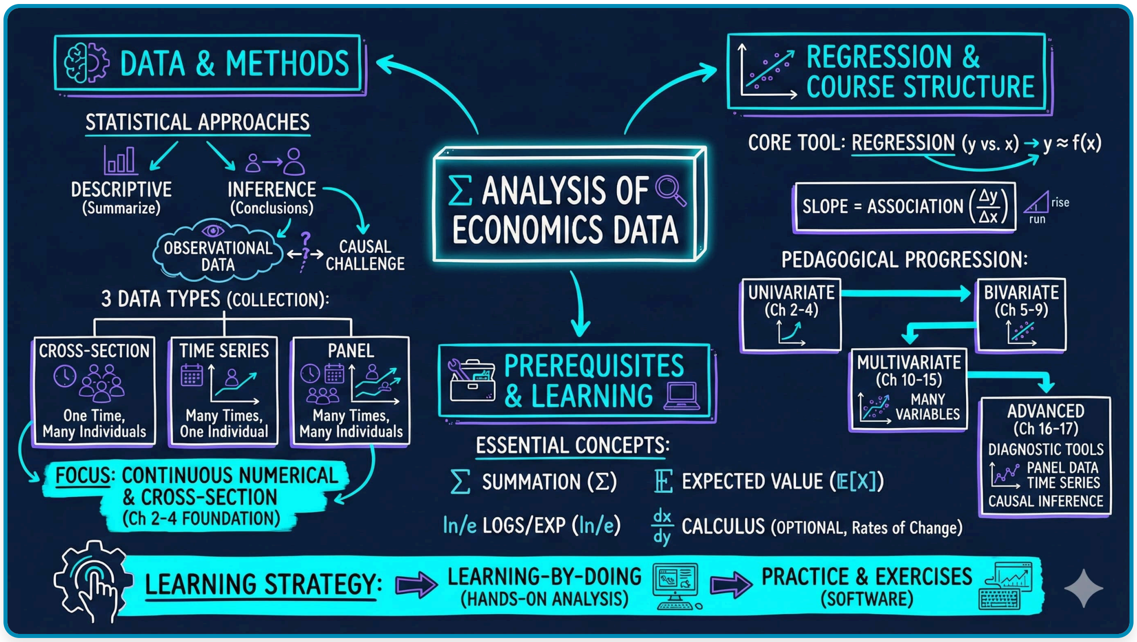
Chapter 1: Analysis of Economics Data

metricsAI: An Introduction to Econometrics with Python and AI in the Cloud

Carlos Mendez



This notebook provides an interactive introduction to regression analysis using Python. You can run all code directly in Google Colab without any local setup required. The data streams directly from GitHub, making this notebook fully self-contained.

Open in Colab

Learning Objectives

By the end of this chapter, you will be able to:

- Distinguish between descriptive analysis and statistical inference
- Identify different types of data (continuous, discrete, categorical)
- Understand the difference between observational and experimental data
- Recognize the three main data collection methods (cross-section, time series, panel)
- Understand the basic concept of regression analysis
- Navigate the structure and organization of this textbook

Chapter Overview

This chapter introduces the fundamental concepts of econometrics and regression analysis. We'll explore how economists use statistical methods to understand relationships in economic data, focusing on a practical example of house prices and house sizes.

What you'll learn:

- What regression analysis is and why it's the primary tool in econometrics
- How to load and explore economic data using Python (pandas)
- How to visualize relationships between variables using scatter plots
- How to fit a simple linear regression model using Ordinary Least Squares (OLS)
- How to interpret regression coefficients in economic terms
- How to use Python's statsmodels package for regression analysis

Dataset used:

- **AED_HOUSE.DTA**: House sale prices for 29 houses in Central Davis, California (1999)
    - Variables: price (sale price in dollars), size (house size in square feet), plus 7 other characteristics

Chapter outline:

- 1.1 What is Regression Analysis?
- 1.2 Load the Data
- 1.3 Preview the Data
- 1.4 Explore the Data
- 1.5 Visualizing the Relationship
- 1.6 Fitting a Regression Line
- 1.7 Interpreting the Results
- 1.8 Visualizing the Fitted Line
- 1.9 Economic Interpretation and Examples
- 1.10 Practice Exercises
- 1.11 Case Studies

Setup

Run this cell first to import all required packages and configure the environment. This sets up:

- Data manipulation (pandas, numpy)
- Statistical modeling (statsmodels)
- Visualization (matplotlib)
- Reproducibility (random seeds)

In [41]:

```
# Import required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.formula.api import ols
import random
import os

# Set random seeds for reproducibility
RANDOM_SEED = 42
random.seed(RANDOM_SEED)
np.random.seed(RANDOM_SEED)
os.environ['PYTHONHASHSEED'] = str(RANDOM_SEED)

# GitHub data URL (data streams directly from here)
GITHUB_DATA_URL = "https://raw.githubusercontent.com/quarcs-lab/data-open/master/AED/"

# Optional: Create directories for saving outputs locally
IMAGES_DIR = 'images'
TABLES_DIR = 'tables'
os.makedirs(IMAGES_DIR, exist_ok=True)
os.makedirs(TABLES_DIR, exist_ok=True)

print("✓ Setup complete! All packages imported successfully.")
print(f"✓ Random seed set to {RANDOM_SEED} for reproducibility.")
print(f"✓ Data will stream from: {GITHUB_DATA_URL}")
```

```
✓ Setup complete! All packages imported successfully.
✓ Random seed set to 42 for reproducibility.
✓ Data will stream from: https://raw.githubusercontent.com/quarcs-lab/data-open/master/AED/
```

1.1 What is Regression Analysis?

Regression analysis is the primary tool economists use to understand relationships between variables. At its core, regression answers questions like: "How does Y change when X changes?"

In our example:

- **Y (dependent variable)**: House sale price (in dollars)

- **X (independent variable)**: House size (in square feet)

The regression line is the "line of best fit" that minimizes the sum of squared distances between actual prices and predicted prices. The mathematical form is:

$$\text{price} = \beta_0 + \beta_1 \times \text{size} + \varepsilon$$

Where:

- β_0 = **intercept** (predicted price when size = 0)
- β_1 = **slope** (change in price for each additional square foot)
- ε = **error term** (random variation not explained by size)

Economic Interpretation:
The slope coefficient β_1 tells us: "On average, how much more expensive is a house that is 1 square foot larger?" This is a measure of **association**, not necessarily causation.

> **Key Concept**: *Descriptive analysis summarizes data using statistics and visualizations, while statistical inference uses sample data to draw conclusions about the broader population. Most econometric analysis involves statistical inference.*

1.2 Load the Data

Let's load the house price dataset directly from GitHub. This dataset contains information on 29 house sales in Central Davis, California in 1999.

In [42]:

```python
# Load the Stata dataset from GitHub
data_house = pd.read_stata(GITHUB_DATA_URL + 'AED_HOUSE.DTA')

print(f"✓ Data loaded successfully!")
print(f"  Shape: {data_house.shape[0]} observations, {data_house.shape[1]} variables")
```

```
✓ Data loaded successfully!
  Shape: 29 observations, 8 variables
```

1.3 Preview the Data

Let's look at the first few rows to understand what variables we have available.

In [43]:

```
# Display first 5 rows
print("First 5 observations:")
print(data_house.head())

print("\nColumn names:")
print(data_house.columns.tolist())
```

```
First 5 observations:
    price  size  bedrooms  bathrooms  lotsize   age  monthsold    list
0  204000  1400         3        2.0        1  31.0          7  199900
1  212000  1600         3        3.0        2  33.0          5  212000
2  213000  1800         3        2.0        2  51.0          4  219900
3  220000  1600         3        2.0        1  49.0          4  229000
4  224500  2100         4        2.5        2  47.0          6  224500

Column names:
['price', 'size', 'bedrooms', 'bathrooms', 'lotsize', 'age', 'monthsold', 'list']
```

Transition: Before jumping into regression analysis, we need to understand our data. Descriptive statistics reveal the scale, variability, and range of our variables—essential for interpreting regression results.

1.4 Explore the Data

Before running any regression, it's essential to understand the data through **descriptive statistics**. Let's look at the key statistics for our variables of interest: price and size.

In [44]:

```
# Summary statistics for all variables
print("=" * 70)
print("DESCRIPTIVE STATISTICS")
print("=" * 70)
print(data_house.describe().round(2))

# Focus on our key variables
print("\n" + "=" * 70)
print("KEY VARIABLES: PRICE AND SIZE")
print("=" * 70)
print(data_house[['price', 'size']].describe().round(2))
```

```
======================================================================
DESCRIPTIVE STATISTICS
======================================================================
           price     size  bedrooms  bathrooms  lotsize    age  monthsold  \
count      29.00    29.00     29.00      29.00    29.00  29.00      29.00
mean   253910.34  1882.76      3.79       2.21     2.14  36.41       5.97
std     37390.71   398.27      0.68       0.34     0.69   7.12       1.68
min    204000.00  1400.00      3.00       2.00     1.00  23.00       3.00
25%    233000.00  1600.00      3.00       2.00     2.00  31.00       5.00
50%    244000.00  1800.00      4.00       2.00     2.00  35.00       6.00
75%    270000.00  2000.00      4.00       2.50     3.00  39.00       7.00
max    375000.00  3300.00      6.00       3.00     3.00  51.00       8.00

            list
count      29.00
mean   257824.14
std     40860.26
min    199900.00
25%    239000.00
50%    245000.00
75%    269000.00
max    386000.00

======================================================================
KEY VARIABLES: PRICE AND SIZE
======================================================================
           price     size
count      29.00    29.00
mean   253910.34  1882.76
std     37390.71   398.27
min    204000.00  1400.00
25%    233000.00  1600.00
50%    244000.00  1800.00
75%    270000.00  2000.00
max    375000.00  3300.00
```

Key observations:

- **Mean house price**: Around $253,910
- **Mean house size**: Around 1,883 square feet
- **Price range**: $204,000 to $375,000
- **Size range**: 1,400 to 3,300 square feet

Notice the variation in both variables - this variation is what allows us to estimate a relationship!

> **Key Concept**: *Economics primarily uses observational data where we observe behavior in uncontrolled settings. Unlike experimental data where conditions can be controlled, observational data requires careful methods to establish relationships and, when possible, causal effects.*

1.5 Visualizing the Relationship

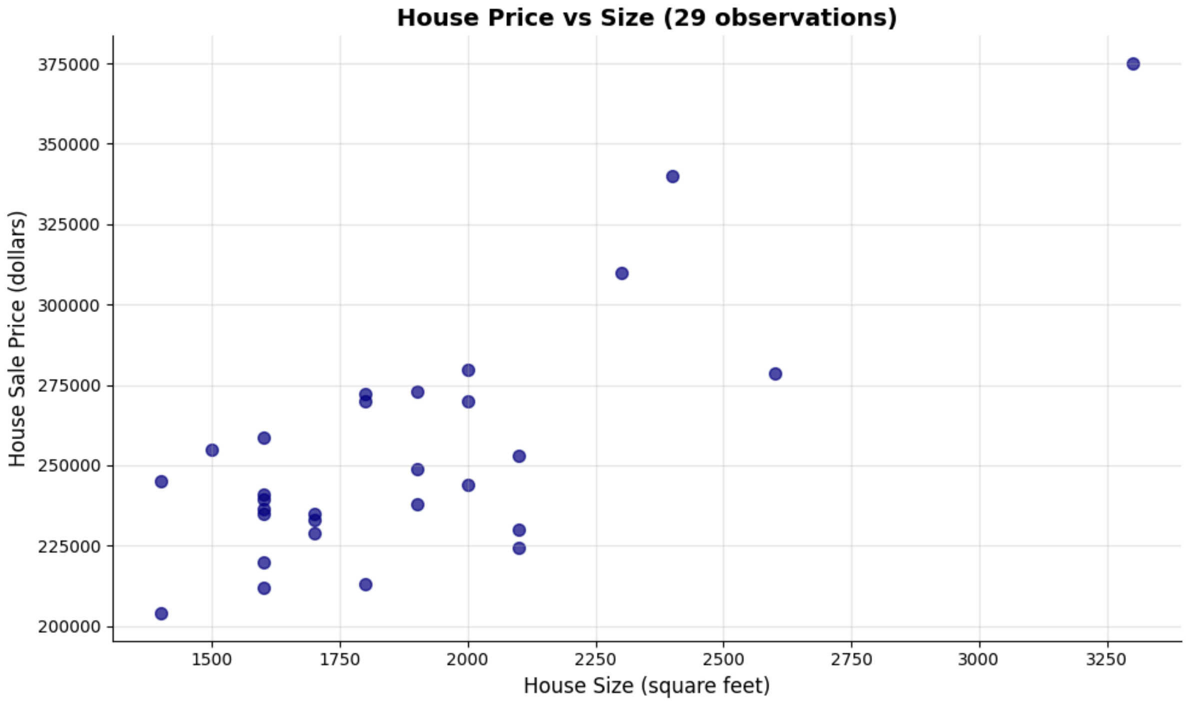
Before running any regression, it's good practice to visualize the relationship between X and Y. A scatter plot helps us:

1. Check if there appears to be a linear relationship
2. Identify any outliers or unusual observations
3. Get an intuitive sense of the strength of the relationship

Let's create a scatter plot of house price vs. house size.

In [45]:

```
# Create scatter plot
fig, ax = plt.subplots(figsize=(10, 6))

# Plot the data points
ax.scatter(data_house['size'], data_house['price'],
           color='navy', s=50, alpha=0.7)

# Labels and formatting
ax.set_xlabel('House Size (square feet)', fontsize=12)
ax.set_ylabel('House Sale Price (dollars)', fontsize=12)
ax.set_title('House Price vs Size (29 observations)', fontsize=14, fontweight='bold')
ax.grid(True, alpha=0.3)
ax.spines[['top', 'right']].set_visible(False)

plt.tight_layout()
plt.show()

print("\nWhat do you see?")
print("- Positive relationship: Larger houses tend to have higher prices")
print("- Roughly linear: The points follow an upward-sloping pattern")
print("- Some scatter: Not all points lie exactly on a line (this is the 'error')")
```




```
What do you see?
- Positive relationship: Larger houses tend to have higher prices
- Roughly linear: The points follow an upward-sloping pattern
- Some scatter: Not all points lie exactly on a line (this is the 'error')
```

Transition: Having visualized a clear positive relationship between house size and price, we're ready to quantify this relationship precisely using regression analysis.

1.6 Fitting a Regression Line

Now we'll fit an **Ordinary Least Squares (OLS)** regression line to these data. OLS chooses the intercept (β_0) and slope (β_1) that **minimize the sum of squared residuals**:

$$\min_{\beta_0, \beta_1} \sum_{i=1}^{n} (\text{price}_i - \beta_0 - \beta_1 \times \text{size}_i)^2$$

In other words, we're finding the line that makes our prediction errors as small as possible (in a squared sense).

We'll use Python's `statsmodels` package, which provides regression output similar to Stata and R.

> **Key Concept**: *Regression analysis quantifies the relationship between variables. In a bivariate regression, the slope coefficient tells us how much the outcome variable (y) changes when the explanatory variable (x) increases by one unit.*

In [46]:

```python
# Fit OLS regression: price ~ size
# The formula syntax is: 'dependent_variable ~ independent_variable'
model = ols('price ~ size', data=data_house).fit()

# Display the full regression output
print("=" * 70)
print("OLS REGRESSION RESULTS: price ~ size")
print("=" * 70)
print(model.summary())
```

```
======================================================================
OLS REGRESSION RESULTS: price ~ size
======================================================================
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.617
Model:                            OLS   Adj. R-squared:                  0.603
Method:                 Least Squares   F-statistic:                     43.58
Date:                Sat, 31 Jan 2026   Prob (F-statistic):           4.41e-07
Time:                        02:18:54   Log-Likelihood:                -332.05
No. Observations:                  29   AIC:                             668.1
Df Residuals:                      27   BIC:                             670.8
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    1.15e+05   2.15e+04      5.352      0.000    7.09e+04    1.59e+05
size          73.7710     11.175      6.601      0.000      50.842      96.700
==============================================================================
Omnibus:                        0.576   Durbin-Watson:                   1.219
Prob(Omnibus):                  0.750   Jarque-Bera (JB):                0.638
Skew:                          -0.078   Prob(JB):                        0.727
Kurtosis:                       2.290   Cond. No.                     9.45e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.45e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

1.7 Interpreting the Results

The regression output contains a lot of information! Let's break down the most important parts:

Key Statistics to Focus On:

1. **Coefficients table** (middle section):
    - **Intercept**: The predicted price when size = 0 (often not economically meaningful)
    - **size**: The slope coefficient - our main interest!
    - **std err**: Standard error (measures precision of the estimate)
    - **t**: t-statistic (coefficient / standard error)
    - **P>|t|**: p-value (tests if coefficient is significantly different from zero)
2. **R-squared** (top right section):
    - Proportion of variation in Y explained by X
    - Ranges from 0 to 1 (higher = better fit)
3. **F-statistic** (top right section):
    - Tests overall significance of the regression
    - Low p-value (Prob F-statistic) means the model is statistically significant

Let's extract and interpret the key coefficients.

In [47]:
```
# Extract key statistics
intercept = model.params['Intercept']
slope     = model.params['size']
r_squared = model.rsquared
n_obs     = int(model.nobs)

print("=" * 70)
print("KEY REGRESSION COEFFICIENTS")
print("=" * 70)
print(f"Intercept (β₀): ${intercept:,.2f}")
print(f"Slope (β₁):  ${slope:,.2f}")
print(f"R-squared: {r_squared:.4f} ({r_squared*100:.2f}%)")
print(f"Number of observations: {n_obs}")

print("\n" + "=" * 70)
print("ECONOMIC INTERPRETATION")
print("=" * 70)
print(f"📌 For every additional square foot of house size,")
print(f"   the sale price increases by approximately ${slope:,.2f}")
print(f"\n📌 The model explains {r_squared*100:.2f}% of the variation in house prices")
print(f"\n📌 The remaining {(1-r_squared)*100:.2f}% is due to other factors not included")
print(f"   (e.g., location, age, condition, neighborhood quality)")
```

```
======================================================================
KEY REGRESSION COEFFICIENTS
======================================================================
Intercept (β₀): $115,017.28
Slope (β₁):  $73.77
R-squared: 0.6175 (61.75%)
Number of observations: 29

======================================================================
ECONOMIC INTERPRETATION
======================================================================
📌 For every additional square foot of house size,
   the sale price increases by approximately $73.77

📌 The model explains 61.75% of the variation in house prices

📌 The remaining 38.25% is due to other factors not included
   (e.g., location, age, condition, neighborhood quality)
```

1.8 Visualizing the Fitted Line

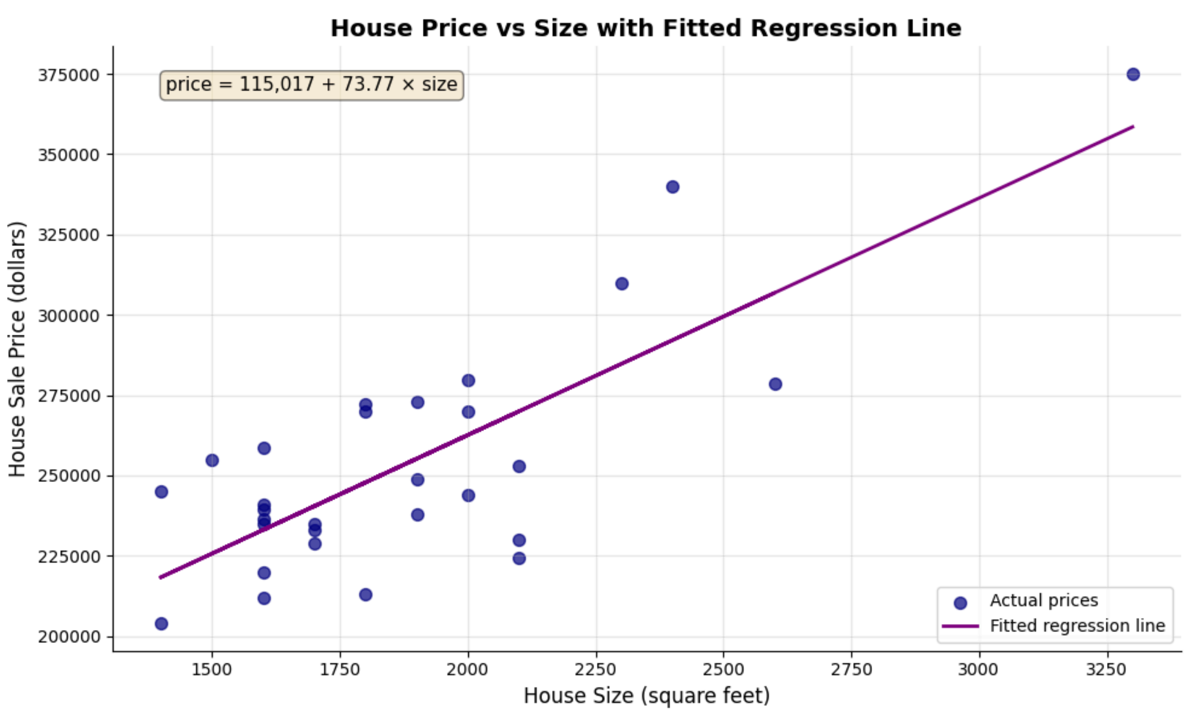
The **fitted regression line** represents our model's predictions. For any given house size, the line shows the predicted price according to our equation:

$$\hat{\text{price}} = \beta_0 + \beta_1 \times \text{size}$$

Let's overlay this fitted line on our scatter plot to see how well it captures the relationship.

In [48]:

```python
# Create scatter plot with fitted regression line
fig, ax = plt.subplots(figsize=(10, 6))

# Plot actual data points
ax.scatter(data_house['size'], data_house['price'],
           color='navy', s=50, label='Actual prices', alpha=0.7)

# Plot fitted regression line
ax.plot(data_house['size'], model.fittedvalues,
        color='purple', linewidth=2, label='Fitted regression line')

# Add equation to plot
equation_text = f'price = {intercept:,.0f} + {slope:.2f} × size'
ax.text(0.05, 0.95, equation_text,
        transform=ax.transAxes, fontsize=11,
        verticalalignment='top',
        bbox=dict(boxstyle='round', facecolor='wheat', alpha=0.5))

# Labels and formatting
ax.set_xlabel('House Size (square feet)', fontsize=12)
ax.set_ylabel('House Sale Price (dollars)', fontsize=12)
ax.set_title('House Price vs Size with Fitted Regression Line',
             fontsize=14, fontweight='bold')
ax.legend(loc='lower right', fontsize=10)
ax.spines[['top', 'right']].set_visible(False)
ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

print("\n📊 The purple line is our 'line of best fit'")
print("   It minimizes the sum of squared vertical distances from each point")
```




```
📊 The purple line is our 'line of best fit'
   It minimizes the sum of squared vertical distances from each point
```

Transition: Statistical output is only meaningful when translated into economic insights. Let's explore what our regression coefficients tell us about housing markets and the limitations of our analysis.

1.9 Economic Interpretation and Examples

Now that we've estimated the regression, let's think about what it means in economic terms.

Practical Implications:

Our estimated slope of approximately **$74 per square foot** means:

- A house that's 100 sq ft larger is predicted to sell for $74 \times 100 =$7,400 more
- A house that's 500 sq ft larger is predicted to sell for $74 \times 500 =$37,000 more

Making Predictions:

We can use our regression equation to predict prices for houses of different sizes. For example, for a 2,000 sq ft house:

$$\hat{\text{price}} = 115,952 + 74.03 \times 2000 = \$264,012$$

Important Caveats:

1. **This is association, not causation**: We can't conclude that adding square footage to a house will increase its value by $74/sq ft. Other factors (like quality of construction) might be correlated with size.
2. **Omitted variables**: Many other factors affect house prices (location, age, condition, amenities). Our simple model ignores these - we'll learn how to include them in later chapters.
3. **Sample-specific**: These results are from 29 houses in Davis, CA in 1999. The relationship might differ in other locations or time periods.
4. **Don't extrapolate too far**: Our data ranges from 1,400 to 3,300 sq ft. Predictions far outside this range (e.g., for a 10,000 sq ft house) may not be reliable.

> **Key Concept**: *Regression results must be interpreted with caution. Association does not imply causation, omitted variables can bias estimates, and predictions should not extrapolate beyond the range of the data.*

Key Takeaways

Statistical Methods and Data Types:

- Econometrics uses two main approaches: descriptive analysis (summarizing data) and statistical inference (drawing population conclusions from samples)
- Economic data are primarily continuous and numerical, though categorical and discrete data are also important
- Economics relies mainly on observational data, making causal inference more challenging than with experimental data
- The three data collection methods are cross-section (individuals at one time), time series (one individual over time), and panel data (individuals over time)
- Each data type requires different considerations for statistical inference, particularly when computing standard errors
- This textbook focuses on continuous numerical data and cross-section analysis as the foundation for more advanced methods

Regression Analysis and Interpretation:

- Regression analysis is the primary tool in econometrics, quantifying how outcome variables (y) vary with explanatory variables (x)
- The simple linear regression model has the form: $y = \beta_0 + \beta_1 x + \varepsilon$, where β_0 is the intercept and β_1 is the slope
- The slope coefficient measures association: how much y changes when x increases by one unit
- OLS (Ordinary Least Squares) finds the best-fitting line by minimizing the sum of squared prediction errors
- R-squared measures the proportion of variation in y explained by x, ranging from 0 to 1 (higher = better fit)
- Economic interpretation focuses on magnitude (size of effect), statistical significance, and practical importance

Practical Application:

- Our house price example: Each additional square foot is associated with a $73.77 increase in price (R^2 = 61.75%)
- Visualization is essential: scatter plots reveal the nature of relationships before fitting regression models
- Regression shows association, not causation—omitted variables and confounding factors require careful consideration
- Predictions should not extrapolate beyond the range of observed data
- Sample-specific results may not generalize to other locations, time periods, or populations

Python Tools and Workflow:

- `pandas` handles data loading, manipulation, and descriptive statistics
- `statsmodels.formula.api.ols()` estimates OLS regression models with R-style formula syntax
- `matplotlib` creates publication-quality scatter plots and visualizations
- Standard workflow: load data → explore descriptively → visualize → model → interpret → validate
- Random seeds ensure reproducibility of results across different runs

Prerequisites and Mathematical Background:

- Summation notation (Σ) expresses formulas concisely and appears throughout econometrics
- Calculus concepts (derivatives, rates of change) help understand marginal effects but are not essential
- Expected values (E[X]) define population parameters like means and variances
- "Learning-by-doing" is the most effective approach: practice with real data and software is essential for mastery

---

Next Steps:

- **Chapter 2**: Univariate data summary (describing single variables)
- **Chapter 3**: The sample mean and sampling distributions
- **Chapter 4**: Statistical inference for the mean (confidence intervals, hypothesis tests)
- **Chapters 5-7**: Deep dive into bivariate regression (extending what we learned here)

You have now mastered: ✓ Loading and exploring economic data in Python ✓ Creating scatter plots to visualize relationships ✓ Estimating simple linear regression models with OLS ✓ Interpreting regression coefficients economically ✓ Understanding the limitations of regression analysis

These foundational concepts are the building blocks for all of econometrics. Everything that follows builds on this introduction!

Practice Exercises

Test your understanding of regression analysis with these exercises:

Exercise 1: Conceptual understanding

- (a) What is the difference between descriptive analysis and statistical inference?
- (b) Why do economists primarily use observational data rather than experimental data?
- (c) Name the three main types of data collection methods and give an example of each.

Exercise 2: Data types

- Classify each of the following as continuous numerical, discrete numerical, or categorical:
    - (a) Annual household income
    - (b) Number of children in a family
    - (c) Employment status (employed, unemployed, not in labor force)
    - (d) Temperature in degrees Celsius

Exercise 3: Regression interpretation

- Suppose you estimate: earnings = 20,000 + 5,000 × education
    - (a) Interpret the intercept coefficient
    - (b) Interpret the slope coefficient
    - (c) Predict earnings for someone with 16 years of education
    - (d) What is the predicted difference in earnings between someone with 12 vs. 16 years of education?

Exercise 4: Using our house price model

- Using the regression equation: price = 115,017 + 73.77 × size
    - (a) Predict the price for a 1,800 sq ft house

- (b) Predict the price for a 2,500 sq ft house
    - (c) What is the predicted price difference between these two houses?
    - (d) Is the intercept economically meaningful in this context? Why or why not?

Exercise 5: Critical thinking about causation

- Our regression shows larger houses have higher prices. Does this mean:
    - (a) Adding square footage to a house will increase its value by $73.77 per sq ft?
    - (b) What other factors might be correlated with both house size and price?
    - (c) How would you design a study to establish a causal relationship?

Exercise 6: R-squared interpretation

- Our model has $R^2 = 0.6175$ (61.75%)
    - (a) What does this number tell us about our model?
    - (b) What factors might explain the remaining 38.25% of variation in prices?
    - (c) Would $R^2 = 1.0$ be realistic for real-world economic data? Why or why not?

Exercise 7: Summation notation

- Calculate: $\sum_{i=1}^{4}(3+2i)$
- Show all steps in your calculation

Exercise 8: Python practice

- Load the house dataset and:
    - (a) Calculate the correlation between price and bedrooms
    - (b) Create a scatter plot of price vs. bedrooms
    - (c) Estimate the regression: price ~ bedrooms
    - (d) Compare the R^2 to our size regression. Which predictor is better?

1.11 Case Studies

Now let's apply what you've learned to real economic research! In this section, you'll explore data from actual published studies, using the same tools and techniques from this chapter.

Why case studies matter:

- See how regression analysis is used in real research
- Practice applying Chapter 1 tools to authentic data
- Develop intuition for economic relationships
- Bridge the gap between textbook examples and research practice

Case Study 1: Economic Convergence Clubs

Research Question: Do countries converge toward similar levels of economic development, or do they form distinct "convergence clubs" with different trajectories?

Background: Traditional economic theory suggests that poor countries should grow faster than rich countries, eventually "catching up" in terms of income and productivity. This is called the convergence hypothesis. However, empirical evidence shows a more complex picture: countries may form distinct groups (clubs) that converge toward different long-run equilibrium levels rather than a single global level.

This Research (Mendez, 2020): Uses modern econometric methods to identify convergence clubs in labor productivity across countries. The analysis examines whether countries follow one common development path or multiple distinct paths, and what factors (capital accumulation, technology, institutions) drive these patterns.

The Data: Panel dataset tracking multiple countries over several years, with 26 variables including:

- **Output measures**: GDP, GDP per capita
- **Productivity**: Labor productivity, total factor productivity (TFP)
- **Capital**: Physical capital stock, capital per worker
- **Human capital**: Years of schooling, human capital index
- **Classifications**: Country codes, regions, income groups

Your Task: Use the descriptive analysis and regression tools from Chapter 1 to explore patterns in the convergence clubs data. You'll investigate productivity gaps, visualize relationships, and begin to understand why some countries develop differently than others.

> **Key Concept: Economic Convergence and Productivity Drivers**
>
> **Beta convergence** *refers to the hypothesis that poor countries will grow faster than rich countries, eventually "catching up" in terms of income and productivity. However, evidence suggests countries may form distinct* **convergence clubs***—groups that converge toward different long-run equilibrium levels rather than a single global level.*
>
> *Labor productivity (output per worker) depends on* **capital accumulation** *(capital per worker) and* **aggregate efficiency** *(total factor productivity or TFP). The regression of productivity on capital captures this association, allowing us to quantify how much of cross-country productivity differences are explained by capital versus efficiency factors.*

Load the Convergence Clubs Data

Let's load two datasets:

1. **Main dataset** (`dat.csv`): Country-year panel data with economic variables
2. **Data dictionary** (`dat-definitions.csv`): Explains what each variable means

The data uses a **multi-index** structure with (country, year) pairs, allowing us to track each country over time.

In [49]:

```
# Import data with sorted multi-index
df1 = pd.read_csv(
    "https://raw.githubusercontent.com/quarcs-lab/mendez2020-convergence-clubs-code-
data/master/assets/dat.csv",
    index_col=["country", "year"]
).sort_index()

# Import data dictionary
df2 = pd.read_csv(
    "https://raw.githubusercontent.com/quarcs-lab/mendez2020-convergence-clubs-code-
data/master/assets/dat-definitions.csv"
)

# Display basic information
print("=" * 70)
print("CONVERGENCE CLUBS DATASET")
print("=" * 70)
print(f"Dataset shape: {df1.shape[0]} observations (country-year pairs), {df1.shape[1]}
variables")
print(f"\nCountries: {len(df1.index.get_level_values('country').unique())} unique
countries")
print(f"Years: {df1.index.get_level_values('year').min()} to
{df1.index.get_level_values('year').max()}")

print("\n" + "=" * 70)
print("FIRST 5 OBSERVATIONS")
print("=" * 70)
print(df1.head(5))
```

```
===========================================================================
CONVERGENCE CLUBS DATASET
===========================================================================
Dataset shape: 2700 observations (country-year pairs), 27 variables

Countries: 108 unique countries
Years: 1990 to 2014

===========================================================================
FIRST 5 OBSERVATIONS
===========================================================================
              id          Y      K       pop         L         s  alpha_it  \
country year
Albania 1990   1  12449.999  31217  3.281453  1.250096  8.497386       NaN
        1991   1  11310.000  30082  3.275438  1.243719  8.442703       NaN
        1992   1  10122.000  28968  3.240613  0.993492  8.388020       NaN
        1993   1  11636.000  29010  3.189623  0.935928  8.333337       NaN
        1994   1  13120.000  29634  3.140634  1.008687  8.278653       NaN

                  GDPpc          lp         h  ...  log_h_raw  log_tfp_raw  \
country year                                   ...
Albania 1990  3794.0508   9959.2344  3.165140  ...   1.152197     5.560511
        1991  3452.9731   9093.6943  3.150347  ...   1.147513     5.479933
        1992  3123.4832  10188.3060  3.135588  ...   1.142817     5.457089
        1993  3648.0798  12432.5870  3.120863  ...   1.138110     5.617254
        1994  4177.5005  13007.0080  3.106171  ...   1.133391     5.707116

              log_GDPpc    log_lp    log_ky     log_h   log_tfp  isocode  \
country year
Albania 1990   8.130327  9.209681  0.880770  1.125790  5.541879      ALB
        1991   8.170837  9.268303  0.869605  1.133772  5.574306      ALB
        1992   8.211625  9.326916  0.858536  1.141821  5.606780      ALB
        1993   8.252907  9.385130  0.847930  1.150036  5.639110      ALB
        1994   8.294488  9.442308  0.838639  1.158521  5.670734      ALB

              hi1990  region
country year
Albania 1990      no  Europe
        1991      no  Europe
        1992      no  Europe
        1993      no  Europe
        1994      no  Europe

[5 rows x 27 columns]
```

In [50]:

```python
print("\n" + "=" * 75)
print("VARIABLE DEFINITIONS")
print("=" * 75)
print(df2)
```

```
============================================================================
VARIABLE DEFINITIONS
============================================================================
         var_name                                          var_def     type
0         country              Standardized country name (from PWT)     cs_id
1            year                                             Year     ts_id
2               Y                                              GDP   numeric
3               K                                 Physical Capital   numeric
4             pop                                       Population   numeric
5               L                                      Labor Force   numeric
6               s                               Years of Schooling   numeric
7        alpha_it                            Variable Capital Share   numeric
8           GDPpc                                   GDP per capita   numeric
9              lp                               Labor Productivity   numeric
10              h                              Human Capital Index   numeric
11             kl                               Capital per Worker   numeric
12             kp                             Capital Productivity   numeric
13             ky                             Capital-Output Ratio   numeric
14            TFP                             Aggregate Efficiency   numeric
15   log_GDPpc_raw                           Log of GDP per capita   numeric
16      log_lp_raw                       Log of Labor Productivity   numeric
17      log_ky_raw                     Log of Capital-Output Ratio   numeric
18       log_h_raw                             Log of Human Capital   numeric
19     log_tfp_raw                  Log of Total Factor Productivity   numeric
20       log_GDPpc                Trend (HP400) of log of GDP per capita   numeric
21          log_lp            Trend (HP400) of log of labor productivity   numeric
22          log_ky          Trend (HP400) of log of Capital-Output Ratio   numeric
23           log_h                 Trend (HP400) of log of Human Capital   numeric
24         log_tfp          Trend (HP400) of log of Aggregate Efficiency   numeric
25          region           Regional group (Classification of the UN)    factor
26          hi1990   High income country (as of 1990, World Bank cl...    factor
27         isocode                               ISO code from the PWT9.0    factor
```

Task 1: Data Exploration (Guided)

Objective: Understand the dataset structure and available variables.

Instructions:

1. Examine the output above to understand the multi-index (country, year) structure
2. Review the data dictionary to identify key productivity variables
3. Check for missing values in key variables
4. Identify the variable names you'll use for subsequent analyses

Key variables to focus on (check exact names in the data dictionary):

- Labor productivity variables
- Capital per worker variables
- GDP per capita measures
- Country classification variables (region, income group)

Run the code above and study the output. What patterns do you notice? How many time periods does each country have?

In [51]:

```
# Your code here: Explore the dataset structure
#
# Suggested explorations:
# 1. Check column names: df1.columns.tolist()
# 2. Check for missing values: df1.isnull().sum()
# 3. Examine a specific country's data
# 4. Count observations per country

# Example: Examine USA's data
# df1.loc['USA']
```

Task 2: Descriptive Statistics (Semi-guided)

Objective: Generate summary statistics for key productivity variables.

Instructions:

1. Select 3-4 key variables related to productivity and capital
2. Generate descriptive statistics (mean, median, std, min, max)
3. Identify countries with highest and lowest productivity levels
4. Calculate the productivity gap between top and bottom performers

Apply what you learned in sections 1.4: Use `.describe()` method like we did with the house price data.

Hint: You'll need to identify the exact variable names from the data dictionary. Look for variables measuring labor productivity, GDP per capita, or capital per worker.

In [52]:

```
# Your code here: Generate descriptive statistics
#
# Steps:
# 1. Identify variable names from data dictionary (check df2)
# 2. Select key variables from df1
# 3. Generate summary statistics using .describe()
# 4. Find countries with max/min values using .idxmax() and .idxmin()
# 5. Calculate gaps between high and low performers

# Example structure:
# key_vars = ['variable1', 'variable2', 'variable3']  # Replace with actual names
# df1[key_vars].describe()
```

> **Key Concept**: *Panel data combines cross-section and time series dimensions, tracking multiple entities (countries) over multiple time periods (years). This structure allows us to study both differences between countries (cross-sectional variation) and changes within countries over time (time series variation). The data is indexed by (country, year) pairs.*

Task 3: Visualizing Productivity Patterns (Semi-guided)

Objective: Create scatter plots to visualize productivity relationships.

Instructions:

1. Create a scatter plot comparing two productivity-related variables
2. Add appropriate axis labels and a descriptive title
3. Optionally: Color-code points by region or income group
4. Interpret the pattern you observe

Apply what you learned in section 1.5: Use matplotlib to create scatter plots like the house price visualization.

Suggested relationships to explore:

- GDP per capita vs. labor productivity
- Capital per worker vs. labor productivity
- Human capital vs. GDP per capita

In [53]:

```
# Your code here: Create scatter plot
#
# Steps:
# 1. Prepare data (select variables, remove missing values)
# 2. Create figure and axis: fig, ax = plt.subplots(figsize=(10, 6))
# 3. Create scatter plot: ax.scatter(x, y, ...)
# 4. Add labels, title, and formatting
# 5. Display and interpret

# Example structure:
# plot_data = df1[['var_x', 'var_y']].dropna()
# fig, ax = plt.subplots(figsize=(10, 6))
# ax.scatter(plot_data['var_x'], plot_data['var_y'], alpha=0.6)
# ax.set_xlabel('Variable X')
# ax.set_ylabel('Variable Y')
# plt.show()

# What pattern do you observe? Positive or negative relationship?
```

Task 4: Time Series Exploration (More Independent)

Objective: Examine productivity trends over time for specific countries.

Instructions:

1. Select 2-3 countries of interest (e.g., USA, China, India, Japan, or countries from your region)
2. Plot labor productivity over time for each country
3. Compare their trajectories: Which countries are growing faster?

4. Calculate the average annual growth rate (optional: percentage change per year)

Hint: Remember that panel data is indexed by (country, year). Use `.loc[country]` to filter data for a specific country.

Questions to answer:

- Are productivity levels converging (getting closer) or diverging (spreading apart)?
- Which country experienced the fastest productivity growth?
- Do you see evidence of convergence clubs (groups following similar paths)?

In [54]:

```
# Your code here: Time series plots
#
# Steps:
# 1. Select countries (e.g., countries = ['USA', 'CHN', 'JPN'])
# 2. For each country, extract time series data
# 3. Create line plot over time
# 4. Compare trajectories

# Example structure:
# countries = ['USA', 'CHN', 'IND']  # Adjust based on actual country codes
# fig, ax = plt.subplots(figsize=(10, 6))
#
# for country in countries:
#     country_data = df1.loc[country]
#     ax.plot(country_data.index, country_data['productivity_var'], label=country)
#
# ax.set_xlabel('Year')
# ax.set_ylabel('Labor Productivity')
# ax.legend()
# plt.show()
```

Task 5: Simple Regression Analysis (INDEPENDENT)

Objective: Estimate the relationship between capital and productivity.

Research Question: Does higher capital per worker lead to higher labor productivity?

Instructions:

1. Prepare regression data (select variables, remove missing values)
2. Estimate OLS regression: `labor_productivity ~ capital_per_worker`
3. Display the regression summary
4. Interpret the slope coefficient economically
5. Report the R-squared value
6. **Critical thinking**: Discuss whether this is association or causation

Apply what you learned in sections 1.6-1.7: Use `ols()` from statsmodels and interpret coefficients.

Important questions:

- What does the slope coefficient tell us?
- Could there be omitted variables affecting both capital and productivity?
- Could reverse causality be a concern (does higher productivity lead to more capital accumulation)?

In [55]:

```
# Your code here: Regression analysis
#
# Steps:
# 1. Prepare data: reg_data = df1[['productivity_var', 'capital_var']].dropna()
# 2. Reset index if needed: reg_data = reg_data.reset_index()
# 3. Estimate regression: model = ols('productivity_var ~ capital_var',
data=reg_data).fit()
# 4. Display summary: print(model.summary())
# 5. Extract and interpret coefficients

# Example structure:
# from statsmodels.formula.api import ols
#
# reg_data = df1[['var_y', 'var_x']].dropna().reset_index()
# model = ols('var_y ~ var_x', data=reg_data).fit()
# print(model.summary())
#
# Interpretation:
# Slope = ___: For every unit increase in capital per worker,
#               labor productivity increases by ___ units
# R² = ___: Capital explains ___% of variation in productivity
```

Task 6: Comparative Analysis (Independent)

Objective: Compare productivity patterns between country groups.

Research Question: Does the capital-productivity relationship differ between high-income and developing countries?

Instructions:

1. If the data has an income group variable, group countries by income level
2. Calculate average productivity and capital for each group
3. Create comparative scatter plots (one color per group)
4. (Advanced) Run separate regressions for each group
5. Compare the slope coefficients: Is the relationship stronger in one group?

This extends Chapter 1 concepts: You're using grouping and comparative analysis to see if relationships vary across subsamples.

Questions to explore:

- Do high-income countries have uniformly higher productivity?

- Is the capital-productivity relationship steeper in developing countries?
- What might explain differences between groups?

In [56]:
```
# Your code here: Comparative analysis
#
# Steps:
# 1. Identify grouping variable (income level, region, etc.)
# 2. Group data: df1.groupby('group_var').mean()
# 3. Create comparative visualizations
# 4. Run regressions by group (optional)

# Example structure for comparative scatter plot:
# fig, ax = plt.subplots(figsize=(10, 6))
#
# for group in df1['group_var'].unique():
#     group_data = df1[df1['group_var'] == group]
#     ax.scatter(group_data['var_x'], group_data['var_y'], label=group, alpha=0.6)
#
# ax.set_xlabel('Capital per Worker')
# ax.set_ylabel('Labor Productivity')
# ax.legend()
# plt.show()

# Advanced: Separate regressions
# for group in groups:
#     group_data = df1[df1['group_var'] == group].dropna()
#     model = ols('var_y ~ var_x', data=group_data).fit()
#     print(f"\n{group}: Slope = {model.params['var_x']:.4f}, R² = {model.rsquared:.4f}")
```

What You've Learned from This Case Study

Through this hands-on exploration of economic convergence data, you've applied all the core Chapter 1 tools:

- **Data loading and exploration**: Worked with real panel data from research
- **Descriptive statistics**: Summarized productivity patterns across countries
- **Visualization**: Created scatter plots and time series to reveal relationships
- **Regression analysis**: Quantified the capital-productivity relationship
- **Critical thinking**: Distinguished association from causation
- **Comparative analysis**: Explored differences between country groups

Connection to the research: The patterns you've discovered—productivity gaps, the role of capital, differences between country groups—are the empirical motivation for the convergence clubs analysis in Mendez (2020). The full research uses advanced methods (covered in later chapters) to formally identify clubs and test convergence hypotheses.

Looking ahead:

- **Chapter 2** will teach you more sophisticated descriptive analysis for univariate data

- **Chapter 3-4** cover statistical inference, allowing you to test hypotheses formally
- **Chapter 5-9** extend regression analysis to multiple predictors and transformations
- **Chapter 10-17** introduce advanced methods like panel data regression—perfect for convergence clubs!

Great work! You've completed Chapter 1 and applied your new skills to real economic research. Continue to Chapter 2 to learn more about data summary and distributions.