Lesson 21: Linear Regression

Data Science with Python – BSc Course

45 Minutes

Learning Objectives

The Problem: A portfolio manager needs to understand how individual stocks respond to market movements. Given historical returns, how do we quantify systematic risk?

After this lesson, you will be able to:

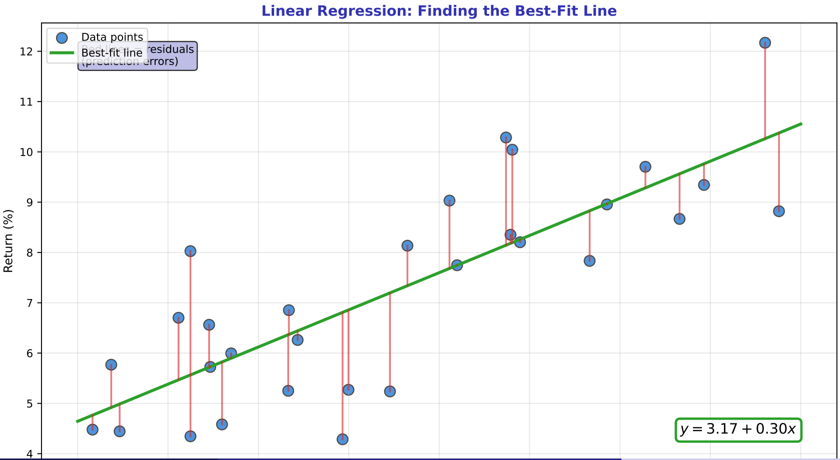
- Understand OLS estimation and the least squares principle
- Fit linear models using sklearn's LinearRegression
- Interpret coefficients (slope as beta, intercept as alpha)
- Estimate CAPM beta to classify stocks by risk profile

Finance Application: Stock classification for portfolio construction

Regression Concept

Finding the Best-Fit Line

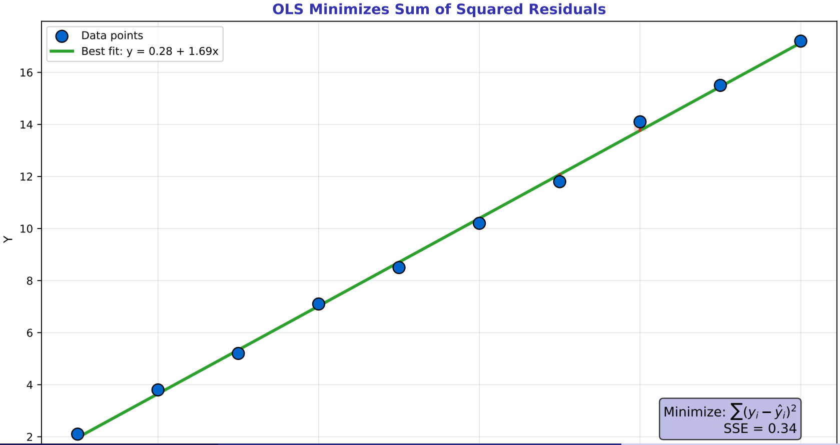
- Linear regression finds the line that best describes the relationship between variables
- In finance: How does stock return respond to market return?
- The slope tells us sensitivity; the intercept tells us baseline return



OLS Formula

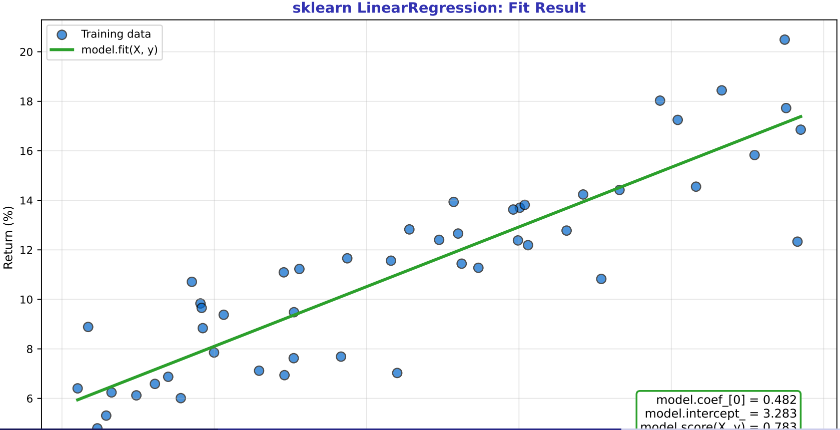
Ordinary Least Squares: The Math Behind Regression

- Goal: Find β_0 (intercept) and β_1 (slope) that minimize $\sum(y_i - \hat{y}_i)^2$
- Closed-form solution: $\beta_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$
- In Python: One line with sklearn (no manual calculation needed)



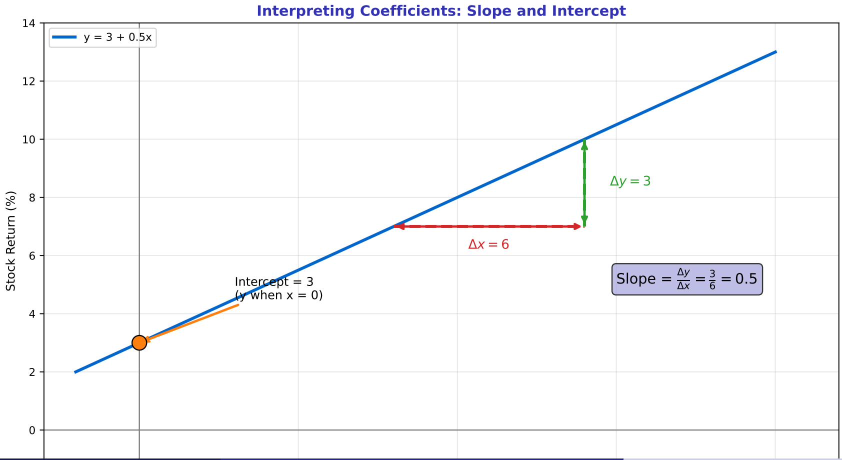
Implementation in Python

- Three steps: import, fit, predict
- `from sklearn.linear_model import LinearRegression`
- `model = LinearRegression().fit(X, y)`
- Access coefficients: `model.coef_` (slope), `model.intercept_`



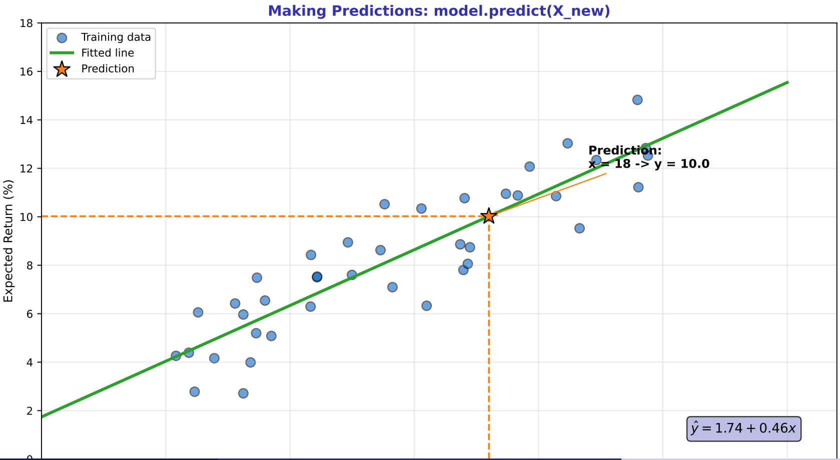
What Do the Numbers Mean?

- **Slope (β_1):** For each 1% market move, stock moves β_1%
- **Intercept (β_0):** Stock's return when market return is zero (alpha)
- Example: If $\beta_1 = 1.2$, stock amplifies market by 20%



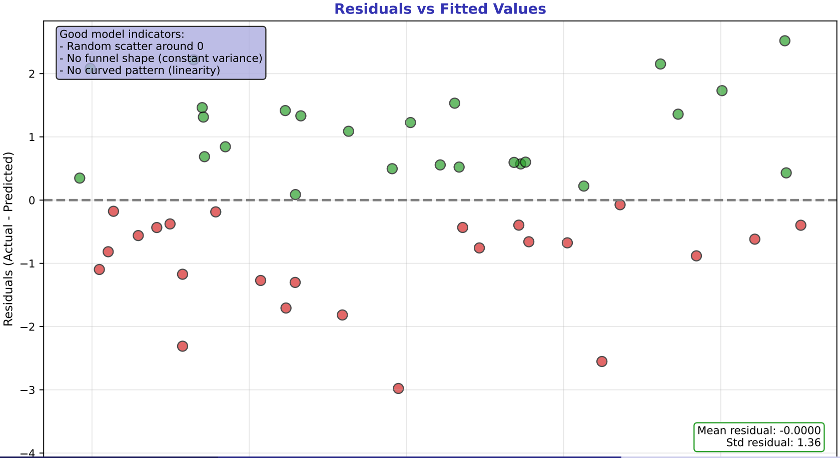
Using the Model for Forecasting

- Once fitted, predict stock return for any market scenario
- `predicted = model.predict([[market_return]])`
- Prediction equation: $\hat{y} = \beta_0 + \beta_1 \cdot x_{market}$



Checking Prediction Quality

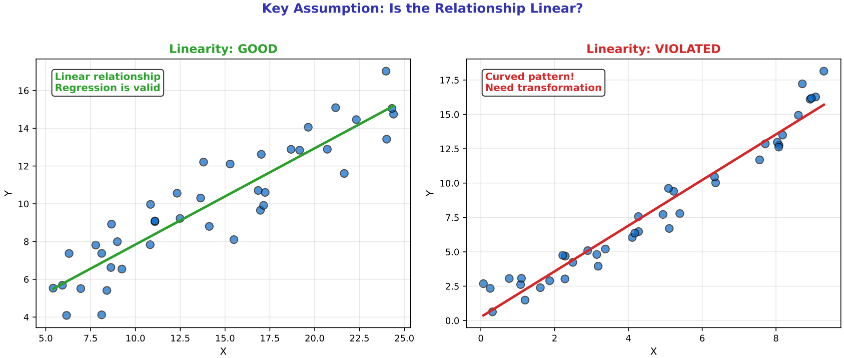
- Residual = Actual - Predicted ($e_i = y_i - \hat{y}_i$)
- Good model: residuals should be random (no pattern)
- Large residuals indicate outliers or model misspecification



Assumptions

When Does Linear Regression Work?

- **Linearity:** Relationship is actually linear (not curved)
- **Independence:** Observations don't influence each other
- **Homoscedasticity:** Variance of errors is constant
- **Normality:** Residuals are normally distributed

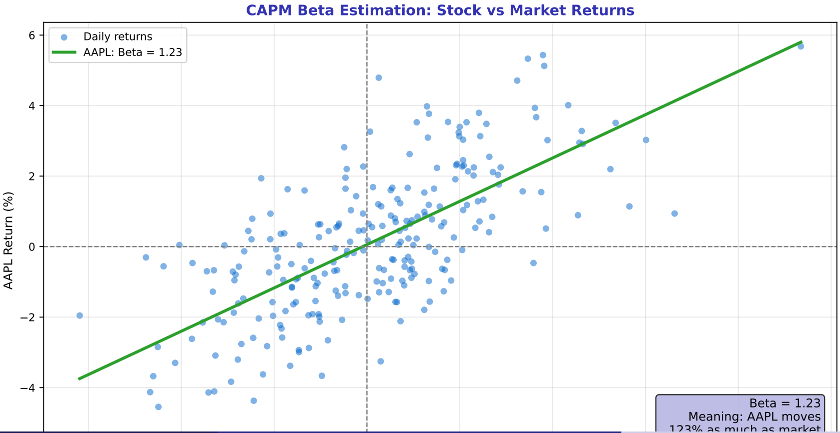


Finance reality: Stock returns often violate these – check residuals before trusting results

CAPM Beta

The Solution: Stock Classification by Systematic Risk

- **Beta > 1:** Aggressive stock (TSLA: $\beta = 1.8$) – amplifies market
- **Beta $= 1$:** Market-tracking (index funds)
- **Beta < 1:** Defensive stock (JNJ: $\beta = 0.7$) – dampens volatility
- Alpha (β_0): Outperformance after risk adjustment



Hands-On Exercise (25 min)

Task: Estimate Beta for Your Favorite Stock

1. Download 1 year of daily returns for a stock (e.g., MSFT) and SPY (market)
2. Fit a linear regression: `model.fit(spy_returns, stock_returns)`
3. Extract and interpret: What is the beta? What is the alpha?
4. Plot the regression line with actual data points
5. Check residuals: Do they look random?

Deliverable: A scatter plot with regression line, annotated with beta and alpha values.

Extension: Compare beta estimates using different time periods (1yr vs 5yr) – do they differ?

Lesson Summary

Problem Solved: We can now quantify systematic risk using CAPM beta estimated via linear regression.

Key Takeaways:

- OLS finds the line that minimizes squared errors
- sklearn: `LinearRegression().fit(X, y)` – three lines of code
- Slope = beta (market sensitivity), Intercept = alpha (skill)
- Beta > 1 aggressive, Beta < 1 defensive

Next Lesson: Regularization (L22) – what happens when we have too many features?

Memory: Beta = slope of stock vs market regression. High beta = high volatility.