

Regression Using Matrices

EPsy 8264

2018-05-14

Recall the model equation we use in simple linear regression

$$Y_i = \beta_0 + \beta_1(X_i) + \epsilon_i$$

where the response variable (Y) is represented as a linear function of the predictor (X) and a residual (ϵ). Equation terms with an i subscript vary across subjects. Terms without an i subscript are the same (fixed) across subjects.

Model Equation Using Matrix Algebra

Using the subject-specific subscripts ($1, 2, 3, \dots, n$), we can write out each subject's equation:

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1(X_1) + \epsilon_1 \\ Y_2 &= \beta_0 + \beta_1(X_2) + \epsilon_2 \\ Y_3 &= \beta_0 + \beta_1(X_3) + \epsilon_3 \\ &\vdots \\ Y_n &= \beta_0 + \beta_1(X_n) + \epsilon_n \end{aligned}$$

These can be arranged into a set of vectors and matrices,

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} \beta_0(1) + \beta_1(X_1) \\ \beta_0(1) + \beta_1(X_2) \\ \beta_0(1) + \beta_1(X_3) \\ \vdots \\ \beta_0(1) + \beta_1(X_n) \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

We can re-write this as,

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ 1 & X_3 \\ \vdots & \\ 1 & X_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Naming these vectors and matrices, we can write the regression model compactly as,¹

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

In the equation above, the \mathbf{X} matrix is sometimes referred to the *design matrix*, the *model matrix*, or the *data matrix*.

¹Typically the dimensions are dropped when we write the regression model. They are included here for completeness.

Estimating the Beta Matrix

When we fit a regression model to a dataset, one of the goals is to estimate the regression coefficients, β_0 and β_1 . We can do this by using the regression equation,

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

Re-writing the equation to isolate the error vector, we get

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

In OLS estimation, we are interested in finding the coefficients that minimize the sum of squared error. The sum of squared error can be expressed in matrix notation as $\epsilon'\epsilon$.

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

Using the rules of transposes and expanding this, we get,

$$\underbrace{\mathbf{Y}'\mathbf{Y}}_{1 \times 1} - \underbrace{\beta'\mathbf{X}'\mathbf{Y}}_{1 \times 1} - \underbrace{\mathbf{Y}'\mathbf{X}\beta}_{1 \times 1} + \underbrace{\beta'\mathbf{X}'\mathbf{X}\beta}_{1 \times 1}$$

Each term is a 1×1 matrix², which means that each term is equal to its transpose. We will re-write the third term $\mathbf{Y}'\mathbf{X}\beta$ as its transpose $\beta'\mathbf{X}'\mathbf{Y}$.

$$\mathbf{Y}'\mathbf{Y} - \beta'\mathbf{X}'\mathbf{Y} - \beta'\mathbf{X}'\mathbf{Y} - \beta'\mathbf{X}'\mathbf{X}\beta$$

Combining the two middle terms,

$$\mathbf{Y}'\mathbf{Y} - 2\beta'\mathbf{X}'\mathbf{Y} - \beta'\mathbf{X}'\mathbf{X}\beta$$

To find the values for the elements in β that minimize the equation, we differentiate with respect to β .

$$\frac{\delta}{\delta\beta} \mathbf{Y}'\mathbf{Y} - 2\beta'\mathbf{X}'\mathbf{Y} - \beta'\mathbf{X}'\mathbf{X}\beta$$

Differentiating this we get

$$-2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\beta$$

We set this equal to zero and solve for β .

$$\begin{aligned} 0 &= -2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\beta \\ 2\mathbf{X}'\mathbf{Y} &= 2\mathbf{X}'\mathbf{X}\beta \\ \mathbf{X}'\mathbf{Y} &= \mathbf{X}'\mathbf{X}\beta \end{aligned}$$

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

²You should verify that the dimension of each term is 1×1 .

$$\begin{aligned}
 (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{Y}) &= (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{X})\beta \\
 (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} &= \mathbf{I}\beta \\
 (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} &= \beta
 \end{aligned}$$

The equation,

$$\beta = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

implies that the vector of regression coefficients can be obtained directly through manipulation of the design matrix and the vector of outcomes. In other words, the coefficients that minimize the sum of squared residuals is a direct function of the data.

Example Using Data

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

```
Y = c(12, 8, 16.26, 13.65, 8.5)
X1 = c(32, 33, 32, 33, 26)
```

Say we wanted to fit a regression model using $X1$ to predict variation in Y . We can estimate the regression coefficients as,

```
# Create design matrix
X = matrix(
  data = c(rep(1, 5), X1),
  ncol = 2
)
```

X

```
##      [,1] [,2]
## [1,]    1  32
## [2,]    1  33
## [3,]    1  32
## [4,]    1  33
## [5,]    1  26
```

```
# Estimate coefficients
B = solve(t(X) %*% X) %*% t(X) %*% Y
B
```

```
##      [,1]
## [1,] -3.8982759
## [2,]  0.4993678
```

Let's compare this to the estimates given in the `lm()` function.

```
lm.1 = lm(Y ~ 1 + X1)
tidy(lm.1)
```

```
##           term      estimate std.error statistic  p.value
## 1 (Intercept) -3.8982759 19.3658346 -0.2012966 0.8533423
## 2           X1   0.4993678  0.6184927  0.8073949 0.4785185
```

They are identical. The `lm()` function actually uses the exact same matrix algebra to compute the regression coefficients that it outputs.

Fitted Values

The $n \times 1$ vector of *fitted values*, $\hat{\mathbf{Y}}$ can be computed as,

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \hat{Y}_3 \\ \vdots \\ \hat{Y}_n \end{bmatrix} = \mathbf{X}\boldsymbol{\beta}$$

In our example, we can compute the fitted values as

```
# Compute fitted values
X %*% B
```

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

```
# Verify using the fitted() function
fitted(lm.1)
```

```
##           1           2           3           4           5
## 12.081494 12.580862 12.081494 12.580862  9.085287
```

H-Matrix (Hat Matrix)

One useful matrix in regression is the **H**-matrix. To obtain the **H**-matrix we substitute the matrix formulation of $\boldsymbol{\beta}$ into the equation to compute the fitted values,

$$\begin{aligned} \hat{\mathbf{Y}} &= \mathbf{X}\boldsymbol{\beta} \\ &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ &= \mathbf{H}\mathbf{Y} \end{aligned}$$

The **H**-matrix is then,

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$$

This matrix has the following properties,

- **H** is a square $n \times n$ matrix
- **H** is a symmetric matrix.
- **H** is an idempotent matrix.
- It can be created completely from the design matrix and its transpose.

In our example,

```
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
```

We can also use the **H**-matrix to compute the fitted values,

```
# Compute fitted values
h %*% Y
```

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

The fitted values can be expressed as linear combinations of the response vector **Y** using coefficients found in **H**. Because of this, **H** is often referred to as the *hat matrix*.

One of the properties of **H** is *idempotency*. This is a property where multiplying a matrix by itself results in the original matrix.

$$\mathbf{H}\mathbf{H} = \mathbf{H}$$

We can verify this using R

```
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
```

Residual Matrix

The $n \times 1$ vector of *residuals*, ϵ can be computed as,

$$\epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix} = \mathbf{Y} - \hat{\mathbf{Y}}$$

Doing a little substitution,

$$\begin{aligned} \epsilon &= \mathbf{Y} - \hat{\mathbf{Y}} \\ &= \mathbf{Y} - \mathbf{X}\beta \\ &= \mathbf{Y} - \mathbf{H}\mathbf{Y} \\ &= (\mathbf{I} - \mathbf{H})\mathbf{Y} \end{aligned}$$

Thus the residuals can also be expressed as linear combinations of the response vector \mathbf{Y} . The matrix $(\mathbf{I} - \mathbf{H})$ has some of the same properties as \mathbf{H} , namely, it is

- Square
- Symmetric
- Idempotent

We can compute the residual vector using R

```
# Create 5x5 identity matrix
i = diag(5)

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

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

```
# Verify the residual vector
resid(lm.1)
```

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

Estimation of Error Variance

Recall that the estimate of the error variance, or Mean Squared Error ($\hat{\sigma}_\epsilon^2$), is

$$\hat{\sigma}_\epsilon^2 = \frac{SS_{\text{Error}}}{df_{\text{Error}}}$$

Using matrices, the sum of squared error (SS_{Error}) is

$$SS_{\text{Error}} = \epsilon' \epsilon$$

The df_{Error} is

$$n - k$$

where k is the number of coefficients being estimated in the model. In the simple regression case, $k = 2$.

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

```
##           [,1]
## [1,] 39.93647
```

```
# Compute df for the residuals
n = 5
num_coef = 2
df_residual = n - num_coef

# Compute error variance estimate
mse = ss_error / df_residual
mse
```

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

We can verify these results by examining the ANOVA decomposition using the `anova()` function.

```
anova(lm.1)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df Sum Sq Mean Sq F value Pr(>F)
## X1         1  8.678   8.678   0.6519 0.4785
## Residuals  3 39.936  13.312
```

Variance-Covariance Matrix of the Regression Coefficients

Remember that the standard error (SE) for a coefficient represents the uncertainty in the estimate. We use them to construct a test statistic (z or t) or confidence interval.

While the SEs are useful for CIs and tests because they are on the same scale as the metric used to measure the estimate, in mathematical applications the variances are more commonly used. We typically provide the variances in a variance-covariance matrix of the coefficients. In this matrix the variances for each of the coefficients is given on the main diagonal of the matrix. The covariances between the coefficients are given in the off-diagonal elements. For example, the variance-covariance matrix for a simple regression model is

$$\sigma_B^2 = \begin{bmatrix} \sigma_{b_0}^2 & \sigma_{b_0, b_1} \\ \sigma_{b_0, b_1} & \sigma_{b_1}^2 \end{bmatrix}$$

Examining this, we find that the variance-covariance matrix is a square, symmetric matrix. To compute the variance-covariance matrix of the regression coefficients

$$\sigma_B^2 = \sigma_\epsilon^2 (\mathbf{X}'\mathbf{X})^{-1}$$

Based on this formula, we see that the variance of the estimator increases when the error term is more noisy (σ_ϵ^2 increases). However, it decreases when the variation in X increases.³ This is because having more observations, allows for more accurate estimation of β . These also affect the size of the covariance terms on the off-diagonal.

In our example,

```
# (X'X)^(-1)
solve(t(X) %*% X)

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

# Variance-covariance matrix
V_b = 13.31216 * solve(t(X) %*% X)
V_b

##           [,1]      [,2]
## [1,] 375.03568 -11.9350400
## [2,] -11.93504  0.3825333
```

The variance of $\hat{\beta}_0$ is 375.04 and the variance for $\hat{\beta}_1$ is 0.38. The covariance between $\hat{\beta}_0$ and $\hat{\beta}_1$ is -11.94 .

Standard Errors for the Coefficients

The standard error for each regression coefficient can be found by computing the square root of the variance estimates.⁴

```
sqrtdiag(V_b)

## [1] 19.3658380  0.6184928
```

³Convince yourself of this.

⁴Verify this with the output from the `tidy()` function.

Correlation Between the Coefficients

Recall that the correlation is the standardized covariance. We can obtain the correlation coefficient between $\hat{\beta}_0$ and $\hat{\beta}_1$, which is more interpretable than the covariance using

$$r_{b_0, b_1} = \frac{\sigma_{b_0, b_1}}{\sqrt{\sigma_{b_0}^2 \sigma_{b_1}^2}}$$

Using the values from our example,

$$\begin{aligned} r_{b_0, b_1} &= \frac{-11.93504}{\sqrt{375.03568 \times 0.3825333}} \\ &= -0.996 \end{aligned}$$

If we have assigned the variance-covariance matrix to an object, we can use indexing to access the different elements to compute the correlation.

```
V_b[1, 2] / sqrt(V_b[1, 1] * V_b[2, 2])
```

```
## [1] -0.9964441
```

The correlation indicates that the regression coefficients are highly, negatively correlated. How can one set of regression coefficients have a correlation? You have to remember the thought experiments about repeated sampling from a population that we talked about in EPsy 8251/8252. The estimated correlation gives the *expected correlation* of the regression coefficients across multiple random samples. In our example, it is telling you that if you did the experiment again and it so happened that the intercept was smaller, it is likely that the slope coefficient would be larger.

In practice, the variance-covariance matrix of the regression coefficients can be obtained directly from R using the `vcov()` function.

```
vcov(lm.1)
```

```
##              (Intercept)              X1
## (Intercept)  375.03555 -11.9350358
## X1          -11.93504   0.3825332
```

Regression Assumptions

There are several assumptions about the errors for the regression model, namely that:

- They are independent of one another;
- The average error is 0; and
- They have constant variance

We can express these assumptions using matrix notation. The independence assumption implies that the covariance between two errors is 0.

$$\sigma(\epsilon_j, \epsilon_k) = 0$$

for all $j \neq k$.

The assumption that the average residual (at each X) is zero can be written as:

$$E(\epsilon) = \mathbf{0}$$

$$E \left(\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The constant variance assumption says that the variance of the residuals (at each X) is the same, σ_ϵ^2 .

Variance-Covariance Matrix of the Residuals

If we combine this with the independence assumption, we can write out the variance-covariance matrix of the residuals.

$$\sigma^2(\epsilon) = \begin{bmatrix} \sigma_\epsilon^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_\epsilon^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_\epsilon^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_\epsilon^2 \end{bmatrix}$$

Notice that the variance-covariance matrix of the residuals is a scalar matrix and can also be expressed as the product of σ_ϵ^2 and an $n \times n$ identity matrix

$$\begin{aligned} \sigma^2(\epsilon) &= \sigma_\epsilon^2 \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \\ &= \sigma_\epsilon^2 \mathbf{I} \end{aligned}$$

In our toy example,

```
# Variance-covariance matrix of residuals
13.31216 * diag(5)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 13.31216  0.00000  0.00000  0.00000  0.00000
## [2,]  0.00000 13.31216  0.00000  0.00000  0.00000
## [3,]  0.00000  0.00000 13.31216  0.00000  0.00000
## [4,]  0.00000  0.00000  0.00000 13.31216  0.00000
## [5,]  0.00000  0.00000  0.00000  0.00000 13.31216
```

Writing the Regression Model

We can use the matrix expressions of the model equation and assumptions to completely specify the regression model. Namely,

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

where $\boldsymbol{\epsilon}$ is a vector of independent random variables with $E(\boldsymbol{\epsilon}) = 0$ and $\boldsymbol{\sigma}^2(\boldsymbol{\epsilon}) = \sigma_{\epsilon}^2 \mathbf{I}$.

References