Assignment 01

Matrix Algebra for Linear Regression

This assignment is worth 20 points.

Unstandardized Regression

1. Write out the elements of the matrix $\mathbf{X}^\intercal\mathbf{X}$, where $\mathbf{X}$ is the design matrix.

```
t(X) %*% X
```

```
##      [,1]    [,2] [,3]   [,4]
## [1,]   10    3020    5    828
## [2,] 3020 1331924  828 213806
## [3,]    5     828    5    828
## [4,]  828  213806  828 213806
```

2. Does $\mathbf{X}^\intercal\mathbf{X}$ have an inverse? Explain.

```
det(t(X) %*% X)
```

```
## [1] 301283491796
```

Yes. Since $\det(\mathbf{X}^\intercal\mathbf{X}) \neq 0$, then $\mathbf{X}^\intercal\mathbf{X}$ has an inverse.

3. Compute (using matrix algebra) and report the vector of coefficients, b for the OLS regression.

```
b = solve(t(X) %*% X) %*% t(X) %*% y
b
```

```
##             [,1]
## [1,] 68.16113912
## [2,] -0.05511209
## [3,] 75.28536912
## [4,] -0.11968567
```

4. Compute (using matrix algebra) and report the variance–covariance matrix of the coefficients.

```
y_hat = X %*% b
e = y - y_hat
mse = (t(e) %*% e) / (10 - 3 - 1)
var_cov_b = as.numeric(mse) * solve(t(X) %*% X)
var_cov_b
```

```
##              [,1]         [,2]         [,3]         [,4]
## [1,]  1161.733848 -2.277506127 -1161.733848  2.277506127
## [2,]    -2.277506  0.005195041     2.277506 -0.005195041
## [3,] -1161.733848  2.277506127  1616.937571 -4.040359754
## [4,]     2.277506 -0.005195041    -4.040360  0.015840293
```

5. Use the values from b (Question 3) and from the variance–covariance matrix you reported in the previous question to find the 95% CI for the coefficient associated with the main-effect of PCI. (Hint: If you need to refresh yourself on how CIs are computed, see here.)

```
moe = qt(.975, df = 6) * sqrt(var_cov_b[2, 2])
b[2] + c(-moe, moe)
```

```
## [1] -0.2314773  0.1212531
```

6. Compute (using matrix algebra) and report the hat-matrix, H. Also show how you would use the values in the hat-matrix to find $\hat{y}_1$ (the predicted value for Algeria).

```
# Compute hat-matrix
H = X %*% solve(t(X) %*% X) %*% t(X)
H
```

```
##                [,1]          [,2]          [,3]          [,4]          [,5]
##  [1,]  9.164420e-01  6.765422e-16 -9.831371e-02  2.298509e-16  2.103921e-01
##  [2,]  7.771561e-16  5.616691e-01 -3.330669e-16  2.491530e-01 -5.551115e-17
##  [3,] -9.831371e-02 -5.607494e-16  3.242125e-01 -8.055622e-17  1.956729e-01
##  [4,]  2.775558e-16  2.491530e-01 -1.509209e-16  2.066802e-01  0.000000e+00
##  [5,]  2.103921e-01 -2.151057e-16  1.956729e-01 -3.642919e-17  2.001507e-01
##  [6,]  9.118885e-02 -3.504141e-16  2.453070e-01 -8.239937e-17  1.984217e-01
##  [7,]  0.000000e+00  9.592912e-02  0.000000e+00  1.858561e-01  0.000000e+00
##  [8,]  2.775558e-16  3.386600e-01 -1.387779e-16  2.188447e-01  0.000000e+00
##  [9,] -1.197092e-01 -6.223320e-16  3.331212e-01 -1.198043e-16  1.953626e-01
## [10,] -6.661338e-16 -2.454113e-01 -1.387779e-16  1.394659e-01 -2.775558e-16
##                [,6]         [,7]          [,8]          [,9]         [,10]
##  [1,]  9.118885e-02 1.084202e-17  3.577867e-16 -1.197092e-01 -4.770490e-16
##  [2,] -1.665335e-16 9.592912e-02  3.386600e-01 -3.747003e-16 -2.454113e-01
##  [3,]  2.453070e-01 1.548783e-16 -2.180873e-16  3.331212e-01  6.793611e-16
##  [4,] -3.469447e-17 1.858561e-01  2.188447e-01 -1.717376e-16  1.394659e-01
##  [5,]  1.984217e-01 5.117434e-17 -8.760354e-17  1.953626e-01  2.463307e-16
##  [6,]  2.165259e-01 4.900594e-17 -1.591609e-16  2.485565e-01  3.417405e-16
##  [7,]  0.000000e+00 2.299466e-01  1.601003e-01  0.000000e+00  3.281678e-01
```

```
##  [8,] -8.326673e-17 1.601003e-01  2.531608e-01 -1.526557e-16  2.923411e-02
##  [9,]  2.485565e-01 1.265806e-16 -2.637322e-16  3.426689e-01  6.754580e-16
## [10,] -2.220446e-16 3.281678e-01  2.923411e-02 -8.326673e-17  7.485434e-01
```

```
# Compute predicted value for Algeria
yhat = H %*% y
yhat[1]
```

```
## [1] 73.52741
```

7. Compute (using matrix algebra) and report the vector of residuals, e.

```
I = diag(10)
e = (I - H) %*% y
e
```

```
##              [,1]
##  [1,]  12.7725938
##  [2,]   3.2612784
##  [3,]  18.4397391
##  [4,]   3.0143684
##  [5,] -58.9056876
##  [6,]  27.5771999
##  [7,]   5.7806892
##  [8,] -10.4372448
##  [9,]   0.1161548
## [10,]  -1.6190911
```

8. Compute (using matrix algebra) and report the estimated value for the RMSE.

```
rmse = sqrt((t(e) %*% e) / (10 - 3 - 1))
rmse
```

```
##          [,1]
## [1,] 28.57229
```

9. Given the assumptions of the OLS model and the RMSE estimate you computed in the previous question, compute and report the variance–covariance matrix of the residuals.

```
as.numeric(rmse)^2 * diag(10)
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
##  [1,] 816.3758   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
##  [2,]   0.0000 816.3758   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
##  [3,]   0.0000   0.0000 816.3758   0.0000   0.0000   0.0000   0.0000   0.0000
##  [4,]   0.0000   0.0000   0.0000 816.3758   0.0000   0.0000   0.0000   0.0000
##  [5,]   0.0000   0.0000   0.0000   0.0000 816.3758   0.0000   0.0000   0.0000
##  [6,]   0.0000   0.0000   0.0000   0.0000   0.0000 816.3758   0.0000   0.0000
##  [7,]   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000 816.3758   0.0000
##  [8,]   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000 816.3758
##  [9,]   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
## [10,]   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
##           [,9]    [,10]
##  [1,]   0.0000   0.0000
##  [2,]   0.0000   0.0000
##  [3,]   0.0000   0.0000
##  [4,]   0.0000   0.0000
##  [5,]   0.0000   0.0000
##  [6,]   0.0000   0.0000
##  [7,]   0.0000   0.0000
##  [8,]   0.0000   0.0000
##  [9,] 816.3758   0.0000
## [10,]   0.0000 816.3758
```

ANOVA Decomposition

10. Compute (using matrix algebra) and report the model, residual, and total sum of squares terms in the ANOVA decomposition table. (2pts)

```
# Total
ss_total = t(y - mean(y)) %*% (y - mean(y))
ss_total
```

```
##          [,1]
## [1,] 20144.37
```

```
# Model
ss_model = t(yhat - mean(y)) %*% (yhat - mean(y))
ss_model
```

```
##          [,1]
## [1,] 15246.11
```

```
# Residual
ss_residual = t(e) %*% e
ss_residual
```

```
##          [,1]
## [1,] 4898.255
```

11. Compute (using matrix algebra) and report the model, residual, and total degrees of freedom terms in the ANOVA decomposition table. (2pts)

```r
# Total
df_total = length(y) - 1
df_total
```

```
## [1] 9
```

```r
# Model
df_model = sum(diag(H)) - 1
df_model
```

```
## [1] 3
```

```r
# Residual
df_residual = df_total - df_model
df_residual
```

```
## [1] 6
```

12. Use the values you obtained in Questions 11 and 12 to compute the model and residual mean square terms.

```r
# Model
ms_model = ss_model / df_model
ms_model
```

```
##          [,1]
## [1,] 5082.037
```

```r
# Residual
ms_residual = ss_residual / df_residual
ms_residual
```

```
##          [,1]
## [1,] 816.3758
```

13. Use the mean square terms you found in Question 13 to compute the F-value for the model (i.e., to test $H_0 : \rho^2 = 0$). Also compute the p-value associated with this F-value. (Hint: If you need to refresh yourself on how F-values or p-values are computed, see here.)

```
# Compute F-statistic
f_obs = ms_model / ms_residual
f_obs
```

```
##          [,1]
## [1,] 6.225119
```

```
# Compute p-value
p = 1 - pf(f_obs, df1 = df_model, df2 = df_residual)
p
```

```
##           [,1]
## [1,] 0.0284259
```

Regression: Effects-Coding

14. Write out the design matrix that would be used to fit the model.

```
##       [,1] [,2]
##  [1,]    1    1
##  [2,]    1   -1
##  [3,]    1    1
##  [4,]    1   -1
##  [5,]    1    1
##  [6,]    1    1
##  [7,]    1   -1
##  [8,]    1   -1
##  [9,]    1    1
## [10,]    1   -1
```

15. Compute (using matrix algebra) and report the vector of coefficients, b, from the OLS regression.

```
b = solve(t(X) %*% X) %*% t(X) %*% y
b
```

```
##       [,1]
## [1,] 79.25
## [2,] 35.25
```

16. Compute (using matrix algebra) and report the variance–covariance matrix for the coefficients.

```
y_hat = X %*% b
e = y - y_hat
mse = (t(e) %*% e) / (10 - 1 - 1)

# Compute variance-covariance matrix
as.numeric(mse) * solve(t(X) %*% X)
```

```
##          [,1]     [,2]
## [1,] 96.48425  0.00000
## [2,]  0.00000 96.48425
```

17. Explain why the sampling variances for the coefficients are the same and why the sampling covariance is zero by referring to computations produced in the matrix algebra. (2pts)

The reason is the computation of the inverse of $\mathbf{X}^\intercal\mathbf{X}$. This produces a 2×2 diagonal matrix that has 0.1 on the main diagonal and zeros on the off-diagonal. Since the diagonal elements are the same, the sampling variances will be equal; since the off-diagonal elements are zero, the covariance will also be zero.

```
solve(t(X) %*% X)
```

```
##      [,1] [,2]
## [1,]  0.1  0.0
## [2,]  0.0  0.1
```