var tpctsat;

run;

\*\*\*\*\*\*\*\*\*\*\*\*PRMINQ2;

%disc2(pval\_col = 0.15,var1=prminq2,d1=disc2tpc);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2tpc;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2perm;

set disc2tpc;

if rank = 0 then ordeqprminq2 = 1;

if rank = 1 then ordeqprminq2 = 2;

if rank = 2 then ordeqprminq2 = 3;

else if rank = 3 then ordeqprminq2 = 4;

else if rank = 4 then ordeqprminq2 = 5;

else if rank = 5 then ordeqprminq2 = 6;

else if rank = 6 then ordeqprminq2 = 7;

else if rank = 7 then ordeqprminq2 = 8;

else if rank = 8 then ordeqprminq2 = 9;

else if rank = 9 then ordeqprminq2 = 10;

else rank = rank;

odseqprminq2 = (avg\_dep)/(1-avg\_dep);

lodseqprminq2 = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2perm;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2perm (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2perm;

Run;

Proc freq data=disc2perm;

table ordeqprminq2 ODSEQprminq2 LODSEQprminq2;

Run;

Proc Means data = disc2perm;

class ordeqprminq2;

var prminq2;

run;

\*\*\*\*\*\*\*\*\*\*\*\*BNKINQ2;

%disc2(pval\_col = 0.15,var1=bnkinq2,d1=disc2perm);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 1;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2perm;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2bnk;

set disc2perm;

if rank = 1 then ordeqbnkinq2 = 1;

if rank = 2 then ordeqbnkinq2 = 2;

if rank = 3 then ordeqbnkinq2 = 3;

else if rank = 4 then ordeqbnkinq2 = 4;

else if rank = 5 then ordeqbnkinq2 = 5;

else if rank = 6 then ordeqbnkinq2 = 6;

else if rank = 7 then ordeqbnkinq2 = 7;

else if rank = 8 then ordeqbnkinq2 = 8;

else if rank = 9 then ordeqbnkinq2 = 9;

else rank = rank;

odseqbnkinq2 = (avg\_dep)/(1-avg\_dep);

lodseqbnkinq2 = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2bnk;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2bnk (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2bnk;

Run;

Proc freq data=disc2bnk;

table ordeqbnkinq2 ODSEQbnkinq2 LODSEQbnkinq2;

Run;

Proc Means data = disc2bnk;

class ordeqbnkinq2;

var bnkinq2;

run;

\*\*\*\*\*\*\*\*\*\*\*\*BNKINQS;

%disc2(pval\_col = 0.15,var1=bnkinqs,d1=disc2bnk);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 1;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2bnk;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2bnks;

set disc2bnk;

if rank = 1 then ordeqbnkinqs = 1;

if rank = 2 then ordeqbnkinqs = 2;

else if rank = 3 then ordeqbnkinqs = 3;

else if rank = 4 then ordeqbnkinqs = 4;

else if rank = 5 then ordeqbnkinqs = 5;

else if rank = 6 then ordeqbnkinqs = 6;

else if rank = 7 then ordeqbnkinqs = 7;

else if rank = 8 then ordeqbnkinqs = 8;

else if rank = 9 then ordeqbnkinqs = 9;

else rank = rank;

odseqbnkinqs = (avg\_dep)/(1-avg\_dep);

lodseqbnkinqs = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2bnks;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2bnks (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2bnks;

Run;

Proc freq data=disc2bnks;

table ordeqbnkinqs ODSEQbnkinqs LODSEQbnkinqs;

Run;

Proc Means data = disc2bnks;

class ordeqbnkinqs;

var bnkinqs;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TBAL;

%disc2(pval\_col = 0.15,var1=tbal,d1=disc2bnks);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 3;

else if rank = 2 then rank = 3;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 9;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2bnks;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2tbal;

set disc2bnks;

if rank = 0 then ordeqtbal = 1;

if rank = 3 then ordeqtbal = 2;

else if rank = 4 then ordeqtbal = 3;

else if rank = 5 then ordeqtbal = 4;

else if rank = 6 then ordeqtbal = 5;

else if rank = 7 then ordeqtbal = 6;

else if rank = 9 then ordeqtbal = 7;

else rank = rank;

odseqtbal = (avg\_dep)/(1-avg\_dep);

lodseqtbal = log ((avg\_dep)/(1-avg\_dep));

if lodseqtbal = . then lodseqtbal = log(0.0000001);

run;

proc freq data=disc2tbal;

tables avg\_dep odseqtbal lodseqtbal;

\*\*\*\*\*\*\* this is weird why is there a warning ;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2tbal (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2tbal;

Run;

Proc freq data=disc2tbal;

table ordeqtbal ODSEQtbal LODSEQtbal;

Run;

Proc Means data = disc2tbal;

class ordeqtbal;

var tbal;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*ROPEN;

%disc2(pval\_col = 0.15,var1=ropen,d1=disc2tbal);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2tbal;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2rop;

set disc2tbal;

if rank = 0 then ordeqropen = 1;

if rank = 1 then ordeqropen = 2;

if rank = 2 then ordeqropen = 3;

else if rank = 3 then ordeqropen = 4;

else if rank = 4 then ordeqropen = 5;

else if rank = 5 then ordeqropen = 6;

else if rank = 6 then ordeqropen = 7;

else if rank = 7 then ordeqropen = 8;

else if rank = 8 then ordeqropen = 9;

else if rank = 9 then ordeqropen = 10;

else rank = rank;

odseqropen = (avg\_dep)/(1-avg\_dep);

lodseqropen = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2rop;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2rop (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2rop;

Run;

Proc freq data=disc2rop;

table ordeqropen ODSEQropen LODSEQropen;

Run;

Proc Means data = disc2rop;

class ordeqropen;

var ropen;

run;

\*\*\*\*\*\*\*\*\*\*\*\*PRMINQS;

%disc2(pval\_col = 0.15,var1=prminqs,d1=disc2rop);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2rop;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2press;

set disc2rop;

if rank = 0 then ordeqprminqs = 1;

if rank = 1 then ordeqprminqs = 2;

if rank = 2 then ordeqprminqs = 3;

else if rank = 3 then ordeqprminqs = 4;

else if rank = 4 then ordeqprminqs = 5;

else if rank = 5 then ordeqprminqs = 6;

else if rank = 6 then ordeqprminqs = 7;

else if rank = 7 then ordeqprminqs = 8;

else if rank = 8 then ordeqprminqs = 9;

else if rank = 9 then ordeqprminqs = 10;

else rank = rank;

odseqprminqs = (avg\_dep)/(1-avg\_dep);

lodseqprminqs = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2press;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2press (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2press;

Run;

Proc freq data=disc2press;

table ordeqprminqs ODSEQprminqs LODSEQprminqs;

Run;

Proc Means data = disc2press;

class ordeqprminqs;

var prminqs;

run;

\*\*\*\*\*\*\*\*\*\*\*TOPENB50;

%disc2(pval\_col = 0.15,var1=TOPENB50,d1=disc2press);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 0;

else if rank = 1 then rank = 2;

else if rank = 2 then rank = 2;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2press;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2topen;

set disc2press;

if rank = 0 then ordeqtopenb50 = 1;

if rank = 2 then ordeqtopenb50 = 2;

else if rank = 3 then ordeqtopenb50 = 3;

else if rank = 4 then ordeqtopenb50 = 4;

else if rank = 5 then ordeqtopenb50 = 5;

else if rank = 6 then ordeqtopenb50 = 6;

else if rank = 7 then ordeqtopenb50 = 7;

else if rank = 8 then ordeqtopenb50 = 8;

else if rank = 9 then ordeqtopenb50 = 9;

else rank = rank;

odseqtopenb50 = (avg\_dep)/(1-avg\_dep);

lodseqtopenb50 = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2topen;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2topen (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2topen;

Run;

Proc freq data=disc2topen;

table ordeqtopenb50 ODSEQtopenb50 LODSEQtopenb50;

Run;

Proc Means data = disc2topen;

class ordeqtopenb50;

var topenb50;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*INQ12;

%disc2(pval\_col = 0.15,var1=inq12,d1=disc2topen);

Proc Print data=temp;

Run;

Proc sort data=temp;

by rank;

Run;

symbol1 v= square color = black i=join;

Proc gplot data = temp;

plot avg\_dep\*rank;

run;

quit;

\*This code get rid of ranks for that specific variable;

\*For Age, we got rid of rank 1 and rank 8;

Data ranked1;

Set ranked;

if rank = 0 then rank = 1;

else if rank = 1 then rank = 1;

else if rank = 2 then rank = 3;

else if rank = 3 then rank = 3;

else if rank = 4 then rank = 4;

else if rank = 5 then rank = 5;

else if rank = 6 then rank = 6;

else if rank = 7 then rank = 7;

else if rank = 8 then rank = 8;

else if rank = 9 then rank = 9;

else rank = rank;

Run;

Proc sort data=ranked1;

by rank;

Run;

Proc sort data=temp;

by rank;

Run;

Proc freq data=ranked1;

tables rank;

run;

Data disc2topen;

Merge ranked1 temp;

by rank;

Run;

\*\*\*\*\*At this point, you can assign the ranks as 1, 2, 3...;

\*as shown in last table;

Data disc2lastv;

set disc2topen;

if rank = 1 then ordeqinq12 = 1;

else if rank = 3 then ordeqinq12 = 2;

else if rank = 4 then ordeqinq12 = 3;

else if rank = 5 then ordeqinq12 = 4;

else if rank = 6 then ordeqinq12 = 5;

else if rank = 7 then ordeqinq12 = 6;

else if rank = 8 then ordeqinq12 = 7;

else if rank = 9 then ordeqinq12 = 8;

else rank = rank;

odseqinq12 = (avg\_dep)/(1-avg\_dep);

lodseqinq12 = log ((avg\_dep)/(1-avg\_dep));

run;

proc freq data=disc2lastv;

tables avg\_dep;

\*Note: at this point, there is ALOT of garbage in the file that needs to be removed...;

Data disc2lastv (drop = \_TYPE\_ avg\_dep avg\_indp max\_dep max\_indp

min\_dep min\_indp numobs pvalue rank std\_dep std\_indp v1);

set disc2lastv;

Run;

Proc freq data=disc2lastv;

table ordeqinq12 ODSEQinq12 LODSEQinq12;

Run;

Proc Means data = disc2lastv;

class ordeqinq12;

var inq12;

\*\* New data set: 158 variables;

Data class.modfile2;

set disc2lastv;

proc means data=class.modfile2 N nmiss;

\*\*\*\*\*\*sampling data : train/score data;

\*proc contents data=class.disc1final;

proc sort data=class.modfile2 out= test1;

by goodbad;

proc surveyselect data=test1 samprate= 0.67 out= moddevfile

seed = 43 outall;

strata goodbad;

proc freq data=moddevfile;

tables goodbad\*selected;

proc contents data=moddevfile;

data train valid;

set moddevfile;

if selected then output train;

else output valid;

proc surveyselect data=test1 outsample

method=srs sampsize=(2000,1000)

seed = 43

strata = goodbad;

proc freq data= oversample;

\*creating training and valid data in out dataset;

\*\*\* 841138 variables;

data class.train;

set train;

\*\*\* 414291 variables;

data class.valid;

set valid;

proc contents data = class.train;

\* log regression deletes some, then one with the highest p values/lowest chi squared get rid of

\* do not do deliq id or goodbad/crelim or matchkey;

\*proc logistics;

Proc Logistic data = class.train desc outest = betas outmodel=scoringdata; \*NO delqid, matchkey, etc. or goodbad;

model GOODBAD = BRNEW BRWCRATE DCWCRATE FFCR49 FININQS INQ12 LAAGE MOSOPEN ORDBNKINQ2 ORDBNKINQS ORDBRADB ORDBRBAL50

ORDBRNEW ORDBRWCRATE ORDDCWCRATE ORDFININQS ORDINQ12 ORDMOSOPEN ORDOT12PTOT ORDPRMINQ2 ORDPRMINQS ORDROPEN ORDTADB

ORDTADB25 ORDTBAL ORDTOPENB50 ORDTPCTSAT ORDTRCR49 ORDTROPENEX ORRATE3 OT12PTOT PRMINQ2 PRMINQS ROPEN TADB TADB25

TBAL TOPENB50 TPCTSAT TRCR49 TROPENEX loddsORDBNKINQ2 loddsORDBNKINQS loddsORDBRADB

loddsORDBRBAL50 loddsORDBRNEW loddsORDBRWCRATE loddsORDDCWCRATE loddsORDFININQS loddsORDINQ12 loddsORDMOSOPEN loddsORDOT12PTOT

loddsORDPRMINQ2 loddsORDPRMINQS loddsORDROPEN loddsORDTADB loddsORDTADB25 loddsORDTBAL loddsORDTOPENB50 loddsORDTPCTSAT

loddsORDTRCR49 loddsORDTROPENEX lodseqbnkinq2 lodseqbnkinqs lodseqbradb lodseqbrbal50 lodseqbrnew lodseqbrwcrate lodseqdcwcrate

lodseqfininqs lodseqinq12 lodseqmosopen lodseqot12ptot lodseqprminq2 lodseqprminqs lodseqropen lodseqtadb lodseqtadb25

lodseqtbal lodseqtopenb50 lodseqtpctsat lodseqtrcr49 lodseqtropenex oddsORDBNKINQ2 oddsORDBNKINQS oddsORDBRADB oddsORDBRBAL50

oddsORDBRNEW oddsORDBRWCRATE oddsORDDCWCRATE oddsORDFININQS oddsORDINQ12 oddsORDMOSOPEN oddsORDOT12PTOT oddsORDPRMINQ2

oddsORDPRMINQS oddsORDROPEN oddsORDTADB oddsORDTADB25 oddsORDTBAL oddsORDTOPENB50 oddsORDTPCTSAT oddsORDTRCR49 oddsORDTROPENEX

odseqbnkinq2 odseqbnkinqs odseqbradb odseqbrbal50 odseqbrnew odseqbrwcrate odseqdcwcrate odseqfininqs odseqinq12 odseqmosopen

odseqot12ptot odseqprminq2 odseqprminqs odseqropen odseqtadb odseqtadb25 odseqtbal odseqtopenb50 odseqtpctsat odseqtrcr49

odseqtropenex ordeqbnkinq2 ordeqbnkinqs ordeqbradb ordeqbrbal50 ordeqbrnew ordeqbrwcrate ordeqdcwcrate ordeqfininqs

ordeqinq12 ordeqmosopen ordeqot12ptot ordeqprminq2 ordeqprminqs ordeqropen ordeqtadb ordeqtadb25 ordeqtbal ordeqtopenb50

ordeqtpctsat ordeqtrcr49 ordeqtropenex

/selection = backward

CTABLE pprob=(0.1 to 0.3 by .02)

LACKFIT RISKLIMITS;

output out = output p = predicted;

score data=class.valid out=class.score;

proc means data=class.train n nmiss mean;

tables oddsORDBRBAL50;

\*(look at sensitivity and specificity and find where they're at their peak by editing pprob),

c stat should be AT LEAST 85%, look at maximum likelihood estimate get rid of next lowest wald chisquare, if

c stat drops below .8 STOP dropping variables;

\*steps in results will show you which ones to get rid of;

\*\*\*\* logistic regression;