Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exe12 - Confidence Intervals NHANES Exercise

Name: Chong Mun Chen

IC Number: 960327-07-5097

Date : 11/7/2023

Introduction : Practising more on solving confidence interval exercises.

Conclusion : Getting familiar with population proportions, means, standard errors, and T-distribution, all of which constructs a confidence interval.

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 [37]:
```
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 [38]:
```
url = "../data_samples/nhanes_2015_2016.csv"
da = pd.read_csv(url)
```

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 [39]:
```
da['SMQ020x'] = da['SMQ020']
reset = {1: 'Yes', 2: 'No', 7: None, 9: None}
da = da.replace({'SMQ020x':reset})
da['SMQ020x']
```

Out[39]:
```
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 [40]:
```
da['RIAGENDRx'] = da['RIAGENDR']
reset = {1:'Male', 2:'Female'}
da = da.replace({'RIAGENDRx':reset})
da['RIAGENDRx']
```

Out[40]:
```
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
```

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

In [41]:
```
dx = da.copy()
dx.dropna(subset=['SMQ020x', 'RIAGENDRx'], inplace=True)
pd.crosstab(dx['SMQ020x'], dx['RIAGENDRx'])
```

Out[41]:

RIAGENDRx	Female	Male
SMQ020x		
No	2066	1340
Yes	906	1413

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

In [42]:
```
reset = {'Yes':1, 'No':0}
dx = dx.replace({'SMQ020x':reset})
dx = dx[['SMQ020x', 'RIAGENDRx']]
dx
```

Out[42]:

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

5725 rows × 2 columns

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

In [43]:
```
dz = dx[['SMQ020x', 'RIAGENDRx']].groupby('RIAGENDRx').agg(['mean', 'size'])
dz
```

Out[43]:

	SMQ020x	
	mean	**size**
RIAGENDRx		
Female	0.304845	2972
Male	0.513258	2753

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:

$$Best\ Estimate \pm Margin\ of\ Error$$

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 differenly 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:

$$Standard\ Error\ for\ Difference\ of\ Two\ Population\ Proportions\ Or\ Means = \sqrt{SE^2_{Proportion\ 1} + SE^2_{Proportion\ 2}}$$

Difference of Two Population Proportions

Calculate the standard error for female

In [44]:
```
import math

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

Out[44]: 0.008444152146214435

Calculate the standard error for male

In [45]:
```
import math

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

Out[45]: 0.009526078653689868

Calculate the difference between these two Standard Errors

In [46]:
```
dse = math.sqrt(sef**2 + sem**2)
dse
```

Out[46]: 0.012729881381407434

Calculate the confidence Interval

In [49]:
```
from scipy.stats import t

dpop = dz['SMQ020x']['mean']['Female'] - dz['SMQ020x']['mean']['Male']

confidence = 0.95
dof = dz['SMQ020x']['size']['Female']
+ dz['SMQ020x']['size']['Male'] - 2

t_multiplier = t.ppf((1 + confidence) / 2, dof)

lower_bound = dpop - (t_multiplier * dse)
upper_bound = dpop + (t_multiplier * dse)

print(t_multiplier)
print(dpop)
print(lower_bound, upper_bound)
```

```
1.9607625111943023
-0.20841304163963553
-0.23337331582424956 -0.1834527674550215
```

Difference of Two Population Means

Now we look into the differences between 2 population means

In [30]:
```
da["BMXBMI"].head()
```

Out[30]:
```
0    27.8
1    30.8
2    28.8
3    42.4
4    20.3
Name: BMXBMI, dtype: float64
```

In [32]:
```
x = da[['BMXBMI', 'RIAGENDRx']].groupby('RIAGENDRx').agg(['mean', 'std', 'size
x
```

Out[32]:

	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 [33]:
```
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 [34]:
```
dsem = math.sqrt(semf**2 + semm**2)
dsem
```

Out[34]:
```
0.18538992862064455
```

The difference between two means for male and female

In [35]:
```
dbm = math.sqrt(x['BMXBMI']['mean']['Female']**2 + x['BMXBMI']['mean']['Male']
dbm
```

Out[35]:
```
41.52803607359479
```

Calculate the confidence interval between two population means

In [36]:
```
cipm = x['BMXBMI']['mean']['Female'] - x['BMXBMI']['mean']['Male']
y = cipm - 2 * dsem
z = cipm + 2 * dsem
print(y,z)
```

```
0.7910936830856763 1.5326533975682544
```