The Gauss–Markov theorem

As before, we assume:

\[ Y = X_1 \beta_1 + \cdots + X_p \beta_p = X^T \beta. \]
The Gauss–Markov theorem

As before, we assume:

\[ Y = X_1 \beta_1 + \cdots + X_p \beta_p = X^T \beta. \]

We observe \( X \in \mathbb{R}^{n \times p}, \ Y \in \mathbb{R}^n \). Then

\[ \hat{\beta}_{LS} = (X^T X)^{-1} X^T Y. \]
The Gauss–Markov theorem

As before, we assume:

\[ Y = X_1 \beta_1 + \cdots + X_p \beta_p = X^T \beta. \]

We observe \( X \in \mathbb{R}^{n \times p}, Y \in \mathbb{R}^n \). Then

\[ \hat{\beta}_{LS} = (X^T X)^{-1} X^T Y. \]

Under some natural assumptions, we can show that \( \hat{\beta}_{LS} \) is the best linear unbiased estimator for \( \beta \).
The Gauss–Markov theorem

As before, we assume:

\[ Y = X_1 \beta_1 + \cdots + X_p \beta_p = X^T \beta. \]

We observe \( X \in \mathbb{R}^{n \times p}, \ Y \in \mathbb{R}^n \). Then

\[ \hat{\beta}_{LS} = (X^T X)^{-1} X^T Y. \]

Under some natural assumptions, we can show that \( \hat{\beta}_{LS} \) is the best linear unbiased estimator for \( \beta \).

Assumptions: \( \ Y = X \beta + \epsilon \), where \( \epsilon \in \mathbb{R}^n \) with:

1. \( \mathbb{E}(\epsilon_i) = 0 \)
2. \( \text{Var}(\epsilon_i) = \sigma^2 < \infty \)
3. \( \text{Cov}(\epsilon_i, \epsilon_j) = 0 \) for all \( i \neq j \).

Note: (3) means that the errors are uncorrelated. In particular, (3) holds if the errors are independent.

The errors need not be normal, nor independent, nor identically distributed.
The Gauss–Markov theorem

As before, we assume:

\[ Y = X_1 \beta_1 + \cdots + X_p \beta_p = X^T \beta. \]

We observe \( X \in \mathbb{R}^{n \times p}, \ Y \in \mathbb{R}^n \). Then

\[ \hat{\beta}_{LS} = (X^T X)^{-1} X^T Y. \]

Under some natural assumptions, we can show that \( \hat{\beta}_{LS} \) is the best linear unbiased estimator for \( \beta \).

Assumptions: \( Y = X \beta + \epsilon \), where \( \epsilon \in \mathbb{R}^n \) with:

1. \( E(\epsilon_i) = 0 \).
2. \( \text{Var}(\epsilon_i) = \sigma^2 < \infty \).
3. \( \text{Cov}(\epsilon_i, \epsilon_j) = 0 \) for all \( i \neq j \).
The Gauss–Markov theorem

As before, we assume:

\[ Y = X_1 \beta_1 + \cdots + X_p \beta_p = X^T \beta. \]

We observe \( X \in \mathbb{R}^{n \times p} \), \( Y \in \mathbb{R}^n \). Then

\[ \hat{\beta}_{LS} = (X^T X)^{-1} X^T Y. \]

Under some natural assumptions, we can show that \( \hat{\beta}_{LS} \) is the best linear unbiased estimator for \( \beta \).

Assumptions: \( Y = X\beta + \epsilon \), where \( \epsilon \in \mathbb{R}^n \) with:

1. \( E(\epsilon_i) = 0 \).
2. \( \text{Var}(\epsilon_i) = \sigma^2 < \infty \).
3. \( \text{Cov}(\epsilon_i, \epsilon_j) = 0 \) for all \( i \neq j \).

Note:

- (3) means that the errors are uncorrelated. In particular, (3) holds if the errors are independent.
- The errors need not be normal, nor independent, nor identically distributed.
Remarks: In our model $Y = X\beta + \epsilon$,

- $X$ is fixed.
- $\epsilon$ is random.
- $Y$ is random.
- $\beta$ is fixed, but unobservable.

We want to estimate $\beta$. 
Remarks: In our model $Y = X\beta + \epsilon$,

- $X$ is fixed.
- $\epsilon$ is random.
- $Y$ is random.
- $\beta$ is fixed, but unobservable.

We want to estimate $\beta$.

A linear estimator of $\beta$, is an estimator of the form $\hat{\beta} = CY$, where $C = (c_{ij}) \in \mathbb{R}^{p \times n}$ is a matrix, and

$$c_{ij} = c_{ij}(X).$$

Note: $\hat{\beta}$ is random since $Y$ is assumed to be random.
Remarks: In our model $Y = X\beta + \epsilon$,

- $X$ is fixed.
- $\epsilon$ is random.
- $Y$ is random.
- $\beta$ is fixed, but unobservable.

We want to estimate $\beta$.

A linear estimator of $\beta$, is an estimator of the form $\hat{\beta} = CY$, where $C = (c_{ij}) \in \mathbb{R}^{p \times n}$ is a matrix, and

$$c_{ij} = c_{ij}(X).$$

Note: $\hat{\beta}$ is random since $Y$ is assumed to be random.

In particular, $\hat{\beta}_{LS} = (X^TX)^{-1}X^TY$ is a linear estimator with $C = (X^TX)^{-1}X^T$. 
Remarks: In our model $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, 
- $\mathbf{X}$ is fixed.
- $\epsilon$ is random.
- $\mathbf{Y}$ is random.
- $\beta$ is fixed, but unobservable.

We want to estimate $\beta$.

A *linear* estimator of $\beta$, is an estimator of the form $\hat{\beta} = C\mathbf{Y}$, where $C = (c_{ij}) \in \mathbb{R}^{p \times n}$ is a matrix, and

$$
c_{ij} = c_{ij}(\mathbf{X}).$$

Note: $\hat{\beta}$ is random since $\mathbf{Y}$ is assumed to be random.

In particular, $\hat{\beta}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ is a linear estimator with 
$C = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$.

An estimator is *unbiased* if $E(\hat{\beta}) = \beta$. 
Ultimately, we want to use $\hat{\beta}$ to predict $Y$, i.e.,
\[
\hat{Y}_i = X_{i1}\hat{\beta}_1 + X_{i2}\hat{\beta}_2 + \cdots + X_{ip}\hat{\beta}_p.
\]
We want to control to error of the prediction.
Ultimately, we want to use $\hat{\beta}$ to predict $Y$, i.e.,
$$\hat{Y}_i = X_{i1}\hat{\beta}_1 + X_{i2}\hat{\beta}_2 + \cdots + X_{ip}\hat{\beta}_p.$$  
We want to control to error of the prediction.

We define the mean squared error (MSE) of a linear combination of the coefficients of $\hat{\beta}$ by

$$\text{MSE}(a^T\hat{\beta}) = E \left[ \left( \sum_{i=1}^{n} a_i(\hat{\beta}_i - \beta_i) \right)^2 \right] \quad (a \in \mathbb{R}^p).$$
Ultimately, we want to use $\hat{\beta}$ to predict $Y$, i.e.,

$$\hat{Y}_i = X_{i1}\hat{\beta}_1 + X_{i2}\hat{\beta}_2 + \cdots + X_{ip}\hat{\beta}_p.$$ We want to control to error of the prediction.

We define the mean squared error (MSE) of a linear combination of the coefficients of $\hat{\beta}$ by

$$\text{MSE}(a^T \hat{\beta}) = E \left[ \left( \sum_{i=1}^{n} a_i(\hat{\beta}_i - \beta_i) \right)^2 \right] \quad (a \in \mathbb{R}^p).$$

**Theorem (Gauss–Markov theorem)**

Suppose $Y = X\beta + \epsilon$ where $\epsilon$ satisfies the previous assumptions. Let $\hat{\beta} = Cy$ be a linear unbiased estimator of $\beta$. Then for all $a \in \mathbb{R}^p$,

$$\text{MSE}(a^T \hat{\beta}_{LS}) \leq \text{MSE}(a^T \hat{\beta}).$$

We say that $\hat{\beta}_{LS}$ is the best linear unbiased estimator (BLUE) of $\beta$. 

The bias-variance tradeoff

Let \( Z = a^T \beta \) and \( \hat{Z} = a^T \hat{\beta} \). (Note: \( Z \) is non-random). Then

\[
MSE(a^T \hat{\beta}) = E \left[ (a^T (\hat{\beta} - \beta))^2 \right] = E \left[ (\hat{Z} - Z)^2 \right] \\
= E(Z^2 - 2Z\hat{Z} + \hat{Z}^2) \\
= E(Z^2) - 2E(Z\hat{Z}) + E(\hat{Z}^2) \\
= Z^2 - 2ZE(\hat{Z}) + \text{Var}(\hat{Z}) + E(\hat{Z})^2 \\
= (Z - E(\hat{Z}))^2 + \text{Var}(\hat{Z}).
\]

\( \text{bias}^2 + \text{variance} \)
The bias-variance tradeoff

Let \( Z = a^T \beta \) and \( \hat{Z} = a^T \hat{\beta} \). (Note: \( Z \) is non-random). Then

\[
MSE(a^T \hat{\beta}) = E \left[ (a^T (\hat{\beta} - \beta))^2 \right] = E \left[ (\hat{Z} - Z)^2 \right] \\
= E(Z^2 - 2Z \hat{Z} + \hat{Z}^2) \\
= E(Z^2) - 2E(Z \hat{Z}) + E(\hat{Z}^2) \\
= Z^2 - 2ZE(\hat{Z}) + \text{Var}(\hat{Z}) + E(\hat{Z})^2 \\
= (Z - E(\hat{Z}))^2 + \text{Var}(\hat{Z}).
\]

Therefore, \( \text{MSE} = \text{Bias-squared} + \text{Variance} \).

As a result, if \( \hat{\beta} \) is unbiased, then \( \text{MSE}(a^T \beta) = \text{Var}(\hat{Z}) \).
We now prove the Gauss–Markov theorem.
We now prove the Gauss–Markov theorem. Using the bias-variance decomposition of MSE, it suffices to show that for every unbiased estimator of $\beta$, 

$$\text{Var}(a^T \hat{\beta}_{LS}) \leq \text{Var}(a^T \hat{\beta}) \quad \forall a \in \mathbb{R}^p.$$
We now prove the Gauss–Markov theorem. Using the bias-variance decomposition of MSE, it suffices to show that for every unbiased estimator of $\beta$,

$$\text{Var}(a^T \hat{\beta}_{LS}) \leq \text{Var}(a^T \hat{\beta}) \quad \forall a \in \mathbb{R}^p.$$ 

**Proof.** Let $\hat{\beta} = CY$ where $C = (X^TX)^{-1}X^T + D$ for some $D \in \mathbb{R}^{p \times n}$. We will compute $E(\hat{\beta})$ and $\text{Var}(a^T \hat{\beta})$.

$$E(\hat{\beta}) = E \left[ ((X^TX)^{-1}X^T + D)Y \right]$$

$$= E \left[ ((X^TX)^{-1}X^T + D)(X\beta + \epsilon) \right]$$

$$= (I + DX)\beta.$$
We now prove the Gauss–Markov theorem. Using the bias-variance decomposition of MSE, it suffices to show that for every unbiased estimator of $\beta$,

$$\text{Var}(a^T \hat{\beta}_{\text{LS}}) \leq \text{Var}(a^T \hat{\beta}) \quad \forall a \in \mathbb{R}^p.$$ 

**Proof.** Let $\hat{\beta} = CY$ where $C = (X^TX)^{-1}X^T + D$ for some $D \in \mathbb{R}^{p \times n}$. We will compute $E(\hat{\beta})$ and $\text{Var}(a^T \hat{\beta})$.

$$E(\hat{\beta}) = E\left[ ((X^TX)^{-1}X^T + D)Y \right]$$
$$= E\left[ ((X^TX)^{-1}X^T + D)(X\beta + \epsilon) \right]$$
$$= (I + DX)\beta.$$ 

In order for $\hat{\beta}$ to be unbiased, we need $DX = 0$. 
We now prove the Gauss–Markov theorem. Using the bias-variance decomposition of MSE, it suffices to show that for every unbiased estimator of $\beta$,

$$\text{Var}(a^T \hat{\beta}_{\text{LS}}) \leq \text{Var}(a^T \hat{\beta}) \quad \forall a \in \mathbb{R}^p.$$ 

**Proof.** Let $\hat{\beta} = C Y$ where $C = (X^T X)^{-1} X^T + D$ for some $D \in \mathbb{R}^{p \times n}$. We will compute $E(\hat{\beta})$ and $\text{Var}(a^T \hat{\beta})$.

$$E(\hat{\beta}) = E \left[ ((X^T X)^{-1} X^T + D) Y \right]$$
$$= E \left[ ((X^T X)^{-1} X^T + D)(X \beta + \epsilon) \right]$$
$$= (I + DX) \beta.$$

In order for $\hat{\beta}$ to be unbiased, we need $DX = 0$. We now compute $\text{Var}(a^T \hat{\beta})$. 

6/14
Recall:

\[ \text{Var}(a^T \hat{\beta}) = a^T \Sigma a, \]

where \( \Sigma = (\text{Cov}(\hat{\beta}_i, \hat{\beta}_j)) = \text{Var}(\hat{\beta}). \)
Recall:

\[ \text{Var}(a^T \hat{\beta}) = a^T \Sigma a, \]

where \( \Sigma = (\text{Cov}(\hat{\beta}_i, \hat{\beta}_j)) = \text{Var}(\hat{\beta}) \). More generally, if \( A \in \mathbb{R}^{p \times p} \), then

\[ \text{Var}(A\hat{\beta}) = A \text{Var}(\hat{\beta}) A^T. \]
Recall:

\[
\text{Var}(a^T \hat{\beta}) = a^T \Sigma a,
\]

where \( \Sigma = \text{Cov}(\hat{\beta}_i,\hat{\beta}_j) = \text{Var}(\hat{\beta}) \). More generally, if \( A \in \mathbb{R}^{p \times p} \), then

\[
\text{Var}(A\hat{\beta}) = A \text{Var}(\hat{\beta}) A^T.
\]

Using these formulas, we obtain

\[
\text{Var}(\hat{\beta}) = \text{Var}(CY)
\]

\[
= C \text{Var}(Y) C^T = \sigma^2 CC^T
\]

\[
= \sigma^2 ((X^T X)^{-1}X^T + D)((X^T X)^{-1}X^T + D)^T
\]

\[
= \sigma^2 (X^T X)^{-1}X^T X (X^T X)^{-1}
\]

\[
+ \sigma^2 \left[ (X^T X)^{-1} \underbrace{X^T D^T}_{= (DX)^T = 0} + \underbrace{DX (X^T X)^{-1} + DD^T}_{= 0} \right]
\]

\[
= \sigma^2 \left[ (X^T X)^{-1} + DD^T \right].
\]
We have shown:

$$\text{Var}(\hat{\beta}) = \sigma^2 (X^T X)^{-1} + \sigma^2 DD^T.$$
We have shown:

$$\text{Var}(\hat{\beta}) = \sigma^2 (X^T X)^{-1} + \sigma^2 D D^T.$$ 

Note that the matrices $(X^T X)^{-1}$ and $D D^T$ are positive semidefinite.
We have shown:

\[ \text{Var}(\hat{\beta}) = \sigma^2 (X^T X)^{-1} + \sigma^2 D D^T. \]

Note that the matrices \((X^T X)^{-1}\) and \(D D^T\) are positive semidefinite.

Therefore,

\[ \text{Var}(a^T \hat{\beta}) = a^T (\sigma^2 (X^T X)^{-1} + \sigma^2 D D^T) a \geq a^T \sigma^2 (X^T X)^{-1} a \]
\[ = \text{Var}(a^T \hat{\beta}_{LS}). \]

This concludes the proof. \(\square\)
Back to bias-variance tradeoff

We saw that

$$\text{MSE}(a^T \hat{\beta}) = (a^T \beta - E(a^T \hat{\beta}))^2 + \text{Var}(a^T \hat{\beta}).$$
We saw that

\[ \text{MSE}(a^T \hat{\beta}) = (a^T \beta - E(a^T \hat{\beta}))^2 + \text{Var}(a^T \hat{\beta}). \]

Moreover, according to the Gauss–Markov theorem, for every unbiased estimator \( \hat{\beta} \),

\[ \text{MSE}(a^T \hat{\beta}_{LS}) = \text{Var}(a^T \hat{\beta}_{LS}) \leq \text{MSE}(a^T \hat{\beta}) \]
Back to bias-variance tradeoff

We saw that

$$\text{MSE}(a^T \hat{\beta}) = (a^T \beta - E(a^T \hat{\beta}))^2 + \text{Var}(a^T \hat{\beta}).$$

Moreover, according to the Gauss–Markov theorem, for every unbiased estimator $\hat{\beta}$,

$$\text{MSE}(a^T \hat{\beta}_{LS}) = \text{Var}(a^T \hat{\beta}_{LS}) \leq \text{MSE}(a^T \hat{\beta})$$

Problems with least squares:

1. Least squares estimates often have large variance, and can have low prediction accuracy (especially when working with small samples).
We saw that

\[
\text{MSE}(a^T \hat{\beta}) = (a^T \beta - E(a^T \hat{\beta}))^2 + \text{Var}(a^T \hat{\beta}).
\]

Moreover, according to the Gauss–Markov theorem, for every unbiased estimator \( \hat{\beta} \),

\[
\text{MSE}(a^T \hat{\beta}_{\text{LS}}) = \text{Var}(a^T \hat{\beta}_{\text{LS}}) \leq \text{MSE}(a^T \hat{\beta})
\]

**Problems with least squares:**

1. Least squares estimates often have large variance, and can have low prediction accuracy (especially when working with small samples).
2. Generally, all the regression coefficients \( \beta_i \) are nonzero, making the model hard to interpret. Often, we want to identify the relevant variables to get the “big picture”.

W e can often increase the prediction accuracy by sacrificing a little bit of bias to reduce the variance of the estimator. We will later examine some useful alternatives to least squares.
We saw that

$$\text{MSE}(a^T \hat{\beta}) = (a^T \beta - E(a^T \hat{\beta}))^2 + \text{Var}(a^T \hat{\beta}).$$

Moreover, according to the Gauss–Markov theorem, for every unbiased estimator $\hat{\beta}$,

$$\text{MSE}(a^T \hat{\beta}_\text{LS}) = \text{Var}(a^T \hat{\beta}_\text{LS}) \leq \text{MSE}(a^T \hat{\beta}).$$

**Problems with least squares:**

1. Least squares estimates often have large variance, and can have low prediction accuracy (especially when working with small samples).

2. Generally, all the regression coefficients $\beta_i$ are nonzero, making the model hard to interpret. Often, we want to identify the *relevant* variables to get the “big picture”.

We can often increase the prediction accuracy by sacrificing a little bit of bias to reduce the variance of the estimator.
We saw that
\[
\text{MSE}(a^T \hat{\beta}) = (a^T \beta - E(a^T \hat{\beta}))^2 + \text{Var}(a^T \hat{\beta}).
\]
Moreover, according to the Gauss–Markov theorem, for every unbiased estimator \(\hat{\beta}\),
\[
\text{MSE}(a^T \hat{\beta}_{LS}) = \text{Var}(a^T \hat{\beta}_{LS}) \leq \text{MSE}(a^T \hat{\beta})
\]

Problems with least squares:

1. Least squares estimates often have large variance, and can have low prediction accuracy (especially when working with small samples).

2. Generally, all the regression coefficients \(\beta_i\) are nonzero, making the model hard to interpret. Often, we want to identify the relevant variables to get the “big picture”.

We can often increase the prediction accuracy by sacrificing a little bit of bias to reduce the variance of the estimator. We will later examine some useful alternatives to least squares.
Training error and test error

A natural way to improve least squares is to force some of the coefficients to be zero.

- Resulting estimator is biased, but can benefit from the bias-variance tradeoff.
- Model is easier to interpret.

Complexity of the model:

A complex model that fits data very well will often make poor predictions.

Overtting.

On the other hand, a very simple model may not capture the complexity of the data.

Undertting.

To test the ability of a model to predict new values:

1. We split our data into 2 parts (training data and test data) as uniformly as possible. People often use 75% training, 25% test.
2. We train our model using the training data only. (This minimizes the training error).
3. We use the fitted model to predict values of the test data and compute the test error.
Training error and test error

A natural way to improve least squares is to force some of the coefficients to be zero.

- Resulting estimator is biased, but can benefit from the bias-variance tradeoff.
- Model is easier to interpret.

Complexity of the model:

- A complex model that fits data very well will often make poor predictions. **Overfitting**.
A natural way to improve least squares is to force some of the coefficients to be zero.

- Resulting estimator is biased, but can benefit from the bias-variance tradeoff.
- Model is easier to interpret.

Complexity of the model:
- A complex model that fits data very well will often make poor predictions. **Overfitting**.
- On the other hand, a very simple model may not capture the complexity of the data. **Underfitting**.
A natural way to improve least squares is to force some of the coefficients to be zero.

- Resulting estimator is biased, but can benefit from the bias-variance tradeoff.
- Model is easier to interpret.

Complexity of the model:

- A complex model that fits data very well will often make poor predictions. **Overfitting.**
- On the other hand, a very simple model may not capture the complexity of the data. **Underfitting.**

To test the ability of a model to predict new values:

1. We split our data into 2 parts (training data and test data) as uniformly as possible. People often use 75% training, 25% test.
Training error and test error

A natural way to improve least squares is to force some of the coefficients to be zero.

- Resulting estimator is biased, but can benefit from the bias-variance tradeoff.
- Model is easier to interpret.

Complexity of the model:

- A complex model that fits data very well will often make poor predictions. **Overfitting**.
- On the other hand, a very simple model may not capture the complexity of the data. **Underfitting**.

To test the ability of a model to predict new values:

1. We split our data into 2 parts (training data and test data) as uniformly as possible. People often use 75% training, 25% test.
2. We fit our model using the training data only. (This minimizes the training error).
A natural way to improve least squares is to force some of the coefficients to be zero.

- Resulting estimator is biased, but can benefit from the bias-variance tradeoff.
- Model is easier to interpret.

Complexity of the model:

- A complex model that fits data very well will often make poor predictions. **Overfitting**.
- On the other hand, a very simple model may not capture the complexity of the data. **Underfitting**.

To test the ability of a model to predict new values:

1. We split our data into 2 parts (training data and test data) as uniformly as possible. People often use 75% training, 25% test.
2. We fit our model using the training data only. (This minimizes the **training error**).
3. We use the fitted model to predict values of the test data and compute the **test error**.
Splitting data into training/test data:

In the case of least squares:

1. $\hat{\beta} = \left( X^T_{\text{train}} X_{\text{train}} \right)^{-1} X^T_{\text{train}} Y_{\text{train}}$.

2. $\hat{Y}_{\text{test}} = X_{\text{test}} \hat{\beta}$.

3. Test error:
   
   $$\text{MSE}_{\text{test}} = \frac{1}{n_2} \sum_{i=1}^{n_2} \left( \hat{Y}_{\text{test},i} - Y_{\text{test},i} \right)^2.$$
Splitting data into training/test data:

In the case of least squares:

1. \( \hat{\beta} = (X_{\text{train}}^T X_{\text{train}})^{-1} X_{\text{train}}^T Y_{\text{train}} \).

Test error:

\[
\text{MSE}_{\text{test}} = \frac{1}{n_2} \sum_{i=1}^{n_2} (\hat{Y}_{\text{test},i} - Y_{\text{test},i})^2.
\]

We choose a model that minimizes the test error.
Splitting data into training/test data:

In the case of least squares:

1. $\hat{\beta} = (X_{\text{train}}^T X_{\text{train}})^{-1} X_{\text{train}}^T Y_{\text{train}}$.
2. $\hat{Y}_{\text{test}} = X_{\text{test}} \beta$.

Test error:

$\text{MSE}_{\text{test}} = \frac{1}{n_{2}} \sum_{i=1}^{n_{2}} (\hat{Y}_{\text{test},i} - Y_{\text{test},i})^2$.

We choose a model that minimizes the test error.
Splitting data into training/test data:

In the case of least squares:

1. $\hat{\beta} = (X_{\text{train}}^T X_{\text{train}})^{-1} X_{\text{train}}^T Y_{\text{train}}$.
2. $\hat{Y}_{\text{test}} = X_{\text{test}} \beta$.
3. Test error:

$$\text{MSE}_{\text{test}} = \frac{1}{n_2} \sum_{i=1}^{n_2} (\hat{Y}_{\text{test},i} - Y_{\text{test},i})^2.$$
Splitting data into training/test data:

1. $\hat{\beta} = (X_{\text{train}}^T X_{\text{train}})^{-1} X_{\text{train}}^T Y_{\text{train}}$.
2. $\hat{Y}_{\text{test}} = X_{\text{test}} \hat{\beta}$.
3. Test error:

$$\text{MSE}_{\text{test}} = \frac{1}{n_2} \sum_{i=1}^{n_2} (\hat{Y}_{\text{test},i} - Y_{\text{test},i})^2.$$ 

We choose a model that minimizes the test error.
Typical behavior of the test and training error, as model complexity is varied.

ESL, Fig 2.11.
Scikit-learn provides a function to split the data automatically for us.
Scikit-learn provides a function to split the data automatically for us.

```python
from sklearn.cross_validation import train_test_split

# Split data into training and test sets
X_train, X_test, y_train, y_test =
    train_test_split(X, y, test_size=0.25,
                     random_state=42)

# Fit model on training data
lin_model = LinearRegression(fit_intercept=True)
lin_model.fit(X_train,y_train)

# Returns the coefficient of determination R^2.
lin_model.score(X_test, y_test)
```
Regression models are often ranked using the \textit{coefficient of determination} called “R squared” and denoted $R^2$.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}.$$
Regression models are often ranked using the *coefficient of determination* called “R squared” and denoted $R^2$.

\[
R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}.
\]

In some sense, the $R^2$ measures “how much better” is the prediction, compared to a constant prediction equal to the average of the $y_i$s.
Regression models are often ranked using the *coefficient of determination* called “R squared” and denoted $R^2$.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}.$$ 

In some sense, the $R^2$ measures “how much better” is the prediction, compared to a constant prediction equal to the average of the $y_i$s.

The score method in sklearn returns the $R^2$. 
Regression models are often ranked using the coefficient of determination called “R squared” and denoted $R^2$.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}.$$ 

In some sense, the $R^2$ measures “how much better” is the prediction, compared to a constant prediction equal to the average of the $y_i$s.

The score method in sklearn returns the $R^2$.

We want a model with a test $R^2$ as close to 1 as possible.