Extension to Multiple Regression

- The simple model with only `AI_Adoption` is useful to show the baseline relationship.
- However, firms differ along many dimensions (size, sector, M&A activity, international exposure).
- To control for these factors, we extend the model to a **multiple linear regression**:

$$\text{Layoffs}_i = \beta_0 + \beta_1 \text{AI\_Adoption}_i + \beta_2 \text{RevenueGrowth}_i + \beta_3 \text{Merger}_i + \beta_4 \text{Ma}$$

- Aim: check whether the AI effect remains strong after adding controls.

New Explanatory Variables

- **RevenueGrowth**: yearly percentage change in firm turnover (proxy for business dynamics)
- **Merger**: binary indicator $= 1$ if the firm was involved in a merger/acquisition in the last year
- **MarketCap**: market capitalization (firm size)
- **Taxes**: total taxes paid (another size/profitability proxy)
- **FDI**: number of foreign direct investments
- **Sector dummies**: Tech / Manufacturing (Finance as baseline)

All these variables were generated synthetically to obtain a richer cross-section of 100 firms.

Data Generation in Python

```
# Synthetic dataset with 100 firms
n = 100
ai_adoption     = np.random.uniform(0, 100, n)
revenue_growth = np.random.normal(5, 10, n)
merger          = np.random.binomial(1, 0.2, n)
market_cap      = np.random.lognormal(mean=2, sigma=0.7, size=n)
taxes           = np.random.normal(50, 15, n)
fdi             = np.random.poisson(3, n)
sector          = np.random.choice(
    ["Tech", "Finance", "Manufacturing"], size=n
)

# Layoffs depends on AI + other covariates + noise
layoffs = (20 + 1.4*ai_adoption - 0.3*revenue_growth
           + 10*merger + 0.8*market_cap + 0.2*taxes
           + 3*fdi + np.random.normal(0, 15, n))
```

We then converted Sector into dummies with `pd.get_dummies(...,` `drop_first=True)`.

Full OLS Specification

```python
# Build DataFrame
df = pd.DataFrame({
    "Layoffs": layoffs,
    "AI_Adoption": ai_adoption,
    "RevenueGrowth": revenue_growth,
    "Merger": merger,
    "MarketCap": market_cap,
    "Taxes": taxes,
    "FDI": fdi,
    "Sector": sector
})

# Convert sector to dummies
df = pd.get_dummies(df, columns=["Sector"], drop_first=True)

y = df["Layoffs"]
X = df.drop(columns=["Layoffs"]).astype(float)
X = sm.add_constant(X)

model_full = sm.OLS(y, X).fit()
print(model_full.summary())
```

This is our **full model**, including all candidate regressors.

Full Model: Main Findings

- **AI_Adoption** remains highly significant ($p < 0.001$): the original relationship is robust.
- **Merger**, **MarketCap**, **Taxes**, and **FDI** are also statistically significant.
- **RevenueGrowth** and the **sector dummies** show high p-values ($p > 0.1$) and are not individually significant.
- $R^2 \approx 0.93$ and Adjusted $R^2 \approx 0.926$: the model explains most of the cross-sectional variation in layoffs.

Model Reduction and Comparison

```python
# keep only significant regressors
X_red = df[[
    "AI_Adoption", "Merger", "MarketCap", "Taxes", "FDI"
]]
X_red = sm.add_constant(X_red)

model_red = sm.OLS(y, X_red).fit()

print(model_red.summary())
```

We compare the two models using AIC, BIC, adjusted R^2 and an F-test for nested models. The reduced model has slightly lower AIC/BIC and almost the same adjusted R^2, so it is preferred for parsimony.

Interpretation of the Multiple Regression

- Even after controlling for size, M&A and international activity, **firms with higher AI adoption still lay off more workers**.
- Some controls capture "structural" reasons for layoffs (e.g. M&A $\Rightarrow$ restructuring).
- Insignificant variables were dropped to avoid overfitting and to present a cleaner specification.
- This mirrors standard econometric practice: start from a rich model, then select the variables that actually explain the outcome.