DATA 201: Time Series Analysis
Linear Regression Model
Lecture 4: LRM with one independent variable

Lulu Wang

Data Analytics
Dickinson College

1/31/2025

Introduction

The Linear Regression Model

Robust Standard Errors

Functional Forms

Objectives

- Review the Linear Regression Model (LRM)
- Discuss its estimation in Python
- Explore applications in financial data

I will review the regression model in broad terms and more details can be found in an introductory statistics/econometrics textbook, such as:

- Stock and Watson, *Introduction to Econometrics*, Pearson
- Wooldridge, *Introductory Econometrics*, A Modern Approach, South-Western

The Linear Regression Model (LRM)

- The linear regression model is given by:

$$Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t,$$

where:

  - Y_t: dependent variable at time t
  - X_t: independent variable / factor / predictor at time t
  - β_0, β_1: coefficients to be estimated
  - ε_t: error term (mean zero, variance σ^2)

- The **expected** (or average) value of Y_t given X_t is:

$$E(Y_t \mid X_t) = \beta_0 + \beta_1 X_t.$$

Interpretation and CAPM

- **Interpretation**:
  - β_0: the expected value of Y_t when $X_t = 0$
  - β_1: the expected change of Y_t for a unit change of X_t
- The Capital Asset Pricing Model (CAPM) is an example of an LRM:

$$R_t^i \;=\; \beta_0 \;+\; \beta_1 \, R_t^{\mathrm{MKT}} + \varepsilon_t,$$

where R_t^i and R_t^{MKT} represent the excess stock and market returns, respectively.

OLS estimation

- Ordinary Least Squares (OLS) is a method to estimate the coefficients β_0 and β_1 from a sample of observations of X_t and Y_t.
- **OLS recipe**: choose the values of $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize the Sum of the Square Residuals (SSR), i.e. $\sum_{t=1}^{T} \hat{\varepsilon}_t^2$.
- Notice that we use ˆ (ˆ) to denote *estimated* quantities of a population parameter (e.g. $\hat{\beta}_1$ vs. β_1).
- In the simple case of the LRM with only one independent variable, we have analytical formulas for the estimators $\hat{\beta}_0$ and $\hat{\beta}_1$.

OLS formulas for the simple LRM

1. $\hat{\beta}_1 = \dfrac{\sigma_{X,Y}}{\sigma_X^2} = \rho_{X,Y}\,\dfrac{\sigma_Y}{\sigma_X}$
   - $\sigma_{X,Y}$: sample covariance of X_t and Y_t
   - σ_X: sample standard deviation of X_t
   - σ_Y: sample standard deviation of Y_t

2. $\hat{\beta}_0 = \overline{Y} - \hat{\beta}_1\,\overline{X}$
   - $\overline{Y}$: sample mean of Y_t
   - $\overline{X}$: sample mean of X_t

OLS Example

- Assume that our dependent variable is the return of the equity market in excess of the risk-free rate, and the independent variable is the DP ratio.
- The LRM is then

$$EP_{t+1}^{CRSP} \;=\; \beta_0 \;+\; \beta_1 \, DP_t \;+\; \varepsilon_t .$$

- The quantity $\beta_0 + \beta_1 \, DP_t$ represents:
  - the regression line as a function of DP,
  - the expected equity premium in the following time period when the current dividend-price ratio is equal to DP_t, that is,

$$E\left(EP_{t+1}^{CRSP} \mid DP_t\right) \;=\; \beta_0 \;+\; \beta_1 \, DP_t .$$

Esimation in Python

- Rather than manually computing these estimates, Python's **statsmodels** automatically calculates regression coefficients, standard errors, and other diagnostic measures:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                ep_crsp   R-squared:                       0.034
Model:                            OLS   Adj. R-squared:                  0.023
Method:                 Least Squares   F-statistic:                     3.061
Date:                Thu, 30 Jan 2025   Prob (F-statistic):             0.0837
Time:                        19:51:46   Log-Likelihood:                -392.57
No. Observations:                  89   AIC:                             789.1
Df Residuals:                      87   BIC:                             794.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1903      5.274     -0.036      0.971     -10.673      10.292
DP             2.1865      1.250      1.749      0.084      -0.298       4.671
==============================================================================
Omnibus:                        3.676   Durbin-Watson:                   1.852
Prob(Omnibus):                  0.159   Jarque-Bera (JB):                3.340
Skew:                          -0.474   Prob(JB):                        0.188
Kurtosis:                       3.021   Cond. No.                         10.9
==============================================================================
```

Fitted Values & Residuals

- Based on the coefficient estimates, we can then calculate the fitted values and the residuals of the regression model.
  - The fitted values:

$$\hat{\text{EP}}_{t+1} = \beta_0 + \beta_1 \times DP_t$$

  - The residuals measure the difference between actual and predicted returns.

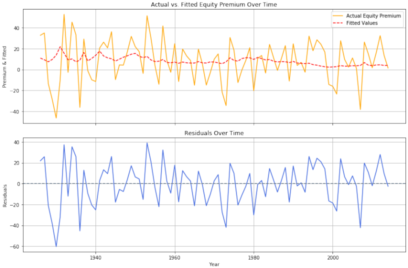


Figure 1: Time series of the realized and predicted equity premium (top) and the residuals obtained as the difference between the realized and predicted equity premium (bottom)

Robust Standard Errors (Overview)

- By default, Python's `statsmodels` often assumes:
  - **Homoskedasticity**: errors have constant variance
  - **No autocorrelation**: errors are independent over time
- If errors are *heteroskedastic* or *correlated*, standard errors from the default approach are not reliable.
- OLS estimates of β_0 and β_1 remain unbiased (under certain conditions), but naive standard errors can understate the true uncertainty.
- Two main robust adjustments:
  - **HC (Heteroskedasticity-Consistent)** for cross-sectional or panel data.
  - **HAC (Heteroskedasticity and Autocorrelation Consistent) or Newey-West** for time series data.

Deciding on Robust Standard Errors in Python

- **Diagnostic or Default?**
  - Run tests for heteroskedasticity (e.g. Breusch-Pagan) or autocorrelation (e.g. Durbin-Watson).
  - Or apply robust/HAC SE by default (slight efficiency loss if errors are actually homoskedastic and uncorrelated).
- Typically, robust/HAC SEs are larger than default SEs, reflecting real-world uncertainties.
- **In Python (statsmodels)**:
  - For cross-sectional data:
    `model = sm.OLS(y, X).fit(cov_type=‘HC3‘)`
  - For time series (Newey-West):
    `model_hac = sm.OLS(y, X).fit(cov_type=‘HAC‘, cov_kwds=‘maxlags‘:  lag_length)`

Newey–West Example in Python

```
=== OLS Results (Default Standard Errors) ===
                            OLS Regression Results
==============================================================================
Dep. Variable:                ep_crsp   R-squared:                       0.034
Model:                            OLS   Adj. R-squared:                  0.023
Method:                 Least Squares   F-statistic:                     3.061
Date:                Mon, 03 Feb 2025   Prob (F-statistic):             0.0837
Time:                        13:02:36   Log-Likelihood:                -392.57
No. Observations:                  89   AIC:                             789.1
Df Residuals:                      87   BIC:                             794.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1903      5.274     -0.036      0.971     -10.673      10.292
DP             2.1865      1.250      1.749      0.084      -0.298       4.671
==============================================================================
Omnibus:                        3.676   Durbin-Watson:                   1.852
Prob(Omnibus):                  0.159   Jarque-Bera (JB):                3.340
Skew:                          -0.474   Prob(JB):                        0.188
Kurtosis:                       3.021   Cond. No.                         10.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
=== OLS Results (HAC/ Newey-West Standard Errors) ===
                            OLS Regression Results
==============================================================================
Dep. Variable:                ep_crsp   R-squared:                       0.034
Model:                            OLS   Adj. R-squared:                  0.023
Method:                 Least Squares   F-statistic:                     2.135
Date:                Mon, 03 Feb 2025   Prob (F-statistic):              0.148
Time:                        13:02:41   Log-Likelihood:                -392.57
No. Observations:                  89   AIC:                             789.1
Df Residuals:                      87   BIC:                             794.1
Df Model:                           1
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1903      5.626     -0.034      0.973     -11.218      10.837
DP             2.1865      1.496      1.461      0.144      -0.746       5.119
==============================================================================
Omnibus:                        3.676   Durbin-Watson:                   1.852
Prob(Omnibus):                  0.159   Jarque-Bera (JB):                3.340
Skew:                          -0.474   Prob(JB):                        0.188
Kurtosis:                       3.021   Cond. No.                         10.9
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC) using 1 lags
```

- The coefficient estimates $(\hat{\beta}_0, \hat{\beta}_1)$ remain the same.
- Standard errors and p-values change to account for heteroskedasticity and/or autocorrelation.

Nonlinear regression models

- The Linear Regression Model (LRM) assumes a linear relationship between X and Y.
- A linear model implies that a **1-unit increase** in X results in a **constant** expected change in Y by β_1.
- However, some relationships are **nonlinear** (e.g., quadratic, logarithmic, exponential).
- Nonlinearity means that the **effect of X on Y varies** depending on the level of X.

Polynomial models

- One way to introduce **nonlinearity** is through the **Quadratic Model**:

$$Y_t = \beta_0 + \beta_1 X_t + \beta_2 X_t^2 + \varepsilon_t$$

- The **effect** of changing X by one unit on Y is given by:

$$\beta_1 + 2\beta_2 X$$

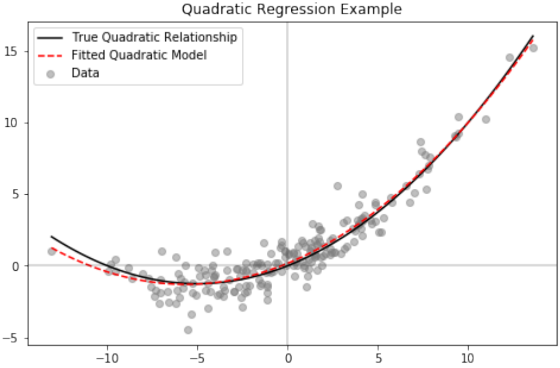
  (depends on X)

- Quadratic regression can still be estimated using **OLS** by adding X^2 as a regressor.

Simulating a Quadratic Model in Python

Example: Simulate $Y_t = 0.5X_t + 0.05X_t^2 + \varepsilon_t$, with:

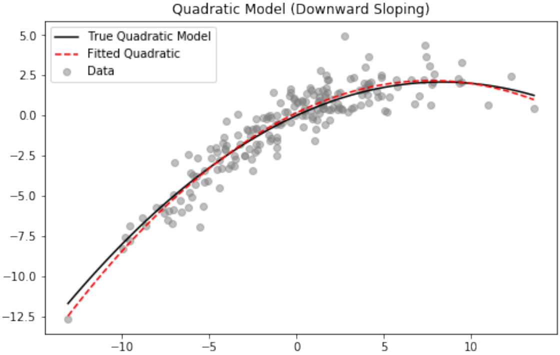
$$X_t \sim N(0, 25), \quad \varepsilon_t \sim N(0, 1)$$



Quadratic Model: Downward Sloping Parabola

- If the coefficient of X^2 is **negative**, the parabola slopes downward at the extremes.
- Below is a simulated quadratic model:

$$Y_t = 0.5X_t - 0.03X_t^2 + \varepsilon_t$$

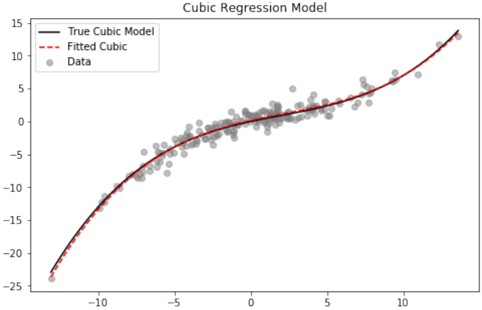


Cubic Regression Model

- A **cubic model** is useful when an **additional curvature** is needed to explain the relationship.
- The model is:

$$Y_t = \beta_0 + \beta_1 X_t + \beta_2 X_t^2 + \beta_3 X_t^3 + \varepsilon_t$$

- Including **higher-order terms** may introduce correlation among regressors, requiring careful evaluation.



Piecewise Linear Model

- This model assumes different slopes **below/above** a threshold m:

$$Y_t = \beta_0 + \beta_1 X_t I(X_t \geq m) + \beta_2 X_t I(X_t < m) + \varepsilon_t$$

- Interpretation:
  - The **effect** of X_t on Y_t is different for $X_t \geq m$ vs. $X_t < m$.
  - The slopes are determined by β_1 and β_2.