

|  |  |  |  |
| --- | --- | --- | --- |
| Academic Year | 2023 | | |
| Semester | Fall | Winter | Summer |
| Course Code - Name | BAN110 | | |
| Instructor | Dr. Razi Iqbal | | |
| Assessment | Projects |  | |

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student Name** | **Role** |
| 130022221 | Sepehr Salehi | Data Cleaning - Outliers |
| 176938215 | Rongzhao Yi | Data Cleaning - Missing value |
| 129620225 | Roohalah Taraf | Merging |
| 129012225 | Yash Deveshbhai Sheth | Business Questions |
| 128023223 | Zil Chetan Sheth | Business Questions |

**Projects**

You are required to choose a project from the list of the projects specified in this document and complete it within groups of max. 5.

Since, this is a group project, it is required to be done in groups of maximum 5. In exceptional cases, there can be 1 member in a group. Each group should have a Group Lead who would be responsible for submitting the project on Blackboard (Please note that not all the members of the group are required to submit the project separately on Blackboard. One submission from the Group Lead would be sufficient).

The detailed requirements for each project are available in this document, so please go through the details and fulfil all the requirements to avoid missing any marks.

Finally, follow the below mentioned instructions carefully.

**Instructions:**

In order to obtain maximum marks in this assessment, please ensure the followings:

* Don’t forget to write your name and ID on the first page of this document. The student IDs and names of all the students in the group should be mentioned along with the roles.
* Submit the project by writing your solution in this document under the Solution heading below. Do not use a separate document. Everything related to the project should be included in this document, e.g., code, screenshots and etc.
* This project has a weightage of **25%** marks of the course.
* This is a group project so **only 1 submission from the group lead is required.**
* Group Leads are required to submit the project on Blackboard as instructed. Submissions through emails will not be accepted.
* The project deadline is **midnight April 9, 2023**. Submissions after the deadline will not be accepted.
* A separate session for presentation and QA for the project will be scheduled.

**Rubric:**

Your assessment will be graded based on the following rubric:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Excellent (7 - 10)** | **Average (4 – 6.9)** | **Poor (<4)** |
| **Project Completion and Code**  **(10)** | The project was completed without any errors and output is as expected. Fulfills all/most of the requirements for the project. | The project was completed with few errors. Fulfills some of the requirements for the project. | The project is incomplete. Does not fulfill all/most of the requirements. |
| **Presentation and QA**  **(10)** | The student has a good contribution to the project. Knows ins and outs of the project.  The student has presented his/her part of the project very well. Knows everything / most of his/her part. | The student has average contribution to the project. Does not know the whole project.  The student has averagely presented his/her part of the project. Knows few of the things about his/her part. | The student has no contribution to the project. Does not know anything / most about the project.  The student has poorly presented the project. Does not know much about the project. |
| **Report**  **(5)** | Student has contributed well in preparing the project report and knows all the aspects of the report. | Student has contributed partially in preparing the project report and knows some aspects of the report. | Student has not contributed in preparing the report. |

**Project Instructions**

You are provided with few datasets on Blackboard under Project Datasets Folder. You are free to pick any dataset you like to work on as a group. You are required to demonstrate at least the following skills in the project:

1. Data Import
   * This phase requires you to import the data from the provided excel file into SAS using Proc Import.
2. Data Cleaning
   * This phase requires you to clean your data before data analysis phase. You should use at least following concepts to complete this phase:
     1. Extract relevant data from the original dataset
     2. Convert a numeric column to character column or vice versa
     3. Create a new column based on existing columns and use it in your analysis
     4. Identify missing values and remove / replace using an appropriate technique
     5. Use built-in SAS function(s) to perform data cleaning, e.g., extracting year from the data column etc.
     6. Identify outliers and deal with them in an appropriate manner
3. Joining and Merging
   * This phase requires you to join / merge your datasets to create a dataset for analysis.
     1. Merge / Join your existing dataset with another dataset. If your Excel file has only 1 sheet, find some data over the Internet that you can merge your existing dataset with.
     2. Make sure the merge in Step 1 is meaningful
     3. Use this merge in your data analysis
4. Data Analysis
   * This phase requires you to analyze your cleaned dataset to answer at least 3 valid business questions. You are free to pick any business questions you like, however, please keep in mind that picking good business questions to answer would result in better marks.
5. Project Report
   * This phase requires you to create a report in MS Word with the following requirements:
     1. Explain each and every phase of the project (from Phase 1 to 4) along with the screenshots of the output and the related SAS code
     2. Include answers to questions in Phase 4 in your report
     3. Create at least 1 graph / chart in your report which can be simply a Box plot to identify outliers etc.
     4. Make sure not to miss any phase and output of its screenshot

**Solution**

# Section 1 - Data Import

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

FILENAME REFFILE '/home/u63025740/BAN110/Life Expectancy Data (2).xlsx';

options validvarname=v7;

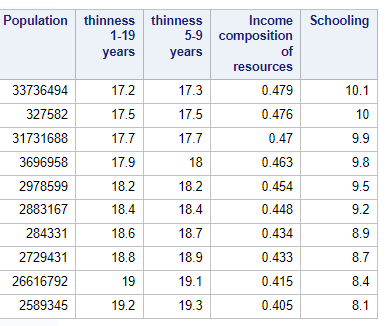
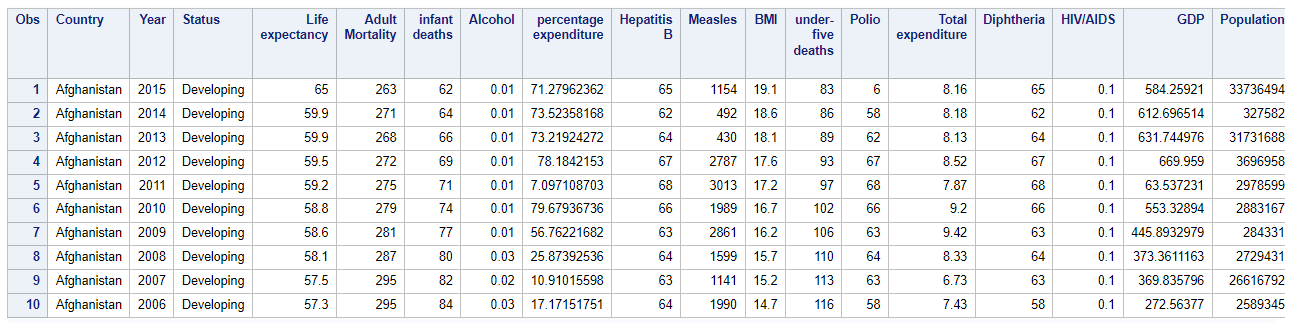
**PROC** **IMPORT** DATAFILE=REFFILE DBMS=XLSX OUT=rawData;

GETNAMES=YES;

**RUN**;

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

**run**;



# Section 2 - Data Cleaning

# Description:

Extract relevant data from the original dataset:

The code reads in a raw dataset named "rawData" and extracts specific columns using a "keep" statement. The extracted columns are then saved to a new dataset named "extractData."

Convert a numeric column to a character column or vice versa:

The code uses PROC CONTENTS to check the format of all variables in the "extractData" dataset. No changes are needed since the format of all variables is correct.

Create a new column based on existing columns and use it in your analysis:

The code creates a new column named "Mortality\_rate" by taking the ratio of "Adult\_Mortality" to "Life\_expectancy" multiplied by 1000. The code also creates a new column named "without\_HepB\_vacc" by taking the ratio of "Hepatitis\_B" to 100. The code then calculates the mean of Mortality\_rate and without\_HepB\_vacc based on each country and saves the output to a temporary dataset named "temp."

Identify missing values and remove / replace using an appropriate technique:

The code uses PROC MEANS to calculate the number of missing values for each variable in the "extractData" dataset. Then, it uses PROC FREQ to identify the number of missing values in each character variable. The code replaces the missing values in the "extractData" dataset with the mean of each variable by country using PROC STANDARD and a merge statement. The final dataset with no missing values is saved to a new dataset named "noMissing."

Use built-in SAS function(s) to perform data cleaning:

The code checks for duplicates in the "noMissing" dataset by creating a new variable named "PK" that concatenates "country" and "year" variables. It then sorts the dataset by PK and removes duplicates using PROC SORT and a data step.

Identify outliers and deal with them in an appropriate manner:

The code uses PROC UNIVARIATE to create histograms of several variables and identifies outliers. It then removes the outliers by creating a new dataset named "lifeExpentancy\_clean" that only includes observations where Life\_expectancy, Adult\_Mortality, GDP, and population are within two standard deviations of their respective means.

\*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 Total\_expenditure);

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=.;

if Total\_expenditure=**0** then

Total\_expenditure=.;

**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**;

Table

Description automatically generated

\*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**;

Graphical user interface, application

Description automatically generated

**proc** **format**;

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

**run**;

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

format \_CHAR\_ $missfmt.;

tables \_CHAR\_ / missing missprint nocum nopercent;

Table

Description automatically generated**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;

Graphical user interface

Description automatically generated**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**;

A picture containing graphical user interface

Description automatically generated

\* 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 Total\_expenditure=. then

Total\_expenditure=Total\_expenditure;

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 Total\_expenditure;

**run**;

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

**run**;

Graphical user interface, application

Description automatically generated

\*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;

\* generating new columns for further analysis;

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:

**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**);

Graphical user interface

Description automatically generated**run**;

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**

Section 4, Question 1:

The first question is about launching a new drug in a country with the highest suicide rates. The code first calculates the maximum value of the "suicide\_per\_100K\_pop" variable from the "lifeExpentancy\_final" dataset using the proc means procedure and stores the result in a new dataset "Maximum." It then filters out the countries with the maximum "suicide\_per\_100K\_pop" value and stores only their names in a new dataset "lifeExpentancy\_final3" using the data step. Finally, the code prints the list of countries in "lifeExpentancy\_final3" using the proc print procedure. The same task is achieved using an SQL query that filters out the country with the maximum sum of "suicide\_per\_100K\_pop" from the "lifeExpentancy\_final" dataset.

Section 4, Question 2:

The second question is about finding the potential market for a vaccine for Hepatitis B. The code first calculates the absolute amount of GDP per capita a country is willing to spend on health using the formula: GDP\*Total\_expenditure/100. It then multiplies the absolute amount of GDP per capita with the percentage of the latest year-olds who have not taken the Hepatitis B vaccine to get the budget a country is willing to spend on this vaccine. The results are stored in a new dataset "potential\_countries." Finally, the list of potential countries is sorted by the budget in descending order using the proc sort procedure.

Section 4, Question 3:

The third question is about building a model to predict life expectancy in different countries based on various variables. The code uses the proc reg procedure to perform a backward selection linear regression with "Life\_expectancy" as the dependent variable and "Adult\_Mortality," "infant\_deaths," "Hepatitis\_B," "gdp," "population," "percentage\_expenditure," "under\_five\_deaths," "Income\_composition\_of\_resources," "Schooling," "Total\_expenditure," and "year" as independent variables. The regression coefficients of the variables are estimated, and the model is built for predicting life expectancy in different countries.

\* section 4

question1

We are a pharmaceutical company and we have launched a new drug that is supposed to help

with depression, anxiety, and other disorders and we want to launch this drug into a

new country so we use this data to get the suicide rates per 100k and launch

the drug in the country with highest suicide rates.;

\*SAS solution;

**proc** **means** data=lifeExpentancy\_final max;

var suicide\_per\_100K\_pop;

output out=Maximum(drop=\_type\_ \_freq\_) max= /autoname;

**run**;

**proc** **print** data=Maximum;

**run**;

Graphical user interface, text, application, Word

Description automatically generated

**data** lifeExpentancy\_final3 (keep=country) ;

set lifeExpentancy\_final;

if \_n\_= **1** then set Maximum;

if suicide\_per\_100K\_pop=suicide\_per\_100K\_pop\_Max;

**run**;

**proc** **print** noobs;

Graphical user interface, application, Word

Description automatically generated**run**;

\*SQL solution;

**proc** **sql**;

select country from lifeExpentancy\_final

where Status='Developed' and year=**2015** group by country HAVING

SUM(suicide\_per\_100K\_pop)=(SELECT MAX(sum) FROM (SELECT country,

SUM(suicide\_per\_100K\_pop) AS sum FROM lifeExpentancy\_final WHERE

Status='Developed' AND year=**2015** GROUP BY country));

**quit**;

Graphical user interface, application

Description automatically generated

\*question2

We are a pharmaceutical company, looking to expand our vaccine for Hepatitis B into the international market.

First, we get the number of latest year -olds (%) who have not taken the Hepatitis B (HepB) vaccine.

Then we multiply GDP per capita with the GDP % they are willing to spend on health.

This will give us the absolute amount of GDP per capita they are willing to spend on health.

Next, we multiply the number of unvaccinated children by the absolute amount of GDP per capita they are willing to spend on health.

This new variable created will help us know the biggest potential market that we should tap.;

**data** potential\_countries (keep= country Nonvaccine\_Hepatit\_B SpendOnHealth budget);

set lifeExpentancy\_final;

SpendOnHealth= GDP\*Total\_expenditure/**100** ;

budget= SpendOnHealth\*Nonvaccine\_Hepatit\_B/**100** ;

where year=**2014**;

format budget Dollar9.3

SpendOnHealth Dollar9.3;

**run**;

**proc** **sort** data=potential\_countries;

by descending budget;

**run**;

**proc** **print** data= potential\_countries (obs=**10**);

**run**;

Table

Description automatically generated

\*question3

as a pharmacy company partnering with WHO we want to build a model to calculate

the coefficient of each variable on life expectancy and build a model to be able

to predict life expectancy of different countries;

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

model Life\_expectancy=Adult\_Mortality infant\_deaths Hepatitis\_B gdp

population percentage\_expenditure under\_five\_deaths

Income\_composition\_of\_resources Schooling Total\_expenditure year/

selection= backward;

**run**;

Table

Description automatically generated

**Diagram

Description automatically generated**

Graphical user interface, diagram

Description automatically generated