**Section 1 - Data Import**

The following code is used to import provided excel file to SAS:

SAS Code:

options validvarname=v7;

**proc** **import** out=life\_exp datafile='/home/u63038704/BAN110/Project/Life Expectancy Data.xlsx' dbms='xlsx';

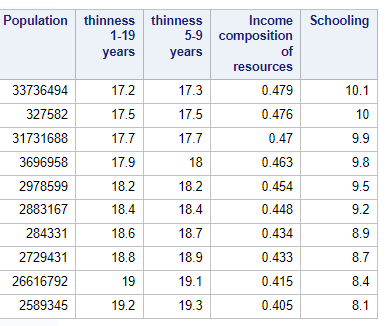
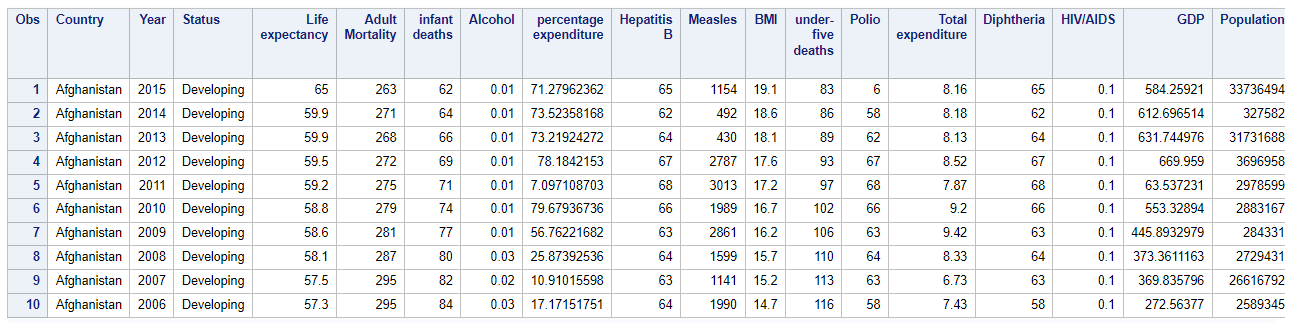
getnames=yes;

**run**;

**proc** **print** data=life\_exp (obs=**10**);

**run**;

Result:



**Section 2 - Data Cleaning**

**Description:**Based on the project instruction, the data cleaning process can be divided into several tasks:

The first task is to extract relevant data from the original dataset using PROC IMPORT and PROC CONTENTS.

The second task is to convert a numeric column to a character column or vice versa if necessary. The format of all variables is already correct.

The third task is to create a new column called "Avg\_Mortality\_Rate" based on existing columns such as Adult\_Mortality and infant\_deaths, and also calculate the percentage of people without HepatitisB vaccination. PROC MEANS is used to create the mean of mortality rate based on existing columns.

The fourth task is to identify missing values and replace them with an appropriate technique. PROC MEANS, PROC FORMAT, and PROC FREQ are used to identify missing values and replace them with the mean of each variable.

The fifth task is to use built-in SAS function(s) to perform data cleaning, such as extracting year from the data column.

The final task is to identify and deal with outliers in an appropriate manner using PROC UNIVARIATE.

**Code:**

\*1. Data Import

Import the data from the provided excel file into SAS using Proc Import;

FILENAME REFFILE '/home/u63055836/BAN110 project/Life Expectancy Data.xlsx';

options validvarname=v7;

**PROC** **IMPORT** DATAFILE=REFFILE DBMS=XLSX OUT=rawData;

GETNAMES=YES;

**RUN**;

\*2. Data Cleaning

2.1. Extract relevant data from the original dataset;

\* Before extract data, we made assumption that

any zero values from the columns are same as missing values;

**data** extractData;

set rawData (keep=country Year status Life\_expectancy Adult\_Mortality

infant\_deaths Hepatitis\_B gdp population percentage\_expenditure \_BMI

under\_five\_deaths Income\_composition\_of\_resources Schooling);

if Life\_expectancy=**0** then

Life\_expectancy=.;

if Adult\_Mortality=**0** then

Adult\_Mortality=.;

if infant\_deaths=**0** then

infant\_deaths=.;

if Hepatitis\_B=**0** then

Hepatitis\_B=.;

if gdp=**0** then

gdp=.;

if population=**0** then

population=.;

if percentage\_expenditure=**0** then

percentage\_expenditure=.;

if \_BMI=**0** then

\_BMI=.;

if under\_five\_deaths=**0** then

under\_five\_deaths=.;

if Income\_composition\_of\_resources=**0** then

Income\_composition\_of\_resources=.;

if Schooling=**0** then

Schooling=.;

**run**;

\*2.2. Convert a numeric column to character column or vice versa:

according to the result of PROC CONTENTS, format of all variables is correct;

**PROC** **CONTENTS** DATA=extractData;

**RUN**;

\*2.3. Create a new column based on existing columns and use it in your analysis

Creating a mean of mortality rate based on existing columns and number percentage of without HepatitisB vaccination

New columns will be created after data cleaning process;

\*2.4. Identify missing values and remove / replace using an appropriate technique;

**proc** **means** data=extractData NMISS mean;

**run**;

**proc** **format**;

value $missfmt ' '='Missing' other='Not Missing';

**run**;

**proc** **freq** data=extractData;

format \_CHAR\_ $missfmt.;

tables \_CHAR\_ / missing missprint nocum nopercent;

**run**;

\* character values do not have missings;

\* find mean for each country;

**proc** **means** data=extractData n mean std;

class country;

output out=temp(drop=\_type\_ \_freq\_) mean=std= /autoname;

**run**;

\* The output data still contains missing value

since some countries' whole column data is missing;

**proc** **means** data=temp nmiss;

**run**;

\* So we replace those missing values by overall mean;

**proc** **standard** data=temp replace out=Mean\_Std;

**run**;

**proc** **means** data=Mean\_Std nmiss;

**run**;

\* replacing missing values with mean of each variable by country;

**proc** **sort** data=extractData out=sortE;

by country;

**run**;

**proc** **sort** data=Mean\_Std out=sortM;

by country;

**run**;

**data** noMissing;

merge sortE sortM;

by country;

if Life\_expectancy=. then

Life\_expectancy=Life\_expectancy\_mean;

if Adult\_Mortality=. then

Adult\_Mortality=Adult\_Mortality\_mean;

if infant\_deaths=. then

infant\_deaths=infant\_deaths\_mean;

if Hepatitis\_B=. then

Hepatitis\_B=Hepatitis\_B\_mean;

if gdp=. then

gdp=gdp\_mean;

if population=. then

population=population\_mean;

if percentage\_expenditure=. then

percentage\_expenditure=percentage\_expenditure\_mean;

if \_BMI=. then

\_BMI=\_BMI\_mean;

if under\_five\_deaths=. then

under\_five\_deaths=under\_five\_deaths\_mean;

if Income\_composition\_of\_resources=. then

Income\_composition\_of\_resources=Income\_composition\_of\_res\_mean;

if Schooling=. then

Schooling=Schooling\_mean;

if country ne '';

keep country year status Life\_expectancy Adult\_Mortality infant\_deaths Hepatitis\_B

gdp population percentage\_expenditure \_BMI under\_five\_deaths

Income\_composition\_of\_resources Schooling;

**run**;

**proc** **means** data=noMissing NMISS mean;

**run**;

\*2.5. Use built-in SAS function(s) to perform data cleaning;

\* Identify if data contains any duplicates by setting country and year as primary key;

**data** findDuplicate;

set noMissing;

PK = compress(cat(country, year),' ');

**run**;

**proc** **sort** data=findDuplicate out=demo;

by PK;

**run**;

**data** duplicateL;

set demo;

by PK;

if first.PK=**0** or last.PK=**0** then output;

**run**;

**proc** **print** data=duplicateL;

**run**;

\* No result is shown from PROC PRINT, there has no duplicate for this dataset;

\*2.6. Identify outliers and deal with them in an appropriate manner;

\*observing outliers using proc univariate;

**proc** **univariate** data=noMissing;

histogram / normal;

**run**;

\* deleting outliers of some important columns;

**proc** **means** data=noMissing;

var Life\_expectancy Adult\_Mortality gdp population;

output out=Mean\_Std1(drop=\_type\_ \_freq\_) mean=std= /autoname;

**run**;

**data** lifeExpentancy\_clean;

set noMissing;

if \_n\_=**1** then

set Mean\_Std1;

if Life\_expectancy\_Mean - **2**\*Life\_expectancy\_StdDev<=Life\_expectancy <=Life\_expectancy\_Mean + **2**\*Life\_expectancy\_StdDev;

if Adult\_Mortality\_Mean - **2**\*Adult\_Mortality\_StdDev<=Adult\_Mortality <=Adult\_Mortality\_Mean + **2**\*Adult\_Mortality\_StdDev;

if gdp\_Mean - **2**\*gdp\_StdDev<=gdp <=gdp\_Mean + **2**\*gdp\_StdDev;

if population\_Mean - **2**\*population\_StdDev<=population <=population\_Mean + **2**\*population\_StdDev;

\* gengerate new columns for futher analysis;

Avg\_Mortality\_Rate=((Adult\_Mortality/**10**)+(Infant\_deaths/**10**))/**2**;

Nonvaccine\_Hepatit\_B=**100**-Hepatitis\_B;

drop Life\_expectancy\_Mean Life\_expectancy\_StdDev Adult\_Mortality\_Mean

Adult\_Mortality\_StdDev gdp\_Mean gdp\_StdDev population\_Mean population\_StdDev;

**run**;

**proc** **univariate** data=lifeExpentancy\_clean;

histogram / normal;

**run**;

**Section 3 - Joining and Merging**

In order to merge existing dataset (Life Expectancy Data) with another dataset, the ‘Suicide’ dataset has been selected. In this dataset, the number and rate of suicide for different Age and Gender groups in 99 different counties are provided for years between 1985 to 2016. This dataset also contains other information about countries like population, Gross domestic product (GDP) and Human development index (HDI). The dataset is available in following link:

<https://www.kaggle.com/datasets/russellyates88/suicide-rates-overview-1985-to-2016>

The following steps are implemented to have two dataset prepared for merging:

**Importing Suicide Dataset:**

SAS Code:

**proc** **import** out=suicide (rename =(**'country-year'n** = country\_year)) datafile='/home/u63038704/BAN110/Project/Suicide Rate Data.csv' dbms=csv Replace;

getnames=yes;

guessingrows=**28000**;

**run**;

**proc** **print** data=suicide (obs=**10**);

**run**;

Result:Graphical user interface

Description automatically generated

Note: The proc import by default scans the first 20 rows to determine the appropriate data type and length of variables. The number of rows can be adjusted using “guessingrows” statement. Without using this statement Country variable length would be set 7, which name of many countries in this variable have more than 7 character in their names.

**Aggregation of variables:**

The suicide dataset provide information for two gender groups and 6 age groups which mean 12 numbers for each year for any country; however, based on “Life Expectancy” data set only the one number for each year for any country is required. Therefore, the detail information to be aggregated to have a single number. There is a variable in Suicide dataset which is concatenation of country name and its related year. This variable is unique for each country in each year and is used for aggregation purpose. After summation and calculating of total suicide rate, the variables of “sex”, “age”, “suicides\_no”, “population”, “HDI for year”, “generation” and “suicides/100k pop” are not required any more. Meanwhile, variables of “gdp\_for\_year” and “pupolation\_total” are provided in Life Expectancy dataset. Therefore, all of these variables are dropped before merging of two datasets.

**proc** **sort** data=suicide out=suicide\_sorted;

by country\_year;

**run**;

**data** suicide\_clean;

set suicide\_sorted;

format suicide\_total **10.** population\_total comma14.0 suicide\_per\_100K\_pop **5.2**;

by country\_year;

if first.country\_year then do;

suicide\_total=**0**;

population\_total =**0**;

end;

suicide\_total + suicides\_no;

population\_total + population;

suicide\_per\_100K\_pop = (suicide\_total/population\_total)\***100000**;

if last.country\_year;

drop sex age suicides\_no population population\_total generation **'HDI for year'n** **'suicides/100k pop'n** **' gdp\_for\_year ($) 'n** **'gdp\_per\_capita ($) 'n**;

**run**;

**proc** **print** data=suicide\_clean (obs=**10**);

**run**;Table

Description automatically generated

**Preparing Life Expectancy Dataset and Joining:**

The ‘Country’ variable can not be used for merging because the name of each country is repeated for different years. Therefore, a new variable in Life Expectancy dataset is defined by concatenating of country name and its related year. This variable which already exists in suicide dataset is used for merging of two dataset.

**data** lifeExpentancy\_country\_year;

set lifeExpentancy\_clean;

country\_year=cat(strip(country),year);

**run**;

**proc** **sort** data=lifeExpentancy\_country\_year out=lifeExpentancy\_sorted;

by country\_year;

**run**;

**data** lifeExpentancy\_final;

merge lifeExpentancy\_sorted(in=lifeExp) suicide\_clean(in=suic);

by country\_year;

if lifeExp and suic;

**run**;

Table

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

**Section 4 - Data Analysis**