Assignment 1_final

April 27, 2022

1 Preparation

[1]:
```
import pandas as pd
import statsmodels.api as sm
from statsmodels.stats.diagnostic import het_white
import numpy as np
```

[2]:
```
temp = pd.read_stata('Assignment_1(STAR).dta')
```

[3]:
```
df = temp[["dist_cod", "county", "district", "enrl_tot", "teachers",␣
↪"computer", "testscr", "comp_stu", "expn_stu", "str", "avginc", "el_pct",␣
↪"read_scr", "math_scr"]]
```

[4]:
```
df.head()
```

[4]:
```
   dist_cod   county                        district  enrl_tot   teachers  \
0   75119.0  Alameda              Sunol Glen Unified     195.0  10.900000
1   61499.0    Butte            Manzanita Elementary     240.0  11.150000
2   61549.0    Butte     Thermalito Union Elementary    1550.0  82.900002
3   61457.0    Butte  Golden Feather Union Elementary     243.0  14.000000
4   61523.0    Butte         Palermo Union Elementary    1335.0  71.500000

   computer     testscr  comp_stu     expn_stu        str     avginc  \
0      67.0  690.799988  0.343590  6384.911133  17.889910  22.690001
1     101.0  661.200012  0.420833  5099.380859  21.524664   9.824000
2     169.0  643.599976  0.109032  5501.954590  18.697226   8.978000
3      85.0  647.700012  0.349794  7101.831055  17.357143   8.978000
4     171.0  640.849976  0.128090  5235.987793  18.671329   9.080333

      el_pct    read_scr    math_scr
0   0.000000  691.599976  690.000000
1   4.583333  660.500000  661.900024
2  30.000002  636.299988  650.900024
3   0.000000  651.900024  643.500000
4  13.857677  641.799988  639.900024
```

2 Q1:

CLRM assumptions A1-A6: 1. Linearity in Parameters 2. Random Sampling 3. Variation in X 4. Zero conditional mean 5. Homoskedasticity 6. Normality

OLS estimators requirements:

1) unbiased: If our linear regression model follows A1-A4 it should be unbiased.

2) BLUE: A1-A5 If our linear regression model follows A1-A5 it should be BLUE.

3) BUE: A1-A6 If our linear regression model follows A1-A6 it should be BUE.

3 Q2:

[17]:
```
Y = df["testscr"]
X = df["str"]
X = sm.add_constant(X)
model = sm.OLS(Y, X)
results = model.fit()
```

[6]:
```
results.summary()
```

[6]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                testscr   R-squared:                       0.051
Model:                            OLS   Adj. R-squared:                  0.049
Method:                 Least Squares   F-statistic:                     22.58
Date:                Wed, 27 Apr 2022   Prob (F-statistic):           2.78e-06
Time:                        20:41:29   Log-Likelihood:                -1822.2
No. Observations:                 420   AIC:                             3648.
Df Residuals:                     418   BIC:                             3657.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        698.9330      9.467     73.825      0.000     680.323     717.543
str           -2.2798      0.480     -4.751      0.000      -3.223      -1.337
==============================================================================
Omnibus:                        5.390   Durbin-Watson:                   0.129
Prob(Omnibus):                  0.068   Jarque-Bera (JB):                3.589
Skew:                          -0.012   Prob(JB):                        0.166
Kurtosis:                       2.548   Cond. No.                         207.
==============================================================================

Notes:
```

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

$SE(\beta_0) = 9.467$

$SE(\beta_1) = 0.48$

4 Q3: β_1 estimates

4.1 t-statistic = -4.751

[7]:
```
results.tvalues
```

[7]:
```
const    73.824514
str      -4.751327
dtype: float64
```

4.2 p-value = 0.000

[8]:
```
results.pvalues
```

[8]:
```
const    6.569925e-242
str       2.783307e-06
dtype: float64
```

4.3 Inference

H_0: $\beta_1 = 0$

Since the p-value is less than 0.05, we reject the null hypothesis.

5 Q4:

HC0: White's (1980) heteroskedasticity robust standard errors

[9]:
```
results_hetero = model.fit(cov_type='HC0')
```

[10]:
```
results_hetero.summary()
```

[10]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                testscr   R-squared:                       0.051
Model:                            OLS   Adj. R-squared:                  0.049
Method:                 Least Squares   F-statistic:                     19.35
Date:                Wed, 27 Apr 2022   Prob (F-statistic):           1.38e-05
```

```
Time:                         20:41:30   Log-Likelihood:                -1822.2
No. Observations:                  420   AIC:                             3648.
Df Residuals:                      418   BIC:                             3657.
Df Model:                            1
Covariance Type:                   HC0
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        698.9330     10.340     67.597      0.000     678.668     719.198
str           -2.2798      0.518     -4.399      0.000      -3.296      -1.264
==============================================================================
Omnibus:                        5.390   Durbin-Watson:                   0.129
Prob(Omnibus):                  0.068   Jarque-Bera (JB):                3.589
Skew:                          -0.012   Prob(JB):                        0.166
Kurtosis:                       2.548   Cond. No.                         207.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
"""
```

$SE(\beta_0) = 10.340$

$SE(\beta_1) = 0.518$

Answer for question 3. won't change.

6 Q5:

R-square = 0.051

A low R-squared value indicates that the independent variable is not explaining much in the variation of your dependent variable regardless of the variable significance, this is letting you know that the identified independent variable, even though significant, is not accounting for much of the mean of your dependent variable. We may want to add more non-correlated independent variables to the model variables that some how relate to the dependent variable.

7 Q6:

[11]:
```
df["str"].describe()
```

[11]:
```
count    420.000000
mean      19.640427
std        1.891812
min       14.000000
25%       18.582360
50%       19.723208
75%       20.871815
```

```
max        25.799999
Name: str, dtype: float64
```

If we have an additional education district with a student teacher ratio of merely 5, the average test score will go up.

8 Q7:

[18]:
```
X_new = df[["str","avginc","expn_stu"]]
X_new = sm.add_constant(X_new)
model = sm.OLS(Y, X_new)
results_new = model.fit()
```

[13]:
```
results_new.summary()
```

[13]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                testscr   R-squared:                       0.519
Model:                            OLS   Adj. R-squared:                  0.516
Method:                 Least Squares   F-statistic:                     149.9
Date:                Wed, 27 Apr 2022   Prob (F-statistic):           7.65e-66
Time:                        20:41:30   Log-Likelihood:                -1679.4
No. Observations:                 420   AIC:                             3367.
Df Residuals:                     416   BIC:                             3383.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        669.7451     13.974     47.928      0.000     642.277     697.213
str           -1.3258      0.437     -3.035      0.003      -2.184      -0.467
avginc         1.8944      0.095     20.039      0.000       1.709       2.080
expn_stu      -0.0035      0.001     -2.616      0.009      -0.006      -0.001
==============================================================================
Omnibus:                        2.414   Durbin-Watson:                   0.693
Prob(Omnibus):                  0.299   Jarque-Bera (JB):                2.489
Skew:                          -0.165   Prob(JB):                        0.288
Kurtosis:                       2.819   Cond. No.                     1.16e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.16e+05. This might indicate that there are
```

```
strong multicollinearity or other numerical problems.
"""
```

- The coef of **str** increases from -2.2798 to -1.3258.
- R-square increases from 0.051 to 0.519.
- Adding more valuable x variables makes the prediction better(considering Adj. R-squared also improves lots).

9 Q8:

Tests under heteroskedasticity assumptions that avginc = expn_stu = 0

[14]:
```
B = np.array(([0,0,1,0],[0,0,0,1]))

print(results_new.f_test(B))
```

```
<F test: F=array([[202.60802797]]), p=3.6662878865406950-62, df_denom=416,
df_num=2>
```

Tests under heteroskedasticity assumptions that each coefficient is jointly statistically significantly different from zero.

[15]:
```
A = np.identity(len(results_new.params))
A = A[1:,:]

print(results_new.f_test(A))
```

```
<F test: F=array([[149.85594469]]), p=7.651663583855308e-66, df_denom=416,
df_num=3>
```

As a result, we can reject the null hypothesis that avginc=0 and expn_stu=0 since the p value is less than given significance level.