

# **MAT555E: Statistical Data Analysis for Computational Sciences**

Fall22-Lecture 03: Multiple Linear Regression

---

Gül İnan

İstanbul Technical University

# Learning Objectives

- Introduction to multiple linear regression
- Least-squares estimation
- Statistical tests
- Model selection criteria
- Implementing a multiple linear regression model with Python `statsmodels` library

# Multiple Linear Regression Model

---

## Multiple linear regression model set-up

- A **regression model** that involves **more than one independent variable** is called a **multiple linear regression (MLR) model**.
- Consider a MLR model with **p independent** variables:

$$Y = \beta_0 + \beta_1 * X_1 + ... + \beta_j * X_j + ... + \beta_p * X_p + \epsilon,$$

- where  $Y$  represents the response variable,  $X_j$ 's ( $j = 1, 2, \dots, p$ ) are independent variables, and with all the assumptions imposed on  $\epsilon$  in SLR.
- Alternatively,

$$E(Y) = \beta_0 + \beta_1 * X_1 + ... + \beta_p * X_p.$$

## Interpretation of regression coefficients

- The parameters  $\beta_j$  ( $j = 0, 1, 2, \dots, p$ ) are called the **regression coefficients**.
- If the **range of the data** includes  $X_1 = X_2 = \dots = X_p = 0$ , then  $\beta_0$  is the expected value of  $Y$  when  $X_1 = X_2 = \dots = X_p = 0$ . Otherwise,  $\beta_0$  has **no physical interpretation**.
- The parameter  $\beta_j$  ( $j = 1, 2, \dots, p$ ) represents the change expected in  $Y$  per unit change in  $X_j$  **when all of the remaining independent variables are held constant**  
 $(\beta_j = \frac{\partial E(Y_i)}{\partial X_j}, j = 1, 2, \dots, p)$ .
- For this reason, the parameters  $\beta_j$  ( $j = 1, 2, \dots, p$ ) are often called **partial regression coefficients**.

## Marginal and partial effects of regression coefficients

- The effect of  $\beta$  in the linear model with a single predictor  $X$  is **usually not the same** as the effect of  $\beta$  of the same variable  $X$  in a model with multiple independent variables.
- The effect  $\beta$  is a marginal effect, **ignoring all other potential independent variables**, whereas  $\beta$  is a partial effect, **conditioning on the other independent variables**.

## Multiple linear regression model set-up

- Suppose we have given a random sample of size  $n$  such that  $\{(X_{i1}, \dots, X_{ip}, Y_i)\}_{i=1}^n$ .
- Then, at individual data point level, we can write down the MLR model as:

$$Y_i = \beta_0 + \beta_1 * X_{i1} + \dots + \beta_p * X_{ip} + \epsilon_i, \quad i = 1, \dots, n.$$

- Alternatively,

$$E(Y_i) = \beta_0 + \beta_1 * X_{i1} + \dots + \beta_p * X_{ip}, \quad i = 1, \dots, n.$$

## Model geometric interpretation

- In MLR, the equation  $E(Y_i) = \beta_0 + \beta_1 * X_{i1} + \dots + \beta_p * X_{ip}$  describes a **p-dimensional regression hyperplane** in the  $(p+1)$ -dimensional space of  $Y, X_1, \dots, X_p$ .
- The parameter  $\beta_0$  is the **intercept** of the **p-dimensional hyperplane**.

In linear algebra: the equation above defines a **p-dimensional hyperplane** where  $\beta_0$  is the off-set from the origin and  $\langle \beta_1, \dots, \beta_p \rangle$  is the vector normal to the hyperplane.



## Recall

- In **SLR model**, the equation  $E(Y_i) = \beta_0 + \beta_1 * X_{i1}$  describes a **regression line** in the **X-Y plane** and the parameter  $\beta_0$  is the **y-intercept**.
- But, we never know the true values of  $\beta_0$  and  $\beta_1$ .
- So, in SLR, actually, we are trying to **estimate the regression line** equation  $\widehat{E(Y_i)} = \widehat{Y_i} = \hat{\beta}_0 + \hat{\beta}_1 * X_{i1}$  or
- In other words, we are trying to **fit the equation**  $\widehat{Y_i} = \hat{\beta}_0 + \hat{\beta}_1 * X_{i1}$  to the data, which is called as **fitted regression line, least-squares line** etc.

## Revisiting Advertising Data

---

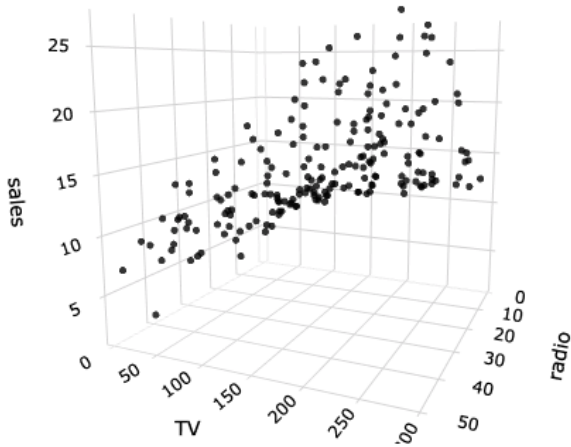
## Revisiting advertising data

- Consider the **advertising** data which consists of sales of a product in 200 different markets, along with advertising budgets for the product in each of those markets for three different media: TV, radio, and newspaper.
- Now assume a **multiple linear regression model** for sales with **two predictors**: TV and radio such that:

$$sales_i = \beta_0 + \beta_1 * TV_i + \beta_2 * radio_i + \epsilon_i$$

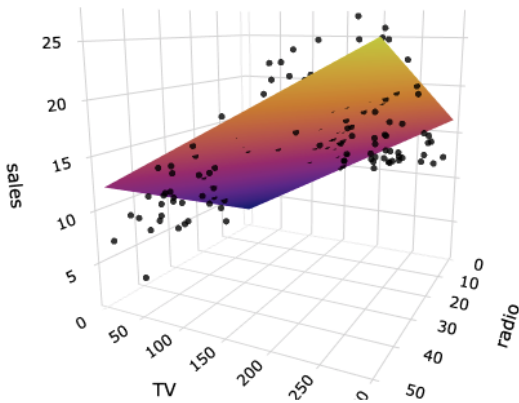
- for  $i = 1, 2, \dots, 200$  and  $\epsilon_i \sim N(0, \sigma^2)$ .

## 3D-scatter plot of advertising data



## Fitting a regression plane

- Since we have **two predictors**, now, we are trying to **fit a regression plane**,  $\widehat{sales}_i = \hat{\beta}_0 + \hat{\beta}_1 * TV_i + \hat{\beta}_2 * radio_i$  to the advertising data.



## Multiple Linear Regression Model Continued

---

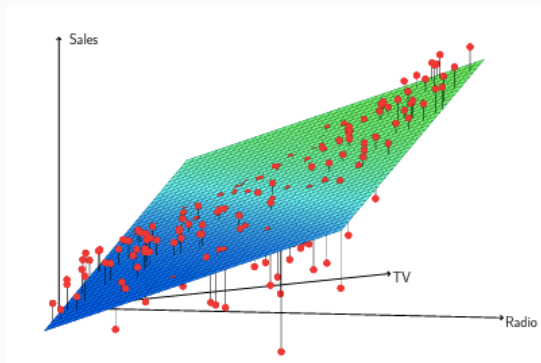
## Least-squares estimation

- In the MLR model, for given a observed sample of size  $n$  such that  $\{(x_{i1}, \dots, x_{ip}, y_i)\}_{i=1}^n$ , the method of least-squares estimates the regression coefficients which **minimizes the residual sum-of-sum squares (RSS)**:

$$RSS(\hat{\beta}) = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 * x_{i1} + \dots + \hat{\beta}_p * x_{ip}))^2.$$

## Least-squares estimation

- For example, for advertising data with two predictors, we are trying to fit a **regression plane**.
- This plane is chosen to **minimize the sum of the squared vertical distances** between each observation (shown in red) and the plane.





## Finding least-squares estimators

- Let  $\mathcal{J}(\beta) = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 * x_{i1} + \dots + \beta_p * x_{ip}))^2$  be the **objective function**.
- To minimize  $\mathcal{J}(\beta)$ , we take partial derivatives with respect to  $\beta_j$  ( $j = 0, 1, 2, \dots, p$ ), set them to zero, and solve the resulting system of equations simultaneously, such that.

$$\left. \frac{\partial \mathcal{J}(\beta)}{\partial \beta_0} \right|_{\hat{\beta}} = -2 \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 * x_{i1} + \dots + \hat{\beta}_p * x_{ip})) = 0 \quad (1)$$

$$\left. \frac{\partial \mathcal{J}(\beta)}{\partial \beta_j} \right|_{\hat{\beta}} = -2 \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 * x_{i1} + \dots + \hat{\beta}_p * x_{ip})) x_{ij} = 0$$

- for  $j = 1, 2, \dots, p$ .

## Finding least-squares estimators

- However, solving the resulting **homogeneous system of  $(p+1)$  linear equations in  $(p+1)$  unknowns** simultaneously is not algebraically easy.

## Matrix notation

- Then, denote  $n \times 1$  response vector  $\mathbf{Y}$ ,  $n \times (p + 1)$  design matrix  $\mathbf{X}$ ,  $(p + 1) \times 1$  regression vector  $\beta$ , and  $n \times 1$  error vector  $\epsilon$ , respectively, as:

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & X_{11} & \cdots & X_{1p} \\ 1 & X_{21} & \cdots & X_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & X_{n1} & \cdots & X_{np} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}.$$

# Matrix notation

- Then, the MLR model can be written in the **matrix form**:

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

- The **least-squares estimates** of  $\beta$  is the solution to the following **optimization** problem.

$$\hat{\beta}_{OLS} = \underset{\beta}{\operatorname{argmin}} \|\epsilon\|_2^2 = \underset{\beta}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 = \underset{\beta}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta),$$

- since  $\sum_{i=1}^n \epsilon_i^2 = \epsilon^T \epsilon = \|\epsilon\|_2^2$ .

## Finding least-squares estimators

- Denote the objective function  $\mathcal{J}(\beta) = (\mathbf{y} - \mathbf{x}\beta)^T(\mathbf{y} - \mathbf{x}\beta)$  which is scalar, find the **gradient vector** of  $\mathcal{J}(\beta)$  with respect to  $\beta$  such that:

$$\nabla \mathcal{J}(\beta) = \frac{\partial \mathcal{J}(\beta)}{\partial \beta} = \begin{pmatrix} \frac{\partial \mathcal{J}(\beta)}{\partial \beta_0} \\ \frac{\partial \mathcal{J}(\beta)}{\partial \beta_1} \\ \vdots \\ \frac{\partial \mathcal{J}(\beta)}{\partial \beta_p} \end{pmatrix},$$

- and then set the **gradient to zero**  $\nabla \mathcal{J}(\beta) = 0$ .

## Finding least-squares estimators

- Let's expand  $\mathcal{J}(\beta)$  first:

$$\begin{aligned}\mathcal{J}(\beta) &= ((\mathbf{y} - \mathbf{x}\beta)^T(\mathbf{y} - \mathbf{x}\beta)) = ((\mathbf{y}^T - \beta^T \mathbf{x}^T)(\mathbf{y} - \mathbf{x}\beta)) \\ &= \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{x}\beta + \beta^T \mathbf{x}^T \mathbf{x}\beta,\end{aligned}$$

- since the dot product  $\mathbf{y}^T \mathbf{x}\beta$  is scalar,  $(\beta^T \mathbf{x}^T \mathbf{y})^T = \mathbf{y}^T \mathbf{x}\beta$ .

## Finding least-squares estimators

- Applying the rules for **differentiation of a scalar with respect to a vector**, we get:

$$\nabla \mathcal{J}(\beta) = \frac{\partial \mathcal{J}(\beta)}{\partial \beta} = -2\mathbf{x}^T \mathbf{y} + 2\mathbf{x}^T \mathbf{x} \beta,$$

- since  $\frac{\partial a^T \beta}{\partial \beta} = a$  and  $\frac{\partial \beta^T \mathbf{S} \beta}{\partial \beta} = 2\mathbf{S} \beta$ , where  $\mathbf{S}$  is a symmetric matrix.

## Finding least-squares estimators

- Setting the gradient vector  $\nabla \mathcal{J}(\beta) = 0$ , we get the **normal equations**:

$$\mathbf{x}^T \mathbf{y} = \mathbf{x}^T \mathbf{x} \beta.$$

- Here we assume that **the number of columns in  $\mathbf{x}$**  is less than the number of rows such as  $(p + 1) < n$  and the rank of  $\mathbf{x}$  is  $(p + 1)$ , hence  $\mathbf{x}$  has a **full column rank**  $(p + 1)$ .



## Least-squares estimators

- Since  $\text{rank}(\mathbf{x}^T \mathbf{x}) = \text{rank}(\mathbf{x}) = (p + 1)$ , this leads the symmetric matrix  $\mathbf{x}^T \mathbf{x}$  to have a **full rank** of  $(p + 1)$ .
- In this case  $\det(\mathbf{x}^T \mathbf{x}) \neq 0$ , that the square matrix  $\mathbf{x}^T \mathbf{x}$  is non-singular and, in turn, is **invertible**.
- Then the normal equations yield the **least-squares estimator** of  $\beta$  as:

$$\hat{\beta}_{OLS} = (\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{y},$$

- where  $\mathbf{x}^T \mathbf{x}$  is a  $(p + 1) \times (p + 1)$  **symmetric matrix** and  $\mathbf{x}^T \mathbf{y}$  is a  $(p + 1) \times 1$  vector.
- Note that  $\mathbf{x}^T \mathbf{x}$  is also known as **Gram matrix** of  $\mathbf{x}$ .

## The form of $\mathbf{x}^T \mathbf{x}$ matrix and $\mathbf{x}^T \mathbf{y}$ vector

$$\mathbf{x}^T \mathbf{x} = \begin{pmatrix} n & \sum_{i=1}^n x_{i1} & \sum_{i=1}^n x_{i2} & \dots & \sum_{i=1}^n x_{ip} \\ \sum_{i=1}^n x_{i1} & \sum_{i=1}^n x_{i1}^2 & \sum_{i=1}^n x_{i1}x_{i2} & \dots & \sum_{i=1}^n x_{i1}x_{ip} \\ . & . & . & . & . \\ . & . & . & . & . \\ \sum_{i=1}^n x_{ip} & \sum_{i=1}^n x_{i1}x_{ip} & \sum_{i=1}^n x_{i2}x_{ip} & \dots & \sum_{i=1}^n x_{ip}^2 \end{pmatrix} \quad \text{and}$$

$$\mathbf{x}^T \mathbf{y} = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_{i1}y_i \\ . \\ . \\ \sum_{i=1}^n x_{ip}y_i \end{pmatrix} .$$

## Issues with Inverse of $\mathbf{x}^T \mathbf{x}$ Matrix

---

## Linearly independent

A collection of  $(p+1)$  elements in a vector space are linearly dependent if at least one element in this collection can be expressed as a linear combination of the remaining  $p$  elements. If no element, however, can be expressed in this fashion, then the  $(p+1)$  elements are linearly independent.

## Linearly independent columns and full column rank

- Here we assumed that our design matrix  $\mathbf{x}$  has a **full column rank**  $(p + 1)$  (for  $(p + 1) < n$ ) which implies that  $(p + 1)$  columns are linearly independent of each other.
- Statistically, this further implies that the explanatory variables, are **linearly independent of each other!!!**
- If the columns of  $\mathbf{x}$  are **linearly related**, then  $\mathbf{x}$  has a column rank  $r$  which is **less than**  $(p+1)$ .

## Linearly dependent columns, multicollinearity problem, and OLS

- This issue in statistics leads to **multicollinearity** problem.
- In statistics, the concept of **multicollinearity** refers to a situation in which **more than two explanatory variables in a multiple regression model are highly linearly related**.
- In case of multicollinearity, the design matrix  $\mathbf{x}$  has less than full column rank and some of the singular values of  $\mathbf{x}$  will be **zero**, this implies that at least one of the eigenvalues of  $\mathbf{x}^T \mathbf{x}$  is equal to zero, then the square matrix  $\mathbf{x}^T \mathbf{x}$  is singular, and therefore the square matrix  $\mathbf{x}^T \mathbf{x}$  cannot be inverted.
- Note: The singular values of  $\mathbf{x}$  are the positive square roots of the eigenvalues of  $\mathbf{x}^T \mathbf{x}$ .

## Multicollinearity problem and OLS

- Under these circumstances, the ordinary least squares estimator  $\hat{\beta}_{OLS} = (\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{y}$  does not exist.
- In practice, such **perfect multicollinearities rarely occur** in **statistical applications**.
- Rather, the columns of  $\mathbf{x}$  may be **nearly** linearly related.
- In this case, the rank of  $\mathbf{x}$  is  $(p+1)$ , but some of the singular values of  $\mathbf{x}$  will be **near zero**.

## Generalized inverse

- The literature shows that any solution of  $\mathbf{x}^T \mathbf{y} = \mathbf{x}^T \mathbf{x} \beta$  is of the form  $\mathbf{G} \mathbf{x}^T \mathbf{y}$  with  $\mathbf{G}$  a generalized inverse of  $\mathbf{x}^T \mathbf{x}$ .
- **Moore Penrose inverse** is the most commonly used generalized matrix inversion approach.



## Generalized inverse with numpy and statsmodels

- For data analysis problems when the code is written from scratch, to avoid matrix inversion failure problems, you can use `numpy.linalg.pinv()` which computes the (Moore-Penrose) pseudo-inverse of a matrix rather than `numpy.linalg.inv()`.
- For example, `OLS.fit()` in statsmodels library uses Moore Penrose inverse as a **default method** to solve the least squares problem.

## Generalized inverse with scikit-learn and scipy

- Interestingly, at first sight, `LinearRegression` in `scikit-learn` seems that it does not use any generalized matrix inversion approach.
- However, in the notes, to my understanding, it says that it depends on `scipy.linalg.lstsq` and `scipy.optimize.nnls`.

## Multicollinearity problem and OLS

- Nevertheless, the presence of multicollinearity in  $\mathbf{x}$  has adverse effects on the least-squares estimate  $\hat{\beta}_{OLS}$ .
- When the determinant of  $(\mathbf{x}^T \mathbf{x})^{-1}$  is close to zero, the matrix elements get very large in magnitude.
- Since  $\hat{\beta}_{OLS} = (\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{y}$  and  $Var(\hat{\beta}) = \sigma^2 (\mathbf{x}^T \mathbf{x})^{-1}$  depends on  $(\mathbf{x}^T \mathbf{x})^{-1}$ ,
- Large variance associated with the elements of  $\hat{\beta}_{OLS}$  can therefore be expected and this causes  $\hat{\beta}_{OLS}$  to become an unreliable estimate for  $\beta$ .
- This problem will take us to the **ridge regression** around Week 7.

## How to check multicollinearity?

- One quick **mathematical way** to check for multicollinearity is to calculate:

$$\kappa(\mathbf{x}) = \sqrt{\frac{e_{max}(\mathbf{x}^T \mathbf{x})}{e_{min}(\mathbf{x}^T \mathbf{x})}},$$

- where  $e_{max}$  and  $e_{min}$  are the maximum and minimum eigenvalues of  $\mathbf{x}^T \mathbf{x}$ , respectively. -  $\kappa(\mathbf{x})$  is less than 10, then there is **no serious problem with multicollinearity**. - Values of  $\kappa(\mathbf{x})$  between 10 and 30 indicate **moderate to strong multicollinearity**, and if  $\kappa(\mathbf{x}) > 30$ , **severe multicollinearity** is implied.

## How to remedy multicollinearity?

- One approach is to calculate the correlation coefficient between each pair of explanatory variables and omit the explanatory variable from the data analysis which is highly correlated with the rest.
- **Variance inflation factor**  $\rightarrow$  HW

## Multiple Linear Regression Model Continued

---

## The fitted values and residuals

- The **fitted value** vector  $\mathbf{\hat{y}}$  is:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y},$$

- where the matrix  $\mathbf{P} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$  is called as **projection matrix**, which results in  $\hat{\mathbf{y}}$  when right-multiplied with  $\mathbf{y}$ .
- The matrix  $\mathbf{P}$  is also an idempotent matrix such that  $\mathbf{P}^2 = \mathbf{P}$ .
- The **residual** vector  $\mathbf{e}$  is:

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) \mathbf{y}.$$

# The sampling distribution of $\hat{\beta}$

- The characteristics of  $\hat{\beta}$  are:
  - Centered at  $\beta$ , i.e.  $E(\hat{\beta}) = \beta$ .
  - $Var(\hat{\beta}) = \sigma^2(\mathbf{x}^T \mathbf{x})^{-1}$ .
  - $\hat{\beta} \sim N(0, \sigma^2(\mathbf{x}^T \mathbf{x})^{-1})$ .
- where the details are:
$$E(\hat{\beta}) = E((\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{y}) = (\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{x} \beta = \beta \text{ and}$$
- $Var(\hat{\beta}) = V((\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{y}) = (\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T V(\mathbf{y}) \mathbf{x} (\mathbf{x}^T \mathbf{x})^{-1} = \sigma^2(\mathbf{x}^T \mathbf{x})^{-1}$  where  $V(\mathbf{y}) = \sigma^2 \mathbf{I}$  and  $\mathbf{x}$  is not random.



## The form of $Var(\hat{\beta})$

- Note that  $Var(\hat{\beta}) = \sigma^2(\mathbf{x}^T \mathbf{x})^{-1}$  is a  $(p+1) \times (p+1)$  **symmetric matrix** called as the **variance-covariance matrix** of  $\hat{\beta}$ :

$$Var(\hat{\beta}) = \begin{pmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0, \hat{\beta}_1) & \dots & Cov(\hat{\beta}_0, \hat{\beta}_p) \\ Cov(\hat{\beta}_1, \hat{\beta}_0) & Var(\hat{\beta}_1) & \dots & Cov(\hat{\beta}_1, \hat{\beta}_p) \\ Cov(\hat{\beta}_2, \hat{\beta}_0) & Cov(\hat{\beta}_2, \hat{\beta}_1) & \dots & Cov(\hat{\beta}_2, \hat{\beta}_p) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(\hat{\beta}_p, \hat{\beta}_0) & Cov(\hat{\beta}_p, \hat{\beta}_1) & \dots & Var(\hat{\beta}_p) \end{pmatrix}.$$

- The variance of an individual OLS estimator  $\hat{\beta}_j$  ( $j = 0, 1, \dots, p$ ),  $Var(\hat{\beta}_j)$ , is the  $j$ -th **diagonal** element of the matrix  $Var(\hat{\beta})$ .

## Estimation of $\sigma^2$

- In many application, the parameter  $Var(\epsilon_i) = \sigma^2$  is **often unknown**.
- As in SLR, we can **estimate it from the data** through the formula:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n e_i^2}{n - p - 1} = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - p - 1} = \frac{(\mathbf{Y} - \hat{\mathbf{Y}})^T (\mathbf{y} - \hat{\mathbf{y}})}{n - p - 1} = \frac{RSS}{n - p - 1}.$$

# Statistical tests

---

## Hypothesis testing for $\beta_j$

- Suppose we wish to test the hypothesis that the **any regression parameter equals 0**, say,  $\beta_j = 0$  ( $j = 0, 1, \dots, p$ ).
- The appropriate **hypothesis** is:

$$H_0 : \beta_j = 0 \quad (2)$$

$$H_1 : \beta_j \neq 0$$

- Under  $H_0$ , the appropriate **test statistic** is:

$$t_j = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)} = \frac{\hat{\beta}_j}{\sqrt{\widehat{Var}(\hat{\beta}_j)}} \sim t_{(n-p-1)},$$

- where  $se(\hat{\beta}_j) = \sqrt{\hat{\sigma}^2 \text{diag}((\mathbf{x}^T \mathbf{x})^{-1})}$ .

## Hypothesis testing for $\beta_j$

- **Decision rule:** We would reject  $H_0 : \beta_j = 0$  at  $\alpha$  significance level, if  $t_j < -t_{1-\alpha/2, n-p-1}$  or  $t_j > t_{1-\alpha/2, n-p-1}$ , where  $t_{1-\alpha/2, n-p-1}$  denotes the  $100(1 - \alpha/2)$  percentile of the  $t$ -distribution with  $n - p - 1$  degrees of freedom.
- The term  $\alpha$  refers to the tolerance level for making a Type I error. (e.g., 10%, 5%, 1%).
- Alternatively, a P-value approach could also be used for decision making. We would reject  $H_0 : \beta_j = 0$  at  $\alpha$  significance level, if  $\text{P-value} < \alpha$ .

## Hypothesis testing for a subset of $q$ of $\beta_j$ 's

- Sometimes we want to test that a **particular subset of  $q$  of the coefficients** are zero.
- This corresponds to a null hypothesis:

$$H_0 : \beta_{p-q+1} = \beta_{p-q+2} = \dots = \beta_p = 0$$

$H_1$  : At least one of them is different from 0.

- where for convenience we have put the **variables chosen for the omission** at the end of the list.
- Under such a null hypothesis, we can **reduce the full model to a smaller model** (the model that uses all the variables except those last  $q$ ).

## Hypothesis testing for a subset of $q$ of $\beta_j$ 's

- Suppose that  $RSS$  and  $RSS_0$  be the residual sum-of-squares based on the least-squares fit of the **full model** and the **reduced smaller model**, respectively.
- If the null hypothesis is true, then these two quantities should be similar.
- Under  $H_0$ , the appropriate **test statistic** is:

$$F = \frac{(RSS_0 - RSS)/q}{RSS/(n - p - 1)} \sim F_{q, (n-p-1)}.$$

- **Decision rule:** We would reject  $H_0$  at  $\alpha$  significance level, if  $F > F_{1-\alpha, q, (n-p-1)}$ , where  $F_{1-\alpha, q, (n-p-1)}$  denotes the  $100(1 - \alpha)$  percentile of the  $F$ -distribution with  $q$  and  $(n - p - 1)$  degrees of freedom.

## Hypothesis testing for a subset of $q$ of $\beta_j$ 's

- For linear regression models, an **individual t-test** is equivalent to an **F-test** for dropping a single coefficient  $\beta_j$  from the model.



# Model Selection Criteria

---

# Akaike Information Criterion

- Akaike Information Criterion (AIC) proposed a general measure of “model badness:”

$$AIC = -2\log(\hat{\beta}) + 2m.$$

- where  $m$  is the number of parameters. The ‘best’ model can be chosen by seeing which has the lowest AIC. We must improve the log likelihood by one unit for every extra parameter.

# Bayesian Information Criterion

- Bayesian Information Criterion (BIC) penalizes complex models more severely is:

$$BIC = -2\log(\hat{\beta}) + m * \log(n).$$

- where  $m$  is the number of parameters and  $n$  is the number of data points.
- Lowest BIC is taken to identify the 'best model', as before.
- BIC tends to favor simpler models than those chosen by AIC.

## References

- Agresti, A., and Kateri, M. (2021). Foundations of Statistics for Data Scientists: With R and Python. Chapman and Hall/CRC.
- Fan, J., Li, R., Zhang, C.H., and Zou, H. (2020). Statistical Foundations of Data Science. Chapman and Hall/CRC.
- Fieller, N. (2018). Basics of Matrix Algebra for Statistics with R. Chapman and Hall/CRC.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). An Introduction to Statistical Learning: With Applications in R. New York: Springer.
- <https://online.stat.psu.edu/stat857/node/45/>
- <https://statproofbook.github.io/D/bic>