STAT 224 Lecture 2
Multiple Linear Regression, Part 1

Yibi Huang
Department of Statistics
University of Chicago

Outline

- What are multiple linear regression models
- Least squares estimation
- Fitted values, residuals, estimate of variance
- Interpretation of regression coefficients

What Are Multiple Linear Regression Models

Deterministic Models (No Errors)

Deterministic describe perfect relationships between variables w/ no errors

$$Y = f(X_1, X_2, \ldots, X_p)$$

Examples:

- Newton's second law of motion:

$$\underset{\text{(Force)}}{F} = \underset{\text{(mass)}}{m} \times \underset{\text{(acceleration)}}{a}$$

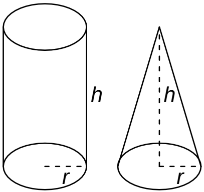
- Ideal gas law: $PV = nRT$

$$\underset{\substack{\text{pressure}\\\text{of gas}}}{P} \times \underset{\substack{\text{volume}\\\text{of gas}}}{V} = \underset{\substack{\text{amount of gas}\\\text{in moles}}}{n} \times \underset{\substack{\text{ideal gass}\\\text{constant}}}{R} \times \underset{\substack{\text{temperature}\\\text{in } ^\circ K}}{T}$$

Example: Timber Volume of Trees

Say we want to model timber volume of a tree as a function of its radius and height. If the trunk of a tree is a cylinder, then

$$\text{volume} = \pi r^2 h, \quad \text{where} \quad \begin{array}{l} r = \text{radius} \\ h = \text{height} \end{array}$$



If the trunk of a tree is a cone, then

$$\text{volume} = \frac{1}{3}\pi r^2 h$$

However, as tree trunks are not exactly cylinders or cones, the formulas above is subject to error. We may model the timber volume of a tree as a function of its radius and height w/ error.

$$\begin{aligned} \text{volume} &= f(r, h) + \varepsilon \\ &= \alpha r^2 h + \varepsilon \quad \text{where } \alpha \text{ is a constant.} \end{aligned}$$

Statistical Models

A Statistical model is a simple, low-dimensional (as fewer predictors as possible) summary of

- the relationship present in the data
- the data-generation process
- the relationship present in the population

Statistical models allow **errors** (**uncertainty**)

$$\underbrace{Y}_{\text{response}} = \underbrace{f(X_1, X_2, \ldots, X_p)}_{\text{deterministic function}} + \underbrace{\varepsilon}_{\text{error (noise)}}$$

Linear Regression Models

In STAT 22400, we focus on *linear* regression models where

$$\begin{aligned} Y &= f(X_1, X_2, \ldots, X_p) + \varepsilon \\ &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p + \varepsilon \end{aligned}$$

The adjective *linear* means the model is linear in its parameters $\beta_0, \beta_1, \ldots, \beta_p$. For example, the following **are** linear regression models

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$$
$$Y = \beta_0 + \beta_1 \log(X) + \varepsilon$$

even though the relationship between Y and X is not linear.

Some Non-linear Models Can Be Turned Linear (1)

Ex 1:

$$\begin{array}{lrcl}
\text{Non-linear model:} & Y & = & \dfrac{X}{\alpha X + \beta} \\
\text{reciprocal} & 1/Y & = & \alpha + \beta(1/X) \\
 & \downarrow & & \downarrow \quad \downarrow \\
\text{Linear model:} & Y' & = & \alpha + \beta X'
\end{array}$$

where $Y' = 1/Y$, $X' = 1/X$.

Some Non-linear Models Can Be Turned Linear (1)

Ex 1:

$$\begin{array}{lrcl}
\text{Non-linear model:} & Y & = & \dfrac{X}{\alpha X + \beta} \\
\text{reciprocal} & 1/Y & = & \alpha + \beta(1/X) \\
 & \downarrow & & \downarrow \quad\quad \downarrow \\
\text{Linear model:} & Y' & = & \alpha + \beta X'
\end{array}$$

where $Y' = 1/Y$, $X' = 1/X$.

Ex 2: Timber volume of trees $\approx cr^2h$ or more generally, $\alpha r^{\beta_1} h^{\beta_2}$

$$\begin{array}{lccccccc}
\text{Non-linear model:} & \text{Volume} & = & \alpha & \times & r^{\beta_1} & \times & h^{\beta_2} \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
\text{Taking logarithm} & \log(\text{Volume}) & = & \log(\alpha) & + & \beta_1 \log(r) & + & \beta_2 \log(h) \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
\text{Linear model:} & Y & = & \beta_0 & + & \beta_1 X_1 & + & \beta_2 X_2
\end{array}$$

where $Y = \log(\text{Volume})$, $X_1 = \log(r) = \log(\text{radius})$, and $X_2 = \log(h) = \log(\text{height})$.

Some Non-linear Models Can Be Turned Linear (2)

Ex 3: Production Function

In economics, the **Cobb-Douglas production function**,

$$V = \alpha K^{\beta_1} L^{\beta_2}, \quad \text{where} \quad \begin{aligned} V &= \text{output} \\ K &= \text{capital} \\ L &= \text{labor} \end{aligned}$$

is a widely used form of the production function to represent the relationship between the amounts of two or more inputs, particularly physical **capital** K and **labor** L, and the amount of **output** V that can be produced by those inputs. Despite of its **nonlinear** from, the production function can be turned into a linear model by taking the log of both sides,

$$\log(V) = \log(\alpha) + \beta_1 \log(K) + \beta_2 \log(L).$$

Which of the Following Models are Linear?

(a) $Y = \beta_0 + \beta_1^X + \varepsilon$

(b) $Y = \beta_0 \beta_1^X \varepsilon$

(c) $Y = \beta_0 + \beta_1 e^X + \varepsilon$

(d) $Y = \beta_0 + \beta_1 X^2 + \beta_2 \log(X) + \varepsilon$

Which of the Following Models are Linear?

(a) $Y = \beta_0 + \beta_1^X + \varepsilon$

(b) $Y = \beta_0 \beta_1^X \varepsilon$

(c) $Y = \beta_0 + \beta_1 e^X + \varepsilon$ .... Linear

(d) $Y = \beta_0 + \beta_1 X^2 + \beta_2 \log(X) + \varepsilon$ .... Linear

Which of the following models can be turned linear after transformation?

(a) $Y = \beta_0 + \beta_1^X + \varepsilon$

(b) $Y = \beta_0 \beta_1^X \varepsilon$

Which of the following models can be turned linear after transformation?

(a) $Y = \beta_0 + \beta_1^X + \varepsilon$
(b) $Y = \beta_0 \beta_1^X \varepsilon$

Ans: (b)

Data for Multiple Linear Regression Models

	SLR		MLR				
	X	Y	X_1	X_2	$\dots$	X_p	Y
case 1:	x_1	y_1	x_{11}	x_{12}	$\dots$	x_{1p}	y_1
case 2:	x_2	y_2	x_{21}	x_{22}	$\dots$	x_{2p}	y_2
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
case n:	x_n	y_n	x_{n1}	x_{n2}	$\dots$	x_{np}	y_n

- For SLR, we observe **pairs** of data values.
- For MLR, we observe **rows** of data values.
- Each row (or pair) is called a **case**, a **record**, or a **data point**
- y_i is the **response** (or **dependent variable**) of the ith case
- There are p **explanatory variables** (or **predictors**, **covariates**), and x_{ik} is the value of the explanatory variable X_k of the ith case

Multiple Linear Regression Models

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \varepsilon_i$$

In the model above,

- ε_i's (errors, or noise) are i.i.d. $N(0, \sigma^2)$
- Parameters include:
  - β_0 = intercept;
  - β_k = regression coefficient (slope) for the kth explanatory variable, $k = 1, \ldots, p$
  - σ^2 = Var(ε_i)= the variance of errors
- Observed (known): $y_i, x_{i1}, x_{i2}, \ldots, x_{ip}$
  Unknown: $\beta_0, \beta_1, \ldots, \beta_p$, σ^2, ε_i's
- Random: ε_i's, y_i's
  Constants (not random): β_k's, σ^2, x_{ik}'s

Multiple Linear Regression Models in Matrix Notation

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{Y_{n\times 1}} = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{p1} \\ 1 & x_{12} & x_{22} & \cdots & x_{p2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{pn} \end{bmatrix}}_{X_{n\times(p+1)}} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}}_{\beta_{(p+1)\times 1}} + \underbrace{\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}}_{\varepsilon_{n\times 1}}$$

or

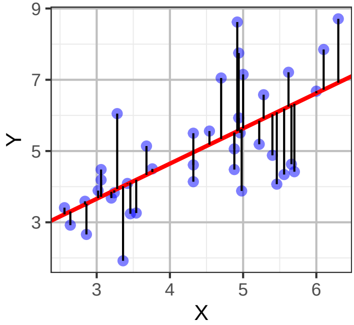
$$Y = X\beta + \varepsilon$$

Least Squares Estimation

Fitting the Model — Least Squares Method

Recall for SLR, the least squares estimate $(\widehat{\beta}_0, \widehat{\beta}_1)$ for (β_0, β_1) is the intercept and slope of the straight line with the minimum sum of squared vertical distances to the data points

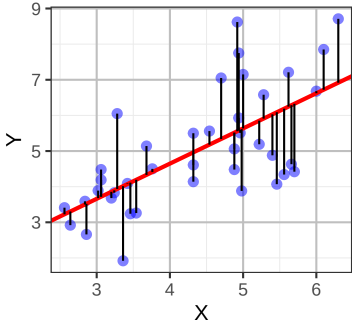
$$\sum_{i=1}^{n}(y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i)^2.$$



Fitting the Model — Least Squares Method

Recall for SLR, the least squares estimate $(\widehat{\beta_0}, \widehat{\beta_1})$ for (β_0, β_1) is the intercept and slope of the straight line with the minimum sum of squared vertical distances to the data points

$$\sum_{i=1}^{n}(y_i - \widehat{\beta_0} - \widehat{\beta_1}x_i)^2.$$



MLR is just like SLR. The least squares estimate $(\widehat{\beta_0}, \ldots, \widehat{\beta_p})$ for $(\beta_0, \ldots, \beta_p)$ is the intercept and slopes of the (hyper)plane with the minimum sum of squared vertical distance to the data points

$$\sum_{i=1}^{n}(y_i - \widehat{\beta_0} - \widehat{\beta_1}x_{i1} - \ldots - \widehat{\beta_p}x_{ip})^2$$

The “Hat” Notation

From now on, we use the “hat” notation to differentiate

- the estimated coefficient $\widehat{\beta}_j$ from
- the actual unknown coefficient β_j

Least Squares Problem for SLR

To find the $(\widehat{\beta}_0, \widehat{\beta}_1)$ that minimize

$$L(\widehat{\beta}_0, \widehat{\beta}_1) = \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i)^2$$

one can set the derivatives of L with respect to $\widehat{\beta}_0$ and $\widehat{\beta}_1$ to 0

$$\frac{\partial L}{\partial \widehat{\beta}_0} = -2 \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i) = 0$$

$$\frac{\partial L}{\partial \widehat{\beta}_1} = -2 \sum_{i=1}^{n} x_i (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i) = 0$$

This results in the 2 equations below in 2 unknowns $\widehat{\beta}_0$ and $\widehat{\beta}_1$.

$$n\widehat{\beta}_0 + \widehat{\beta}_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$\widehat{\beta}_0 \sum_{i=1}^{n} x_i + \widehat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

Least Squares Problem for SLR

$$n\widehat{\beta}_0 + \widehat{\beta}_1 \sum\nolimits_{i=1}^{n} x_i = \sum\nolimits_{i=1}^{n} y_i$$

$$\widehat{\beta}_0 \sum\nolimits_{i=1}^{n} x_i + \widehat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

Least Squares Problem for SLR

$$n\widehat{\beta}_0 + \widehat{\beta}_1 \overbrace{\sum_{i=1}^{n} x_i}^{=n\bar{x}} = \overbrace{\sum_{i=1}^{n} y_i}^{=n\bar{y}}$$

$$\widehat{\beta}_0 \underbrace{\sum_{i=1}^{n} x_i}_{=n\bar{x}} + \widehat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

Least Squares Problem for SLR

$$n\widehat{\beta}_0 + \widehat{\beta}_1 \overbrace{\sum_{i=1}^n x_i}^{=n\bar{x}} = \overbrace{\sum_{i=1}^n y_i}^{=n\bar{y}} \xrightarrow{\text{divide by } n} \widehat{\beta}_0 + \widehat{\beta}_1 \bar{x} = \bar{y} \Rightarrow \widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}$$

$$\widehat{\beta}_0 \underbrace{\sum_{i=1}^n x_i}_{=n\bar{x}} + \widehat{\beta}_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

Least Squares Problem for SLR

$$n\widehat{\beta}_0 + \widehat{\beta}_1 \overbrace{\sum_{i=1}^{n} x_i}^{=n\bar{x}} = \overbrace{\sum_{i=1}^{n} y_i}^{=n\bar{y}} \overset{\text{divide by } n}{\Longrightarrow} \widehat{\beta}_0 + \widehat{\beta}_1 \bar{x} = \bar{y} \Rightarrow \widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}$$

$$\widehat{\beta}_0 \underbrace{\sum_{i=1}^{n} x_i}_{=n\bar{x}} + \widehat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i \Longrightarrow \widehat{\beta}_0 n\bar{x} + \widehat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

Least Squares Problem for SLR

$$n\widehat{\beta}_0 + \widehat{\beta}_1 \overbrace{\sum_{i=1}^n x_i}^{=n\bar{x}} = \overbrace{\sum_{i=1}^n y_i}^{=n\bar{y}} \overset{\text{divide by } n}{\Longrightarrow} \widehat{\beta}_0 + \widehat{\beta}_1 \bar{x} = \bar{y} \Rightarrow \widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}$$

$$\widehat{\beta}_0 \underbrace{\sum_{i=1}^n x_i}_{=n\bar{x}} + \widehat{\beta}_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i \Longrightarrow \widehat{\beta}_0 n\bar{x} + \widehat{\beta}_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

Replacing $\widehat{\beta}_0$ with $\bar{y} - \widehat{\beta}_1 \bar{x}$ in the second equation, we get

$$(\bar{y} - \widehat{\beta}_1 \bar{x}) n\bar{x} + \widehat{\beta}_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

$$\Longleftrightarrow \widehat{\beta}_1 \left(\sum_{i=1}^n x_i^2 - n\bar{x}^2 \right) = \sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}$$

$$\Longleftrightarrow \widehat{\beta}_1 = \frac{\sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^n x_i^2 - n\bar{x}^2}$$

HW

- Show that

$$\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^{n}(x_i - \bar{x})y_i = \left(\sum_{i=1}^{n} x_i y_i\right) - n\bar{x}\bar{y}.$$

- Show that

$$\sum_{i=1}^{n}(x_i - \bar{x})^2 = \left(\sum_{i=1}^{n} x_i^2\right) - n\bar{x}^2.$$

Hence, there are 3 formulae for LS estimate of the slope:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})y_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Least Squares Problem for MLR

To find the $(\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_p)$ that minimize

$$L(\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_p) = \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_{i1} - \ldots - \widehat{\beta}_p x_{ip})^2$$

one can set the derivatives of L with respect to $\widehat{\beta}_j$ to 0

$$\frac{\partial L}{\partial \widehat{\beta}_0} = -2 \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_{i1} - \ldots - \widehat{\beta}_p x_{ip})$$

$$\frac{\partial L}{\partial \widehat{\beta}_k} = -2 \sum_{i=1}^{n} x_{ik}(y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_{i1} - \ldots - \widehat{\beta}_p x_{ip}), \; k = 1, 2, \ldots, p$$

and then equate them to 0. This results in a system of $(p+1)$ equations in $(p+1)$ unknowns on the next page.

Least Squares Problem for MLR

The least squares estimate $(\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_p)$ is the solution to the following system of equations, called the **normal equations**.

$$
\begin{array}{lllll}
\widehat{\beta}_0 \;\cdot\; n & +\widehat{\beta}_1 \sum_{i=1}^n x_{i1} & +\cdots+\widehat{\beta}_p \sum_{i=1}^n x_{ip} & = \sum_{i=1}^n y_i \\
\widehat{\beta}_0 \sum_{i=1}^n x_{i1} & +\widehat{\beta}_1 \sum_{i=1}^n x_{i1}^2 & +\cdots+\widehat{\beta}_p \sum_{i=1}^n x_{i1}x_{ip} & = \sum_{i=1}^n x_{i1}y_i \\
 & & \vdots & \\
\widehat{\beta}_0 \sum_{i=1}^n x_{ik} & +\widehat{\beta}_1 \sum_{i=1}^n x_{ik}x_{i1} & +\cdots+\widehat{\beta}_p \sum_{i=1}^n x_{ik}x_{ip} & = \sum_{i=1}^n x_{ik}y_i \\
 & & \vdots & \\
\widehat{\beta}_0 \sum_{i=1}^n x_{ip} & +\widehat{\beta}_1 \sum_{i=1}^n x_{ip}x_{i1} & +\cdots+\widehat{\beta}_p \sum_{i=1}^n x_{ip}^2 & = \sum_{i=1}^n x_{ip}y_i
\end{array}
$$

Least Squares Problem for MLR

The least squares estimate $(\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_p)$ is the solution to the following system of equations, called the **normal equations**.

$$
\begin{array}{llll}
\widehat{\beta}_0 \cdot n & + \widehat{\beta}_1 \sum_{i=1}^n x_{i1} & + \cdots + \widehat{\beta}_p \sum_{i=1}^n x_{ip} & = \sum_{i=1}^n y_i \\
\widehat{\beta}_0 \sum_{i=1}^n x_{i1} & + \widehat{\beta}_1 \sum_{i=1}^n x_{i1}^2 & + \cdots + \widehat{\beta}_p \sum_{i=1}^n x_{i1}x_{ip} & = \sum_{i=1}^n x_{i1}y_i \\
 & & \vdots & \\
\widehat{\beta}_0 \sum_{i=1}^n x_{ik} & + \widehat{\beta}_1 \sum_{i=1}^n x_{ik}x_{i1} & + \cdots + \widehat{\beta}_p \sum_{i=1}^n x_{ik}x_{ip} & = \sum_{i=1}^n x_{ik}y_i \\
 & & \vdots & \\
\widehat{\beta}_0 \underbrace{\sum_{i=1}^n x_{ip}}_{\text{known}} & + \widehat{\beta}_1 \underbrace{\sum_{i=1}^n x_{ip}x_{i1}}_{\text{known}} & + \cdots + \widehat{\beta}_p \underbrace{\sum_{i=1}^n x_{ip}^2}_{\text{known}} & = \underbrace{\sum_{i=1}^n x_{ip}y_i}_{\text{known}}
\end{array}
$$

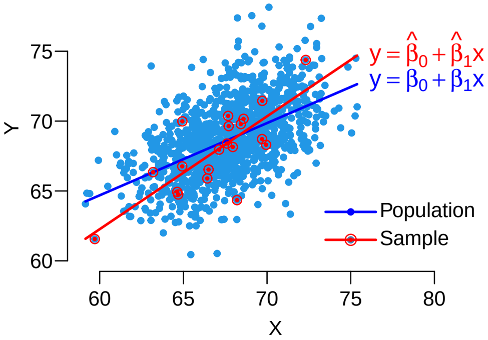
- In matrix notation, the normal equation is $(X^T X)\widehat{\beta} = X^T Y$, and the least squares estimate is $\widehat{\beta} = (X^T X)^{-1} X^T Y$
- Don't worry about solving the equations.
  R and other software can do the computation for us.

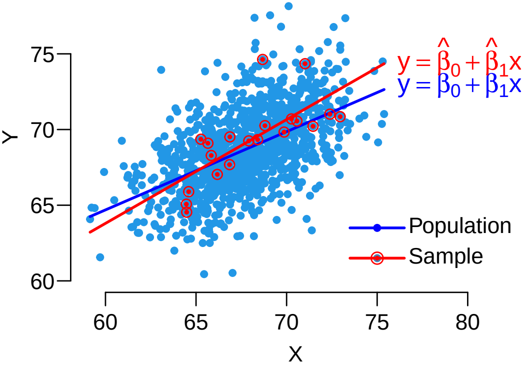
Parameters v.s. Estimates

Note β_i's are the coefficients of the MLR model, and $\widehat{\beta}_i$'s are the estimates of β_i's.

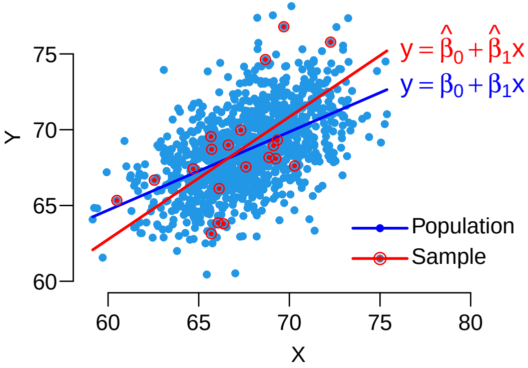
For SLR mdodel,

- $y = \beta_0 + \beta_1 x$ is the least square line for the **population**.
- $y = \widehat{\beta}_0 + \widehat{\beta}_1 x$ is the least square line for a **sample**

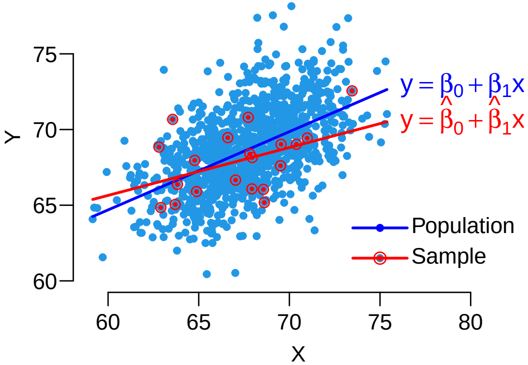




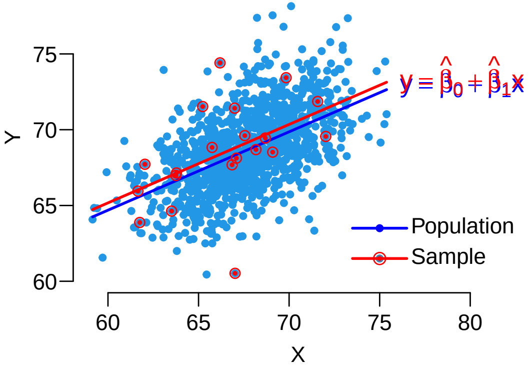
$y = \beta_0 + \beta_1 x$	$y = \widehat{\beta}_0 + \widehat{\beta}_1 x$
least-square regression line of the population	least-square regression line of the sample
fixed	random, changes from sample to sample
unknown	can be calculated from sample
of interest	not of interest



$y = \beta_0 + \beta_1 x$	$y = \widehat{\beta_0} + \widehat{\beta_1} x$
least-square regression line of the population	least-square regression line of the sample
fixed	random, changes from sample to sample
unknown	can be calculated from sample
of interest	not of interest



$y = \beta_0 + \beta_1 x$	$y = \widehat{\beta_0} + \widehat{\beta_1} x$
least-square regression line of the population	least-square regression line of the sample
fixed	random, changes from sample to sample
unknown	can be calculated from sample
of interest	not of interest



$y = \beta_0 + \beta_1 x$	$y = \widehat{\beta_0} + \widehat{\beta_1} x$
least-square regression line of the population	least-square regression line of the sample
fixed	random, changes from sample to sample
unknown	can be calculated from sample
of interest	not of interest

Fitted Values, Residuals, Estimate of σ^2

Fitted Values

The fitted value or predicted value:

$$\widehat{y_i} = \widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \ldots + \widehat{\beta}_p x_{ip}$$

Again, the "*hat*" notation is used.

- $\widehat{y_i}$ is the fitted value
- y_i is the actual observed value

Errors and Residuals

- One cannot directly compute the errors

$$\varepsilon_i = y_i - \beta_0 - \beta_1 x_{i1} - \ldots - \beta_p x_{ip}$$

  since the coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are **unknown**.

- The errors ε_i can be estimated by the residuals e_i defined as:

$$\begin{aligned} \text{residual } e_i &= \text{observed } y_i - \text{predicted } y_i \\ &= y_i - \widehat{y_i} \\ &= y_i - \underbrace{(\widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \ldots + \widehat{\beta}_p x_{ip})}_{\text{predicted } y_i} \end{aligned}$$

- $e_i \neq \varepsilon_i$ in general since $\widehat{\beta}_j \neq \beta_j$

Properties of Residuals

Recall the LS estimate $(\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_p)$ satisfies the equations

$$\sum_{i=1}^{n} \underbrace{(y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_{i1} - \ldots - \widehat{\beta}_p x_{ip})}_{= y_i - \widehat{y}_i = e_i = \text{residual}} = 0 \text{ and}$$

$$\sum_{i=1}^{n} x_{ik} \overbrace{(y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_{i1} - \ldots - \widehat{\beta}_p x_{ip})} = 0, \; k = 1, 2, \ldots, p.$$

The residuals e_i hence have the properties

$$\underbrace{\sum_{i=1}^{n} e_i = 0}_{\text{Residuals add up to 0.}} \quad , \quad \underbrace{\sum_{i=1}^{n} x_{ik} e_i = 0, \; k = 1, 2, \ldots, p.}_{\text{Residuals are orthogonal to predictors.}}$$

The two properties combined imply that **the residuals have 0 correlation with each of the p predictors** since

$$\mathrm{Cov}(X_k, e) = \frac{1}{n-1}\left(\underbrace{\sum_{i=1}^{n} x_{ik} e_i}_{=0} - n\bar{x}_k \underbrace{\bar{e}}_{=0} \right) = 0$$

Mean Square Error (MSE) — Estimate of σ^2

The variance σ^2 of the errors ε_i's is estimated by the **mean square error (MSE)**, the sum of squares of residuals divided by $n - p - 1$.

$$\text{MSE} = \frac{\sum_{i=1}^{n} e_i^2}{n - p - 1} = \frac{\sum_{i=1}^{n} (y_i - \widehat{y_i})^2}{n - p - 1}$$

Why divided by $n - p - 1$ instead of by n?

- A simple reason is it takes at least $p + 1$ observations to estimate $\beta_0, \beta_1, \ldots, \beta_p$. Need at least $p + 2$ observations to get non-zero residuals to determine the variability of the estimate

Mean Square Error (MSE) — Estimate of σ^2

The variance σ^2 of the errors ε_i's is estimated by the **mean square error (MSE)**, the sum of squares of residuals divided by $n - p - 1$.

$$\text{MSE} = \frac{\sum_{i=1}^{n} e_i^2}{n-p-1} = \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{n-p-1}$$

Why divided by $n - p - 1$ instead of by n?

- A simple reason is it takes at least $p + 1$ observations to estimate $\beta_0, \beta_1, \ldots, \beta_p$. Need at least $p + 2$ observations to get non-zero residuals to determine the variability of the estimate
- We will show (in the next Lecture) that *MSE is an unbiased estimator for σ^2*.

Example: The Auto Data

`Auto` data of 9 variables about 392 car models in the 1980s. The variables include

- `acceleration`: Time to accelerate from 0 to 60 mph (in seconds)
- `horsepower`: Engine horsepower
- `weight`: Vehicle weight (lbs.)

Description of all 9 variables: https://rdrr.io/cran/ISLR/man/Auto.html

You can download the data at

https://www.stat.uchicago.edu/~yibi/s224/data/Auto.txt

Please **change the working directory** to the folder where `Auto.txt` is stored, and load the data as follows.

```
Auto = read.table("Auto.txt", h=T)
```

How to Do Regression in R?

```
lm(acceleration ~ weight + horsepower, data=Auto)

Call:
lm(formula = acceleration ~ weight + horsepower, data = Auto)

Coefficients:
(Intercept)       weight   horsepower
    18.4358       0.0023      -0.0933
```

The `lm()` command above asks R to fit the model

$$\text{acceleration} = \beta_0 + \beta_1 \text{weight} + \beta_2 \text{horsepower} + \varepsilon$$

and R gives us the regression equation

$$\widehat{\text{acceleration}} = 18.4358 + 0.0023\,\text{weight} - 0.0933\,\text{horsepower}$$

More R Commands

```
lm1 = lm(acceleration ~ weight + horsepower, data=Auto)
lm1$coef          # show the estimated beta's
(Intercept)      weight  horsepower
  18.435791    0.002302   -0.093313
```

More R Commands

```
lm1 = lm(acceleration ~ weight + horsepower, data=Auto)
lm1$coef          # show the estimated beta's
(Intercept)      weight  horsepower
  18.435791    0.002302   -0.093313
```

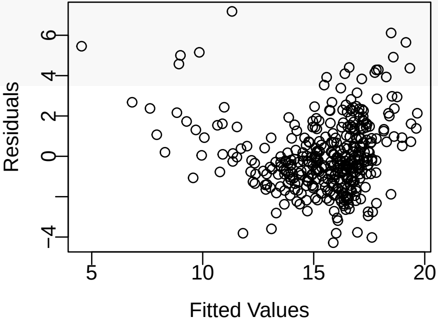
```
lm1$fit           # show the fitted values
lm1$res           # show the residuals
```

More R Commands

```
lm1 = lm(acceleration ~ weight + horsepower, data=Auto)
lm1$coef          # show the estimated beta's
(Intercept)       weight  horsepower
  18.435791     0.002302   -0.093313
```

```
lm1$fit           # show the fitted values
lm1$res           # show the residuals
```

```
plot(lm1$fit,lm1$res,
     xlab="Fitted Values",
     ylab="Residuals")
```



Interpretation of Regression Coefficients

Interpretation of the Intercept β_0

β_0 = intercept = the mean value of Y when all X_j' are 0.

- may have no practical meaning
  e.g., β_0 is meaningless in the `Auto` model as no car has 0 weight

Interpretation of the regression coefficient for β_j

β_j = the regression coefficient for X_j, is the mean change in the response Y when X_j is increased by one unit **holding other X_i's constant**.

- Also called the **partial regression coefficients** because they are *adjusted for the other covariates*
- Interpretation of β_j depends on the presence of other predictors in the model
  e.g., the 2 β_1's in the 2 models below have different interpretations

$$\text{Model } 1: \quad Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

$$\text{Model } 2: \quad Y = \beta_0 + \beta_1 X_1 + \varepsilon.$$

Something Wrong?

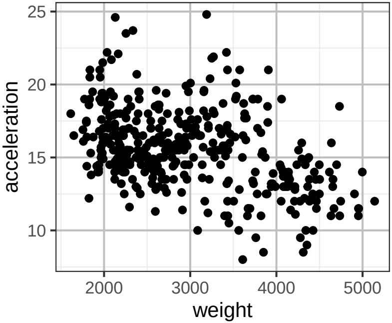
```
# Model 1
lm(acceleration ~ weight, data=Auto)$coef
(Intercept)      weight
  19.572666   -0.001354
# Model 2
lm(acceleration ~ weight + horsepower, data=Auto)$coef
(Intercept)      weight  horsepower
  18.435791    0.002302   -0.093313
```

The coefficient $\widehat{\beta}_1$ for weight is *negative* in the Model 1 but *positive* in the Model 2.

Do heavier cars require more or less time to accelerate from 0 to 60 mph?

Effect of weight Not Controlling for Other Predictors

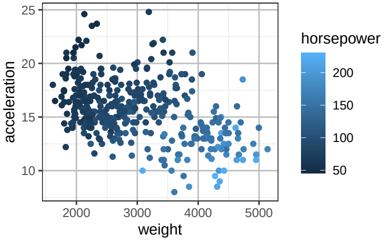
```
library(ggplot2)
ggplot(Auto, aes(x=weight, y=acceleration)) + geom_point()
```

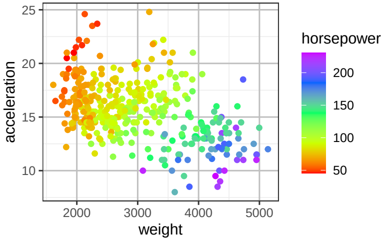


From the scatter plot above, are `weight` and `acceleration` are positively or negatively associated? Do heavier vehicles generally require more or less time to accelerate from 0 to 60 mph? Is that reasonable?

Effect of weight Controlling for horsepower (1)

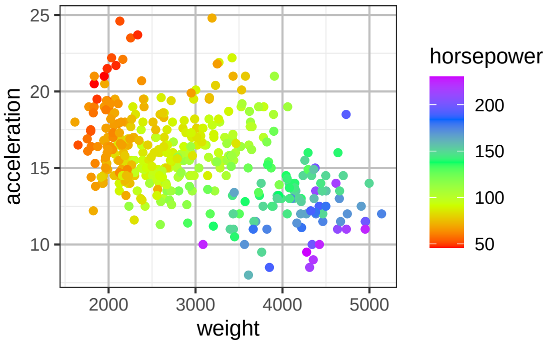
```r
ggplot(Auto, aes(x=weight, y=acceleration, col=horsepower)) +
  geom_point()
ggplot(Auto, aes(x=weight, y=acceleration, col=horsepower)) +
  geom_point() + scale_color_gradientn(colours = rainbow(5))
```





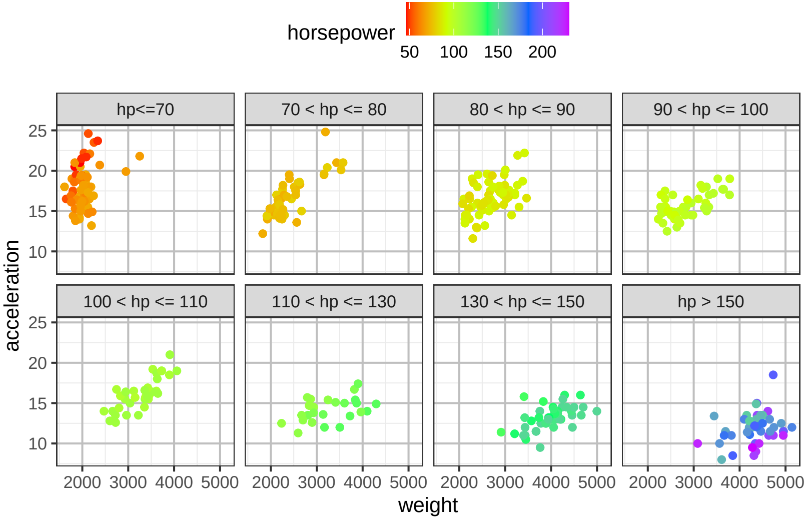
Effect of weight Controlling for horsepower (2)

```
ggplot(Auto, aes(x=weight, y=acceleration, col=horsepower)) +
  geom_point() + scale_color_gradientn(colours = rainbow(5))
```



Consider car models of similar `horsepower` (similar color), are `weight` and `acceleration` positively or negatively correlated?

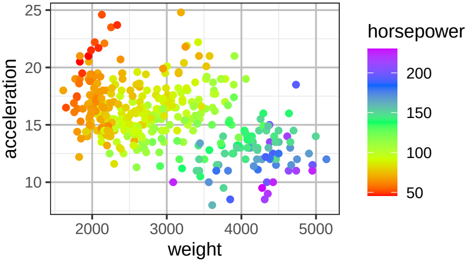
Effect of `weight` Controlling for `horsepower` (3)



R codes for the plot on the previous page

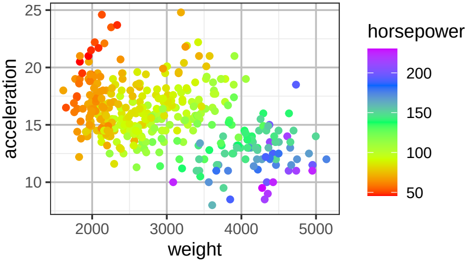
```
Auto$hp = cut(Auto$horsepower,
              breaks=c(45,70, 80, 90,100,110, 130, 150,230),
              labels=c("hp<=70", "70 < hp <= 80", "80 < hp <= 90",
                       "90 < hp <= 100", "100 < hp <= 110",
                       "110 < hp <= 130",
                       "130 < hp <= 150", "hp > 150"))
ggplot(Auto, aes(x=weight, y=acceleration, col=horsepower)) +
  geom_point() + scale_color_gradientn(colours = rainbow(5)) +
  facet_wrap(~hp, nrow=2) + theme(legend.position="top")
```

Example: Auto Data — Simpson's Paradox



Why is the association btw `acceleration` and `weight` flipped from positive to negative when `horsepower` is ignored?

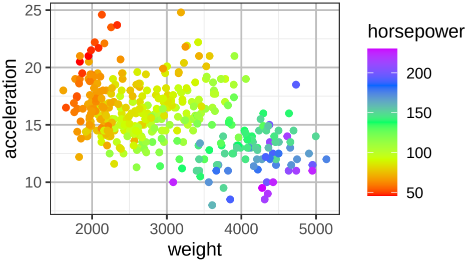
Example: Auto Data — Simpson's Paradox



Why is the association btw `acceleration` and `weight` flipped from positive to negative when `horsepower` is ignored?

- Heavier vehicles (purple dots) tend to have more horsepower while lighter ones (red dots) tend to have less

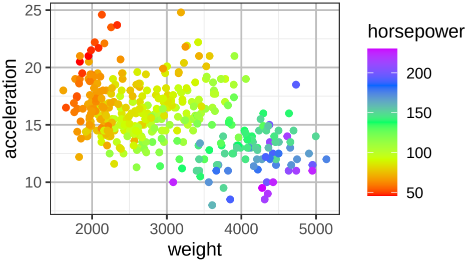
Example: Auto Data — Simpson's Paradox



Why is the association btw `acceleration` and `weight` flipped from positive to negative when `horsepower` is ignored?

- Heavier vehicles (purple dots) tend to have more horsepower while lighter ones (red dots) tend to have less
- Vehicles with more horsepower (purple dots) require less time to accelerate while those with less (red dots) require more

Example: Auto Data — Simpson's Paradox



Why is the association btw `acceleration` and `weight` flipped from positive to negative when `horsepower` is ignored?

- Heavier vehicles (purple dots) tend to have more horsepower while lighter ones (red dots) tend to have less
- Vehicles with more horsepower (purple dots) require less time to accelerate while those with less (red dots) require more
- Hence, when ignoring `horsepower`, it looks like heavier vehicles require less time to accelerate, though heavier vehicles require more time to accelerate after the effect of `horsepower` is adjusted (which means considering only vehicles with similar `horsepower`)

What We Mean by “Adjusted for Other Coveriates”?

For a multiple linear regression model with p predictors

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

β_j represents the effect of X_j on the respone variable Y after it has been **adjusted** for all of $X_1, \ldots, X_p$ except X_j.

What does “adjusted for” mean?

What We Mean by “Adjusted for Other Coveriates” (2)?

The LS estimate $\widehat{\beta}_j$ for β_j in the MLR model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

would be identical to the slope for the SLR model computed as follows.

1. Regress Y on all other X_k's except X_j
2. Regress X_j on all other X_k's except X_j
3. Fit a SLR model using the residuals from Step 1 as the response and the residuals from Step 2 as the predictor.

What We Mean by “Adjusted for Other Coveriates” (2)?

The LS estimate $\widehat{\beta}_j$ for β_j in the MLR model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

would be identical to the slope for the SLR model computed as follows.

1. Regress Y on all other X_k's except X_j
2. Regress X_j on all other X_k's except X_j
3. Fit a SLR model using the residuals from Step 1 as the response and the residuals from Step 2 as the predictor.

Moreover, the intercept obtained in Step 3 would be 0.

What We Mean by “Adjusted for Other Coveriates” (2)?

The LS estimate $\widehat{\beta}_j$ for β_j in the MLR model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

would be identical to the slope for the SLR model computed as follows.

1. Regress Y on all other X_k's except X_j
2. Regress X_j on all other X_k's except X_j
3. Fit a SLR model using the residuals from Step 1 as the response and the residuals from Step 2 as the predictor.

Moreover, the intercept obtained in Step 3 would be 0.

This proof of this result involves complicated matrix algebra and hence is omitted. We just illustrate with an example.

For the Auto Data, recall we have fit the model

$$\text{acceleration} = \beta_0 + \beta_1 \text{weight} + \beta_2 \text{horsepower} + \varepsilon$$

and obtained the estimate for β_1 to be $\widehat{\beta_2} = 0.0023$.

Step 1. Regress `acceleration` on `horsepower`. Let `RY` be the residuals of this model.

```
RY = lm(acceleration ~ horsepower, data=Auto)$res
```

Step 2. Regress `weight` on `horsepower`. Let `RWT` be the residuals of this model.

```
RWT = lm(weight ~ horsepower, data=Auto)$res
```

Step 3. Regress `RY` on `RWT`.

```
lm(RY ~ RWT)$coef
(Intercept)         RWT
  7.352e-17   2.302e-03
```

Observe that

- the **estimated intercept is exactly 0** (slightly off due to rounding error)
- the estimated coefficient for `RWT` is *exactly same* estimated coefficient for `weight` in the model.

```
lm(acceleration ~ weight + horsepower, data=Auto)$coef
(Intercept)      weight  horsepower
  18.435791    0.002302   -0.093313
```

$$RY = \text{acceleration} - \tilde{\beta}_0 - \tilde{\beta}_1 \text{horsepower}$$
$$= \text{the part of } \texttt{acceleration} \text{ not explained by } \texttt{horsepower}$$

`weight` might be correlated with other predictors in the model.

$$\text{weight} = \check{\beta}_0 + \check{\beta}_1 \text{horsepower} + \text{error}$$

We can break `weight` into 2 components:

- a part that's linear w/ of `horsepower`, and
- the part `RWT` that is uncorrelated with `horsepower`

The first part is useless in predicting `acceleration` since `horsepower` haS been included in the model. Only `RWT` provides the additional information that `horsepower` cannot provide.