STS7102: Linear Models and Extensions

Linear Regression

Qiang Sun, Ph.D. <qiang.sun@mbzuai.ac.ae>

MBZUAI

September 18, 2025

Reading

- Chapter 2, Ding (2025).
- Chapter 3, Ding (2025).

Simple Linear Regression

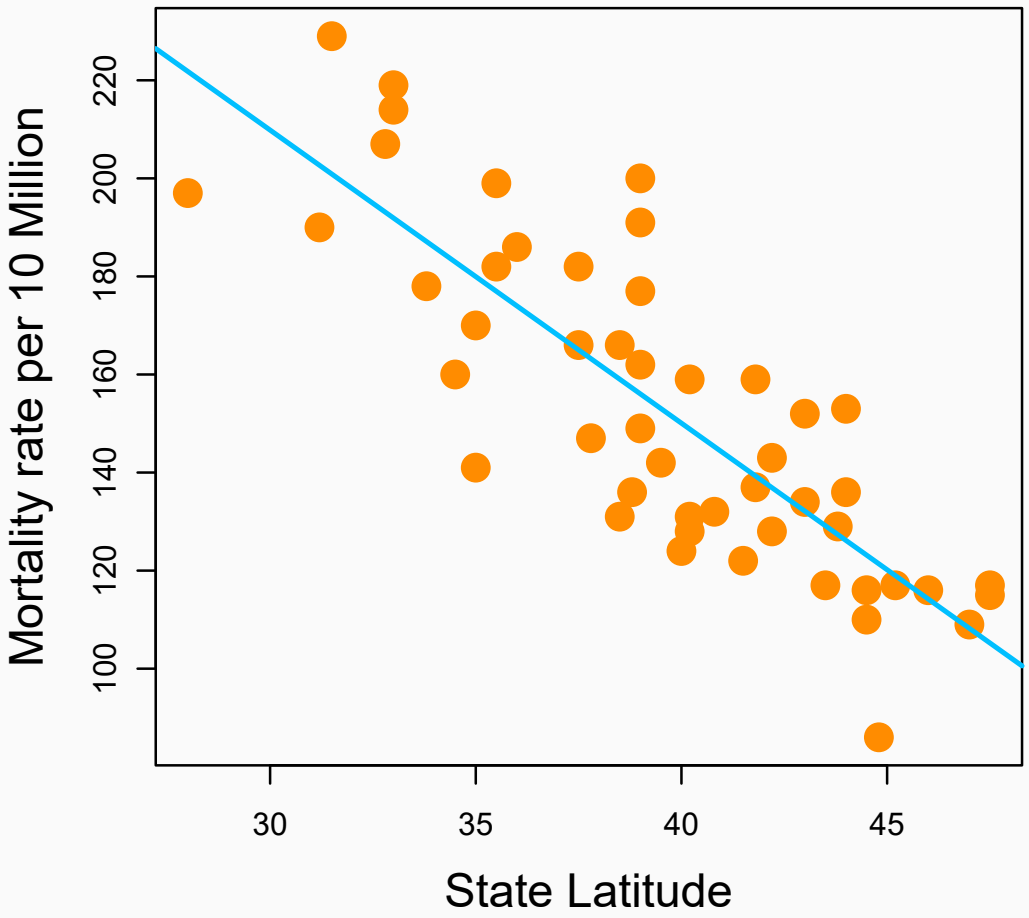
Simple Linear Regression

- Let's look at a simplified version of a linear model — with only one predictor.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

- X is the predictor, Y is the outcome, and ϵ is a random error.
- β_0 and β_1 are unknown regression coefficients that we want to estimate.
- Suppose that a researcher can set the value of X, and perform experiments to observe Y. Repeatedly perform such experiments on different values of X will allow us to collect a set of data.

Simple Linear Regression



Skin cancer mortality rate per 10 million (1950s) by state latitude.

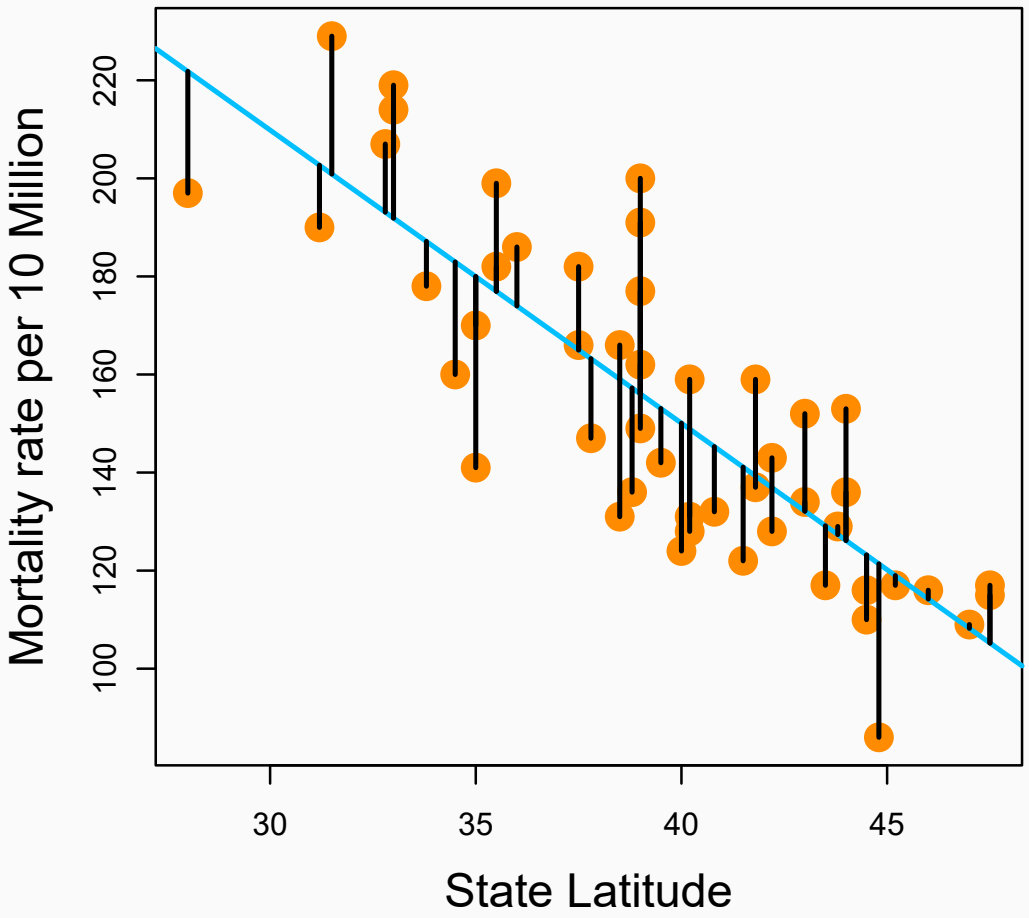
Simple Linear Regression

- What is the optimal line (with intercept $\widehat{\beta}_0$ and slope $\widehat{\beta}_1$) that describes this relationship based on the observed data?
- There are n observations, and for each $i \in 1, \ldots n$, we have
  - y_i the observed mortality rate for state i
  - x_i the latitude for state i
- Usually this is obtained by minimizing the sum of squared errors:

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

- Interpretation: $y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i$ measures the (vertical) distance between the observed point and the fitted line.

Simple Linear Regression



Skin cancer mortality rate per 10 million (1950s) by state latitude.

Minimizing the SSE

- How to minimize the sum of squared errors (SSE)?

$$
\begin{aligned}
(\widehat{\beta}_0, \widehat{\beta}_1) =& \arg\min_{\beta_0, \beta_1} \text{SSE} \\
=& \arg\min_{\beta_0, \beta_1} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2
\end{aligned}
$$

- It is usually believed that the technique was first discovered around 1805 by Adrien Marie Legendre (1752-1833).
- This is a quadratic function of both β_0 and β_1, hence is convex about its argument
- Take the derivative with respect to the parameters and set to zero:

$$
\frac{\partial \text{SSE}}{\partial \beta_0} = 0 \quad \text{and} \quad \frac{\partial \text{SSE}}{\partial \beta_1} = 0
$$

Simple Linear Regression

Legendre (left) and Fourier (right).

Minimizing the SSE

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

$$\frac{\partial \text{SSE}}{\partial \beta_0} = 2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \overset{\text{set}}{=} 0$$

$$\frac{\partial \text{SSE}}{\partial \beta_1} = 2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)x_i \overset{\text{set}}{=} 0$$

$$\implies \quad \widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}$$

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

where r_{xy} is the sample correlation coefficient, and s_y and s_x are the sample standard error.

Example 1

Suppose we observe 8 sample points:

$$\mathbf{x} = (0.7, -0.1, 0.4, 0.3, -2.2, -2.5, -0.4, -1.3)^\mathsf{T},$$
$$\mathbf{y} = (2.2, -1.0, -0.5, 2.8, -2.8, -3.4, 0.1, -2.1)^\mathsf{T}.$$

Find the optimal simple linear regression line that describes the data.

Example 1

Suppose we observe 8 sample points:

$$\mathbf{x} = (0.7, -0.1, 0.4, 0.3, -2.2, -2.5, -0.4, -1.3)^\mathsf{T},$$
$$\mathbf{y} = (2.2, -1.0, -0.5, 2.8, -2.8, -3.4, 0.1, -2.1)^\mathsf{T}.$$

Find the optimal simple linear regression line that describes the data.

- First we calculate $\bar{x} = -0.6375$, $s_x = 1.221167$, $\bar{y} = -0.5875$, and $s_y = 2.235709$.
- Calculate $\widehat{\beta}_1 = r_{xy}\frac{s_y}{s_x} = 1.593462$
- Calculate $\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1\bar{x} = 0.4283319$

Example 1

```python
import numpy as np
np.set_printoptions(precision=6, suppress=True)  # change the default
    number of decimals
import statsmodels.api as sm

# input the data
x = np.array([0.7, -0.1, 0.4, 0.3, -2.2, -2.5, -0.4, -1.3])
y = np.array([2.2, -1.0, -0.5, 2.8, -2.8, -3.4, 0.1, -2.1])
# calculate the means
xbar = np.mean(x)
ybar = np.mean(y)

# calculate beta1
beta1 = np.sum((x - xbar)*(y - ybar)) / np.sum((x - xbar)**2)
print(f"{beta1:.6f}")  # 1.593462
# calculate beta0
beta0 = ybar - beta1 * xbar
print(f"{beta0:.6f}")  # 0.428332
```

Output:

```
1.593462
0.428332
```

Example 1

```
# alternative way to calculate beta1 using correlation coefficient
beta1_alt = np.corrcoef(x, y)[0,1] * np.std(y, ddof=0) / np.std(x, ddof
    =0)
print(f"{beta1_alt:.6f}")  # 1.593462

# add a column of ones for the intercept
X = sm.add_constant(x)

# fit linear regression (equivalent to lm(y~x) in R)
model = sm.OLS(y, X).fit()

# print the estimated coefficients
print(model.params)  # [0.428332, 1.593462]
```

Output:

```
1.593462
0.428332
1.593462
```

Example 2

Suppose a researcher wants to perform a linear regression on the samples he collected. However, the original data was lost, and he has only the access to some summary statistics

$$n = 8, \qquad \bar{x} = 0.2875, \qquad \bar{y} = 0.0075,$$
$$\sum_{i=1}^{n} x_i y_i = 1.5941, \qquad s_x = 0.5460704.$$

Can you still figure out the regression line based on these available information?

Example 2

Suppose a researcher wants to perform a linear regression on the samples he collected. However, the original data was lost, and he has only the access to some summary statistics

$$n = 8, \qquad \bar{x} = 0.2875, \qquad \bar{y} = 0.0075,$$
$$\sum_{i=1}^{n} x_i y_i = 1.5941, \qquad s_x = 0.5460704.$$

Can you still figure out the regression line based on these available information?

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{(n-1)s_x^2} = 0.7554315$$
$$\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x} = -0.2096866$$

Example 2

Suppose a researcher wants to perform a linear regression on the samples he collected. However, the original data was lost, and he has only the access to some summary statistics

$$n = 8, \qquad \bar{x} = 0.2875, \qquad \bar{y} = 0.0075,$$
$$\sum_{i=1}^{n} x_i y_i = 1.5941, \qquad s_x = 0.5460704.$$

Can you still figure out the regression line based on these available information?

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{(n-1)s_x^2} = 0.7554315$$
$$\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x} = -0.2096866$$

To perform linear regressions, the original data is not necessary as long as some key statistics are calculated.

Multiple Linear Regression

Multiple Linear Regression

- Usually a linear regression is perform a number of predictor:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon.$$

- The techniques that we used earlier on simple linear regression can still be applied, but the calculation becomes very tedious.
- We have to setup $p + 1$ equations (taking derivatives of the SSE) and jointly solve for the optimizer.
- We are going to introduce a matrix representation of the solution that makes things easier.
- The distribution of the estimator will also be derived, which makes hypothesis testing possible.

Matrix representation

- The data that we have (from n such experiments) can be summarized into the following matrices:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}_{n\times 1} \quad \text{and} \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}_{n\times(p+1)}$$

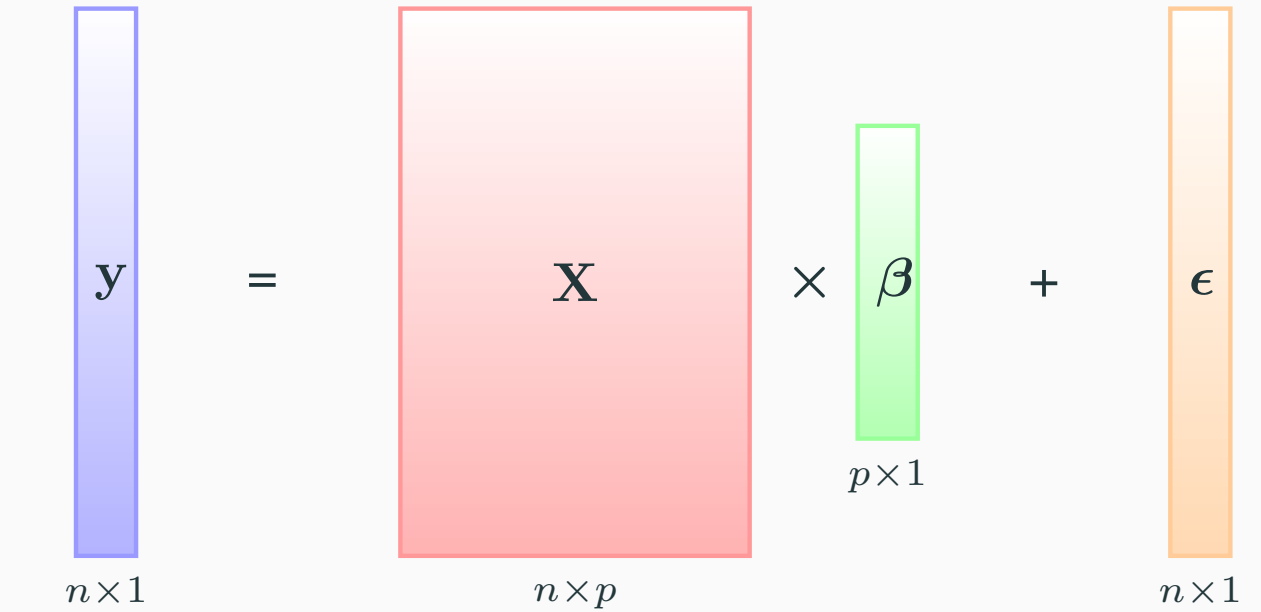
- The parameter vector $\boldsymbol{\beta}$ that we are interested has $p+1$ entries:

$$\boldsymbol{\beta}_{(p+1)\times 1} = (\beta_0, \beta_1, \ldots, \beta_p)^{\mathsf{T}}$$

- The linear regression can be represented as

$$\mathbf{y}_{n\times 1} = \mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1} + \boldsymbol{\epsilon}_{n\times 1}$$

Matrix representation



To clarify some notations

	Random Variable	Realization	Estimation
Outcome	Y	y	$\widehat{y}$, $\bar{y}$
Outcome of n samples	$\mathbf{Y}$	$\mathbf{y}$	$\widehat{\mathbf{y}}$
Predictor	$X, X_1, \ldots, X_p,$	x, x_i, x_{ij}	
Predictor of n samples		$\mathbf{X}$, $\mathbf{x}_j$	
Coefficients			$\widehat{\boldsymbol{\beta}}$
Error	ϵ		
Error of n samples	$\boldsymbol{\epsilon}$		$\mathbf{e}$

Matrix representation

- We can still calculate the sum of squared errors (SSE), based on any proposed $\boldsymbol{\beta}$ estimation

$$\begin{aligned}\text{SSE} &= \sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 = \sum_{i=1}^{n}(y_i - x_i^{\mathsf{T}}\widehat{\boldsymbol{\beta}})^2 \\ &= \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|_2^2,\end{aligned}$$

  where x_i is the ith row of the design matrix $\mathbf{X}$, and $\|\cdot\|_2$ is called the ℓ_2-norm (Euclidean norm):

$$\|\mathbf{a}\|_2 = \sqrt{\textstyle\sum_{i=1}^{n} a_i^2} = \sqrt{\mathbf{a}^{\mathsf{T}}\mathbf{a}}, \quad \text{and} \quad \|\mathbf{a}\|_2^2 = \textstyle\sum_{i=1}^{n} a_i^2 = \mathbf{a}^{\mathsf{T}}\mathbf{a}$$

- We need to minimize the SSE

Matrix representation

- Again, we take derivative of the SSE and obtain a $p+1$ dimensional vector

$$\begin{aligned}\frac{\partial \text{SSE}}{\partial \boldsymbol{\beta}} &= 2\sum_{i=1}^{n} x_i(y_i - x_i^{\mathsf{T}}\widehat{\boldsymbol{\beta}}) \\ &= 2\left(\mathbf{X}^{\mathsf{T}}\mathbf{y} - \mathbf{X}^{\mathsf{T}}\mathbf{X}\widehat{\boldsymbol{\beta}}\right).\end{aligned}$$

- Setting the above to be 0, we have $p+1$ equations represented in the matrix form:

$$\mathbf{X}^{\mathsf{T}}\mathbf{y} = \mathbf{X}^{\mathsf{T}}\mathbf{X}\widehat{\boldsymbol{\beta}},$$

  which is called the normal equations.
- Validate that this is exactly the equations we had for the simple linear regression (p = 1 case). What is the design matrix $\mathbf{X}$?
- How to solve this?

Matrix representation

- In most of the cases $\mathbf{X}^\mathsf{T}\mathbf{X}$ is a positive definite matrix, this means we can multiple $(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$ on both sides of the normal equations and obtain

$$\begin{aligned} & (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{X}\widehat{\boldsymbol{\beta}} \\ \Longrightarrow \quad & (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \widehat{\boldsymbol{\beta}} \end{aligned}$$

  which gives us the solution.
- Why $\mathbf{X}^\mathsf{T}\mathbf{X}$ is usually positive definite? What if it is not? — The column vectors of $\mathbf{X}$ will be linearly dependent. This causes trouble...

Example 3

ID	Intercept	X_1	X_2	Y
1	1	0	1	11
2	1	11	5	15
3	1	11	4	13
4	1	7	3	14
5	1	4	1	0
6	1	10	4	19
7	1	5	4	16
8	1	8	2	8

- Setup the design matrix and response vector
- Perform MLR using solutions to the normal equation.

Example 3

```python
import numpy as np

# set up the design matrix
X1 = np.array([0, 11, 11, 7, 4, 10, 5, 8])
X2 = np.array([1, 5, 4, 3, 1, 4, 4, 2])

# design matrix with intercept
X = np.column_stack((np.ones(len(X1)), X1, X2))

# response vector
y = np.array([11, 15, 13, 14, 0, 19, 16, 8]).reshape(-1, 1)

# closed-form OLS solution: (X'X)^(-1) X'y
beta = np.linalg.inv(X.T @ X) @ (X.T @ y)

print(beta)
```

Output:

```
[[ 3.7]
 [-0.7]
 [ 4.4]]
```

Example 3

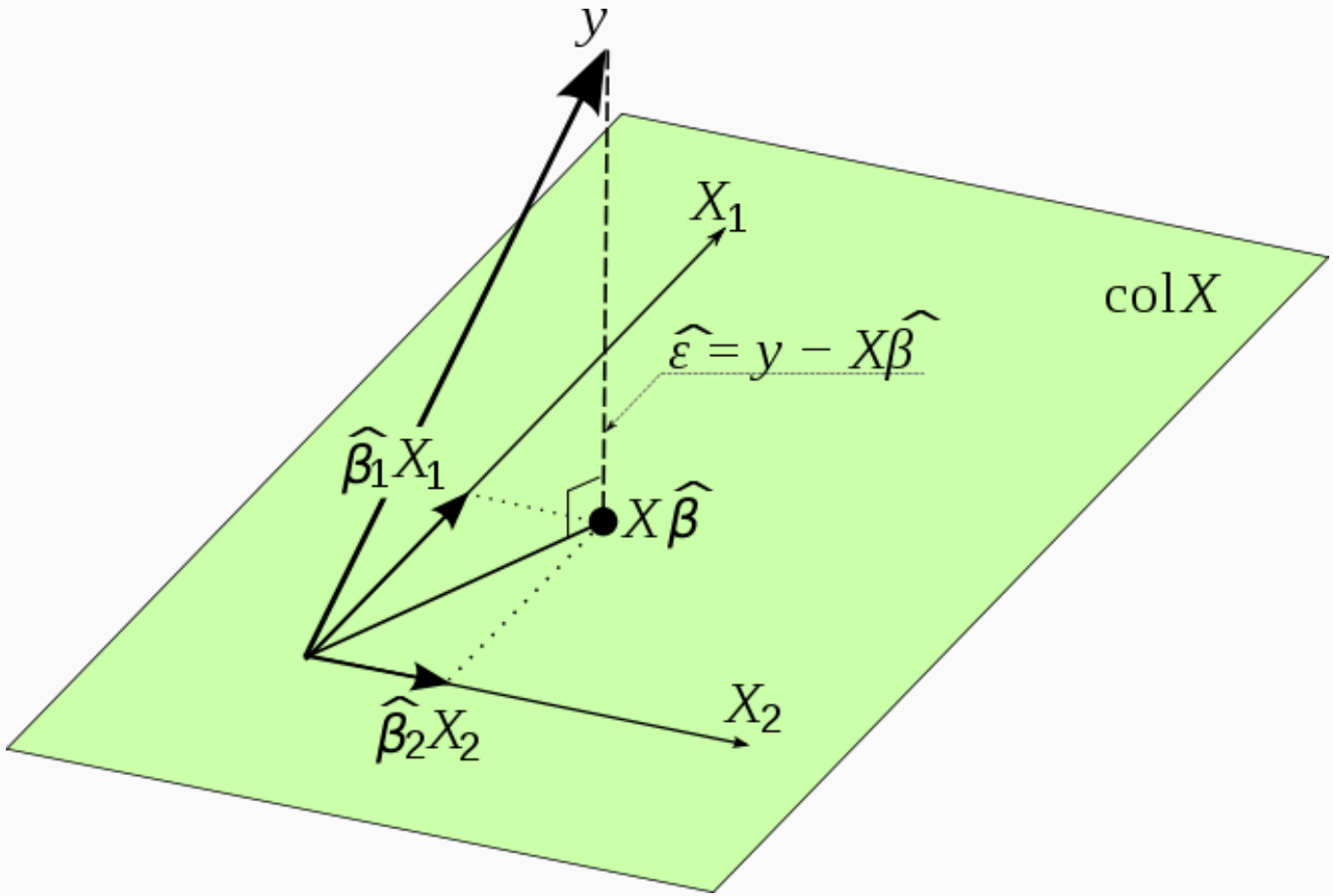
```
import statsmodels.api as sm

# Fit linear regression (like lm() in R)
model = sm.OLS(y, X).fit()

# Print summary coefficients
print(model.params)
```

Output:

```
const    3.7
x1      -0.7
x2       4.4
dtype: float64
```

Geometric interpretation

- Linear regression can be viewed as projecting the vector $\mathbf{y}$ onto a hyperplane defined by the column space of $\mathbf{X}$



Geometric interpretation

- The column vectors of $\mathbf{X}$ are

$$\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{pmatrix}, \quad \begin{pmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{pmatrix}, \quad \cdots \quad \begin{pmatrix} x_{np} \\ x_{np} \\ \vdots \\ x_{np} \end{pmatrix}$$

- Any element in the column space $\text{col}(\mathbf{X})$ of $\mathbf{X}$ can be expressed as their linear combinations:

$$\beta_0 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \beta_1 \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{pmatrix} + \beta_2 \begin{pmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{pmatrix} + \cdots + \beta_p \begin{pmatrix} x_{np} \\ x_{np} \\ \vdots \\ x_{np} \end{pmatrix} = \mathbf{X}\boldsymbol{\beta}$$

Geometric interpretation

- Among all these kind of linear combinations (search through the entire column space of $\mathbf{X}$, namely col$(\mathbf{X})$), find the one closest to $\mathbf{y}$.
- How to define "closest"? — Euclidean distance, the ℓ_2 norm.
- This is the same as projecting the vector $\mathbf{y}$ onto the space col$(\mathbf{X})$ (shown in the previous plot).
- The projection is $\widehat{\mathbf{y}}$, and the remaining part $\mathbf{e} = \mathbf{y} - \widehat{\mathbf{y}}$ will be orthogonal to the space col$(\mathbf{X})$.
- There are some easy ways to calculate this project.

Special case: orthogonal design matrix

- Usually it is difficult to calculate the inverse matrix $(\mathbf{X}^\mathsf{T}\mathbf{X})$, however, there is a special case when $\mathbf{X}^\mathsf{T}\mathbf{X}$ is an diagonal matrix, i.e., only the diagonal elements are non-zero.
- This happens when the columns of $\mathbf{X}$ are orthogonal to each other.
- An example:

Intercept	X_1	X_2	Y
1	1	1	1
1	1	-1	2
1	-1	1	3
1	-1	-1	4

- Calculate the regression coefficients by hand.

Hand calculation of the $\widehat{\beta}$

- We first get $\mathbf{X}^\mathsf{T}\mathbf{X}$, which is a diagonal matrix

$$\mathbf{X}^\mathsf{T}\mathbf{X} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

- The inverse of that is just taking the inverse of each element:

$$(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1} = \begin{pmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{pmatrix}$$

- Multiple that to the $\mathbf{X}^\mathsf{T}\mathbf{y}$, we have

$$(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \begin{pmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{pmatrix} \begin{pmatrix} 10 \\ -4 \\ -2 \end{pmatrix} = \begin{pmatrix} 2.5 \\ -1 \\ -0.5 \end{pmatrix}$$

Geometric interpretation

- Let $\mathbf{H} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}$ be a projection matrix referred to as the "hat" matrix

$$\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{H}\mathbf{y}$$
$$\mathbf{e} = \mathbf{y} - \widehat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}$$

- $\mathbf{H}$ is idempotent: $\mathbf{H}$ is symmetric and $\mathbf{H}\mathbf{H} = \mathbf{H}$

Proof.

$$\begin{aligned}
\mathbf{H}^\mathsf{T} &= \left(\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\right)^\mathsf{T} \\
&= \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T} = \mathbf{H} \\
\mathbf{H}\mathbf{H} &= \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T} \\
&= \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T} = \mathbf{H}
\end{aligned}$$

□

Example

- Load the `cheddar` data, model `taste` using all three covaraites (with intercept): Acetic, H2S and Lactic.
- Calculate the following quantities to perform MLR:
  - $\mathbf{X}^\top\mathbf{X}$, and check if it is positive definite
  - The parameter estimates $\widehat{\boldsymbol{\beta}}$
  - $\mathbf{H}$, calculate SSE and $\widehat{\sigma}^2$, what is the degrees of freedom?
  - The coefficient of determination R^2

Example

- Load the `cheddar` data, model `taste` using all three covaraites (with intercept): Acetic, H2S and Lactic.
- Calculate the following quantities to perform MLR:
  - $\mathbf{X}^\top\mathbf{X}$, and check if it is positive definite
  - The parameter estimates $\widehat{\boldsymbol{\beta}}$
  - $\mathbf{H}$, calculate SSE and $\widehat{\sigma}^2$, what is the degrees of freedom?
  - The coefficient of determination R^2
- If a researcher wants to use at most 2 covariates, which is the best model?

A: Acetic + H2S; B: H2S + Lactic; C: Acetic + Lactic

Example

- Load the `cheddar` data, model `taste` using all three covaraites (with intercept): Acetic, H2S and Lactic.
- Calculate the following quantities to perform MLR:
  - $\mathbf{X}^\top\mathbf{X}$, and check if it is positive definite
  - The parameter estimates $\widehat{\boldsymbol{\beta}}$
  - $\mathbf{H}$, calculate SSE and $\widehat{\sigma}^2$, what is the degrees of freedom?
  - The coefficient of determination R^2
- If a researcher wants to use at most 2 covariates, which is the best model?

A: Acetic + H2S; B: H2S + Lactic; C: Acetic + Lactic

- Question: Will MLR with X_1 and X_2 always outperforms the model using X_1 only?

The sum of squares

- Recall that SST $=$ SSR $+$ SSE

$$\begin{aligned}
\text{SST} &= \|\mathbf{y} - \bar{y}\mathbf{1}\|_2^2 \\
\text{SSE} &= \|(\mathbf{I} - \mathbf{H})\mathbf{y}\|_2^2 = \mathbf{y}^\mathsf{T}(\mathbf{I} - \mathbf{H})^\mathsf{T}(\mathbf{I} - \mathbf{H})\mathbf{y} \\
&= \mathbf{y}^\mathsf{T}(\mathbf{I} - \mathbf{H})\mathbf{y} \\
\text{SSR} &= \|\mathbf{X}\widehat{\boldsymbol{\beta}} - \bar{y}\mathbf{1}\|_2^2
\end{aligned}$$

- $\mathbf{1}$ is a vector of length n, with each element being 1.
- SST resides in $n - 1$ dimensions; SSE in $n - p - 1$ dimensions; SSR in p dimensions.
- Careful: Sometimes people count the intercept as one of the p dimensions, we didn't.