Regression using Matrices

We will use the following toy data set to illustrate how regression is carried out via matrix algebra.

```
> myData = data.frame(
    wage   = c(12, 8, 16.26, 13.65, 8.5),
    age    = c(32, 33, 32, 33, 26),
    sex    = c("M", "F", "M", "M", "M")
    )

> myData

   wage age sex
1 12.00  32   M
2  8.00  33   F
3 16.26  32   M
4 13.65  33   M
5  8.50  26   M
```

We can now present linear regression in matrix terms. Begin with the linear regression model

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \quad \text{where } i = 1, \ldots, n$$

This implies

$$\begin{aligned}
Y_1 &= 1\beta_0 + \beta_1 X_1 + \epsilon_1 \\
Y_2 &= 1\beta_0 + \beta_1 X_2 + \epsilon_2 \\
&\vdots \\
Y_n &= 1\beta_0 + \beta_1 X_n + \epsilon_n
\end{aligned}$$

We can now arrange these into appropriate vectors and matrices.

$$\underset{n\times 1}{\mathbf{Y}} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \qquad \underset{n\times 2}{\mathbf{X}} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \qquad \underset{2\times 1}{\boldsymbol{\beta}} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \qquad \underset{n\times 1}{\boldsymbol{\epsilon}} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

We can write the model using matrix terms very compactly as

$$\underset{n\times 1}{\mathbf{Y}} = \underset{n\times 2}{\mathbf{X}}\,\underset{2\times 1}{\boldsymbol{\beta}} + \underset{n\times 1}{\boldsymbol{\epsilon}}$$

Dimension
of product

Next we turn to the assumptions of the errors. Since $\mathbf{E}\{\varepsilon_i\} = 0$,

$$\underset{n\times 1}{\mathbf{E}\{\boldsymbol{\epsilon}\}} = \underset{n\times 1}{\mathbf{0}}$$

We also make the assumption that the errors have constant variance σ^2 and that they are independent of one another $\sigma(\varepsilon_i, \varepsilon_j)=0$ for $i \neq j$. The **variance-covariance matrix of the error terms** can therefore be expressed as

$$\underset{n\times n}{\boldsymbol{\sigma}^2\{\boldsymbol{\epsilon}\}} = \begin{bmatrix} \sigma_\epsilon^2 & 0 & \cdots & 0 \\ 0 & \sigma_\epsilon^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_\epsilon^2 \end{bmatrix}$$

This scalar matrix can also be expressed as

$$\underset{n\times n}{\boldsymbol{\sigma}^2\{\boldsymbol{\epsilon}\}} = \underset{n\times n}{\sigma^2\mathbf{I}}$$

Thus, the **regression model** can be completely expressed in matrix terms as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\varepsilon}$ is a vector of independent random variables with $\mathbf{E}\{\boldsymbol{\varepsilon}\}=0$ and $\boldsymbol{\sigma}^2\{\boldsymbol{\varepsilon}\}= \sigma^2\mathbf{I}$

Solving for β

We can re-express the **error vector** as

$$\boldsymbol{\epsilon} = \mathbf{Y} - \mathbf{X}\boldsymbol{\beta}$$

The *method of least squares* minimizes the **sum of squared errors** which is expressed as

$$\boldsymbol{\epsilon}'\boldsymbol{\epsilon} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

$$= \underset{1\times n}{\mathbf{Y}'}\ \underset{n\times 1}{\mathbf{Y}} - \underset{1\times 2}{\boldsymbol{\beta}'}\ \underset{2\times n}{\mathbf{X}'}\ \underset{n\times 1}{\mathbf{Y}} - \underset{1\times n}{\mathbf{Y}'}\ \underset{n\times 2}{\mathbf{X}}\ \underset{2\times 1}{\boldsymbol{\beta}} + \underset{1\times 2}{\boldsymbol{\beta}'}\ \underset{2\times n}{\mathbf{X}'}\ \underset{n\times 2}{\mathbf{X}}\ \underset{2\times 1}{\boldsymbol{\beta}}$$

Note that each term is a 1x1 matrix, which means that each term is equal to its transpose. We will re-write the third term **Y'X β** as its transpose **β'X'Y**

$$= \underset{1\times n}{\mathbf{Y}'}\ \underset{n\times 1}{\mathbf{Y}} - \underset{1\times 2}{\boldsymbol{\beta}'}\ \underset{2\times n}{\mathbf{X}'}\ \underset{n\times 1}{\mathbf{Y}} - \underset{1\times 2}{\boldsymbol{\beta}'}\ \underset{2\times n}{\mathbf{X}'}\ \underset{n\times 1}{\mathbf{Y}} + \underset{1\times 2}{\boldsymbol{\beta}'}\ \underset{2\times n}{\mathbf{X}'}\ \underset{n\times 2}{\mathbf{X}}\ \underset{2\times 1}{\boldsymbol{\beta}}$$

Combining the two middle terms we get

$$= \underset{1\times n}{\mathbf{Y}'}\ \underset{n\times 1}{\mathbf{Y}} - 2\underset{1\times 2}{\boldsymbol{\beta}'}\ \underset{2\times n}{\mathbf{X}'}\ \underset{n\times 1}{\mathbf{Y}} + \underset{1\times 2}{\boldsymbol{\beta}'}\ \underset{2\times n}{\mathbf{X}'}\ \underset{n\times 2}{\mathbf{X}}\ \underset{2\times 1}{\boldsymbol{\beta}}$$

To find the values for the elements in $\boldsymbol{\beta}$ that minimize the equation, we differentiate with respect to β_0 and β_1 ($\boldsymbol{\beta}$)

$$\frac{\partial}{\partial \boldsymbol{\beta}}(\underset{1\times n}{\mathbf{Y}'}\underset{n\times 1}{\mathbf{Y}} - 2\underset{1\times 2}{\boldsymbol{\beta}'}\underset{2\times n}{\mathbf{X}'}\underset{n\times 1}{\mathbf{Y}} + \underset{1\times 2}{\boldsymbol{\beta}'}\underset{2\times n}{\mathbf{X}'}\underset{n\times 2}{\mathbf{X}}\underset{2\times 1}{\boldsymbol{\beta}})$$

Differentiating, we get...

$$= -2\underset{2\times n}{\mathbf{X}'}\underset{n\times 1}{\mathbf{Y}} + 2\underset{2\times n}{\mathbf{X}'}\underset{n\times 2}{\mathbf{X}}\underset{2\times 1}{\boldsymbol{\beta}}$$

Solving this equation for zero...

$$\begin{aligned}\underset{2\times 1}{\mathbf{0}} &= -2\underset{2\times n}{\mathbf{X}'}\underset{n\times 1}{\mathbf{Y}} + 2\underset{2\times n}{\mathbf{X}'}\underset{n\times 2}{\mathbf{X}}\underset{2\times 1}{\boldsymbol{\beta}} \\ &= -\underset{2\times n}{\mathbf{X}'}\underset{n\times 1}{\mathbf{Y}} + \underset{2\times n}{\mathbf{X}'}\underset{n\times 2}{\mathbf{X}}\underset{2\times 1}{\boldsymbol{\beta}}\end{aligned}$$

Adding **X'Y** to both sides

$$\underset{2\times n}{\mathbf{X}'}\underset{n\times 1}{\mathbf{Y}} = \underset{2\times n}{\mathbf{X}'}\underset{n\times 2}{\mathbf{X}}\underset{2\times 1}{\boldsymbol{\beta}}$$

Now we pre-multiply both sides of the equation by $(\mathbf{X'X})^{-1}$

$$\underset{2\times 2}{(\mathbf{X}'\mathbf{X})^{-1}}\underset{2\times n}{\mathbf{X}'}\underset{n\times 1}{\mathbf{Y}} = \underset{2\times 2}{(\mathbf{X}'\mathbf{X})^{-1}}\underset{2\times n}{\mathbf{X}'}\underset{n\times 2}{\mathbf{X}}\underset{2\times 1}{\boldsymbol{\beta}}$$

$$\underset{2\times 2}{(\mathbf{X}'\mathbf{X})^{-1}}\underset{2\times n}{\mathbf{X}'}\underset{n\times 1}{\mathbf{Y}} = \underset{2\times 2}{\mathbf{I}}\underset{2\times 1}{\boldsymbol{\beta}}$$

This means

$$\underset{2\times 1}{\boldsymbol{\beta}} = \underset{2\times 2}{(\mathbf{X}'\mathbf{X})^{-1}}\underset{2\times n}{\mathbf{X}'}\underset{n\times 1}{\mathbf{Y}}$$

The vector of regression coefficients can be obtained directly through manipulation of the design matrix and the vector of outcomes.

$$\underset{2\times 1}{\boldsymbol{\beta}} = \underset{2\times 2}{(\mathbf{X}'\mathbf{X})^{-1}} \underset{2\times n}{\mathbf{X}'} \underset{n\times 1}{\mathbf{Y}}$$

```
> Y = myData$wage
> X = matrix(c(rep(1, 5), myData$age), ncol = 2)

> b = solve(t(X) %*% X) %*% t(X) %*% Y
> b

                  [,1]
(Intercept) -3.8982759
age          0.4993678
```

When we estimate the $\boldsymbol{\beta}$ matrix from sample data we refer to it as **b**.

$$\mathbf{b} = \begin{bmatrix} -3.8983 \\ 0.4994 \end{bmatrix}$$

Fitted Values

Let the vector of **fitted values** be

$$\underset{n\times 1}{\hat{\mathbf{Y}}} = \begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \vdots \\ \hat{Y}_n \end{bmatrix}$$

In matrix notation

$$\underset{n\times 1}{\hat{\mathbf{Y}}} = \underset{n\times 2}{\mathbf{X}} \; \underset{2\times 1}{\mathbf{b}}$$

$$\underset{n\times 1}{\hat{\mathbf{Y}}} = \underset{n\times 2}{\mathbf{X}} \underset{2\times 1}{\mathbf{b}}$$

```
> fitted_values = X %*% b
> fitted_values

      [,1]
1 12.081494
2 12.580862
3 12.081494
4 12.580862
5  9.085287
```

H Matrix

$$\underset{n\times 1}{\hat{\mathbf{Y}}} = \underset{n\times 2}{\mathbf{X}}\ \underset{2\times 1}{\mathbf{b}}$$

Substituting the expression for **b**

$$\underset{n\times 1}{\hat{\mathbf{Y}}} = \underset{n\times 2}{\mathbf{X}}\ \underset{2\times 2}{(\mathbf{X}'\mathbf{X})^{-1}}\ \underset{2\times n}{\mathbf{X}'}\ \underset{n\times 1}{\mathbf{Y}}$$

This is often expressed as

$$\underset{n\times 1}{\hat{\mathbf{Y}}} = \underset{n\times n}{\mathbf{H}}\ \underset{n\times 1}{\mathbf{Y}}$$

where

$$\underset{n\times n}{\mathbf{H}} = \underset{n\times 2}{\mathbf{X}}\ \underset{2\times 2}{(\mathbf{X}'\mathbf{X})^{-1}}\ \underset{2\times n}{\mathbf{X}'}$$

H is a square *n* x *n* matrix that can be created completely from the design matrix and its transpose.

```
> h = X %*% solve(t(X) %*% X) %*% t(X)

> h

           1           2          3           4           5
1 0.21839080  0.24137931 0.21839080  0.24137931  0.08045977
2 0.24137931  0.29310345 0.24137931  0.29310345 -0.06896552
3 0.21839080  0.24137931 0.21839080  0.24137931  0.08045977
4 0.24137931  0.29310345 0.24137931  0.29310345 -0.06896552
5 0.08045977 -0.06896552 0.08045977 -0.06896552  0.97701149
```

The fitted values can be expressed as linear combinations of the response vector **Y** using coefficients found in **H**.

```
> h %*% Y

       [,1]
1 12.081494
2 12.580862
3 12.081494
4 12.580862
5  9.085287
```

Because of this, **H** is often referred to as the *hat matrix*.

H is a symmetric matrix and also has a special property called **idempotency**:

$$\underset{n\times n}{\mathbf{H}}\;\underset{n\times n}{\mathbf{H}} = \underset{n\times n}{\mathbf{H}}$$

```
> h %*% h

           1           2          3           4           5
1 0.21839080  0.24137931 0.21839080  0.24137931  0.08045977
2 0.24137931  0.29310345 0.24137931  0.29310345 -0.06896552
3 0.21839080  0.24137931 0.21839080  0.24137931  0.08045977
4 0.24137931  0.29310345 0.24137931  0.29310345 -0.06896552
5 0.08045977 -0.06896552 0.08045977 -0.06896552  0.97701149
```

Residuals

Let the vector of residuals be

$$\underset{n\times 1}{\mathbf{e}} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

In matrix notation

$$\begin{aligned} \underset{n\times 1}{\mathbf{e}} &= \underset{n\times 1}{\mathbf{Y}} - \underset{n\times 2}{\mathbf{X}}\ \underset{2\times 1}{\mathbf{b}} \\ &= \underset{n\times 1}{\mathbf{Y}} - \underset{n\times 1}{\hat{\mathbf{Y}}} \end{aligned}$$

Since the fitted values can be expressed as **HY**, the residual vector can also be expressed as

$$\underset{n\times 1}{\mathbf{e}} = \underset{n\times 1}{\mathbf{Y}} - \underset{n\times n}{\mathbf{H}}\ \underset{n\times 1}{\mathbf{Y}}$$

The residuals can also be expressed as linear combinations of the response vector **Y**.

Using the distributive property as it relates to matrices, we re-write this as

$$\underset{n\times 1}{\mathbf{e}} = (\underset{n\times n}{\mathbf{I}} - \underset{n\times n}{\mathbf{H}})\ \underset{n\times 1}{\mathbf{Y}}$$

The matrix (**1** - **H**), like **H**, is *symmetric* and *idempotent*.

$$\underset{n\times 1}{\mathbf{e}} = (\underset{n\times n}{\mathbf{I}} - \underset{n\times n}{\mathbf{H}})\underset{n\times 1}{\mathbf{Y}}$$

```
> i = diag(5)

> residuals = (i - h) %*% Y
> residuals

         [,1]
1 -0.08149425
2 -4.58086207
3  4.17850575
4  1.06913793
5 -0.58528736
```

Mean Squared Error (MSE)

Recall a Mean Square (or variance estimate) is the ratio of the sum of squares to its degrees of freedom.

$$\text{MSE} = \frac{\text{SS}_{\text{error}}}{df_{\text{error}}} = \frac{\sum (e_i)^2}{n-p}$$

In matrix notation

$$\text{SS}_{\text{Error}} = \mathbf{e}'\mathbf{e}$$

```
# Compute SS for the residuals
> SSE = t(residuals) %*% residuals
> SSE

         [,1]
[1,] 39.93647


# Compute df for the residuals
> num_coef = 2
> df_residual = nrow(myData) - num_coef

> MSE = SSE / df_residual
> MSE

         [,1]
[1,] 13.31216
```

Standard Errors for the Coefficients

The SEs for the coefficients represent the uncertainty in the estimates.

The uncertainty is also correlated between the coefficients (unless you have a balanced design.)

We represent the uncertainty and the correlation in the estimated variance–covariance matrix of the coefficients,

$$\mathbf{V}_\beta \hat{\sigma}^2$$

where

$$\mathbf{V}_\beta = (\mathbf{X}^t \mathbf{X})^{-1}$$

and

$$\hat{\sigma}^2 = \mathrm{MSE}$$

```
> V_b = solve(t(X) %*% X)
> V_b

             [,1]        [,2]
[1,] 28.1724138 -0.89655172
[2,] -0.8965517  0.02873563
```

$$\mathbf{V}_\beta = \begin{bmatrix} 28.17 & -0.90 \\ -0.90 & 0.03 \end{bmatrix}$$

$$\hat{\sigma}^2 = 13.31$$

$$\begin{bmatrix} 28.17 & -0.90 \\ -0.90 & 0.03 \end{bmatrix} (13.31) = \boxed{\begin{bmatrix} 375.0 & -11.9 \\ -11.9 & 0.04 \end{bmatrix}}$$

```
> var_cov = as.numeric(MSE) * V_b
> var_cov

             [,1]        [,2]
[1,] 375.03555 -11.9350358
[2,] -11.93504   0.3825332
```

The diagonal elements are the *variances* of the regression coefficients and the off-diagonal elements are the *covariances*.

We can also compute the estimated SE for each coefficient and the correlation between the coefficients.

$$\mathbf{V}_\beta = \begin{bmatrix} 28.17 & -0.90 \\ -0.90 & 0.03 \end{bmatrix}$$

$$\text{SE}_{\beta_0} = (\sqrt{\mathbf{V}_{11}})(\hat{\sigma}) \qquad \text{SE}_{\beta_1} = (\sqrt{\mathbf{V}_{22}})(\hat{\sigma}) \qquad r_{\beta_1,\beta_2} = \frac{\mathbf{V}_{12}}{\sqrt{\mathbf{V}_{11}\mathbf{V}_{22}}}$$

$$\text{SE}_{\beta_0} = (\sqrt{28.17})(3.65) = 19.37$$

$$\text{SE}_{\hat{\beta}_1} = (\sqrt{0.03})(3.65) = 0.62$$

$$r_{\hat{\beta}_1,\hat{\beta}_2} = \frac{-0.90}{\sqrt{(28.17)(0.03)}} = -0.996$$

```
# Compute SEs
> sqrt(diag(var_cov))

[1] 19.3658346  0.6184927

# Compute correlation between coefficients
> V_b[1, 2] / sqrt(prod(diag(V_b)))

[1] -0.9964441
```

There are several built-in function in R that compute these values for us in practice.

```
# Fit the linear model
> lm.1 = lm(wage ~ 1 + age, data = myData)


# Regression summary
> summary(lm.1)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.8983    19.3658  -0.201    0.853
age           0.4994     0.6185   0.807    0.479


# Analysis of variance decomposition
> anova(lm.1)

Analysis of Variance Table

Response: wage
          Df Sum Sq Mean Sq F value Pr(>F)
age        1  8.678   8.678  0.6519 0.4785
Residuals  3 39.936  13.312
```

```r
# Obtain the regression coefficients
> coef(lm.1)

(Intercept)         age
 -3.8982759   0.4993678


# Get the fitted values
> fitted(lm.1)

        1         2         3         4         5
12.081494 12.580862 12.081494 12.580862  9.085287


# Get the residuals
> resid(lm.1)

          1           2           3           4           5
-0.08149425 -4.58086207  4.17850575  1.06913793 -0.58528736
```

```
# Obtain the model (X) matrix
> model.matrix(lm.1)

  (Intercept) age
1           1  32
2           1  33
3           1  32
4           1  33
5           1  26




# Obtain the variance-covariance matrix of the coefficients
> vcov(lm.1)

            (Intercept)         age
(Intercept)   375.03555 -11.9350358
age           -11.93504   0.3825332
```