```
Libname Group5 "/home/u58677578/BAN110/Project";

/* Loading Data set from txt file */

Data Group5.Auto_Mpg;
Infile "/home/u58677578/BAN110/Project/auto-mpg.data";
Informat Name $30.;
INput @1 Mpg 4.
      @8 Cylinders 1.
      @12 Displacement 5.
      @23 Horsepower 5.
      @34 Weight 5.
      @45 Acceleration 4.
      @52 Year 2.
      @56 Origin 1.
      @58 Name & $30.;
Format Mpg 4.1  Displacement 5.1 Horsepower 5.1 Weight 6.1
       Acceleration 4.1;
Run;


/* Checking Meta Data Description */

Proc Contents Data=Group5.Auto_Mpg;
Run;

Title 'Lisitng First 10 Observations';
Proc Print Data = Group5.Auto_Mpg (obs = 5);
Run;

/* Descriptive statistics and distribution of Targeet variable MPG  */

Title 'Descrpitive Statitics for MPG';
Proc Means Data = Group5.Auto_Mpg;
Var mpg;
Run;

Title 'Histogram of MPG';
Proc Sgplot Data = Group5.Auto_Mpg;
Histogram Mpg;
Density Mpg;
Density Mpg / type=kernel;
Run;


/*      Categorical Values      */


options nolabel;
Title 'Listing Frequencies for Cylinders';
Proc Freq Data=Group5.Auto_Mpg;
Tables Cylinders Year Origin / nocum missing;
Run;
Title;


/*      Checking for missing categorical values using informat method       */
```

```
Proc Format;
Value Origin_Check
        1,2,3 = 'Valid'
        other = 'Invalid';

Value Cyl_Check
        3,4,5,6,8 = 'Valid'
        other = 'Invalid';

Value Year_Check
    70-82 = 'Valid'
    other = 'Invalid';
Run;
```

```
Data _null_;
File Print;
Set Group5.Auto_Mpg (Keep = Name Cylinders Year Origin);
If put(Cylinders, Cyl_Check.) = 'Invalid' then put
    'Missing observation of Cylinders = ' _n_  name Cylinders =;
Else if Put(Year, Year_Check.) = 'Invalid' then put
    'Missing observation of Year = ' _n_ name Year =;
Else if put(Origin,Origin_Check.) = 'Invalid' then put
    'Missing observation of Origin = ' _n_ name Origin = ;
Run;

Title 'Checking for Missing values of Categorical variables';
Proc Freq Data=Group5.Auto_Mpg;
Tables Cylinders Year Origin / nocum nopercent;
Format Cylinders Cyl_Check. Year Year_Check. Origin Origin_Check.;
Run;


/* Converting Date from 2 digit number to Date9. format */
```

```
Data Group5.Auto_Mpg;
Set Group5.Auto_Mpg;
Year_new = Cat('03/01/19',Year);
Year = year(input(Year_new, mmddyy10.));
Drop Year_new;
Run;

Title 'Printing first 5 observations';
Proc Print Data = Group5.Auto_Mpg (obs = 5);
Var Name Year;
Run;


/*      Derive Vehicle Make and Model from Name         */
```

```
Data Group5.Auto_Mpg;
Set Group5.Auto_Mpg;
Name = Propcase(Compress(Name,'"'));
Array model_n [6] $20. Model1-Model6;
Do i = 1 to 6;
Model_n [i] = compress(Scan(Name,i),"'");
End;
If _n_ = 293 then Model3 = '';
Brand = Model1;
Model = Catx('',Model2,model3,model4,model5,model6);
```

```
Drop Model1-Model6 Name i;
Run;


/*      Checking Errors in Brand        */

Title 'Checking errors in Brand';
Proc Freq Data = Group5.Auto_Mpg;
Tables Brand / nocum nopercent;
Run;

/*Correcting Spelling errors for variable Brand */

Data Group5.Auto_Mpg;
Set Group5.Auto_Mpg;
Brand = Tranwrd(Brand,'Vw','Volkswagen');
Brand = Tranwrd(Brand,'Vokswagen','Volkswagen');
Brand = Tranwrd(Brand,'Chevroelt','Chevrolet');
Brand = Tranwrd(Brand,'Chevy','Chevrolet');
Brand = Tranwrd(Brand,'Maxda','Mazda');
Brand = Tranwrd(Brand,'Toyouta','Toyota');
Run;


Data Group5.Auto_Mpg;
Retain Brand Model Cylinders Year Origin MPG Displacement
        Horsepower Weight Acceleration;
Set Group5.Auto_Mpg;
Run;


Title 'Checking Corrected Brands';
Proc Freq Data = Group5.Auto_Mpg;
Tables Brand*Origin / nocum nopercent norow nocol;
Run;

/*      Numerical Variables     */


options nolabel;
Proc MEans Data = Group5.Auto_Mpg
n nmiss min max mean median mode stddev var ;
Var mpg acceleration displacement weight horsepower;
Run;


/* Checking Missing Numeric Observations */

Title 'Identifying Missing numeric values';
Data _null_;
File print;
Set Group5.Auto_Mpg;
Array Numeric [*] _NUMERIC_;
Do i = 1 to Dim(Numeric);
If missing(numeric(i)) then put
    'Missing Observation ' Brand = Model = Mpg = Cylinders = Displacement = Horsepower =
    Weight = Acceleration = ;
End;
Run;
```

```
/* Checking Mean Horsepower for various Cylinder categories */

Proc Means Data = Group5.Auto_Mpg;
Class Cylinders;
Var Horsepower;
Run;

/* Replacing missing horsepower with mean horespower grouped by Cylinders */

Proc Sort Data = Group5.Auto_Mpg; by Cylinders; Run;

Proc Stdize data = Group5.Auto_Mpg out = Group5.Auto_Mpg
reponly method = mean;
by cylinders;
Run;


/* Calculatin a new variable Power-Weight Ratio */

Data Group5.Auto_Mpg;
Set Group5.Auto_Mpg;
PWR = horsepower/weight;
Run;

/*      Detecting outliers before Imputing missing values of Horsepower     */


Proc Univariate Data = Group5.Auto_Mpg plots;
Var mpg acceleration displacement weight horsepower pwr;
Run;

/* After Checking we see variable Acceleration has normal distribution. Hence, we will use
   Standard Deviation method to detect Outliers     */


Proc Means Data = Group5.Auto_Mpg noprint;
Var  Acceleration;
Output out = Means (drop = _type_ _freq_)
    Mean =
    Std = / autoname;
Run;

Proc Means Data = Group5.Auto_Mpg noprint;
Var pwr;
Output out = IQR (drop = _type_ _freq_)
    Q1 =
    Q3 =
    Qrange = / autoname;
Run;

/* Detecting OUtliers for Power-Weight Ration using Inter Quartile Range */

Title 'Listing Outliers for Power-Weight Ratio';
Data _NULL_;
Set Group5.Auto_Mpg (keep = pwr Brand Model);
File Print;
If _n_ =1 then set IQR;
If pwr < pwr_Q1 - 1.5*pwr_Qrange or
  pwr > pwr_Q3 + 1.5*pwr_Qrange then
Put 'Outlier detected for ' Brand Model ' Power-Weight ratio = ' pwr;
```

```
Run;
Title;

Title 'Listing Outliers for Power-Weight Ratio';
Data Group5.Auto_Mpg;
Set Group5.Auto_Mpg;
If _n_ =1 then set IQR;
If pwr < pwr_Q1 - 1.5*pwr_Qrange or
  pwr > pwr_Q3 + 1.5*pwr then delete;
Drop pwr_Q1 pwr_Q3 pwr_Qrange;
Run;
Title;

/* Detecting Outliers for Acceleration */

Title 'Listing Outliers for Acceleration';
Data _NULL_;
Set Group5.Auto_Mpg (keep = Acceleration Brand Model);
File Print;
If _n_ =1 then set Means;
If Acceleration <= Acceleration_Mean - 2*Acceleration_StdDev or
  Acceleration > Acceleration_Mean + 2*Acceleration_StdDev then
Put 'Outlier detected for ' Brand Model ' where Acceleration = ' Acceleration;
Run;

Title 'Listing Outliers for Acceleration';
Data Group5.Auto_Mpg;
Set Group5.Auto_Mpg;
If _N_ = 1 then set means;
If Acceleration < Acceleration_Mean - 2*Acceleration_StdDev or
  Acceleration > Acceleration_Mean + 2*Acceleration_StdDev then delete;
Drop Acceleration_MEan Acceleration_StdDev;
Run;

Proc Univariate Data = Group5.Auto_Mpg plots;
Var Acceleration;
Run;


/* Checking Skewness of Variable Horsepower using QQplot and Histogram */

Title 'Histogram for Horsepower';
Proc sGplot Data = Group5.Auto_Mpg;
Histogram horsepower;
Density horsepower;
Density horsepower / type=kernel;
Run;


Proc Gchart Data = Group5.Auto_Mpg;
vbar horsepower;
Run;


Title 'QQ-Plot for Horsepower';
Proc Univariate Data = Group5.Auto_Mpg;
Var horsepower;
qqplot ;
Run;
```

```
/* Applying Log10 transformation on Horsepower */

Data Log_test;
Set Group5.Auto_Mpg;
LogHP = Log(horsepower);
Run;

Title 'Histogram of Horsepower after Log Transformation';
Proc sGplot Data = log_test;
Histogram loghp;
Density loghp;
Density loghp/ type=kernel;
Run;

Title 'QQ-Plot of Horsepower after Log Transformation';
Proc Univariate Data = log_test plots;
Var Loghp;
Run;

Title 'Listing First 5 Observations from Final Dataset';
Proc Print Data = group5.auto_mpg (obs = 5);
Run;

Data Group5.Auto_Mpg;
Set Group5.Auto_Mpg;
Label   Brand = 'Brand of the Vehicle'
        Model = 'Model name of vehicle'
        Cylinders = 'Number of Cylinders. Categorical Variable which can take following values:
                    4, 6 or 8'
        Year = 'The year in which the vehicle was manufactured'
        Origin = 'Country of Origin of the Vehicle Brand. Has the following categories:
                    Unites States  = 1
                    Germany =2
                    Japan = 3'
        MPG = 'City fuel cycle measured in miles/gallon'
        Displacement = 'Engine size of vehicle measured in cubic centimetres(CC)'
        Horsepower = 'Horsepower of the vehicle'
        Weight = 'Weight of vehicle in lbs'
        Acceleration = 'Time taken to reach from 0-60 mph'
        PWR = 'Power to weight ratio of vehicle measured as hp/lbs';
Run;

options label;
Proc Contents Data = Group5.Auto_Mpg;
ODS Select variables;
Run;

Proc sgplot data = group5.auto_mpg;
histogram mpg;
density mpg;
density mpg / type = kernel;
Run;

Proc sgplot data = group5.auto_mpg;
reg x = horsepower y = mpg  / cli clm;
Run;

Proc sgplot data = group5.auto_mpg;
reg x = weight y = mpg  / cli clm;
Run;
```

```
Proc sgplot data = group5.auto_mpg;
reg x = pwr y = mpg  / cli clm;
Run;
```

```
Proc sgplot data = group5.auto_mpg;
reg x = displacement y = mpg  / cli clm;
Run;
```

```
Proc Univariate Data = group5.auto_mpg plots;
```