Regression_ttest_review

April 15, 2021

[32]:
```
import pandas as pd
raw_data=pd.read_csv("FB_Data.csv")
raw_data=raw_data.dropna()
raw_data=raw_data[raw_data["Date"]!='6-Apr']
raw_data
```

[32]:

	Date	Advertisement	Daily_Reach	Daily_Likes	Cost_per_Result	Daily_Male_Likes	Daily_Female_Likes	Daily_Male_Reach
2	7-Apr	Photo	1214	48	0.61	11	37	406
3	7-Apr	Video	1711	55	0.53	21	34	595
4	8-Apr	Photo	1745	42	0.65	12	30	683
5	8-Apr	Video	2169	53	0.54	18	35	924
6	9-Apr	Photo	1299	39	0.70	7	32	465
7	9-Apr	Video	2543	57	0.55	14	43	984
8	10-Apr	Photo	1624	34	0.74	10	24	536
9	10-Apr	Video	4392	44	0.58	12	32	1850
10	11-Apr	Photo	3939	41	0.72	16	25	1693
11	11-Apr	Video	5652	55	0.58	15	40	2710
12	12-Apr	Photo	2347	31	0.75	10	21	1207
13	12-Apr	Video	3902	47	0.58	22	25	1821
14	13-Apr	Photo	1471	36	0.77	13	23	786
15	13-Apr	Video	3156	46	0.59	13	33	1606

	Daily_Female_Reach	18-34_Male	35-54_Male	55+_Male	18-34_Female	\
2	808	1	2	8	3	
3	1116	5	5	11	3	
4	1062	1	1	10	0	
5	1245	4	8	6	2	
6	834	1	1	5	1	
7	1559	2	4	8	2	
8	1088	0	3	7	0	
9	2542	4	3	5	3	
10	2246	2	2	12	1	
11	2942	6	6	3	5	
12	1140	2	3	5	2	
13	2081	5	3	14	3	
14	685	1	3	9	2	
15	1550	3	2	8	3	

	35-54_Female	55+_Female
2	3	31
3	4	27
4	0	30
5	7	26
6	1	30
7	2	39
8	2	22
9	4	25
10	1	23
11	7	28
12	1	18
13	2	20
14	1	20
15	2	28

[33]:
```
#Basic Comparison
data=raw_data
print("Photos")
print(data[data['Advertisement']=="Photo"].mean())
print("Videos")
print(data[data['Advertisement']=="Video"].mean())
```

```
Photos
Daily_Reach           1948.428571
Daily_Likes             38.714286
Cost_per_Result          0.705714
Daily_Male_Likes        11.285714
Daily_Female_Likes      27.428571
Daily_Male_Reach       825.142857
```

```
Daily_Female_Reach    1123.285714
18-34_Male               1.142857
35-54_Male               2.142857
55+_Male                 8.000000
18-34_Female             1.285714
35-54_Female             1.285714
55+_Female              24.857143
dtype: float64
Videos
Daily_Reach           3360.714286
Daily_Likes             51.000000
Cost_per_Result          0.564286
Daily_Male_Likes        16.428571
Daily_Female_Likes      34.571429
Daily_Male_Reach      1498.571429
Daily_Female_Reach    1862.142857
18-34_Male               4.142857
35-54_Male               4.428571
55+_Male                 7.857143
18-34_Female             3.000000
35-54_Female             4.000000
55+_Female              27.571429
dtype: float64
```

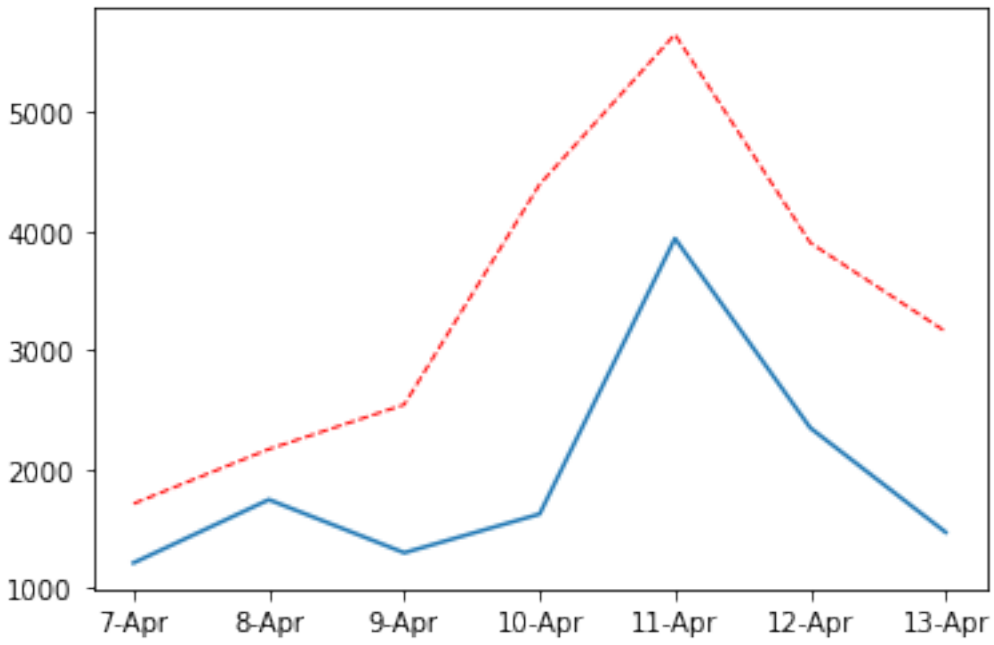
[34]:
```
from scipy.stats import ttest_ind, ttest_ind_from_stats
print(ttest_ind(data[data['Advertisement']=="Photo"]["Daily_Reach"],␣
 ↪data[data['Advertisement']=="Video"]["Daily_Reach"], equal_var=False,␣
 ↪nan_policy
='omit'))
```

```
Ttest_indResult(statistic=-2.2253642426851576, pvalue=0.04865621616362275)
```

1 Basic Analysis

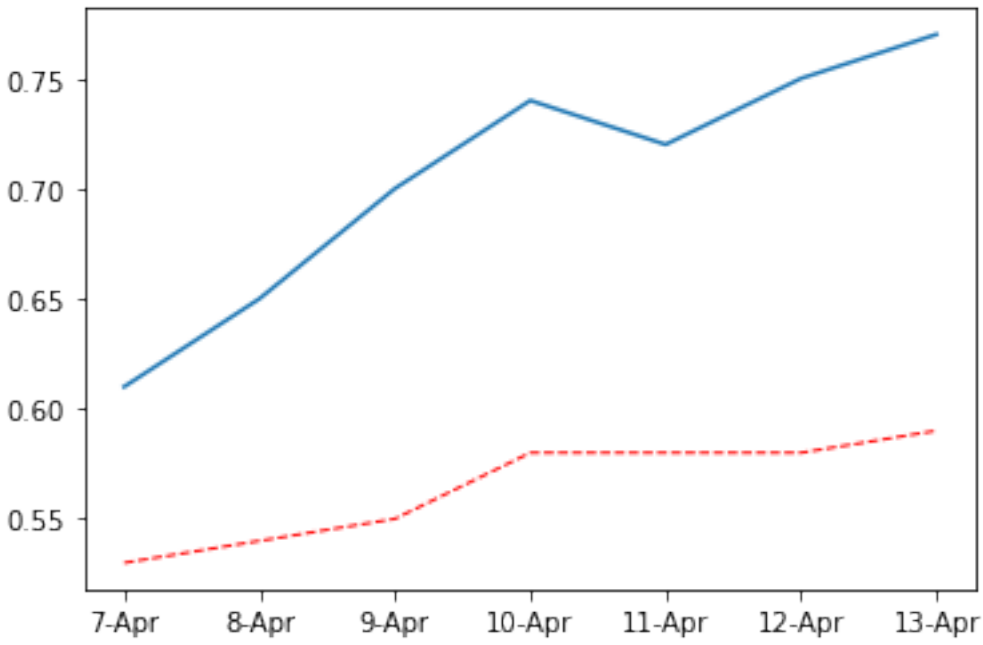
[35]:
```
import matplotlib.pyplot as plt
```

[36]:
```
plt.
 ↪plot(data[data['Advertisement']=="Photo"]['Date'],data[data['Advertisement']=="Photo"]["Daily
plt.
 ↪plot(data[data['Advertisement']=="Photo"]['Date'],data[data['Advertisement']=="Video"]["Daily
        linewidth=1.0,
        linestyle='--' )
plt.show()
```



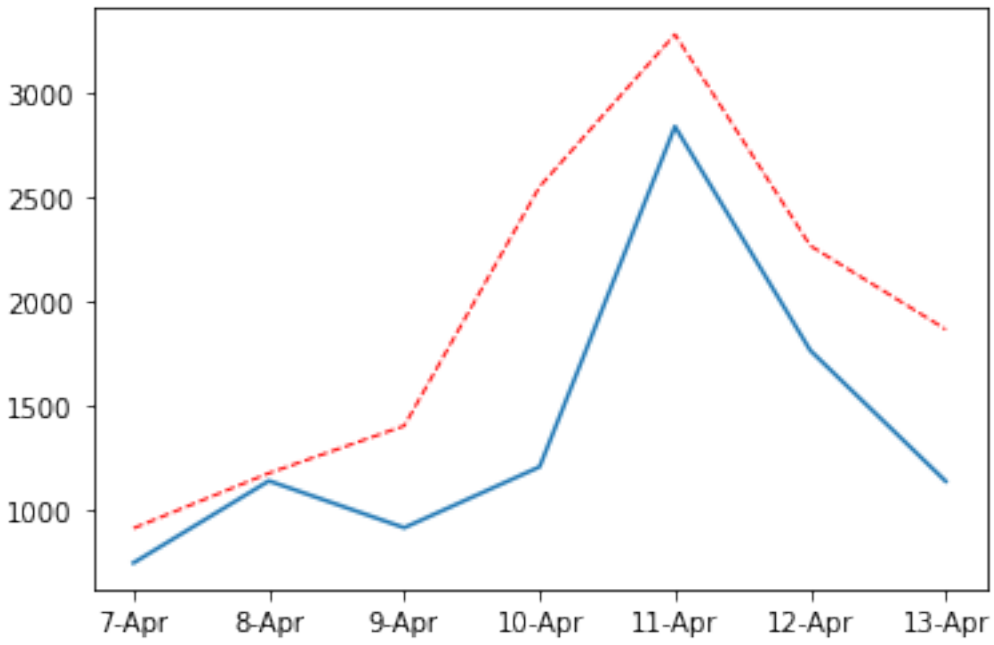
The conversion rate for photo is higher.

[37]:
```
plt.
 ↪plot(data[data['Advertisement']=="Photo"]['Date'],data[data['Advertisement']=="Photo"]["Cost_
plt.
 ↪plot(data[data['Advertisement']=="Photo"]['Date'],data[data['Advertisement']=="Video"]["Cost_
        linewidth=1.0,
        linestyle='--' )
plt.show()
```



The video has lower cost per reach.

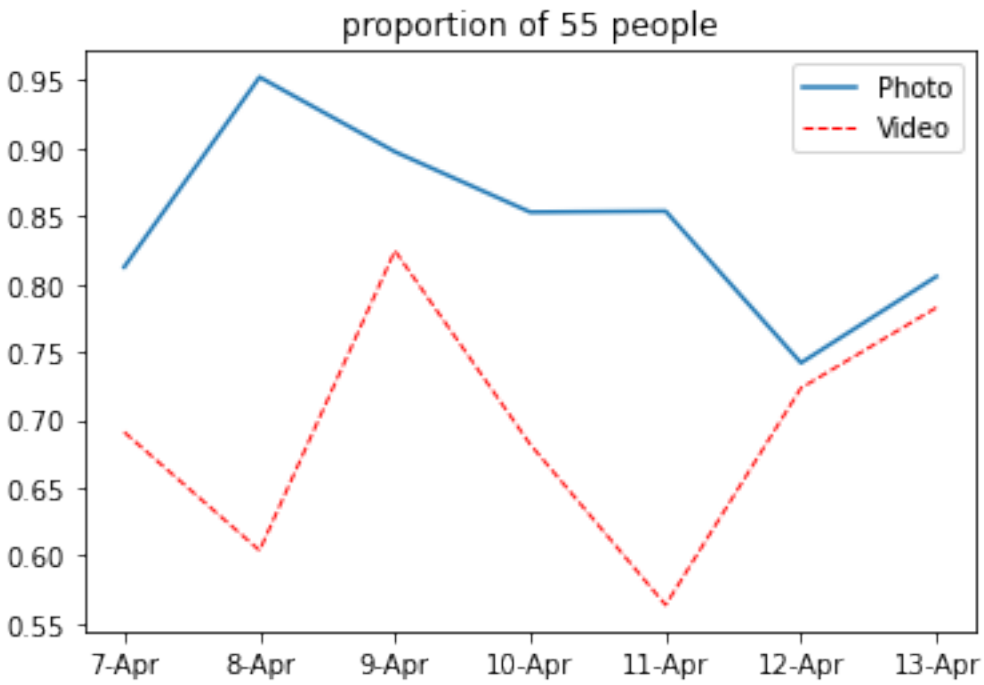
[38]:
```
data['interaction']=data['Cost_per_Result']*data['Daily_Reach']
plt.
 ↪plot(data[data['Advertisement']=="Photo"]['Date'],data[data['Advertisement']=="Photo"]["inter
plt.
 ↪plot(data[data['Advertisement']=="Photo"]['Date'],data[data['Advertisement']=="Video"]["inter
        linewidth=1.0,
        linestyle='--' )
plt.show()
```



From the picture, the video is better.

2 age distribution

[40]:
```
data['55_prop']=(data["55+_Female"]+data["55+_Male"])/data['Daily_Likes']
plt.
 ↪plot(data[data['Advertisement']=="Photo"]['Date'],data[data['Advertisement']=="Photo"]['55_pr
plt.
 ↪plot(data[data['Advertisement']=="Photo"]['Date'],data[data['Advertisement']=="Video"]['55_pr
        linewidth=1.0,
        linestyle='--' )
plt.legend()
plt.title("proportion of 55 people")
plt.show()
```



[41]:
```
print(ttest_ind(data[data['Advertisement']=="Photo"]["55_prop"],␣
 ↪data[data['Advertisement']=="Video"]["55_prop"], equal_var=False, nan_policy
='omit'))
```

```
Ttest_indResult(statistic=3.4513248113063186, pvalue=0.005396928249093133)
```

consider age , overall peopel above 55 clicks the like button most. There is a significant difference among proportion of liked across these two types of ads. The data is unstructured and mostly composed of 55 age or above.

3 gender distribution

[15]:
```
print(ttest_ind(data[data['Advertisement']=="Photo"]["Daily_Female_Reach"],␣
 ↪data[data['Advertisement']=="Video"]["Daily_Female_Reach"], equal_var=False,␣
 ↪nan_policy
='omit'))
```

```
Ttest_indResult(statistic=-2.2714049306284414, pvalue=0.0437341913520168)
```

[16]:
```
print(ttest_ind(data[data['Advertisement']=="Photo"]["Daily_Male_Reach"],␣
 ↪data[data['Advertisement']=="Video"]["Daily_Male_Reach"], equal_var=False,␣
 ↪nan_policy
='omit'))
```

```
Ttest_indResult(statistic=-2.0741028987973813, pvalue=0.06407199668531965)
```

The total number of reach and gender distribution is not similiar across different groups, which may shows further lack of externality.

[ ]:
```
data[data['Advertisement']=="Photo"]["Daily_Female_Reach"]
```

The distribution of ages in both groups are not the same.

[42]:
```
data['male_reach_prop']=data["Daily_Male_Reach"]/data["Daily_Reach"]
data['female_reach_prop']=data["Daily_Female_Reach"]/data["Daily_Reach"]
print(ttest_ind(data[data['Advertisement']=="Photo"]["male_reach_prop"],␣
 ↪data[data['Advertisement']=="Video"]["male_reach_prop"], equal_var=False,␣
 ↪nan_policy
='omit'))
```

```
Ttest_indResult(statistic=-0.5449183898347112, pvalue=0.5972355588137508)
```

4 Cost and likes

However, the distribution(proportion) of the gender for both group is roughly the same.

[22]:
```
print(ttest_ind(data[data['Advertisement']=="Photo"]["female_reach_prop"],␣
 ↪data[data['Advertisement']=="Video"]["female_reach_prop"], equal_var=False,␣
 ↪nan_policy
='omit'))
```

```
Ttest_indResult(statistic=0.5449183898347099, pvalue=0.5972355588137518)
```

[23]:
```
ttest_ind(data[data['Advertisement']=="Photo"]['Cost_per_Result'],␣
 ↪data[data['Advertisement']=="Video"]['Cost_per_Result'], equal_var=False,␣
 ↪nan_policy
='omit')
```

[23]:
```
Ttest_indResult(statistic=6.024948132556825, pvalue=0.00031634266018202624)
```

The difference of cost per reach among photo and video ads is statistically significant.

[43]:
```
import statsmodels.formula.api as smf

smf.ols("Cost_per_Result ~ Advertisement", data).fit().summary()
```

```
D:\anaconda3\lib\site-packages\scipy\stats\stats.py:1603: UserWarning:
kurtosistest only valid for n>=20 ... continuing anyway, n=14
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

[43]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Cost_per_Result   R-squared:                       0.752
Model:                            OLS   Adj. R-squared:                  0.731
Method:                 Least Squares   F-statistic:                     36.30
Date:                Fri, 16 Apr 2021   Prob (F-statistic):           5.98e-05
Time:                        01:03:16   Log-Likelihood:                 24.971
No. Observations:                  14   AIC:                            -45.94
Df Residuals:                      12   BIC:                            -44.66
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==========
                               coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
Intercept                    0.7057      0.017     42.517      0.000       0.670
0.742
Advertisement[T.Video]      -0.1414      0.023     -6.025      0.000      -0.193
-0.090
==============================================================================
Omnibus:                        2.611   Durbin-Watson:                   0.597
Prob(Omnibus):                  0.271   Jarque-Bera (JB):                1.235
Skew:                          -0.726   Prob(JB):                        0.539
Kurtosis:                       3.092   Cond. No.                         2.62
==============================================================================

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

The video has lower cost compared with photo and is significant.

[8]:
```
smf.ols("Cost_per_Result ~ Advertisement + Daily_Reach", data).fit().summary()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1603:
UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=14
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

[8]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:        Cost_per_Result   R-squared:                       0.789
Model:                            OLS   Adj. R-squared:                  0.750
```

```
Method:                 Least Squares   F-statistic:                     20.54
Date:                Thu, 15 Apr 2021   Prob (F-statistic):           0.000193
Time:                        12:57:45   Log-Likelihood:                 26.108
No. Observations:                  14   AIC:                            -46.22
Df Residuals:                      11   BIC:                            -44.30
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
==========
                               coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
Intercept                    0.6778      0.026     26.446      0.000       0.621
0.734
Advertisement[T.Video]      -0.1617      0.027     -6.017      0.000      -0.221
-0.103
Daily_Reach              1.432e-05   1.03e-05      1.393      0.191   -8.31e-06
3.7e-05
==============================================================================
Omnibus:                        1.600   Durbin-Watson:                   1.159
Prob(Omnibus):                  0.449   Jarque-Bera (JB):                0.378
Skew:                          -0.373   Prob(JB):                        0.828
Kurtosis:                       3.305   Cond. No.                     7.11e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 7.11e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

[54]:
```
data['likes_per_reach']=data['Daily_Likes']/data['Daily_Reach']
ttest_ind(data[data['Advertisement']=="Photo"]['likes_per_reach'],␣
 ↪data[data['Advertisement']=="Video"]['likes_per_reach'], equal_var=False,␣
 ↪nan_policy
='omit')
```

[54]:
```
Ttest_indResult(statistic=1.078598035077019, pvalue=0.3023813750812078)
```

[62]:
```
smf.ols("Cost_per_Result ~ Advertisement +likes_per_reach", data).fit().summary()
```

```
D:\anaconda3\lib\site-packages\scipy\stats\stats.py:1603: UserWarning:
kurtosistest only valid for n>=20 ... continuing anyway, n=14
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

[62]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:        Cost_per_Result   R-squared:                       0.874
Model:                            OLS   Adj. R-squared:                  0.851
Method:                 Least Squares   F-statistic:                     38.03
Date:                Fri, 16 Apr 2021   Prob (F-statistic):           1.15e-05
Time:                        01:20:55   Log-Likelihood:                 29.703
No. Observations:                  14   AIC:                            -53.41
Df Residuals:                      11   BIC:                            -51.49
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
==========
                               coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
Intercept                    0.7831      0.027     29.252      0.000       0.724
0.842
Advertisement[T.Video]      -0.1592      0.018     -8.692      0.000      -0.199
-0.119
likes_per_reach             -3.3315      1.022     -3.260      0.008      -5.581
-1.082
==============================================================================
Omnibus:                        1.707   Durbin-Watson:                   1.870
Prob(Omnibus):                  0.426   Jarque-Bera (JB):                0.335
Skew:                           0.308   Prob(JB):                        0.846
Kurtosis:                       3.441   Cond. No.                         134.
==============================================================================

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

The photo will reduce the cost per result, while the increased like per reach will reduce the cost per result too. The higher conversion rate(from view to like) means that the ads is more effective.

[55]:
```
smf.ols("likes_per_reach ~ Advertisement", data).fit().summary()
```

```
D:\anaconda3\lib\site-packages\scipy\stats\stats.py:1603: UserWarning:
kurtosistest only valid for n>=20 ... continuing anyway, n=14
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

[55]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:        likes_per_reach   R-squared:                       0.088
Model:                            OLS   Adj. R-squared:                  0.012
Method:                 Least Squares   F-statistic:                     1.163
Date:                Fri, 16 Apr 2021   Prob (F-statistic):              0.302
Time:                        01:07:46   Log-Likelihood:                46.792
No. Observations:                  14   AIC:                            -89.58
Df Residuals:                      12   BIC:                            -88.31
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==========
                               coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
Intercept                    0.0232      0.003      6.653      0.000       0.016
0.031
Advertisement[T.Video]      -0.0053      0.005     -1.079      0.302      -0.016
0.005
==============================================================================
Omnibus:                        0.823   Durbin-Watson:                   0.427
Prob(Omnibus):                  0.663   Jarque-Bera (JB):                0.770
Skew:                           0.398   Prob(JB):                        0.680
Kurtosis:                       2.173   Cond. No.                         2.62
==============================================================================

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

[61]:
```
smf.ols("likes_per_reach ~ male_reach_prop+female_reach_prop", data).fit().
↪summary()
```

```
D:\anaconda3\lib\site-packages\scipy\stats\stats.py:1603: UserWarning:
kurtosistest only valid for n>=20 ... continuing anyway, n=14
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

[61]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:        likes_per_reach   R-squared:                       0.381
```

```
Model:                            OLS   Adj. R-squared:                  0.330
Method:                 Least Squares   F-statistic:                     7.399
Date:                Fri, 16 Apr 2021   Prob (F-statistic):            0.0186
Time:                        01:15:15   Log-Likelihood:                 49.506
No. Observations:                  14   AIC:                            -95.01
Df Residuals:                      12   BIC:                            -93.73
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
=====
                       coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-----
Intercept            0.0095      0.002      4.583      0.001       0.005
0.014
male_reach_prop     -0.0369      0.016     -2.291      0.041      -0.072
-0.002
female_reach_prop    0.0463      0.015      3.189      0.008       0.015
0.078
==============================================================================
Omnibus:                        0.400   Durbin-Watson:                   1.002
Prob(Omnibus):                  0.819   Jarque-Bera (JB):                0.512
Skew:                           0.240   Prob(JB):                        0.774
Kurtosis:                       2.195   Cond. No.                     1.24e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 1.37e-31. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
"""
```

The coeffcient for proportion of male will decrease the conversion rate, while more female will increase the conversion rate.