OEIT6 - Data Analytics

Experiment 3: Analyze statistical data using Python

Name: Gaurav Panchal

UID: 2019120046

```
import pandas as pd
from scipy import stats
from statsmodels.stats import weightstats as stests
```

```
df[['bp_before','bp_after']].describe()
df.head(5)
```

	patient	sex	agegrp	bp_before	bp_after
0	1	Male	30-45	143	153
1	2	Male	30-45	163	170
2	3	Male	30-45	153	168
3	4	Male	30-45	153	142
4	5	Male	30-45	146	141

```
ttest,pval = stats.ttest_rel(df['bp_before'], df['bp_after'])
print(pval)
```

```
0.0011297914644840823
```

```
if pval<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

```
 reject null hypothesis
```

```
ztest ,pval1 = stests.ztest(df['bp_before'], x2=df['bp_after'], value=0,alternative='two-s
print(float(pval1))
```

```
0.002162306611369422
```

```
df_anova = pd.read_csv('PlantGrowth.csv')
df_anova = df_anova[['weight','group']]

grps = pd.unique(df_anova.group.values)
d_data = {grp:df_anova['weight'][df_anova.group == grp] for grp in grps}
```

```
F, p = stats.f_oneway(d_data['ctrl'], d_data['trt1'], d_data['trt2'])

print("p-value for significance is: ", p)

if p<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

```
 p-value for significance is:  0.0159099583256229
 reject null hypothesis
```

```
import statsmodels.api as sm
from statsmodels.formula.api import ols

df_anova2 = pd.read_csv("https://raw.githubusercontent.com/Opensourcefordatascience/Data-s
```

```
    model = ols('Yield ~ C(Fert)*C(Water)', df_anova2).fit()

    # Seeing if the overall model is significant
    print(f"Overall model F({model.df_model: .0f},{model.df_resid: .0f}) = {model.fvalue:
```

```
 Overall model F( 3, 16) =  4.112, p =  0.0243
```

```
model.summary()
```

OLS Regression Results

Dep. Variable:	Yield	**R-squared:**	0.435
Model:	OLS	**Adj. R-squared:**	0.330
Method:	Least Squares	**F-statistic:**	4.112
Date:	Mon, 21 Jan 2019	**Prob (F-statistic):**	0.0243
Time:	16:06:07	**Log-Likelihood:**	-50.996
No. Observations:	20	**AIC:**	110.0
Df Residuals:	16	**BIC:**	114.0
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	31.8000	1.549	20.527	0.000	28.516	35.084
C(Fert)[T.B]	-1.9600	2.191	-0.895	0.384	-6.604	2.684
C(Water)[T.Low]	-1.8000	2.191	-0.822	0.423	-6.444	2.844
C(Fert)[T.B]:C(Water)[T.Low]	-3.5200	3.098	-1.136	0.273	-10.088	3.048

Omnibus:	3.427	**Durbin-Watson:**	2.963
Prob(Omnibus):	0.180	**Jarque-Bera (JB):**	1.319
Skew:	-0.082	**Prob(JB):**	0.517
Kurtosis:	1.752	**Cond. No.**	6.85

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
res = sm.stats.anova_lm(model, typ= 2)
res
```

	sum_sq	df	F	PR(>F)
C(Fert)	69.192	1.0	5.766000	0.028847
C(Water)	63.368	1.0	5.280667	0.035386
C(Fert):C(Water)	15.488	1.0	1.290667	0.272656
Residual	192.000	16.0	NaN	NaN

```
df_chi = pd.read_csv('chi-test.csv')


contingency_table=pd.crosstab(df_chi["Gender"],df_chi["Like Shopping?"])
print('contingency_table :-\n',contingency_table)

     contingency_table :-
      Like Shopping?  No  Yes
     Gender
     Female           2    3
     Male             2    2


#Observed Values
Observed_Values = contingency_table.values
print("Observed Values :-\n",Observed_Values)

     Observed Values :-
      [[2 3]
      [2 2]]


b=stats.chi2_contingency(contingency_table)
Expected_Values = b[3]
print("Expected Values :-\n",Expected_Values)

     Expected Values :-
      [[2.22222222 2.77777778]
      [1.77777778 2.22222222]]


no_of_rows=len(contingency_table.iloc[0:2,0])
no_of_columns=len(contingency_table.iloc[0,0:2])
df11=(no_of_rows-1)*(no_of_columns-1)
print("Degree of Freedom:-",df)
alpha = 0.05

     Degree of Freedom:- 1



from scipy.stats import chi2
chi_square=sum([(o-e)**2./e for o,e in zip(Observed_Values,Expected_Values)])
chi_square_statistic=chi_square[0]+chi_square[1]
print("chi-square statistic:-",chi_square_statistic)
```

```
     [0.05 0.04]
     chi-square statistic:- 0.09000000000000008
```

```
critical_value=chi2.ppf(q=1-alpha,df=df11)
print('critical_value:',critical_value)
```

```
     critical_value: 3.841458820694124
```

```
#p-value
p_value=1-chi2.cdf(x=chi_square_statistic,df=df11)
print('p-value:',p_value)
```

```
     p-value: 0.7641771556220945
```

```
print('Significance level: ',alpha)
print('Degree of Freedom: ',df11)
print('chi-square statistic:',chi_square_statistic)
print('critical_value:',critical_value)
print('p-value:',p_value)
```

```
     Significance level:  0.05
     Degree of Freedom:  1
     chi-square statistic: 0.09000000000000008
     critical_value: 3.841458820694124
     p-value: 0.7641771556220945
```

```
if chi_square_statistic>=critical_value:
    print("Reject H0,There is a relationship between 2 categorical variables")
else:
    print("Retain H0,There is no relationship between 2 categorical variables")

if p_value<=alpha:
    print("Reject H0,There is a relationship between 2 categorical variables")
else:
    print("Retain H0,There is no relationship between 2 categorical variables")
```

```
     Retain H0,There is no relationship between 2 categorical variables
     Retain H0,There is no relationship between 2 categorical variables
```

Conclusion:

Ronald Coase said “*Torture the data, and it will confess to Anything*”. For that confession of data, Hypothesis Testing could be used to interpret and draw conclusions about the population using sample data. A Hypothesis Test helps in making a decision as to which mutually exclusive statement about the population is best supported by sample data.

Colab paid products - Cancel contracts here