ECON 0150 | Economic Data Analysis

The economist's data analysis pipeline.

Part 4.4 | OLS Assumptions; Multiple Regression

OLS Assumptions

Our test results are only valid when the model assumptions are valid.

1. ***Linearity**: The relationship between X and Y is linear*

2. ***Independence**: Observations are independent from each other*

3. ***Homoskedasticity**: Equal error variance across all values of X*

4. ***Normality**: Errors are normally distributed*

Model Diagnostics: Why Check Assumptions?

Assumption violations affect our inferences

If assumptions are violated:

- *Coefficient estimates may be biased*
- *Standard errors may be wrong*
- *p-values may be misleading*
- *Predictions may be unreliable*

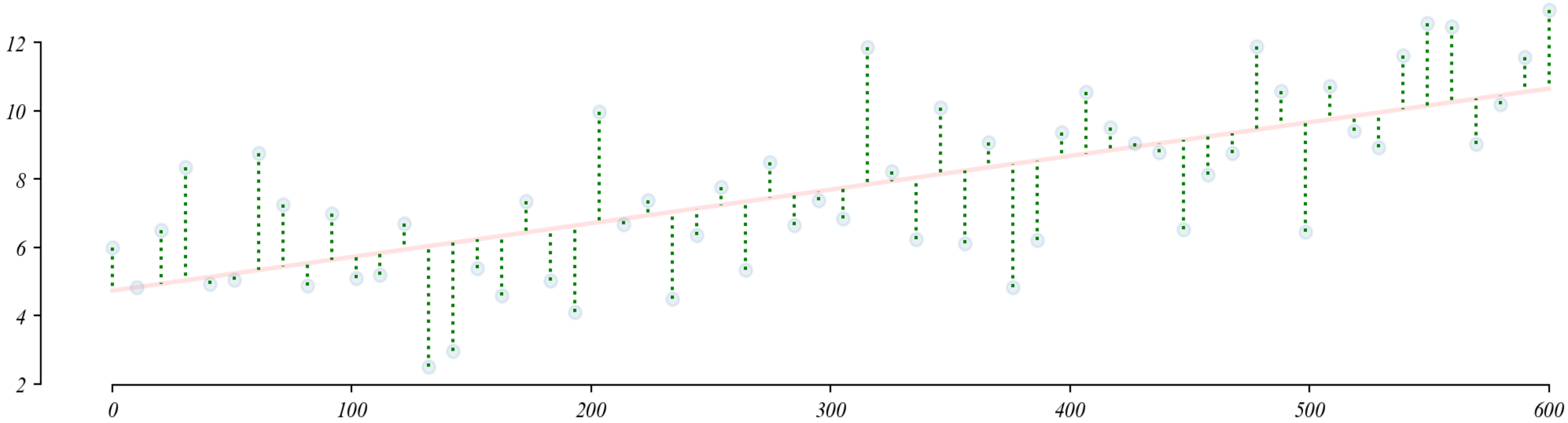
> to test whether the model is ‘specified’, we can calculate the residuals and the model predictions

Example: Education and Income

Is income higher for those more highly educated?

Model Residuals

… we can directly examine the error of the model.

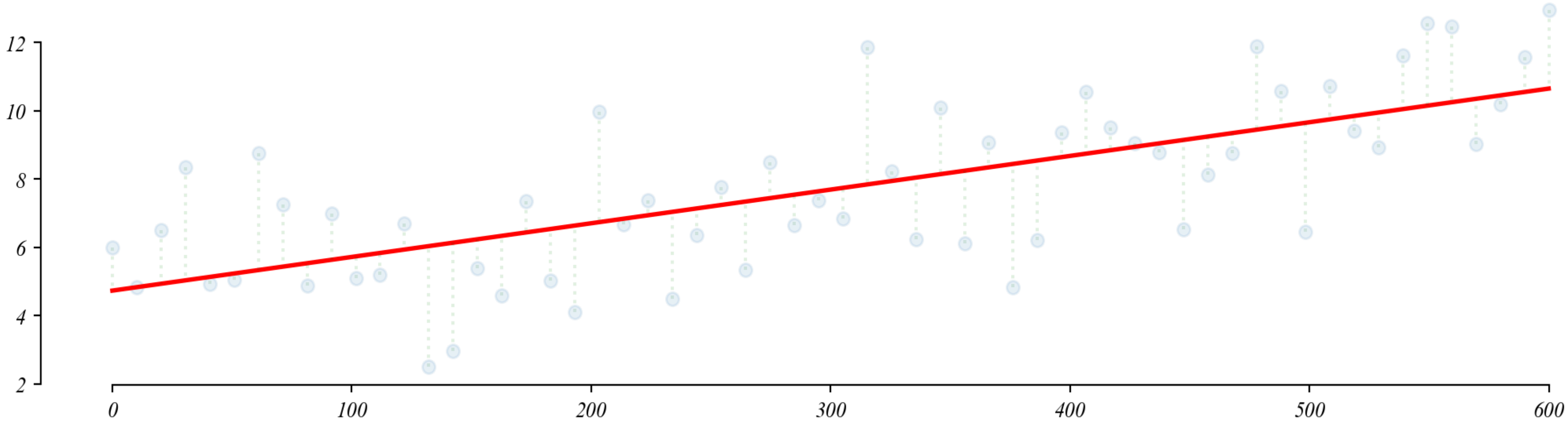



```
# Calculate residuals
residuals = model.resid
residuals.hist()
```

> this is ε

Model Predictions

... we can directly examine the predictions of the model.




```
# Calculate predictions
predictions = model.predict()
predictions.hist()
```

> *this is $\hat{y}$, the model prediction*

Residual Plot

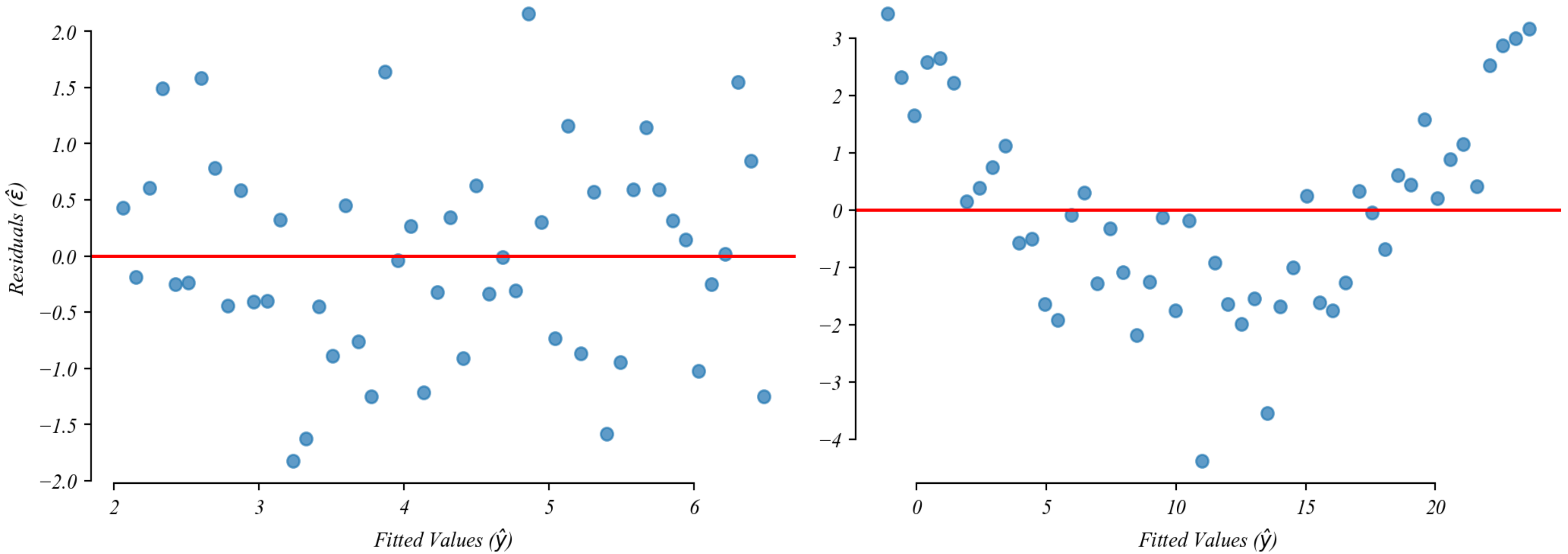
... we can directly observe the error according to the model estimates.

```
plt.scatter(predictions, residuals)
```

Assumption 1: Checking for Linearity

The error term should be unrelated to the fitted value.

> *which one of these figures shows linearity?*

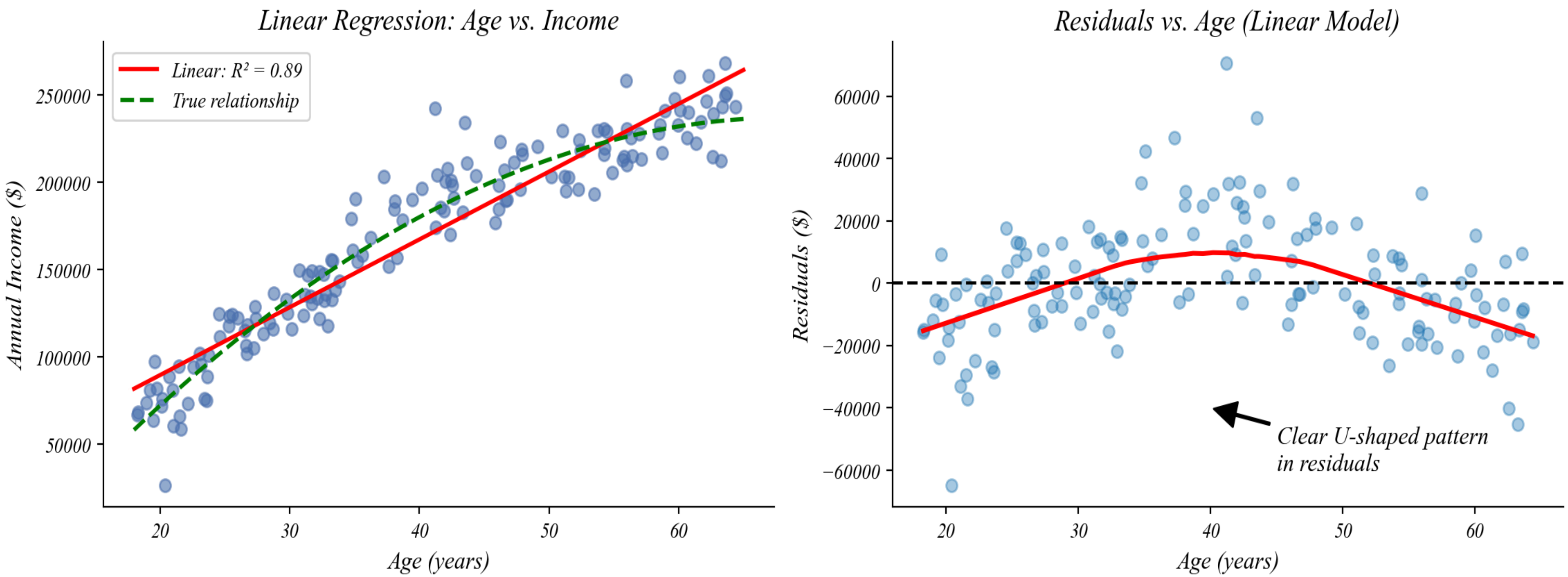


> *the left one is what we want to see*

> *residual plots should show that the model is equally wrong everywhere*

Non-Linear Relationships

A non-linear relationship will produce non-linear residuals.



> *sometimes relationships aren't linear*

> *linear model misses curvature, leading to systematic errors*

> *check your residuals*

Handling Non-Linear Relationships

Transform variables to become linear

> *here, adding a squared term captures the curvature in our data*

$$\text{income} = \beta_0 + \beta_1 \text{age} + \beta_2 \text{age}^2 + \varepsilon$$

instead of

$$\text{income} = \beta_0 + \beta_1 \text{age} + \varepsilon$$

```
df['age_squared'] = df['age']**2
quadratic_model = smf.ols('income ~ age + age_squared', data=df).fit()
```

> *coefficient interpretations change:*

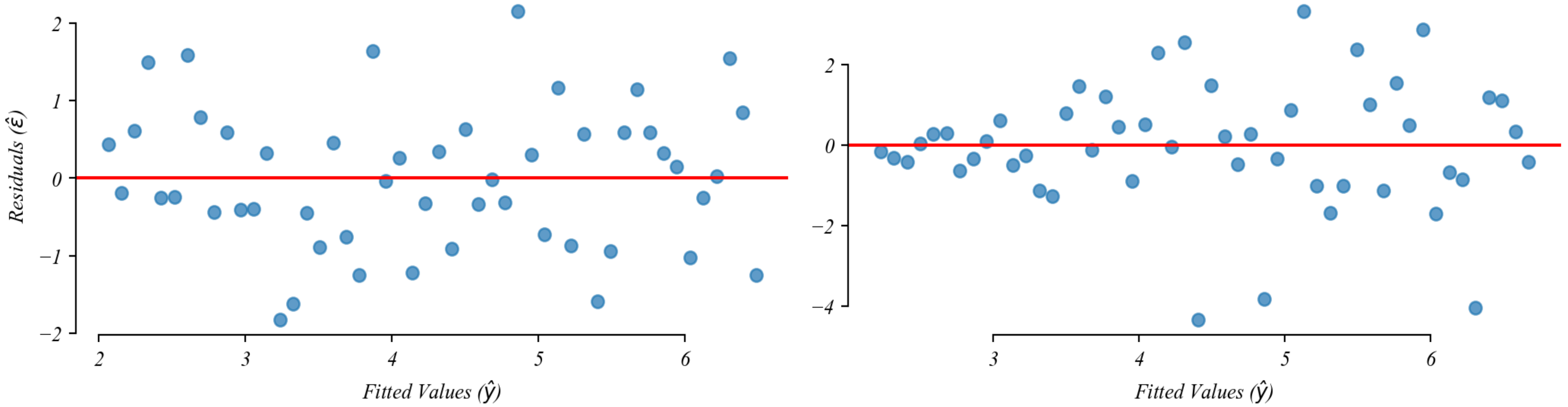
- β_1 = *effect of age when age = 0 (not very meaningful here)*
- β_2 = *how the effect of age changes as age increases*

> *other common transformations: log(y) ~ x or y ~ log(x) or log(y) ~ log(x)*

Assumption 3: Homoskedasticity

Residuals should be spread out the same everywhere.

> *which one of these figures shows homoskedasticity?*



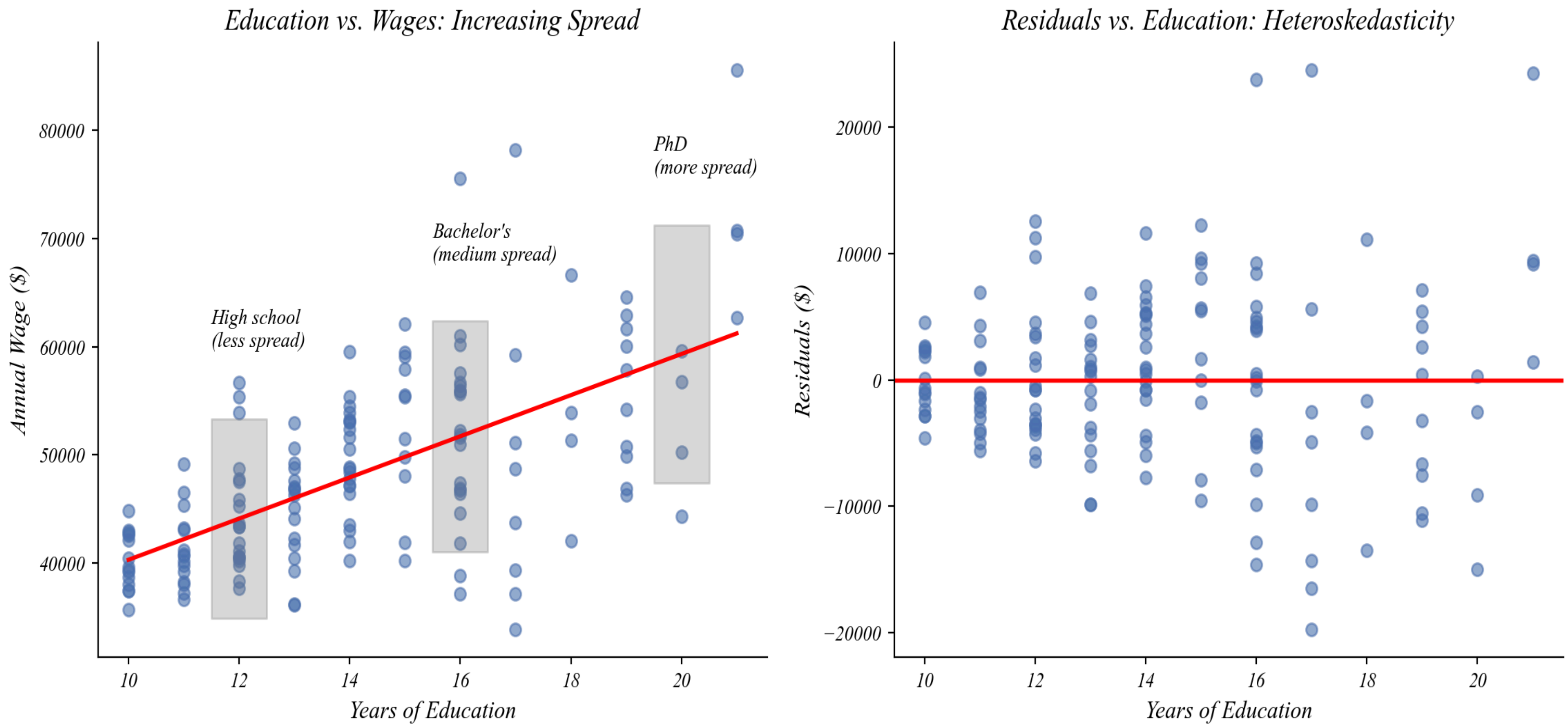
> *the left figure shows constant variability (homoskedasticity)*

> *the right one has increasing variability (heteroskedasticity)*

> *residual plots should show that the model is equally wrong everywhere*

Heteroskedasticity

When the spread of residuals changes across values of X

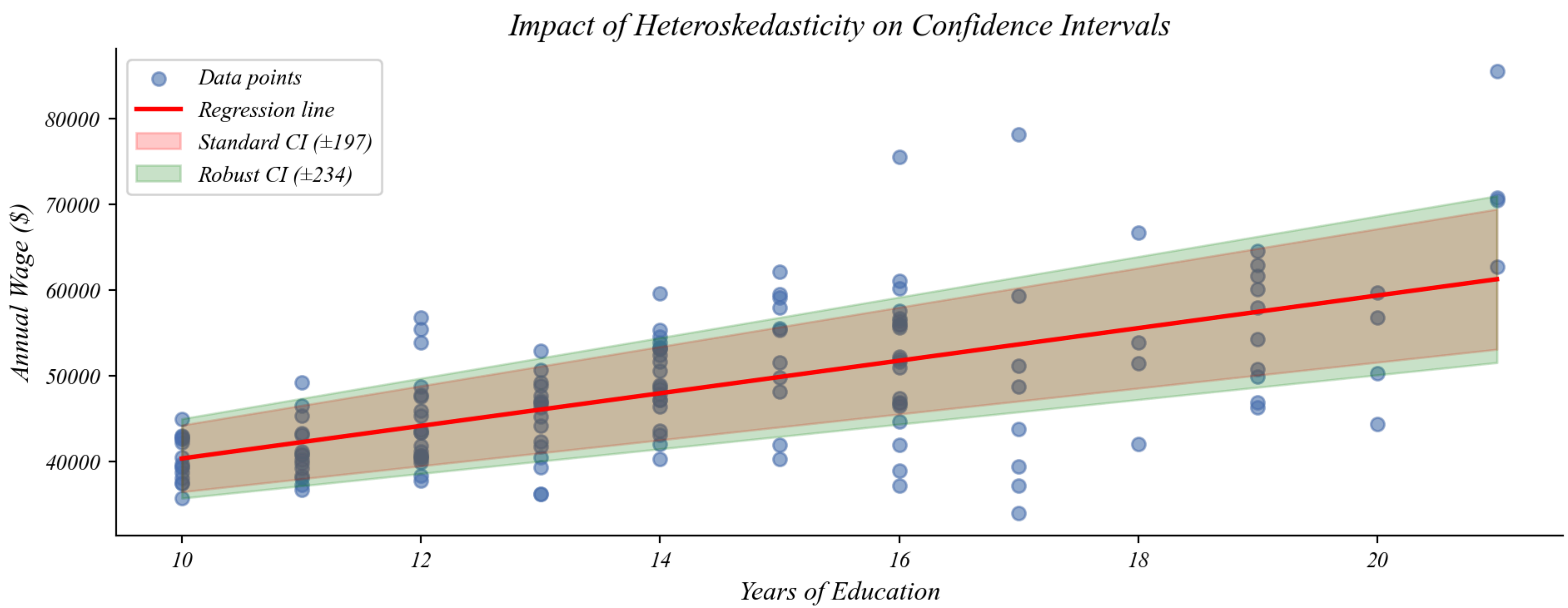


> notice how the spread of points increases with more education

> PhD wages vary more than high school wages

Heteroskedasticity

It affects how we measure uncertainty in our estimates



> *standard methods assume constant spread (homoskedasticity)*

> *like using the wrong ruler to measure uncertainty*

> *with heteroskedasticity, we need robust standard errors*

> *these adjust for the changing spread in our data*

Handling Heteroskedasticity

Robust standard errors give more accurate measures of uncertainty

```
# Fit the model with robust standard errors (HC3: heteroskedastic-constant)
robust_model = smf.ols('wages ~ education', data=df).fit(cov_type='HC3')
```

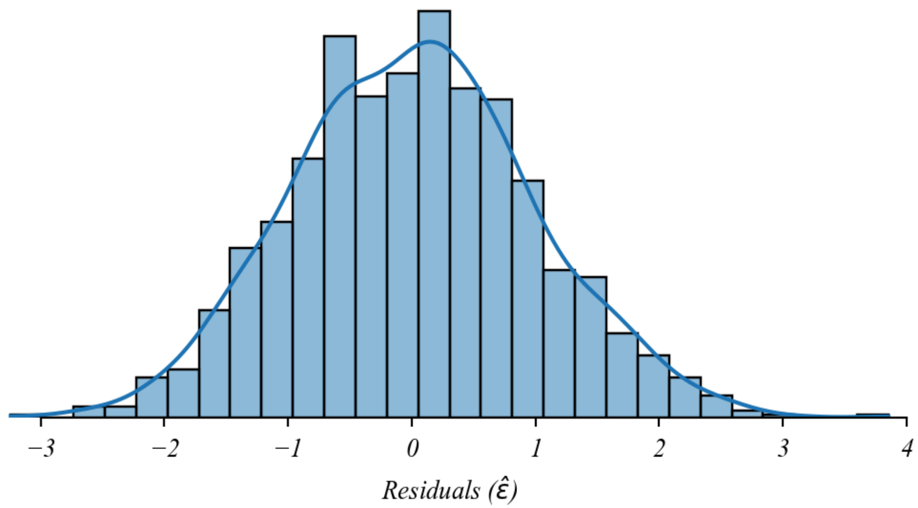
> *robust standard errors give more accurate confidence intervals*

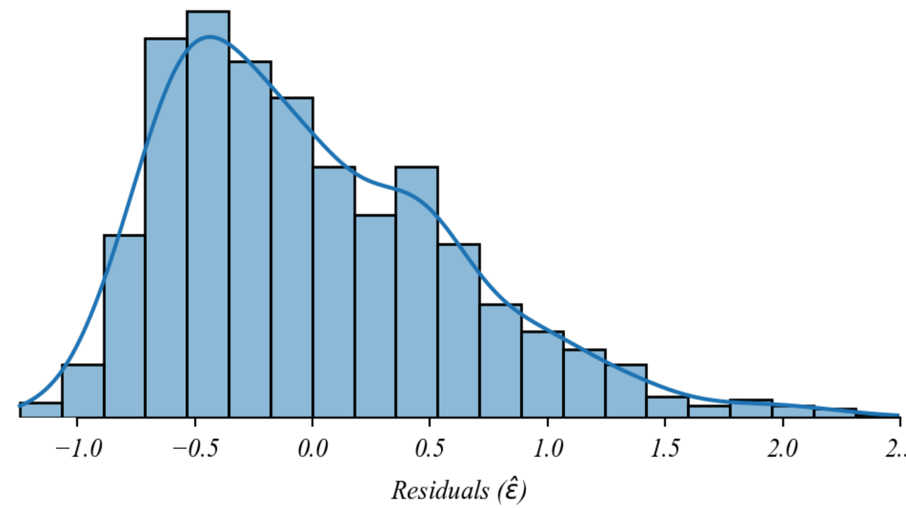
> *and more reliable hypothesis tests*

> *especially important when heteroskedasticity is pronounced*

Assumption 4: Normality

Residuals should be normally distributed





> *left shows a nice bell shape (roughly normally distributed)*

> *right shows a skewed distribution (not normally distributed)*

> *by the CLT we can still use regression without this if the sample is large*

Multiple Regression

Wages depend on more than just education

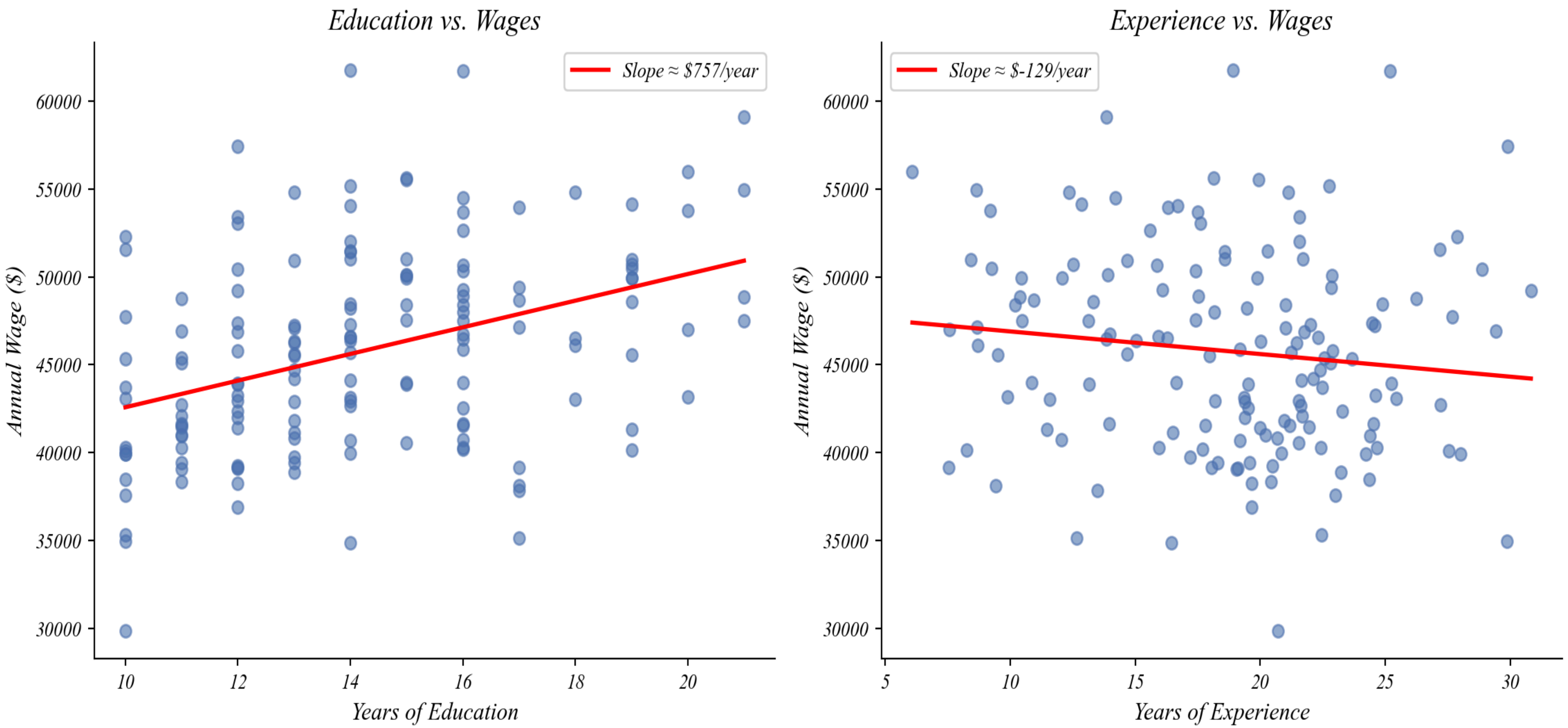
Wages also depend on:

- *Experience*
- *Industry*
- *Location*
- *And many other factors*

> how can we handle multiple relationships at once?

Modeling Relationships Separately

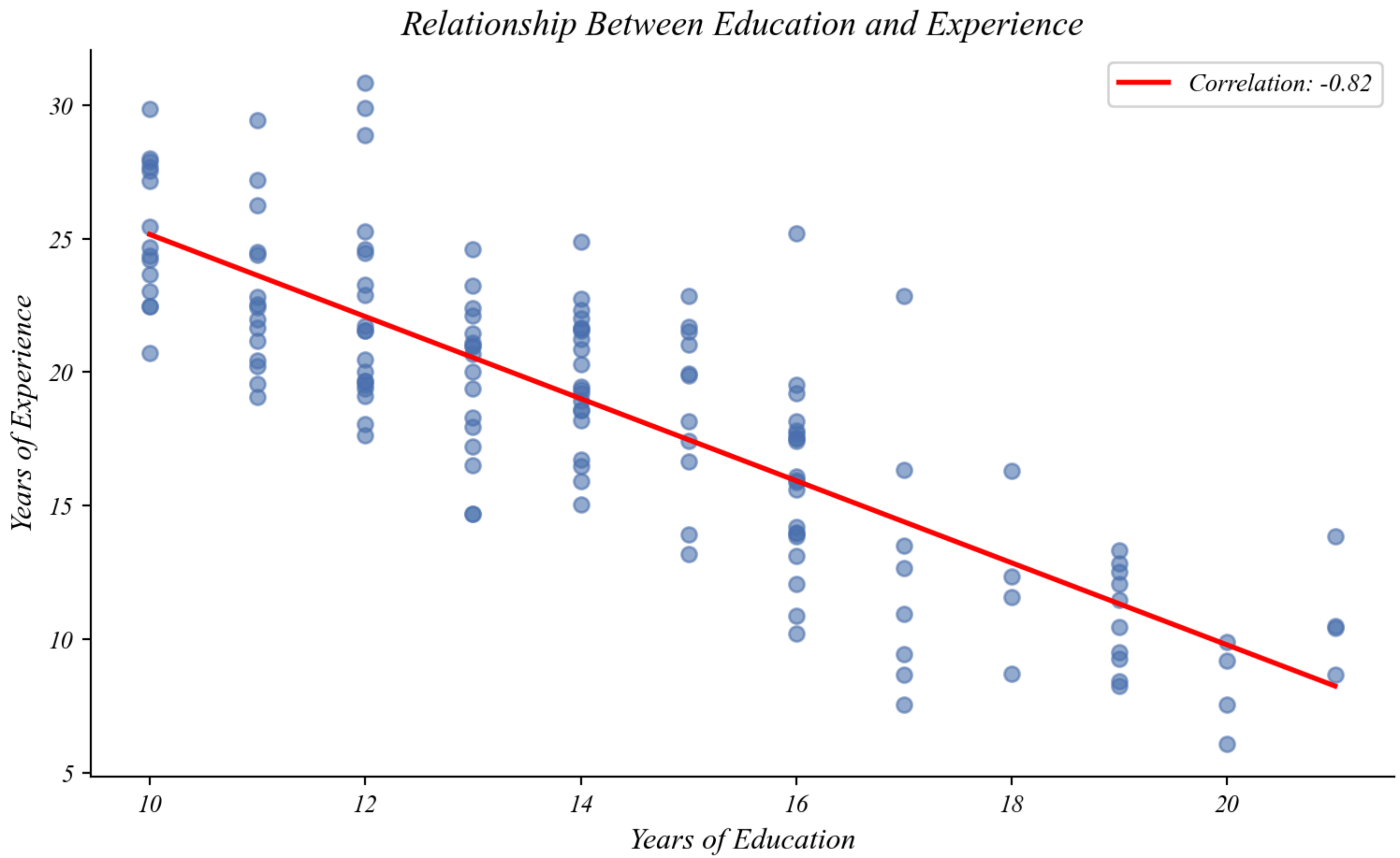
What if we build a regression model for both relationships separately?



> does this mean years of experience has a negative relationship with wages?

The Challenge: Related Variables

Education and experience are correlated!

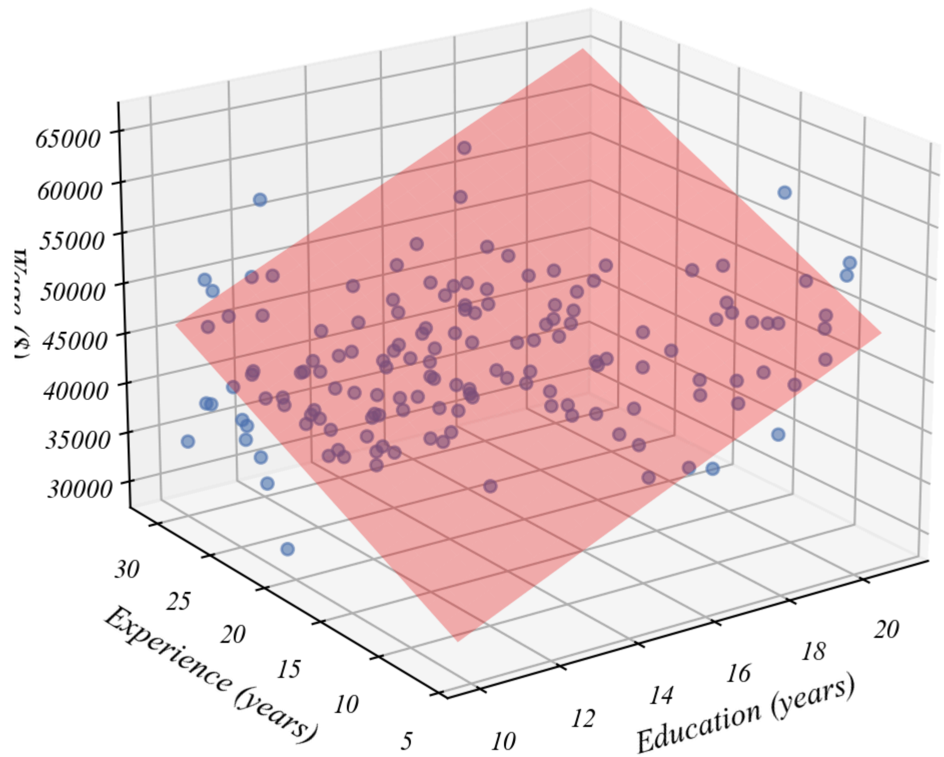


> more education usually means less work experience

> if we look at one without accounting for the other, we get misleading results

Multiple Regression

We can adjust for multiple variables simultaneously.



> multiple regression gives each variable's effect "holding others constant"

The Multiple Regression Equation

Extending the best-fitting line to multiple dimensions

Single Variable:

$$\text{Wage} = \beta_0 + \beta_1 \times \text{Education} + \epsilon$$

Multiple Variables:

$$\text{Wage} = \beta_0 + \beta_1 \times \text{Education} + \beta_2 \times \text{Experience} + \epsilon$$

Interpretation:

- β_0 *= Base wage (intercept)*
- β_1 *= Effect of one more year of education, holding experience constant*
- β_2 *= Effect of one more year of experience, holding education constant*

Example: Testing with Multiple Regression

We can test individual variables or groups of variables

```
import statsmodels.formula.api as smf

# Fit multiple regression model
model = smf.ols('INCLOG10 ~ EDU + AGE', data=data).fit()
```

> *can test each one like before (t-test)*

> *"Are education AND age related to wages?"*

> *does this mean the model without* `AGE` *was wrong?*

> *how do we know if we've included everything?*

Indicator (dummy) Variables

… we can easily turn numerical or categorical variables into indicator variables.

```
# 1. Simple binary indicator (above/below threshold)
model1 = smf.ols('INCLOG10 ~ I(EDU > 12)', data=data).fit()
```

```
# 2. Multiple thresholds/categories
model2 = smf.ols('INCLOG10 ~ I(EDU > 12) + I(EDU < 9)', data=data).fit()
```

```
# 3. Indicators from existing categorical variable
model3 = smf.ols('INCLOG10 ~ EDU + C(DEGFIELD) data=data).fit()
```

Looking Forward

Next steps in building the general linear model…

Next topics:

- *Omitted variable bias*
- *Fixed effects*
- *Multicollinearity*
- *Causality*
- *Basic time series*
- *Multiple slope models*