Jacob Miller - Homework 3

February 20, 2020

1 Setup

[1]:
```
import pandas as pd
import numpy as np
import patsy
import scipy
import statsmodels.api as sm
from astropy.table import Table
from sympy import symbols
```

[2]:
```
def title_print(text):
    '''
    Used throughout to print section titles
    '''
    text_len = len(text)
    print()
    print('#' * (text_len + 4))
    print('|', text, '|')
    print('#' * (text_len + 4))
```

2 Problem 3.1

[3]:
```
df = pd.read_excel('Data/data-table-B1.xlsx')
df = df.rename(columns = {'y': 'Games_won',
                          'x1': 'Rushing_yards',
                          'x2': 'Passing_yards',
                          'x3': 'Punting_average',
                          'x4': 'Field_goal_percentage',
                          'x5': 'Turnover_differential',
                          'x6': 'Penalty_yards',
                          'x7': 'Percent_rushing',
                          'x8': 'Opponent_rushing_yards',
                          'x9': 'Opponent_passing_yards'})
```

[4]:
```python
title_print('Problem 3.1a')
y, X = patsy.dmatrices('Games_won ~ Passing_yards + Percent_rushing + \
                        Opponent_rushing_yards', df)

parens = np.matmul(X.T, X)
Xs = np.matmul(np.linalg.inv(parens), X.T)
b_hat = np.round(np.matmul(Xs, y), 4)

print('y_hat = {} + {} * x_2 + {} * x_7 + {} * x_8'.format(b_hat[0],
                                                           b_hat[1],
                                                           b_hat[2],
                                                           b_hat[3]))
```

```
################
| Problem 3.1a |
################
y_hat = [-1.8084] + [0.0036] * x_2 + [0.194] * x_7 + [-0.0048] * x_8
```

[5]:
```python
title_print('Problem 3.1b')
results = sm.OLS(y, X).fit()
results.model.data.design_info = X.design_info

# Note statsmodels prints out ANOVA for each individual regressor
aov_table = sm.stats.anova_lm(results, typ = 1)

print(results.summary())
print('\n--- Analysis of Variance table ---\n{}'.format(aov_table))
print('\nRegression F: {}'.format(round(results.fvalue, 2)))
print('Regression p: {}'.format(round(results.f_pvalue, 4)))
print('\n--> Regression is significant <--')
```

```
################
| Problem 3.1b |
################
                            OLS Regression Results
==============================================================================
Dep. Variable:              Games_won   R-squared:                       0.786
Model:                            OLS   Adj. R-squared:                  0.760
Method:                 Least Squares   F-statistic:                     29.44
Date:                Thu, 20 Feb 2020   Prob (F-statistic):           3.27e-08
Time:                        17:27:18   Log-Likelihood:                -52.532
No. Observations:                  28   AIC:                             113.1
Df Residuals:                      24   BIC:                             118.4
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
```

```
==========
                            coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
----------
Intercept                -1.8084      7.901     -0.229      0.821     -18.115
14.498
Passing_yards             0.0036      0.001      5.177      0.000       0.002
0.005
Percent_rushing           0.1940      0.088      2.198      0.038       0.012
0.376
Opponent_rushing_yards   -0.0048      0.001     -3.771      0.001      -0.007
-0.002
==============================================================================
Omnibus:                       0.665   Durbin-Watson:                   1.492
Prob(Omnibus):                 0.717   Jarque-Bera (JB):                0.578
Skew:                          0.321   Prob(JB):                        0.749
Kurtosis:                      2.712   Cond. No.                     7.42e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 7.42e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

--- Analysis of Variance table ---
                          df      sum_sq     mean_sq          F        PR(>F)
Passing_yards            1.0   76.193400   76.193400  26.172055  3.100132e-05
Percent_rushing          1.0  139.500820  139.500820  47.917840  3.697874e-07
Opponent_rushing_yards   1.0   41.400062   41.400062  14.220716  9.377699e-04
Residual                24.0   69.870004    2.911250        NaN           NaN

Regression F: 29.44
Regression p: 0.0

--> Regression is significant <--
```

[6]:
```
title_print('Problem 3.1c')
t_stat = -scipy.stats.t.ppf(0.025, len(X) - 2)
t_values = abs(np.round(results.tvalues[1:], 3))
p_values = np.round(results.pvalues[1:], 3)
table = Table([['B2', 'B7', 'B8'], t_values, p_values],
              names = ('Coef', 't_0', 'p-value'))

print(table)
print('\nt-statistic = {}'.format(round(t_stat, 3)))
```

```
print('\n--> abs(t_0) > t-statistic, so all are significant <--')
```

```
################
| Problem 3.1c |
################
Coef   t_0  p-value
---- ------ -------
  B2 5.177     0.0
  B7 2.198   0.038
  B8 3.771   0.001

t-statistic = 2.056

--> abs(t_0) > t-statistic, so all are significant <--
```

[7]:
```
title_print('Problem 3.1d')
print('R^2 = {}%'.format(round(100 * results.rsquared, 2)))
print('Adj-R^2 = {}%'.format(round(100 * results.rsquared_adj, 2)))
```

```
################
| Problem 3.1d |
################
R^2 = 78.63%
Adj-R^2 = 75.96%
```

[8]:
```
title_print('Problem 3.1e')
y2, X2 = patsy.dmatrices('y ~ Passing_yards + Opponent_rushing_yards', df)

parens2 = np.matmul(X2.T, X2)
Xs2 = np.matmul(np.linalg.inv(parens2), X2.T)
b_hat2 = np.round(np.matmul(Xs2, y2), 4)

results2 = sm.OLS(y2, X2).fit()
results2.model.data.design_info = X2.design_info

partial_F = round((results.ess - results2.ess) / results.mse_resid, 2)

print('reduced y_hat = {} + {} * x_2 + {} * x_8'.format(b_hat2[0],
                                                        b_hat2[1],
                                                        b_hat2[2]))
print('partial F = {}'.format(partial_F))
print('\n--> {} < {}, therefore B7 is significant <--'.format(partial_F,
      round(results.fvalue, 2)))
```

```
################
```

```
| Problem 3.1e |
################
reduced y_hat = [14.7127] + [0.0031] * x_2 + [-0.0068] * x_8
partial F = 4.83

--> 4.83 < 29.44, therefore B7 is significant <--
```

3 Problem 3.10

[9]:
```
df = pd.read_excel('Data/data-table-B11.xlsx')
```

```
WARNING *** OLE2 inconsistency: SSCS size is 0 but SSAT size is non-zero
```

[10]:
```
title_print('Problem 3.10a')
y, X = patsy.dmatrices('Quality ~ Clarity + Aroma + Body + Flavor + \
                        Oakiness', df)

parens = np.matmul(X.T, X)
Xs = np.matmul(np.linalg.inv(parens), X.T)
b_hat = np.round(np.matmul(Xs, y), 4)

print('y_hat = {} + {} * x_1 + {} * x_2 + {} * x_3 + {} * x_4 + {} * x_5'.\
      format(b_hat[0], b_hat[1], b_hat[2], b_hat[3], b_hat[4], b_hat[5]))
```

```
#################
| Problem 3.10a |
#################
y_hat = [3.9969] + [2.3395] * x_1 + [0.4826] * x_2 + [0.2732] * x_3 + [1.1683] *
x_4 + [-0.684] * x_5
```

[11]:
```
title_print('Problem 3.10b')
results = sm.OLS(y, X).fit()
results.model.data.design_info = X.design_info

aov_table = sm.stats.anova_lm(results, typ = 1)

print(results.summary())
print('\n--- Analysis of Variance table ---\n{}'.format(aov_table))
print('\nRegression F: {}'.format(round(results.fvalue, 2)))
print('Regression p: {}'.format(round(results.f_pvalue, 4)))
print('\n--> Regression is significant <--')
```

```
#################
| Problem 3.10b |
```

```
#################
                            OLS Regression Results
==============================================================================
Dep. Variable:                Quality   R-squared:                       0.721
Model:                            OLS   Adj. R-squared:                  0.677
Method:                 Least Squares   F-statistic:                     16.51
Date:                Thu, 20 Feb 2020   Prob (F-statistic):           4.70e-08
Time:                        17:30:51   Log-Likelihood:                -56.378
No. Observations:                  38   AIC:                             124.8
Df Residuals:                      32   BIC:                             134.6
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      3.9969      2.232      1.791      0.083      -0.549       8.543
Clarity        2.3395      1.735      1.349      0.187      -1.194       5.873
Aroma          0.4826      0.272      1.771      0.086      -0.072       1.038
Body           0.2732      0.333      0.821      0.418      -0.404       0.951
Flavor         1.1683      0.304      3.837      0.001       0.548       1.789
Oakiness      -0.6840      0.271     -2.522      0.017      -1.236      -0.132
==============================================================================
Omnibus:                        1.181   Durbin-Watson:                   0.837
Prob(Omnibus):                  0.554   Jarque-Bera (JB):                1.020
Skew:                          -0.384   Prob(JB):                        0.601
Kurtosis:                       2.770   Cond. No.                         134.
==============================================================================

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

--- Analysis of Variance table ---
            df     sum_sq    mean_sq          F        PR(>F)
Clarity    1.0   0.125210   0.125210   0.092645  7.628120e-01
Aroma      1.0  77.353210  77.353210  57.235072  1.286336e-08
Body       1.0   6.414421   6.414421   4.746149  3.684165e-02
Flavor     1.0  19.049819  19.049819  14.095314  6.945740e-04
Oakiness   1.0   8.597755   8.597755   6.361638  1.683272e-02
Residual  32.0  43.248006   1.351500        NaN           NaN

Regression F: 16.51
Regression p: 0.0

--> Regression is significant <--
```

[12]:
```
title_print('Problem 3.10c')
t_stat = -scipy.stats.t.ppf(0.025, len(X) - 2)
t_values = abs(np.round(results.tvalues[1:], 3))
p_values = np.round(results.pvalues[1:], 3)
table = Table([['B1', 'B2', 'B3', 'B4', 'B5'], t_values, p_values],
              names = ('Coef', 't_0', 'p-value'))

print(table)
print('\nt-statistic = {}'.format(round(t_stat, 3)))
print('\n--> B4, B5: abs(t_0) > t-statistic so these are significant <--')
```

```
#################
| Problem 3.10c |
#################
Coef  t_0  p-value
---- ----- -------
  B1 1.349   0.187
  B2 1.771   0.086
  B3 0.821   0.418
  B4 3.837   0.001
  B5 2.522   0.017

t-statistic = 2.028

--> B4, B5: abs(t_0) > t-statistic so these are significant <--
```

[13]:
```
title_print('Problem 3.10d')
y2, X2 = patsy.dmatrices('Quality ~ Aroma + Flavor', df)

parens = np.matmul(X2.T, X2)
Xs2 = np.matmul(np.linalg.inv(parens), X2.T)
b_hat2 = np.round(np.matmul(Xs2, y2), 4)
results2 = sm.OLS(y2, X2).fit()
results2.model.data.design_info = X2.design_info

table = Table([['R^2', 'Adj-R^2'],
               [round(100 * results.rsquared, 2),
                round(100 * results.rsquared_adj, 2)],
               [round(100 * results2.rsquared, 2),
                round(100 * results2.rsquared_adj, 2)]],
               names = (' ', 'Full model', 'Reduced model'))
print(table)
print('\n--> Very similar, so models are similar <--')
```

```
#################
| Problem 3.10d |
```

```
#################
        Full model Reduced model
------- ---------- -------------
    R^2      72.06         65.86
Adj-R^2      67.69          63.9

--> Very similar, so models are similar <--
```

[14]:
```
title_print('Problem 3.10e')
ci_1 = np.round(results.conf_int()[4], 3)
ci_2 = np.round(results2.conf_int()[2], 3)

print('Full model: {} to {}'.format(ci_1[0], ci_1[1]))
print('Reduced model: {} to {}'.format(ci_2[0], ci_2[1]))
print('\n--> Very similar again, so similar models <--')
```

```
#################
| Problem 3.10e |
#################
Full model: 0.548 to 1.789
Reduced model: 0.58 to 1.76

--> Very similar again, so similar models <--
```

4 Problem 3.25 (Note: This is problem 3.21 in 4th edition of textbook)

[15]:
```
title_print('Problem 3.25a')
b, b0, b1, b2, b3, b4 = symbols('b b0 b1 b2 b3 b4')
y, x1, x2, x3, x4, eps = symbols('y x1 x2 x3 x4 eps')
gamma_0, gamma_1, z = symbols('gamma_0 gamma_1 z')

beta = np.array([[b0], [b1], [b2], [b3], [b4]])
X = np.array([1, x1, x2, x3, x4])
y = np.matmul(X, beta) + eps

# H0: b1 = b2 = b3 = b4
beta2 = np.array([[b0], [b1], [b1], [b1], [b1]])
y2 = np.matmul(X, beta2) + eps

T = np.array([[0, 1, -1, 0, 0],
              [0, 0, 1, -1, 0],
              [0, 0, 0, 1, -1]])
c = np.array([[0], [b], [b], [b], [b]])
```

```python
print('y = {}'.format(y))
print('H0: b1 = b2 = b3 = b4 = b')
print('\nTherefore: b1 - b2 = 0, b2 - b3 = 0, b3 - b4 = 0')
print('\nT = \n{}\n\nbeta = \n{}\n\nc = \n{}'.format(T, beta, c))
print('\ny = {}'.format(y2))
print('\nWhere:\ngamma_0 = b0\ngamma_1 = b\nz = x1 + x2 + x3 + x4')
print('\n--> Reduced model: y = gamma_0 + gamma_1 * z + eps <--')
```

```
#################
| Problem 3.25a |
#################
y = [b0 + b1*x1 + b2*x2 + b3*x3 + b4*x4 + eps]
H0: b1 = b2 = b3 = b4 = b

Therefore: b1 - b2 = 0, b2 - b3 = 0, b3 - b4 = 0

T =
[[ 0  1 -1  0  0]
 [ 0  0  1 -1  0]
 [ 0  0  0  1 -1]]

beta =
[[b0]
 [b1]
 [b2]
 [b3]
 [b4]]

c =
[[0]
 [b]
 [b]
 [b]
 [b]]

y = [b0 + b1*x1 + b1*x2 + b1*x3 + b1*x4 + eps]

Where:
gamma_0 = b0
gamma_1 = b
z = x1 + x2 + x3 + x4

--> Reduced model: y = gamma_0 + gamma_1 * z + eps <--
```

```
[16]: title_print('Problem 3.25b')

beta = np.array([[b0], [b1], [b2], [b3], [b4]])
X = np.array([1, x1, x2, x3, x4])
y = np.matmul(X, beta) + eps

# H0: b1 = b2, b3 = b4
beta2 = np.array([[b0], [b1], [b1], [b3], [b3]])
y2 = np.matmul(X, beta2) + eps

T = np.array([[0, 1, -1, 0, 0],
              [0, 0, 0, 1, -1]])
c = np.array([[0], [0]])

print('y = {}'.format(y))
print('H0: b1 = b2, b3 = b4')
print('\nTherefore: b1 - b2 = 0, b3 - b4 = 0')
print('\nT = \n{}\n\nbeta = \n{}\n\nc = \n{}'.format(T, beta, c))
print('\ny = {}'.format(y2))
print('\nWhere:\ngamma_0 = b0\ngamma_1 = b1\ngamma_3 = b3')
print('z1 = x1 + x2\nz3 = x3 + x4')
print('\n--> Reduced model: y = gamma_0 + gamma_1 * z1 + gamma_3 * z3 <--')
```

```
#################
| Problem 3.25b |
#################
y = [b0 + b1*x1 + b2*x2 + b3*x3 + b4*x4 + eps]
H0: b1 = b2, b3 = b4

Therefore: b1 - b2 = 0, b3 - b4 = 0

T =
[[ 0  1 -1  0  0]
 [ 0  0  0  1 -1]]

beta =
[[b0]
 [b1]
 [b2]
 [b3]
 [b4]]

c =
[[0]
 [0]]
```

```
y = [b0 + b1*x1 + b1*x2 + b3*x3 + b3*x4 + eps]

Where:
gamma_0 = b0
gamma_1 = b1
gamma_3 = b3
z1 = x1 + x2
z3 = x3 + x4

--> Reduced model: y = gamma_0 + gamma_1 * z1 + gamma_3 * z3 <--
```