BAN110: Data preparation and Handling

Group Project Report: Suicide Rate dataset

BAN110: Data preparation and Handling

Introduction

The dataset we used is 'Suicide Rate'. It is a suicides survey data conducted from 1985 to 2016. This is compiled dataset pulled from four other datasets linked by time and place, and was built to find signals correlated to increased suicide rates among different cohorts globally, across the socio-economic spectrum. This project will help us to find patterns and gain some insights on the Suicide Rates, and hopefully help

We used SAS software to perform all the operations.

Dataset and task description:

Variable	Description	Type	Length
country	Country from which data was collected	Char	7
year	Year in which data was collected	Num	8
sex	Gender of the person committing suicide	Char	6
age	Age of the person committing suicide	Char	11
suicides_no	Number of Suicides	Num	8
population	Total number of People in that country	Num	8
suicidesPer100k_pop	Number of suicides per 100k people	Num	8
country_year	Country and Year	Char	11
HDI_for_year	Human Development Index, growth of the people to measure overall growth of country	Num	8

BAN110: Data preparation and Handling

gdp_per_capita_dollars	GDP per capita in dollars	Num	8
Gdp_for_year_dollars	GDP per year in dollars	Char	15
generation	Generation (Gen X, Y, Z, etc)	Char	15

Loading Data:

```
libname clean '~/BAN110/Project';

/* Data Import  */
proc import datafile="~/BAN110/Project/master2.csv"
out=clean.suicideRates
dbms=CSV replace;
guessingrows=max;
run;

proc print data=clean.suiciderates(obs=50);
run;
```

```
/* Column type conversion*/
data clean.suicideRates;
set clean.suicideRates;

instant = _n_;

gdp_for_year_dollars1 = input(gdp_for_year_dollars,comma15.);
drop gdp_for_year_dollars;
rename gdp_for_year_dollars1 = gdp_for_year_dollars;
run;
```

Obs	country	year	sex	age	suicides_no	population	suicidesPer100k_pop	country_year	HDI_for_year	gdp_per_capita_dollars	generation	instant	gdp_for_year_dollars
1	Albania	1987	male	15-24 years	21	312900	6.71	Albania1987	.	796	Generation X	1	2156624900
2	Albania	1987	male	35-54 years	16	308000	5.19	Albania1987	.	796	Silent	2	2156624900
3	Albania	1987	female	15-24 years	14	289700	4.83	Albania1987	.	796	Generation X	3	2156624900
4	Albania	1987	male	75+ years	1	21800	4.59	Albania1987	.	796	G.I. Generation	4	2156624900
5	Albania	1987	male	25-34 years	9	274300	3.28	Albania1987	.	796	Boomers	5	2156624900
6	Albania	1987	female	75+ years	1	35600	2.81	Albania1987	.	796	G.I. Generation	6	2156624900
7	Albania	1987	female	35-54 years	6	278800	2.15	Albania1987	.	796	Silent	7	2156624900
8	Albania	1987	female	25-34 years	4	257200	1.56	Albania1987	.	796	Boomers	8	2156624900
9	Albania	1987	male	55-74 years	1	137500	0.73	Albania1987	.	796	G.I. Generation	9	2156624900
10	Albania	1987	female	5-14 years	0	311000	0	Albania1987	.	796	Generation X	10	2156624900

We used the above code to import the data from the csv file. Also, we introduced a new variable ‘instant’, which you can see in the output image. This new variable will act as a unique row identifier, similar to an Id or Row Id. ‘Instant’ variable was very crucial to perform further calculations in this project.

Checking and Correcting Errors:

BAN110: Data preparation and Handling

```
title 'List of Values and their Frequency Table for Categorical Variables';
proc freq data=clean.suicideRates;
tables _character_ / missing nocum;
run;
```

sex	Frequency	Percent
female	13910	50.00
male	13910	50.00

age	Frequency	Percent
15-24 years	4642	16.69
25-34 years	4642	16.69
35-54 years	4642	16.69
5-14 years	4610	16.57
55-74 years	4642	16.69
75+ years	4642	16.69

generation	Frequency	Percent
Boomers	4990	17.94
G.I. Generation	2744	9.86
Generation X	6408	23.03
Generation Z	1470	5.28
Millenials	5844	21.01
Silent	6364	22.88

country	Frequency	Percent
Albania	264	0.95
Antigua and Barbuda	324	1.16
Argentina	372	1.34
Armenia	298	1.07
Aruba	168	0.60

country_year	Frequency	Percent
Albania1987	12	0.04
Albania1988	12	0.04
Albania1989	12	0.04
Albania1992	12	0.04
Albania1993	12	0.04

```
title 'List of Numerical Variables';
proc means data=clean.suicideRates n min max mean nmiss q1 q3 range;
run;
```

List of Numerical Variables

The MEANS Procedure

Variable	N	Minimum	Maximum	Mean	N Miss	Lower Quartile	Upper Quartile	Range
year	27820	1985.00	2016.00	2001.26	0	1995.00	2008.00	31.0000000
suicides_no	27820	0	22338.00	242.5744069	0	3.0000000	131.0000000	22338.00
population	27820	278.0000000	43805214.00	1844793.62	0	97497.00	1486195.50	43804936.00
suicidesPer100k_pop	27820	0	224.9700000	12.8160974	0	0.9200000	16.6200000	224.9700000
HDI_for_year	8364	0.4830000	0.9440000	0.7766011	19456	0.7130000	0.8550000	0.4610000
gdp_per_capita_dollars	27820	251.0000000	126352.00	16866.46	0	3447.00	24874.00	126101.00
instant	27820	1.0000000	27820.00	13910.50	0	6955.50	20865.50	27819.00
gdp_for_year_dollars	27820	46919625.00	989930542279	114115739402	0	8985352832	136631966609	989883622654

With the help of above code, we were successfully able to display the values and their frequencies for both Character and Numeric Variables.

We have included all the important parameters to check numerical variables such as Minimum, Maximum, Mean, Number of Missing Values (N Miss), Lower Quartile (Q1), Upper Quartile (Q3) and Range.

```
title "Checking Missing Character Values";
    proc format;
        value $Count_Missing ' '  = 'Missing'
                          other = 'Nonmissing';
    run;

proc freq data=clean.suicideRates;
       tables _character_ / nocum missing;
       format _character_ $Count_Missing.;
run;
```

Checking Missing Character Values

The FREQ Procedure

country	Frequency	Percent
Nonmissing	27820	100.00

sex	Frequency	Percent
Nonmissing	27820	100.00

age	Frequency	Percent
Nonmissing	27820	100.00

Checking Missing Numeric Values

The FREQ Procedure

year	Frequency	Percent
Nonmissing	27820	100.00

suicides_no	Frequency	Percent
Nonmissing	27820	100.00

population	Frequency	Percent
Nonmissing	27820	100.00

BAN110: Data preparation and Handling

```
title "Checking Missing Numeric Values";
    proc format;
       value Count_Missing .  = 'Missing'
                     other = 'Nonmissing';
    run;

proc freq data=clean.suicideRates;
       tables _numeric_ / nocum missing;
       format _numeric_ Count_Missing.;
run;
```

In order to check the missing values, we used format procedure. We can see the major difference in the code of checking missing values for Character and Numeric was the $ sign.

$ is used for Character variables, whereas it is not required in Numeric variables.

So, from the outputs we can see that character variables don't have any missing values.

However, the Numeric variable 'HDI_for_Year' has 19456 Missing values.

BAN110: Data preparation and Handling

```
title "Correcting Errors by Deletion (Character Variables)";
data clean.suicideRatesErrors;
 set clean.suicideRates;
 if cmiss(of _all_) then delete;
run;

proc format;
      value $Count_Missing ' '  = 'Missing'
                     other = 'Nonmissing';
run;

proc freq data=clean.suicideRatesErrors;
      tables _character_ / nocum missing;
      format _character_ $Count_Missing.;
run;

title "Correcting Errors by Deletion (Numeric Variables)";
data missing_delete;
 set clean.suicideRates;
 if HDI_for_year=. then delete;
run;
proc format;
      value Count_Missing .  = 'Missing'
                     other = 'Nonmissing';
    run;

proc freq data=missing_delete;
      tables _numeric_ / nocum missing;
      format _numeric_ Count_Missing.;
run;
```

Correcting Errors by Deletion (Character Variables)

The FREQ Procedure

country	Frequency	Percent
Nonmissing	8364	100.00

sex	Frequency	Percent
Nonmissing	8364	100.00

age	Frequency	Percent
Nonmissing	8364	100.00

country_year	Frequency	Percent
Nonmissing	8364	100.00

generation	Frequency	Percent
Nonmissing	8364	100.00

Correcting Errors by Deletion (Numeric Variables)

The FREQ Procedure

year	Frequency	Percent
Nonmissing	8364	100.00

suicides_no	Frequency	Percent
Nonmissing	8364	100.00

population	Frequency	Percent
Nonmissing	8364	100.00

suicidesPer100k_pop	Frequency	Percent
Nonmissing	8364	100.00

HDI_for_year	Frequency	Percent
Nonmissing	8364	100.00

gdp_per_capita_dollars	Frequency	Percent
Nonmissing	8364	100.00

instant	Frequency	Percent
Nonmissing	8364	100.00

gdp_for_year_dollars	Frequency	Percent
Nonmissing	8364	100.00

To correct the errors by deletion of character variables, we used cmiss in the if statement to check for missing character values and then delete. Later, we used format statement in proc freq to display any other missing values present in the dataset. However, as it is evident in the output that we didn't find any missing values in the character variables.

To correct the errors by deletion of numeric variables, we used the if statement to check for missing values in the variable HDI_for_Year and then delete. Later, we used format statement in proc freq to display any other missing values present in the dataset. However, as it is evident in the output that we deleted the existing missing values and further there are no missing values in the dataset.

It is important to note that deleting such a huge amount of missing values from a dataset can cause data inaccuracy in the final result.

BAN110: Data preparation and Handling

```
title "Extreme Observations Table";
ods select ExtremeObs;
proc univariate data=missing_delete nextrobs=10;
id instant;
var suicides_no suicidesPer100k_pop;
histogram / normal;
run;
```

Extreme Observations Table

The UNIVARIATE Procedure
Variable: suicides_no

Extreme Observations					
Lowest			Highest		
Value	instant	Obs	Value	instant	Obs
0	27544	8280	8445	27186	8150
0	27328	8208	8545	26972	8080
0	26548	7956	8961	27198	8162
0	26476	7932	9263	27030	8090
0	26475	7931	10332	27090	8102
0	26474	7930	11396	27187	8151
0	26416	7920	11455	27199	8163
0	26415	7919	11681	27162	8126
0	26414	7918	11763	27174	8138
0	26413	7917	11767	27150	8114

Extreme Observations Table

The UNIVARIATE Procedure
Variable: suicidesPer100k_pop

Extreme Observations					
Lowest			Highest		
Value	instant	Obs	Value	instant	Obs
0	27544	8280	124.95	1907	421
0	27328	8208	125.22	23901	7189
0	26548	7956	125.46	8355	2545
0	26476	7932	131.17	4545	1249
0	26475	7931	131.90	15105	4669
0	26474	7930	141.91	15046	4658
0	26416	7920	144.15	11473	3481
0	26415	7919	144.85	15045	4657
0	26414	7918	165.96	11413	3469
0	26413	7917	187.06	24333	7249

We used an extreme observations table to check the highest and lowest values in suicides_no and suicidesPer100k_pop variables. We can see that the highest value in suicides_no is **11767** and the lowest value is **0**. Similarly, in the suicidesPer100k_pop, the highest value is **187.05** and the lowest value is **0**.

BAN110: Data preparation and Handling

```
title "Ten Highest and Lowest Values for suicidesPer100k_pop";
proc sort data=missing_delete
out=clean.Tmp;
by suicidesPer100k_pop;
run;
data missing_delete1;
if 0 then set clean.Tmp nobs=Number_of_Obs;
High = Number_of_Obs - 9;
call symputx('High_Cutoff',High);
stop;
run;
data missing_delete1;
set clean.Tmp(obs=10) /* 10 lowest values */
clean.Tmp(firstobs=&High_Cutoff); /* 10 highest values */
file print;
if _n_ le 10 then do;
if _n_ = 1 then put / "Ten Lowest Values";
put "Instant = " instant @16 "Value = " suicidesPer100k_pop;
end;
else if _n_ ge 11 then do;
if _n_ = 11 then put / "10 Highest Values";
put "Instant = " instant @18 "Value = " suicidesPer100k_pop;
end;
run;
```

Ten Highest and Lowest Values for suicidesPer100k_pop

```
Ten Lowest Values
Instant = 143  Value = 0
Instant = 144  Value = 0
Instant = 193  Value = 0
Instant = 194  Value = 0
Instant = 195  Value = 0
Instant = 196  Value = 0
Instant = 197  Value = 0
Instant = 198  Value = 0
Instant = 199  Value = 0
Instant = 200  Value = 0
10 Highest Values
Instant = 1907   Value = 124.95
Instant = 23901  Value = 125.22
Instant = 8355   Value = 125.46
Instant = 4545   Value = 131.17
Instant = 15105  Value = 131.9
Instant = 15046  Value = 141.91
Instant = 11473  Value = 144.15
Instant = 15045  Value = 144.85
Instant = 11413  Value = 165.96
Instant = 24333  Value = 187.06
```

```
title "Ten Highest and Lowest Values for suicides_no";
proc sort data=missing_delete
out=clean.Tmp;
by suicides_no;
run;
data missing_delete2;
if 0 then set clean.Tmp nobs=Number_of_Obs;
High = Number_of_Obs - 9;
call symputx('High_Cutoff',High);
stop;
run;
data missing_delete2;
set clean.Tmp(obs=10) /* 10 lowest values */
clean.Tmp(firstobs=&High_Cutoff); /* 10 highest values */
file print;
if _n_ le 10 then do;
if _n_ = 1 then put / "Ten Lowest Values";
put "Instant = " instant @16 "Value = " suicides_no;
end;
else if _n_ ge 11 then do;
if _n_ = 11 then put / "10 Highest Values";
put "Instant = " instant @18 "Value = " suicides_no;
end;
run;
```

Ten Highest and Lowest Values for suicides_no

```
Ten Lowest Values
Instant = 143  Value = 0
Instant = 144  Value = 0
Instant = 193  Value = 0
Instant = 194  Value = 0
Instant = 195  Value = 0
Instant = 196  Value = 0
Instant = 197  Value = 0
Instant = 198  Value = 0
Instant = 199  Value = 0
Instant = 200  Value = 0
10 Highest Values
Instant = 27186  Value = 8445
Instant = 26972  Value = 8545
Instant = 27198  Value = 8961
Instant = 27030  Value = 9263
Instant = 27090  Value = 10332
Instant = 27187  Value = 11396
Instant = 27199  Value = 11455
Instant = 27162  Value = 11681
Instant = 27174  Value = 11763
Instant = 27150  Value = 11767
```

Just to confirm the accuracy of our extreme observation table, we performed top 10 highest and lowest values for both suicides_no and suicidesPer100k_pop variables.
It is clear in the output that the results we got in the top 10 highest and lowest values are the same as the values in our extreme observations table.

BAN110: Data preparation and Handling

Handling and Treating the missing Values:

Missing values frequently encountered while collecting the data. So here in this dataset we checked for missing values by using the proc freq function.

To check the missing values in the present dataset we used the code as shown below and we get the output that only the HDI_for_year variable has missing values. All other variables have no missing values.

HDI_for_year has 19456 frequency with almost 69.94% missing values in the dataset. Code and output of this is as follows:

Checking Missing Values

The FREQ Procedure

country	Frequency	Percent
Notmissing	27820	100.00

sex	Frequency	Percent
Notmissing	27820	100.00

age	Frequency	Percent
Notmissing	27820	100.00

country_year	Frequency	Percent
Notmissing	27820	100.00

HDI_for_year	Frequency	Percent
Missing	19456	69.94
Notmissing	8364	30.06

gdp_for_year_dollars	Frequency	Percent
Notmissing	27820	100.00

generation	Frequency	Percent

gdp_for_year_dollars	Frequency	Percent
Notmissing	27820	100.00

generation	Frequency	Percent
Notmissing	27820	100.00

year	Frequency	Percent
Notmissing	27820	100.00

suicides_no	Frequency	Percent
Notmissing	27820	100.00

population	Frequency	Percent
Notmissing	27820	100.00

suicidesPer100k_pop	Frequency	Percent
Notmissing	27820	100.00

gdp_per_capita_dollars	Frequency	Percent
Notmissing	27820	100.00

```
title "Checking Missing Values";
proc format;
value $Count_Missing ' ' = 'Missing'
other = 'Notmissing';
value Count_Missing . = 'Missing'
other = 'Notmissing';
run;

proc freq data=clean.suiciderates;
tables _character_ / nocum missing;
format _character_ $Count_Missing.;
tables _numeric_ / nocum missing;
format _numeric_ Count_Missing.;
run;
```

BAN110: Data preparation and Handling

For Treating and handling the missing values we examine the variable HDI_for_year by univariate procedure and we get frequency and total number of missing values.

We have character datatype for the variable HDI_for_year however it contains numerical value so we changed its datatype from character to number the code for the same is shown as follows.

After changing the data type we replace the missing values by it's previous values. Those countries who do not have previous values we replace those values by zero.

For that we impute the missing value the code is as shown in the next slide and output for the same is as follows.

After treating and replacing missing values we run proc print statements to check its frequency and print few observations as shown in the output. And we get its original frequency 27820 with 100% and no missing value for the variable HDI_for_year.

The codes and output is as follows:

BAN110: Data preparation and Handling

Treating Missing values

```
data clean.suiciderates1;
set clean.suiciderates;
HDI_for_year = input( HDI_for_year,best12.);
run;
```

HDI_for_year	Frequency	Percent
Notmissing	27820	100.00

Examining missing values

```
title HDI_for_year;
proc freq data=clean.suiciderates;
table HDI_for_Year;
run;
```

HDI_for_year

The FREQ Procedure

HDI_for_year	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	8364	100.00	8364	100.00
Frequency Missing = 19456				

```
proc univariate data=clean.suiciderates PLOT;
 VAR HDI_for_year;
run;
```

```
/* Imputing Missing Values */
data clean.suiciderates1_fin;
   update clean.suiciderates (obs=0) clean.suiciderates;
   by country;
   output;
run;

data clean.suicideRates;
set clean.suiciderates1_fin;
if missing(HDI_for_year) then do HDI_for_year= 0;
end;
run;

 proc freq data=clean.suiciderates;
 tables _character_ / nocum missing;
 format _character_ $Count_Missing.;
 tables _numeric_ / nocum missing;
 format _numeric_ Count_Missing.;
 run;

proc print data=clean.suiciderates(obs=50);
run;
```

Dataset after Imputing the missing values

Obs	country	year	sex	age	suicides_no	population	suicidesPer100k_pop	country_year	HDI_for_year	gdp_per_capita_dollars	generation	instant	gdp_for_year_dollars
205	Albania	2006	female	15-24 years	0	283138	0	Albania2006	0.695	3235	Millenials	205	8992642349
206	Albania	2006	female	25-34 years	0	186391	0	Albania2006	0.695	3235	Generation X	206	8992642349
207	Albania	2006	female	35-54 years	0	388746	0	Albania2006	0.695	3235	Boomers	207	8992642349
208	Albania	2006	female	5-14 years	0	267316	0	Albania2006	0.695	3235	Millenials	208	8992642349
209	Albania	2006	female	55-74 years	0	215907	0	Albania2006	0.695	3235	Silent	209	8992642349

BAN110: Data preparation and Handling

Detecting and Removing Outliers:

After studying the data, we have developed a summary table for different numerical variables and made the relevant decision for outliers in the variables. Further details will be provided after this

Summary of Detecting and Removing Outliers:

Variables	Number of Outliers	Distribution	Decision for Outliers
Year	0	Normal	NA
Population	4180	Right Skewed	Remove
SuicidesPer100k_pop	2046	Right Skewed	Remove
GDP_for_years_dollars	3088	Right Skewed	Remove
GDP_per_capita_dollars	1016	Right Skewed	Remove

As you can see from the summary table, except for the year variables, all the other variables are right skewed. Hence, we choose the Interquartile Range Method to detect and remove the outliers instead of the trimming by statistics method because the latter requires the data to be normally distributed.

We have attached the code to detect outliers as follows, using population as an example,

Code to Detect Outliers:

```
title"Q1 Q3 and  Interquartile range of population";

proc means data=clean.suicideRates Q1 Q3 Qrange;
    var population;
    output out=clean.population Q1=
    Q3=
    QRange= / autoname;
run;

proc print data = clean.population;
run;

title"Outliers of population based on the interquantile range method";

data _null_;
    file print;
    set clean.suicideRates(keep=instant population);

    if _n_=1 then
        set clean.population;

    if population le population_Q1 - 1.5*population_QRange and not missing(population)
        or population ge population_Q3 + 1.5*population_QRange then
            put "Possible Outlier for instant " instant "value of population is "
            population ;
run;
```

BAN110: Data preparation and Handling

Results of Outliers Detection:

Outliers of population based on the interquantile range method

```
Possible Outlier for instant 651 value of population is 3619000
Possible Outlier for instant 656 value of population is 3622000
Possible Outlier for instant 663 value of population is 3580000
Possible Outlier for instant 668 value of population is 3691000
Possible Outlier for instant 675 value of population is 3616600
Possible Outlier for instant 680 value of population is 3755500
Possible Outlier for instant 687 value of population is 3672300
Possible Outlier for instant 692 value of population is 3820400
Possible Outlier for instant 699 value of population is 3729600
Possible Outlier for instant 704 value of population is 3885600
Possible Outlier for instant 711 value of population is 3787800
Possible Outlier for instant 716 value of population is 3950100
Possible Outlier for instant 723 value of population is 3847300
Possible Outlier for instant 728 value of population is 4014100
Possible Outlier for instant 735 value of population is 3934581
Possible Outlier for instant 740 value of population is 4083427
Possible Outlier for instant 747 value of population is 3990829
Possible Outlier for instant 753 value of population is 4143203
Possible Outlier for instant 759 value of population is 4045105
Possible Outlier for instant 764 value of population is 4200203
Possible Outlier for instant 771 value of population is 4096814
Possible Outlier for instant 776 value of population is 4253771
Possible Outlier for instant 784 value of population is 4145861
Possible Outlier for instant 789 value of population is 4306817
Possible Outlier for instant 797 value of population is 4191889
Possible Outlier for instant 801 value of population is 4355651
Possible Outlier for instant 809 value of population is 4237124
```

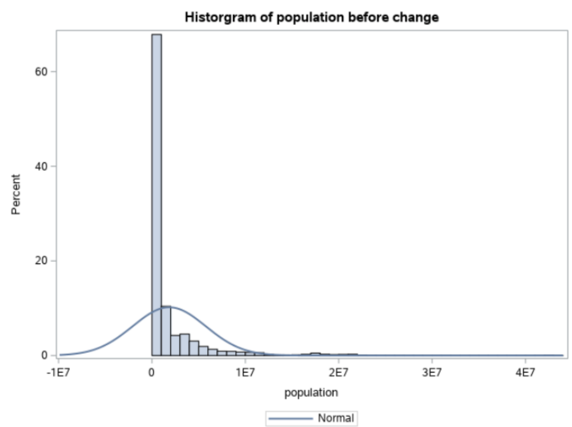
We also developed a code to obtain the total number of Outliers to derive the number in the summary table as shown previously.

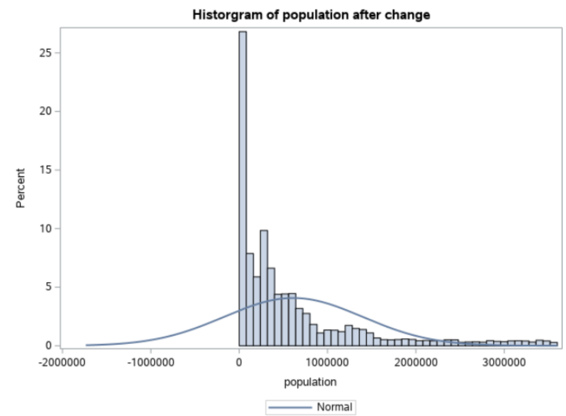
Code to obtain the total number of outliers:

```
data clean.population;
set clean.population;
n=1;
run;
data clean.suicideRates;
set clean.suicideRates;
n=1;
run;
data clean.outliersPopulation; merge clean.suicideRates(in =d1) clean.population(in =d2);
by n;
if d1 =1 and d2 =1;
keep  instant population population_Q1 population_Q3 population_QRange;
run;

data clean.outliersPopulation;
set clean.outliersPopulation;
    retain below_Q1 above_Q3 ;
    if population >= (population_Q3 + 1.5*population_QRange) then do;
        above_Q3 = 1;
    end;
    else above_Q3 = 0;
    if population <= (population_Q1 - 1.5*population_QRange) then do;
    below_Q1 = 1;
    end;
    else below_Q1 = 0;
    if missing(above_Q3) then above_Q3 = 0;
        if missing(below_Q1) then below_Q1 = 0;
run;

title "Total outliers for population ";
proc means data=clean.outliersPopulation sum;
var above_Q3 below_Q1;
run;
```

Firstly we try to remove the outliers for individual variables to determine whether there is any distribution change after removing them.

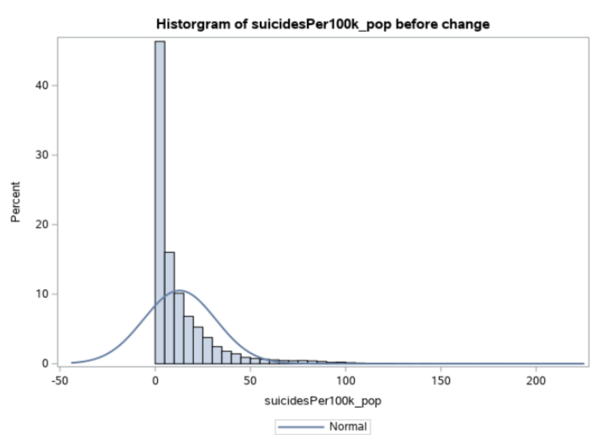
For Population:

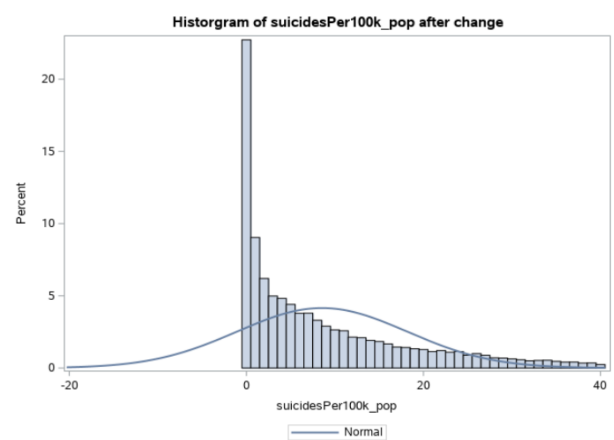




As you can see, the distribution remains to be right skewed with a tail to the right. The total number of outliers is also insignificant as compared to the total rows. Hence we decide to remove the outliers for the population variable.

For SuicidesPer100k_pop:

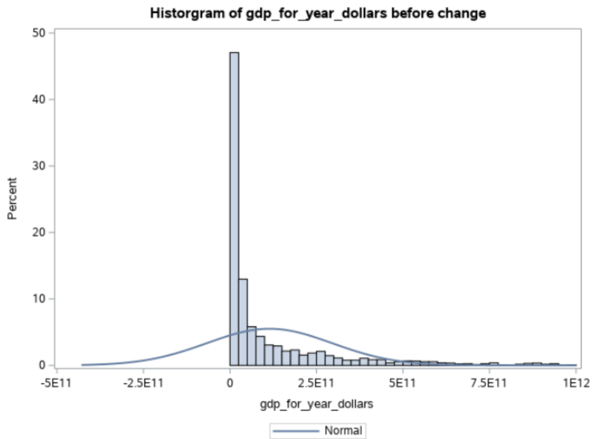


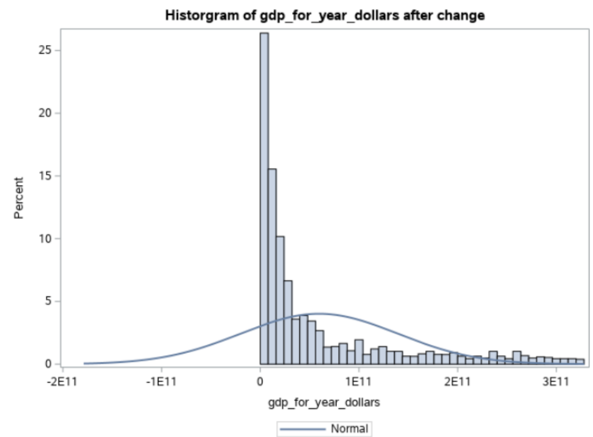


Similarly, the distribution remains to be right skewed with a tail to the right. The total number of outliers is also insignificant as compared to the total rows. Hence we decide to remove the outliers for the SuicidesPer100k_pop variable.

For GDP_for_years_dollars:

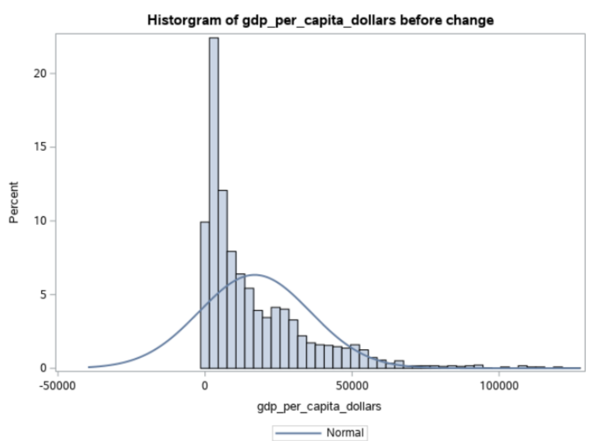
BAN110: Data preparation and Handling

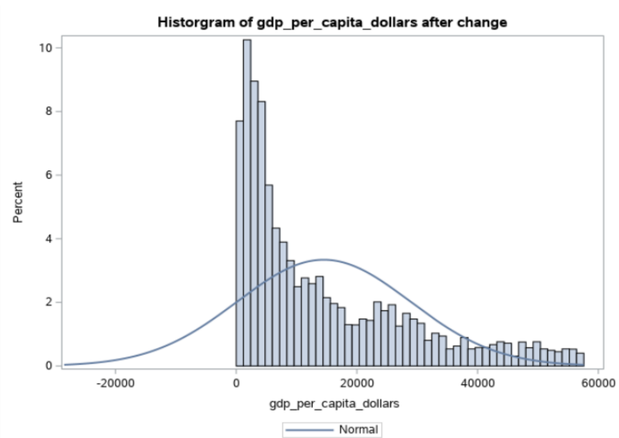




Similarly, the distribution remains to be right skewed with a tail to the right. The total number of outliers is also insignificant as compared to the total rows. Hence we decide to remove the outliers for the GDP_for_years_dollars variable.

For GDP_per_capita_dollars :





Similarly, the distribution remains to be right skewed with a tail to the right. The total number of outliers is also insignificant as compared to the total rows. Hence we decide to remove the outliers for the GDP_per_capita_dollars variable.

We used the code below to remove the outliers:

BAN110: Data preparation and Handling

```
/* remove outliers for individual vairable and check for distribution change */
data clean.OutliersRm_population;
    file print;
    set clean.suicideRates(keep=instant population);

    if _n_=1 then
        set clean.population;

    if population le population_Q1 - 1.5*population_QRange and not missing(population)
    or population ge population_Q3 + 1.5*population_QRange then
            delete;
run;
```

After reviewing the dataset for each individual variable after removing the outliers, we used the following code to combine them.

Code to combined dataset after removing outliers:

```
/* Since distribution remain similar after removing outliers, it's safe to remove all those outliers  */
proc sort data = clean.OutliersRm_population(keep = instant);
by instant;
run;
proc sort data = clean.OutliersRm_suicidesper100(keep = instant);
by instant;
run;
proc sort data = clean.OutliersRm_gdpcapital(keep = instant);
by instant;
run;
proc sort data = clean.OutliersRm_gdpyear(keep = instant);
by instant;
run;
data clean.suicide_final_OutliersRm;
   merge clean.suicideRates (in=d0) clean.OutliersRm_population (in=d1)
   clean.OutliersRm_suicidesper100 (in=d2) clean.OutliersRm_gdpcapital (in=d3) clean.OutliersRm_gdpyear (in=d4) ;
   by instant;
   if d1=1 and d2=1 and d3=1 and d4=1;
   drop n;
run;
```

After removing outliers and combining them, we have 19347 rows in the dataset.

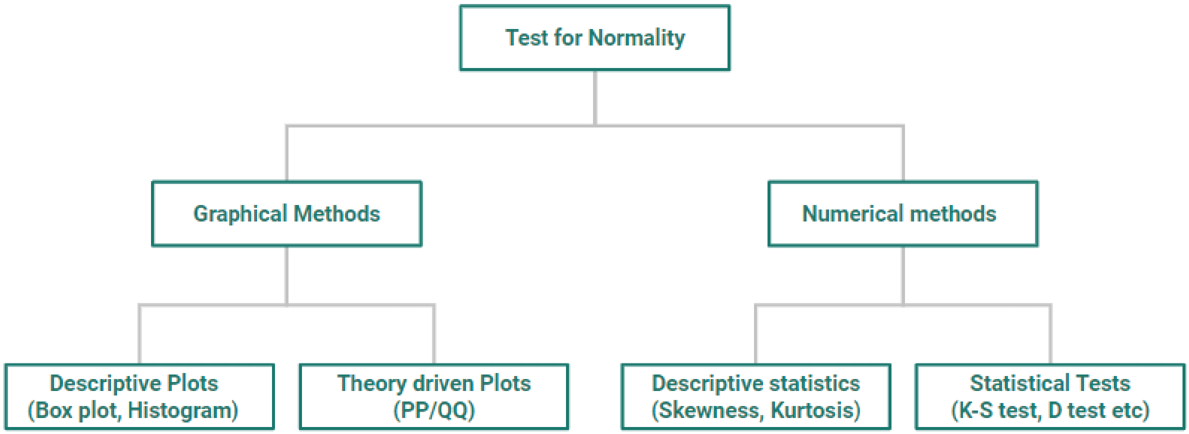
Results of removing outliers:

Total rows: 19347 Total columns: 13 Rows 1-100

	country	year	sex	age	suicides_no
1	Albania	1987	male	15-24 years	21
2	Albania	1987	male	35-54 years	16
3	Albania	1987	female	15-24 years	14
4	Albania	1987	male	75+ years	1
5	Albania	1987	male	25-34 years	9
6	Albania	1987	female	75+ years	1
7	Albania	1987	female	35-54 years	6
8	Albania	1987	female	25-34 years	4
9	Albania	1987	male	55-74 years	1
10	Albania	1987	female	5-14 years	0
11	Albania	1987	female	55-74 years	0
12	Albania	1987	male	5-14 years	0
13	Albania	1988	female	75+ years	2
14	Albania	1988	male	15-24 years	17

BAN110: Data preparation and Handling

Test for Normality and distribution:



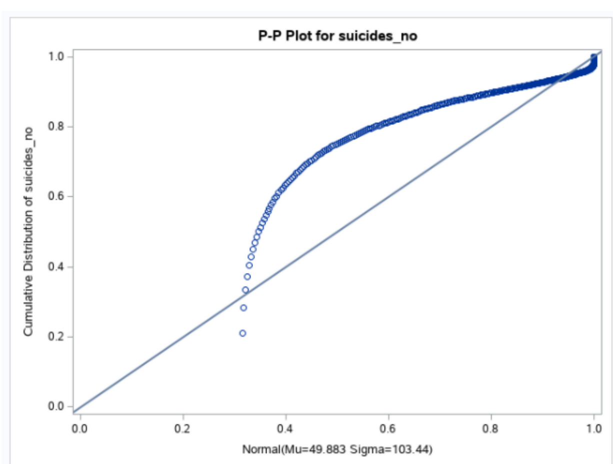
We have tested the normality of our numerical variables using the Theory driven PP Plots.

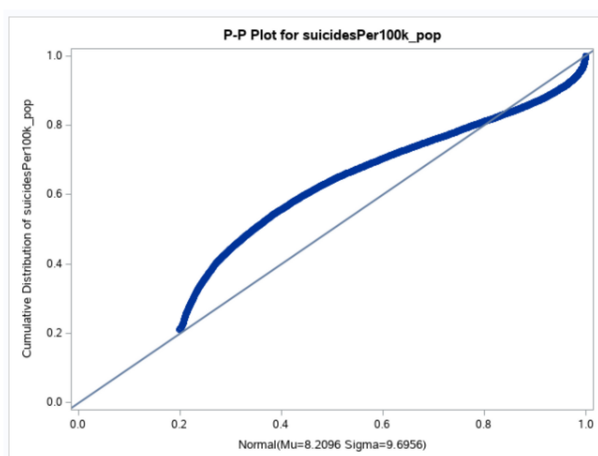
The probability-probability plot (P-P plot or percent plot) compares an empirical cumulative distribution function of a variable with a specific theoretical cumulative distribution function (e.g., the standard normal distribution function).

Code:

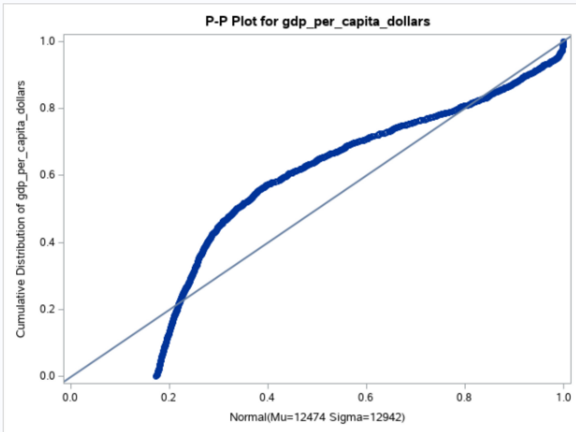
```
proc univariate data=clean.suicide_final_OutliersRm;
ppplot;
run;
```

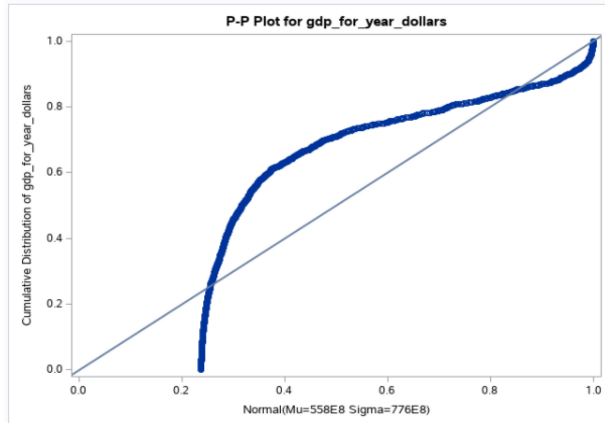
Output:





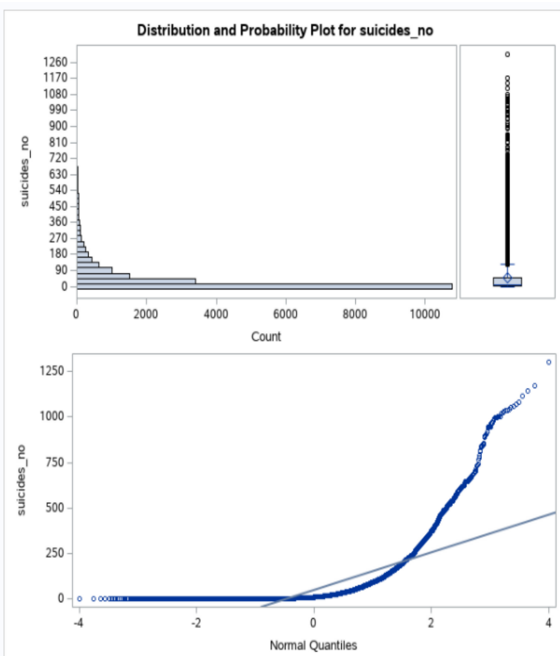
BAN110: Data preparation and Handling

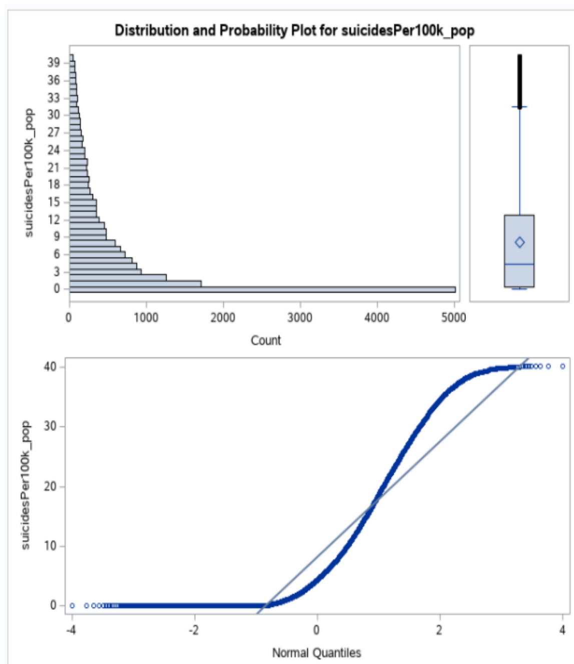




For all the above PP Plots of the variables we observe that the data points are not distributed along the straight line.

Distribution and Probability plot for **suicides_no** and **suicidesPer100k_pop**





We see Right-skewed distribution for our numerical variables. The presence of skewed distribution or outlier influences has an effect on the analysis because some types of analyses accept only normal (or close to normal) distribution.

BAN110: Data preparation and Handling

Applying Transformations:

Let's apply log and root transformations to **suicides_no** and **suicidesPer100k_pop** and try to achieve normality for our variables.

log transformation: The most frequently used transformation to transform a right-skewed distribution is the log transformation . Note that the logarithm is defined only for positive values. In the case of negative values, a constant has to be added to the data in order to make them all positive.

root transformation: Another transformation that normalizes data is the root transformation .

Code:

```
/* Transformations */
Data clean.suiciderates_transformed;
SET clean.suicide_final_OutliersRm;
 log_suicides_no = log(suicides_no+1);
 log_suicidesPer100k_pop = log(suicidesPer100k_pop+1);
 root4_suicides_no = (suicides_no+1) ** 0.25;
 root4_suicidesPer100k_pop = (suicidesPer100k_pop+1) ** 0.25;
RUN;
```

Note: We have added a constant to the variables after checking its minimum values to avoid undefined values.

```
proc means data=clean.suicide_final_OutliersRm min;
var suicides_no suicidesPer100k_pop;
run;
```

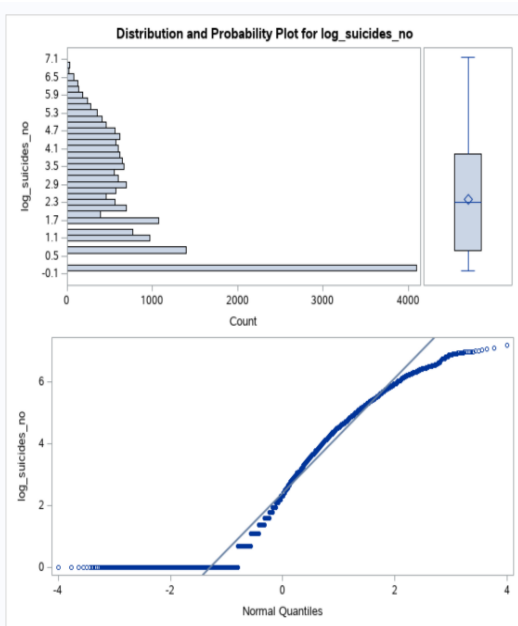
The MEANS Procedure

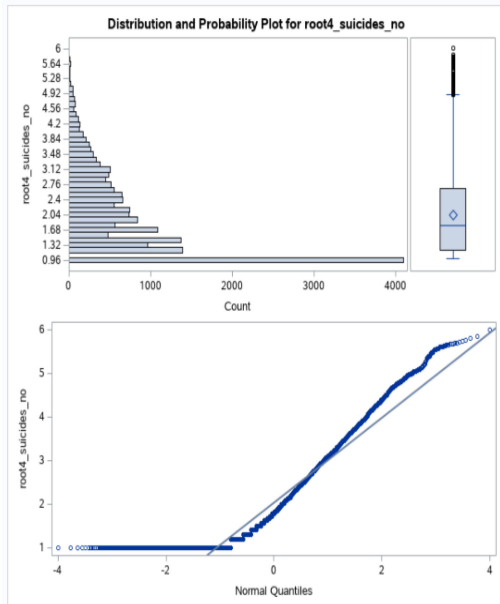
Variable	Minimum
suicides_no	0
suicidesPer100k_pop	0

```
ODS select TestsForNormality Plots;
PROC UNIVARIATE DATA = clean.suiciderates_transformed NORMAL PLOT;
 RUN;
ODS select All;
```

Output:

Lets compare the log and root transformations





Based on the above plots of log and root transformation on **suicides_no** variable respectively, we can say that we have achieved normality for our variable.

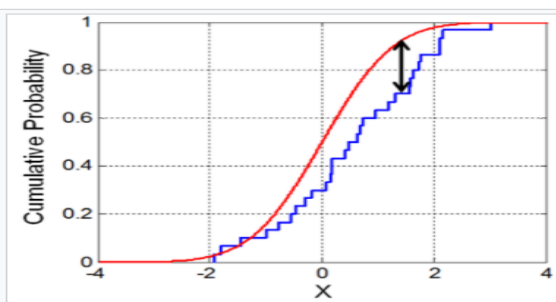


Illustration of the Kolmogorov–Smirnov statistic. The red line is a model CDF, the blue line is an empirical CDF, and the black arrow is the K–S statistic.

The **Kolmogorov–Smirnov** statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples. Smaller value of KS Statistic indicates higher achievable normality.

BAN110: Data preparation and Handling

Let's check the statistical parameters for both the variables.

The UNIVARIATE Procedure
Variable: log_suicides_no

Tests for Normality				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.11522	Pr > D	<0.0100
Cramer-von Mises	W-Sq	43.64535	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	344.5417	Pr > A-Sq	<0.0050

The UNIVARIATE Procedure
Variable: root4_suicides_no

Tests for Normality				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.144006	Pr > D	<0.0100
Cramer-von Mises	W-Sq	83.73077	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	548.2816	Pr > A-Sq	<0.0050

The UNIVARIATE Procedure
Variable: suicides_no

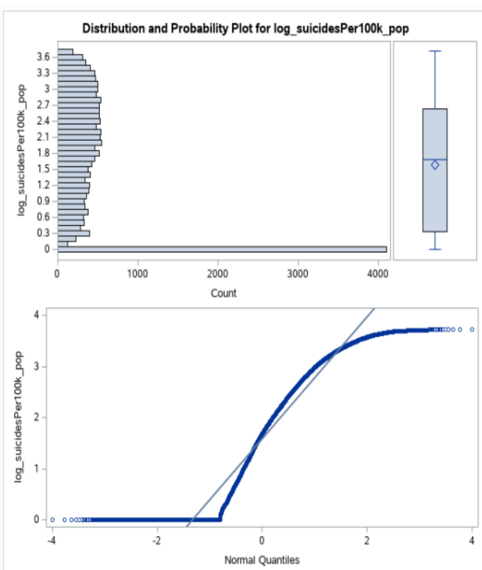
Tests for Normality				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.314823	Pr > D	<0.0100
Cramer-von Mises	W-Sq	585.9781	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	2966.667	Pr > A-Sq	<0.0050

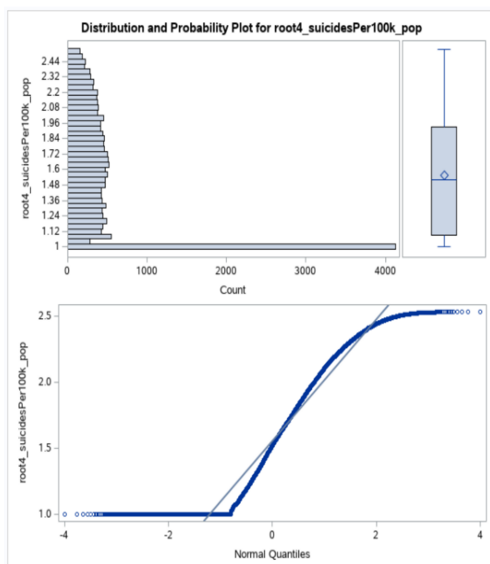
From the Kolmogorov-Smirnov statistic we see, for example, in our case the log transformation performs better than the root transformation for suicides_no

KS statistic log_suicides_no < KS statistic root4_suicides_no <KS statistic suicides_no

0.11522 < 0.144006 < 0.314823

Similarly we have examined the log and root transformations of **suicidesPer100k_pop** variable**.**





BAN110: Data preparation and Handling

The UNIVARIATE Procedure
Variable: log_suicidesPer100k_pop

Tests for Normality				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.120472	Pr > D	<0.0100
Cramer-von Mises	W-Sq	60.45628	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	470.7949	Pr > A-Sq	<0.0050

The UNIVARIATE Procedure
Variable: root4_suicidesPer100k_pop

Tests for Normality				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.113371	Pr > D	<0.0100
Cramer-von Mises	W-Sq	58.81612	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	443.5282	Pr > A-Sq	<0.0050

Based on this K-S statistic for **suicidesPer100k_pop** we see that root transformation performs better for this case.

Conclusion

We have successfully completed all the Data Preparation and Handling steps on the suicide dataset. This includes

- Data Import
- Checking and correcting errors
- Checking Missing Values
- Treating Missing Values
- Detecting and Removing Outliers
- Test for Normality and Distribution
- Applying Normality Transformations
- Testing Normality for Transformed Variables