July 17, 2015

SAS code only for Practical Guide to Logistic Regression

Joseph M Hilbe, *Arizona State University*

Joseph M. Hilbe: *Practical Guide to Logistic Regression*
Full SAS Code by Yang Liu

Table of Contents

Chapter 1 - SAS Code

Section 1.4

```
*Import medpar as a temporary dataset;
proc import datafile="c:\data\medpar.dta" out=medpar dbms=dta replace;
run;

*Print the first six observations;
proc print data=medpar (obs=6);
run;

*Build the logistic model;
proc genmod data=medpar descending;
      model died=hmo white/ dist=binomial link=logit;
run;

*Another way to build the logistic model;
proc logistic data=medpar descending;
      model died=hmo white / clparm=both;
run;
```

Chapter 2 - SAS Code

Section 2.1

```
*Create a new dataset with binary variables x and y;
data xdta;
      input x y @@;
      datalines;
 1 1 0 0 1 0 0 1 1
 0 1 1 1 0 0 1 0 1
 ;
run;


*Build the logistic model;
proc genmod data=xdta descending;
      model y=x / dist=binomial link=logit;
      output out=residual resdev=deviance;
run;


*Another way to build the logistic model;
proc logistic data=xdta descending;
      model y=x / clparm=both;
      output out=residual resdev=deviance;
run;

*Statistics of deviance residual;
proc means data=residual min q1 median q3 max maxdec=4;
      var deviance;
run;
```

```
*Generate a table of y by x;
proc freq data=xdta;
      tables y*x / norow nocol nocum nopercent;
run;


*Estimate statement with exp option provides the odds ratio;
proc genmod data=xdta descending;
      model y=x / dist=binomial link=logit;
      estimate "Intercept" Intercept 1 / exp;
      estimate "x" x 1 / exp;
run;
```

Section 2.2

```
*Build the logistic model;
proc genmod data=xdta descending;
      model y=x / dist=binomial link=logit;
run;

*Create a dataset to make calculations;
data data1;
      set xdta;
      if x=1 then xb=1.0986-1.5041*1;
      else if x=0 then xb=1.0986-1.5041*0;
      mu=1/(1+exp(-xb));
      o=mu/(1-mu);
      or=o;
      if or < 1 then or=0.667/3;
      coeff=log(or);
      format mu 4.2 o or xb coeff 7.4;
run;

*Print the dataset;
proc print data=data1;
      var x o;
run;

*Print the whole dataset;
proc print data=data1;
run;
```

Section 2.3

```
*Build the logistic model- covb option provides var-cov matrix;
proc genmod data=xdta descending;
      model y=x / dist=binomial  link=logit covb;
run;

*Use SAS interactive matrix language;
proc iml;
      vcov={1.33333 -1.33333,
           -1.33333  2.16667};
      se=sqrt(diag(vcov));
      print se;
quit;

*Logistic regression with OIM standard error;
proc surveylogistic data=xdta;
      model y(event='1')=x;
run;

*Covb option provides var-cov matrix;
proc genmod data=xdta descending;
      model y=x / dist=binomial link=logit covb;
run;

*Calculations of odds ratio and model statistics;
proc iml;
      vcov={1.33333 -1.33333,
           -1.33333  2.16667};
      coef={1.0986, -1.5041};
      or=exp(coef);
      se=sqrt(diag(vcov));
      ose=se*or;
      print or [format = 7.4] ose [format = 7.4];

      zscore=coef/se;
      delta=ose;
      z=zscore[,+];
      pvalue=2*(1-probnorm((abs(z))));
      print z pvalue;

      se1=se[,+];
      loci=coef-quantile('normal', 0.975)*se1;
      upci=coef+quantile('normal', 0.975)*se1;
      expl=exp(loci);
      expu=exp(upci);
      print or [format=7.4] delta [format=7.4] z [format=7.4]
            pvalue [format=7.4] expl [format=7.4] expu [format=7.4];
quit;

*Clparm=both provides both PL and Wald confidence intervals;
proc logistic data=xdta descending;
      model y=x / clparm=both;
run;
```

Section 2.4

```
*Import medpar as a temporary dataset;
proc import datafile="c:\ado\medpar.dta" out=medpar dbms=dta replace;
run;

*Print the first six observations;
proc print data=medpar (obs=6);
run;

*Generate the frequency table of type and output the dataset;
proc freq data=medpar;
      tables type / out=freq;
run;


*Build the logistic model with class;
proc genmod data=medpar descending;
      class type (ref='1') / param = ref;
      model died=type / dist=binomial link=logit;
      estimate "Intercept" Intercept 1 / exp;
      estimate "type2" type 1 0 / exp;
      estimate "type3" type 0 1 / exp;
      output out=residual resdev=deviance;
run;

*Set up format for variable type;
proc format;
      value typefmt 1="Elective Admit"
                    2="Urgent Admit"
                    3="Emergency Admit";
run;

*Logistic regression with controlled reference;
proc genmod data=medpar descending;
      class type (ref='Elective Admit') / param = ref;
      model died=type / dist=binomial link=logit;
      estimate "Intercept" Intercept 1 / exp;
      estimate "type2" type 1 0 / exp;
      estimate "type3" type 0 1 / exp;
      format type typefmt.;
run;

*Logistic regression with controlled reference;
proc genmod data=medpar descending;
      class type (ref='Emergency Admit') / param = ref;
      model died=type / dist=binomial link=logit;
      estimate "Intercept" Intercept 1 / exp;
      estimate "type2" type 1 0 / exp;
      estimate "type3" type 0 1 / exp;
      format type typefmt.;
run;

*Generate the frequency table for variable type;
proc freq data=medpar;
      tables type;
```

```
      format type typefmt.;
run;

*Re-categorized variable type;
data medpar1;
      set medpar;
      if type in (2,3) then type=2;
run;

*Generate the frequency table for re-categorized variable type;
proc freq data=medpar1;
      tables type;
run;

*Logistic regression with re-categorized type;
proc genmod data=medpar1 descending;
      class type (ref='1') / param = ref;
      model died=type / dist=binomial link=logit;
      estimate "Intercept" Intercept 1 / exp;
      estimate "type2" type 1 0 / exp;
      estimate "type3" type 0 1 / exp;
run;
```

Section 2.5

```
*Summary for variable los;
proc means data=medpar min q1 median mean q3 max maxdec=3;
      var los;
run;


*Build the generalized additive model;
proc gam data=medpar;
      model died (event='1')=spline(los) / dist=binomial;
run;

*Import badhealth as a temporary dataset;
proc import datafile="c:\data\badhealth.dta" out=badhealth dbms=dta
replace;
run;

*Print the first six observations;
proc print data=badhealth (obs=6);
run;

*Generate the frequency table for variable badh;
proc freq data=badhealth;
      tables badh / nocum nopercent;
run;

*Summary for variable age;
proc means data=badhealth min q1 median mean q3 max maxdec=2;
      var age;
      output out=center mean=;
run;
```

```
*Create a macro variable;
proc sql;
      select age into: mean
      from center;
quit;

*Summary for variable numvisit;
proc means data=badhealth min q1 median mean q3 max maxdec=3;
      var numvisit;
run;

*Center the age;
data badhealth1;
      set badhealth;
      cage=age-&mean;
run;

*Summary for centered age;
proc means data=badhealth1 min q1 median mean q3 max maxdec=3;
      var cage;
run;

*Provide the std;
proc means data=badhealth std;
      var age;
      output out=stderror std=;
run;

*Create a macro variable;
proc sql;
      select age into: std
      from stderror;
quit;

*Scale age with a different way;
proc standard data=badhealth mean=0 std=&std out=cenage;
      var age;
run;

*Build the logistic model;
proc genmod data=badhealth descending;
      model badh=age / dist=binomial link=logit;
run;

*Build the logistic model with centered age;
proc genmod data=badhealth1 descending ;
      model badh=cage / dist=binomial  link=logit;
run;

*Standardize age and output the sage dataset;
proc standard data=badhealth mean=0 std=1 out=sage;
      var age;
run;

*Build the logistic model with standardized age;
proc genmod data=sage descending ;
      model badh=age / dist=binomial  link=logit;
```

```
run;
```

Section 2.6

```
*Build the logistic model and output model prediction;
proc genmod data=medpar descending;
      model died=white / dist=binomial link=logit;
      output out=etab pred=fitb;
run;

*Generate the frequency table for fitb;
proc freq data=etab;
      tables fitb / nocum nopercent;
run;

*Build the logistic model and output model prediction;
proc genmod data=medpar descending;
      model died=white / dist=binomial link=logit;
      output out=etac pred=fitc;
run;

*Summary for variable fitc;
proc means data=etac min q1 median mean q3 max maxdec=5;
      var fitc;
run;

*Create a dataset to make calculations;
data prob;
      xb20=-0.3617 - 0.0305*20;
      mu20=1/(1+exp(-xb20));
run;

*Print the variable mu20;
proc print data=prob;
      var mu20;
run;

*Build the logistic model and output confidence intervals;
proc genmod data=medpar descending;
      model died=los / dist=binomial  link=logit;
      output out=cl pred=mu lower=loci upper=upci;
run;

*Summary for confidence intervals;
proc means data=cl min q1 median mean q3 max maxdec=5;
      var loci mu upci;
run;



*Graph scatter plot;
proc sgplot data=cl;
      scatter x=los y=mu;
      scatter x=los y=loci;
      scatter x=los y=upci;
run;
```

Chapter 3 - SAS Code

Section 3.1

```
*Import edrelig as a temporary dataset;
proc import datafile="c:\data\edrelig.dta" out=edrelig dbms=dta
replace;
run;

*Print the first six observations;
proc print data=edrelig (obs=6);
run;

*Generate a table of educlevel;
proc freq data=edrelig;
      tables educlevel / nocum nopercent;
run;


*Build logistic model and obtain odds ratio & covariance matrix;
proc genmod data=edrelig descending;
      class educlevel (ref='AA') / param = ref;
      model religious=age male kids educlevel/dist=binomial link=logit
      covb;
      estimate "Intercept" Intercept 1 / exp;
      estimate "Age" age 1 / exp;
      estimate "Male" male 1 / exp;
      estimate "Kid" kids 1 / exp;
      estimate "BA" educlevel 1 0 / exp;
      estimate "MA/PhD" educlevel 0 1 / exp;
run;


*Calculations of odds ratio and model statistics;
proc iml;
      vcov={0.10887 -0.002631 -0.003488 -0.006413 -0.02009 -0.008667,
           -0.002631 0.0001073 -0.000201 -0.000882 -0.000099 -0.000310,
           -0.003488 -0.000201 0.03449 0.0000180 -0.004585 -0.01575,
           -0.006413 -0.000882 0.0000180 0.04425 0.005386 0.003250,
           -0.02009 -0.000099 -0.004585 0.005386 0.04336 0.02292,
           -0.008667 -0.000310 -0.01575 0.003250 0.02292 0.05117};
      coef={-1.4352, 0.0398, 0.1900, 0.1239, -0.4723, -0.4954};
      se=sqrt(diag(vcov));
      se1=se[,+];
      zscore=coef/se;
      z=zscore[,+];
      or=exp(coef);
      print or;
      delta=se*or;
      pvalue=2*(1-probnorm((abs(z))));
      loci=coef-quantile('normal', 0.975)*se1;
      upci=coef+quantile('normal', 0.975)*se1;
      expl=exp(loci);
      expu=exp(upci);
      print or [format=7.4] delta [format=7.4] z [format=7.4]
```

```
            pvalue [format=7.4] expl [format=7.4] expu [format=7.4];
      odds1=1/or[5];
      odds2=1/or[6];
      print odds1 odds2;
quit;
```

Section 3.2

```
*Build the logistic model and obtain the deviance residual;
proc genmod data=edrelig descending;
      class educlevel (ref='AA') / param = ref;
      model religious=age male kids educlevel/dist=binomial link=logit;
      output out=residual resdev=deviance;
run;

*Statistics of deviance residual;
proc means data=residual min q1 median mean q3 max maxdec=4;
      var deviance;
run;



*Build the logistic model and obtain the Person residual;
proc genmod data=edrelig descending;
      class educlevel (ref='AA') / param = ref;
      model religious=age male kids educlevel/dist=binomial link=logit;
      output out=residuals reschi=pearson;
run;

*Pearson Chi2 statistic;
proc sql;
      create table pr as
      select sum(pearson**2) as pchi2, sum(pearson**2)/595 as disp
      from residuals;
quit;

*Print the Chi2 statistic;
proc print data=pr;
run;



*Build the logistic model and obtain statistic;
proc genmod data=edrelig descending;
      class educlevel (ref='AA') / param = ref;
      model religious=age male kids educlevel/dist=binomial link=logit;
      output out=obstats leverage=hat stdreschi=stdp stdresdev=stddev;
run;

*Summary for statistic;
proc means data=obstats min q1 median mean q3 max maxdec=6;
      var hat stdp stddev;
run;
```

Section 3.3

```
*Build the logistic model with class;
proc genmod data=medpar descending;
      class type (ref='1') / param = ref;
      model died=white hmo los type / dist=binomial link=logit covb;
run;
```

Section 3.4

```
*Create a macro variable;
proc sql;
      select disp into: disp
      from pr;
quit;


*Calculations of scaling standard error;
proc iml;
      vcov={0.04799 -0.04125 -0.003194 -0.000563 -0.005717 -0.002297,
           -0.04125  0.04373 -0.001326  0.0000612 0.002537  0.0007290,
           -0.003194 -0.001326 0.02287   0.0000272 0.001369  0.003236,
           -0.000563 0.0000612 0.0000272 0.0000608 -0.000122 -0.000374,
           -0.005717 0.002537  0.001369 -0.000122  0.02083   0.005140,
           -0.002297 0.0007290 0.003236 -0.000374  0.005140  0.05263};
      coef={-0.7202,0.3037,0.0272,-0.0372,0.4179,0.9338};
      se=sqrt(diag(vcov));
      se1=se[,+];
      print coef se1;

      scse=se1*sqrt(&disp);
      print coef se1 scse;
quit;


*Sort the dataset;
proc sort data=medpar;
      by descending type;
run;

*Use quasilikelihood function to generate scaling standard error;
proc glimmix data=medpar order=data;
      class type;
      model died (event='1')=white hmo los type/dist=binary link=logit
      solution;
      random _RESIDUAL_;
run;

*Generate the robust standard errors;
proc surveylogistic data=medpar;
      class type (ref='1') / param=ref;
      model died (event='1')=white hmo los type;
run;
```

```
*Generate the bootstrapped standard errors;
%macro bootstrap (Nsamples);
proc surveyselect data=medpar out=boot
      seed=30459584 method=urs samprate=1 rep=&nsamples.;
run;

proc genmod data=boot descending;
      class type (ref='1') / param=ref;
      model died =white hmo los type / dist=binomial link=logit;
      freq numberhits;
      by replicate;
      ods output ParameterEstimates=est;
run;

data est1;
      set est;
      parameter1=parameter;
      if parameter="Scale" then delete;
      if level1=2 then parameter1="type2";
      else if level1=3 then parameter1="type3";
run;

proc means data=est1 mean;
      class parameter1;
      var StdErr;
run;
%mend;
%bootstrap(100);
```

Section 3.5

```
*Build the logistic model;
proc genmod data=medpar descending;
      model died =white los / dist=binomial link=logit;
run;


*Build the logistic model with interaction;
proc genmod data=medpar descending;
      model died =white los white*los/ dist=binomial link=logit;
run;


*Generate odds ratios for los from 1 to 40;
data ior;
      do i=1 to 40;
      or=exp(0.7709+(-0.0478*i));
      output;
      end;
run;

*Print the calculated odds ratios;
```

```
proc print data=ior;
run;
```

Chapter 4 - SAS Code

Section 4.1

```
*Build the logistic model and obtain the Person residual;
proc genmod data=medpar descending;
      class type (ref='1') / param=ref;
      model died=white hmo los type / dist=binomial link=logit;
      output out=residuals reschi=pearson;
run;

*Pearson Chi2 statistic;
proc sql;
      create table pr as
      select sum(pearson**2) as pchi2, 1492 as df,
             1-probchi(sum(pearson**2), 1492) as pvalue
      from residuals;
quit;

*Print the Chi2 statistic;
proc print data=pr;
      title "Pearson Chi GOF";
run;

*Type3 option provides the likelihood ratio test;
proc genmod data=medpar descending;
      class type (ref='1') / param=ref;
      model died=white hmo los type / dist=binomial link=logit type3;
run;

*Anscombe residuals can be obtained as a model output in the
SAS/Insight, not in SAS command language;

*Create new variables;
data mylgg;
      set medpar;
      if died=1 then dead=1;
      else if died=0 then alive=1;
      drop died;
      m=sum(alive, dead);
run;

*Transform the dataset;
proc sql;
      create table mylgg1 as
      select white as white, hmo as hmo, type as type, count(alive) as
      alive, count(dead) as dead, count(m) as m
      from mylgg
      group by white, hmo, type;
quit;
```

```
*Print the dataset with selected variables;
proc print data=mylgg1;
      var white hmo type m;
run;

*Print the whole dataset;
proc print data=mylgg1;
run;

*Obstats option provides all the residuals and statistics in Table 4.2;
proc genmod data=medpar descending;
      class type (ref='1') / param=ref;
      model died=white hmo los type / dist=binomial link=logit obstats;
      ods output obstats=stats;
run;

*Square the standardized deviance residual;
data stats1;
      set stats;
      stresdev2=stresdev**2;
run;

*Plot the square of standardized deviance residuals and mu;
proc gplot data=stats1;
      symbol v=circle color=black;
      plot stresdev2*pred / vref=4 cvref=red;
run;

*Plot the leverage and std Pearson residual;
proc gplot data=stats1;
      symbol v=circle color=black;
      plot leverage*streschi / href=0 chref=red;
run;

*Sort the dataset;
proc sort data=medpar out=medpar1;
      by white hmo los type;
run;

*Calculate the sum of the dead;
proc means data=medpar1 sum;
      by white hmo los type;
      var died;
      output out=summary sum=dead;
run;

*Create a new variable alive;
data summary1;
      set summary;
      alive=_freq_-dead;
      drop _type_ _freq_;
run;

*Print the dataset;
proc print data=summary1;
run;
```

```
*Build the logistic model with numeric variables;
proc genmod data=medpar descending;
      model died=los type / dist=binomial link=logit;
run;

*Output the los;
proc freq data=medpar;
      tables los / out=los;
run;

*Prepare for the conditional effects plot;
data effect;
      set los;
      k1=-0.8714+(-0.0376)*los+0.4386*1;
      r1=1/(1+exp(-k1));
      k2=-0.8714+(-0.0376)*los+0.4386*2;
      r2=1/(1+exp(-k2));
      k3=-0.8714+(-0.0376)*los+0.4386*3;
      r3=1/(1+exp(-k3));
run;

*Graph the conditional effects plot;
proc sgplot data=effect;
      scatter x=los y=r1;
      scatter x=los y=r2;
      scatter x=los y=r3;
      xaxis label='LOS';
      yaxis label='Type of Admission' grid values=(0 to 0.4 by 0.1);
      title 'P[Death] within 48 hr admission';
run;
```

Section 4.2

```
*Build the logistic model and output model prediction;
proc genmod data=medpar descending;
      class type (ref='1') / param=ref;
      model died=white hmo los type / dist=binomial link=logit;
      output out=fit pred=mu;
run;

*Calculate the mean;
proc means data=fit mean;
      var mu;
run;

*Build the logistic model and output classification table & ROC curve;
proc logistic data=medpar descending plots(only)=ROC;
      class type (ref='1') / param=ref;
      model died=white hmo los type / outroc=ROCdata ctable pprob=(0 to
1 by 0.0025);
      ods output classification=ctable;
run;

*Sensitivity and specificity plot;
symbol1 interpol=join color=vibg height=0.1 width=2;
symbol2 interpol=join color=depk height=0.1 width=2;
```

```
axis1 label=("Probability") order=(0 to 1 by 0.25);
axis2 label=(angle=90 "Sensitivity Specificity %") order=(0 to 100 by
25);
proc gplot data=ctable;
      plot sensitivity*problevel specificity*problevel /
      overlay haxis=axis1 vaxis=axis2 legend;
run;

*Approximate cutoff point can be found when sensitivity and specificity
are closest/equal in the classification table;
```

Section 4.3

```
*Lackfit option provides the Hosmer-Lemeshow GOF test;
proc logistic data=medpar descending;
      class type (ref='1') / param=ref;
      model died=white hmo los type / lackfit;
run;
```

Section 4.4

```
*Import hivlgold as a temporary dataset;
proc import datafile="c:\data\hivlgold.dta" out=HIV dbms=dta replace;
run;

*Build the weighted logistic model;
proc genmod data=HIV descending;
      class cd4 (ref='0') cd8 (ref='0') / param = ref;
      weight cases;
      model infec= cd4 cd8 / dist=binomial link=logit;
run;

*Build the Firth's penalized logistic model;
proc logistic data=HIV descending;
      class cd4 (ref='0') cd8 (ref='0') / param = ref;
      weight cases;
      model infec= cd4 cd8 / firth clodds=pl;
run;
```

Section 4.5

```
*Import azheart as a temporary dataset;
proc import datafile="c:\data\azheart.dta" out=azheart dbms=dta
replace;
run;

*Print the first six observations;
proc print data=azheart (obs=6);
run;

*Generate a table of died by procedure and type;
proc freq data=azheart;
      tables died*procedure died*type / norow nocol nocum nopercent;
run;
```

```
*Build the logistic model and obtain odds ratio & statistics;
proc genmod data=azheart descending;
      model died=procedure type / dist=binomial link=logit;
      estimate "Intercept" Intercept 1 / exp;
      estimate "Procedure" procedure 1 / exp;
      estimate "Type" type 1 / exp;
run;

*Build the quasibinomial logistic model;
proc glimmix data=azheart;
      model died (event='1')=procedure type/dist=binary link=logit
      solution covb;
      random _RESIDUAL_;
run;

*Calculations of odds ratio and model statistics;
proc iml;
      vcov={1.5113 -1.3315 -0.7666,
           -1.3315  1.5288  0.4765,
           -0.7666  0.4765  1.2369};
      coef={-3.2479,2.5380,0.5415};
      se=sqrt(diag(vcov));
      se1=se[,+];
      zscore=coef/se;
      z=zscore[,+];
      or=exp(coef);
      delta=se*or;
      pvalue=2*(1-probnorm((abs(z))));
      loci=coef-quantile('normal', 0.975)*se1;
      upci=coef+quantile('normal', 0.975)*se1;
      expl=exp(loci);
      expu=exp(upci);
      print or [format=7.4] delta [format=7.4] z [format=7.4]
            pvalue [format=7.4] expl [format=7.4] expu [format=7.4];
quit;

*Build the exact logistic model;
proc genmod data=azheart descending;
      model died=procedure type / dist=binomial link=logit;
      exact procedure type / estimate=both;
run;
```

Section 4.6

```
*Create a temporary dataset;
data mydata;
      input y x count @@;
      datalines;
1 1 8 1 0 6 0 1 5 0 0 4
;
run;

*Build the logistic model with weight and obtain odds ratio;
proc genmod data=mydata descending;
      weight count;
```

```
      model y=x / dist=binomial link=logit;
      estimate "Intercept" Intercept 1 / exp;
      estimate "x" x 1 / exp;
run;

*Create a temporary dataset;
data mydata2;
      input grade gender type count @@;
      datalines;
0 0 1 3 0 0 2 4 0 0 3 2 0 1 1 2
0 1 2 4 0 1 3 3 1 0 1 2 1 0 2 1
1 0 3 6 1 1 1 3 1 1 2 2 1 1 3 4
;
run;

*Build the logistic model with weight and obtain odds ratio;
proc genmod data=mydata2 descending;
      class type (ref='1') / param=ref;
      weight count;
      model grade=gender type / dist=binomial link=logit;
      estimate "Intercept" Intercept 1 / exp;
      estimate "Gender" gender 1 / exp;
      estimate "Type2" type 1 0 / exp;
      estimate "Type3" type 0 1 / exp;
run;
```

Chapter 5.2 - SAS Code

Section 5.2

```
*Create a temporary dataset;
data obser;
      input y x1 x2 x3 @@;
      datalines;
1 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 0 0 1
1 0 0 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 0
;
run;

*Build the logistic model;
proc genmod data=obser descending;
      model y=x1 x2 x3 / dist=binomial link=logit;
run;

*Create a temporary dataset;
data grp;
      input y cases x1 x2 x3 @@;
      datalines;
1 3 1 0 1 1 1 1 1 1
2 2 0 0 1 0 1 0 1 1
2 2 1 0 0 0 1 0 1 0
;
run;
```

```
*Build the logistic model;
proc genmod data=grp descending;
      model y/cases=x1 x2 x3 / dist=binomial link=logit;
run;

*Build the logistic model;
proc genmod data=grp descending;
      class x1 (ref='0') x2 (ref='0') x3 (ref='0') / param=ref;
      model y/cases=x1 x2 x3 / dist=binomial link=logit;
run;
```

Section 5.4

```
*Sort the dataset;
proc sort data=medpar out=medpar1;
      by white hmo los type;
run;

*Calculate the sum of the dead;
proc means data=medpar1 sum;
      by white hmo los type;
      var died;
      output out=summary sum=dead;
run;

*Create a new variable alive;
data summary1;
      set summary;
      alive=_freq_-dead;
      cases=_freq_;
      drop _type_ _freq_;
run;

*Obstats option provides all the residuals and useful statistics;
proc genmod data=summary1 descending;
      class type (ref='1')/ param=ref;
      model dead/cases=white hmo los type / dist=binomial link=logit
      obstats;
      ods output obstats=allstats;
run;

*Plot the leverage and std Pearson residual;
proc gplot data=allstats;
      symbol v=circle color=black;
      plot leverage*streschi / href=0 chref=red;
run;

*Plot the standardized deviance residuals and mu;
proc gplot data=allstats;
      symbol v=circle color=black;
      plot stresdev*pred / vref=2 cvref=red;
run;
```

Section 5.5

```
*Import titanicgrp as a temporary dataset;
proc import datafile="c:\data\titanicgrp.dta" out=azheart dbms=dta
replace;
run;

*Print the dataset;
proc print data=titanicgrp;
run;

*Build the logistic model and obtain odds ratio & covariance matrix;
proc genmod data=titanicgrp descending;
      class class (ref='3')/ param=ref;
      model survive/cases=age sex class/dist=binomial link=logit covb;
      estimate "Intercept" Intercept 1 / exp;
      estimate "ageadults" age 1 / exp;
      estimate "sexman" sex 1 / exp;
      estimate "class" class 1 0 / exp;
      estimate "class" class 0 1 / exp;
run;

*Build the logistic mode with robust adjustment;
proc glimmix data=titanicgrp order=data empirical=hc0;
      class class;
      model survive/cases=age sex class / dist=binomial link=logit
      solution covb;
      random _RESIDUAL_;
run;

*Calculations of odds ratio and model statistics;
proc iml;
      vcov={0.5798 -0.4722 -0.2001 0.04322 -0.05299,
            -0.4722 0.5089 0.1235 -0.1162 -0.06026,
            -0.2001 0.1235 0.2227 -0.1112 -0.01121,
             0.04322 -0.1162 -0.1112 0.2726 0.1219,
            -0.05299 -0.06026 -0.01121 0.1219 0.3524};
      coef={1.2955, -1.0556, -2.3695, 1.7664, 0.7558};
      or=exp(coef);
      rse=sqrt(diag(vcov));
      rse1=rse[,+];
      ORrse=rse*or;
      pvalue=2*(1-probnorm((abs(1/rse1))));
      print or ORrse pvalue;
quit;

*Build the Beta binomial model;
proc fmm data=titanicgrp;
      class class;
      model survive/cases=age sex class / dist=betabinomial;
run;
```

Chapter 6 - SAS Code

Section 6.2

```
*Import rwm1984 as a temporary dataset;
proc import datafile="c:\data\rwm1984.dta" out=rwm1984 dbms=dta
replace;
run;

*Print the first 6 observations;
proc print data=rwm1984 (obs=6);
run;

*Generate the frequency table of outwork ;
proc freq data=rwm1984;
      tables outwork / norow nocol nocum nopercent;
run;

*Summary for continuous variables;
proc means data=rwm1984 min q1 median mean q3 max maxdec=3;
      var docvis age;
      output out=center mean=;
run;

*Create the macro variables;
proc sql;
      select age into: meanage from center;
      select docvis into: meandoc from center;
quit;

*Center the continuous variables;
data R84;
      set rwm1984;
      cage=age-&meanage;
      cdoc=docvis-&meandoc;
run;

*Build the logistic model and obtain odds ratio & statistics;
proc genmod data=R84 descending;
      model outwork=cdoc female kids cage / dist=binomial link=logit;
      estimate "Intercept" Intercept 1 / exp;
      estimate "Cdoc" cdoc 1 / exp;
      estimate "Female" female 1 / exp;
      estimate "Kids" kids 1 / exp;
      estimate "Cage" cage 1 / exp;
run;

*Build the quasibinomial logistic model;
proc glimmix data=R84;
      model outwork (event='1')=cdoc female kids cage / dist=binary
      link=logit solution covb;
      random _RESIDUAL_;
run;

*Calculations of odds ratio and model statistics;
proc iml;
```

```
      vcov={0.007147 -0.00000601 -0.00505 -0.00461 -0.00015,
            -0.00000601 0.000043 -0.00002 0.000022 -0.00000226,
            -0.00505 -0.00002 0.007444 0.000595 0.000062,
            -0.00461 0.000022 0.000595 0.008796 0.000192,
            -0.00015 -0.00000226 0.000062 0.000192 0.000019};
      coef={-2.0103,0.02443,2.2568,0.3580,0.05438};
      se=sqrt(diag(vcov));
      se1=se[,+];
      zscore=coef/se;
      z=zscore[,+];
      or=exp(coef);
      delta=se*or;
      pvalue=2*(1-probnorm((abs(z))));
      loci=coef-quantile('normal', 0.975)*se1;
      upci=coef+quantile('normal', 0.975)*se1;
      expl=exp(loci);
      expu=exp(upci);
      print or [format=7.4] delta [format=7.4] z [format=7.4]
            pvalue [format=7.4] expl [format=7.4] expu [format=7.4];
quit;

*Bayesian logistic regression;
proc genmod data=R84 descending;
      model outwork=cdoc female kids cage / dist=binomial link=logit;
      bayes seed=10231995 nbi=5000 nmc=100000
      coeffprior=uniform diagnostics=all
      statistics=(summary interval) plots=all;
run;

*Create the normal prior;
data prior;
      input _type_ $ Intercept cdoc cage;
      datalines;
Var 1e5 1e5 1e5
Mean 0 0 0
;
run;

*Bayesian logistic regression with normal prior;
proc genmod  data=R84 descending;
      model outwork=cdoc female kids cage / dist=binomial link=logit;
      bayes seed=10231995 nbi=5000 nmc=100000
      coeffprior=normal(input=prior) diagnostics=all
      statistics=(summary interval) plots=all;
run;
```