625.661 - Homework Two

Eric Niblock

February 22, 2022

1. **An article in the Journal of Pharmaceutical Sciences (80, 971 – 977, 1991) presents data on the observed mole fraction solubility of a solute at a constant temperature, along with x_1 = dispersion partial solubility, x_2 = dipolar partial solubility, and x_3 = hydrogen bonding Hansen partial solubility. The response y is the negative logarithm of the mole fraction solubility.**

   (a) **Fit a complete quadratic model to the data.**

   A complete quadratic model was fit to the data in the attached PDF. The result is:

$$\begin{aligned}\hat{y} =& -1.80 + 0.388x_1 + 0.207x_2 - 0.056x_3 - 0.015x_1^2 - 0.005x_2^2 \\ &+ 0.002x_3^2 - 0.020x_1x_2 + 0.001x_1x_3 - 0.001x_2x_3\end{aligned} \tag{1}$$

   (b) **Test for significance of regression, and construct t-statistics for each model parameter. Interpret these results.**

   A complete test for significance of regression has been performed in the attached PDF. We can see that all of the p-values are greater than 0.05, and therefore none are significant. The F-value of 21.12 is significant, and the regression is significant.

   (c) **Use the extra-sum-of-squares method to test the contribution of all second-order terms to the model.**

We have that,

$$SS_R(\beta_4,\beta_5,\beta_6,\beta_7,\beta_8,\beta_9|\beta_0,\beta_1,\beta_2,\beta_3) = SS_R(\beta_1,...,\beta_9|\beta_0) - SS_R(\beta_1,\beta_2,\beta_3|\beta_0) \tag{2}$$

And then,

$$F_0 = \frac{SS_R(\beta_4,...,\beta_9|\beta_0,...,\beta_3)}{6\,MS_{Res}} = 1.781 \tag{3}$$

With the calculation of the value given in the attached PDF. This value is not significant. Therefore we find that the introduction of second-order terms is not significant to the model.

2. **Consider the wine quality of young red wines data in Table B.19. Regressor x_1 is an indicator variable.**

   (a) **Use x_1 as the only regressor. Perform a regression analysis on your generated data.**

   The attached PDF shows the generation of the following regression analysis,

$$\hat{y} = 15.37 + 0.344x_1 \tag{4}$$

   (b) **Perform a 1-way analysis of variance on your generated data.**

   Our one-way ANOVA test for significance has revealed that the model is not significant. The results are provided in the attached PDF, with an F-value of 0.1877 and a corresponding p-value of 0.670.

3. **Use the data you generated in Problem 2 and include x_5 (wine color). Perform a thorough regression analysis of your generated data including the variables x_1, x_5, and y. Discuss the results and draw conclusions. State the assumptions for your analysis.**

   The attached PDF shows the generation of the following regression analysis,

   $$\hat{y} = 11.99 - 0.010x_1 + 0.792x_5 \tag{5}$$

   We assume that the residuals are normally distributed with a mean of zero and a constant standard deviation. Furthermore, we assume that x_1 and x_5 are not strongly correlated. Regression analysis reveals that x_1 is not significant, which was also shown to be true in the previous problem. However, we find that x_5 is highly significant and the model overall is significant given the provided F-value.

4. **Bonus Problem.**

   (a) **Create a numerical example to confirm that solving the minimization problem related to X and y is the same as solving the minimization problem related to D and U [*rephrased*].**

   A numerical example was created and provided in the attached PDF. It verifies that the generated linear models are identical regardless of if we use X and y or D and U. Additionally, the predicated value of y_0 is equivalent to the last determined coefficient of the model using D and U.

   (b) **Prove mathematically that the "Magnificent Dummy" method can generate an unbiased estimator of $E(y|x_0)$ and an unbiased estimator of the variance of the unbiased estimator of $E(y|x_0)$.**

   This problem was not attempted.

Problem 1: Data Loading and Selection

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
df = pd.DataFrame(np.array([[0.22200,    7.3,    0.0,    0.0 ],
[   0.39500,    8.7,    0.0,    0.3    ],
[    0.42200,    8.8,    0.7,    1.0    ],
[    0.43700 ,   8.1 ,   4.0,    0.2    ],
[    0.42800  ,  9.0,    0.5,    1.0    ],
[    0.46700,    8.7,    1.5,    2.8    ],
[    0.44400,    9.3,    2.1,    1.0    ],
[    0.37800,    7.6,    5.1,    3.4    ],
[    0.49400,    10.0,    0.0,    0.3    ],
[    0.45600,    8.4,    3.7,    4.1    ],
[    0.45200,    9.3,    3.6,    2.0    ],
[    0.11200,    7.7,    2.8,    7.1    ],
[    0.43200,    9.8,    4.2,    2.0   ],
[    0.10100,    7.3,    2.5,    6.8    ],
[    0.23200,    8.5,    2.0,    6.6   ],
[    0.30600,    9.5,    2.5,    5.0  ],
[    0.09230,    7.4,    2.8,    7.8  ],
[    0.11600,    7.8,    2.8,    7.7  ],
[    0.07640,    7.7,    3.0 ,   8.0   ],
[    0.43900,    10.3,    1.7,    4.2 ],
[    0.09440,    7.8,    3.3,    8.5  ],
[    0.11700,    7.1,    3.9,    6.6    ],
[    0.07260,    7.7,    4.3,    9.5   ],
[    0.04120,    7.4,    6.0,    10.9 ],
[    0.25100,    7.3,    2.0 ,   5.2  ],
[    0.00002,   7.6,    7.8,    20.7]]))
df.columns = ["y", "x1", "x2", "x3"]
```

In [3]:
```
n = 18

sample = df.sample(n)
Xtemp = np.array(sample[['x1','x2','x3']])
sample
```

Out[3]:

	y	x1	x2	x3
1	0.39500	8.7	0.0	0.3
0	0.22200	7.3	0.0	0.0
19	0.43900	10.3	1.7	4.2
4	0.42800	9.0	0.5	1.0
5	0.46700	8.7	1.5	2.8
25	0.00002	7.6	7.8	20.7
20	0.09440	7.8	3.3	8.5

	y	x1	x2	x3
14	0.23200	8.5	2.0	6.6
2	0.42200	8.8	0.7	1.0
6	0.44400	9.3	2.1	1.0
13	0.10100	7.3	2.5	6.8
17	0.11600	7.8	2.8	7.7
21	0.11700	7.1	3.9	6.6
10	0.45200	9.3	3.6	2.0
22	0.07260	7.7	4.3	9.5
3	0.43700	8.1	4.0	0.2
15	0.30600	9.5	2.5	5.0
16	0.09230	7.4	2.8	7.8

Problem 1: Part (a) and (b)

In [4]:
```
X = np.zeros((n,10))
X[:,0] = np.ones(n)
X[:,1:4] = Xtemp
X[:,4] = Xtemp[:,0]**2
X[:,5] = Xtemp[:,1]**2
X[:,6] = Xtemp[:,2]**2
X[:,7] = Xtemp[:,0]*Xtemp[:,1]
X[:,8] = Xtemp[:,0]*Xtemp[:,2]
X[:,9] = Xtemp[:,1]*Xtemp[:,2]
```

In [5]:
```
import statsmodels.api as sm

y = np.array(sample[['y']])
mod = sm.OLS(y, X)
resultsQ = mod.fit()
print(resultsQ.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.960
Model:                            OLS   Adj. R-squared:                  0.914
Method:                 Least Squares   F-statistic:                     21.12
Date:                Fri, 25 Feb 2022   Prob (F-statistic):           0.000119
Time:                        10:58:07   Log-Likelihood:                 36.011
No. Observations:                  18   AIC:                            -52.02
Df Residuals:                       8   BIC:                            -43.12
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.8023      1.185     -1.521      0.167      -4.536       0.931
x1             0.3875      0.278      1.394      0.201      -0.253       1.028
```

```
x2             0.2065      0.199      1.037      0.330      -0.253       0.666
x3            -0.0558      0.105     -0.533      0.609      -0.297       0.185
x4            -0.0150      0.016     -0.910      0.390      -0.053       0.023
x5            -0.0047      0.020     -0.236      0.819      -0.050       0.041
x6             0.0017      0.002      0.910      0.390      -0.003       0.006
x7            -0.0201      0.018     -1.110      0.299      -0.062       0.022
x8             0.0012      0.010      0.118      0.909      -0.023       0.025
x9            -0.0013      0.008     -0.156      0.880      -0.020       0.018
==============================================================================
Omnibus:                       22.612   Durbin-Watson:                   2.162
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               30.598
Skew:                           2.004   Prob(JB):                     2.27e-07
Kurtosis:                       7.973   Cond. No.                     1.42e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
ed.
[2] The condition number is large, 1.42e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
C:\Users\Eric\Anaconda3\lib\site-packages\scipy\stats\stats.py:1604: UserWarning: kurtos
istest only valid for n>=20 ... continuing anyway, n=18
  "anyway, n=%i" % int(n))
```

Part (c)

In [6]:

```
SSR1 = resultsQ.ess
MS_res = resultsQ.ssr/(n-10)
print('Sum of Squares Regression for Second Order Model:     ', SSR1)
print('MS_res for Second Order Model:                        ', MS_res)
```

```
Sum of Squares Regression for Second Order Model:      0.4580432326246042
MS_res for Second Order Model:                         0.002409850121924478
```

In [7]:

```
y = np.array(sample[['y']])
X = np.zeros((n,4))
X[:,0] = np.ones(n)
X[:,1:4] = Xtemp
mod = sm.OLS(y, X)
resultsL = mod.fit()
SSR2 = resultsL.ess
print('Sum of Squares Regression for First Order Model:     ', SSR2)
```

```
Sum of Squares Regression for First Order Model:      0.43228515574053106
```

In [8]:

```
F = abs((SSR2 - SSR1)/(6*MS_res))
print('F-value:   ',F)
```

```
F-value:    1.7814439059736398
```

Problem 2: Data Loading and Selection

In [197…

```
df = pd.read_excel(r'C:\Users\maste\Downloads\linear_regression_5e_data_sets\linear_reg
                   '\Appendices\data-table-B19.XLS')
```

In [198…

```
n = 20

sample = df.sample(n)
X = np.array(sample[' x_1 '])
y = np.array(sample['y'])
sample
```

Out[198…

	y	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_10
22	15.3	1	3.69	122	8.00	5.05	1.90	3.15	0.27	23	0.063
25	14.3	1	3.76	100	5.55	3.25	1.15	2.10	0.34	12	0.042
3	17.3	0	3.86	99	12.85	7.70	3.90	3.80	0.35	22	0.076
10	14.0	0	3.91	81	3.90	2.15	1.00	1.15	0.32	7	0.023
13	12.8	0	3.92	96	5.00	2.70	1.40	1.30	0.35	7	0.026
20	15.7	1	3.75	120	8.80	5.50	1.85	3.65	0.39	19	0.073
16	16.3	1	3.76	22	8.20	5.00	1.85	3.15	0.25	25	0.063
28	14.0	1	3.76	104	8.70	5.10	2.25	2.85	0.34	17	0.057
5	16.5	0	3.85	61	10.30	6.20	2.50	3.70	0.38	20	0.074
14	18.5	1	3.87	89	9.15	5.60	1.95	3.65	0.46	16	0.073
21	15.5	1	3.98	94	5.45	3.05	1.50	1.55	0.41	8	0.031
24	14.8	1	3.74	10	7.90	4.75	1.95	2.80	0.25	23	0.056
17	16.3	1	3.76	77	8.35	5.05	1.90	3.15	0.37	17	0.063
19	16.0	1	3.88	85	6.85	4.10	1.50	2.60	0.33	16	0.052
18	16.0	1	3.98	58	10.15	6.00	2.60	3.40	0.38	18	0.068
9	14.0	0	3.47	178	3.60	2.25	0.75	1.50	0.37	8	0.030
26	14.3	1	3.91	73	4.65	2.70	0.95	1.75	0.36	10	0.035
0	19.2	0	3.85	66	9.35	5.65	2.40	3.25	0.33	19	0.065
11	13.8	0	3.75	108	5.80	3.20	1.60	1.60	0.38	8	0.032
15	17.3	1	3.97	59	10.25	6.10	2.40	3.70	0.40	19	0.074

In [199…

```
X = sm.add_constant(X.T)
mod = sm.OLS(y, X)
results = mod.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.010
Model:                            OLS   Adj. R-squared:                 -0.045
Method:                 Least Squares   F-statistic:                    0.1877
Date:                Wed, 23 Feb 2022   Prob (F-statistic):              0.670
Time:                        08:48:03   Log-Likelihood:                -37.859
No. Observations:                  20   AIC:                             79.72
Df Residuals:                      18   BIC:                             81.71
```

```
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         15.3714      0.640     24.017      0.000      14.027      16.716
x1             0.3440      0.794      0.433      0.670      -1.324       2.012
==============================================================================
Omnibus:                        2.036   Durbin-Watson:                   2.494
Prob(Omnibus):                  0.361   Jarque-Bera (JB):                1.407
Skew:                           0.641   Prob(JB):                        0.495
Kurtosis:                       2.788   Cond. No.                         3.14
==============================================================================

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

Problem 3

In [200…

```
X = np.array(sample[[' x_1 ', ' x_5 ']])
y = np.array(sample['y'])
```

In [201…

```
X = sm.add_constant(X)
mod = sm.OLS(y, X)
results = mod.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.544
Model:                            OLS   Adj. R-squared:                  0.490
Method:                 Least Squares   F-statistic:                     10.13
Date:                Wed, 23 Feb 2022   Prob (F-statistic):            0.00127
Time:                        08:48:20   Log-Likelihood:                -30.117
No. Observations:                  20   AIC:                             66.23
Df Residuals:                      17   BIC:                             69.22
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         11.9948      0.880     13.636      0.000      10.139      13.851
x1            -0.0102      0.560     -0.018      0.986      -1.192       1.172
x2             0.7918      0.178      4.457      0.000       0.417       1.167
==============================================================================
Omnibus:                        3.452   Durbin-Watson:                   2.117
Prob(Omnibus):                  0.178   Jarque-Bera (JB):                1.786
Skew:                           0.692   Prob(JB):                        0.409
Kurtosis:                       3.475   Cond. No.                         17.0
==============================================================================

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

Bonus Problem

In [103…

```
## Random design matrix

X = np.zeros((20,4))
X[:,1:] = np.random.rand(20,3)
X[:,0] = np.ones(20)
X
```

Out[103…

```
array([[1.        , 0.56996571, 0.46408687, 0.65422843],
       [1.        , 0.52736153, 0.82396288, 0.80884639],
       [1.        , 0.35751224, 0.78162348, 0.30272197],
       [1.        , 0.80242513, 0.62653233, 0.64040214],
       [1.        , 0.14485958, 0.6235301 , 0.62807942],
       [1.        , 0.16817815, 0.01441573, 0.08797511],
       [1.        , 0.70064569, 0.6655653 , 0.22324781],
       [1.        , 0.07933594, 0.6871496 , 0.15452432],
       [1.        , 0.29633117, 0.15743266, 0.12315529],
       [1.        , 0.83807179, 0.0417141 , 0.17964486],
       [1.        , 0.28176226, 0.43043453, 0.65195485],
       [1.        , 0.59339496, 0.01212021, 0.2241093 ],
       [1.        , 0.10424003, 0.98504602, 0.91874039],
       [1.        , 0.16217311, 0.0605311 , 0.80698914],
       [1.        , 0.43028094, 0.74062269, 0.62712085],
       [1.        , 0.20573278, 0.78352811, 0.64986957],
       [1.        , 0.85048792, 0.48169398, 0.84734647],
       [1.        , 0.81298923, 0.0052779 , 0.18649551],
       [1.        , 0.32534837, 0.61101436, 0.42037858],
       [1.        , 0.97507466, 0.5703657 , 0.32206271]])
```

In [108…

```
## Random response matrix

y = np.random.rand(20)
y
```

Out[108…

```
array([0.9710332 , 0.51807184, 0.95035383, 0.60686395, 0.63126885,
       0.96507317, 0.89245471, 0.40897841, 0.76527396, 0.43031051,
       0.16397734, 0.84150824, 0.31283437, 0.0021449 , 0.25351408,
       0.25689649, 0.47285929, 0.97783217, 0.03923143, 0.41009382])
```

In [107…

```
## Random point, x0

x0 = np.random.rand(3)
x0
```

Out[107…

```
array([0.14444866, 0.23720879, 0.96802746])
```

In [112…

```
## Construction of matrix U

U = np.append(y,0)
U
```

Out[112…

```
array([0.9710332 , 0.51807184, 0.95035383, 0.60686395, 0.63126885,
       0.96507317, 0.89245471, 0.40897841, 0.76527396, 0.43031051,
       0.16397734, 0.84150824, 0.31283437, 0.0021449 , 0.25351408,
       0.25689649, 0.47285929, 0.97783217, 0.03923143, 0.41009382,
       0.        ])
```

In [117…

```
## Construction of matrix D

D = np.zeros((21,5))
D[:-1,:4] = X
D[-1,0] = 1
D[-1,1:4] = x0
D[-1,-1] = -1
D
```

Out[117…

```
array([[ 1.        ,  0.56996571,  0.46408687,  0.65422843,  0.        ],
       [ 1.        ,  0.52736153,  0.82396288,  0.80884639,  0.        ],
       [ 1.        ,  0.35751224,  0.78162348,  0.30272197,  0.        ],
       [ 1.        ,  0.80242513,  0.62653233,  0.64040214,  0.        ],
       [ 1.        ,  0.14485958,  0.6235301 ,  0.62807942,  0.        ],
       [ 1.        ,  0.16817815,  0.01441573,  0.08797511,  0.        ],
       [ 1.        ,  0.70064569,  0.6655653 ,  0.22324781,  0.        ],
       [ 1.        ,  0.07933594,  0.6871496 ,  0.15452432,  0.        ],
       [ 1.        ,  0.29633117,  0.15743266,  0.12315529,  0.        ],
       [ 1.        ,  0.83807179,  0.0417141 ,  0.17964486,  0.        ],
       [ 1.        ,  0.28176226,  0.43043453,  0.65195485,  0.        ],
       [ 1.        ,  0.59339496,  0.01212021,  0.2241093 ,  0.        ],
       [ 1.        ,  0.10424003,  0.98504602,  0.91874039,  0.        ],
       [ 1.        ,  0.16217311,  0.0605311 ,  0.80698914,  0.        ],
       [ 1.        ,  0.43028094,  0.74062269,  0.62712085,  0.        ],
       [ 1.        ,  0.20573278,  0.78352811,  0.64986957,  0.        ],
       [ 1.        ,  0.85048792,  0.48169398,  0.84734647,  0.        ],
       [ 1.        ,  0.81298923,  0.0052779 ,  0.18649551,  0.        ],
       [ 1.        ,  0.32534837,  0.61101436,  0.42037858,  0.        ],
       [ 1.        ,  0.97507466,  0.5703657 ,  0.32206271,  0.        ],
       [ 1.        ,  0.14444866,  0.23720879,  0.96802746, -1.        ]])
```

In [121…

```
## Linear regression using y and X

mod1 = sm.OLS(y, X)
results1 = mod1.fit()
print(results1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.300
Model:                            OLS   Adj. R-squared:                  0.169
Method:                 Least Squares   F-statistic:                     2.285
Date:                Tue, 22 Feb 2022   Prob (F-statistic):              0.118
Time:                        16:31:25   Log-Likelihood:                -1.5409
No. Observations:                  20   AIC:                             11.08
Df Residuals:                      16   BIC:                             15.06
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.6921      0.194      3.559      0.003       0.280       1.104
x1             0.2360      0.237      0.998      0.333      -0.265       0.737
x2             0.0171      0.246      0.070      0.945      -0.504       0.539
x3            -0.5618      0.279     -2.016      0.061      -1.152       0.029
==============================================================================
Omnibus:                        0.799   Durbin-Watson:                   1.151
Prob(Omnibus):                  0.671   Jarque-Bera (JB):                0.714
Skew:                          -0.141   Prob(JB):                        0.700
Kurtosis:                       2.118   Cond. No.                         6.44
==============================================================================
```

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

In [122…

```python
## Linear regression using U and D

mod2 = sm.OLS(U, D)
results2 = mod2.fit()
print(results2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.388
Model:                            OLS   Adj. R-squared:                  0.235
Method:                 Least Squares   F-statistic:                     2.537
Date:                Tue, 22 Feb 2022   Prob (F-statistic):             0.0806
Time:                        16:31:26   Log-Likelihood:                -1.1057
No. Observations:                  21   AIC:                             12.21
Df Residuals:                      16   BIC:                             17.43
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.6921      0.194      3.559      0.003       0.280       1.104
x1             0.2360      0.237      0.998      0.333      -0.265       0.737
x2             0.0171      0.246      0.070      0.945      -0.504       0.539
x3            -0.5618      0.279     -2.016      0.061      -1.152       0.029
x4             0.1865      0.355      0.525      0.607      -0.566       0.939
==============================================================================
Omnibus:                        0.521   Durbin-Watson:                   1.236
Prob(Omnibus):                  0.771   Jarque-Bera (JB):                0.600
Skew:                          -0.144   Prob(JB):                        0.741
Kurtosis:                       2.224   Cond. No.                         8.55
==============================================================================

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

In [137…

```python
## y0 by use of model 1

y0 = results1.params@np.append(1,x0).T
y0
```

Out[137… 0.18646883407741743