MIS581 Capstone: Business Intelligence and Data Analytics: FEMA Public Assistance Policy Change Proposal

Craig Westerman

Colorado State University Global

Dr. Lisa Bryan

06/05/22

FILENAME REFFILE '/home/u50061858/M581/M5\_581/PublicAssistanceFundedProjectsDetails\_000--000000.xlsx';

PROC IMPORT DATAFILE=REFFILE

DBMS=XLSX

OUT=M581.'IMPORT.FIPS'n;

GETNAMES=YES;

RUN;

PROC CONTENTS DATA=M581.'IMPORT.FIPS'n; RUN;

/\*One-Way Frequency for FIPS by IncidentType \*/

proc sort data=M581.'IMPORT.FIPS'n out=Work.SortTempTableSorted;

by incidentType;

run;

proc freq data=Work.SortTempTableSorted order=freq;

tables FIPS / plots=(freqplot cumfreqplot);

by incidentType;

run;

proc delete data=Work.SortTempTableSorted;

run;

/\* FIPS by Declared Year (showing most storms per yr by FIPS) \*/

proc sort data=M581.'IMPORT.FIPS'n out=Work.SortTempTableSorted;

by DeclaredYear;

run;

proc freq data=Work.SortTempTableSorted order=freq;

tables FIPS / plots=(freqplot cumfreqplot);

by DeclaredYear;

run;

proc delete data=Work.SortTempTableSorted;

run;

/\* Pull States with Disasters by year \*/

proc sort data=M581.'IMPORT.FIPS'n out=Work.SortTempTableSorted;

by state incidentType;

run;

/\* Exploring Data \*/

proc univariate data=Work.SortTempTableSorted;

ods select Histogram;

var DeclaredYear;

histogram DeclaredYear;

by state incidentType;

run;

proc delete data=Work.SortTempTableSorted;

run;

/\* Distribution of Years for Disaster counts & trends over 20 yrs\*/

roc univariate data=M581.'IMPORT.FIPS'n;

ods select Histogram;

var DeclaredYear;

histogram DeclaredYear;

run;

proc univariate data=M581.'IMPORT.FIPS'n;

ods select ProbPlot QQPlot;

var DeclaredYear;

/\* Checking for Normality \*/

probplot DeclaredYear / normal(mu=est sigma=est);

qqplot DeclaredYear / normal(mu=est sigma=est);

run;

/\* Disaster Yr by State by Disaster # (DR) \*/

proc sort data=M581.'IMPORT.FIPS'n out=Work.SortTempTableSorted;

by state DeclaredYear;

run;

/\* Exploring Data \*/

proc univariate data=Work.SortTempTableSorted;

ods select Histogram;

var disasterNumber;

histogram disasterNumber / normal;

inset n / position=ne;

by state DeclaredYear;

run;

proc delete data=Work.SortTempTableSorted;

run;

\* Proc Sort of FIPS by Disaster Year \*/

proc sort data=M581.'IMPORT.FIPS'n out=Work.SortTempTableSorted;

by ' totalObligated'n;

run;

proc freq data=Work.SortTempTableSorted order=freq;

tables FIPS / plots=(freqplot cumfreqplot);

weight DeclaredYear;

by ' totalObligated'n;

run;

proc delete data=Work.SortTempTableSorted;

run;

/\* Data Exploration table analysis by DR# with Decl Yr by states by Incidents \*/

proc sort data=M581.'IMPORT.FIPS'n out=WORK.TempSorted5215;

by incidentType;

run;

/\* Scatter plot matrix macro \*/

%macro scatterPlotMatrix(xVars=, title=, groupVar=);

proc sgscatter data=WORK.TempSorted5215;

matrix &xVars / %if(&groupVar ne %str()) %then

%do;

group=&groupVar legend=(sortorder=ascending) %end;

;

title &title;

by incidentType;

run;

title;

%mend scatterPlotMatrix;

/\* Histogram (one-way or two-way) \*/

%macro DEHisto(data=, avar=, classVar=);

%local i numAVars numCVars cVar cVar1 cVar2;

%let numAVars=%Sysfunc(countw(%str(&avar), %str( ), %str(q)));

%let numCVars=%Sysfunc(countw(%str(&classVar), %str( ), %str(q)));

%if(&numAVars>0 & &numCVars>0) %then

%do;

%if(&numCVars=1) %then

%do;

%let cVar=%scan(%str(&classVar), 1, %str( ), %str(q));

proc sql noprint;

select count(distinct &cVar) into :nrows from &data;

quit;

/\* One-way histogram \*/

proc univariate data=&data noprint;

var &avar;

class &cVar;

histogram &avar / nrows=&nrows;

run;

%end;

%else

%do;

/\* One-way histogram of each class variable \*/

%do i=1 %to %eval(&numCVars);

%let cVar=%scan(%str(&classVar), &i, %str( ), %str(q));

proc sql noprint;

select count(distinct &cVar) into :nrows from &data;

quit;

proc univariate data=&data noprint;

var &avar;

class &cVar;

histogram &avar / nrows=&nrows;

run;

%end;

/\* Two-way histogram \*/

%let cVar1=%scan(%str(&classVar), 1, %str( ),

%str(q));

%let cVar2=%scan(%str(&classVar), 2, %str( ), %str(q));

proc sql noprint;

select count(distinct &cVar1) into :nrows from &data;

quit;

proc sql noprint;

select count(distinct &cVar2) into :ncols from &data;

quit;

proc univariate data=&data noprint;

var &avar;

class &cVar1 &cVar2;

histogram &avar / nrows=&nrows ncols=&ncols;

run;

%end;

%end;

%mend DEHisto;

/\* By group histogram (one-way or two-way) \*/

%macro byGroupDEHisto(data=, level=, num\_level=, byVars=, num\_byvars=, avar=,

classVar=);

%local i j dsid whereClause varnum rc groupInfo;

%do j=1 %to &num\_byvars;

%let varName&j=%scan(%str(&byVars), &j);

%end;

%do i=1 %to &num\_level;

/\* Get group variable values. \*/

data \_null\_;

i=&i;

set &level point=i;

%do j=1 %to &num\_byvars;

call symputx("x&j", strip(&&varName&j), 'l');

%end;

stop;

run;

/\* Build proc sql where clause. \*/

%let dsid=%sysfunc(open(&data));

%let whereClause=;

%do j=1 %to %eval(&num\_byvars-1);

%let varnum=%sysfunc(varnum(&dsid, &&varName&j));

%if(%sysfunc(vartype(&dsid, &varnum))=C) %then

%let whereClause=&whereClause.&&varName&j.="&&x&j"%str( and );

%else

%let whereClause=&whereClause.&&varName&j.=&&x&j.%str( and );

%end;

%let varnum=%sysfunc(varnum(&dsid, &&varName&num\_byvars));

%if(%sysfunc(vartype(&dsid, &varnum))=C) %then

%let whereClause=&whereClause.&&varName&num\_byvars.="&&x&num\_byvars";

%else

%let whereClause=&whereClause.&&varName&num\_byvars.=&&x&num\_byvars;

%let rc=%sysfunc(close(&dsid));

/\* Subsetting the data set. \*/

proc sql noprint;

create table WORK.tempData as select \* from &data

where &whereClause;

quit;

/\* Build plot group info. \*/

%let groupInfo=;

%do j=1 %to %eval(&num\_byvars-1);

%let groupInfo=&groupInfo.&&varName&j.=&&x&j%str( );

%end;

%let groupInfo=&groupInfo.&&varName&num\_byvars.=&&x&num\_byvars;

/\* Create by group Histogram. \*/

title (&groupInfo);

%DEHisto(data=WORK.tempData, avar=&avar, classVar=&classVar);

title;

%end;

%mend byGroupDEHisto;

%scatterPlotMatrix(xVars=disasterNumber DeclaredYear,

title="Scatter plot matrix grouped by state", groupVar=state);

%let num\_level=0;

data WORK.level(keep=incidentType unique\_level);

set WORK.TempSorted5215 end=lastobs;

retain unique\_level 0;

by incidentType;

if last.incidentType then

do;

unique\_level+1;

output;

end;

if lastobs then

call symputx('num\_level', strip(put(unique\_level, best.)), 'l');

run;

%byGroupDEHisto(data=WORK.TempSorted5215, level=WORK.level,

num\_level=&num\_level, byVars=incidentType, num\_byvars=1, avar=disasterNumber

DeclaredYear, classVar=state);

proc datasets library=WORK noprint;

delete TempSorted5215 level tempData;

run;

/\* Model & Forecasting of TotalObligation by Delcared Year by State \*/

proc sort data=M581.'IMPORT.FIPS'n out=Work.preProcessedData;

by state DeclaredYear;

run;

proc arima data=Work.preProcessedData plots

(only)=(series(corr crosscorr) residual(corr normal)

forecast(forecastonly));

identify var=' totalObligated'n;

estimate method=ML;

forecast lead=12 back=0 alpha=0.05 id=DeclaredYear interval=day;

outlier;

by state;

run;

quit;

proc delete data=Work.preProcessedData;

run;

/\* Heat Map of Yr by State \*/

ods graphics / reset width=10in height=8in imagemap;

proc sgplot data=M581.'IMPORT.FIPS'n;

heatmap x=state y=DeclaredYear / name='HeatMap' discretey;

gradlegend 'HeatMap';

xaxis grid valuesrotate=diagonal;

run;

ods graphics / reset;

/\* Sort data by totalObligated (to identify top obligated FIPS) \*/

proc sort data=M581.'IMPORT.FIPS'n out=work.sortDS

(drop=disasterNumber declarationDate applicationTitle applicantId

damageCategoryCode dcc countyCode stateCode stateNumberCode ' projectAmount'n

' federalShareObligated'n hash lastRefresh id) equals;

by descending ' totalObligated'n;

run;

/\* Frequencies for Categorical Values by IncidentType \*/

proc freq data=M581.'IMPORT.FIPS'n;

tables incidentType / plots=(freqplot);

run;