```
ods pdf file='~/contDataAna/Project/Data/PDF_SAScode.pdf';

libname dataset '~/contDataAna/Project/Data';

%let datafolder =  ~/contDataAna/Project/Data;
/***************/
/* import data */
/***************/
proc import datafile = "&datafolder/support.txt"
            out= dataset.support
            dbms=dlm replace;
  guessingrows=max;
  getnames=yes;
  datarow=2;
run;

/* Convert 'NA' to missing (.) and ensure all columns are numeric */
data support (keep = age sex dzclass num_co edu_num slos totcst_num);
    set dataset.support;
    if totcst = 'NA' then totcst = '';
    if edu = 'NA' then edu = '';
    totcst_num = input(totcst, best32.);
    edu_num = input(edu, best32.);
run;

/* add dummy variables for dzclass and sex */
data support_fact;
  set support;
  if dzclass=1 then dzclass1=1; else dzclass1=0;
  if dzclass=2 then dzclass2=1; else dzclass2=0;
  if dzclass=3 then dzclass3=1; else dzclass3=0;
  if dzclass = 4 then dzclass4 = 1; else dzclass4 = 0;
  if sex = 1 then sex1 = 1;  else sex1 = 0;
  if sex = 2 then sex2 = 1; else sex2 = 0;
  rename totcst_num = totcst;
  rename edu_num = edu;

run;

 /* add interaction terms and log(tot_cst)*/
data support_fact;
    set support_fact;
    totcst_log = log(totcst);
    intagedzclass1 = age * dzclass1;
    intagedzclass2 = age * dzclass2;
    intagedzclass3 = age * dzclass3;
    intagedzclass4 = age * dzclass4;
    intsex1dzclass1 = sex1 * dzclass1;
    intsex1dzclass2 = sex1 * dzclass2;
    intsex1dzclass3 = sex1 * dzclass3;
    intsex1dzclass4 = sex1 * dzclass4;
    intsex2dzclass1 = sex2 * dzclass1;
    intsex2dzclass2 = sex2 * dzclass2;
    intsex2dzclass3 = sex2 * dzclass3;
    intsex2dzclass4 = sex2 * dzclass4;
    intnum_codzclass1 = num_co * dzclass1;
    intnum_codzclass2 = num_co * dzclass2;
    intnum_codzclass3 = num_co * dzclass3;
    intnum_codzclass4 = num_co * dzclass4;
    intedudzclass1 = edu * dzclass1;
    intedudzclass2 = edu * dzclass2;
    intedudzclass3 = edu * dzclass3;
    intedudzclass4 = edu * dzclass4;
    intagesex1 = age * sex1;
    intagesex2 = age * sex2;
    intagenum_co = age * num_co;
    intageedu = age * edu;
    intnum_cosex1 = num_co * sex1;
    intnum_cosex2 = num_co * sex2;
    intedusex1 = edu * sex1;
    intedusex2 = edu * sex2;
    intedunum_co = num_co * edu;
run;

/*******************************/
/* split in train and test data */
/*******************************/
/*note: for descriptive statistics all data is used, not only the training data */

proc surveyselect data=support_fact seed=123 rate=.75 outall out=support_fact2;
run;
```

```
data train test;
  set support_fact2;
  if Selected = 1 then output train;
                  else output test;
run;

/*************************/
/* descriptive statistics */
/*************************/
/* overview frequence and where missing data */
proc freq data= support_fact;
    table dzclass totcst age sex edu num_co/missing;
run;
proc means data = support_fact missing;
    var dzclass totcst age sex edu num_co;
run;
/* only missing data in edu and totcst (log(totcst)) */

/*
proc freq data= support_fact;
    table  edu*totcst totcst*dzclass totcst*sex totcst*num_co edu*dzclass edu*sex edu*num_co;
run;
*/
/*Binning of totcst into a categorical variable where cat 100 represents missing values*/
data support_fact;
    set support_fact;
    if totcst <= 400000 then cst_cat = 9;
    if totcst <= 350000 then cst_cat = 8;
    if totcst <= 300000 then cst_cat = 7;
    if totcst <= 250000 then cst_cat = 5;
    if totcst <= 200000 then cst_cat = 4;
    if totcst <= 150000 then cst_cat = 3;
    if totcst <= 100000 then cst_cat = 2;
    if totcst <= 50000 then cst_cat = 1;
    if totcst = . then cst_cat = 100;
    edu_ms = edu;
    if edu = . then edu_ms = 100;
run;

/*
proc freq data= support_fact;
    table  edu_ms*cst_cat cst_cat*dzclass cst_cat*sex cst_cat*num_co edu_ms*dzclass edu_ms*sex edu_ms*num_co;
run;
*/

/* Histograms showing distribution of included parameters for which the cost/edu value is missing */

%macro histogram(var=,cat=cst_cat);
proc sgplot data = support_fact;
title "distribution of missing data in cost";
  histogram &var;
  where cst_cat = 100;
run;
%mend;

%histogram(var=age);
%histogram(var=num_co);
%histogram(var=edu_ms);
%histogram(var = dzclass);
%histogram(var = sex);

%macro histogram(var=,cat=edu_ms);
proc sgplot data = support_fact;
title "distribution of missing data in education";
  histogram &var;
  where edu_ms = 100;
run;
%mend;

%histogram(var=age);
%histogram(var=num_co);
%histogram(var=cst_cat);
%histogram(var = dzclass);
%histogram(var = sex);

/* Distribution of important variables for which the cost is highest*/

/* Note: outlier and leverage plots are available from line 354 */

%macro histogram(var=,cat=edu_ms);
proc sgplot data = support_fact;
```

```
title "distribution of outliers in cost";
  histogram &var;
  where cst_cat >= 6 and cst_cat <= 9;
run;
%mend;

%histogram(var=age);
%histogram(var=num_co);
%histogram(var=edu_ms);
%histogram(var = dzclass);
%histogram(var = sex);


/* scatter plot matrix*/
proc sgscatter data=support_fact;
  title "Scatterplot Matrix";
  matrix totcst totcst_log age num_co edu/ diagonal=(histogram normal);
run;

/* correlations */
proc corr data=support_fact;
  var totcst totcst_log age sex dzclass num_co edu;
run;

/* boxplots */
%macro box(var=,cat=dzclass);
proc sgplot data = support_fact;
  vbox &var/ category = &cat;
run;
%mend;

%box(var=age);
%box(var=num_co);
%box(var=edu);
%box(var = totcst);
%box(var = totcst_log)

/* table to get bivariate interactions of categorical data */
proc freq data=support_fact;
  tables sex*dzclass /chisq expected norow nocol nopercent;
run;

/***********/
/* model 1 */
/***********/
proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3;
run;

/***********/
/* model 2 */
/***********/
proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age sex1 num_co edu;
run;

/***********************************/
/* model 3 - forward model building */
/***********************************/
/* including 2nd parameter */
proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age/ vif;
    test age=0;
run;

proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 num_co/ vif;
    test num_co = 0;
run;

proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 edu/ vif;
    test edu = 0;
run;

proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 sex1/ vif;
    test sex1 = 0;
run;

proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 sex2/ vif;
```

```
    test sex2 = 0;
run;

/* add age to model -
    check in which functional form with partial residual plot*/
proc reg data=train;
  model totcst_log = dzclass1 dzclass2 dzclass3 age/ partial;
run;

/* add age linearly -
    check if 3rd variable should be included in model*/
proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age num_co/ vif;
    test num_co = 0;
run;

proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age edu/ vif;
    test edu = 0;
run;

proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age sex1/ vif;
    test sex1 = 0;
run;

proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age sex2/ vif;
    test sex2 = 0;
run;

/* add num_co to model -
    check in which functional form with partial residual plot*/
proc reg data=train;
  model totcst_log = dzclass1 dzclass2 dzclass3 age num_co/ partial;
run;

/* add num_co linearly -
    check if 3rd variable should be included in model*/
proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age num_co edu/ vif;
    test edu = 0;
run;

proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age num_co sex1/ vif;
    test sex1 = 0;
run;

proc reg data = train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age num_co sex2/ vif;
    test sex2 = 0;
run;

/* no 3rd contributes significantly enough to add,
 need for pruning checked, VIF stays under 10 => no pruning needed*/

/* check if there are interactions that should be added */
proc glm data=train;
  class dzclass;
  model totcst_log = dzclass age num_co dzclass*age;
run;

proc glm data=train;
  class dzclass;
  model totcst_log = dzclass age num_co dzclass*num_co;
run;

proc glm data=train;
  class dzclass;
  model totcst_log = dzclass age num_co num_co*age;
run;

*this one is checked because descriptive statistics suggested that this might be contributing to the variation of tot_cst;
proc glm data=train;
  class dzclass;
  model totcst_log = dzclass age num_co edu*sex;
run;

/* no significant interaction terms */

/* final model */
```

```
proc glm data = train;
   class dzclass;
   model totcst_log = dzclass1 dzclass2 dzclass3 age num_co / solution;
   output out=resid r=rman p=pman student=student cookd=cook lcl=Lower ucl=Upper dffits=dff;
   store glmmodel;
run;


/*********************************************************/
/* model diagnostics and outlier detection final model */
/*********************************************************/
/* check the squared residuals as a diagnostic */
data resid2;
  set resid;
  rman2=rman**2;
  n=_n_;
run;
proc sgplot data=resid2;
  scatter x=pman y=rman2;
  refline 0 / axis=y lineattrs=(color=red);
run;
proc sgplot data=resid2;
  scatter x=n y=cook;
run;
proc sgplot data=resid2;
  scatter x=n y=dff;
run;
proc reg data=train plots(label)=(CooksD RStudentByLeverage DFFITS DFBETAS);
  model totcst_log = dzclass1 dzclass2 dzclass3 age num_co / r influence;
  output out=lev h=leverage student=stud;
run;

/* Plot distributions of variables for cases with a high leverage or residuals*/

%macro histogram(var=,cat=leverage);
proc sgplot data = lev;
title "distribution of highest leverage cases";
  histogram &var;
    where leverage >=0.019;
run;
%mend;

%histogram(var=age);
%histogram(var=num_co);
%histogram(var=edu_ms);
%histogram(var = dzclass);
%histogram(var = totcst_log);

%macro histogram(var=,cat=stud);
proc sgplot data = lev;
title "distribution of outlier cases (studentised residuals)";
  histogram &var;
  where stud >= 2 or stud <=-2;
run;
%mend;

%histogram(var=age);
%histogram(var=num_co);
%histogram(var=edu_ms);
%histogram(var = dzclass);
%histogram(var = totcst_log);

/***********************************/
/* interpret parameters of interest */
/***********************************/
/* interpret and test parameters of interest */
proc reg data=train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age num_co;
    diseaseClass: test dzclass1=0, dzclass2=0, dzclass3=0;
run;
quit;

proc reg data=train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age num_co;
    diseaseClass: test dzclass1=0;
run;
quit;

proc reg data=train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age num_co;
    diseaseClass: test dzclass2=0;
run;
```

```
quit;

proc reg data=train;
    model totcst_log = dzclass1 dzclass2 dzclass3 age num_co;
    diseaseClass: test dzclass3=0;
run;
quit;

/*************************************************/
/* estimate average cost all people in the sample */
/*************************************************/
proc plm restore=glmmodel;
    score data=support_fact2 out=fullScored
    pred=Predicted lcl=Lower ucl=Upper residual=res STDERR=std;
run;

data fullScored2;
    set fullScored;
    n=_n_;
run;

proc sgplot data=fullScored2;
  scatter x=n y=res / group=Selected;
run;

proc sgplot data=fullScored2;
      scatter x=n y=Predicted / group=Selected;
      scatter x=n y=totcst_log / group=Selected;
run;

/*******************************/
/* fit final model to test data */
/*******************************/
/* use final model for prediction on test data */
proc plm restore = glmmodel;
   score data=test out=ScoreResults_test pred=Predicted lcl=Lower ucl=Upper residual= res STDERR=std;
run;
proc print data = ScoreResults_test;
run;
proc plm restore = glmmodel;
   score data=train out=ScoreResults_train pred=Predicted lcl=Lower ucl=Upper residual= res STDERR=std;
run;
proc print data = ScoreResults_train;

data TestScored2;
    set ScoreResults_test;
    n=_n_;
run;

proc sgplot data=TestScored2;
      scatter x=n y=Predicted;
      scatter x=n y=totcst_log;
run;

proc sgplot data=TestScored2;
  scatter x=n y=res;
run;

/* compare estimates and SE's with training set */
proc sgplot data=fullScored2;
  histogram std / group=selected transparency=0.5;       /* SAS 9.4m2 */
  density std / type=kernel group=selected; /* overlay density estimates */
run;

proc sgplot data=fullScored2;
  histogram predicted / group=selected transparency=0.5;       /* SAS 9.4m2 */
  density predicted / type=kernel group=selected; /* overlay density estimates */
run;

proc sgplot data = fullScored2;
    scatter x = totcst_log y = std/group = selected;
run;

proc sgplot data = fullScored2;
    scatter x = totcst_log y = predicted/group = selected;
run;

/* derive prediction intervals training data and assess coverage with test-data*/
* get amount of times values from test data set fall withinn prediction interval - prediction intervals were already deriv
data ScoreResults_test;
    set ScoreResults_test;
```