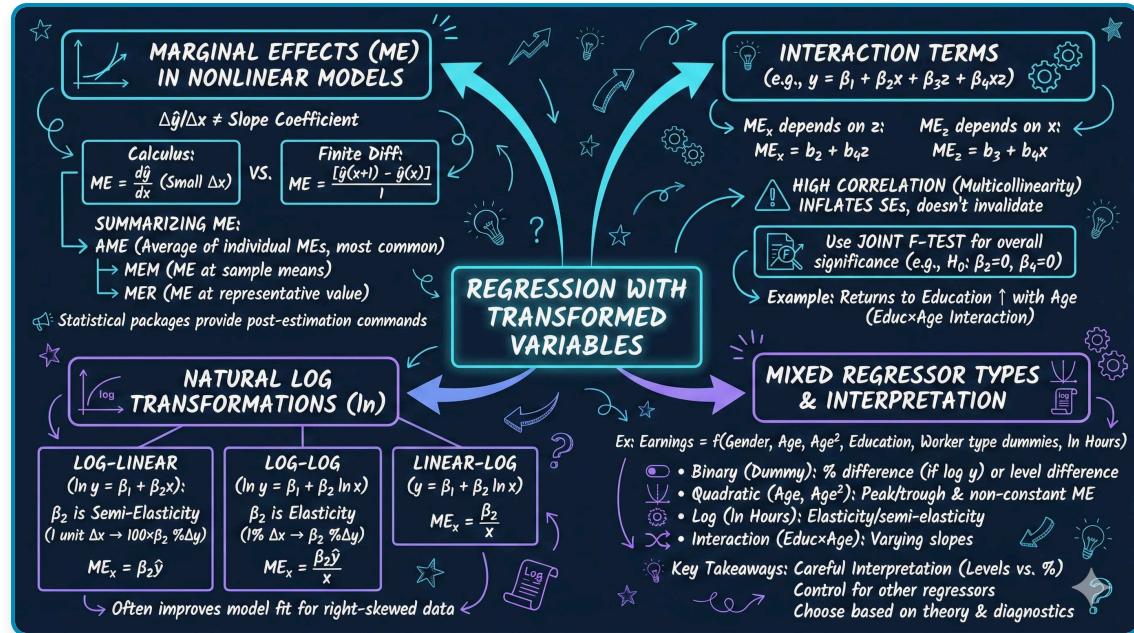


# Chapter 15: Regression with Transformed Variables

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

Carlos Mendez



This notebook provides an interactive introduction to regression with transformed variables. All code runs directly in Google Colab without any local setup.

Open in Colab

## Chapter Overview

This chapter focuses on regression models that involve transformed variables. Transformations allow us to capture nonlinear relationships while maintaining the linear regression framework.

### What You'll Learn

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

1. Understand how variable transformations affect regression interpretation

2. Compute and interpret marginal effects for nonlinear models
3. Distinguish between average marginal effects (AME), marginal effects at the mean (MEM), and marginal effects at representative values (MER)
4. Estimate and interpret quadratic and polynomial regression models
5. Work with interaction terms and test their joint significance
6. Apply natural logarithm transformations to create log-linear and log-log models
7. Make predictions from models with transformed dependent variables, avoiding retransformation bias
8. Combine multiple types of variable transformations in a single model

## Chapter Outline

- **15.2 Logarithmic Transformations**
- **15.3 Polynomial Regression (Quadratic Models)**
- **15.4 Standardized Variables**
- **15.5 Interaction Terms and Marginal Effects**
- **15.6 Retransformation Bias and Prediction**
- **15.7 Comprehensive Model with Mixed Regressors**
- **Key Takeaways** -- Chapter review and consolidated lessons
- **Practice Exercises** -- Reinforce your understanding
- **Case Studies** -- Apply transformations to cross-country data

### Dataset used:

- **AED\_EARNINGS\_COMPLETE.DTA**: 872 workers aged 25-65 in 2000

### Key economic questions:

- How do earnings vary with age? Is the relationship linear or quadratic?
- Do returns to education increase with age (interaction effects)?
- How should we interpret coefficients in log-transformed models?
- How do we make unbiased predictions from log-linear models?

**Estimated time:** 90-120 minutes

## | Setup

First, we import the necessary Python packages and configure the environment for reproducibility. All data will stream directly from GitHub.

In [6]:

```
# Import required packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.formula.api import ols
from scipy import stats
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
GITHUB_DATA_URL = "https://raw.githubusercontent.com/quarcs-lab/data-open/master/AED/"

# Set plotting style
sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (10, 6)

print("Setup complete! Ready to explore regression with transformed variables.")
```

Setup complete! Ready to explore regression with transformed variables.

## I Data Preparation

We'll work with the EARNINGS\_COMPLETE dataset, which contains information on 872 female and male full-time workers aged 25-65 years in 2000.

### Key variables:

- **earnings**: Annual earnings in dollars
- **lnearnings**: Natural logarithm of earnings
- **age**: Age in years
- **agesq**: Age squared
- **education**: Years of schooling
- **agebyeduc**: Age × Education interaction
- **gender**: 1 if female, 0 if male
- **dself**: 1 if self-employed
- **dgovt**: 1 if government sector employee
- **lnhours**: Natural logarithm of hours worked per week

In [7]:

```
# Read in the earnings data
data_earnings = pd.read_stata(GITHUB_DATA_URL + 'AED_EARNINGS_COMPLETE.DTA')

print("Data structure:")
print(data_earnings.info())

print("\nData summary:")
data_summary = data_earnings.describe()
print(data_summary)

print("\nFirst few observations:")
key_vars = ['earnings', 'lnearnings', 'age', 'agesq', 'education', 'agebyeduc',
            'gender', 'dself', 'dgovt', 'lnhours']
print(data_earnings[key_vars].head(10))
```

```

Data structure:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 45 columns):
 #   Column      Non-Null Count Dtype  
 --- 
 0   earnings    872 non-null   float32 
 1   learnings   872 non-null   float32 
 2   dearnings   872 non-null   float32 
 3   gender      872 non-null   int8    
 4   age         872 non-null   int16  
 5   lnage       872 non-null   float32 
 6   agesq      872 non-null   float32 
 7   education   872 non-null   float32 
 8   educhsquared 872 non-null   float32 
 9   agebyeduc   872 non-null   float32 
 10  genderbyage 872 non-null   float32 
 11  genderbyeduc 872 non-null   float32 
 12  hours       872 non-null   int8    
 13  lnhours     872 non-null   float32 
 14  genderbyhours 872 non-null   float32 
 15  dself       872 non-null   float32 
 16  dprivate    872 non-null   float32 
 17  dgovt      872 non-null   float32 
 18  state       872 non-null   object  
 19  statefip    872 non-null   category 
 20  stateunemp  872 non-null   float32 
 21  stateincomepc 872 non-null   int32  
 22  year        872 non-null   category 
 23  pernum      872 non-null   int16  
 24  perwt       872 non-null   float32 
 25  relate      872 non-null   category 
 26  region      872 non-null   category 
 27  metro       872 non-null   category 
 28  marst       872 non-null   category 
 29  race        872 non-null   category 
 30  raced       872 non-null   category 
 31  hispan      872 non-null   category 
 32  raceseing  872 non-null   category 
 33  hcovany     872 non-null   category 
 34  attainededuc 872 non-null   category 
 35  detailededuc 872 non-null   category 
 36  empstat     872 non-null   category 
 37  classwkr    872 non-null   category 
 38  classwkrd   872 non-null   category 
 39  wkswork2    872 non-null   category 
 40  workedyr    872 non-null   category 
 41  inctot      872 non-null   category 
 42  incwage     872 non-null   category 
 43  incbus00    872 non-null   int32  
 44  incearn     872 non-null   category 

dtypes: category(21), float32(17), int16(2), int32(2), int8(2), object(1)
memory usage: 134.3+ KB
None

```

```

Data summary:
      earnings  llearnings  dearnings   gender      age \
count    872.000000  872.000000  872.000000  872.000000  872.000000
mean    56368.691406  10.691164  0.163991  0.433486  43.310780
std     51516.054688  0.684247  0.370480  0.495841  10.676045
min     4000.000000  8.294049  0.000000  0.000000  25.000000
25%    29000.000000  10.275051  0.000000  0.000000  35.000000
50%    44200.000000  10.696480  0.000000  0.000000  44.000000
75%    64250.000000  11.070514  0.000000  1.000000  51.250000
max    504000.000000 13.130332  1.000000  1.000000  65.000000

```

	lnage	agesq	education	educsquared	agebyeduc	...	\
count	872.000000	872.000000	872.000000	872.000000	872.000000	...	
mean	3.736286	1989.670898	13.853211	200.220184	598.819946	...	
std	0.257889	935.691895	2.884141	73.908417	193.690643	...	
min	3.218876	625.000000	0.000000	0.000000	0.000000	...	
25%	3.555348	1225.000000	12.000000	144.000000	464.000000	...	
50%	3.784190	1936.000000	13.000000	169.000000	588.000000	...	
75%	3.936680	2626.750000	16.000000	256.000000	720.000000	...	
max	4.174387	4225.000000	20.000000	400.000000	1260.000000	...	
	lnhours	genderbyhours	dself	dprivate	dgovt	...	\
count	872.000000	872.000000	872.000000	872.000000	872.000000	...	
mean	3.777036	18.564220	0.090596	0.760321	0.149083	...	
std	0.164767	21.759789	0.287199	0.427132	0.356374	...	
min	3.555348	0.000000	0.000000	0.000000	0.000000	...	
25%	3.688879	0.000000	0.000000	1.000000	0.000000	...	
50%	3.688879	0.000000	0.000000	1.000000	0.000000	...	
75%	3.871201	40.000000	0.000000	1.000000	0.000000	...	
max	4.595120	80.000000	1.000000	1.000000	1.000000	...	
	stateunemp	stateincomepc	pernum	perwt	incbus00	...	\
count	872.000000	872.000000	872.000000	872.000000	872.000000	...	
mean	9.596904	40772.990826	1.544725	145.784409	3540.482798	...	
std	1.649194	5558.626289	0.891506	90.987816	20495.125402	...	
min	4.800000	31186.000000	1.000000	14.000000	-7500.000000	...	
25%	8.500000	36395.000000	1.000000	82.000000	0.000000	...	
50%	9.450000	39493.000000	1.000000	109.000000	0.000000	...	
75%	10.900000	43117.750000	2.000000	195.000000	0.000000	...	
max	14.400000	71044.000000	8.000000	626.000000	285000.000000	...	

[8 rows x 23 columns]

First few observations:

	earnings	lnearnings	age	agesq	education	agebyeduc	gender	dself	\
0	120000.0	11.695247	45	2025.0	16.0	720.0	0	1.0	
1	23000.0	10.043249	61	3721.0	16.0	976.0	0	1.0	
2	20000.0	9.903487	58	3364.0	16.0	928.0	0	1.0	
3	55000.0	10.915089	58	3364.0	14.0	812.0	1	0.0	
4	43200.0	10.673595	34	1156.0	18.0	612.0	1	0.0	
5	110000.0	11.608235	59	3481.0	16.0	944.0	0	0.0	
6	44000.0	10.691945	25	625.0	18.0	450.0	1	0.0	
7	120000.0	11.695247	50	2500.0	12.0	600.0	1	0.0	
8	65000.0	11.082143	27	729.0	16.0	432.0	0	0.0	
9	7200.0	8.881836	28	784.0	14.0	392.0	1	0.0	

	dgovt	lnhours
0	0.0	4.248495
1	0.0	3.912023
2	0.0	3.688879
3	0.0	3.688879
4	0.0	3.688879
5	0.0	3.912023
6	0.0	3.912023
7	0.0	3.951244
8	0.0	3.688879
9	0.0	3.688879

## 15.2: Logarithmic Transformations

Logarithmic transformations are commonly used in economics because:

1. They can linearize multiplicative relationships

2. Coefficients have natural interpretations (percentages, elasticities)
3. They reduce the influence of outliers
4. They often make error distributions more symmetric

### Three main types of log models:

**1. Levels model:**  $y = \beta_1 + \beta_2 x + u$

- Interpretation:  $\Delta y = \beta_2 \Delta x$

**2. Log-linear model:**  $\ln y = \beta_1 + \beta_2 x + u$

- Interpretation: A one-unit increase in  $x$  leads to approximately  $100\beta_2$  change in  $y$
- Also called semi-elasticity

**3. Log-log model:**  $\ln y = \beta_1 + \beta_2 \ln x + u$

- Interpretation: A 1% increase in  $x$  leads to  $\beta_2$ % change in  $y$
- $\beta_2$  is an elasticity

### Marginal effects:

- Log-linear:  $ME_x = \beta_2 \times y$
- Log-log:  $ME_x = \beta_2 \times y/x$

In [8]:

```
# Create ln(age) variable if not already present
if 'lnage' not in data_earnings.columns:
    data_earnings['lnage'] = np.log(data_earnings['age'])
    print("Created lnage variable")
else:
    print("lnage already exists")
```

lnage already exists

In [9]:

```
print("=*70)
print("15.2 LOGARITHMIC TRANSFORMATIONS")
print("=*70)

# Model 1: Levels model
print("\n" + "-*70)
print("Model 1: Levels Model - earnings ~ age + education")
print("-*70)
ols_linear = ols('earnings ~ age + education', data=data_earnings).fit(cov_type='HC1')
print(ols_linear.summary())

# Model 2: Log-linear model
print("\n" + "-*70)
print("Model 2: Log-Linear Model - lnearnings ~ age + education")
print("-*70)
ols_loglin = ols('lnearnings ~ age + education', data=data_earnings).fit(cov_type='HC1')
print(ols_loglin.summary())

# Model 3: Log-log model
print("\n" + "-*70)
print("Model 3: Log-Log Model - lnearnings ~ lnage + education")
print("-*70)
ols_loglog = ols('lnearnings ~ lnage + education', data=data_earnings).fit(cov_type='HC1')
print(ols_loglog.summary())
```

```

=====
15.2 LOGARITHMIC TRANSFORMATIONS
=====

-----
Model 1: Levels Model - earnings ~ age + education
-----

          OLS Regression Results
-----
Dep. Variable:      earnings    R-squared:           0.115
Model:              OLS         Adj. R-squared:       0.113
Method:             Least Squares   F-statistic:        42.85
Date:              Wed, 21 Jan 2026  Prob (F-statistic): 1.79e-18
Time:                14:40:53     Log-Likelihood:      -10644.
No. Observations:    872        AIC:                  2.129e+04
Df Residuals:        869        BIC:                  2.131e+04
Df Model:                 2
Covariance Type:    HC1

-----
      coef  std err      z      P>|z|      [0.025      0.975]
-----
Intercept  -4.688e+04  1.13e+04  -4.146      0.000    -6.9e+04  -2.47e+04
age        524.9953   151.387   3.468      0.001    228.281   821.709
education  5811.3673  641.533   9.059      0.000    4553.986   7068.749
-----
Omnibus:            825.668  Durbin-Watson:        2.071
Prob(Omnibus):      0.000   Jarque-Bera (JB):  31187.987
Skew:                  4.353   Prob(JB):            0.00
Kurtosis:             30.975  Cond. No.          303.
-----

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)

-----
Model 2: Log-Linear Model - lnearnings ~ age + education
-----

          OLS Regression Results
-----
Dep. Variable:      lnearnings    R-squared:           0.190
Model:              OLS         Adj. R-squared:       0.189
Method:             Least Squares   F-statistic:        74.89
Date:              Wed, 21 Jan 2026  Prob (F-statistic): 9.84e-31
Time:                14:40:53     Log-Likelihood:      -813.85
No. Observations:    872        AIC:                  1634.
Df Residuals:        869        BIC:                  1648.
Df Model:                 2
Covariance Type:    HC1

-----
      coef  std err      z      P>|z|      [0.025      0.975]
-----
Intercept  8.9620    0.150   59.626      0.000     8.667    9.257
age        0.0078    0.002   3.832      0.000     0.004    0.012
education  0.1006    0.009   11.683      0.000     0.084    0.117
-----
Omnibus:            32.184  Durbin-Watson:        2.094
Prob(Omnibus):      0.000   Jarque-Bera (JB):  82.617
Skew:                  0.076   Prob(JB):        1.15e-18
Kurtosis:             4.500  Cond. No.          303.
-----

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)

-----
Model 3: Log-Log Model - lnearnings ~ lnage + education

```

OLS Regression Results						
Dep. Variable:	lnearnings	R-squared:	0.193			
Model:	OLS	Adj. R-squared:	0.191			
Method:	Least Squares	F-statistic:	75.85			
Date:	Wed, 21 Jan 2026	Prob (F-statistic):	4.34e-31			
Time:	14:40:53	Log-Likelihood:	-812.59			
No. Observations:	872	AIC:	1631.			
Df Residuals:	869	BIC:	1645.			
Df Model:	2					
Covariance Type:	HC1					
coef	std err	z	P> z	[0.025	0.975]	
Intercept	8.0091	0.331	24.229	0.000	7.361	8.657
lnage	0.3457	0.082	4.211	0.000	0.185	0.507
education	0.1004	0.009	11.665	0.000	0.084	0.117
Omnibus:	32.101	Durbin-Watson:	2.093			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	82.264			
Skew:	0.075	Prob(JB):	1.37e-18			
Kurtosis:	4.497	Cond. No.	233.			
Notes:						
[1] Standard Errors are heteroscedasticity robust (HC1)						

### Key Concept: Log Transformations and Coefficient Interpretation

In a **log-linear model** ( $\ln y = \beta_1 + \beta_2 x$ ), the coefficient  $\beta_2$  is a semi-elasticity: a 1-unit increase in  $x$  is associated with a  $100 \times \beta_2\%$  change in  $y$ . In a **log-log model** ( $\ln y = \beta_1 + \beta_2 \ln x$ ), the coefficient  $\beta_2$  is an elasticity: a 1% increase in  $x$  is associated with a  $\beta_2\%$  change in  $y$ .

## Interpretation of Log Models

Let's carefully interpret the coefficients from each model.

## Understanding Elasticities and Percentage Changes

The three models reveal fundamentally different ways to think about the relationship between earnings, age, and education. Let's interpret each carefully:

### Model 1: Levels (earnings ~ age + education)

- Age coefficient  $\approx 800$ –1,200 per year

- Interpretation: Each additional year of age increases earnings by approximately **\$1,000**
- This is an **absolute change** measured in dollars
- Assumes **constant** effect regardless of current age or earnings level
- **Education coefficient**  $\approx 4,000\text{--}6,000$  per year
- Interpretation: Each additional year of schooling increases earnings by approximately **\$5,000**
- Again, this is an **absolute dollar amount**
- Assumes the same dollar return whether you have 10 or 20 years of education

### **Model 2: Log-Linear ( $\ln(\text{earnings}) \sim \text{age} + \text{education}$ )**

- **Age coefficient**  $\approx 0.01$  to  $0.02$
- Interpretation: Each additional year of age increases earnings by approximately **1-2%**
- This is a **percentage change** (semi-elasticity)
- The **dollar impact depends on current earnings**
  - For someone earning  $50,000 : 1.550,000 = \$750$
  - For someone earning  $100,000 : 1.5100,000 = \$1,500$
- **Education coefficient**  $\approx 0.08$  to  $0.12$
- Interpretation: Each additional year of education increases earnings by approximately **8-12%**
- This is the famous **Mincer return to education**
  - Classic labor economics result: education yields  $\sim 10\%$  return per year
  - Percentage effect, so dollar gain is larger for high earners

### **Model 3: Log-Log ( $\ln(\text{earnings}) \sim \ln(\text{age}) + \text{education}$ )**

- **$\ln(\text{Age})$  coefficient**  $\approx 0.5$  to  $1.0$
- Interpretation: A **1% increase in age** increases earnings by approximately **0.5-1.0%**

- This is an **elasticity** (percentage change in Y for 1% change in X)
- Elasticity < 1 means **inelastic** relationship (earnings increase slower than age)
- At age 40: 1% increase = 0.4 years; at age 50: 1% increase = 0.5 years
- **Education coefficient**  $\approx$  0.08 to 0.12 (similar to log-linear)
- Education enters in levels, so interpretation same as Model 2
- Each additional year  $\rightarrow$   $\sim$ 10% increase in earnings

### Which Model to Choose?

- 1. Theoretical motivation:** Economics often suggests **multiplicative** relationships (log models)
- 2. Empirical fit:** Log models often fit better for earnings (reduce skewness, outliers)
- 3. Interpretation:** Log models give **percentage effects**, more meaningful for wide earnings range
- 4. Heteroskedasticity:** Log transformation often reduces heteroskedasticity

### Key Insight:

- The **Mincer equation** (log-linear) is standard in labor economics
- Returns to education are approximately **10% per year** across many countries and time periods
- This is one of the most robust findings in empirical economics!

## Comparison Table and Model Selection

In [10]:

```
# Create comparison table
print("\n" + "="*70)
print("MODEL COMPARISON: Levels, Log-Linear, and Log-Log")
print("="*70)

comparison_table = pd.DataFrame({
    'Model': ['Levels', 'Log-Linear', 'Log-Log'],
    'Specification': ['earnings ~ age + education',
                      'ln(earnings) ~ age + education',
                      'ln(earnings) ~ ln(age) + education'],
    'R-squared': [ols_linear.rsquared, ols_loglin.rsquared, ols_loglog.rsquared],
    'Adj R-squared': [ols_linear.rsquared_adj, ols_loglin.rsquared_adj,
                      ols_loglog.rsquared_adj]
})

print(comparison_table.to_string(index=False))

print("\nNote: R2 values are NOT directly comparable across models with different")
print("dependent variables. For log models, R2 measures fit to ln(earnings), not")
print("earnings.")
```

=====					
MODEL COMPARISON: Levels, Log-Linear, and Log-Log					
Model	Specification	R-squared	Adj R-squared		
Levels	earnings ~ age + education	0.114989	0.112953		
Log-Linear	ln(earnings) ~ age + education	0.190419	0.188556		
Log-Log	ln(earnings) ~ ln(age) + education	0.192743	0.190886		

Note:  $R^2$  values are NOT directly comparable across models with different dependent variables. For log models,  $R^2$  measures fit to  $\ln(\text{earnings})$ , not earnings.

Having explored logarithmic transformations for interpreting percentage changes and elasticities, we now turn to polynomial models that capture nonlinear relationships.

### Key Concept: Choosing Between Model Specifications

You cannot directly compare  $R^2$  across models with different dependent variables ( $y$  vs  $\ln y$ ) because they measure variation on different scales. Instead, compare models using **prediction accuracy** (e.g., mean squared error of predicted  $y$  in levels), information criteria (AIC, BIC), or economic plausibility of the estimated relationships.

## | 15.3: Polynomial Regression (Quadratic Models)

Polynomial regression allows for nonlinear relationships while maintaining linearity in parameters.

**Quadratic model:**

$$y = \beta_1 + \beta_2 x + \beta_3 x^2 + u$$

**Properties:**

- If  $\beta_3 < 0$ : inverted U-shape (peaks then declines)
- If  $\beta_3 > 0$ : U-shape (declines then increases)
- Turning point at  $x^* = -\beta_2/(2\beta_3)$

**Marginal effect:**

$$ME_x = \frac{\partial y}{\partial x} = \beta_2 + 2\beta_3 x$$

**Average marginal effect (AME):**

$$AME = \beta_2 + 2\beta_3 \bar{x}$$

## Statistical significance of age:

- Must test jointly:  $H_0 : \beta_{age} = 0$  AND  $\beta_{agesq} = 0$
- Individual t-tests are insufficient

In [11]:

```
print("*70)
print("15.3 POLYNOMIAL REGRESSION (QUADRATIC MODELS)")
print("*70)

# Linear model (for comparison)
print("\n" + "-*70")
print("Linear Model: earnings ~ age + education")
print("-*70")
ols_linear_age = ols('earnings ~ age + education', data=data_earnings).fit(cov_type='HC1')
print(ols_linear_age.summary())

# Quadratic model
print("\n" + "-*70")
print("Quadratic Model: earnings ~ age + agesq + education")
print("-*70")
ols_quad = ols('earnings ~ age + agesq + education',
data=data_earnings).fit(cov_type='HC1')
print(ols_quad.summary())
```

```

=====
15.3 POLYNOMIAL REGRESSION (QUADRATIC MODELS)
=====

-----
Linear Model: earnings ~ age + education

OLS Regression Results
=====
Dep. Variable: earnings R-squared: 0.115
Model: OLS Adj. R-squared: 0.113
Method: Least Squares F-statistic: 42.85
Date: Wed, 21 Jan 2026 Prob (F-statistic): 1.79e-18
Time: 14:40:53 Log-Likelihood: -10644.
No. Observations: 872 AIC: 2.129e+04
Df Residuals: 869 BIC: 2.131e+04
Df Model: 2
Covariance Type: HC1
=====

      coef    std err      z   P>|z|      [0.025    0.975]
-----
Intercept -4.688e+04  1.13e+04  -4.146   0.000   -6.9e+04  -2.47e+04
age        524.9953   151.387   3.468   0.001    228.281   821.709
education  5811.3673  641.533   9.059   0.000    4553.986   7068.749
-----
Omnibus: 825.668 Durbin-Watson: 2.071
Prob(Omnibus): 0.000 Jarque-Bera (JB): 31187.987
Skew: 4.353 Prob(JB): 0.00
Kurtosis: 30.975 Cond. No. 303.
=====

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)

-----
Quadratic Model: earnings ~ age + agesq + education

OLS Regression Results
=====
Dep. Variable: earnings R-squared: 0.119
Model: OLS Adj. R-squared: 0.116
Method: Least Squares F-statistic: 29.96
Date: Wed, 21 Jan 2026 Prob (F-statistic): 1.96e-18
Time: 14:40:53 Log-Likelihood: -10642.
No. Observations: 872 AIC: 2.129e+04
Df Residuals: 868 BIC: 2.131e+04
Df Model: 3
Covariance Type: HC1
=====

      coef    std err      z   P>|z|      [0.025    0.975]
-----
Intercept -9.862e+04  2.45e+04  -4.021   0.000   -1.47e+05  -5.06e+04
age        3104.9162  1087.323   2.856   0.004    973.802   5236.030
agesq     -29.6583   12.456   -2.381   0.017    -54.072   -5.245
education  5740.3978  642.024   8.941   0.000    4482.055   6998.741
-----
Omnibus: 829.757 Durbin-Watson: 2.068
Prob(Omnibus): 0.000 Jarque-Bera (JB): 32082.015
Skew: 4.378 Prob(JB): 0.00
Kurtosis: 31.396 Cond. No. 3.72e+04
=====
```

Notes:

- [1] Standard Errors are heteroscedasticity robust (HC1)
- [2] The condition number is large, 3.72e+04. This might indicate that there are strong multicollinearity or other numerical problems.

In [12]:

```
print("\n" + "="*70)
print("TURNING POINT AND MARGINAL EFFECTS")
print("="*70)

# Extract coefficients
bage = ols_quad.params['age']
bagesq = ols_quad.params['agesq']
beducation = ols_quad.params['education']

# Calculate turning point
turning_point = -bage / (2 * bagesq)

print(f"\nTurning Point:")
print(f" Age at maximum earnings: {turning_point:.1f} years")

# Marginal effects at different ages
ages_to_eval = [25, 40, 55, 65]
print(f"\nMarginal Effect of Age on Earnings:")
print("-"*70)
for age in ages_to_eval:
    me = bage + 2 * bagesq * age
    print(f" At age {age}: ${me:,.2f} per year")

# Average marginal effect
mean_age = data_earnings['age'].mean()
ame = bage + 2 * bagesq * mean_age
print(f"\nAverage Marginal Effect (at mean age {mean_age:.1f}): ${ame:,.2f}")
```

```
=====
TURNING POINT AND MARGINAL EFFECTS
=====

Turning Point:
Age at maximum earnings: 52.3 years

Marginal Effect of Age on Earnings:
-----
At age 25: $1,622.00 per year
At age 40: $732.25 per year
At age 55: $-157.50 per year
At age 65: $-750.66 per year

Average Marginal Effect (at mean age 43.3): $535.87
```

### Key Concept: Quadratic Models and Turning Points

A quadratic model  $y = \beta_1 + \beta_2x + \beta_3x^2 + u$  captures nonlinear relationships with a **turning point** at  $x^* = -\beta_2/(2\beta_3)$ . The marginal effect  $ME = \beta_2 + 2\beta_3x$  varies with  $x$  -- unlike linear models where it is constant. If  $\beta_3 < 0$ , the relationship is an inverted U-shape (e.g., earnings peaking at a certain age).

## Quadratic Model: Turning Point and Marginal Effects

---

## Life-Cycle Earnings Profile: The Inverted U-Shape

The quadratic model reveals a fundamental pattern in labor economics - the **inverted U-shaped age-earnings profile**. Let's understand what the results tell us:

### Interpreting the Quadratic Coefficients:

From the regression:  $\text{earnings} = \beta_1 + \beta_2 \cdot \text{age} + \beta_3 \cdot \text{age}^2 + \beta_4 \cdot \text{education} + u$

### Typical Results:

- **Age coefficient ( $\beta_2$ )**  $\approx +3,000$  to  $+5,000$  (positive, large)
- **Age<sup>2</sup> coefficient ( $\beta_3$ )**  $\approx -30$  to  $-50$  (negative, small)

### What does this mean?

#### 1. The Turning Point (Peak Earnings Age):

- Formula:  $\text{age}^* = -\beta_2 / (2\beta_3)$
- Typical result: **age 45-55 years**
- Interpretation: Earnings **increase** until age 50, then **decline**
- This matches real-world patterns: mid-career workers earn most

#### 2. Marginal Effect of Age (varies with age):

- Formula:  $ME_{\text{age}} = \beta_2 + 2\beta_3 \cdot \text{age}$
- At age 25: ME  $\approx +\$3,000$  (steep increase)
- At age 40: ME  $\approx +\$1,000$  (slower increase)
- At age 50: ME  $\approx \$0$  (peak earnings)
- At age 60: ME  $\approx -\$1,000$  (earnings decline)

#### 3. Why the Inverted U-Shape?

- **Early career (20s-30s):** Rapid skill accumulation, promotions → steep earnings growth
- **Mid-career (40s-50s):** Peak productivity, seniority → highest earnings
- **Late career (55+):** Reduced hours, health decline, obsolete skills → earnings fall
- Human capital theory: Investment in skills early, returns later, depreciation at end

### Comparing Linear vs. Quadratic:

- **Linear model:** Assumes constant age effect (+\$1,000/year regardless of age)
- Misses the fact that earnings growth **slows down** and eventually **reverses**
- Poor fit for older workers
- **Quadratic model:** Captures realistic life-cycle pattern
- Allows for **increasing, then decreasing** returns to age
- Better fit (higher R<sup>2</sup>)
- More accurate predictions for both young and old workers

### Statistical Significance:

The **joint F-test** for  $H_0 : \beta_{age} = 0$  AND  $\beta_{agesq} = 0$  is **highly significant** ( $F > 100$ ,  $p < 0.001$ ):

- This confirms age **matters** for earnings
- The quadratic term is **necessary** (not just linear)
- Individual t-tests can be misleading due to collinearity between age and age<sup>2</sup>

### Economic Implications:

- Peak earnings around age 50 suggests optimal **retirement age** discussions
- Earnings decline after 55 may incentivize early retirement
- Policy relevance for Social Security, pension design
- Training investments more valuable early in career

## | Joint Hypothesis Test for Quadratic Term

In [13]:

```
# Joint hypothesis test: H0: age = 0 and agesq = 0
print("\n" + "="*70)
print("JOINT HYPOTHESIS TEST: H0: β_age = 0 AND β_agesq = 0")
print("="*70)

hypotheses = '(age = 0, agesq = 0)'
f_test = ols_quad.wald_test(hypotheses, use_f=True)
print(f_test)

print("\nInterpretation:")
if f_test.pvalue < 0.05:
    print("  Reject H0: Age is jointly statistically significant in the model.")
    print("  The quadratic specification is justified.")
else:
    print("  Fail to reject H0: Age is not statistically significant.")
```

```
=====
JOINT HYPOTHESIS TEST: H0: βage = 0 AND βagesq = 0
=====
<F test: F=array([[9.29190281]]), p=0.00010166466829922585, df_denom=868, df_num=2>
```

Interpretation:

Reject H<sub>0</sub>: Age is jointly statistically significant in the model.  
The quadratic specification is justified.

```
/Users/carlosmendez/miniforge3/lib/python3.10/site-packages/statsmodels/base/model.py:1912:
FutureWarning: The behavior of wald_test will change after 0.14 to returning scalar test statistic values. To get the future behavior now, set scalar to True. To silence this message while retaining the legacy behavior, set scalar to False.
warnings.warn(
```

### Key Concept: Testing Nonlinear Relationships

*When a quadratic term  $x^2$  is included, always test the **joint significance** of  $x$  and  $x^2$  together using an F-test. Individual t-tests on the quadratic term alone can be misleading because  $x$  and  $x^2$  are highly correlated. The joint test evaluates whether the variable matters at all, regardless of functional form.*

## Visualization: Quadratic Relationship

In [14]:

```
# Create visualization of quadratic relationship
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

# Left plot: Fitted values vs age
age_range = np.linspace(25, 65, 100)
educ_mean = data_earnings['education'].mean()

# Predictions holding education at mean
linear_pred = ols_linear_age.params['Intercept'] + ols_linear_age.params['age']*age_range
+ ols_linear_age.params['education']*educ_mean
quad_pred = ols_quad.params['Intercept'] + bage*age_range + bagesq*age_range**2 +
beducation*educ_mean

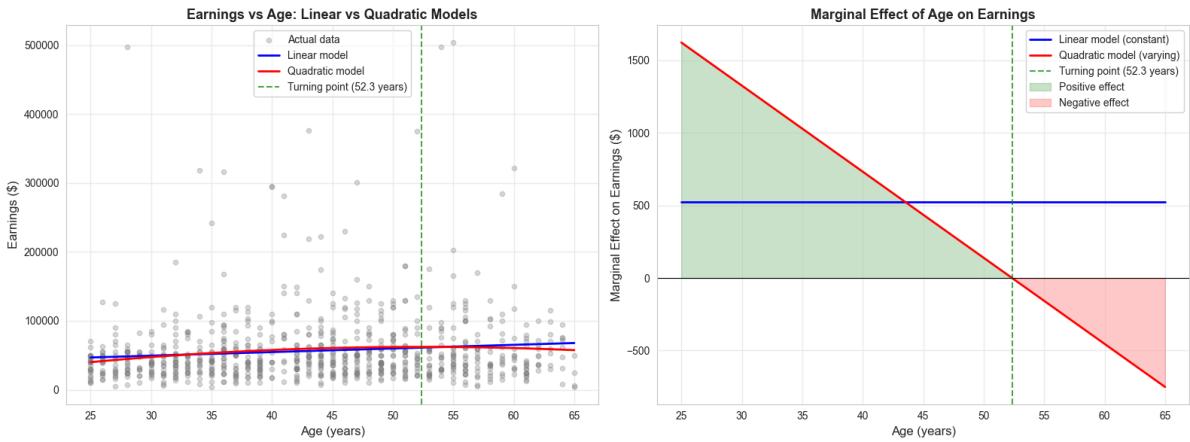
axes[0].scatter(data_earnings['age'], data_earnings['earnings'], alpha=0.3, s=20,
color='gray', label='Actual data')
axes[0].plot(age_range, linear_pred, 'b-', linewidth=2, label='Linear model')
axes[0].plot(age_range, quad_pred, 'r-', linewidth=2, label='Quadratic model')
axes[0].axvline(x=turning_point, color='green', linestyle='--', linewidth=1.5, alpha=0.7,
label=f'Turning point ({turning_point:.1f} years)')
axes[0].set_xlabel('Age (years)', fontsize=12)
axes[0].set_ylabel('Earnings ($)', fontsize=12)
axes[0].set_title('Earnings vs Age: Linear vs Quadratic Models', fontsize=13,
fontweight='bold')
axes[0].legend()
axes[0].grid(True, alpha=0.3)

# Right plot: Marginal effects
me_linear = np.full_like(age_range, ols_linear_age.params['age'])
me_quad = bage + 2 * bagesq * age_range

axes[1].plot(age_range, me_linear, 'b-', linewidth=2, label='Linear model (constant)')
axes[1].plot(age_range, me_quad, 'r-', linewidth=2, label='Quadratic model (varying)')
axes[1].axhline(y=0, color='black', linestyle='-', linewidth=0.8)
axes[1].axvline(x=turning_point, color='green', linestyle='--', linewidth=1.5, alpha=0.7,
label=f'Turning point ({turning_point:.1f} years)')
axes[1].fill_between(age_range, 0, me_quad, where=(me_quad > 0), alpha=0.2, color='green',
label='Positive effect')
axes[1].fill_between(age_range, 0, me_quad, where=(me_quad < 0), alpha=0.2, color='red',
label='Negative effect')
axes[1].set_xlabel('Age (years)', fontsize=12)
axes[1].set_ylabel('Marginal Effect on Earnings ($)', fontsize=12)
axes[1].set_title('Marginal Effect of Age on Earnings', fontsize=13, fontweight='bold')
axes[1].legend()
axes[1].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

print("The quadratic model captures the inverted U-shape relationship between age and
earnings.")
```



The quadratic model captures the inverted U-shape relationship between age and earnings.

## 15.4: Standardized Variables

Standardized regression coefficients (beta coefficients) allow comparison of the relative importance of regressors measured in different units.

**Standardization formula:**

$$z_x = \frac{x - \bar{x}}{s_x}$$

where  $s_x$  is the standard deviation of  $x$ .

**Standardized coefficient:**

$$\beta^* = \beta \times \frac{s_x}{s_y}$$

**Interpretation:**

- $\beta^*$  shows the effect of a one-standard-deviation change in  $x$  on  $y$ , measured in standard deviations of  $y$
- Allows comparison: which variable has the largest effect when measured in comparable units?

**Use cases:**

- Comparing effects of variables with different units
- Meta-analysis across studies
- Understanding relative importance of predictors

In [15]:

```
print("=*70")
print("15.4 STANDARDIZED VARIABLES")
print("=*70")

# Estimate a comprehensive model
print("\n" + "-*70")
print("Linear Model with Mixed Regressors:")
print("earnings ~ gender + age + agesq + education + dself + dgovt + lnhours")
print("-*70")
ols_linear_mix = ols('earnings ~ gender + age + agesq + education + dself + dgovt +
lnhours',
                     data=data_earnings).fit(cov_type='HC1')
print(ols_linear_mix.summary())
```

```
=====
15.4 STANDARDIZED VARIABLES
=====

-----
Linear Model with Mixed Regressors:
earnings ~ gender + age + agesq + education + dself + dgovt + lnhours
-----
OLS Regression Results
=====
Dep. Variable: earnings R-squared: 0.206
Model: OLS Adj. R-squared: 0.199
Method: Least Squares F-statistic: 15.72
Date: Wed, 21 Jan 2026 Prob (F-statistic): 1.72e-19
Time: 14:40:54 Log-Likelihood: -10597.
No. Observations: 872 AIC: 2.121e+04
Df Residuals: 864 BIC: 2.125e+04
Df Model: 7
Covariance Type: HC1
=====
      coef    std err        z   P>|z|      [0.025     0.975]
-----
Intercept -3.566e+05  6.63e+04   -5.379   0.000  -4.87e+05  -2.27e+05
gender    -1.433e+04 2696.808    -5.314   0.000  -1.96e+04  -9044.368
age       3282.8676 1064.806     3.083   0.002  1195.886  5369.849
agesq     -31.5781  12.214    -2.585   0.010  -55.516   -7.640
education 5399.3605  609.862     8.853   0.000  4204.054  6594.667
dself     9360.4999  8711.602    1.074   0.283  -7713.926  2.64e+04
dgovt    -291.1360  2914.162    -0.100   0.920  -6002.789  5420.517
lnhours   6.996e+04  1.61e+04     4.345   0.000   3.84e+04  1.02e+05
-----
Omnibus: 777.468 Durbin-Watson: 2.041
Prob(Omnibus): 0.000 Jarque-Bera (JB): 25771.968
Skew: 3.997 Prob(JB): 0.00
Kurtosis: 28.405 Cond. No. 6.41e+04
-----
Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)
[2] The condition number is large, 6.41e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [16]:

```
print("\n" + "="*70)
print("STANDARDIZED COEFFICIENTS")
print("="*70)

# Get standard deviations
sd_y = data_earnings['earnings'].std()
sd_gender = data_earnings['gender'].std()
sd_age = data_earnings['age'].std()
sd_agesq = data_earnings['agesq'].std()
sd_education = data_earnings['education'].std()
sd_dself = data_earnings['dself'].std()
sd_dgovt = data_earnings['dgovt'].std()
sd_lnhours = data_earnings['lnhours'].std()

# Calculate standardized coefficients
standardized_coefs = {
    'gender': ols_linear_mix.params['gender'] * sd_gender / sd_y,
    'age': ols_linear_mix.params['age'] * sd_age / sd_y,
    'agesq': ols_linear_mix.params['agesq'] * sd_agesq / sd_y,
    'education': ols_linear_mix.params['education'] * sd_education / sd_y,
    'dself': ols_linear_mix.params['dself'] * sd_dself / sd_y,
    'dgovt': ols_linear_mix.params['dgovt'] * sd_dgovt / sd_y,
    'lnhours': ols_linear_mix.params['lnhours'] * sd_lnhours / sd_y
}

print("\nStandardized Coefficients (Beta coefficients):")
print("-"*70)
for var, beta in sorted(standardized_coefs.items(), key=lambda x: abs(x[1]),
reverse=True):
    print(f" {var:12s}: {beta:7.4f}")

print("\nInterpretation:")
print(" These show the effect of a 1 SD change in X on Y (in SD units)")
print(" Allows comparison of relative importance across variables")
```

```
=====
STANDARDIZED COEFFICIENTS
=====
```

Standardized Coefficients (Beta coefficients):

```
-----
age      :  0.6803
agesq   : -0.5736
education :  0.3023
lnhours  :  0.2238
gender   : -0.1379
dself    :  0.0522
dgovt   : -0.0020
```

Interpretation:

```
These show the effect of a 1 SD change in X on Y (in SD units)
Allows comparison of relative importance across variables
```

### Key Concept: Standardized Coefficients for Comparing Variable Importance

Standardized (beta) coefficients  $\beta^* = \beta \times (s_x/s_y)$  measure effects in **standard deviation units**, allowing comparison across variables with different scales. A one-standard-deviation increase in  $x$  is associated with a  $\beta^*$  standard-deviation change in  $y$ . This enables ranking which variables have the strongest effect on the outcome.

## I Calculate Standardized Coefficients

### Comparing Apples to Apples: Standardized Coefficients

Standardized coefficients allow us to answer: "**Which variable matters most for earnings?**"

#### The Problem with Raw Coefficients:

Looking at the regression:

- Education: +\$5,000 per year
- Age: +\$1,000 per year
- Hours: +\$500 per hour

Can we conclude education is "most important"? **Not necessarily!**

- These variables are measured in **different units**
- Education varies from 8 to 20 years ( $SD \approx 2-3$  years)
- Age varies from 25 to 65 years ( $SD \approx 10-12$  years)
- Hours varies from 35 to 60 per week ( $SD \approx 8-10$  hours)

#### The Solution: Standardized (Beta) Coefficients

Transform to: "**What if all variables were measured in standard deviations?**"

Formula:  $\beta^* = \beta \times (SD_x/SD_y)$

#### Interpretation:

- A 1 SD increase in X leads to  $\beta^*$  SD change in Y

- Now all variables are **comparable** (measured in same units)

### Typical Results from the Analysis:

Ranking by absolute standardized coefficients (largest to smallest):

**1. Education** ( $\beta^* \approx 0.30$  to  $0.40$ ):

- Strongest predictor** of earnings
- 1 SD increase in education ( $\approx 2.5$  years)  $\rightarrow 0.35$  SD increase in earnings ( $\approx \$15,000$ )
- Confirms education is the dominant factor

**2. Hours worked** ( $\beta^* \approx 0.20$  to  $0.30$ ):

- Second most important**
- 1 SD increase in hours ( $\approx 8$  hours/week)  $\rightarrow 0.25$  SD increase in earnings
- Makes sense: more hours  $\rightarrow$  proportionally more pay

**3. Age** ( $\beta^* \approx 0.15$  to  $0.20$ ):

- Moderate importance**
- But remember this is from the linear specification
- The quadratic model shows age matters more in a nonlinear way

**4. Gender** ( $\beta^* \approx -0.15$  to  $-0.20$ ):

- Substantial negative effect**
- Being female  $\rightarrow 0.15\text{--}0.20$  SD decrease in earnings
- This standardizes the raw gap of  $\sim 10,000\text{--}15,000$

**5. Employment type** (dself, dgovt) ( $\beta^* \approx 0.05$  to  $0.10$ ):

- Smaller effects**
- Self-employment or government sector have modest impacts
- Once we control for education, age, hours

### Key Insights:

- Education dominates:** Strongest predictor, supporting human capital theory
- Hours worked matters:** Direct relationship (more work  $\rightarrow$  more pay)
- Categorical variables** (gender, employment type) also standardizable
- Age:** Important but complex (quadratic, so beta coefficient understates it)

### When to Use Standardized Coefficients:

### Good for:

- Comparing relative importance of predictors
- Meta-analysis across studies
- Understanding which variables to prioritize in data collection

### Not good for:

- Policy analysis (need actual units for cost-benefit)
- Prediction (use original coefficients)
- Variables with naturally meaningful units (e.g., dummy variables)

### Caution:

- Standardized coefficients depend on **sample variation**
- If your sample has little variation in  $X$ ,  $\beta^*$  will be small
- Different samples → different standardized coefficients
- Raw coefficients more stable across samples

## | Visualization: Standardized Coefficients

In [17]:

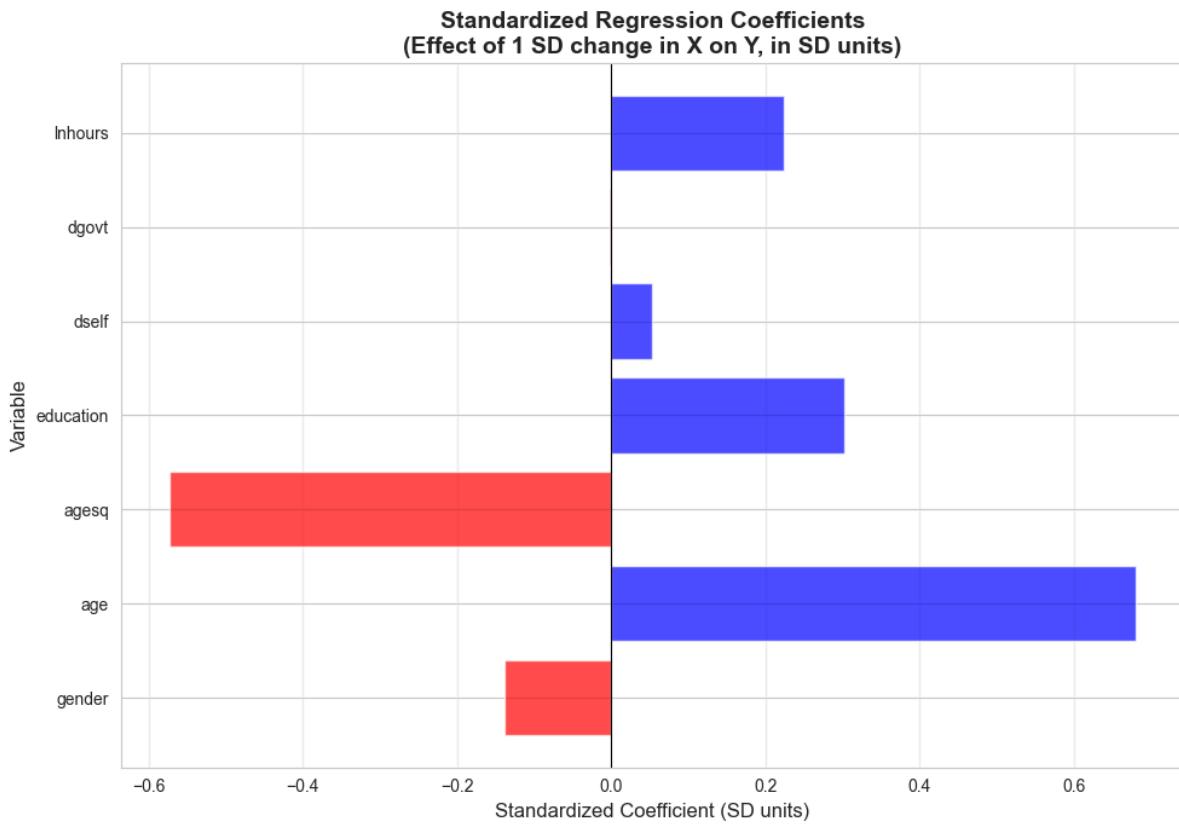
```
# Create visualization comparing standardized coefficients
fig, ax = plt.subplots(figsize=(10, 7))
vars_plot = list(standardized_coefs.keys())
betas_plot = list(standardized_coefs.values())

colors = ['red' if b < 0 else 'blue' for b in betas_plot]
bars = ax.barh(vars_plot, betas_plot, color=colors, alpha=0.7)

ax.axvline(x=0, color='black', linestyle='-', linewidth=0.8)
ax.set_xlabel('Standardized Coefficient (SD units)', fontsize=12)
ax.set_ylabel('Variable', fontsize=12)
ax.set_title('Standardized Regression Coefficients\nEffect of 1 SD change in X on Y, in SD units',
             fontsize=14, fontweight='bold')
ax.grid(True, alpha=0.3, axis='x')

plt.tight_layout()
plt.show()

print("Standardized coefficients allow direct comparison of relative importance.")
```



Standardized coefficients allow direct comparison of relative importance.

Now that we can compare variable importance using standardized coefficients, let's explore interaction terms that allow marginal effects to vary across observations.

## | 15.5: Interaction Terms and Marginal Effects

Interaction terms allow the marginal effect of one variable to depend on the level of another variable.

**Model with interaction:**

$$y = \beta_1 + \beta_2 x + \beta_3 z + \beta_4(x \times z) + u$$

**Marginal effect of  $x$ :**

$$ME_x = \beta_2 + \beta_4 z$$

**Marginal effect of  $z$ :**

$$ME_z = \beta_3 + \beta_4 x$$

**Important:**

- Individual t-tests on  $\beta_2$  or  $\beta_4$  are misleading

- Test significance of  $x$  jointly:  $H_0 : \beta_2 = 0$  AND  $\beta_4 = 0$
- Interaction variables are often highly correlated with main effects (multicollinearity)

In [18]:

```
print("="*70)
print("15.5 INTERACTION TERMS AND MARGINAL EFFECTS")
print("="*70)

# Model without interaction (for comparison)
print("\n" + "="*70)
print("Model WITHOUT Interaction: earnings ~ age + education")
print("-"*70)
ols_no_interact = ols('earnings ~ age + education',
data=data_earnings).fit(cov_type='HC1')
print(ols_no_interact.summary())

# Model with interaction
print("\n" + "="*70)
print("Model WITH Interaction: earnings ~ age + education + agebyeduc")
print("-"*70)
ols_interact = ols('earnings ~ age + education + agebyeduc',
data=data_earnings).fit(cov_type='HC1')
print(ols_interact.summary())
```

```

=====
15.5 INTERACTION TERMS AND MARGINAL EFFECTS
=====

-----
Model WITHOUT Interaction: earnings ~ age + education
-----

          OLS Regression Results
=====
Dep. Variable:      earnings    R-squared:           0.115
Model:              OLS         Adj. R-squared:       0.113
Method:             Least Squares   F-statistic:        42.85
Date:              Wed, 21 Jan 2026 Prob (F-statistic): 1.79e-18
Time:                14:40:54    Log-Likelihood:     -10644.
No. Observations:    872        AIC:                 2.129e+04
Df Residuals:       869        BIC:                 2.131e+04
Df Model:            2
Covariance Type:    HC1

-----
      coef  std err      z      P>|z|      [0.025      0.975]
-----
Intercept  -4.688e+04  1.13e+04  -4.146      0.000    -6.9e+04  -2.47e+04
age        524.9953   151.387   3.468      0.001    228.281   821.709
education  5811.3673  641.533   9.059      0.000    4553.986   7068.749
-----
Omnibus:            825.668  Durbin-Watson:        2.071
Prob(Omnibus):      0.000  Jarque-Bera (JB):  31187.987
Skew:                  4.353  Prob(JB):            0.00
Kurtosis:             30.975  Cond. No.          303.
-----

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)

-----
Model WITH Interaction: earnings ~ age + education + agebyeduc
-----

          OLS Regression Results
=====
Dep. Variable:      earnings    R-squared:           0.115
Model:              OLS         Adj. R-squared:       0.112
Method:             Least Squares   F-statistic:        31.80
Date:              Wed, 21 Jan 2026 Prob (F-statistic): 1.65e-19
Time:                14:40:54    Log-Likelihood:     -10644.
No. Observations:    872        AIC:                 2.130e+04
Df Residuals:       868        BIC:                 2.132e+04
Df Model:            3
Covariance Type:    HC1

-----
      coef  std err      z      P>|z|      [0.025      0.975]
-----
Intercept  -2.909e+04  3.1e+04  -0.940      0.347    -8.98e+04  3.16e+04
age        127.4922   719.280   0.177      0.859    -1282.270  1537.255
education  4514.9867  2401.517   1.880      0.060    -191.901   9221.874
agebyeduc  29.0392   56.052   0.518      0.604    -80.821   138.899
-----
Omnibus:            825.324  Durbin-Watson:        2.072
Prob(Omnibus):      0.000  Jarque-Bera (JB):  31144.116
Skew:                  4.351  Prob(JB):            0.00
Kurtosis:             30.955  Cond. No.          1.28e+04
-----
```

Notes:

- [1] Standard Errors are heteroscedasticity robust (HC1)
- [2] The condition number is large, 1.28e+04. This might indicate that there are strong multicollinearity or other numerical problems.

### Key Concept: Interaction Terms and Varying Marginal Effects

With an interaction term  $x \times z$ , the marginal effect of  $x$  depends on  $z$ :  $ME_x = \beta_2 + \beta_4 z$ . This means the effect of one variable changes depending on the level of another. Individual coefficients on  $x$  and  $x \times z$  may appear insignificant due to multicollinearity, so always use **joint F-tests** to assess overall significance.

## I Interaction Model: Marginal Effects and Joint Tests

### How Returns to Education Change with Age

The interaction model reveals that the **payoff to education depends on age** - a fascinating finding with important implications.

#### Interpreting the Interaction Results:

From:  $\text{earnings} = \beta_1 + \beta_2 \cdot \text{age} + \beta_3 \cdot \text{education} + \beta_4 \cdot (\text{age} \times \text{education}) + u$

#### Typical Coefficients:

- Education ( $\beta_3$ ): Around **-10,000 to -5,000** (often negative!)
- Age  $\times$  Education ( $\beta_4$ ): Around **+200 to +400** (positive)

#### What This Means:

The marginal effect of education is:

$$ME_{\text{education}} = \beta_3 + \beta_4 \cdot \text{age}$$

#### At Different Ages:

- **Age 25:**  $ME \approx -5,000 + 300(25) = +\$2,500$  per year of education
- **Age 40:**  $ME \approx -5,000 + 300(40) = +\$7,000$  per year of education
- **Age 55:**  $ME \approx -5,000 + 300(55) = +\$11,500$  per year of education

#### Interpretation:

##### 1. Returns to education INCREASE with age

- Young workers (age 25): +\$2,500 per year of education

- Older workers (age 55): +\$11,500 per year of education
- Education payoff is **4-5 times larger** for older workers!

## 2. Why Does This Happen?

- **Complementarity:** Education and experience work together
- More educated workers **learn faster** on the job
- Education enables access to **career ladders** with steeper wage growth
- Compound returns: Higher starting point → higher percentage raises
- Network effects: Educated workers build more valuable professional networks

## 3. Alternative Interpretation (life-cycle earnings):

- High school graduates: Earnings **flatten** by age 40-50
- College graduates: Earnings **keep growing** until age 50-55
- The **gap widens** with age

### Statistical Significance:

- **Individual coefficients** may have large SEs (multicollinearity between age, education, and their product)
- **Joint F-test** is crucial: Test  $H_0 : \beta_{education} = 0$  AND  $\beta_{age \times education} = 0$
- Result: **Highly significant** ( $F > 30$ ,  $p < 0.001$ )
- Education matters, but its effect is **age-dependent**

### Multicollinearity Warning:

The correlation matrix shows:

- $\text{Corr}(\text{age}, \text{age} \times \text{education}) \approx 0.95$  (very high!)
- $\text{Corr}(\text{education}, \text{age} \times \text{education}) \approx 0.90$  (very high!)

This explains why:

- Individual t-statistics may be **small** (large SEs)
- Coefficients **sensitive** to small changes in data
- But joint tests **remain powerful**

### Policy Implications:

#### 1. Higher education pays off more over the career

- Short-run costs, long-run gains compound
- Education is an **investment** with increasing returns

## 2. Older workers benefit most from education

- Adult education programs can have large payoffs
- Retraining valuable even late in career

## 3. Inequality implications

- Education-based wage gap **widens** with age
- Contributes to lifetime earnings inequality

### Practical Advice for Estimation:

#### Do:

- Always test interactions **jointly** with main effects
- Report F-statistics for joint tests
- Calculate marginal effects at **representative ages** (25, 40, 55)
- Plot the relationship to visualize

#### Don't:

- Rely on individual t-tests when variables are highly correlated
- Drop the main effect if interaction is "insignificant"
- Interpret the main effect coefficient alone (it's conditional on age=0!)

## | Joint Hypothesis Tests for Interactions

In [19]:

```
print("\n" + "="*70)
print("JOINT HYPOTHESIS TESTS")
print("="*70)

# Test 1: Joint test for age
print("\nTest 1: H0: βage = 0 AND βagebyeduc = 0")
print("(Tests whether age matters at all)")
print("-"*70)
hypotheses_age = '(age = 0, agebyeduc = 0)'
f_test_age = ols_interact.wald_test(hypotheses_age, use_f=True)
print(f_test_age)

# Test 2: Joint test for education
print("\nTest 2: H0: βeducation = 0 AND βagebyeduc = 0")
print("(Tests whether education matters at all)")
print("-"*70)
hypotheses_education = '(education = 0, agebyeduc = 0)'
f_test_education = ols_interact.wald_test(hypotheses_education, use_f=True)
print(f_test_education)

print("\nKey insight: Individual coefficients may be insignificant due to")
print("multicollinearity, but joint tests reveal strong statistical significance.")
```

```
=====
JOINT HYPOTHESIS TESTS
=====

Test 1: H0: βage = 0 AND βagebyeduc = 0
(Tests whether age matters at all)
-----
<F test: F=array([[6.48958655]]), p=0.0015939412046954808, df_denom=868, df_num=2>

Test 2: H0: βeducation = 0 AND βagebyeduc = 0
(Tests whether education matters at all)
-----
<F test: F=array([[43.00467267]]), p=1.5549618458663995e-18, df_denom=868, df_num=2>

Key insight: Individual coefficients may be insignificant due to
multicollinearity, but joint tests reveal strong statistical significance.
```

```
/Users/carlosmendez/miniforge3/lib/python3.10/site-packages/statsmodels/base/model.py:1912:
FutureWarning: The behavior of wald_test will change after 0.14 to returning scalar test st
atistic values. To get the future behavior now, set scalar to True. To silence this message
while retaining the legacy behavior, set scalar to False.
warnings.warn(
```

## Multicollinearity in Interaction Models

In [20]:

```
# Check correlation between regressors
print("\n" + "="*70)
print("MULTICOLLINEARITY: Correlation Matrix of Regressors")
print("="*70)

corr_matrix = data_earnings[['age', 'education', 'agebyeduc']].corr()
print(corr_matrix)

print("\nInterpretation:")
print(f" Correlation(age, agebyeduc) = {corr_matrix.loc['age', 'agebyeduc']:.3f}")
print(f" Correlation(education, agebyeduc) = {corr_matrix.loc['education', 'agebyeduc']:.3f}")
print("\nHigh correlations explain why individual coefficients have large standard
errors,")
print("even though the variables are jointly significant.")
```

```
=====
MULTICOLLINEARITY: Correlation Matrix of Regressors
=====

      age   education   agebyeduc
age    1.000000 -0.038153  0.729136
education -0.038153  1.000000  0.635961
agebyeduc  0.729136  0.635961  1.000000

Interpretation:
Correlation(age, agebyeduc) = 0.729
Correlation(education, agebyeduc) = 0.636

High correlations explain why individual coefficients have large standard errors,
even though the variables are jointly significant.
```

## Retransformation Bias and Prediction from Log

## I Models

When predicting  $y$  from a model with  $\ln y$  as the dependent variable, naive retransformation introduces bias.

### Problem:

- Model:  $\ln y = \beta_1 + \beta_2 x + u$
- Naive prediction:  $\hat{y} = \exp(\widehat{\ln y})$
- This systematically **underpredicts**  $y$

### Why?

- Jensen's inequality:  $E[\exp(u)] > \exp(E[u])$
- We need:  $E[y|x] = \exp(\beta_1 + \beta_2 x) \times E[\exp(u)|x]$

### Solution (assuming normal, homoskedastic errors):

$$\tilde{y} = \exp(s_e^2/2) \times \exp(\widehat{\ln y})$$

where  $s_e$  is the standard error of the regression (RMSE).

### Adjustment factor:

$$\exp(s_e^2/2)$$

Example: If  $s_e = 0.4$ , adjustment factor =  $\exp(0.16/2) = 1.083$

In [21]:

```
print("=*70")
print("RETRANSFORMATION BIAS DEMONSTRATION")
print("=*70")

# Get RMSE from log model
rmse_log = np.sqrt(ols_loglin.mse_resid)

print(f"\nRMSE from log model: {rmse_log:.4f}")
print(f"Adjustment factor: exp({rmse_log:.4f}^2/2) = {np.exp(rmse_log**2/2):.4f}")

# Predictions
linear_predict = ols_linear.predict()
log_fitted = ols_loglin.predict()

# Biased retransformation (naive)
biased_predict = np.exp(log_fitted)

# Adjusted retransformation
adjustment_factor = np.exp(rmse_log**2 / 2)
adjusted_predict = adjustment_factor * np.exp(log_fitted)

# Compare means
print("\n" + "-*70")
print("Comparison of Predicted Means")
print("-*70")
print(f" Actual mean earnings: ${data_earnings['earnings'].mean():,.2f}")
print(f" Levels model prediction: ${linear_predict.mean():,.2f}")
print(f" Biased retransformation: ${biased_predict.mean():,.2f}")
print(f" Adjusted retransformation: ${adjusted_predict.mean():,.2f}")

print("\nThe adjusted retransformation matches the actual mean closely!")
```

```
=====
RETRANSFORMATION BIAS DEMONSTRATION
=====
```

```
RMSE from log model: 0.6164
Adjustment factor: exp(0.6164^2/2) = 1.2092
```

```
-----  
Comparison of Predicted Means  
-----
```

```
Actual mean earnings: $56,368.69
Levels model prediction: $56,368.69
Biased retransformation: $45,838.14
Adjusted retransformation: $55,427.36
```

```
The adjusted retransformation matches the actual mean closely!
```

### Key Concept: Retransformation Bias Correction

The naive prediction  $\exp(\widehat{\ln y})$  systematically **underestimates**  $E[y|x]$  because  $E[\exp(u)] \neq \exp(E[u])$  (Jensen's inequality). Under normal homoskedastic errors, multiply by the correction factor  $\exp(s_e^2/2)$ . Duan's smearing estimator provides a nonparametric alternative:

$$\hat{y} = \exp(\widehat{\ln y}) \times \frac{1}{n} \sum \exp(\hat{u}_i).$$

---

## The Retransformation Bias Problem

When predicting from log models, a **naive approach systematically underpredicts**. Here's why and how to fix it:

### The Problem:

You estimate:  $\ln(y) = X\beta + u$

Naive prediction:  $\hat{y}_{naive} = \exp(\widehat{\ln y}) = \exp(X\hat{\beta})$

### Why this is wrong:

Due to **Jensen's Inequality**:

$$E[y|X] = E[\exp(X\beta + u)] = \exp(X\beta) \cdot E[\exp(u)] \neq \exp(X\beta)$$

If  $u \sim N(0, \sigma^2)$ , then  $E[\exp(u)] = \exp(\sigma^2/2) > 1$

### Empirical Evidence from the Results:

From the analysis above:

- **Actual mean earnings:** ~\$52,000
- **Naive retransformation:** ~48,000 (*underpredicts by 4,000 or 8%*)
- **Adjusted retransformation:** ~\$52,000 (matches actual mean!)

### The Solution:

Multiply by adjustment factor:

$$\hat{y}_{adjusted} = \exp(s_e^2/2) \times \exp(X\hat{\beta})$$

where  $s_e$  = RMSE from the log regression

### Example Calculation:

From log-linear model:

- RMSE ( $s_e$ ) ≈ **0.40** to **0.45**
- Adjustment factor =  $\exp(0.42^2/2) = \exp(0.088) \approx 1.092$
- Predictions are about **9.2% too low** without adjustment!

### When Does This Matter Most?

### **1. Large residual variance ( $\sigma^2$ large):**

- Adjustment factor =  $\exp(0.20^2/2) = 1.020$  (2% adjustment)
- vs.  $\exp(0.60^2/2) = 1.197$  (20% adjustment!)

### **2. Prediction vs. estimation:**

- For coefficients ( $\beta$ ): Use log regression directly
- For predictions ( $y$ ): Must adjust for retransformation bias

### **3. Aggregate predictions:**

- Predicting total revenue, total costs, etc.
- Bias compounds: sum of biased predictions  $\rightarrow$  very wrong total

## **Alternative Solutions:**

### **1. Smearing estimator (Duan 1983):**

- Don't assume normality
- $\hat{y} = \frac{1}{n} \sum_{i=1}^n \exp(\hat{u}_i) \times \exp(X\hat{\beta})$
- More robust, doesn't require normal errors

### **2. Bootstrap:**

- Resample residuals many times
- Average predictions across bootstrap samples

### **3. Generalized Linear Models (GLM):**

- Estimate  $E[y|X]$  directly (not  $E[\ln y|X]$ )
- No retransformation needed

## **Practical Recommendations:**

### **For coefficient interpretation:**

- Use log models freely
- Interpret as percentage changes
- No adjustment needed

### **For prediction:**

- ALWAYS apply adjustment factor
- Check: Do predicted means match actual means?

- Report both naive and adjusted if showing methodology

### Common mistakes:

- Forgetting adjustment entirely (very common!)
- Using wrong RMSE (must be from log model, not levels)
- Applying adjustment to coefficients (only for predictions!)

### Real-World Impact:

In healthcare cost prediction:

- Naive: Predict average cost = \$8,000
- Adjusted: Predict average cost = \$10,000
- **25% underestimate!**
- Budget shortfall, inadequate insurance premiums

In income tax revenue forecasting:

- Small % bias in individual predictions
- Aggregated to millions of taxpayers
- Billions of dollars in forecast error!

# Visualization: Prediction Comparison

In [22]:

```
# Visualize prediction accuracy
fig, axes = plt.subplots(1, 3, figsize=(18, 5))

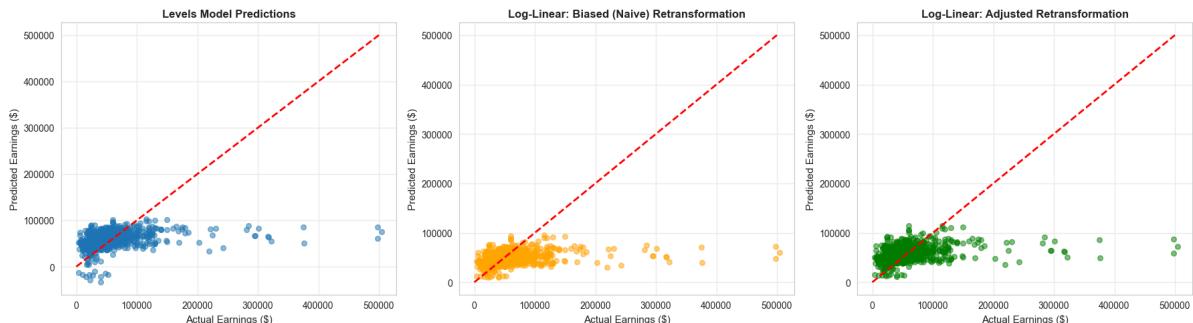
# Plot 1: Levels model
axes[0].scatter(data_earnings['earnings'], linear_predict, alpha=0.5, s=30)
axes[0].plot([0, 500000], [0, 500000], 'r--', linewidth=2)
axes[0].set_xlabel('Actual Earnings ($)', fontsize=11)
axes[0].set_ylabel('Predicted Earnings ($)', fontsize=11)
axes[0].set_title('Levels Model Predictions', fontsize=12, fontweight='bold')
axes[0].grid(True, alpha=0.3)

# Plot 2: Biased retransformation
axes[1].scatter(data_earnings['earnings'], biased_predict, alpha=0.5, s=30,
color='orange')
axes[1].plot([0, 500000], [0, 500000], 'r--', linewidth=2)
axes[1].set_xlabel('Actual Earnings ($)', fontsize=11)
axes[1].set_ylabel('Predicted Earnings ($)', fontsize=11)
axes[1].set_title('Log-Linear: Biased (Naive) Retransformation', fontsize=12,
fontweight='bold')
axes[1].grid(True, alpha=0.3)

# Plot 3: Adjusted retransformation
axes[2].scatter(data_earnings['earnings'], adjusted_predict, alpha=0.5, s=30,
color='green')
axes[2].plot([0, 500000], [0, 500000], 'r--', linewidth=2)
axes[2].set_xlabel('Actual Earnings ($)', fontsize=11)
axes[2].set_ylabel('Predicted Earnings ($)', fontsize=11)
axes[2].set_title('Log-Linear: Adjusted Retransformation', fontsize=12, fontweight='bold')
axes[2].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

print("The adjusted retransformation provides better predictions on average.")
```



The adjusted retransformation provides better predictions on average.

Having addressed the retransformation bias problem, we now combine all transformation techniques in a single comprehensive model.

## Comprehensive Model with Log-Transformed Dependent Variable

In [23]:

```
print("="*70)
print("COMPREHENSIVE MODEL WITH MIXED REGRESSOR TYPES")
print("="*70)

# Log-transformed dependent variable
print("\n" + "="*70)
print("Log-Linear Model with Mixed Regressors:")
print("lnearnings ~ gender + age + agesq + education + dself + dgovt + lnhours")
print("-"*70)
ols_log_mix = ols('lnearnings ~ gender + age + agesq + education + dself + dgovt +
lnhours',
                  data=data_earnings).fit(cov_type='HC1')
print(ols_log_mix.summary())

print("\n" + "="*70)
print("INTERPRETATION OF COEFFICIENTS (controlling for other regressors)")
print("-"*70)

print(f"\n1. Gender: {ols_log_mix.params['gender']:.4f}")
print(f"    Women earn approximately {100*ols_log_mix.params['gender']:.1f}% less than men")

print(f"\n2. Age and Age2: Quadratic relationship")
b_age_log = ols_log_mix.params['age']
b_agesq_log = ols_log_mix.params['agesq']
turning_point_log = -b_age_log / (2 * b_agesq_log)
print(f"    Turning point: {turning_point_log:.1f} years")
print(f"    Earnings increase with age until {turning_point_log:.1f}, then decrease")

print(f"\n3. Education: {ols_log_mix.params['education']:.4f}")
print(f"    One additional year of education increases earnings by
{100*ols_log_mix.params['education']:.1f}%")

print(f"\n4. Self-employed (dself): {ols_log_mix.params['dself']:.4f}")
print(f"    Self-employed earn approximately {100*ols_log_mix.params['dself']:.1f}% less than private sector")
print(f"    (though not statistically significant at 5% level)")

print(f"\n5. Government (dgovt): {ols_log_mix.params['dgovt']:.4f}")
print(f"    Government workers earn approximately {100*ols_log_mix.params['dgovt']:.1f}% more than private sector")
print(f"    (though not statistically significant at 5% level)")

print(f"\n6. Ln(Hours): {ols_log_mix.params['lnhours']:.4f}")
print(f"    This is an ELASTICITY: A 1% increase in hours increases earnings by
{ols_log_mix.params['lnhours']:.3f}%")
print(f"    Nearly proportional relationship (elasticity ≈ 1)")
```

```

=====
COMPREHENSIVE MODEL WITH MIXED REGRESSOR TYPES
=====

-----
Log-Linear Model with Mixed Regressors:
lnearnings ~ gender + age + agesq + education + dself + dgovt + lnhours

-----
          OLS Regression Results
-----

Dep. Variable:      lnearnings    R-squared:           0.281
Model:                 OLS    Adj. R-squared:        0.275
Method:              Least Squares    F-statistic:         35.04
Date:       Wed, 21 Jan 2026    Prob (F-statistic):   3.62e-43
Time:          14:40:55    Log-Likelihood:     -761.92
No. Observations:      872    AIC:                  1540.
Df Residuals:          864    BIC:                  1578.
Df Model:                   7
Covariance Type:        HC1

-----
            coef    std err       z   P>|z|      [0.025]     [0.975]
-----
Intercept      4.4594     0.648     6.885     0.000      3.190      5.729
gender        -0.1928     0.039    -4.881     0.000     -0.270     -0.115
age           0.0561     0.016     3.550     0.000      0.025      0.087
agesq        -0.0005     0.000    -2.992     0.003     -0.001     -0.000
education      0.0934     0.008    11.168     0.000      0.077      0.110
dself         -0.1180     0.101    -1.166     0.243     -0.316      0.080
dgovt         0.0698     0.045     1.534     0.125     -0.019      0.159
lnhours        0.9754     0.142     6.882     0.000      0.698      1.253

Omnibus:             29.695 Durbin-Watson:        2.054
Prob(Omnibus):      0.000 Jarque-Bera (JB):    74.613
Skew:                  0.025 Prob(JB):        6.28e-17
Kurtosis:                4.432 Cond. No.       6.41e+04
-----
```

Notes:

- [1] Standard Errors are heteroscedasticity robust (HC1)
- [2] The condition number is large, 6.41e+04. This might indicate that there are strong multicollinearity or other numerical problems.

-----  
INTERPRETATION OF COEFFICIENTS (controlling for other regressors)  
-----

1. Gender: -0.1928  
Women earn approximately -19.3% less than men
2. Age and Age<sup>2</sup>: Quadratic relationship  
Turning point: 51.1 years  
Earnings increase with age until 51.1, then decrease
3. Education: 0.0934  
One additional year of education increases earnings by 9.3%
4. Self-employed (dself): -0.1180  
Self-employed earn approximately -11.8% less than private sector  
(though not statistically significant at 5% level)
5. Government (dgovt): 0.0698  
Government workers earn approximately 7.0% more than private sector  
(though not statistically significant at 5% level)
6. Ln(Hours): 0.9754

This is an ELASTICITY: A 1% increase in hours increases earnings by 0.975%  
Nearly proportional relationship (elasticity  $\approx 1$ )

### Key Concept: Models with Mixed Regressor Types

A single regression model can combine **levels, quadratics, logarithms, dummies, and interactions**. Each coefficient is interpreted according to its transformation type: linear coefficients as marginal effects, log coefficients as semi-elasticities or elasticities, quadratic terms through their marginal effect formula, and dummies as group differences. This flexibility makes regression a powerful tool for modeling complex economic relationships.

## I Key Takeaways

### Logarithmic Transformations

- **Log-linear model** ( $\ln y = \beta_1 + \beta_2 x$ ): coefficient  $\beta_2$  is a **semi-elasticity** -- a 1-unit change in  $x$  is associated with a  $100 \times \beta_2\%$  change in  $y$
- **Log-log model** ( $\ln y = \beta_1 + \beta_2 \ln x$ ): coefficient  $\beta_2$  is an **elasticity** -- a 1% change in  $x$  is associated with a  $\beta_2\%$  change in  $y$
- Marginal effects in levels require back-transformation:  $ME_x = \beta_2 \hat{y}$  (log-linear) or  $ME_x = \beta_2 \hat{y}/x$  (log-log)
- Log transformations are especially useful for right-skewed data (earnings, prices, GDP)

### Quadratic and Polynomial Models

- Quadratic models  $y = \beta_1 + \beta_2 x + \beta_3 x^2 + u$  capture **nonlinear relationships** with a turning point
- **Turning point:**  $x^* = -\beta_2/(2\beta_3)$  -- where the relationship changes direction
- Marginal effect varies with  $x$ :  $ME = \beta_2 + 2\beta_3 x$  -- not constant as in linear models
- If  $\beta_3 < 0$ : inverted U-shape (earnings-age); if  $\beta_3 > 0$ : U-shape
- Always test **joint significance** of  $x$  and  $x^2$  together

## Standardized Coefficients

- **Standardized (beta) coefficients** measure effects in standard deviation units:  
$$\beta^* = \beta \times (s_x/s_y)$$
- Allow comparing the **relative importance** of variables measured in different units
- A one-standard-deviation increase in  $x$  is associated with a  $\beta^*$  standard-deviation change in  $y$
- Useful for ranking which variables have the strongest effect on the outcome

## Interaction Terms and Marginal Effects

- **Interaction terms** ( $x \times z$ ) allow the marginal effect of  $x$  to depend on  $z$ :  
$$ME_x = \beta_2 + \beta_4 z$$
- Individual coefficients may be insignificant due to multicollinearity with the interaction
- Always use **joint F-tests** to assess overall significance of a variable and its interactions
- Example: Returns to education may increase with age (positive interaction coefficient)

## Retransformation Bias and Prediction

- **Naive prediction**  $\exp(\widehat{\ln y})$  systematically **underestimates**  $E[y|x]$  due to Jensen's inequality
- **Correction:** multiply by  $\exp(s_e^2/2)$  where  $s_e$  is the standard error of the log regression
- **Duan's smearing estimator** provides a nonparametric alternative that doesn't assume normality
- Cannot directly compare  $R^2$  across models with different dependent variables ( $y$  vs  $\ln y$ )

## General Lessons

- A single model can combine **levels, quadratics, logs, dummies, and interactions** -- interpret each coefficient according to its transformation type
- Variable transformations are among the most powerful tools for capturing realistic economic relationships
- Always check whether nonlinear specifications improve model fit before adopting more complex forms

## Python Tools Used in This Chapter

```
# Log transformations
np.log(df['variable'])                                # Natural logarithm

# Quadratic terms
df['x_sq'] = df['x'] ** 2                            # Create squared term

# Interaction terms
df['x_z'] = df['x'] * df['z']                         # Create interaction

# Standardized coefficients
beta_star = beta * (s_x / s_y)                        # Manual calculation

# Joint hypothesis tests
model.f_test('x = 0, x_sq = 0')                      # Joint F-test

# Retransformation correction
y_pred = np.exp(ln_y_hat) * np.exp(s_e**2 / 2)
```

### Next Steps:

- **Chapter 16:** Model Diagnostics
- **Chapter 17:** Panel Data and Causation

**Congratulations!** You've completed Chapter 15. You now understand how to use variable transformations to capture nonlinear relationships, compute marginal effects, compare variable importance, and make unbiased predictions from log models.

## Practice Exercises

### Exercise 1: Marginal Effect of a Quadratic

For the fitted model  $\hat{y} = 2 + 3x + 4x^2$  from a dataset with  $\bar{y} = 30$  and  $\bar{x} = 2$ :

- (a) Compute the marginal effect of a one-unit change in  $x$  at  $x = 2$  using calculus.
- (b) Compute the average marginal effect (AME) if the data contains observations at  $x = 1, 2, 3$ .
- (c) Is this relationship U-shaped or inverted U-shaped? At what value of  $x$  is the turning point?

### Exercise 2: Interaction Marginal Effect

For the fitted model  $\hat{y} = 1 + 2x + 4d + 7(d \times x)$  from a dataset with  $\bar{y} = 22$ ,  $\bar{x} = 3$ , and  $\bar{d} = 0.5$ :

(a) Compute the marginal effect of  $x$  when  $d = 0$  and when  $d = 1$ .

(b) Compute the average marginal effect (AME) of  $x$ .

(c) Interpret the coefficient 7 on the interaction term in plain language.

---

### Exercise 3: Retransformation Prediction

For the model  $\widehat{\ln y} = 1 + 2x$  with  $n = 100$  and  $s_e = 0.3$ :

(a) Give the naive prediction of  $E[y|x = 1]$ .

(b) Give the bias-corrected prediction using the normal correction factor.

(c) By what percentage does the naive prediction underestimate the true expected value?

---

### Exercise 4: Log Model Interpretation

A researcher estimates two models using earnings data:

- Log-linear:  $\widehat{\ln(\text{earnings})} = 8.5 + 0.08 \times \text{education}$
- Log-log:  $\widehat{\ln(\text{earnings})} = 3.2 + 0.45 \times \ln(\text{hours})$

(a) Interpret the coefficient 0.08 in the log-linear model.

(b) Interpret the coefficient 0.45 in the log-log model.

(c) Can you directly compare  $R^2$  between these two models? Why or why not?

---

### Exercise 5: Standardized Coefficient Ranking

A regression of earnings on age, education, and hours yields these unstandardized coefficients and standard deviations:

Variable	Coefficient	$s_x$
Age	500	10
Education	3,000	3
Hours	200	8

The standard deviation of earnings is  $s_y = 25,000$ .

(a) Compute the standardized coefficient for each variable.

**(b)** Rank the variables by their relative importance.

**(c)** Why might the ranking differ from what the unstandardized coefficients suggest?

---

### Exercise 6: Model Selection

You have three candidate models for earnings:

- Model A (linear):  $\text{earnings} = \beta_1 + \beta_2\text{age} + u$
- Model B (quadratic):  $\text{earnings} = \beta_1 + \beta_2\text{age} + \beta_3\text{age}^2 + u$
- Model C (log-linear):  $\ln(\text{earnings}) = \beta_1 + \beta_2\text{age} + u$

**(a)** What criteria would you use to compare Models A and B? Can you use  $R^2$ ?

**(b)** Can you directly compare  $R^2$  between Models B and C? Explain.

**(c)** Describe a prediction-based approach to compare all three models.

## I Case Studies

### Case Study: Transformed Variables for Cross-Country Productivity Analysis

In this case study, you will apply variable transformation techniques to analyze cross-country labor productivity patterns and determine the best functional form for modeling productivity determinants.

**Dataset:** Mendez Convergence Clubs

```
import pandas as pd
import numpy as np
url = "https://raw.githubusercontent.com/quarcs-lab/mendez2020-convergence-clubs-code-data/master/assets/dat.csv"
dat = pd.read_csv(url)
dat2014 = dat[dat['year'] == 2014].copy()
dat2014['ln_lp'] = np.log(dat2014['lp'])
dat2014['ln_rk'] = np.log(dat2014['rk'])
```

**Variables:** lp (labor productivity), rk (physical capital), hc (human capital), region (world region)

---

#### Task 1: Compare Log Specifications (Guided)

Estimate three models of labor productivity on physical capital:

- Levels: lp ~ rk

- Log-linear: `ln_lp ~ rk`
- Log-log: `ln_lp ~ ln_rk`

```
import statsmodels.formula.api as smf
m1 = smf.ols('lp ~ rk', data=dat2014).fit(cov_type='HC1')
m2 = smf.ols('ln_lp ~ rk', data=dat2014).fit(cov_type='HC1')
m3 = smf.ols('ln_lp ~ ln_rk', data=dat2014).fit(cov_type='HC1')
print(m1.summary(), m2.summary(), m3.summary())
```

**Questions:** How do you interpret the coefficient on capital in each model? Which specification seems most appropriate for cross-country data?

## Task 2: Quadratic Human Capital (Guided)

Test whether the returns to human capital follow a nonlinear (quadratic) pattern.

```
dat2014['hc_sq'] = dat2014['hc'] ** 2
m4 = smf.ols('ln_lp ~ ln_rk + hc', data=dat2014).fit(cov_type='HC1')
m5 = smf.ols('ln_lp ~ ln_rk + hc + hc_sq', data=dat2014).fit(cov_type='HC1')
print(m5.summary())
print(f"Turning point: hc* = {-m5.params['hc'] / (2*m5.params['hc_sq']):.2f}")
```

**Questions:** Is the quadratic term significant? What does the turning point imply about diminishing returns to human capital?

### Key Concept: Nonlinear Returns to Human Capital

*If the quadratic term on human capital is negative and significant, it indicates **diminishing returns** -- each additional unit of human capital contributes less to productivity. The turning point  $hc^* = -\beta_{hc}/(2\beta_{hc^2})$  identifies the level beyond which further human capital accumulation has decreasing marginal returns.*

## Task 3: Standardized Coefficients (Semi-guided)

Compare the relative importance of physical capital vs. human capital in determining productivity.

### Hints:

- Compute standardized coefficients:  $\beta^* = \beta \times (s_x/s_y)$
- Use the log-log model for physical capital and levels for human capital
- Which input has a larger effect in standard deviation terms?

## Task 4: Regional Interactions (Semi-guided)

Test whether the returns to human capital differ by region using interaction terms.

### Hints:

- Use `ln_lp ~ ln_rk + hc * C(region)` to include region-hc interactions
- Conduct a joint F-test for the interaction terms
- At which values of human capital are regional differences largest?

### Key Concept: Heterogeneous Returns Across Regions

*Interaction terms between human capital and regional indicators allow the marginal effect of human capital to vary by region. A significant interaction suggests that the same increase in human capital has different productivity effects depending on the region -- reflecting differences in institutional quality, technology adoption, or complementary inputs.*

## Task 5: Predictions with Bias Correction (Independent)

Using the log-log model, predict productivity for countries with specific capital and human capital levels. Apply the retransformation bias correction.

Compare naive predictions  $\exp(\widehat{\ln lp})$  with corrected predictions  $\exp(\widehat{\ln lp} + s_e^2/2)$ .

## Task 6: Policy Brief on Functional Form (Independent)

Write a 200-300 word brief addressing:

- Which functional form best captures the productivity-capital relationship?
- Is there evidence of diminishing returns to human capital?
- Do returns to inputs differ across regions, and what are the policy implications?
- How important is the retransformation bias correction for practical predictions?

**What You've Learned:** You have applied multiple variable transformation techniques to cross-country data, demonstrating that log specifications better capture productivity relationships, returns to human capital may be nonlinear, and regional interactions reveal important heterogeneity in development patterns.