



**Program Code: J620-002-4:2020**

**Program Name: FRONT-END SOFTWARE DEVELOPMENT**

**Title : Exe12 - Confidence Intervals NHANES Exercise**

**Name: Phua Yan Han**

**IC Number: 050824070059**

**Date : 11/7/23**

**Introduction :**

**Conclusion :**



## **Exercise 1: Confidence Intervals - NHANES**

This exercise, we are going to practice on how to load data, clean/manipulate a dataset, and construct a confidence interval for the difference between two population proportions and means.

We will use the 2015-2016 wave of the NHANES data for our analysis.

For our population proportions, we will analyze the difference of proportion between female and male smokers. The column that specifies smoker and non-smoker is "SMQ020" in our dataset.

For our population means, we will analyze the difference of mean of body mass index within our female and male populations. The column that includes the body mass index value is "BMXBMI".

Additionally, the gender is specified in the column "RIAGENDR".

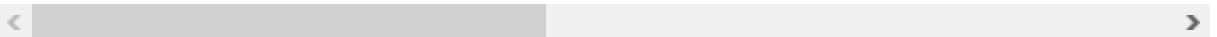
```
In [49]: import pandas as pd
import numpy as np
import matplotlib
matplotlib.use('Agg')
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

```
In [50]: url = "../Data files/nhanes_2015_2016.csv"
da = pd.read_csv(url)
da
```

Out[50]:

	SEQN	ALQ101	ALQ110	ALQ130	SMQ020	RIAGENDR	RIDAGEYR	RIDRETH1	DMDCTZ1
0	83732	1.0	NaN	1.0	1	1	62	3	1.0
1	83733	1.0	NaN	6.0	1	1	53	3	2.0
2	83734	1.0	NaN	NaN	1	1	78	3	1.0
3	83735	2.0	1.0	1.0	2	2	56	3	1.0
4	83736	2.0	1.0	1.0	2	2	42	4	1.0
...	...	...	...	...	...	...	...	...	...
5730	93695	2.0	2.0	NaN	1	2	76	3	1.0
5731	93696	2.0	2.0	NaN	2	1	26	3	1.0
5732	93697	1.0	NaN	1.0	1	2	80	3	1.0
5733	93700	NaN	NaN	NaN	1	1	35	3	2.0
5734	93702	1.0	NaN	2.0	2	2	24	3	1.0

5735 rows × 28 columns



## Investigating and Cleaning Data

Create a new column named 'SMQ020x' and store data from column 'SMQ020' with following replacements:

- 1 to "Yes"
- 2 to "No"
- 7 to NaN
- 9 to NaN

```
In [51]: da['SMQ020x']=da['SMQ020']
reset = {1:'Yes',2:'No',7:None,9:None}
da = da.replace({'SMQ020x':reset})
da['SMQ020x']
```

```
Out[51]: 0      Yes
1      Yes
2      Yes
3      No
4      No
...
5730   Yes
5731   No
5732   Yes
5733   Yes
5734   No
Name: SMQ020x, Length: 5735, dtype: object
```

Create a new column named 'RIAGENDRx' and store data from column 'RIAGENDR' with following replacements:

- 1 to "Male"
- 2 to "Female"

```
In [52]: da['RIAGENDRx']=da['RIAGENDR']
reset = {1:'Male',2:'Female'}
da = da.replace({'RIAGENDRx':reset})
da['RIAGENDRx']
```

```
Out[52]: 0      Male
1      Male
2      Male
3      Female
4      Female
...
5730   Female
5731   Male
5732   Female
5733   Male
5734   Female
Name: RIAGENDRx, Length: 5735, dtype: object
```

In [53]: da

Out[53]:

	SEQN	ALQ101	ALQ110	ALQ130	SMQ020	RIAGENDR	RIDAGEYR	RIDRETH1	DMDCTZ1
0	83732	1.0	NaN	1.0	1	1	62	3	1.0
1	83733	1.0	NaN	6.0	1	1	53	3	2.0
2	83734	1.0	NaN	NaN	1	1	78	3	1.0
3	83735	2.0	1.0	1.0	2	2	56	3	1.0
4	83736	2.0	1.0	1.0	2	2	42	4	1.0
...	...	...	...	...	...	...	...	...	...
5730	93695	2.0	2.0	NaN	1	2	76	3	1.0
5731	93696	2.0	2.0	NaN	2	1	26	3	1.0
5732	93697	1.0	NaN	1.0	1	2	80	3	1.0
5733	93700	NaN	NaN	NaN	1	1	35	3	2.0
5734	93702	1.0	NaN	2.0	2	2	24	3	1.0

5735 rows × 30 columns



Drop all NAs from both SMQ020x & RIAGENDRx and store into a new dataframe named 'dx'.  
Plot the following crosstab using pd.crosstab library.

```
In [54]: dx = da[da['SMQ020x'] !=None]
pd.crosstab(dx['SMQ020x'],dx['RIAGENDRx'])
```

Out[54]:

RIAGENDRx	Female	Male
SMQ020x		
No	2066	1340
Yes	906	1413

Replace dx['SMQ020x'] "Yes" to 1 and "No" to 0.

```
In [55]: reset = {'Yes':1, 'No':0}
dx = dx.replace({'SMQ020x':reset})
dx=dx[['SMQ020x', 'RIAGENDRx']]
dx
```

Out[55]:

	SMQ020x	RIAGENDRx
0	1.0	Male
1	1.0	Male
2	1.0	Male
3	0.0	Female
4	0.0	Female
...	...	...
5730	1.0	Female
5731	0.0	Male
5732	1.0	Female
5733	1.0	Male
5734	0.0	Female

5735 rows × 2 columns

Calculate the 'mean' and 'size' and store into a new dataframe called dz

```
In [56]: dz = dx[['SMQ020x', 'RIAGENDRx']].groupby('RIAGENDRx').agg(['mean', 'size'])
dz
```

Out[56]:

	SMQ020x	
	mean	size
RIAGENDRx		
Female	0.304845	2976
Male	0.513258	2759

## Constructing Confidence Intervals

Now that we have the population proportions of male and female smokers, we can begin to calculate confidence intervals. From lecture, we know that the equation is as follows:

$$\text{Best Estimate} \pm \text{Margin of Error}$$

Where the *Best Estimate* is the **observed population proportion or mean** from the sample and the *Margin of Error* is the **t-multiplier**.

The equation to create a 95% confidence interval can also be shown as:

*Population Proportion or Mean  $\pm (t - multiplier * Standard Error)$*

The Standard Error is calculated differently for population proportion and mean:

$$Standard Error for Population Proportion = \sqrt{\frac{Population Proportion * (1 - Population Proportion)}{Number Of Observations}}$$

$$Standard Error for Mean = \frac{Standard Deviation}{\sqrt{Number Of Observations}}$$

Lastly, the standard error for difference of population proportions and means is:

$$Standard Error for Difference of Two Population Proportions Or Means = \sqrt{SE_1^2 + SE_2^2}$$

Calculate the standard error for female

```
In [71]: import math
p = dz['SMQ020x']['mean']['Female']
n = dz['SMQ020x']['size']['Female']
sef = math.sqrt(p*(1-p)/n)
sef
```

Out[71]: 0.008438475404634192

Calculate the standard error for male

```
In [72]: p = dz['SMQ020x']['mean']['Male']
n = dz['SMQ020x']['size']['Male']
sem = math.sqrt((p*(1-p))/n)
sem
```

Out[72]: 0.009515714829783395

Calculate the difference between these two Standard Errors

```
In [59]: dse = math.sqrt(sef**2+sem**2)
dse
```

Out[59]: 1.0871506400747817

Calculate the confidence Interval

```
In [60]: ci = dz['SMQ020x']['mean']['Male']-dz['SMQ020x']['mean']['Female']
x = ci-2*dse
y = ci+2*dse
print(x,y)
print(dse)
```

```
-1.9658882385099279 2.382714321789199
1.0871506400747817
```

## Difference of Two Population Means

Now we look into the differences between 2 population means

```
In [61]: da["BMXBMI"].head()
```

```
Out[61]: 0    27.8
1    30.8
2    28.8
3    42.4
4    20.3
Name: BMXBMI, dtype: float64
```

```
In [62]: x = da[['BMXBMI', 'RIAGENDRx']].groupby('RIAGENDRx').agg(['mean', 'std', 'size'])
x
```

```
Out[62]:
```

	BMXBMI		
	mean	std	size
RIAGENDRx			
Female	29.939946	7.753319	2976
Male	28.778072	6.252568	2759

Calculate the Standard Error for Mean for both female and male

```
In [63]: s = x['BMXBMI']['std']['Female']
n = x['BMXBMI']['size']['Female']
semf = s/math.sqrt(n)
s = x['BMXBMI']['std']['Male']
n = x['BMXBMI']['size']['Male']
semm = s/math.sqrt(n)
print(semf,semm)
```

```
0.14212522940758335 0.11903715722332033
```

Calculate the difference between 2 Standard Error for Mean

```
In [64]: dsem = math.sqrt(semf**2+semm**2)
dsem
```

```
Out[64]: 0.18538992862064455
```

The difference between two means for male and female

```
In [65]: dbm = math.sqrt(x['BMXBMI']['mean']['Female']**2+x['BMXBMI']['mean']['Male']**2)
dbm
```

```
Out[65]: 41.52803607359479
```

Calculate the confidence interval between two population means

```
In [66]: clpm = x['BMXBMI']['mean']['Female'] - x['BMXBMI']['mean']['Male']
y = clpm - 2*dsem
z = clpm + 2*dsem

print(y,z)
```

```
0.7910936830856763 1.5326533975682544
```