```
                        /* DATA EXPLORATION AND PREPROCESSING */

/* Clearing log window */
DM "log; clear; ";

/* Data Loading */
PROC IMPORT OUT=Nutrition
    DATAFILE="/home/u59360783/Nutrition.csv"
    DBMS=CSV
    REPLACE;
    GETNAMES=YES;
RUN;

PROC IMPORT OUT=Region_mapping
    DATAFILE="/home/u59360783/us_regions.csv"
    DBMS=CSV
    REPLACE;
    GETNAMES=YES;
RUN;

/* Backing up master data  */
data Nutrition_bkup;
set Nutrition;
run;

/* Displaying few Observations to Confirm Import */
proc print data=Nutrition_bkup(obs=5); /* Display the first 5 observations */
run;

/* Knowing more about Data Structures such as data types, length of each features */
proc contents data=Nutrition_bkup_2;
run;

/* Counting entries and Missing values */
PROC MEANS DATA=Nutrition_bkup NMISS;
RUN;

/* Identifying Duplicate Records */
proc sort data=Nutrition_bkup
out=data_dup nodup;
by _all_;
run;

data Nutrition_bkup_1;
  set Nutrition_bkup;

  /* Create a new variable to store the income category names */
  length income_category $30;
  length Activity_Category $30;

  /* Categorize income ranges and assign names */
  if income = "Data not reported" then income_category = "Data not reported";
  else if income = "$75,000 or greater" then income_category = "High Income";
  else if income = "$50,000 - $74,999" then income_category = "Upper Middle Income";
  else if income = "$35,000 - $49,999" then income_category = "Middle Income";
  else if income = "$25,000 - $34,999" then income_category = "Lower Middle Income";
  else if income = "$15,000 - $24,999" then income_category = "Low Income";
  else if income = "Less than $15,000" then income_category = "Very Low Income";
  else income_category = "Unknown";

  /* Create a new variable "Category" based on column names */
  if upcase(Question) eq 'PERCENT OF ADULTS WHO ENGAGE IN NO LEISURE-TIME PHYSICAL ACTIVITY' then
    Activity_Category = 'Physical_Inactivity';
  else if upcase(Question) eq 'PERCENT OF ADULTS AGED 18 YEARS AND OLDER WHO HAVE OBESITY' then
    Activity_Category = 'Obesity';
  else if upcase(Question) eq 'PERCENT OF ADULTS AGED 18 YEARS AND OLDER WHO HAVE AN OVERWEIGHT CLASSIFICATION' then
    Activity_Category = 'Overweight';
  else if upcase(Question) eq 'PERCENT OF ADULTS WHO ACHIEVE AT LEAST 300 MINUTES A WEEK OF MODERATE-INTENSITY AEROBIC PHYSICAL ACTIVITY OR 150 MINUTES A WEEK OF VIGO
    Activity_Category = 'Physical_Activity_300min';
  else if upcase(Question) eq 'PERCENT OF ADULTS WHO ACHIEVE AT LEAST 150 MINUTES A WEEK OF MODERATE-INTENSITY AEROBIC PHYSICAL ACTIVITY OR 75 MINUTES A WEEK OF VIGOR
    Activity_Category = 'Physical_Activity_150min_and_Muscle_Strengthening';
  else if upcase(Question) eq 'PERCENT OF ADULTS WHO ACHIEVE AT LEAST 150 MINUTES A WEEK OF MODERATE-INTENSITY AEROBIC PHYSICAL ACTIVITY OR 75 MINUTES A WEEK OF VIGOR
    Activity_Category = 'Physical_Activity_150min';
  else if upcase(Question) eq 'PERCENT OF ADULTS WHO ENGAGE IN MUSCLE-STRENGTHENING ACTIVITIES ON 2 OR MORE DAYS A WEEK' then
    Activity_Category = 'Muscle_Strengthening';
  else if upcase(Question) eq 'PERCENT OF ADULTS WHO REPORT CONSUMING FRUIT LESS THAN ONE TIME DAILY' then
    Activity_Category = 'Fruit_Consumption';
  else if upcase(Question) eq 'PERCENT OF ADULTS WHO REPORT CONSUMING VEGETABLES LESS THAN ONE TIME DAILY' then
    Activity_Category = 'Vegetable_Consumption';
  else
    Activity_Category = 'Other';
run;

/* Display the result */
proc print data=Nutrition_bkup_1(obs=10);
run;

/* Initialize macro variable */
%let income_Category_values = ;
%let Activity_Category_values = ;

/* Use PROC SQL to get distinct values and store in the macro variable */
proc sql noprint;
  select distinct income_Category
  into :income_Category_values separated by ','
  from Nutrition_bkup_1;
  select distinct Activity_Category
  into :Activity_Category_values separated by ','
  from Nutrition_bkup_1;
quit;

/* Display the macro variable */
%put &income_Category_values;
%put &Activity_Category_values;

options validvarname=any;
data Nutrition_bkup_2(rename=("Age(years)"n=Age "Race/Ethnicity"n=Race_Ethnicity));
set Nutrition_bkup_1;
run;

/* Additional data processing */
data Nutrition_bkup_2;
format Gender $10.;
length Gender $10.;
format Age $10.;
```

```
length Age $10.;
set Nutrition_bkup_2;

Latitude = input(compress(tranwrd(scan(GeoLocation,1,','), '(', ''),' '), best12.);
Longitude = input(compress(tranwrd(scan(GeoLocation,2,','), ')', ''),' '), best12.);

Data_Value = Data_Value/100;
Data_Value_Alt = Data_Value_Alt/100;
Low_Confidence_Limit = Low_Confidence_Limit/100;
High_Confidence_Limit = High_Confidence_Limit/100;

/* Assign a default value if variable1 is blank or null */
if Age="" or Age="." then Age="Others";
if Gender="" then Gender="Others";
if Education="" then Education="Others";
if Race_Ethnicity="" then Race_Ethnicity="Others";
if Data_Value="" then Data_Value=0;
if Sample_Size="" then Sample_Size=0;
run;

proc freq data=Nutrition_bkup_2;
  tables Education*Race_Ethnicity / noprint out=DistinctValues(keep= Education Race_Ethnicity );
run;

/* Sorting data before joining two tables */
proc sort data=Nutrition_bkup_2 out=Nutrition_bkup_2;by LocationDesc;run;
proc sort data=Region_mapping out=Region_mapping;by State;run;

/* Keeping relevant features */
data Nutrition_bkup_3;
merge Nutrition_bkup_2 (rename=(LocationDesc=State YearStart=Year) IN=X) Region_mapping(IN=Y);
by State;
if X;

drop YearEnd Datasource Class Topic Data_Value_Unit Data_Value_Type Data_Value_Alt
     Data_Value_Footnote_Symbol Data_Value_Footnote Low_Confidence_Limit
     High_Confidence_Limit Total GeoLocation ClassID TopicID QuestionID
     DataValueTypeID LocationID StratificationCategoryId1 StratificationID1 LocationAbbr;
run;

/* Export final dataset to excel */
proc export data=Nutrition_bkup_3
  outfile='/home/u59360783/Nutrition_F.xlsx'
  dbms=xlsx
```