Jacob Miller - Homework 2

February 6, 2020

[1]:
```
import numpy as np
from scipy import stats
```

[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))
```

[3]:
```
##############
# Problem 1 #
##############
X = np.array([[-8], [-4], [0], [4], [8]])
X = np.concatenate((np.ones(X.shape), X), axis = 1)
Y = np.array([[11.7], [11], [10.2], [9], [7.8]])
```

[4]:
```
#################
# Calculations #
#################
# Constants
alpha = 0.05
n = len(Y)
t_stat = stats.t.ppf(alpha / 2, len(Y) - 2)
x_hat = np.mean(X[:, 1])

# SS_T = sum(y_i**2) - (sum(y_i)**2) / n
sum_y_squared = sum(Y ** 2)
squared_sum_y = sum(Y) ** 2
SS_T = sum_y_squared - squared_sum_y / n

# S_xx = sum(x_i**2) - sum(x_i)**2 / n
sum_xx = sum(X[:, 1] ** 2)
sum_x_squared = (sum(X[:, 1])) ** 2
```

```
S_xx = sum_xx - sum_x_squared / n

# S_xy = sum(y_i * x_i) - sum(y_i) * sum(x_i) / n
sum_xy = sum(X[:, 1].reshape(-1, 1) * Y)
sum_x_sum_y = sum(X[:, 1]) * sum(Y)
S_xy = sum_xy - sum_x_sum_y / n
```

[5]:
```
################
# Problem 1.a #
################
title_print('Problem 1.a')

X_inv = np.linalg.inv(np.matmul(X.T, X))
b_hat = np.matmul(np.matmul(X_inv, X.T), Y)

print(b_hat)
```

```
################
| Problem 1.a |
################
[[ 9.94 ]
 [-0.245]]
```

[6]:
```
################
# Problem 1.b #
################
title_print('Problem 1.b')

# SS_res = SS_T - B_hat[1] * S_xy
SS_res = SS_T - b_hat[1] * S_xy

# MS_res (variance) = SS_Res / (n-2)
MS_res = SS_res / (n - 2)

var_cov = MS_res * X_inv

print(var_cov)
```

```
################
| Problem 1.b |
################
[[0.00986667 0.        ]
 [0.         0.00030833]]
```

[7]:
```
################
# Problem 1.c #
################
title_print('Problem 1.c')

X_0 = -6
y_0 = b_hat[0] + b_hat[1] * X_0

print(y_0)
```

```
################
| Problem 1.c |
################
[11.41]
```

[8]:
```
################
# Problem 1.d #
################
title_print('Problem 1.d')

X_h = np.concatenate((np.ones(1), np.array([X_0])), axis = 0)
inside = np.matmul(np.matmul(X_h.T, X_inv), X_h)
var_y = MS_res * inside

print(var_y)
```

```
################
| Problem 1.d |
################
[0.02096667]
```

[9]:
```
################
# Problem 1.e #
################
title_print('Problem 1.e')

# Prediction interval - note t_stat is negative so ranges are flipped
parens = 1 + 1/n + (X_0 - x_hat) ** 2 / S_xx
constant = t_stat * np.sqrt(MS_res * parens)

y_low = y_0 + constant
y_high = y_0 - constant

print('-> Prediction Interval <-\n{} <= y_0 <= {}'.format(y_low, y_high))
```

```
###############
| Problem 1.e |
###############
-> Prediction Interval <-
[10.5662015] <= y_0 <= [12.2537985]
```

[10]:
```
##############
# Problem 2 #
##############
title_print('Problem 2')

print('Recall b_hat:\n{}'.format(b_hat))
print('\n(X\')^(-1):\n{}'.format(X_inv))
print('\n(X\')^(-1) * X\':\n{}'.format(np.matmul(X_inv, X.T)))
print('\n(X\')^(-1) * X\' * y:\n{}'.format(np.matmul(
                                          np.matmul(X_inv, X.T),
                                          Y)))
```

```
##############
| Problem 2 |
##############
Recall b_hat:
[[ 9.94 ]
 [-0.245]]

(X')^(-1):
[[0.2     0.     ]
 [0.      0.00625]]

(X')^(-1) * X':
[[ 0.2    0.2    0.2    0.2    0.2  ]
 [-0.05  -0.025  0.     0.025  0.05 ]]

(X')^(-1) * X' * y:
[[ 9.94 ]
 [-0.245]]
```