/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Code Name : Representativeness analysis.sas

Author : Kinsuk Ghatak

Last Modified on : 25th May , 2021

Purpose : The code contains steps pertaining to all the relevant data processing and analysis on representativeness understanding between GLCS application

development , LRDR and calibration portfolios. It calculates and generates the summary statistics [mean , median , p1, p90, p99, skewness , cusrtosis],

box plots, box plots by industry and so on.

Apart from that this also helps in doing several Exploratory analysis: like

1. CRR Migration anaysis

2. Turnover bucket wise analysis for LRDR data set

3. CRR wise distribution

4. Geography wise distribution

Dependency : Final data sets of live data , development data and calibration data

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*Set up libraries \*/

libname glcs spde "/sas/data/rmad/analytics/data\_output/model\_development/development/md005\_glcs\_pd\_v32/working"

datapath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md005\_glcs\_pd\_v32/working/data")

indexpath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md005\_glcs\_pd\_v32/working/index" )

hdfshost=default

outencoding=asciiany

;

libname MEP\_REU SPDE "/sas/data/rmad/analytics/data\_output/model\_development/development/md031\_test"

datapath=("/sas/data/rmad/analytics/data\_output/model\_development/development/md031\_test")

indexpath=("/sas/data/rmad/analytics/data\_output/model\_development/development/md031\_test")

hdfshost=default PARTSIZE = 1048576;

libname mep\_glcs spde "/sas/data/rmad/analytics/data\_output/model\_development/development/md033\_glcs\_irb\_repair"

datapath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md033\_glcs\_irb\_repair/data")

indexpath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md033\_glcs\_irb\_repair/index")

hdfshost=default partsize=1048576;

;

Libname RMADDREF meta library = 'RMAD\_D\_REFERENCE\_DATA' metaout=data;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 1 : DATA EXTRACTION , MAPPING , PROCESSING

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*Read the country region mapping table\*/

proc sql;

create table country\_region\_mapping as

select distinct country\_code\_2\_alpha as COUNTRY\_CODE,

country\_description as COUNTRY,

HSBC\_REGION

from RMADDREF.RMAD\_D\_REF\_FULL\_COUNTRY\_CODE\_MAP; quit;

PROC FORMAT;

VALUE CRR\_RANK\_EQV

0.1 = 1

1.1 = 2

1.2 = 3

2.1 = 4

2.2 = 5

3.1 = 6

3.2 = 7

3.3 = 8

4.1 = 9

4.2 = 10

4.3 = 11

5.1 = 12

5.2 = 13

5.3 = 14

6.1 = 15

6.2 = 16

7.1 = 17

7.2 = 18

8.1 = 19

8.2 = 20

8.3 = 21

9.0 = 22

10.0 = 23

;

QUIT;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EXTRACTION OF GLCS DEV DATA AND MERGING CARM\_INSTANCE INFORMATION TO IT

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

data glcs\_dev;

set glcs.glcs\_dev\_scored;

new\_ra5\_id\_new2=input(new\_ra5\_id,8.);

rename crrrank=crr\_32\_rk;

rename industry\_sector=industry;

run;

data glcs\_dev;

set glcs\_dev;

length industry2 $200.;

if industry="Basic Materials" then industry2="Basic\_Mat";

else if industry="Consumer, Cyclical" then industry2="Consum\_Cycl";

else if industry="Consumer, Non-cyclical" then industry2="Consum\_NoCycl";

else industry2=industry;

drop industry;

rename industry2 =industry;

run;

proc sql; create table glcs\_dev as select a.\*,b.HSBC\_REGION from glcs\_dev a left join country\_region\_mapping b on a.country=b.country\_code;quit;

/\* Saving GLCS Dev data with region information to permanent library\*/

/\*data mep\_glcs.glcs\_dev;\*/

/\*set glcs\_dev;\*/

/\*run;\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EXTRACTION OF GLCS CALIBRATION DATA AND MERGE CRR RANKS :

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*data glcs.glcs\_calib\_sample;\*/

/\*set work.callibration\_sample\_sas;\*/

/\*run;\*/

data glcs\_calib;

set glcs.glcs\_calib\_sample(keep= Score Final\_PD current\_final\_crr calibrated\_pd model\_crr\_pre\_calibration model\_crr\_post\_calibration);

run;

DATA glcs\_calib;

SET glcs\_calib;

mdl\_crr\_pre\_calib\_rk= PUT(model\_crr\_pre\_calibration,CRR\_RANK\_EQV.);

mdl\_crr\_post\_calib\_rk =PUT(model\_crr\_post\_calibration,CRR\_RANK\_EQV.);

mdl\_crr\_precalib\_rk=input(mdl\_crr\_pre\_calib\_rk,8.);

mdl\_crr\_postcalib\_rk=input(mdl\_crr\_post\_calib\_rk,8.);

drop mdl\_crr\_pre\_calib\_rk mdl\_crr\_post\_calib\_rk;

RUN;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*\*\*\*\*Extract live data set for GLCS Portfolio and merge HSBC infor and then split it into LRDR , application data sets \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

proc sql; create table mep\_glcs.glcs\_crr32\_dec\_irb\_20may21 as select a.\*,b.HSBC\_REGION from mep\_glcs.glcs\_crr32\_dec\_irb\_20may21 a left join country\_region\_mapping b on a.COUNTRY\_OF\_INCORPORATION\_CODE=b.country\_code;quit;

proc freq data=mep\_glcs.glcs\_crr32\_dec\_irb\_24apr21; tables hsbc\_region;run;

proc sql; create table mep\_glcs.nonglcs\_crr32\_dec\_irb\_20may21 as select a.\*,b.HSBC\_REGION from mep\_glcs.nonglcs\_crr32\_dec\_irb\_20may21 a left join country\_region\_mapping b on a.COUNTRY\_OF\_INCORPORATION\_CODE=b.country\_code;quit;

proc freq data=mep\_glcs.nonglcs\_crr32\_dec\_irb\_20may21; tables hsbc\_region;run;

%let vars\_to\_keep =

ra\_customerid

customerid

reln\_id\_orig

GRA\_GCDU\_Global\_id

yr\_orig

cust\_id\_orig

final\_flag3

intodef\_orig

gcdu\_global\_id\_orig

yr

approval\_date\_orig

customername

carm\_instance\_orig

country\_use

industry

COI

COUNTRY\_OF\_INCORPORATION\_CODE

hsbc\_region

Total\_Assets

net\_income

Tot\_liablities\_o\_Tot\_Assets

ebit\_margin

Total\_Revenue

ffo\_o\_net\_debt

EBITDA\_To\_Tot\_Interest

Cashflow\_o\_Interest\_Expense

Ln\_Total\_Assets\_df

Net\_Income\_df\_rk

Tot\_liab\_o\_Tot\_Assets\_rk

EBIT\_Margin\_rk

FFO\_o\_Net\_Debt\_rk

Total\_Revenue\_rk

EBITDA\_To\_Tot\_Interest\_rk

Cashflow\_o\_Interest\_Expense\_Rk

PRODU473s

DOMIN656s

CHANG484\_imps

CYCLI202s

YEARS558s

MAJOR669\_imps

Security\_risk

commitment\_to\_pay

Institutional\_Effectiveness

Interest\_Charges\_ratio

Current\_Acc\_o\_gdp

finalscore

finscore

coresubj

coresubjmerton

coremerton

lnpd

pd

model\_crr\_orig

model\_crr\_par

crr\_32

final\_crr\_orig

final\_crr\_par

crrrank

crr\_32\_rk

app\_year

;

data glcs\_all (keep= &vars\_to\_keep.);

set mep\_glcs.glcs\_crr32\_dec\_irb\_20may21(rename=(Sovereigns\_Commitment\_to\_Pay=commitment\_to\_pay Current\_Account\_o\_GDP=Current\_Acc\_o\_gdp ));

if EAD\_LIMIT\_FLAG = "ACTIVE";

app\_year=YEAR(approval\_date\_orig);

crr\_32\_rank=PUT(crr\_32,CRR\_RANK\_EQV.);

crr\_32\_rk=input(crr\_32\_rank,8.);

drop crr\_32\_rank;

net\_income=input(net\_inc,8.);

drop net\_inc ;

run;

/\*60,280\*/

data non\_glcs\_all (keep= &vars\_to\_keep.);

set mep\_glcs.nonglcs\_crr32\_dec\_irb\_20may21(rename=(Sovereigns\_Commitment\_to\_Pay=commitment\_to\_pay Current\_Account\_o\_GDP=Current\_Acc\_o\_gdp ));

if EAD\_LIMIT\_FLAG = "ACTIVE";

app\_year=YEAR(approval\_date\_orig);

crr\_32\_rank=PUT(crr\_32,CRR\_RANK\_EQV.);

crr\_32\_rk=input(crr\_32\_rank,8.);

drop crr\_32\_rank;

net\_income=input(net\_inc,8.);

drop net\_inc ;

run;

/\*41,977\*/

data glcs\_total;

set glcs\_all non\_glcs\_all;

run;

/\*\* EXTRACT ONLY PARNETS FROM THE APPENDED FINAL DATA SET : \*\*/

data final\_glcs\_appended\_parent;

set glcs\_total;

if Final\_Flag3="Parent";

run;

/\*42,401\*/

/\*\* Keep only latest records per aproval date for each customer \*\*\*/

proc sql; create table glcs\_full\_parent as select \* from final\_glcs\_appended\_parent

group by reln\_id\_orig,customerid,approval\_date\_orig

having app\_year=max(app\_year);

quit;

/\*42,401\*/

%let vars\_to\_keep =

ra\_customerid

customerid

reln\_id\_orig

cust\_id\_orig

hsbc\_region

yr

approval\_date\_orig

customername

carm\_instance\_orig

country\_use

industry

COI

Total\_Assets

net\_income

Tot\_liablities\_o\_Tot\_Assets

ebit\_margin

Total\_Revenue

ffo\_o\_net\_debt

EBITDA\_To\_Tot\_Interest

Cashflow\_o\_Interest\_Expense

Ln\_Total\_Assets\_df

Net\_Income\_df\_rk

Tot\_liab\_o\_Tot\_Assets\_rk

EBIT\_Margin\_rk

FFO\_o\_Net\_Debt\_rk

Total\_Revenue\_rk

EBITDA\_To\_Tot\_Interest\_rk

Cashflow\_o\_Interest\_Expense\_Rk

PRODU473s

DOMIN656s

CHANG484\_imps

CYCLI202s

YEARS558s

MAJOR669\_imps

Security\_risk

commitment\_to\_pay

Institutional\_Effectiveness

Interest\_Charges\_ratio

Current\_Acc\_o\_gdp

finalscore

finscore

coresubj

coresubjmerton

coremerton

lnpd

pd

model\_crr\_orig

model\_crr\_par

crr\_32

final\_crr\_orig

final\_crr\_par

crrrank

crr\_32\_rk

app\_year

;

data glcs\_full\_portfolio\_2;

set glcs\_full\_parent;

length industry2 $200.;

if industry="Basic Materials" then industry2="Basic\_Mat";

else if industry="Consumer, Cyclical" then industry2="Consum\_Cycl";

else if industry="Consumer, Non-Cyclical" then industry2="Consum\_NoCycl";

else industry2=industry;

drop industry;

rename industry2 =industry;

run;

data glcs\_app;

set glcs\_full\_portfolio\_2;

if app\_year in (2019,2020);

run;

/\*7,332\*/

data glcs\_lrdr;

set glcs\_full\_portfolio\_2;

if app\_year not in (2020);

run;

/\*38,867\*/

/\*distribution analysis for various hsbc region accross different data sets \*/

proc freq data=glcs\_app;tables hsbc\_region;run;

proc freq data=glcs\_lrdr;tables hsbc\_region;run;

proc freq data=glcs\_dev;tables hsbc\_region;run;

/\*Saving the full , application and LRDR datasets into permanent libraries\*/

data mep\_glcs.glcs\_full\_portfolio;

set glcs\_full\_portfolio\_2;

run;

data mep\_glcs.glcs\_app;

set glcs\_app;

run;

data mep\_glcs.glcs\_lrdr;

set glcs\_lrdr;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

data glcs\_lrdr;

set mep\_glcs.glcs\_lrdr;

run;

data glcs\_app;

set mep\_glcs.glcs\_app;

run;

data glcs\_full\_portfolio\_2;

data mep\_glcs.glcs\_full\_portfolio;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 2 : MACRO AND ANALYSIS TO GENERATE VARIABLE DISTRIBUTION TABLES AND BOX PLOTS AT DIFFERENT LEVELS :

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Macros to generate Variable distribution carm\_site wise and also at total level \*/

%let data\_sets=

glcs\_app

glcs\_lrdr

glcs\_dev

;

%let var\_list\_total =

Total\_Assets

Net\_Income

Tot\_liablities\_o\_Tot\_Assets

ebit\_margin

Total\_Revenue

ffo\_o\_net\_debt

EBITDA\_To\_Tot\_Interest

Cashflow\_o\_Interest\_Expense

Ln\_Total\_Assets\_df

Net\_Income\_df\_rk

Tot\_liab\_o\_Tot\_Assets\_rk

EBIT\_Margin\_rk

FFO\_o\_Net\_Debt\_rk

Total\_Revenue\_rk

EBITDA\_To\_Tot\_Interest\_rk

Cashflow\_o\_Interest\_Expense\_Rk

PRODU473s

DOMIN656s

CHANG484\_imps

CYCLI202s

YEARS558s

MAJOR669\_imps

Security\_risk

commitment\_to\_pay

Institutional\_Effectiveness

Interest\_Charges\_ratio

Current\_Acc\_o\_gdp

finalscore

finscore

coresubj

coresubjmerton

coremerton

lnpd

pd

crr\_32\_rk

;

%let ind\_list=

Basic\_Mat

Communications

Consum\_Cycl

Consum\_NoCycl

Diversified

Energy

Industrial

Technology

Utilities

;

%macro var\_dist\_loop(var\_list\_name);

data var\_dist\_all;

set \_NULL\_;

run;

%do k=1 %to %sysfunc(countw(&data\_sets.));

%let data\_set\_var\_dist=%scan(&data\_sets.,&k.);

proc means data=&data\_set\_var\_dist. stackodsputput N Nmiss min max std median mean p1 p5 p10 p25 p50 p75 p90 p95 p99 skweness kurtosis;

vars &var\_list\_name.;

ods output summary=Total\_&data\_set\_var\_dist.;

run;

data Total\_&data\_set\_var\_dist.;

set Total\_&data\_set\_var\_dist.;

length date\_set\_name $200.;

date\_set\_name="&data\_set\_var\_dist.";

run;

data dist\_&data\_set\_var\_dist.;

set \_NULL\_;

run;

%do i=1 %to %sysfunc(countw(&ind\_list.));

%let industry=%scan(&ind\_list.,&i.);

data &data\_set\_var\_dist.\_&industry.;

set &data\_set\_var\_dist.;

if industry="&industry.";

run;

proc means data=&data\_set\_var\_dist.\_&industry. stackodsputput N Nmiss min max std median mean p1 p5 p10 p25 p50 p75 p90 p95 p99 skweness kurtosis;

vars &var\_list\_name.;

ods output summary=Tot\_&data\_set\_var\_dist.\_&industry.;

run;

data Tot\_&data\_set\_var\_dist.\_&industry.;

set Tot\_&data\_set\_var\_dist.\_&industry.;

length date\_set\_name $200.;

length industry\_name $200.;

date\_set\_name="&data\_set\_var\_dist.";

industry\_name="&industry.";

data dist\_&data\_set\_var\_dist.;

set dist\_&data\_set\_var\_dist. Tot\_&data\_set\_var\_dist.\_&industry.;

run;

run;

%end;

data var\_dist\_all;

set var\_dist\_all Total\_&data\_set\_var\_dist.;

run;

%end;

%mend;

%var\_dist\_loop(&var\_list\_total.);

data var\_dist\_all;

set var\_dist\_all;

variable\_name=lowcase(Variable);

drop Variable ;

rename variable\_name=Variable;

run;

data var\_dist\_all\_tagged;

length var\_flag $200.;

set var\_dist\_all;

if Variable in ('total\_assets',

'net\_income',

'tot\_liablities\_o\_tot\_assets',

'ebit\_margin',

'total\_revenue',

'ffo\_o\_net\_debt',

'ebitda\_to\_tot\_interest',

'cashflow\_o\_interest\_expense') then var\_flag="Raw Financials";

else if Variable in ( 'ln\_total\_assets\_df',

'net\_income\_df\_rk',

'tot\_liab\_o\_tot\_assets\_rk',

'ebit\_margin\_rk',

'ffo\_o\_net\_debt\_rk',

'total\_revenue\_rk',

'ebitda\_to\_tot\_interest\_rk',

'cashflow\_o\_interest\_expense\_rk') then var\_flag="Transformed Financials";

else if Variable in ('produ473s',

'domin656s' ,

'chang484\_imps' ,

'cycli202s' ,

'years558s' ,

'major669\_imps') then var\_flag="Qualitatives";

else if Variable in ( 'security\_risk',

'commitment\_to\_pay',

'institutional\_effectiveness',

'interest\_charges\_ratio',

'current\_acc\_o\_gdp')then var\_flag='Macroeconomics';

else if Variable in ( 'finalscore',

'finscore' ,

'coresubj',

'coresubjmerton',

'coremerton' ,

'lnpd',

'pd',

'crr\_32\_rk') then var\_flag='Scores';

run;

proc sort data=var\_dist\_all\_tagged out=var\_dist\_all; by var\_flag variable ;run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Representative analysis on the application , callibration data set on the basis of scores :

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

data glcs\_calib;

set glcs\_calib;

rename mdl\_crr\_postcalib\_rk=crr\_32\_rk Calibrated\_PD=pd ;

run;

%let data\_sets\_calib\_rep=

glcs\_app

glcs\_calib

;

%let calib\_distrib\_vars=

pd

crr\_32\_rk

;

%macro var\_dist\_calib\_loop(var\_list\_name);

data var\_dist\_all;

set \_NULL\_;

run;

%do k=1 %to %sysfunc(countw(&data\_sets\_calib\_rep.));

%let data\_set\_var\_dist=%scan(&data\_sets\_calib\_rep.,&k.);

proc means data=&data\_set\_var\_dist. stackodsputput N Nmiss min max std median mean p1 p5 p10 p25 p50 p75 p90 p95 p99 skweness kurtosis;

vars &var\_list\_name.;

ods output summary=Total\_&data\_set\_var\_dist.;

run;

data Total\_&data\_set\_var\_dist.;

set Total\_&data\_set\_var\_dist.;

length date\_set\_name $200.;

date\_set\_name="&data\_set\_var\_dist.";

run;

data var\_dist\_all;

set var\_dist\_all Total\_&data\_set\_var\_dist.;

run;

%end;

%mend;

%var\_dist\_calib\_loop(&calib\_distrib\_vars.);

/\*\* Generate Plots and graphs for the different variables \*\*/

data dev\_scores;

set glcs\_dev (keep= industry finalscore finscore coresubj coresubjmerton coremerton lnpd pd crr\_32\_rk);

data\_flag="Development data";

run;

data lrdr\_scores;

set glcs\_lrdr (keep= industry finalscore finscore coresubj coresubjmerton coremerton lnpd pd crr\_32\_rk);

data\_flag="LRDR data";

run;

data appl\_scores;

set glcs\_app (keep= industry finalscore finscore coresubj coresubjmerton coremerton lnpd pd crr\_32\_rk);

data\_flag="Application data";

run;

data dev\_app;

set dev\_scores appl\_scores;

run;

data app\_lrdr;

set appl\_scores lrdr\_scores;

run;

/\*plotting final and fin scores by dev-app and app-lrdr\*\*/

proc sgplot data=dev\_app;

vbox finalscore / group=data\_flag;

run;

proc sgplot data=app\_lrdr;

vbox finalscore / group=data\_flag;

run;

proc sgplot data=dev\_app;

vbox finscore/ group=data\_flag;

run;

proc sgplot data=app\_lrdr;

vbox finscore /group=data\_flag;

run;

/\*plotting final and fin scores by industry category for dev-app and app-lrdr\*\*/

proc sgplot data=dev\_app;

vbox finalscore /category=industry group=data\_flag;

run;

proc sgplot data=app\_lrdr;

vbox finalscore /category=industry group=data\_flag;

run;

proc sgplot data=dev\_app;

vbox finscore /category=industry group=data\_flag;

run;

proc sgplot data=app\_lrdr;

vbox finscore /category=industry group=data\_flag;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 4 : CRR WISE DISTRIBUTION ANALYSIS FOR APPLICATION , LRDR , CALIBRATION AND DEV DATA SETS :

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*analysing applicationvs calibration pd figures \*\*/

data appl\_pd;

set glcs\_app (keep=&calib\_distrib\_vars.);

data\_flag="Application Data";

run;

data calib\_pd;

set glcs\_calib (keep=&calib\_distrib\_vars.);

data\_flag="Calibration Data";

run;

data app\_calib;

set appl\_pd calib\_pd;

run;

proc sgplot data=app\_calib;

vbox pd / group=data\_flag;

run;

/\*\*generating CRR distribution accross different data sets \*/

proc freq data= glcs\_dev ;tables crr;run;

proc freq data= glcs\_lrdr ;tables crr\_32;run;

proc freq data= glcs\_app ;tables crr\_32;run;

proc freq data=glcs\_calib;tables Model\_CRR\_post\_calibration;run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 5 : TURNOVER CUTOFF WISE DISTRIBUTION ANALYSIS FOR LRDR DATA SET :

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\* LRDR data set turnover cutoff wise anlysis with respect to CRR \*\*/

proc freq data=glcs\_lrdr;

where Total\_Revenue>700;

tables crr\_32 ;run;

proc freq data=glcs\_lrdr;

where Total\_Revenue>650;

tables crr\_32 ;run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 6 : CRR MIGRATION ANALYSIS :

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%macro crr\_migration\_analysis(year,prev\_year);

data glcs\_live\_&year.;

set mep\_glcs.glcs\_full\_portfolio;

if app\_year =&year. and cust\_id\_orig ne ' ' and crr\_32 ne ' ' ;

run;

proc sql; create table glcs\_live\_filtd\_&year. as select \* from glcs\_live\_&year.

group by cust\_id\_orig having approval\_date\_orig=max(approval\_date\_orig);

quit;

data glcs\_live\_filtd\_&year.;

set glcs\_live\_filtd\_&year.;

if intodef\_orig=1 then glcs\_crr\_32="Default" ;

else glcs\_crr\_32=crr\_32 ;

drop crr\_32;

rename glcs\_crr\_32=crr\_32;

run;

data glcs\_live\_&prev\_year.;

set mep\_glcs.glcs\_full\_portfolio;

if app\_year =&prev\_year. and cust\_id\_orig ne ' ' and crr\_32 ne ' ' ;

run;

proc sql; create table glcs\_live\_filtd\_&prev\_year. as select \* from glcs\_live\_&prev\_year.

group by cust\_id\_orig having approval\_date\_orig=max(approval\_date\_orig);

quit;

data glcs\_live\_filtd\_def\_&prev\_year.;

set glcs\_live\_filtd\_&prev\_year.;

if intodef\_orig=1;

run;

proc freq data=glcs\_live\_filtd\_def\_&prev\_year. output ; tables crr\_32 \* intodef\_orig;run;

proc sql; create table crr\_mig\_&prev\_year.\_&year. as select a.cust\_id\_orig as cust\_id\_&prev\_year.,a.crr\_32 as model\_crr\_&prev\_year.,

b.cust\_id\_orig as cust\_id\_&year.,b.crr\_32 as model\_crr\_&year. from glcs\_live\_filtd\_&prev\_year. a

left join glcs\_live\_filtd\_&year. b

on a.reln\_id\_orig=b.reln\_id\_orig and a.cust\_id\_orig=b.cust\_id\_orig ;

quit;

data crr\_mig\_&prev\_year.\_&year.;

set crr\_mig\_&prev\_year.\_&year.;

if model\_crr\_&year. =' ' then model\_crr\_&year.= model\_crr\_&prev\_year.;

run;

/\*data crr\_mig\_&prev\_year.\_&year.;\*/

/\*set crr\_mig\_&prev\_year.\_&year.;\*/

/\*if model\_crr\_&year. =' ' then model\_crr\_new\_&year.= 0;\*/

/\*else model\_crr\_new\_&year.= model\_crr\_&year.;\*/

/\*drop model\_crr\_&year.;\*/

/\*rename model\_crr\_new\_&year.= model\_crr\_&year.;\*/

/\*run;\*/

proc freq data=crr\_mig\_&prev\_year.\_&year. ;

tables model\_crr\_&prev\_year. \* model\_crr\_&year. /nocum nocol nopercent norow ;

run;

%mend;

%crr\_migration\_analysis(2019,2018);

%crr\_migration\_analysis(2020,2019);

%crr\_migration\_analysis(2009,2008);

data glcs\_live\_filtd\_2018;

set mep\_glcs.glcs\_full\_portfolio;

if yr=2018 and intodef\_orig=1;

run;

proc freq data=glcs\_live\_filtd\_2018; tables crr\_32 \* intodef\_orig/ nocum nocol nopercent norow ;run;

data glcs\_live\_filtd\_2008;

set mep\_glcs.glcs\_full\_portfolio;

if yr=2008 and intodef\_orig=1;

run;

proc freq data=glcs\_live\_filtd\_2008; tables crr\_32 \* intodef\_orig/ nocum nocol nopercent norow ;run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Code Name : KS\_KW\_Computation.sas

Author : Kinsuk Ghatak

Last Modified on : 25th May , 2021

Purpose : The script helps in calculating KS and KW statistics for the different drivers as well as outcome variables like scores etc.

Dependency : Representative analysis

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Set up libraries \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

libname glcs spde "/sas/data/rmad/analytics/data\_output/model\_development/development/md005\_glcs\_pd\_v32/working"

datapath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md005\_glcs\_pd\_v32/working/data")

indexpath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md005\_glcs\_pd\_v32/working/index" )

hdfshost=default

outencoding=asciiany

;

\*MEP\_REU;

libname MEP\_REU SPDE "/sas/data/rmad/analytics/data\_output/model\_development/development/md031\_test"

datapath=("/sas/data/rmad/analytics/data\_output/model\_development/development/md031\_test")

indexpath=("/sas/data/rmad/analytics/data\_output/model\_development/development/md031\_test")

hdfshost=default PARTSIZE = 1048576;

libname mep\_glcs spde "/sas/data/rmad/analytics/data\_output/model\_development/development/md033\_glcs\_irb\_repair"

datapath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md033\_glcs\_irb\_repair/data")

indexpath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md033\_glcs\_irb\_repair/index")

hdfshost=default partsize=1048576;

;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 1 : DATA EXTRACTION AND PREPARATION FOR KS-KW COMPUTATION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%let KS\_var\_list =

Total\_Assets

Net\_Income

Tot\_liablities\_o\_Tot\_Assets

ebit\_margin

Total\_Revenue

ffo\_o\_net\_debt

EBITDA\_To\_Tot\_Interest

Cashflow\_o\_Interest\_Expense

Ln\_Total\_Assets\_df

Net\_Income\_df\_rk

Tot\_liab\_o\_Tot\_Assets\_rk

EBIT\_Margin\_rk

Total\_Revenue\_rk

FFO\_o\_Net\_Debt\_rk

EBITDA\_To\_Tot\_Interest\_rk

Cashflow\_o\_Interest\_Expense\_Rk

PRODU473s

DOMIN656s

CHANG484\_imps

CYCLI202s

YEARS558s

MAJOR669\_imps

Security\_risk

commitment\_to\_pay

Institutional\_Effectiveness

Interest\_Charges\_ratio

Current\_Acc\_o\_gdp

finalscore

finscore

coresubj

coresubjmerton

coremerton

lnpd

pd

crr\_32\_rk

;

data glcs\_dev;

set glcs.glcs\_dev\_scored;

new\_ra5\_id\_new2=input(new\_ra5\_id,8.);

rename crrrank=crr\_32\_rk;

rename industry\_sector=industry;

run;

data glcs\_dev;

set glcs\_dev;

length industry2 $200.;

if industry="Basic Materials" then industry2="Basic\_Mat";

else if industry="Consumer, Cyclical" then industry2="Consum\_Cycl";

else if industry="Consumer, Non-cyclical" then industry2="Consum\_NoCycl";

else industry2=industry;

drop industry;

rename industry2 =industry;

run;

data glcs\_dev\_2 ;

rename year=app\_year ;

set glcs\_dev(keep= &KS\_var\_list. industry year);

data\_name="Development Data";

run;

data glcs\_dev\_2;

length industry\_2 $300.;

set glcs\_dev\_2;

if industry="Basic Materials" then industry\_2="Basic\_Mat" ;

else if industry="Consumer, Cyclical" then industry\_2="Consum\_Cycl";

else if industry="Consumer, Non-Cyclical" then industry\_2="Consum\_NoCycl";

else industry\_2=industry;

drop industry;

rename industry\_2=industry;

run;

data glcs\_lrdr;

set mep\_glcs.glcs\_lrdr;

data\_name="LRDR Dataset";

run;

data glcs\_app;

set mep\_glcs.glcs\_app;

data\_name="Application Dataset";

run;

data glcs\_calib;

set glcs.glcs\_calib\_sample(keep= Score Final\_PD current\_final\_crr calibrated\_pd model\_crr\_pre\_calibration model\_crr\_post\_calibration);

run;

DATA glcs\_calib;

SET glcs\_calib;

mdl\_crr\_pre\_calib\_rk= PUT(model\_crr\_pre\_calibration,CRR\_RANK\_EQV.);

mdl\_crr\_post\_calib\_rk =PUT(model\_crr\_post\_calibration,CRR\_RANK\_EQV.);

mdl\_crr\_precalib\_rk=input(mdl\_crr\_pre\_calib\_rk,8.);

mdl\_crr\_postcalib\_rk=input(mdl\_crr\_post\_calib\_rk,8.);

drop mdl\_crr\_pre\_calib\_rk mdl\_crr\_post\_calib\_rk;

RUN;

data glcs\_calib;

set glcs\_calib;

rename mdl\_crr\_postcalib\_rk=crr\_32\_rk Calibrated\_PD=pd ;

data\_name="Calibration Data";

run;

%let calib\_distrib\_vars=

data\_name

pd

crr\_32\_rk

;

data appl\_pd;

set glcs\_app (keep=&calib\_distrib\_vars.);

data\_flag="Application Data";

run;

data calib\_pd;

set glcs\_calib (keep=&calib\_distrib\_vars.);

data\_flag="Calibration Data";

run;

data glcs\_calib\_vs\_app;

set appl\_pd calib\_pd;

run;

data glcs\_dev\_Vs\_app;

set glcs\_dev\_2 glcs\_app ;

run;

data glcs\_dev\_Vs\_lrdr;

set glcs\_dev\_2 glcs\_lrdr;

run;

data glcs\_app\_Vs\_lrdr;

set glcs\_app glcs\_lrdr;

run;

%let score\_variables=

finalscore

finscore

pd

;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 2 : DEFINING KS , KW OMPUTATION MACROS AND RUNNING THE MACROS WITH LOOP FOR DIFFERENT COMBINATIONS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%macro KS\_macro(dataset, var\_list);

data KS\_vars;

set \_NULL\_;

run;

%do n=1 %to %sysfunc(countw(&var\_list.));

%let variable=%scan(&var\_list.,&n.);

ods trace on;

ods output KS2Stats=KS\_Result;

ods output KruskalWallisTest=KW\_Result;

proc npar1way data=&dataset.;

class data\_name ;

var &variable.;

run;

data KS\_Result\_2;

set KS\_Result(keep=Variable Label2 cValue2 );

rename Label2=Statistic cValue2=Value;

metric="KS";

run;

data KW\_Result\_2;

set KW\_Result(keep=Variable Label1 cValue1);

rename Label1=Statistic cValue1=Value;

metric="KW";

run;

data KS\_KW;

set KS\_Result\_2 KW\_Result\_2;

run;

data KS\_KW;

set KS\_KW;

if Statistic='Pr > KSa' or Statistic='Pr > Chi' ;

run;

data KS\_vars;

set KS\_vars KS\_KW;

run;

data KS\_vars;

set KS\_Vars;

Variable\_small=lowcase(Variable);

drop Variable;

rename Variable\_small=Variable;

run;

data KS\_vars;

length var\_flag $300.;

set KS\_vars;

if Variable in ('total\_assets',

'net\_income',

'tot\_liablities\_o\_tot\_assets',

'ebit\_margin',

'total\_revenue',

'ffo\_o\_net\_debt',

'ebitda\_to\_tot\_interest',

'cashflow\_o\_interest\_expense') then var\_flag="Raw Financials";

else if Variable in ('ln\_total\_assets\_df',

'net\_income\_df\_rk',

'tot\_liab\_o\_tot\_assets\_rk',

'ebit\_margin\_rk',

'ffo\_o\_net\_debt\_rk',

'total\_revenue\_rk',

'ebitda\_to\_tot\_interest\_rk',

'Cashflow\_o\_interest\_expense\_rk') then var\_flag="Transformed Financials";

else if Variable in('produ473s',

'domin656s' ,

'chang484\_imps' ,

'cycli202s' ,

'years558s' ,

'major669\_imps') then var\_flag="Qualitatves";

else if Variable in ( 'security\_risk',

'commitment\_to\_pay',

'institutional\_effectiveness',

'interest\_charges\_ratio',

'current\_acc\_o\_gdp')then var\_flag='Macroeconomics';

else if Variable in ( 'finalscore',

'finscore' ,

'coresubj',

'coresubjmerton',

'coremerton' ,

'lnpd',

'pd',

'crr\_32\_rk') then var\_flag='Scores';

run;

%end;

%mend;

/\*\*Dev vs application : \*\*/

%KS\_macro(glcs\_dev\_Vs\_app, &score\_variables.);

/\*\*Dev vs LRDR : \*\*/

%KS\_macro(glcs\_dev\_Vs\_lrdr, &KS\_var\_list.);

/\*\*LRDR vs application : \*\*/

%KS\_macro(glcs\_app\_Vs\_lrdr, &score\_variables.);

%let ks\_data\_sets=

glcs\_dev\_Vs\_app

glcs\_app\_Vs\_lrdr

glcs\_calib\_vs\_app

;

options mlogic mprint symbolgen;

%macro KS\_Compute\_Loop ();

data KS\_All;

set \_NULL\_;

run;

%do i=1 %to %sysfunc(countw(&ks\_data\_sets.));

%let dataset\_ks=%scan(&ks\_data\_sets.,&i.);

%KS\_macro(&dataset\_ks., &score\_variables.);

data KS\_vars\_&dataset\_ks.;

set KS\_vars;

length dataset\_name $200.;

dataset\_name="&dataset\_ks.";

run;

data KS\_All;

set KS\_All KS\_vars\_&dataset\_ks.;

run;

%end;

%mend;

%KS\_Compute\_Loop ();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Code Name : PSI\_Calculation.sas

Author : Kinsuk Ghatak

Last Modified on : 25th May , 2021

Purpose : The script helps in calculating Population Stability Index of different outcome variables accross the several datasets

Dependency : Representative analysis

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Set up libraries \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

libname glcs spde "/sas/data/rmad/analytics/data\_output/model\_development/development/md005\_glcs\_pd\_v32/working"

datapath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md005\_glcs\_pd\_v32/working/data")

indexpath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md005\_glcs\_pd\_v32/working/index" )

hdfshost=default

outencoding=asciiany

;

\*MEP\_REU;

libname MEP\_REU SPDE "/sas/data/rmad/analytics/data\_output/model\_development/development/md031\_test"

datapath=("/sas/data/rmad/analytics/data\_output/model\_development/development/md031\_test")

indexpath=("/sas/data/rmad/analytics/data\_output/model\_development/development/md031\_test")

hdfshost=default PARTSIZE = 1048576;

libname mep\_glcs spde "/sas/data/rmad/analytics/data\_output/model\_development/development/md033\_glcs\_irb\_repair"

datapath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md033\_glcs\_irb\_repair/data")

indexpath= ("/sas/data/rmad/analytics/data\_output/model\_development/development/md033\_glcs\_irb\_repair/index")

hdfshost=default partsize=1048576;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 1 : DATA EXTRACTION AND PREPARATION FOR PSI COMPUTATION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROC FORMAT;

VALUE CRR\_RANK\_EQV

0.1 = 1

1.1 = 2

1.2 = 3

2.1 = 4

2.2 = 5

3.1 = 6

3.2 = 7

3.3 = 8

4.1 = 9

4.2 = 10

4.3 = 11

5.1 = 12

5.2 = 13

5.3 = 14

6.1 = 15

6.2 = 16

7.1 = 17

7.2 = 18

8.1 = 19

8.2 = 20

8.3 = 21

9.0 = 22

10.0 = 23

;

QUIT;

data glcs\_dev;

set mep\_glcs.glcs\_dev;

run;

data glcs\_dev;

set glcs\_dev;

length industry2 $200.;

if industry="Basic Materials" then industry2="Basic\_Mat";

else if industry="Consumer, Cyclical" then industry2="Consum\_Cycl";

else if industry="Consumer, Non-cyclical" then industry2="Consum\_NoCycl";

else industry2=industry;

drop industry;

rename industry2 =industry;

run;

data glcs\_dev\_2 ;

rename year=app\_year ;

set glcs\_dev;

data\_name="Development Data";

run;

data glcs\_dev\_2;

length industry\_2 $300.;

set glcs\_dev\_2;

if industry="Basic Materials" then industry\_2="Basic\_Mat" ;

else if industry="Consumer, Cyclical" then industry\_2="Consum\_Cycl";

else if industry="Consumer, Non-Cyclical" then industry\_2="Consum\_NoCycl";

else industry\_2=industry;

drop industry;

rename industry\_2=industry;

run;

data glcs\_dev\_2 ;

set glcs\_dev\_2;

if crr\_32 <=3.3 then Model\_CRR\_Band="1.1 to 3.3";

else if crr\_32 <=6.2 then Model\_CRR\_Band ="4.1 to 6.2";

else if crr\_32 >=7.1 then Model\_CRR\_Band ="7.1 above";

run;

data glcs\_lrdr;

set mep\_glcs.glcs\_lrdr;

data\_name="LRDR Dataset";

if crr\_32 <=3.3 then Model\_CRR\_Band="1.1 to 3.3";

else if crr\_32 <=6.2 then Model\_CRR\_Band ="4.1 to 6.2";

else if crr\_32 >=7.1 then Model\_CRR\_Band ="7.1 above";

run;

data glcs\_app;

set mep\_glcs.glcs\_app;

data\_name="Application Dataset";

if crr\_32 <=3.3 then Model\_CRR\_Band="1.1 to 3.3";

else if crr\_32 <=6.2 then Model\_CRR\_Band ="4.1 to 6.2";

else if crr\_32 >=7.1 then Model\_CRR\_Band ="7.1 above";

run;

data glcs\_calib;

set glcs.glcs\_calib\_sample(keep= Score Final\_PD current\_final\_crr calibrated\_pd model\_crr\_pre\_calibration model\_crr\_post\_calibration);

run;

DATA glcs\_calib;

SET glcs\_calib;

mdl\_crr\_pre\_calib\_rk= PUT(model\_crr\_pre\_calibration,CRR\_RANK\_EQV.);

mdl\_crr\_post\_calib\_rk =PUT(model\_crr\_post\_calibration,CRR\_RANK\_EQV.);

mdl\_crr\_precalib\_rk=input(mdl\_crr\_pre\_calib\_rk,8.);

mdl\_crr\_postcalib\_rk=input(mdl\_crr\_post\_calib\_rk,8.);

drop mdl\_crr\_pre\_calib\_rk mdl\_crr\_post\_calib\_rk;

RUN;

data glcs\_calib;

set glcs\_calib;

rename mdl\_crr\_postcalib\_rk=crr\_32\_rk Calibrated\_PD=pd ;

data\_name="Calibration Data";

run;

%let calib\_distrib\_vars=

data\_name

pd

crr\_32\_rk

;

%let KS\_var\_list =

Total\_Assets

Net\_Income\_df\_rk

Tot\_liablities\_o\_Tot\_Assets

ebit\_margin

Total\_Revenue

ffo\_o\_net\_debt

EBITDA\_To\_Tot\_Interest

Cashflow\_o\_Interest\_Expense

Ln\_Total\_Assets\_df

Net\_Income\_df\_rk

Tot\_liab\_o\_Tot\_Assets\_rk

EBIT\_Margin\_rk

FFO\_o\_Net\_Debt\_rk

Total\_Revenue\_rk

EBITDA\_To\_Tot\_Interest\_rk

Cashflow\_o\_Int\_Exp\_rk

PRODU473s

DOMIN656s

CHANG484\_imps

CYCLI202s

YEARS558s

MAJOR669\_imps

Security\_risk

commitment\_to\_pay

Institutional\_Effectiveness

Interest\_Charges\_ratio

Current\_Acc\_o\_gdp

;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 2: PSI Computation loops :

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*PSI FOR CONTINUOUS VARIABLES;

options mlogic mprint symbolgen;

%macro PSI(traindata, testdata, variable, group);

proc sql;

select count(\*) into: total\_train from &traindata; quit;

proc sql;

select count(\*) into: total\_test from &testdata; quit;

proc rank data=&traindata. out= ranktable groups=&group;

var &variable.;

ranks &variable.\_rank;

run;

proc sql;

create table train\_table as

select &variable.\_rank as decile , count(\*) as train\_obs,

max(&variable.) as maxval

from ranktable group by &variable.\_rank; quit;

data train\_table;

set train\_table;

minval = lag(maxval);

if decile = 0 then minval = -10\*\*100;

if decile = &group-1 then maxval = 10\*\*100;

run;

proc sql;

create table test\_ext as

select A.\*, B.Decile from &testdata as A full join train\_table as B

on A.&variable. > minval and A.&variable. <= maxval;

quit;

proc sql;

create table test\_table as

select decile, count(\*) as test\_obs

from test\_ext group by decile; quit;

proc sql;

create table combined as

select train\_table.\*, test\_table.test\_obs, train\_obs/&total\_train as DevPercnt, test\_obs/&total\_test as HOPercnt

from train\_table full outer join test\_table on train\_table.decile = test\_table.decile;

run;

proc sql;

create table PSI\_&variable as

select \*, (DevPercnt - HOPercnt) as dev\_ho, log(DevPercnt / HOPercnt) as Log\_Dev\_HO, (DevPercnt - HOPercnt) \*log(DevPercnt / HOPercnt) as PSI

from combined;

run;

/\*DATA PSI\_&variable;\*/

/\*SET PSI\_&variable;\*/

/\*if \_n\_ = 3 then PSI = 0.3;\*/

/\*run;\*/

DATA PSI\_&variable;

set PSI\_&variable;

if decile ne '.';

run;

PROC SQL;

CREATE TABLE PSI\_SUM AS

SELECT SUM(PSI) AS PSI FROM PSI\_&variable;

QUIT;

DATA PSI\_&variable;

SET PSI\_&variable PSI\_SUM;

RUN;

DATA PSI\_&variable;

SET PSI\_&variable;

IF (\_N\_ = &group+1 and PSI < 0.1) THEN RESULT = "GREEN";

ELSE IF (\_N\_ = &group+1 AND 0.1 <= PSI < 0.25) THEN RESULT = "AMBER";

ELSE IF (\_N\_ = &group+1 AND PSI >= 0.25) THEN RESULT = "RED";

RUN;

DATA PSI\_&variable;

SET PSI\_&variable;

id= \_N\_;

run;

proc datasets;

delete ranktable train\_table test\_ext test\_table PSI\_SUM;

run;

%mend;

%let psi\_vars=

finscore

finalscore

pd

;

%macro PSI\_cont\_loop(traindata, testdata, var\_list,group);

data psi\_cont\_vars;

set \_NULL\_;

run;

%do n=1 %to %sysfunc(countw(&var\_list.));

%let variable=%scan(&var\_list.,&n.);

%PSI(&traindata.,&testdata.,&variable.,&group.);

DATA PSI\_&variable;

length Variable $100.;

SET PSI\_&variable;

Variable="&variable.";

No\_Of\_Buckets= &group.;

run;

proc sql;

create table summ\_&variable as select Variable,PSI,Result,id,No\_Of\_Buckets from PSI\_&variable

having id=max(id);

quit;

data psi\_cont\_vars;

set psi\_cont\_vars summ\_&variable;

run;

%end;

%mend;

%PSI\_cont\_loop(glcs\_app,glcs\_dev\_2,&psi\_vars.,5);

%PSI\_cont\_loop(glcs\_app,glcs\_lrdr,&psi\_vars.,5);

%PSI\_cont\_loop(glcs\_app,glcs\_calib,&psi\_vars.,5);

\*PSI FOR CATEGORICAL VARIABLE;

options mlogic mprint symbolgen;

%macro PSI\_categ(traindata, testdata, category);

proc sql;

select count(\*) into: total\_train from &traindata; quit;

proc sql;

select count(\*) into: total\_test from &testdata; quit;

/\*proc rank data=&traindata. out= ranktable groups=&group;\*/

/\*var &variable.;\*/

/\*ranks &variable.\_rank;\*/

/\*run;\*/

proc sql;

create table train\_table as

select distinct &category., count(\*) as train\_obs,count(\*)/&total\_train. as train\_obs\_perc

from &traindata.

group by &category.;

quit;

proc sql;

create table test\_table as

select distinct &category., count(\*) as test\_obs,count(\*)/&total\_test. as test\_obs\_perc

from &testdata.

group by &category.;

quit;

proc sql;

create table combined as

select train\_table.\*, test\_table.\*

from train\_table left join test\_table on train\_table.&category. = test\_table.&category.;

run;

proc sql;

create table PSI\_&category as

select \*, (train\_obs\_perc - test\_obs\_perc) as train\_test, log(train\_obs\_perc / test\_obs\_perc) as Log\_train\_test,

(train\_obs\_perc - test\_obs\_perc) \* log(train\_obs\_perc / test\_obs\_perc) as PSI

from combined;

run;

DATA PSI\_&category ;

set PSI\_&category ;

if &category. ne '.';

run;

PROC SQL;

CREATE TABLE PSI\_SUM AS

SELECT SUM(PSI) AS PSI FROM PSI\_&category;

QUIT;

DATA PSI\_&category;

SET PSI\_&category PSI\_SUM;

RUN;

proc sql;

select count(distinct(&category.)) into: group from PSI\_&category;

quit;

DATA PSI\_&category;

SET PSI\_&category;

IF (\_N\_ = &group.+1 and PSI < 0.1) THEN RESULT = "GREEN";

ELSE IF (\_N\_ = &group.+1 AND 0.1 <= PSI < 0.25) THEN RESULT = "AMBER";

ELSE IF (\_N\_ = &group.+1 AND PSI >= 0.25) THEN RESULT = "RED";

RUN;

DATA PSI\_&category;

SET PSI\_&category;

id= \_N\_;

run;

proc datasets;

delete ranktable train\_table test\_ext test\_table PSI\_SUM;

run;

%mend;

%let categ\_vars=

crr\_32\_rk

/

;

%macro PSI\_categ\_loop(traindata, testdata);

data psi\_categ\_vars;

set \_NULL\_;

run;

%do n=1 %to %sysfunc(countw(&categ\_vars.));

%let category=%scan(&categ\_vars.,&n.);

proc sql;

select count(distinct &category) into: buckets from &traindata.;

quit;

%PSI\_categ(&traindata., &testdata., &category.);

DATA PSI\_&category;

length Variable $100.;

SET PSI\_&category;

Variable="&category.";

No\_Of\_Buckets= &buckets.;

run;

proc sql;

create table summ\_&category as select Variable,PSI,Result,id,No\_Of\_Buckets from PSI\_&category

having id=max(id);

quit;

data psi\_categ\_vars;

set psi\_categ\_vars summ\_&category;

run;

%end;

%mend;

%PSI\_categ\_loop(glcs\_app,glcs\_dev\_2);

%PSI\_categ\_loop(glcs\_app,glcs\_lrdr);

%PSI\_categ\_loop(glcs\_app,glcs\_calib);

%PSI\_categ\_loop(glcs\_lrdr\_650,glcs\_lrdr\_700);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION 3 : glcs Turnover threshold CRR distribution checks

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

data glcs\_lrdr\_700;

set mep\_glcs.glcs\_lrdr;

if total\_revenue>700;

run;

proc freq data=glcs\_lrdr\_700;tables crr\_32;run;

proc freq data=glcs\_lrdr\_650;tables crr\_32;run;

data glcs\_lrdr\_650;

set mep\_glcs.glcs\_lrdr;

if total\_revenue>650;

run;