CODE

Code 1: Dataload

```
LIBNAME healthin '/home/u62518985/';
FILENAME hidata '/home/u62518985/SAS HealthInsurance/00healthinsurance.csv';
PROC IMPORT
  DATAFILE = hidata
  OUT = healthin.healthinsurancedata
  DBMS = CSV
  REPLACE;
  GETNAMES = YES;
RUN;
```

Code 2: Data Labelling

```
title "Heath Insurance Data";
data healthin.healthinsurancedata;
set healthin.healthinsurancedata;
label
age ='age beneficiary'
age_group='Beneficiary age, 0-><=median age, 1->>median age'
sex ='insurance contractor gender'
sex_status = 'insurance contractor gender - 0=female, 1=male'
bmi = 'Body mass index'
children = 'Number of children covered by health insurance / Number of dependents'
smoker = 'Smoking status'
smoker_status = 'Smoking status - 0=No,1=Yes'
region = 'Rhe beneficiarys residential area in the US'
region_value ='Residential area in the US seperated from 0 to 3 based on each region'
charges = 'Individual medical costs billed by health insurance'
charges_seperation= 'Charges seperated based on the median medical costs - <=median=0,
>median=1 ';
run;
```

Code 3: If Else Loop

```
data healthin.healthinsurancedata;
  set healthin.healthinsurancedata;
  if sex = 'female' then sex_status = 0;
  else if sex = 'male' then sex_status = 1;
run;

data healthin.healthinsurancedata;
  set healthin.healthinsurancedata;
  if region='northwest' then region_value=0;
  else if region='northeast' then region_value=1;
  else if region='southwest' then region_value=2;
  else if region='southeast' then region_value=3;
run;

data healthin.healthinsurancedata;
```

```
  set healthin.healthinsurancedata;
  if smoker = 'no' then smoker_status = 0;
  else if smoker = 'yes' then smoker_status = 1;
run;

data healthin.healthinsurancedata;
  set healthin.healthinsurancedata;
  if age<=39 then age_group = 0;
  else if age>39 then age_group = 1;
run;

data healthin.healthinsurancedata;
  set healthin.healthinsurancedata;
  if charges<=9382.03 then charges_seperation_median = 0;
  else if charges>9382.03 then charges_seperation_median = 1;
run;
data healthin.healthinsurancedata;
  set healthin.healthinsurancedata;
  if charges<=13270.42 then charges_seperation_mean = 0;
  else if charges>13270.42 then charges_seperation_mean = 1;
run;
data healthin.healthinsurancedata;
  set healthin.healthinsurancedata;
  if charges<=16657.72 then charges_seperation = 0;
  else if charges>16657.72 then charges_seperation= 1;
run;
```

Code 4: Finding if there is any missing value in the original data

```
proc means data=healthin.healthinsurancedata nmiss;
var age bmi children charges sex_status region_value smoker_status;
run;
```

Code 5: Performing Descriptive Statistics for Continuous Variables- Age, bmi, children and charges

```
proc means data=healthin.healthinsurancedata mean median skew stddev var maxdec=2 q1 q3;
var age bmi children charges;
run;

proc means data=healthin.healthinsurancedata n mean median skew stddev var maxdec=2 kurtosis;
class sex;
var age bmi children charges;
run;

proc means data=healthin.healthinsurancedata n mean median skew stddev var maxdec=2;
class smoker;
var age bmi children charges;
run;


proc means data=healthin.healthinsurancedata n mean median skew stddev var maxdec=2;
class region;
```

```
var age bmi children charges;
run;
```

Code 6: Performing descriptive statistics on categorical variables : sex, region and smoking status.

```
proc univariate data=healthin.healthinsurancedata;
var age bmi children charges;
run;

PROC UNIVARIATE DATA=healthin.healthinsurancedata;
  VAR age;
  HISTOGRAM / NORMAL;
RUN;

PROC UNIVARIATE DATA=healthin.healthinsurancedata;
  VAR bmi;
  HISTOGRAM / NORMAL;
RUN;

PROC UNIVARIATE DATA=healthin.healthinsurancedata;
  VAR children;
  HISTOGRAM / NORMAL;
RUN;

PROC UNIVARIATE DATA=healthin.healthinsurancedata;
  VAR charges;
  HISTOGRAM / NORMAL;
RUN;

proc corr data=healthin.healthinsurancedata;
var age sex_status bmi children smoker_status region_value charges;
run;

/*correlation*/
proc corr data=healthin.healthinsurancedata;
var charges age bmi children smoker_status sex_status region_value;
run;

PROC Freq DATA=healthin.healthinsurancedata;
  table smoker;
RUN;

PROC Freq DATA=healthin.healthinsurancedata;
  table sex;
RUN;

PROC Freq DATA=healthin.healthinsurancedata;
  table region;
RUN;
```

```
PROC UNIVARIATE DATA=healthin.healthinsurancedata;
  VAR age bmi children  charges;
  QQPLOT / NORMAL;
RUN;

PROC UNIVARIATE DATA=healthin.healthinsurancedata;
  VAR age;
  QQPLOT / NORMAL;
RUN;

PROC UNIVARIATE DATA=healthin.healthinsurancedata;
  VAR bmi;
  QQPLOT / NORMAL;
RUN;

PROC UNIVARIATE DATA=healthin.healthinsurancedata;
  VAR children   ;
  QQPLOT / NORMAL;
RUN;

PROC UNIVARIATE DATA=healthin.healthinsurancedata;
  VAR charges;
  QQPLOT / NORMAL;
RUN;

proc freq data=healthin.healthinsurancedata;
tables sex region smoker;
run;

proc gchart data=healthin.healthinsurancedata;
vbar age;
run;

proc gchart data=healthin.healthinsurancedata;
vbar bmi;
run;

proc gchart data=healthin.healthinsurancedata;
vbar children;
run;

proc gchart data=healthin.healthinsurancedata;
vbar charges;
run;

proc sgplot data=healthin.healthinsurancedata;
vbox age;
run;
```

```
proc sgplot data=healthin.healthinsurancedata;
vbox bmi;
run;

proc sgplot data=healthin.healthinsurancedata;
vbox children;
run;

proc sgplot data=healthin.healthinsurancedata;
vbox charges/ fill;
run;
```

Code 7: Performing CMH and the assumption of cell frequency

```
/*checking if the expected frequency count for each cell in the contingency table is greater than 5.*/

proc freq data=healthin.healthinsurancedata;
tables sex_status smoker_status region_value;
run;

proc freq data=healthin.healthinsurancedata;
tables smoker_status*sex_status/cmh;
weight region_value;
run;

proc freq data=healthin.healthinsurancedata;
tables smoker_status*region_value/cmh;
weight sex_status;
run;

proc freq data=healthin.healthinsurancedata;
tables region_value*sex_status/cmh;
weight smoker_status;
run;
```

Code 8: Performing Ttest, assumption of normality and homogeneity, Wilcoxin test

```
/* Check normality assumption */
proc univariate data=healthin.healthinsurancedata normal;
  var charges;
  class smoker;
run;

/* Check homogeneity of variances assumption */
proc glm data=healthin.healthinsurancedata;
  class smoker;
  model charges = smoker;
  means smoker / hovtest=levene;
run;
```

```
/*test for homogenity fails*/
proc npar1way data=healthin.healthinsurancedata wilcoxon;
   class smoker;
   var charges;
run;

/*ttest assumption failed
proc ttest data=healthin.healthinsurancedata;
  class smoker;
  var charges;
run;

-----------------------------------------------------------------------------------------------*/

/* Check normality assumption */
proc univariate data=healthin.healthinsurancedata normal;
  var charges;
  class sex;
run;

/* Check homogeneity of variances assumption */
proc glm data=healthin.healthinsurancedata;
  class sex;
  model charges = sex;
  means sex / hovtest=levene;
  output out=healthin.asst residual=residual p=predicted;
run;

proc gplot data=healthin.asst;
plot predicted*residual;
run;

proc sgplot data=healthin.asst;
scatter x=predicted y=residual;
run;


/* Assumption of homogenity failed
proc ttest data=healthin.healthinsurancedata;
  class sex;
  var charges;
run;
*/

proc npar1way data=healthin.healthinsurancedata wilcoxon;
   class sex;
   var charges;
```

```
run;
/*no relation btwn sex and charges*/
/*----------------------------------------------------------------------------------------------*/

proc univariate data=healthin.healthinsurancedata normal;
  var charges;
  class age_group;
run;

/* Check homogeneity of variances assumption */
proc glm data=healthin.healthinsurancedata;
  class age_group;
  model charges = age_group;
  means age_group / hovtest=levene;
run;

/*assumption of homogenity did not fail*/
proc ttest data=healthin.healthinsurancedata;
  class age_group;
  var charges;
run;
```

Code 9: Performing ANOVA, assumption of homogeneity and normality and npar1way test

```
proc univariate data=healthin.healthinsurancedata normal;
  var charges;
  class region;
run;

/* Check homogeneity of variances assumption */
proc glm data=healthin.healthinsurancedata;
  class region;
  model charges = region;
  means region / hovtest=levene;
run;

/*Levene's test for equality of variances is significant (i.e. p-value < 0.05), it suggests that the
assumption of homogeneity of variances has been violated.
In such a case, the results of the ANOVA may not be reliable.
*Assumption of homogenerity fails*/

proc npar1way data=healthin.healthinsurancedata;
   class region;
   var charges;
   run;
```

```
/*for children*/

proc univariate data=healthin.healthinsurancedata normal;
  var charges;
  class children;
run;

proc glm data=healthin.healthinsurancedata;
  class children;
  model charges = children;
  means children / hovtest=levene;
run;
```

Code 10: Performing Logistic Regression and its assumptions

```
proc logistic data=healthin.healthinsurancedata descending;
model charges_seperation= age bmi children sex_status region_value
smoker_status/selection=stepwise;
output out=healthin.logisticoutput resdev=resdev predicted=predicted;
run;

/*linearity*/
proc sgplot data=healthin.logisticoutput;
scatter x=predicted y=resdev;
xaxis label='Predicted';
yaxis label='Residuals' ;
run;
/*independence*/
data healthin.logisticoutput;
set healthin.logisticoutput;
obs=_n_;
run;

proc sgplot data=healthin.logisticoutput;
scatter x=resdev y=obs;
yaxis label='Observation';
xaxis label='Residuals' ;
run;
/*correlation*/
proc corr data=healthin.healthinsurancedata;
var age bmi children sex_status region_value smoker_status;
run;

proc logistic data=healthin.healthinsurancedata descending;
model charges_seperation_median= age bmi children sex_status region_value / selection=stepwise;
output out=healthin.logisticoutputmedian resdev=resdev predicted=predicted;
run;
```

```
/*linearity*/
proc sgplot data=healthin.logisticoutputmedian;
scatter x=predicted y=resdev;
xaxis label='Predicted';
yaxis label='Residuals' ;
run;

/*independence*/
data healthin.logisticoutputmedian;
set healthin.logisticoutputmedian;
obs=_n_;
run;

proc sgplot data=healthin.logisticoutputmedian;
scatter x=resdev y=obs;
yaxis label='Observation';
xaxis label='Residuals' ;
run;

/*correlation*/
proc corr data=healthin.healthinsurancedata;
var age bmi children sex_status region_value smoker_status;
run;



/*quasi complete seperation
proc logistic data=healthin.healthinsurancedata ;
model charges_seperation_median=smoker_status;
run;

proc autoreg data=healthin.healthinsurancedata;
  model charges_seperation= age bmi children sex_status smoker_status region_value / dw=4
dwprob;
run;*/
```

Code 11: Performing Multiple Regression and its assumptions

```
proc reg data=healthin.healthinsurancedata;
model charges = age bmi children smoker_status sex_status region_value / vif ;
output out=healthin.residuals residual=residual predicted=predicted COOKD=COOKD  ;
run;

proc autoreg data=healthin.healthinsurancedata;
  model charges = age bmi children smoker_status sex_status region_value / dw=4 dwprob;
run;
```

```
data healthin.residuals;
set healthin.residuals;
order = _n_;
run;

/*Assumption of independence order vs residuals*/
title 'Assumption of Independence';
proc sgplot data=healthin.residuals;
scatter x=order y=residual;
xaxis label='Observation';
yaxis label='Residuals';
run;

/*Assumption of variance*/
title 'Assumption of Variance';
proc sgplot data=healthin.residuals;
scatter x=predicted y=residual;
xaxis label='Observation';
yaxis label='Residuals';
run;

/*Assumption of normality*/
title 'Assumption of Normality';
proc univariate data=healthin.healthinsurancedata;
var charges;
histogram /normal;
run;

/*Assumption of Linearity*/
title'Assumption of Linearity';
proc sgplot data=healthin.gammaresiduals;
scatter x=charges y=predicted / markerattrs=(symbol=circlefilled);
lineparm x=0 y=0 slope=1;
xaxis label='Charges';
yaxis label='Predicted Charges';
run;

/*Assumption of Linearity
proc sgplot data=healthin.residuals;
scatter x=predicted y=charges;
lineparm x=0 y=0 slope=1;
xaxis label='Predicted Values';
yaxis label='Actual Values';
run;
*/


/*----------------------------------Gamma------------------------------------------------------------*/
```

```
proc genmod data=healthin.healthinsurancedata;
model charges = age bmi children smoker_status sex_status region_value / dist=gamma  link=log;
output out=healthin.gammaresiduals resdev=resdev predicted=predicted ;
run;

proc genmod data=healthin.healthinsurancedata;
model charges = age/ dist=gamma link=inverse ;
output out=healthin.gammaresiduals resdev=resdev predicted=predicted ;
run;

proc autoreg data=healthin.healthinsurancedata;
   model charges = age bmi children smoker_status sex_status region_value / dw=4 dwprob;
run;

proc reg data=healthin.healthinsurancedata;
model charges=age;
run;

/*Assumption of independence*/
title 'Assumption of Independence';
proc sgplot data=healthin.gammaresiduals;
scatter x=resdev y=charges;
xaxis label='Observation';
yaxis label='Residuals';
run;

data healthin.residuals;
set healthin.residuals;
order = _n_;
run;

/*Assumption of independence order vs residuals*/
title 'Assumption of Independence';
proc sgplot data=healthin.residuals;
scatter x=order y=residual;
xaxis label='Observation';
yaxis label='Residuals';
run;

/*Assumption of Linearity*/
title'Assumption of Linearity';
proc sgplot data=healthin.gammaresiduals;
scatter x=charges y=predicted / markerattrs=(symbol=circlefilled);
xaxis label='Charges';
yaxis label='Predicted Charges';
run;

quit;
```

Code 12: Performing Gamma Regression and its Assumptions

```
/*--------------------------------Gamma--------------------------------------------------------*/
proc genmod data=healthin.healthinsurancedata;
model charges = age bmi children smoker_status sex_status region_value / dist=gamma  link=log  ;
output out=healthin.gammaresiduals resdev=resdev predicted=predicted ;
run;

data healthin.gammaresiduals;
set healthin.gammaresiduals;
order = _n_;
run;

/*Assumption of independence order vs residuals*/
title 'Assumption of Independence';
proc sgplot data=healthin.gammaresiduals;
scatter x=order y=resdev;
xaxis label='Observation';
yaxis label='Residuals' min=-2 max=3 values=(-0.75 to .10 by 0.05);
run;
data healthin.gammaresiduals;
set healthin.gammaresiduals;
logpred=log(predicted);
run;



/*Assumption of Linearity*/
title'Assumption of Linearity';
proc sgplot data=healthin.gammaresiduals;
scatter y=resdev x=logpred / markerattrs=(symbol=circlefilled);
  refline 0 / lineattrs=(color=black);
xaxis label='Log of Predicted Values' values=(8 to 9.5 by .25);
yaxis label='Residual deviance' min=-5 max=1 values=(-1 to 2 by 0.05) ;
run;

quit;
```