Lab #8: Chi-Square-Confidence Intervals-Hypothesis Testing (3)

Objectives:

1. Chi-Square Analysis examples on Contingency Tables
2. Hypothesis Test and Confidence Intervals for Proportions
3. Hypothesis testing using T-Tests, Pooled and Normal

Chi-Square Analysis

Suppose that we want to analyze the following contingency table:

		Vaccinated		
		Yes	No	Total
Autism	Yes Observed	440	184	
	Yes Expected			
	No Observed	1240	584	
	No Expected			
Total				

In order to import a contingency table in SAS we need to create a new dataset with the two categorical variables and all the possible cases. We also need to define a dummy variable to hold all the numerical entries. We call that variable counts. To create the dataset we use the code:

```
DATA Vaccines;
INPUT Autism $ Vaccinated $ count;
datalines;
Yes Yes 440
Yes No 184
No Yes 1240
No No 584
;
```

Notice the use of $ after the variables Autism and Vaccinated. Those are there to remind SAS that these are categorical variables. Count, being a numerical variable, does not need a $ after it.

This has created and stored our dataset. In order for us to turn it into a contingency table and do the chi square analysis we need to use the procedure freq. as follows:

```
TITLE 'Contingency table and chi square analysis';
PROC FREQ data=Vaccines ORDER=DATA;
tables Autism*Vaccinated/ chisq expected norow nocol nopercent;
weight count;
RUN;
```

The command: **`tables Autism*Vaccinated`** instructs SAS to create a contingency table using the two variables Autism and Vaccinated. The rest of this command works as follows:

chisq : do the chi square analysis
expected: compute and display the expected values for each cell
norow: do not show the row marginal probabilities
nocol: do not show the column marginal probabilities
nopercent: do not show general probabilities

The command **ORDER=DATA** tells the procedure freq to present the data in the order we put them in (so no flipping the YES and NO anymore!).

The output is something like this:

Frequency
Expected

Table of Autism by Vaccinated

Autism	Vaccinated		
	no	yes	Total
no	584 572.24	1240 1251.8	1824
yes	184 195.76	440 428.24	624
Total	768	1680	2448

And the statistics table is:

Statistic	DF	Value	Prob
Chi-Square	1	1.3827	0.2396
Likelihood Ratio Chi-Square	1	1.3931	0.2379
Continuity Adj. Chi-Square	1	1.2676	0.2602
Mantel-Haenszel Chi-Square	1	1.3821	0.2397

Statistic	DF	Value	Prob
Phi Coefficient		0.0238	
Contingency Coefficient		0.0238	
Cramer's V		0.0238	

Once again out of all the tests we focus on the first one. We say thus that the p value is 0.2396 which is much larger than 0.05. So we fail to reject the null hypothesis. This dataset does not provide enough evidence to support the claim that Autism and Vaccines are dependent.

Hypothesis Testing and confidence intervals for proportions

In the following section we will explore with sas the idea of a hypothesis testing with Z-scores for proportions.
It is estimated that 8% of the population are colorblind. Pingelap, is a south pacific island with strange nickname. It is called "the color blind island", since a 26.4% of the population, 66 people, is colorblind and almost everybody has some sort of color deficiency. Let's test if the two ratios are truly different if we know that the island has 250 people.

First we need to create a dataset in sas. Let's call it Color and have two variables one for location and the other for states of colorblindness.

```
DATA Color;
INPUT Location $ Colorblindness $ count;
datalines;
World Yes 8
World No 92
Pingelap Yes 26.4
Pingelap No 73.6
;
```

This has created our dataset. Now to view it and do the z-test comparison of the ratios we need to use the following code:

```
TITLE "Z score for proportions test and 95% confidence ";
PROC FREQ data=Color ORDER=DATA;
tables Location*Colorblindness/  riskdiff(equal var=null cl=wald) norow nocol
nopercent alpha=0.05;
weight count;
RUN;
```

Once again the procedure frequency will output the table in a nice 2 by 2 format and once more we have
norow: do not show the row marginal probabilities
nocol: do not show the column marginal probabilities
nopercent: do not show general probabilities

The creation of the z-tables and happens because of the command **riskdiff**. In the documentation for this command in SAS we learn that the binomial proportions are called "risks," so a "risk difference" is a difference in proportions.
The command **equal var=null** tells sas that we don't expect the variability of the two proportions to be equal and the command cl=wald instructs it to use the Wald Method to compute the standard error, which is the formula we learned in class i.e.

$$SE = \sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}}$$

You can learn more about the **procedure freq** here:

http://support.sas.com/documentation/cdl/en/statug/68162/HTML/default/viewer.htm#statug_freq_syntax08.htm

As you can see the 95% interval is computed as well for the difference of the two proportions. This is obtained by adjusting the command **alpha=0.05.**

Let's change that to 0.1 and see what happens with the code:

```
TITLE "Z score for proportions test and 90% confidence ";
PROC FREQ data=Color ORDER=DATA;
tables Location*Colorblindness/  riskdiff(equal var=null cl=wald) norow nocol
nopercent alpha=0.01;
weight count;
RUN;
```

Obviously there is no change in the proportions difference test and we get:

Proportion (Risk) Difference Test	
H0: P1 - P2 = 0 Wald Method	
Proportion Difference	-0.1840
ASE (H0)	0.0534
Z	-3.4477
One-sided Pr < Z	0.0003
Two-sided Pr > |Z|	0.0006
Column 1 (Colorblindness = Yes)	

What changes in the confidence limits which gives us now:

Confidence Limits for the Proportion (Risk) Difference		
Proportion Difference = -0.1840		
Type	**90% Confidence Limits**	
Wald	-0.2691	-0.0989
Column 1 (Colorblindness = Yes)		

Hypothesis testing using T-Tests, Pooled and Normal

As you recall, the **PROC IMPORT** statement is the best way to enter external data sets. The CSV file we will be using is called "Qs3small.csv". Download and save it in your favorite folder and mark the complete path to it. Then use the following code to import it, making sure you put the correct path on the **DATAFILE** argument. Save the dataset as Qs3.

```
PROC IMPORT OUT= Qs3
            DATAFILE= "put the complete path here…\Qs3small.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
```

This dataset contains some of the results from questionnaire 3. Let's try to see if there is any difference of the reported understanding of the lectures based on gender. To keep things tidy lets first sort our dataset based on gender using the code:

```
PROC SORT Data=Qs3;
by Gender;
RUN;
```

And then lets print our dataset with proc print.

```
PROC PRINT Data=Qs3;
RUN;
```

Let's also do some basic summary statistics on the average understanding of the whole population using proc univariate as follows:

```
TITLE "Summary statistics for Lectures, all students";
PROC UNIVARIATE Data=Qs3 plot;
var Lectures;
RUN;
```

We notice that the total **mean is 3.6951** and **the standard deviation is 0.757**. We proceed to compute the summary statistics based on different genders with the slightly modified proc univariate

```
TITLE "Summary statistics for Lectures, by gender ";
PROC UNIVARIATE Data=Qs3 plot;
var Lectures;
by Gender;
RUN;
```

Notice that instead of getting 2 categories (M, F) we got 3! Fear not, one corresponds to one observation that was missing the gender, so it became a group on its own.

Gender=F

Moments

N	18	**Sum Weights**	18
Mean	3.61111111	**Sum Observations**	65
Std Deviation	0.63142126	**Variance**	0.39869281
Skewness	0.51682132	**Kurtosis**	-0.6481221
Uncorrected SS	241.5	**Corrected SS**	6.77777778
Coeff Variation	17.4855119	**Std Error Mean**	0.14882742

Gender=M

Moments

N	22	**Sum Weights**	22
Mean	3.75	**Sum Observations**	82.5
Std Deviation	0.86945522	**Variance**	0.75595238
Skewness	-0.5080564	**Kurtosis**	0.6958176
Uncorrected SS	325.25	**Corrected SS**	15.875
Coeff Variation	23.1854726	**Std Error Mean**	0.18536848

As we notice, we need to compare the two means, but we have different number of observations and different standard deviations. Fear not! SAS will do the computation for us immediately.

But,first we need to take care of that missing value otherwise SAS will not know which groups to compare. To do that we will create a new set called Qs3clean, by removing that observation without a gender entry as follows:

```
DATA Qs3Clean;
SET Qs3;
if Gender = ' ' then delete;
RUN;
```

As you can see the new dataset Qs3Clean is created by using the original set Qs3 using the if statement, that says if the variable gender is empty then delete it (very logical!)
Let's see how this new dataset looks like with proc print.

```
TITLE "Questionnaire 3 without missing Gender";
PROC PRINT Data=Qs3Clean;
RUN;
```

So now we can continue with the ttest analysis which is basically the following simple code:

```
TITLE "T-test comparison between Lectures for different genders";
ods graphics on;
PROC TTEST Data=Qs3Clean cochran ci=equal umpu;
class Gender;
var Lectures;
RUN;
ods graphics off;
```

Notice that we need to define the variable for which we are doing the comparison as class and the numerical variable as var (usual)
Also I turned on the graphics just in case I had turned it off at some point, to get all the graphs produced. Finally the commands Cochran ci=equal umpu instruct SAS to unclude the Cochran approximation of confidence intervals and ttest (another method of computation more "datadriven") and ci=equal tells sas to create an equal tailed confidence interval (default). The UMPU specifies an interval using the "best" variance.

There are various different outputs popping up. The following table contains the summary statistics for the two genders and their computed difference

Gender	N	Mean	Std Dev	Std Err	Minimum	Maximum
F	18	3.6111	0.6314	0.1488	3.0000	5.0000
M	22	3.7500	0.8695	0.1854	1.5000	5.0000
Diff (1-2)		-0.1389	0.7721	0.2454		

Then we get 95% confidence intervals for F, M and the difference if we used pooled standard errors or not (Satterthwaite method)

Gender	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev		95% UMPU CL Std Dev	
F		3.6111	3.2971	3.9251	0.6314	0.4738	0.9466	0.4654	0.9257
M		3.7500	3.3645	4.1355	0.8695	0.6689	1.2425	0.6593	1.2206
Diff (1-2)	**Pooled**	-0.1389	-0.6356	0.3579	0.7721	0.6310	0.9951	0.6258	0.9856
Diff (1-2)	**Satterthwaite**	-0.1389	-0.6203	0.3425					

Finally we get the corresponding p values for a two sided ttest comparison and again we get all cases including a new one:

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	38	-0.57	0.5747
Satterthwaite	Unequal	37.534	-0.58	0.5625
Cochran	Unequal	.	-0.58	0.5658

The pooled is the one we did in class and Satterhwaite is the standard with the uneven ones. Cochran, is another method of computing p-values using a completely different approximation that computes variances and standard errors through the data.
We will focus on the first two depending on what the problem calls for. In this case though any of the three is ok. And even if we did a one-sided hypothesis (halving all of them) the p values would be much bigger than 0.05, so we fail to reject the null hypothesis. In other words the dataset does not provide enough evidence to show a difference in understanding of the Lecture between genders.

Finally the table

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	21	17	1.90	0.1847

contains the F value and the corresponding p value for the comparison of the two variances. Notice that we also get the histograms for free with the QQ-plots in case we are wondering if the results follow a normal distribution.

To learn more about the ttest procedure follow this link:

https://support.sas.com/documentation/cdl/en/statug/63962/HTML/default/viewer.htm#statug_ttest_sect002.htm