# Linear Regression Homework 5

**Name**: Eric Yuan

**UNI**: qy2205

## 1. Covariance matrix of $Y = AX$

Since $\boldsymbol{A}$ is a constant $\boldsymbol{k} \times \boldsymbol{n}$ matrix. So we have $\boldsymbol{E}[\boldsymbol{AX}] = \boldsymbol{AE}[\boldsymbol{X}]$

$\mathrm{Var[AX]} = \mathrm{E[(AX - E(AX))(AX - E[AX])^T]}$

$= \mathrm{E[(AX - E(AX))(AX - E[AX])^T]}$

$= \mathrm{E[(A(X - E(X))(X - E[X])^T A^T]}$

$= \mathrm{AE[(X - E(X))(X - E[X])^T]A^T}$

$= \mathrm{AVar[X]A^T}$

$= \mathrm{A\Sigma A^T}$

## 2. Show that $\mathbf{t-test}$ and $\mathbf{F-test}$ are equivalent in the sense that the $\mathbf{T^2 = F}$ where $\mathbf{T}$ is the $\mathbf{t-statistic}$ and $\mathbf{F}$ is the $\mathbf{F-statistic}$

Full model:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$

Where:

$$\varepsilon \sim N(0, \sigma)$$

$$t = \frac{\widehat{\beta_1}}{\sqrt{(X^TX)^T\widehat{\sigma}^2}} \sim t_{n-p-1}$$

$$z = \frac{\widehat{\beta_1}}{\sqrt{(X^TX)^T\sigma^2}} \sim N(0,1)$$

$$\frac{\widehat{\beta_1}^2}{(X^TX)^T\sigma^2} \sim \chi^2_{(1)}$$

Restricted model:

$$F_{H_0} = \frac{(SS^F_{Reg} - SS^R_{Reg})(DOF^F_{Reg} - DOF^R_{Reg})}{SS^F_{Res} - DOF^F_{Res}} \sim F(m, n-p-1)$$

$$= \frac{(SS^F_{Reg} - SS^R_{Reg})/1}{SS^F_{Res}/(n-p-1)} \sim F(1, n-p-1)$$

Since $SSE = \hat{\sigma}^2$

$$\frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{(n-p-1)}$$

$$\mathrm{t}^2 = \frac{\chi^2_{(1)}/1}{\chi^2_{(n-p-1)}/(n-p-1)} \sim \mathrm{F}(1, \mathrm{n}-\mathrm{p}-1)$$

Thus $\mathrm{t}^2 = \mathrm{F}$

## 3. Chapter 6 Problem a, b, c

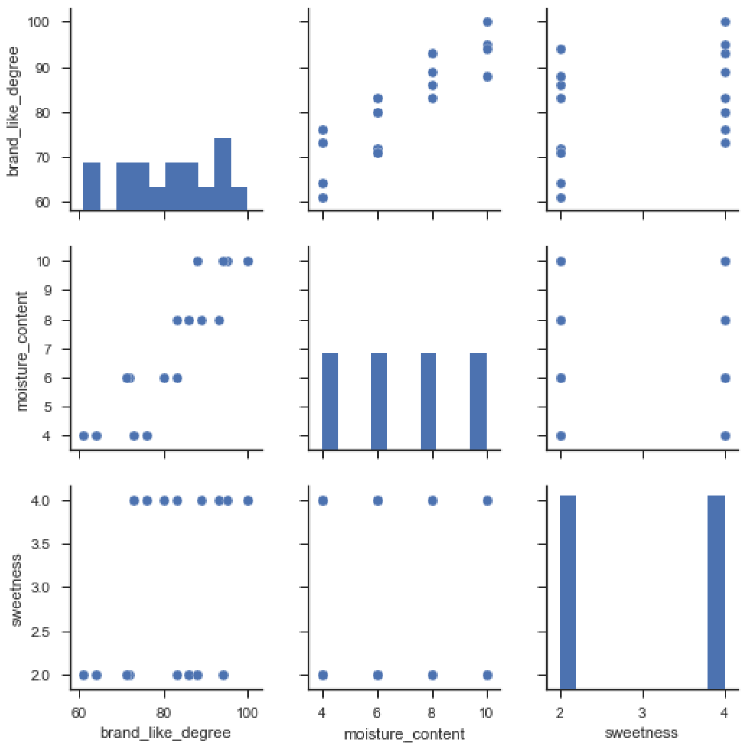
(a) Basic data cleaning with Python

```
import pandas as pd
import numpy as np
import seaborn as sns
sns.set(style="ticks", color_codes=True)
```

```
with open('CH06PR05.txt') as f:
    data = f.readlines()
df = pd.DataFrame(list(map(lambda x: x.split(), data)))
df.columns = ['brand_like_degree', 'moisture_content', 'sweetness']
df = df.astype(float)
df.head()
```

| | brand_like_degree | moisture_content | sweetness |
|---|---|---|---|
| **0** | 64.0 | 4.0 | 2.0 |
| **1** | 73.0 | 4.0 | 4.0 |
| **2** | 61.0 | 4.0 | 2.0 |
| **3** | 76.0 | 4.0 | 4.0 |
| **4** | 72.0 | 6.0 | 2.0 |

Scatter plot matrix



Correlation Matrix

| | brand_like_degree | moisture_content | sweetness |
|---|---|---|---|
| **brand_like_degree** | 1.000000 | 0.892393 | 0.394581 |
| **moisture_content** | 0.892393 | 1.000000 | 0.000000 |
| **sweetness** | 0.394581 | 0.000000 | 1.000000 |

Based on the results, we could see that the correlation between moisture content and degree of brand liking is very high. There is no correlation between moisture content and sweetness.

(b) Based on 6.1

$$Y = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$$

```
from statsmodels.formula.api import ols
model = ols('brand_like_degree~moisture_content + sweetness', df).fit()
print(model.summary())
print(model._results.params)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      brand_like_degree   R-squared:                       0.952
Model:                            OLS   Adj. R-squared:                  0.945
Method:                 Least Squares   F-statistic:                     129.1
Date:                Mon, 29 Oct 2018   Prob (F-statistic):           2.66e-09
Time:                        21:54:25   Log-Likelihood:                -36.894
No. Observations:                  16   AIC:                             79.79
Df Residuals:                      13   BIC:                             82.11
Df Model:                           2
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
Intercept           37.6500      2.996     12.566      0.000      31.177      44.123
moisture_content     4.4250      0.301     14.695      0.000       3.774       5.076
sweetness            4.3750      0.673      6.498      0.000       2.920       5.830
==============================================================================
Omnibus:                        0.766   Durbin-Watson:                   2.313
Prob(Omnibus):                  0.682   Jarque-Bera (JB):                0.647
Skew:                           0.049   Prob(JB):                        0.724
Kurtosis:                       2.020   Cond. No.                         35.9
==============================================================================
```

Based on the result, the regression function is:

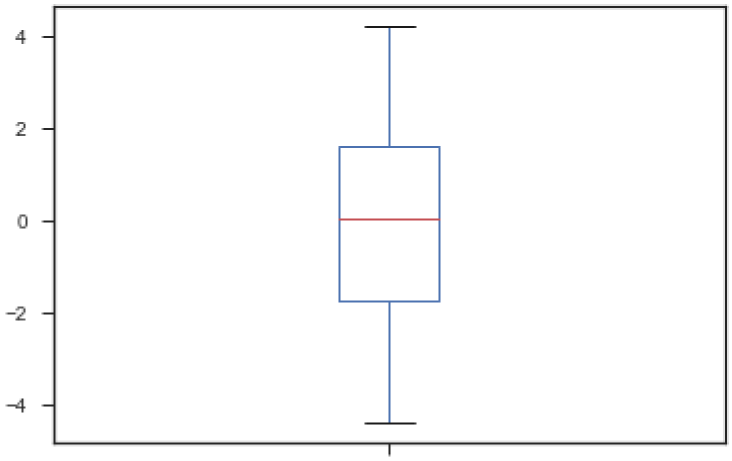
$$Y = 37.65 + 4.425 Moisture\ content + 4.375 Sweetness + \varepsilon$$

$b_1$ is the coefficients of regression model, which means when $Moisture\ content$ improve 1, degree of brand liking improve 4.425, when sweetness improve 1, degree of brand liking improve 4.375.

(c) Residual

```
model.resid.plot(kind = 'box')
```

<matplotlib.axes._subplots.AxesSubplot at 0xc5601d0>




We could see the residual almost follow the normal distribution and also don't have large variance, which means the assumptions of model are correct.

## 4. Chapter 6 Problem 7

(1)

```
print('R square:', round(model.rsquared, 3))
```

R square: 0.952

R square is 0.952, which means the factor sweetness and moisture content have strong relationship with degree of brand liking.

(2)

```
SStotal = sum((df['brand_like_degree'] - df['brand_like_degree'].mean())**2)
SSreg = sum((model.fittedvalues - df['brand_like_degree'].mean())**2)
SSres = SStotal - SSreg
R2 = SSreg/SStotal
round(R2, 6)
```

0.952059

Based on the result, the multiple and single determination $R^2$ are the same.

## 5. Chapter 6 Problem 8

(a)

The confidence interval for $E(Y_h)$ is

$$\widehat{Y_h} \pm t\left(1-\frac{a}{2}; n-2\right) s\{\widehat{Y_h}\}$$

$$s^2\{\widehat{Y_h}\} = X_h{}' s^2\{b\} X_h$$

$\mathrm{X_h} = \begin{pmatrix} 1 \\ 5 \\ 4 \end{pmatrix}$

$\mathrm{b} = \begin{pmatrix} 37.65 \\ 4.425 \\ 4.375 \end{pmatrix}$

$\widehat{\mathrm{Y_h}} = \mathrm{X_h}'\mathrm{b} = 77.275$

$\mathrm{MSE} = 7.2538$ calculated with Python

$\mathrm{s^2\{b\} = MSE(X'X)^{-1}}$

$\mathrm{s^2\{\widehat{Y_h}\}} = 1.269$

```
from scipy.stats import t
MSE = SSres/(len(df)-3)
xh = np.array([1, 5, 4])
b = np.array([37.65, 4.425, 4.375])
yh = np.dot(xh.T, b)
t = t.ppf(0.995, len(df)-3)
X = df[['moisture_content', 'sweetness']].values
X = np.hstack([np.ones(len(df)).reshape(-1, 1), X])
ciup = yh + t*np.sqrt(MSE)*np.sqrt(np.dot(np.dot(np.sqrt(xh.T), np.linalg.inv(np.dot(X.T, X))), xh))
cilow = yh - t*np.sqrt(MSE)*np.sqrt(np.dot(np.dot(np.sqrt(xh.T), np.linalg.inv(np.dot(X.T, X))), xh))
round(cilow,3), round(ciup,3)
```

```
(74.475, 80.075)
```

The confidence interval is:

$[74.475, 80.075]$

(b)

```
ciup = yh + t*np.sqrt(MSE)*np.sqrt(1 + np.dot(np.dot(np.sqrt(xh.T), np.linalg.inv(np.dot(X.T, X))), xh))
cilow = yh - t*np.sqrt(MSE)*np.sqrt(1 + np.dot(np.dot(np.sqrt(xh.T), np.linalg.inv(np.dot(X.T, X))), xh))
round(cilow,3), round(ciup,3)
```

```
(68.693, 85.857)
```

Confidence interval is [68.693, 85.857]

## 6. Chapter 6 Problem 25

Set $Y_i = Y_i - 4X_{i2}$, Use the model

$$Y_i' = \beta_0 + \beta_1 X_{i1} + \beta_3 X_{i3} + \varepsilon_i$$

to get the fitted line. Thus, we have $lm(Y - 4X_2 \sim X_1 + X_3)$