ISOM 2600 – Introduction to Business Analytics

Topic 3: Multiple Linear Regression

Readings: Chapter 15

About this Course

1 — Python Basic
- List
- NumPy
- SciPy
- Plotting

2 — Exploratory Data Analysis
- Pandas
- EDA
- Methods for Time Series

3 — Multiple Linear Regression
- SLR
- MLR

4 — Clustering

Two Key Goals of Regression

<u>Explanation:</u> understand the relationship between a predictor variable X_i and response variable Y

- ❑ How does a change in advertising spend X impact the sales Y

<u>Prediction:</u> use the known values of predictor variable $(X_1, X_2, \dots, X_n)$ to predict the response variable Y

- ❑ Given ad spend X_1, store size X_2 , and foot traffic X_3, predict the next month's sales Y

```python
import statsmodels.api as sm
```

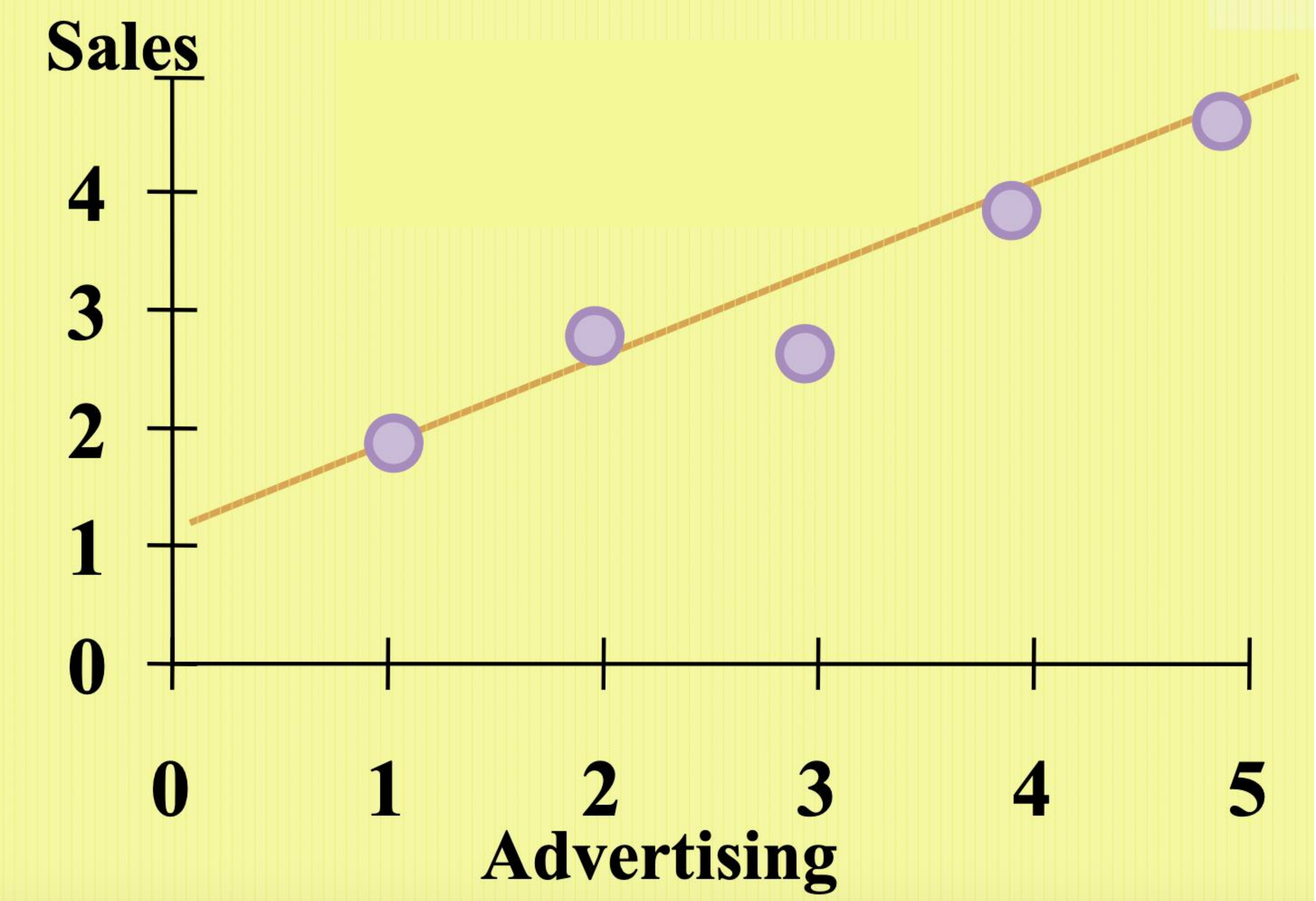
Functions	**Purpose**
model = sm.OLS().fit()	Fit a Linear regression model and return a model object
sm.add_constant()	Add intercept term to model
model.summary()	Display a detailed summary of regression results, coefficient estimation, p-value…
model.predict()	Predict y for a new input x
model.fittedvalues	Return the predicted (fitted) values for the training data
model.resid	Return the residual = actual value – predicted value
model.rsquared	Return the R^2
model.mse_resid	Return the Mean Squared Error

Simple Linear Model (SLM) Review

Example: Ads vs Sales

How to describe the relationship between these two variables?

Ads ($ 1000)	Sales Volume (1000 cups)
1	2
2	3
3	3
4	4
5	5



Example: Ads vs Sales

How to interpret the fitted line? $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x = 1.3 + 0.7x$

Interpretation/Explanation:

- The **slope** $\hat{\beta}_1$: mean sales volume y is expected to increase by 0.7 units for each unit increase in Ads x (or the estimated change in y on average is 0.7 unit associated with one unit increase in Ads x)

- The **intercept** $\hat{\beta}_0$: average value of sales is 1.3 units when Ads $x = 0$

Prediction:

If the Ads $x = 1.5$ ($1500), the prediction of **mean sales** is

$$\hat{y} = 1.3 + 0.7 * 1.5 = 2.35 \text{ (2350 cups)}$$

Example: Ads vs Sales

How accurate is the model? $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x = 1.3 + 0.7x$

Calculate the MSE and RMSE:

x_i	y_i	$\hat{y}_i = 1.3 + 0.7x_i$	$y_i - \hat{y}_i$	$(y_i - \hat{y}_i)^2$
1	2	2	0	0
2	3	2.7	0.3	0.09
3	3	3.4	-0.4	0.16
4	4	4.1	-0.1	0.01
5	5	4.8	0.2	0.04
Total				$SSE = 0.3$

MSE

$$s_e^2 = \frac{SSE}{n-2}$$

$$= \frac{0.3}{5-2} = 0.1$$

RMSE

$$s_e = \sqrt{0.1} = 0.3162$$

Review of the definitions in later part

Example: Ads vs Sales

How useful is the model? $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x = 1.3 + 0.7x$

Calculate the R^2: 94.2% of the variation in sales volume is explained by the the regression model on Ads spend

x_i	y_i	$\hat{y}_i = 1.3 + 0.7x_i$	$(y_i - \hat{y}_i)^2$	$(y_i - \bar{y})^2$
1	2	2	0	1.96
2	3	2.7	0.09	0.16
3	3	3.4	0.16	0.16
4	4	4.1	0.01	0.36
5	5	4.8	0.04	2.56
Total			$SSE = 0.3$	$SST = 5.2$

$$R^2 = 1 - \frac{SSE}{SST} = 0.942$$

Modeling Steps

1. Specify model structure: response variable and independent variable(s)

2. Check the model assumptions **(LINE Assumptions)**

3. Slice Data: Train and Test Datasets

4. Build Models with **Library: statsmodels**

5. Evaluate the models: R-squared, MSE and RMSE, Hypothesis Testing

6. Use model for prediction and estimation

Model Structure

General Form: relationship between the response variable Y and the explanatory variable X is a **linear function**

Population y-intercept

Population Slope

Random Error

$$Y = \underbrace{\beta_0 + \beta_1 X}_{\text{Deterministic Part: Linear Form}} + \varepsilon$$

Response Variable

Explanatory Variable

Data and Notation

$$y = \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \qquad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon_n \end{bmatrix}$$

Unobservable Part

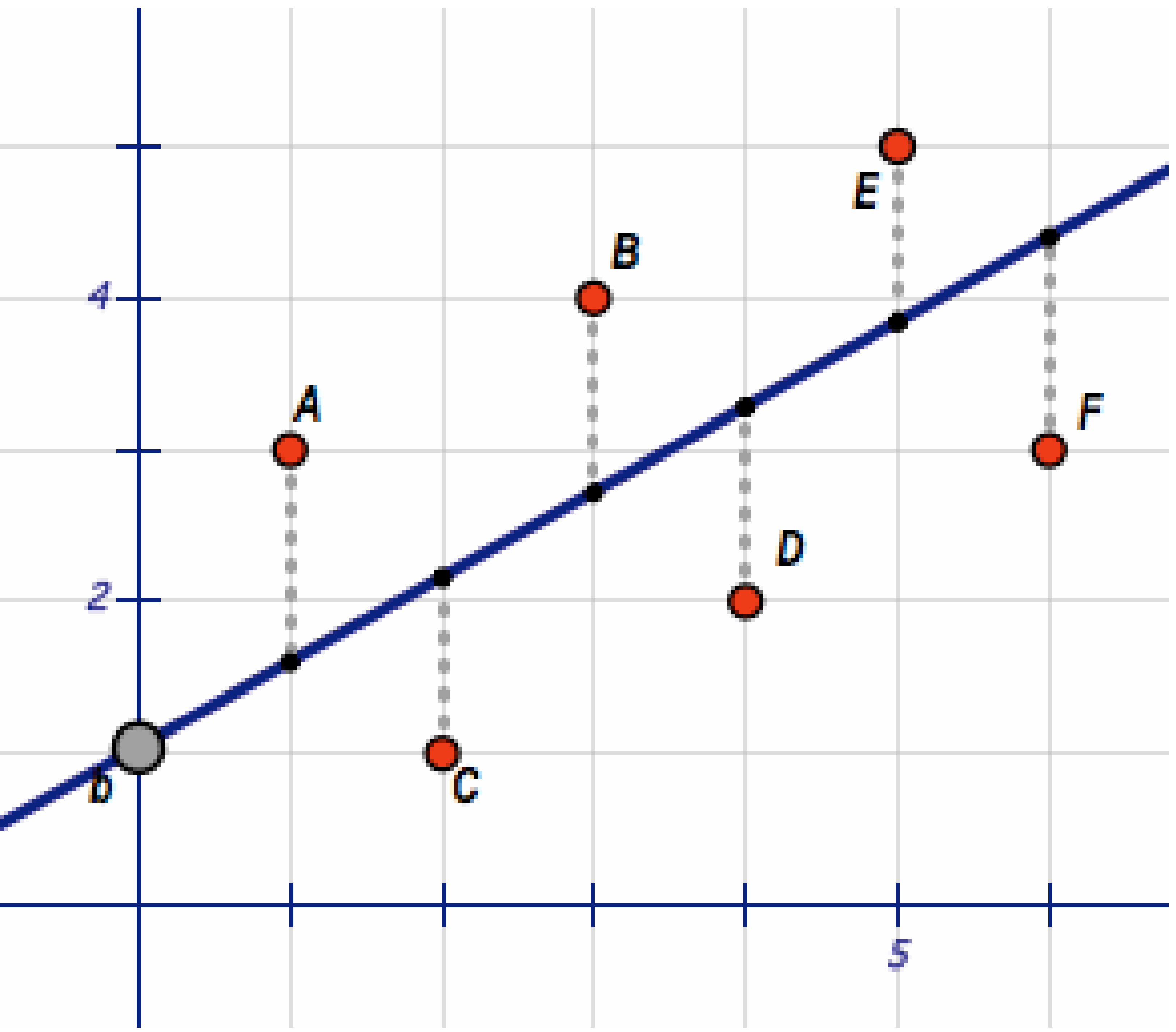
“LINE” Model Assumptions

$\varepsilon_i \sim N(0, \sigma^2)$ for $i = 1, \dots, n$ and $\varepsilon_1, \dots, \varepsilon_n$ are independent and identical distributed (iid)

- **L**inearity Assumption: ε_i have mean equal to zero $E(\varepsilon_i) = 0$
- **I**ndependence Assumption: ε_i are independent of each other
- **N**ormality Assumption: ε_i are normally distributed
- **E**qual Variance: ε_i have equal variance σ_ε^2

How to check whether the assumptions hold?
Residual Analysis

Least Square Estimation



Find the "Best-fitting" line:

- "Best-fitting" means minimizing the **vertical difference/residual** between actual y values and predicted values $\hat{y}$
- The residual is the **error of prediction**: $e_i = y_i - \hat{y}_i$
- Define Sum of Squared Errors **(SSE):**
  - SSE = $\sum e_i^2 = \sum (y_i - \hat{y}_i)^2$
- **Least Square Estimation (LSE):** choose the estimators $\hat{\beta}_0$,$\hat{\beta}_1$ such that SSE is minimized

LSE Formulas

Mathematically, to calculate the estimates of the coefficients/parameters, we use the following formulas:

$$\hat{\beta}_1 = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})/(n-1)}{\sum_i (x_i - \overline{x})^2/(n-1)}$$

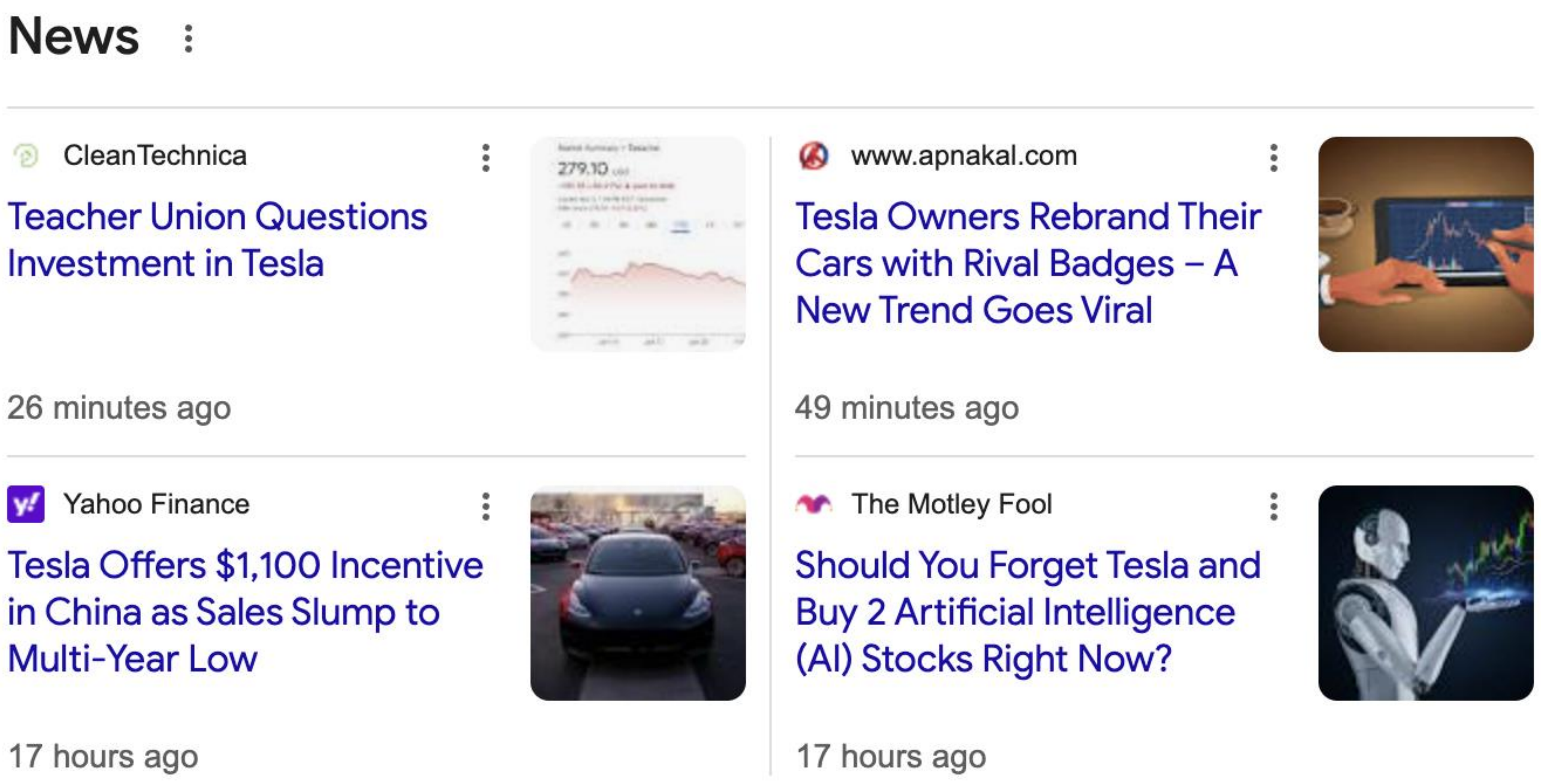
$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

The **regression line** or **fitted line** that estimates the equation of the simple linear model is

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

Case: S&P 500 and Tesla

We want to analyze how **TSLA** relate to the **market** or whether it is aggressive or defensive significantly



Case: S&P 500 and Tesla

The **S&P 500 Index** is a market-capitalization-weighted index of the 500 leading publicly traded companies in the US, widely seen as a key indicator of overall stock market performance

Tesla (TSLA) is one of the most talked-about and actively traded stocks in the market, known for its volatility, strong brand, and disruptive innovation

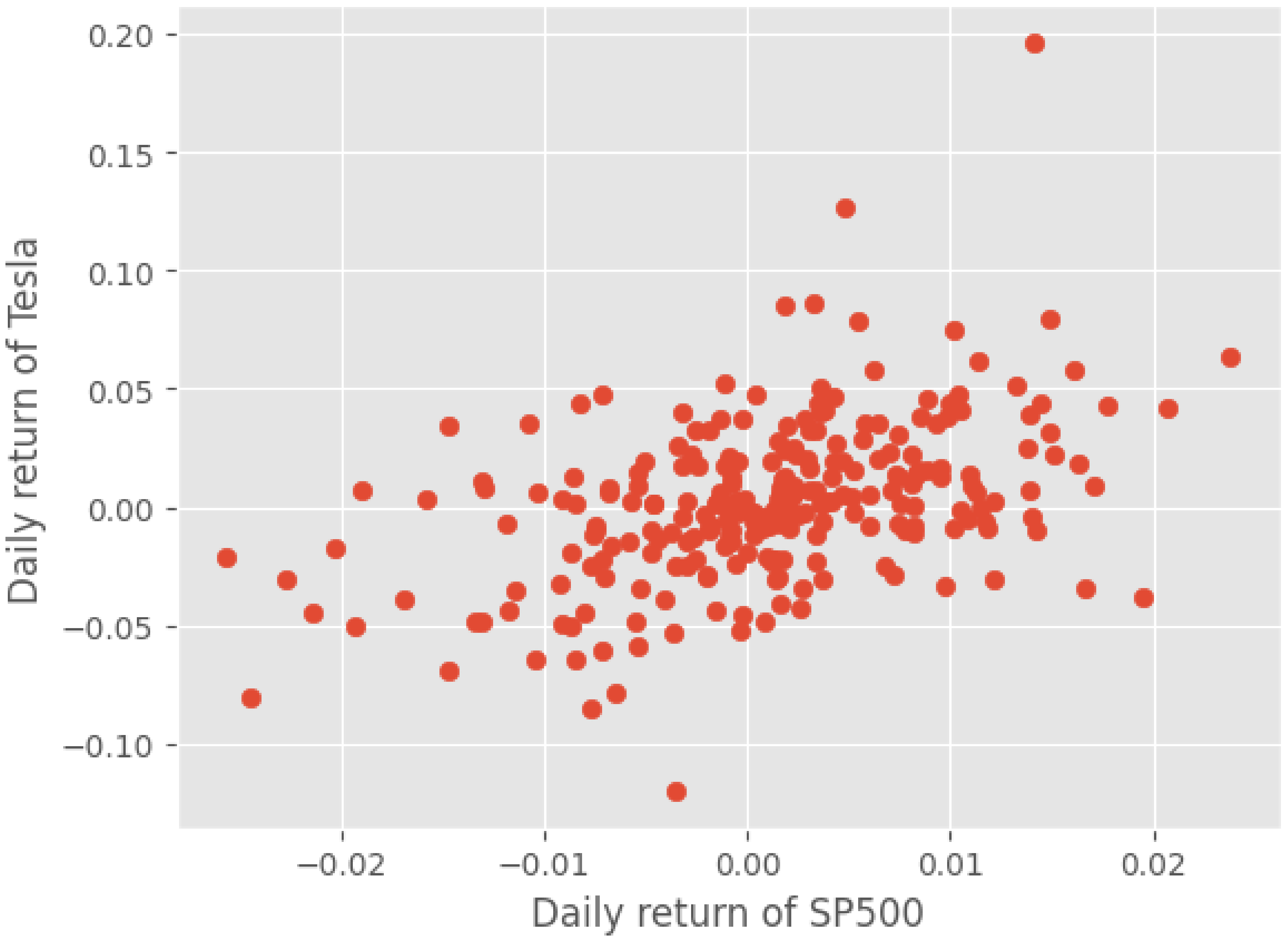
```python
returns = isom2600.data.returnsp500_tesla()
returns.head()
```

	SP500	Tesla
Date		
2021-01-04	-0.014755	0.034152
2021-01-05	0.007083	0.007317
2021-01-06	0.005710	0.028390
2021-01-07	0.014847	0.079447
2021-01-08	0.005492	0.078403

Is S&P 500 associated with Tesla?

Scatterplot

```
plt.scatter(returns["SP500"],returns["Tesla"])
plt.xlabel("Daily return of SP500")
plt.ylabel("Daily return of Tesla")
plt.show()
```



Question: Is there a clear pattern?

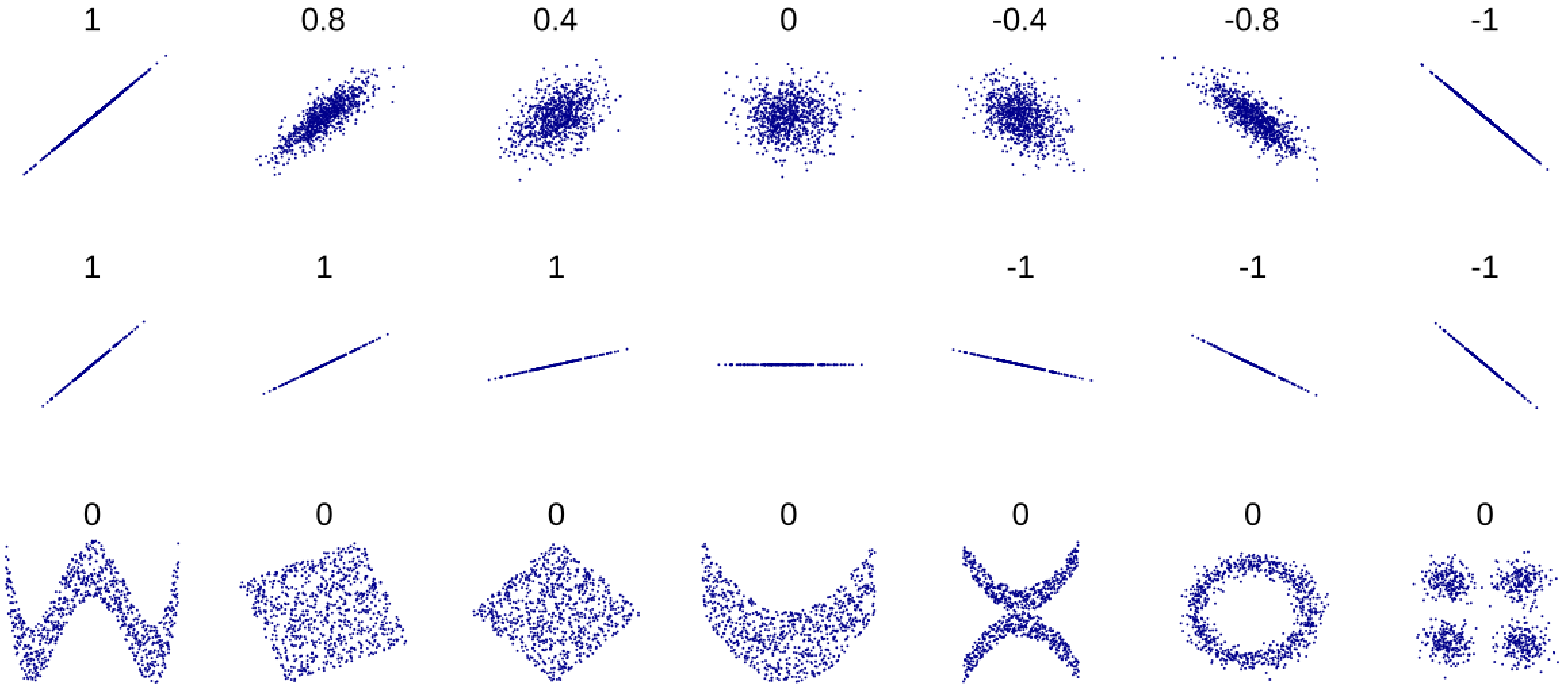
- ❑ **Direction:** trend up (positive)
- ❑ **Curvature:** linear
- ❑ **Variation:** considerable
- ❑ **Outliers:** no apparent outliers

```
returns.corr()
```

	SP500	Tesla
SP500	1.000000	0.448253
Tesla	0.448253	1.000000

Extra: Association and Correlation

<u>Note:</u> correlation coefficient can be applied to evaluate the **strength of association** only if the pattern is **linear**



Strong Non-linear Association

Case: S&P 500 and Tesla

The **Single-Index Model (SIM)** is a simple linear regression model to estimate the relationship between $\boldsymbol{R}$: ***a specific stock's return (here is TSLA)*** and the $\boldsymbol{R_m}$: ***overall market return (here is SP500).*** It assumes that a single factor (market return) explains most of a stock's return variations

$$R = \alpha + \beta * R_m + \varepsilon$$

- ❑ α is the **intercept parameter** (β_0 in general form), the stock's **alpha** (expected return when market return $R_{\mathrm{m}} = 0$)
- ❑ β is the **slope parameter** (β_1 in general form), the stock's **beta** (sensitivity to market return) – systematic risk
- ❑ ε is the idiosyncratic risk (random error), it is the firm-specific risk which is unrelated to market – unsystematic risk

Case: S&P 500 and Tesla

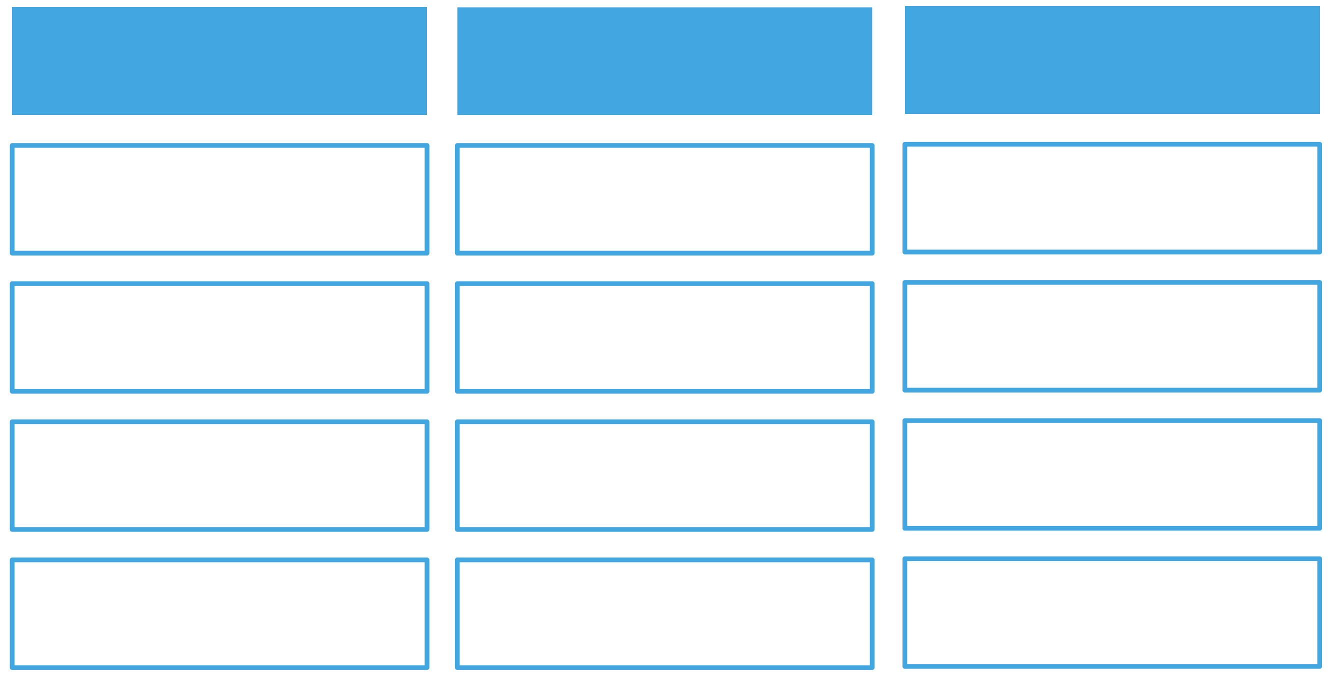
Split data into training and testing sets: 80% and 20% (shuffle data first)

Train model in the training set, and **evaluate** the fitted model in both the testing set and training set

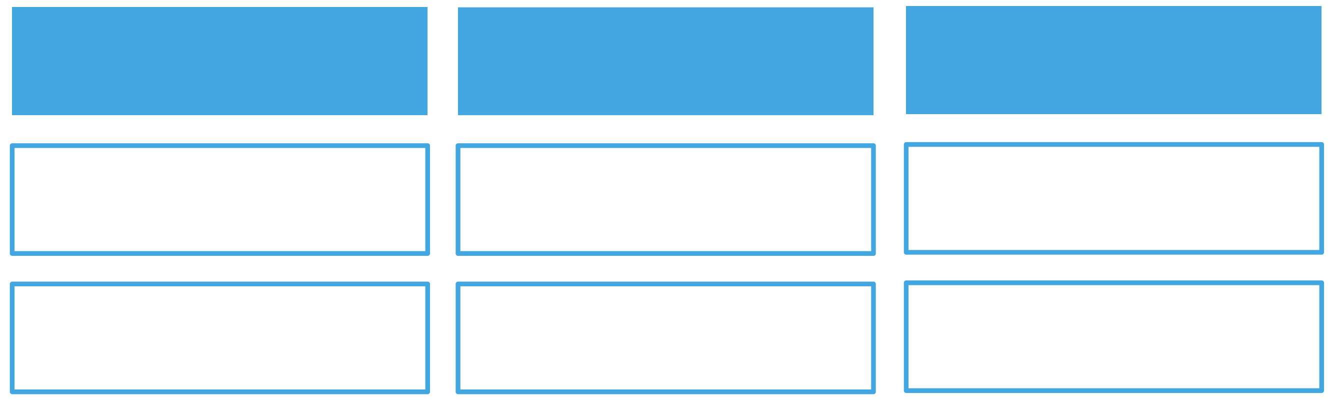
```
trainsize = int(len(returns)*0.8)
train = returns.iloc[:trainsize]
test = returns.iloc[trainsize:]
print("No of examples in train: ", train.shape)
print("No of examples in test: ", test.shape)
```

```
No of examples in train:  (201, 2)
No of examples in test:  (51, 2)
```

Training Data



Testing Data



Case: S&P 500 and Tesla

Build the model with the statsmodels (sm) library on training set

```
model = sm.OLS(train['Tesla'], sm.add_constant(train['SP500'])).fit()
```

- With Intercept: add_constant to x column if your model is $Y = \beta_0 + \beta_1 X + \varepsilon$
- Without Intercept: no need to add_constant to predictor x if $Y = \beta_1 X + \varepsilon$

What is the difference between those two models?

Note: you should include intercept unless there is strong theoretical reason to exclude it

Case: S&P 500 and Tesla

Model without Intercept

```python
modelno = sm.OLS(train['Tesla'], train['SP500']).fit()
# without add constant, force b0=0
print(modelno.summary())
```

```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                  Tesla   R-squared (uncentered):                   0.207
Model:                            OLS   Adj. R-squared (uncentered):              0.203
Method:                 Least Squares   F-statistic:                              52.08
Date:                Thu, 06 Mar 2025   Prob (F-statistic):                    1.08e-11
Time:                        08:45:18   Log-Likelihood:                          428.03
No. Observations:                 201   AIC:                                     -854.1
Df Residuals:                     200   BIC:                                     -850.7
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
SP500          1.8042      0.250      7.217      0.000       1.311       2.297
==============================================================================
Omnibus:                       58.902   Durbin-Watson:                   2.229
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              292.279
Skew:                           1.010   Prob(JB):                     3.41e-64
Kurtosis:                       8.551   Cond. No.                         1.00
==============================================================================
```

Model with Intercept

```python
model = sm.OLS(train['Tesla'], sm.add_constant(train['SP500'])).fit()
# fit model=build model= estimate intercept and slope
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Tesla   R-squared:                       0.205
Model:                            OLS   Adj. R-squared:                  0.201
Method:                 Least Squares   F-statistic:                     51.28
Date:                Thu, 06 Mar 2025   Prob (F-statistic):           1.52e-11
Time:                        08:39:20   Log-Likelihood:                 428.03
No. Observations:                 201   AIC:                            -852.1
Df Residuals:                     199   BIC:                            -845.5
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0002      0.002     -0.099      0.921      -0.004       0.004
SP500          1.8071      0.252      7.161      0.000       1.309       2.305
==============================================================================
Omnibus:                       58.854   Durbin-Watson:                   2.229
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              291.774
Skew:                           1.010   Prob(JB):                     4.39e-64
Kurtosis:                       8.546   Cond. No.                         124.
==============================================================================
```

What does P_value of intercept mean?

Hypothesis Testing

A regression model is not likely to be useful unless there is a **significant relationship** between X and Y

When **<u>no linear relationship</u>** exists between two variables, the regression line should be horizontal, i.e. $\beta_1 = 0$ .The best estimate we have is $\overline{y}$

To test for **significance**, we use the following **hypothesis (*default in Python*):**

$H_0: \beta_1 = 0)$ No linear relationship) vs $H_a: \beta_1 \neq 0$
$H_0: \beta_0 = 0)$ No Intercept) vs $H_a: \beta_0 \neq 0$

Extra: we will test $H_a: \beta_1 > 0 \ or \ H_a: \beta_1 < 0$ **or** $H_a: \beta_1 > a \ or \ H_a: \beta_1 < a$ **when we have more information**

Case: S&P 500 and Tesla

Make prediction

Single Value Prediction: predict value of one response variable: daily return of Tesla ($\hat{y}$) when the independent variable $x = -0.05$

```
model.predict([1, -0.05])
```

```
array([-0.09055942])
```

All Predictions: predict value of the response variable for all SP500 returns in the training dataset

```
yhat = model.predict(sm.add_constant(train['SP500']))
```

Case: S&P 500 and Tesla

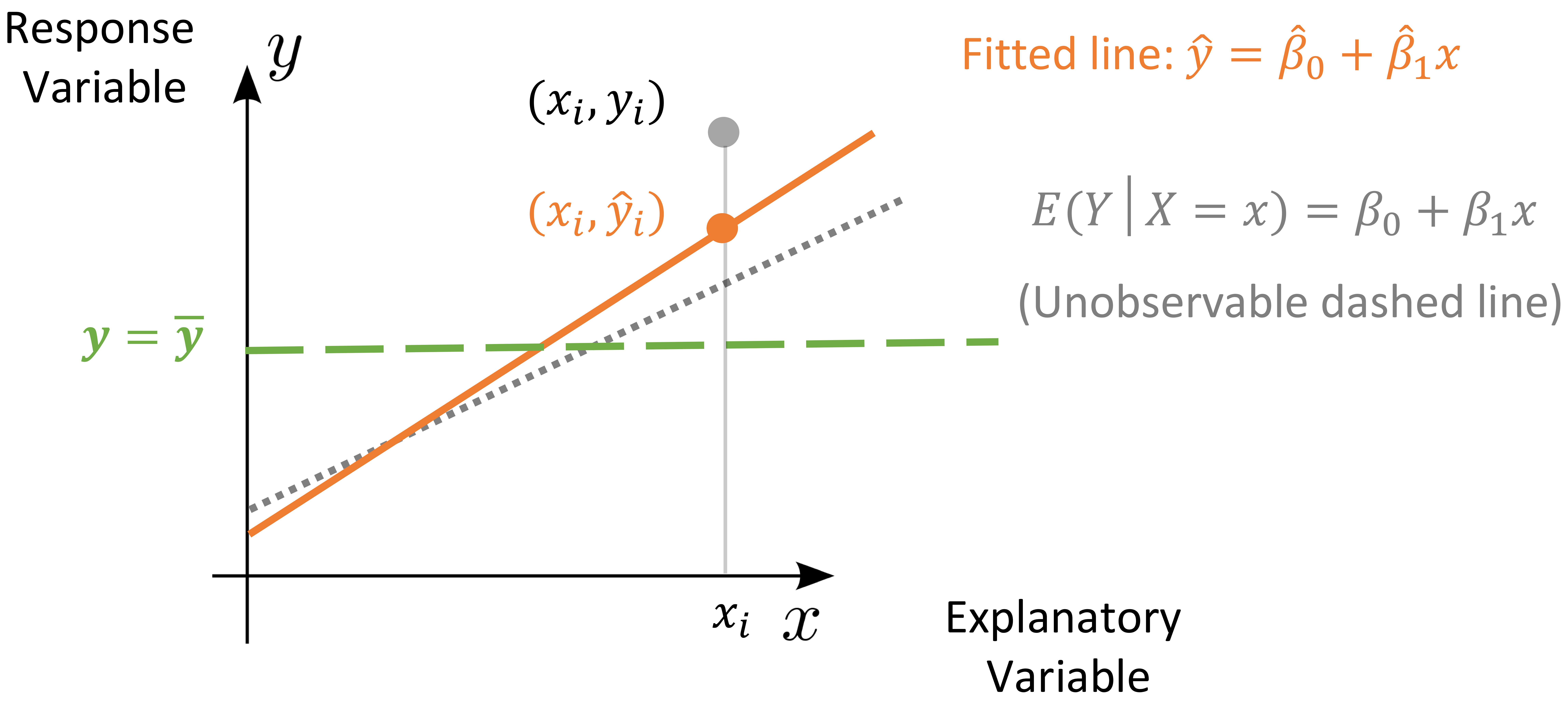
Add the predictions to the training set as a new column

```
train['Tesla_Pred'] = model.fittedvalues
train
```

Date	SP500	Tesla	Tesla_Pred
2021-01-04	-0.014755	0.034152	-0.026867
2021-01-05	0.007083	0.007317	0.012595
2021-01-06	0.005710	0.028390	0.010115
2021-01-07	0.014847	0.079447	0.026627
2021-01-08	0.005492	0.078403	0.009721

Create a new column in the train DataFrame as“Tesla_Pred”, the new column contains the predictions of Tesla price for all SP500 in the training dataset: model.fittedvalues (It is equivalent to yhat in the previous page)

Residual



Response Variable
y
(x_i, y_i)
$(x_i, \hat{y}_i)$
$y = \bar{y}$
x_i
x
Explanatory Variable
Fitted line: $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$
$E(Y \mid X = x) = \beta_0 + \beta_1 x$
(Unobservable dashed line)

Case: S&P 500 and Tesla

Get the residuals $e_i = \mathrm{y_i} - \hat{y}_i$

```
residuals = model.resid
train['Residuals'] = residuals
train.head()
```

Date	SP500	Tesla	Tesla_Pred	Residuals
2021-01-04	-0.014755	0.034152	-0.026867	0.061019
2021-01-05	0.007083	0.007317	0.012595	-0.005278
2021-01-06	0.005710	0.028390	0.010115	0.018275
2021-01-07	0.014847	0.079447	0.026627	0.052819
2021-01-08	0.005492	0.078403	0.009721	0.068682

MSE and RMSE

Residuals $e_i = \mathrm{y_i} - \hat{y}_i$ is the difference between the observed value of y_i and the predicted value $\widehat{y_i}$, it is the point estimate of ε_i error term

Mean Squared Error (MSE) $s_e^2 = \ MSE \ = \frac{SSE}{n-2} = \frac{\sum(y_i - \hat{y}_i)^2}{n-2}$, it is the **point estimate** of the variance σ^2 of the error term

Root Mean Squared Error (RMSE) is also called **Standard Error**

$$s_e = \sqrt{MSE} = \sqrt{\frac{SSE}{n-2}} = \sqrt{\frac{\sum e_i^2}{n-2}}$$

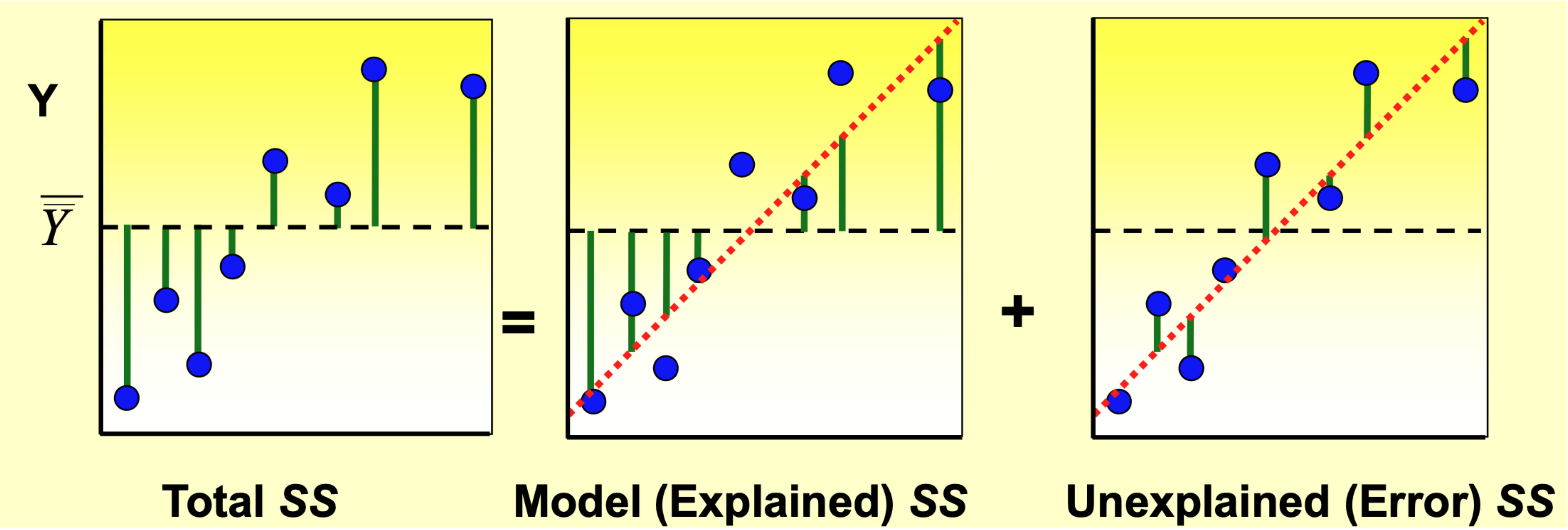
This is the **point estimate** of the standard deviation σ of the **error term**

R^2

Overall Variability in Y

Explained in part by → The regression model

Remains, in part, unexplained → The error



$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2 \quad SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \quad SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

R^2

How useful is a particular regression model?

The fraction of the variation that is explained determines the goodness of the regression, and is called the **coefficient of determination**, which is represented by the symbol R^2

R^2 measures the **proportion of the variation** in Y that is explained by the the **regression model**

$$R^2 = \frac{\text{Explained Variation}}{\text{Total Variation}} = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

Case: S&P 500 and Tesla

Evaluate the model performance in training set: R^2, MSE and RMSE

```
mse = model.mse_resid
print(mse)
```

```
0.0008359973379095491
```

```
r_squared = model.rsquared
print(r_squared)
```

```
0.20489272018556537
```

```
rmse = np.sqrt(mse)
print(rmse)
```

```
0.028913618554403546
```

Case: S&P 500 and Tesla

Extra: compare the model performance in testing set and training set

Performance Metric	MSE	RMSE
Training Data	0.00084	0.02891
Test Data	0.00148	0.03845

Note: the model performs better on the training set than on the testing set, because model has already seen the learned from the training data, testing data is new to the model

Residual Analysis

If a regression line works well, it captures the underlying pattern, the residuals should look like they have been randomly and independently selected from normally distributed populations having mean 0 and variance σ^2

- Residuals e_i versus explanatory variables x_i (pairs of data): the plot should have **no pattern**

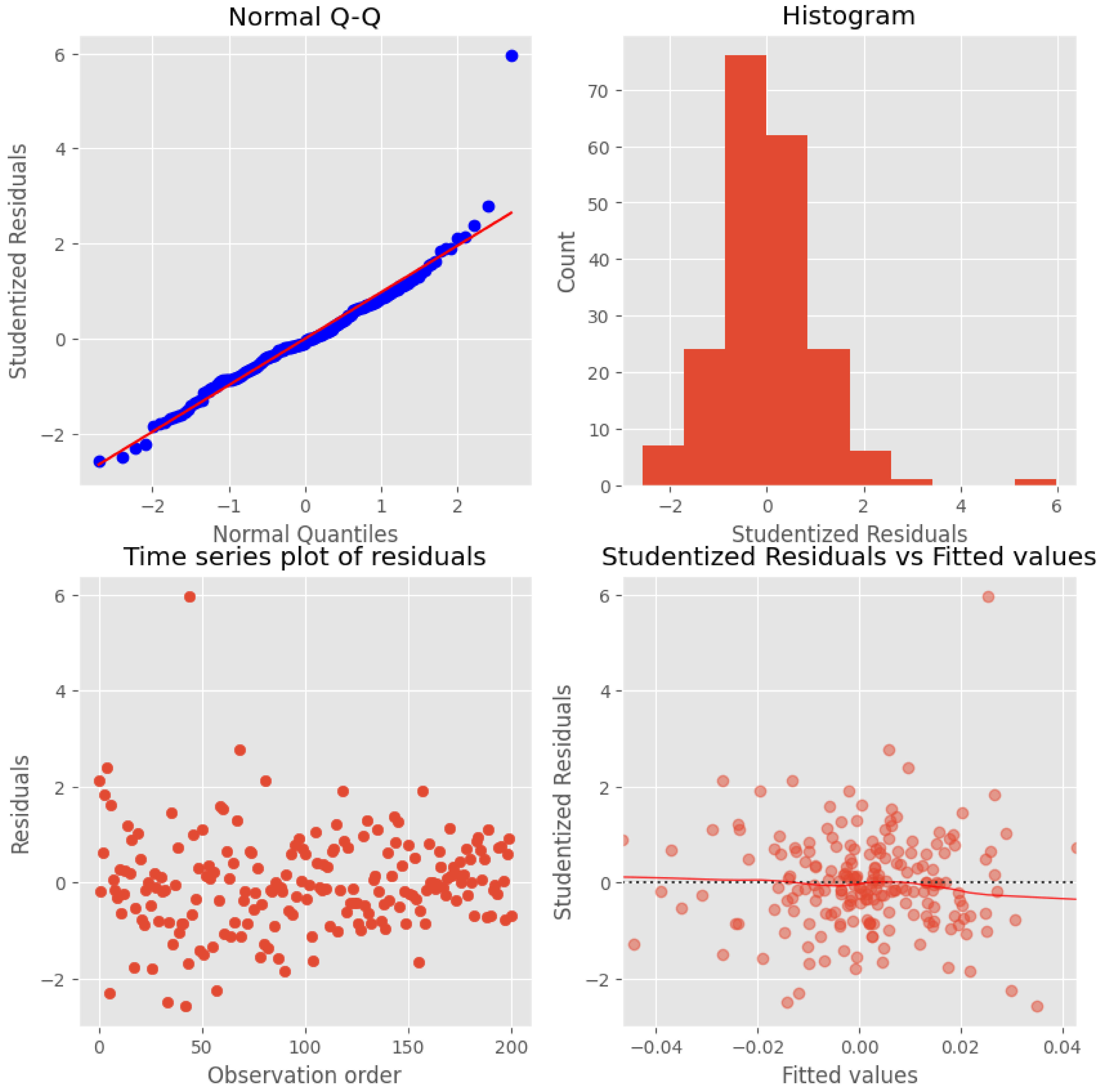
- Residuals e_i versus predicted $\hat{y}$ - **no pattern**

- Inspect the **histogram** and **QQ plot** of the residuals – the histogram approximates **normal**

Case: S&P 500 and Tesla

Normality

Random and Independent



Case: S&P 500 and Tesla

How to interpret the α and β in the model? (Interpretation of Parameters)

α is the **abnormal return** (independent of the market)

- If α is positive significantly, the stock outperforms the market

β measures the risk exposure of the specific stock to the overall market risk

- $\beta > 1$ means stock is **more volatile** than the market (aggressive stock)
- $0 < \beta < 1$ means stock is **less volatile** than the market (defensive stock)
- $\beta < 0$ means moving **opposite** to the market

Case: S&P 500 and Tesla

Interpret the fitted line:

$$R = -0.0002 + 1.8071 * R_m + \varepsilon$$

- **Intercept $\widehat{\alpha} = \mathbf{0.02\%}$**: the estimated return of Tesla stock when the market return is zero. It is positive but not significantly, meaning Tesla has no significant alpha (no excess return independent of the market)

- **Slope Estimation $\widehat{\beta} = \mathbf{1.8071}$**: If the market moves by **1%**, the stock tends to move by **1.8071%** on average. Tesla is **more volatile** than the market. It is a high-beta stock, which is more aggressive with higher risk and higher expected return