SDS 4130/5130: Linear Statistical Models

Topic 4: Multiple Linear Regression Model III

Fitted and Residual Values, Hat Matrix, and MSE

Maxine Yu[1]

[1] Department of Statistics and Data Science
https://maxineyu.github.io/personal_web/

Washington University in St. Louis

September 3, 2025

Today's Class

1. Fitted Values, the Hat Matrix, and Residuals

2. Estimation of σ^2, $SSE = SS_{Res}$, and MSE

Fitted Values

- The fitted value corresponding to the levels of the regressor variables $X_1 = x_1, \ldots, X_k = x_k$ is defined as

$$\hat{\mathbf{y}} = \hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j x_j = \mathbf{x}'\hat{\boldsymbol{\beta}},$$

  where $\mathbf{x}' = [1, x_1, \ldots, x_k]$ and $\hat{\boldsymbol{\beta}} = [\hat{\beta}_0, \ldots, \hat{\beta}_k]'$ is the OLSE.
- $\hat{\mathbf{y}}$ is a point estimate of the mean value of the response, $E[Y|X_1 = x_1, \ldots, X_k = x_k] = \beta_0 + \sum_{j=1}^{k} \beta_j x_j$, for a case with predictor values $X_1 = x_1, \ldots, X_k = x_k$.
- The fitted value $\hat{y}_i$ corresponding to the i^{th}-case is

$$\hat{y}_i = \mathbf{x}_i'\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j x_{ij}, \quad i = 1, \ldots, n,$$

  where $\mathbf{x}_i' = [1, x_{i1}, \ldots, x_{ik}]$.

Vector of Fitted Values

- The column vector

$$\hat{\mathbf{y}} = [\hat{y}_1, \ldots, \hat{y}_n]'$$

  is called the vector of fitted values.

- If we recall that the design matrix **X** takes the form:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1k} \\ 1 & x_{21} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \ldots & x_{nk} \end{bmatrix},$$

  we have the following short-hand formula for $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}$$

- The fitted values can be found using the function `fitted(model)` in R.

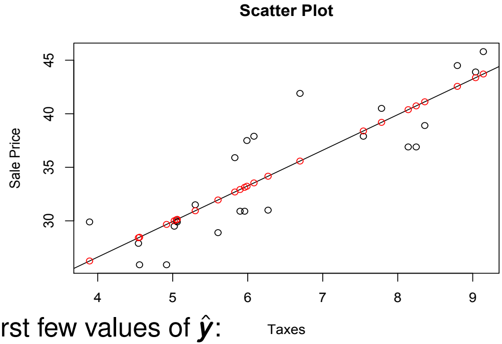
Example 1

Recall the Example 1 from Lect. 2 regarding sale prices of houses. Let us regress the sale price y against the first predictor x_1 (taxes). The following code shows how to generate the scatter plot of y vs. x_1, the least-squares regression line, and the fitted values (red points):

```
> data(table.b4)
> mymodel<-lm(y~x1,data=table.b4)
> plot(table.b4$x1,table.b4$y,xlab='Taxes',ylab='Sale Price',main=
> abline(coef(mymodel))
> points(table.b4$x1,fitted(mymodel),col='red')
```

Example 1. Continued...

We generate the following graph:



These are the first few values of $\hat{\boldsymbol{y}}$:

```
>fitted(mymodel)
       1        2        3        4        5        6        7 ...
30.01118 28.42247 28.47034 30.14050 26.25531 32.92732 31.94962...
```

The Hat Matrix

- It has some advantages to write things as linear combinations of the observations vector $\mathbf{y} = [y_1, \ldots, y_n]'$. This is easy to do for $\hat{\mathbf{y}}$ since we known that $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ and, thus,

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} =: \mathbf{H}\mathbf{y},$$

  where **H** is called the hat matrix and is defined as

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

- This matrix maps the vector of observed values into a vector of fitted values.

Comments

The hat matrix $\mathbf{H} = [h_{ij}]$ allows us to write the fitted values as linear combinations of the observed response variables y_j's:

$$\hat{y}_i = \sum_{j=1}^{n} h_{ij} y_j.$$

The entry h_{ij} is sometimes called the leverage of the j^{th} response in the i^{th} fitted value and can be interpreted as the effect that j^{th} response in the i^{th} fitted value.

Vector Of Residuals

- The difference between the observed and the fitted values,

$$e_i := y_i - \hat{y}_i,$$

  is called the residual of the i^{th}-case.

- The column vector $\mathbf{e} = [e_1, \ldots, e_n]'$ is called the vector of residuals.

- It is easy to see that

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{H}\mathbf{y} = (\mathbf{I} - \mathbf{H})\mathbf{y},$$

  where $\mathbf{I} = \mathbf{I}_{n \times n}$ is the $n \times n$ identity matrix.

- The residuals can be found using the function `resid(model)` in R.

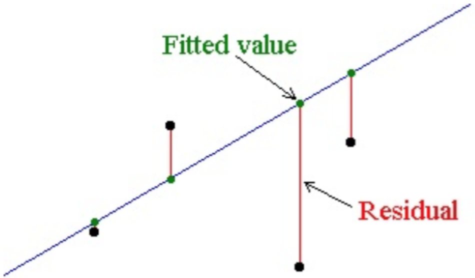
Interpretation Of Residuals

- Important: The residual e_i **is not** the error term $\varepsilon_i = Y_i - \mathbb{E}(Y_i)$;
  But, if the model has a good fit so that $\hat{\beta} \approx \beta$, then the residual can be though as a proxy for the error:

$$e_i \approx \varepsilon_i.$$

- In particular, residuals are useful in assessing the statistical properties of the errors ε_i and analyzing the appropriateness of the model (model diagnostics).

Illustration of Fitted Values and Residuals for $k = 1$

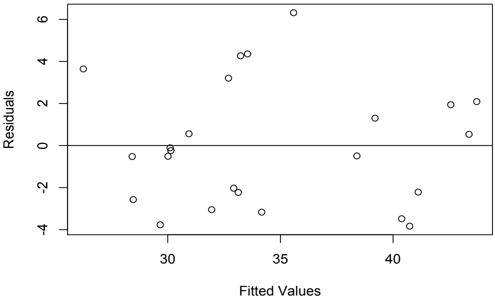


- The blue line represents the least-squares line $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$
- The black dots are the data points (x_i, y_i)
- The green dots are the fitted values $(x_i, \hat{y}_i)$
- The vertical distance between the green and black dots are the residuals e_i

Example 2

This is a continuation of the code of the Example 1. The following code generates a plot of the residuals against the fitted values:

```
>plot(fitted(mymodel),resid(mymodel),xlab='Fitted Values',
ylab='Residuals')
>abline(h=0)
```



Properties of the Hat Matrix

The Hat Matrix $\mathbf{H} = [h_{ij}] := \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \in \mathbb{R}^{n \times n}$ has some interesting properties:

1. Symmetric: $\mathbf{H}' = \mathbf{H}$
2. Idempotent: $\mathbf{H}\mathbf{H} = \mathbf{H}$
3. $\mathbf{I} - \mathbf{H}$ is also symmetric and idempotent
4. $\mathbf{H}\mathbf{X} = \mathbf{X}$ and $\mathbf{X}'\mathbf{H} = \mathbf{X}'$.
5. The trace[1] $tr(\mathbf{H}) := \sum_{i=1}^{n} h_{ii}$ equals the number of coefficients $p = k + 1$:

$$tr(\mathbf{H}) = \mathbf{tr}(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}') = \mathbf{tr}(\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}) = \mathbf{tr}(\boldsymbol{I}_{\mathbf{p}\times\mathbf{p}}) = \mathbf{p}, \tag{1}$$

   (where we used the property $tr(\mathbf{AB}) = tr(\mathbf{BA})$)

6. $tr(\mathbf{I} - \mathbf{H}) = n - p$

[1] The trace of a matrix is the sum of the matrix's diagonal elements.

Estimation of σ^2 I

- As an application of the previous formulas and concepts, we now devise a "good" estimator for $\sigma^2 = \mathrm{Var}(\varepsilon_i)$, the variance of the regression errors ε_i.
- For simplicity, suppose for now that the errors $\varepsilon_1, \ldots, \varepsilon_n$ are iid (independent identically distributed). Recall from elementary statistics that the natural estimator of σ^2 is the sample variance of the ε_i's:
$$\frac{\sum_{i=1}^{n}(\varepsilon_i - \bar{\varepsilon})^2}{n-1}.$$
- However, we don't observe directly the errors ε_i!. Nevertheless, it is natural to use its observable proxies, the residuals e_i:
$$\tilde{\sigma}^2 = \frac{\sum_{i=1}^{n}(e_i - \bar{e})^2}{n-1} = \frac{\sum_{i=1}^{n} e_i^2}{n-1},$$
where the last equality is because, as it turns out, $\bar{e} = \frac{\sum_{i=1}^{n} e_i}{n}$ is always equal to 0 (see p. 19 below).

Estimation of σ^2 II

- The numerator of $\tilde{\sigma}^2$ is called the sum of square errors or sum of the residuals squares:

$$SSE = SS_{Res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2.$$

- As it turns out $\mathbb{E}[SS_{Res}] = \sigma^2(n-p)$ (see pages 17-18 below). Then, $E(\tilde{\sigma}^2) = \frac{1}{n-1}E(SS_{Res}) = \sigma^2\left(\frac{n-p}{n-1}\right) < \sigma^2$ because $p = k+1 \geq 2$.
- However, the following estimator is UNBIASED for σ^2:

$$s^2 = \hat{\sigma}^2 = \frac{SS_{Res}}{n-p} = \frac{\sum_{i=1}^{n} e_i^2}{n-p}.$$

This is called Mean-Square Error and is sometimes denoted as MSE or MS_{Res}. From now on, we shall use s^2 as our estimator for σ^2.

Connection between residuals and errors

To show the unbiasedness statement of the last point above, we first need to connect the residuals $\boldsymbol{e}$ with the errors ε. Remember that $\mathbf{e} = (\mathbf{I} - \mathbf{H})\mathbf{y}$ and $\mathbf{y} = \mathbf{X}\beta + \varepsilon$. Then,

$$\boldsymbol{e} = (\mathbf{I} - \mathbf{H})(\mathbf{X}\beta + \varepsilon) = (\mathbf{I} - \mathbf{H})\mathbf{X}\beta + (\mathbf{I} - \mathbf{H})\varepsilon.$$

But, the first term cancels out because
$(\mathbf{I} - \mathbf{H})\mathbf{X}\beta = (\mathbf{IX} - \mathbf{HX})\beta = (\mathbf{X} - \mathbf{X})\beta = \mathbf{0}$ (here, we used that $\mathbf{HX} = \mathbf{X}$). Then,

$$\boldsymbol{e} = (\mathbf{I} - \mathbf{H})\varepsilon.$$

Verification that $E(SS_{Res}) = \sigma^2(n-p)$ I

- Note that
$$SS_{Res} = \sum_{i=1}^{n} e_i^2 = \mathbf{e}'\mathbf{e}.$$
- Then, using the connection between $\mathbf{e}$ and $\boldsymbol{\varepsilon}$ of the previous page,
$$SS_{Res} = ((\mathbf{I}-\mathbf{H})\varepsilon)'(\mathbf{I}-\mathbf{H})\varepsilon = \varepsilon'(\mathbf{I}-\mathbf{H})'(\mathbf{I}-\mathbf{H})\varepsilon = \varepsilon'(\mathbf{I}-\mathbf{H})\varepsilon,$$
because $\mathbf{I}-\mathbf{H}$ is symmetric and idempotent. We conclude that
$$\mathbb{E}\left[SS_{Res}\right] = \mathbb{E}\left[\boldsymbol{\varepsilon}'(\mathbf{I}-\mathbf{H})\boldsymbol{\varepsilon}\right].$$
- Let us denote $\boldsymbol{A} = \mathbf{I}-\mathbf{H}$. We can show that
$$\boldsymbol{\varepsilon}'\mathbf{A}\boldsymbol{\varepsilon} = \sum_{i,j=1}^{n} a_{i,j}\varepsilon_i\varepsilon_j,$$
where $a_{i,j}$ is the (i,j) entry of $\boldsymbol{A}$.

Verification that $E(SS_{Res}) = \sigma^2(n-p)$ II

- Then,

$$\mathbb{E}[SS_{Res}] = \mathbb{E}(\boldsymbol{\varepsilon}'\mathbf{A}\boldsymbol{\varepsilon}) = \sum_{i,j=1}^{n} a_{i,j}\mathbb{E}(\varepsilon_i\varepsilon_j).$$

- Remember the assumptions we have for the errors (see Topic 2, page 5):
  - $\mathbb{E}(\varepsilon_i) = 0$,
  - $\mathrm{Var}(\varepsilon_i) = \sigma^2$ (Homoskedasticity Assumption),
  - $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$, for $i \neq j$.

  Then, when $i = j$, $\mathbb{E}(\varepsilon_i\varepsilon_j) = \mathbb{E}(\varepsilon_i^2) = \mathrm{Var}(\varepsilon_i) = \sigma^2$, and, when $i \neq j$, $\mathbb{E}(\varepsilon_i\varepsilon_j) = \mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$. Thus, we have $\sum_{i,j=1}^{n} a_{i,j}\mathbb{E}(\varepsilon_i\varepsilon_j) = \sigma^2 \sum_{i=1}^{n} a_{i,i} = \sigma^2\mathrm{tr}(\mathbf{A}) = \mathrm{tr}\,(\mathbf{I} - \mathbf{H})$.

- Finally, using the last property of page 13,

$$\mathbb{E}[SS_{Res}] = \sigma^2\mathrm{tr}\,(\mathbf{I} - \mathbf{H}) = \sigma^2(n-p).$$

Verification of $\bar{e} = 0$

- To show that $\bar{e} = 0$ or, equivalently, $\sum_{i=1}^{n} e_i = 0$, we can use the first equation of the normal equations given in page 5 of Topic 3:

$$-2\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \sum_{j=1}^{k}\hat{\beta}_j x_{ij}\right) = -2\sum_{i=1}^{n} e_i = 0.$$

Finding $\hat{\sigma}^2$ in R I

1. The value of $\hat{\sigma}$ is directly reported in the summary of the `lm` function as the “Residual standard error".
2. The following shows three different methods to compute the MSE $\hat{\sigma}^2$ in the case of Table.B4 in Montgomery et al. (sale prices of 24 houses), regressing the sale price y against x_1 (taxes) and x_3 (lot size). Exercise: Run the code and check you get the same value.

Finding $\hat{\sigma}^2$ in R II

```
# Extracting the residual standard error or MSE
mymodel<-lm(y~x1+x3,data=table.b4)
# Method 1
sigma(mymodel)
# Method 2
(summ<-summary(mymodel))
(MSE<-summ$sig^2)
# method 3
e<-resid(mymodel)
(SS_Res<-t(e)%*%e)
(MSE2<-SS_Res/(24-3))
```

Note the use of `resid` to find the residuals, `t(A)` to find the transpose, and $\% * \%$ for matrix multiplication.

Estimated Variance-Covariance Matrix of $\hat{\beta}$ in R I

- The natural estimate of the covariance matrix $\mathrm{Cov}(\hat{\beta}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$ is

$$s^2(\hat{\beta}) := \widehat{\mathrm{Cov}(\hat{\beta})} := \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$$

- The function `vcov(mymodel)` returns the above matrix.
- The following code show the use of this function.

Estimated Variance-Covariance Matrix of $\hat{\beta}$ in R II

```
# Regress House Sale Price against x1=taxes (in thousand dollars)
and x3 lot size (in thousand of square feet)
data(table.b4)
mymodel<-lm(y~x1+x3,data=table.b4)
summ<-summary(mymodel)
> # Extracting the Covariance Matrix of the OLSE
> # Method 1: Use of vcov function:
> (V=vcov(mymodel))
            (Intercept)         x1         x3
(Intercept)   6.8693708 -0.9133021 -0.1066774
x1           -0.9133021  0.2987683 -0.1658113
x3           -0.1066774 -0.1658113  0.1937260
```

Estimated Variance-Covariance Matrix of $\hat{\beta}$ in R III

Let us verify the answer using matrix operations:

```
# Method 2: by hand
> X<-model.matrix(mymodel)
> C<-solve(t(X)%*% X)
> sigma(mymodel)^2*C
            (Intercept)         x1         x3
(Intercept)   6.8693708 -0.9133021 -0.1066774
x1           -0.9133021  0.2987683 -0.1658113
x3           -0.1066774 -0.1658113  0.1937260
```

Note the use of the function `model.matrix(mymodel)` to extract the design matrix **X** and `solve(A)` to find A^{-1}.