

ECON 0150 | Economic Data Analysis

The economist's data analysis pipeline.

Part 5.2 | Time Series: Differences, Seasonality, and Elasticities

Time Series Analysis

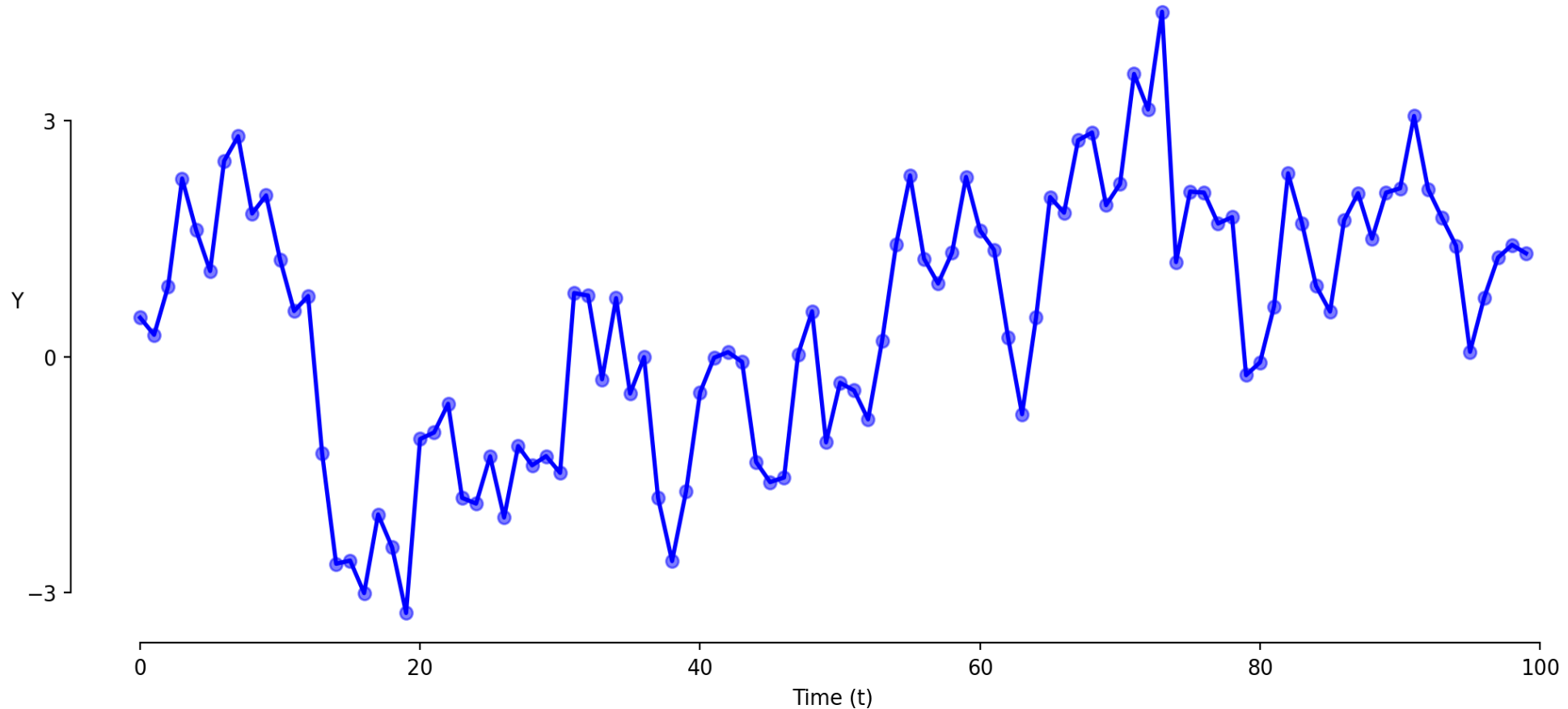
Modeling relationships through time

Key Questions:

- *How do we analyze data that changes over time?*
- *What do trends, cycles, and seasonal patterns tell us?*
- *How do we transform time series data for the general linear model?*

The Challenge of Time Series

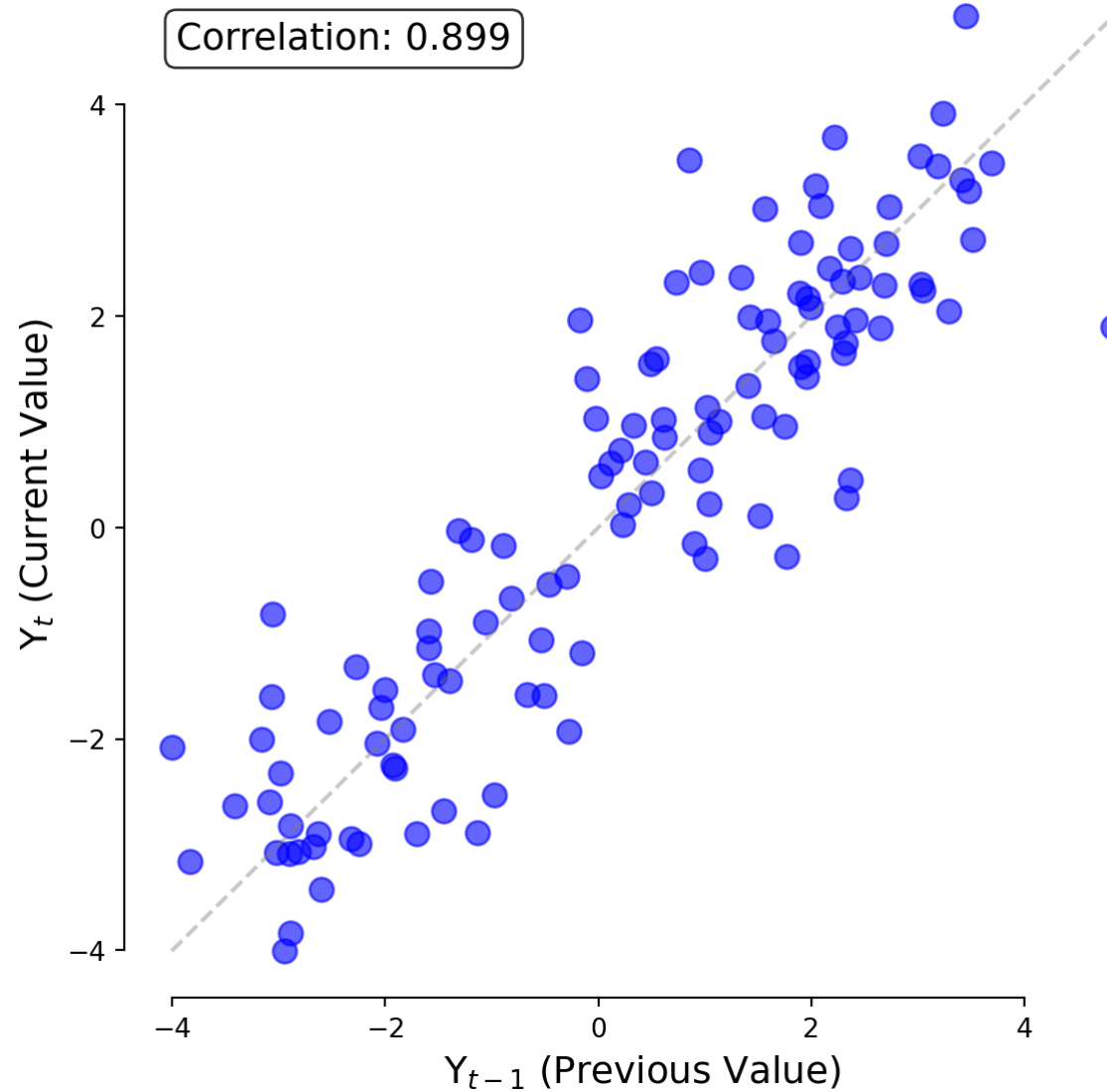
Data related in time has a special problem of autocorrelation.



> observations are related to their past values (serial correlation; autocorrelation)

Autocorrelation

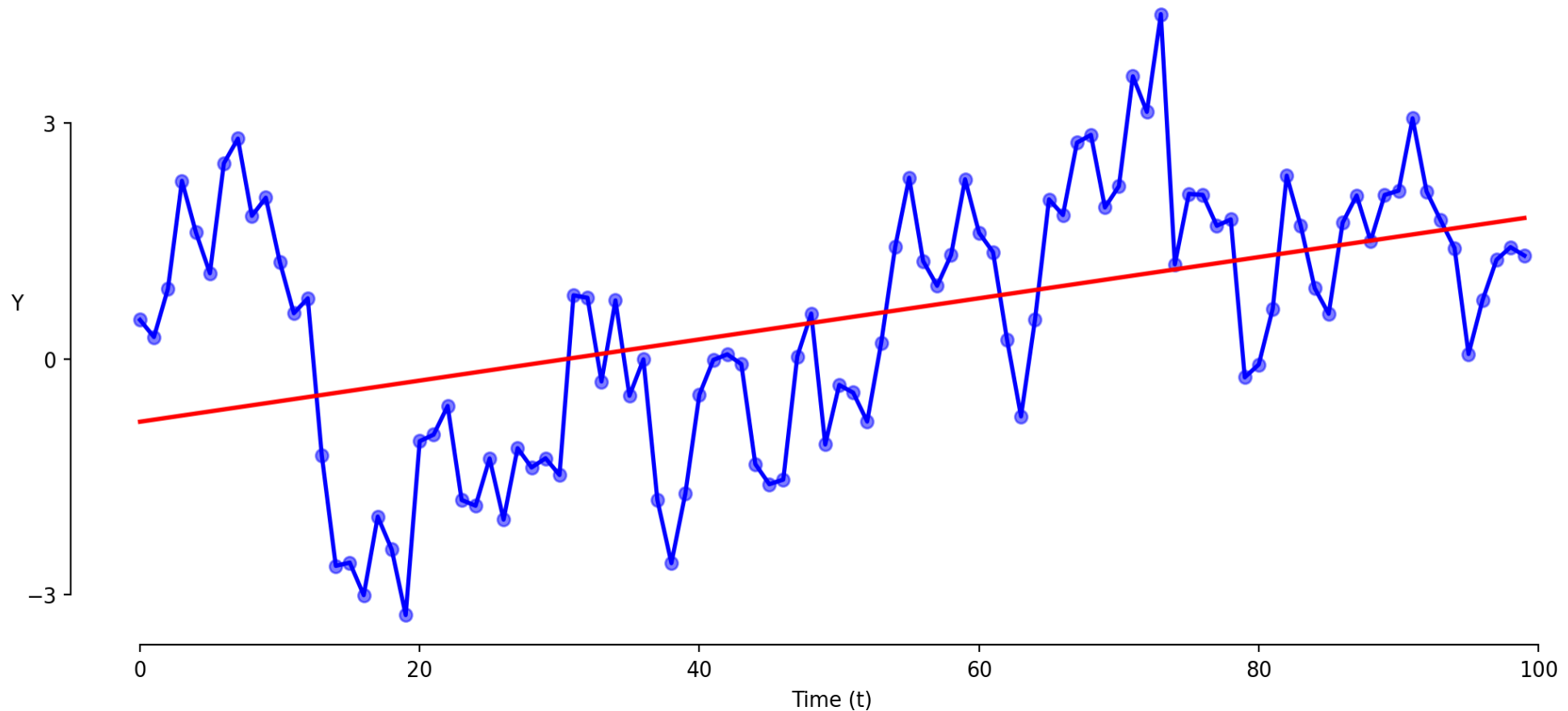
Values in time series are typically related to their own past values.



Model 1: Levels Regression

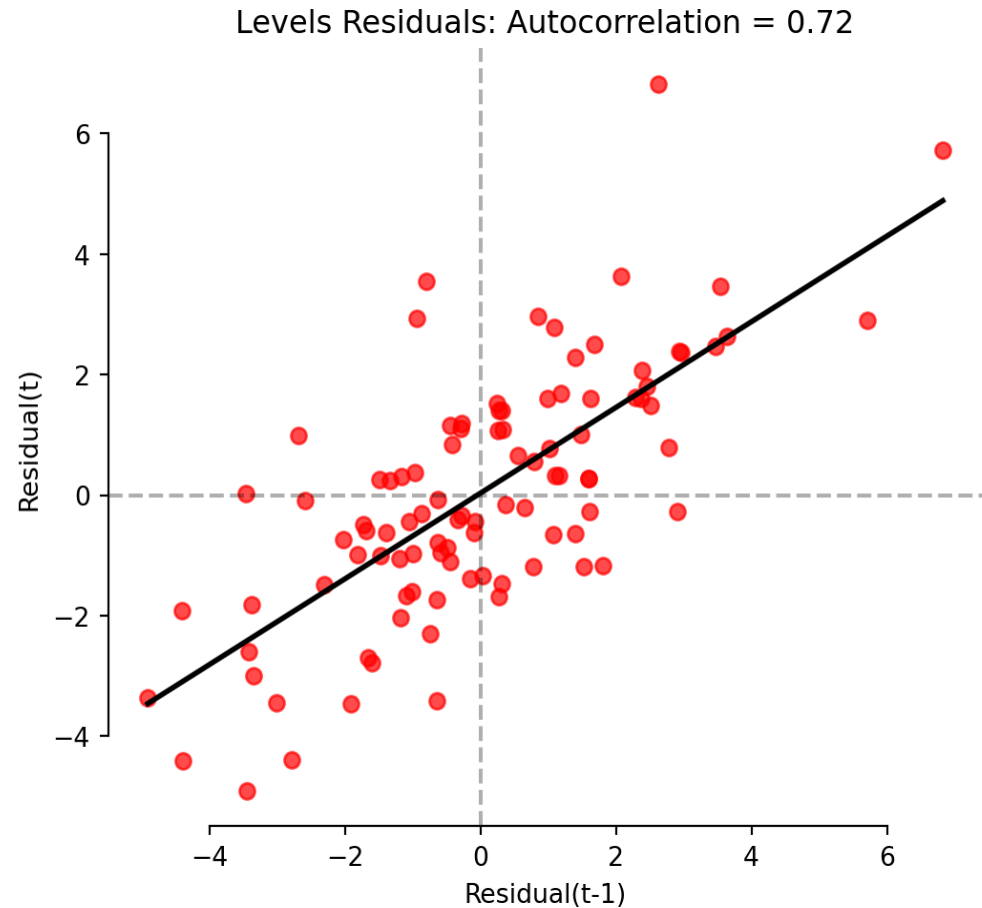
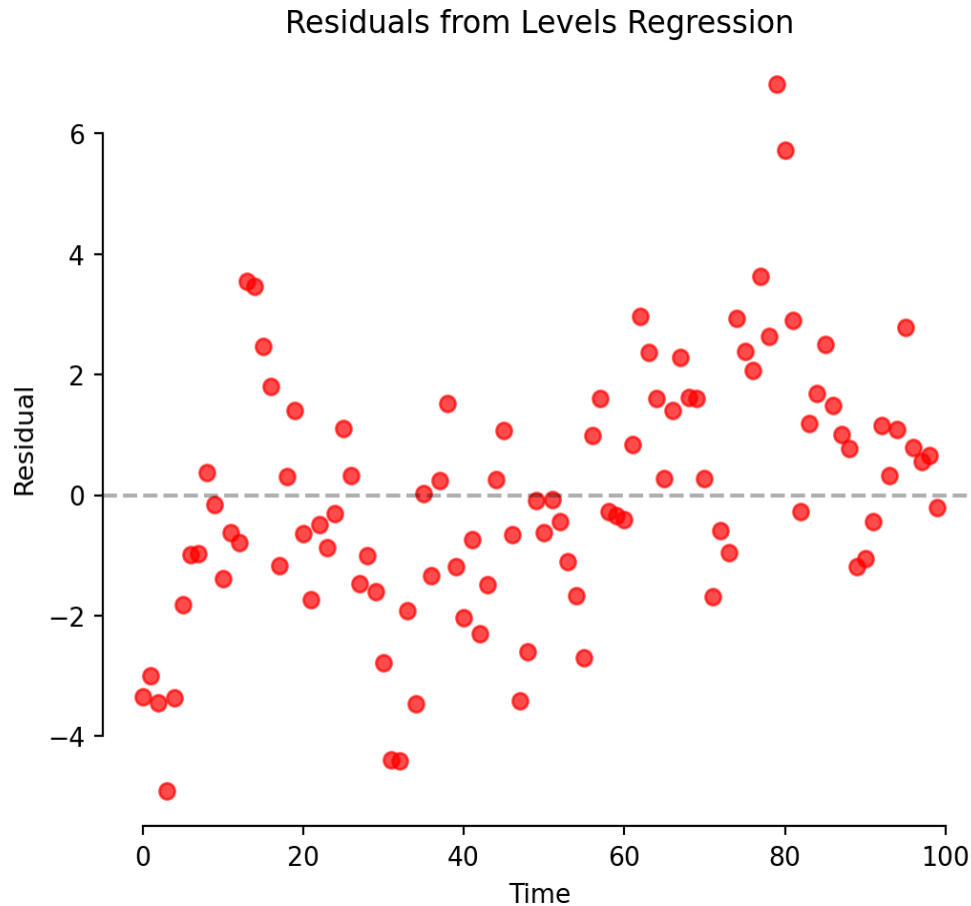
The standard approach has problems with time series.

$$Y = \beta_0 + \beta_1 \times t + \varepsilon$$



OLS Assumption: No Autocorrelation

The confidence level of regression requires that the error terms are independent.



> *levels regression shows strong patterns in residuals (autocorrelation)*

Model 1: Levels Regression

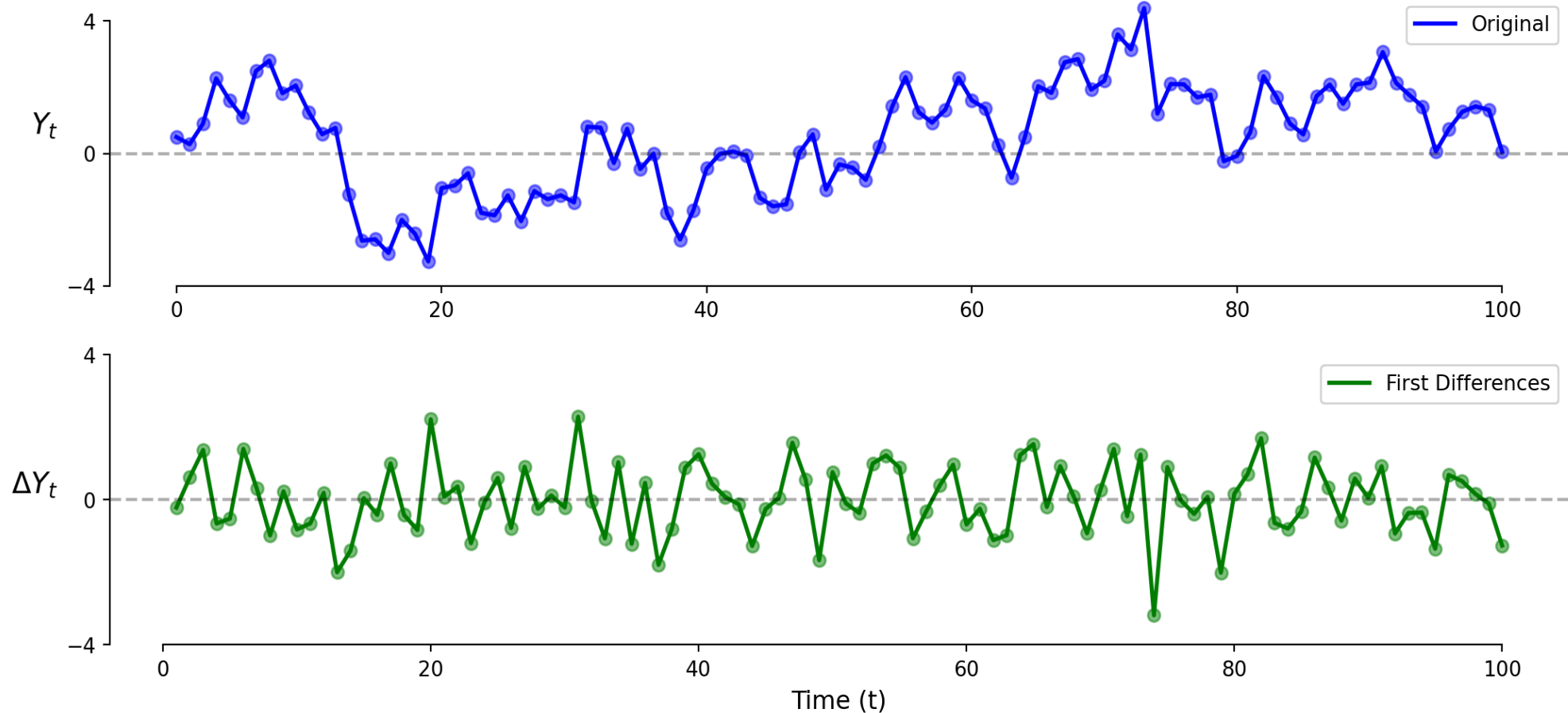
The standard approach has problems with time series.

$$Y = \beta_0 + \beta_1 \times t + \varepsilon$$

- > common trends can create spurious correlations*
- > error terms are serially correlated, violating regression assumptions*
- > potentially misleading significance levels due to violated assumptions*
- > differencing substantially reduces the autocorrelation problem*

Model 2: First Differences

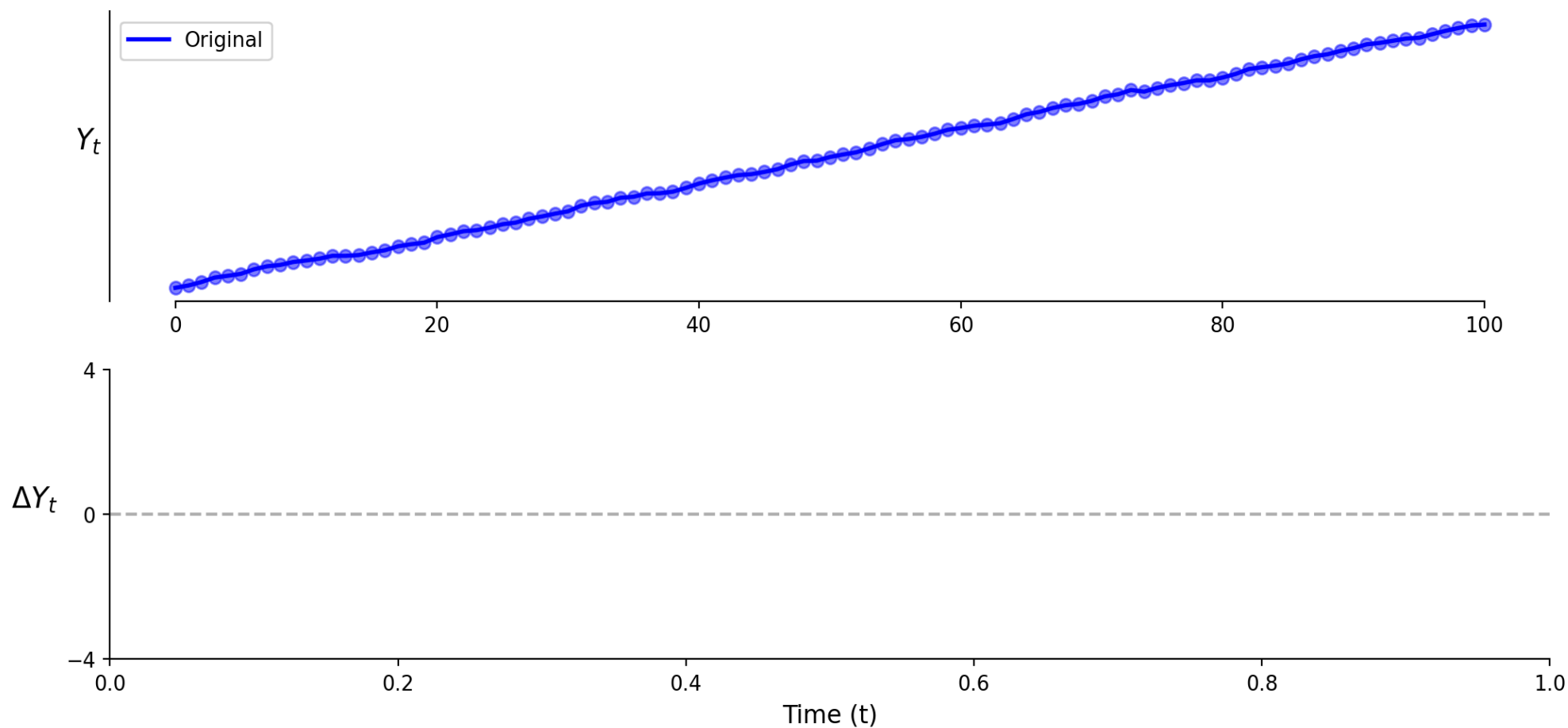
Focusing on changes rather than levels: $\Delta Y_t = Y_t - Y_{t-1}$



- > *differences (correctly in this case) shows no relationship*
- > *what would the first differences look like if there was a positive trend?*

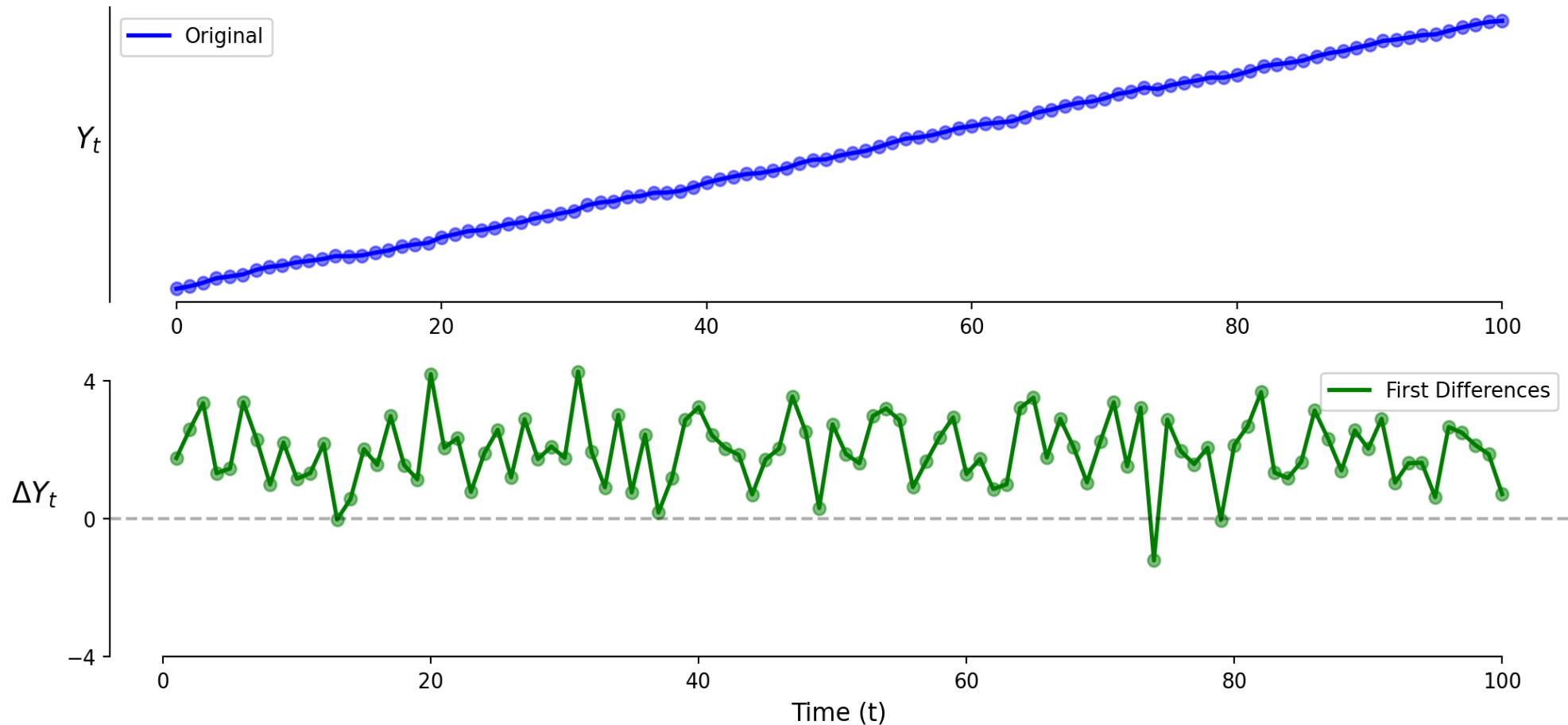
Model 2: First Differences

Focusing on changes rather than levels: $\Delta Y_t = Y_t - Y_{t-1}$



Model 2: First Differences

Focusing on changes rather than levels: $\Delta Y_t = Y_t - Y_{t-1}$

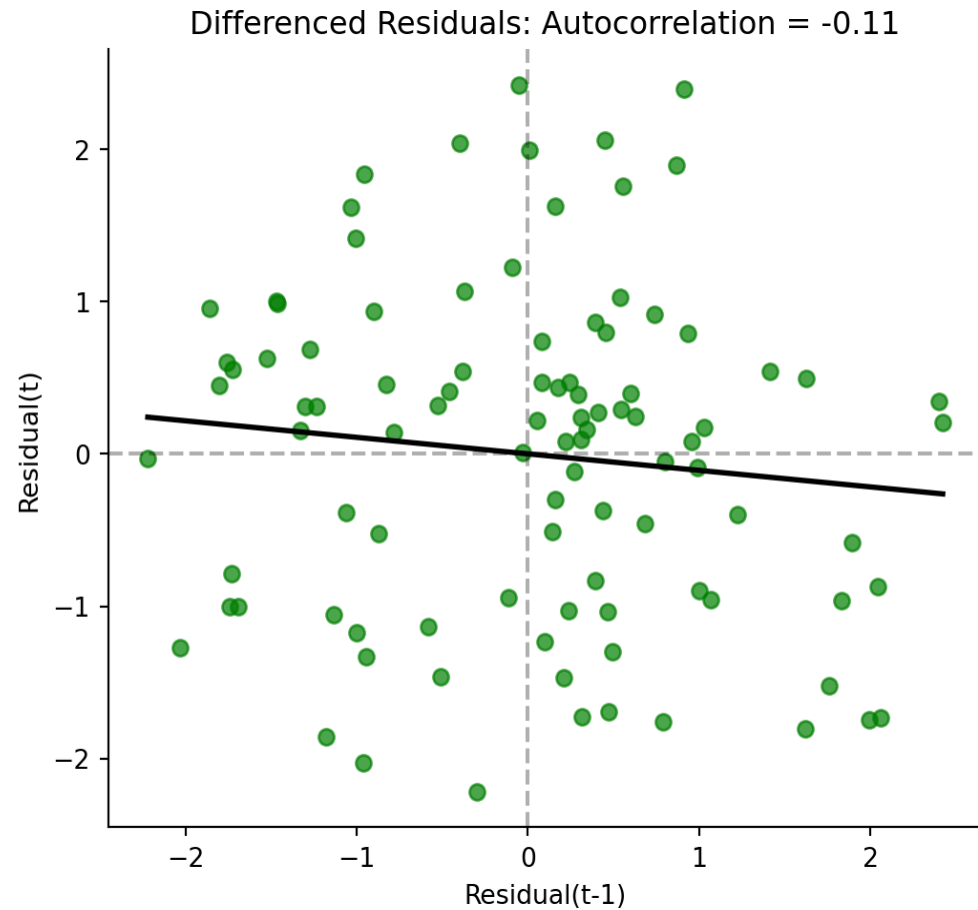
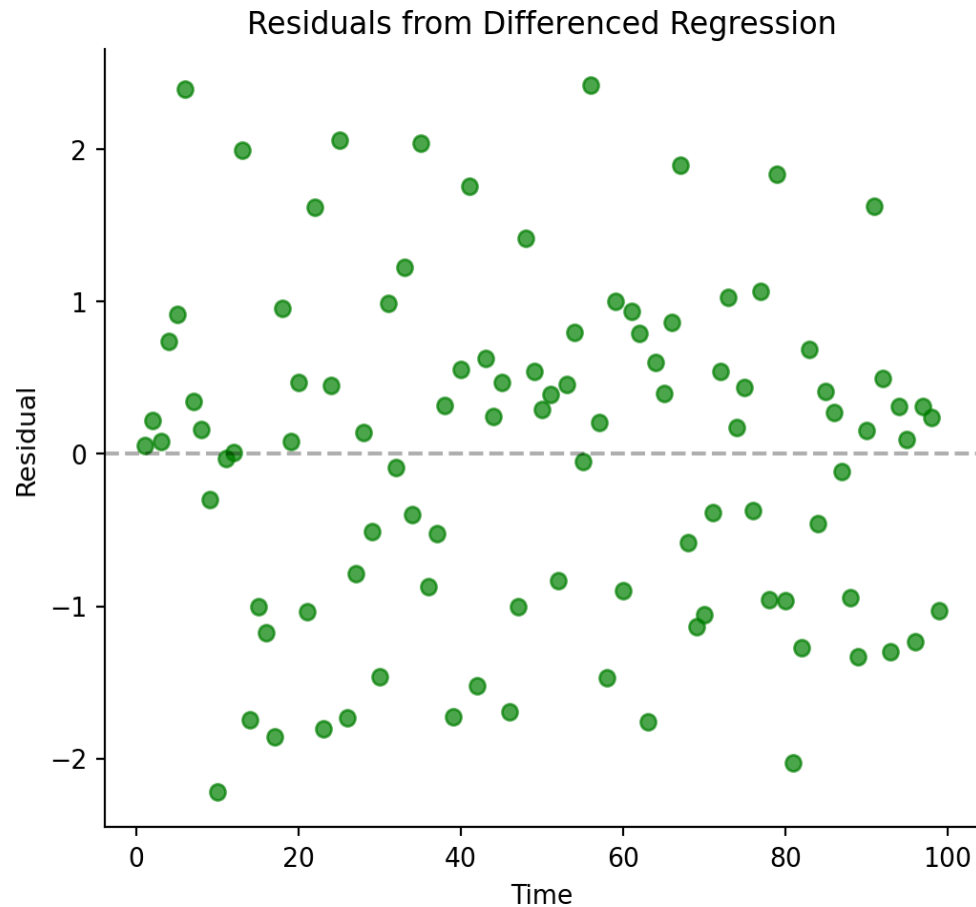


> *the vertical intercept is positive!*

> *differences (correctly in this case) shows the relationship as an intercept*

The Time Series Challenge

The problem of autocorrelated residuals.



> *differencing substantially reduces the autocorrelation problem*

Model 2: The Code

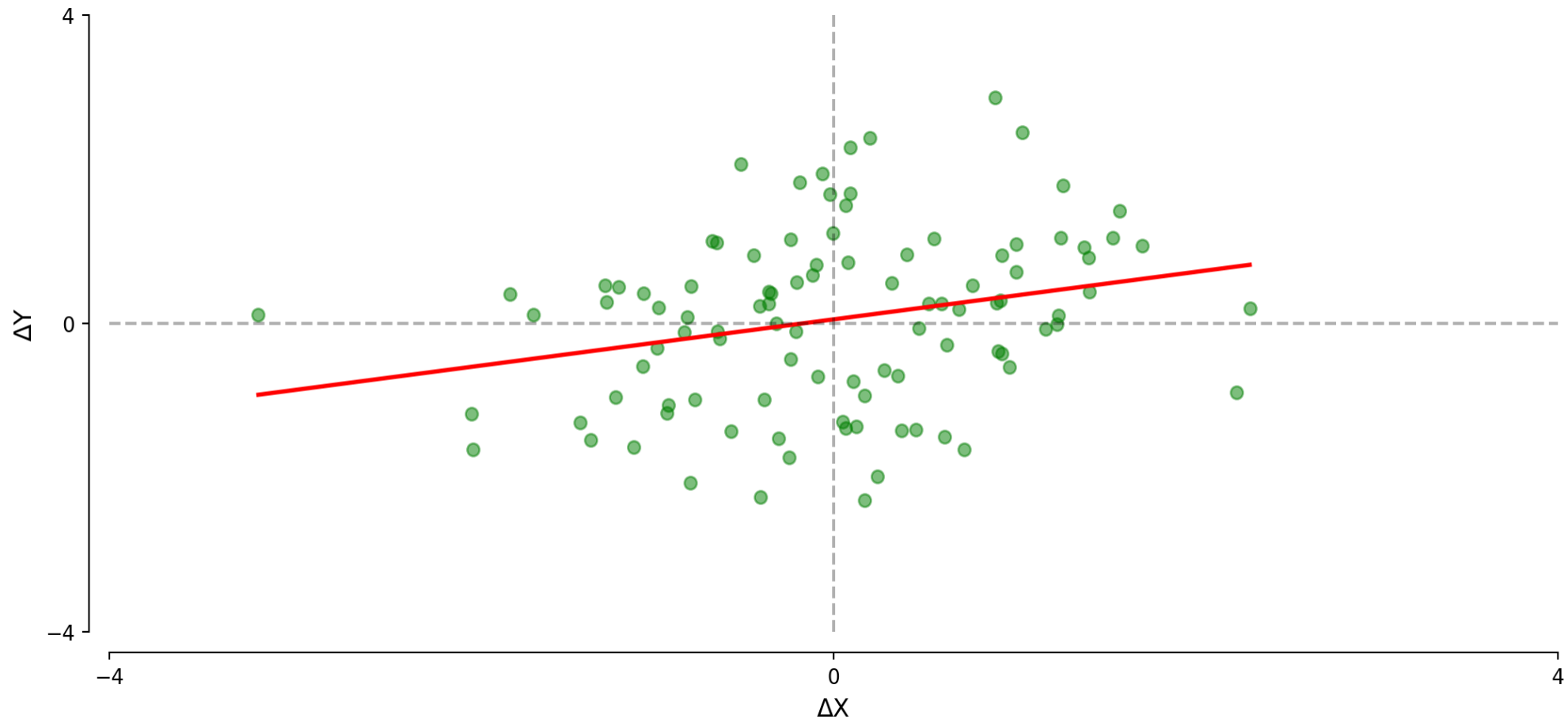
Implementing a first differences regression

```
1 # 1. Create first differences variables
2 data['gdp_diff'] = data['gdp'].diff()
3 data['unemployment_diff'] = data['unemployment'].diff()
4
5 # 2. Drop the first row which has NaN due to differencing
6 data = data.dropna()
7
8 # 3. Fit the differences model
9 model2 = smf.ols('gdp_diff ~ unemployment_diff', data=data).fit()
10 print(model2.summary().tables[1])
```

Model 3: First Differences Regression

Relating changes in X to changes in Y through time (t).

> *we can also relate two time series variables X and Y using differences*



$$\Delta Y_t = \beta_0 + \beta_1 \times \Delta X_t + \varepsilon_t$$

Model 3: First Differences Regression

Relating changes in X to changes in Y .

$$\Delta Y_t = \beta_0 + \beta_1 \times \Delta X_t + \varepsilon_t$$

- > removes trends that could cause spurious correlations*
- > reduces serial correlation in the error terms*
- > β_0 captures time trend in Y*
- > clear interpretation: how do changes in X relate to changes in Y ?*
- > β_1 captures the short-term relationship between variables*

Model 3: The Code

Implementing a first differences regression

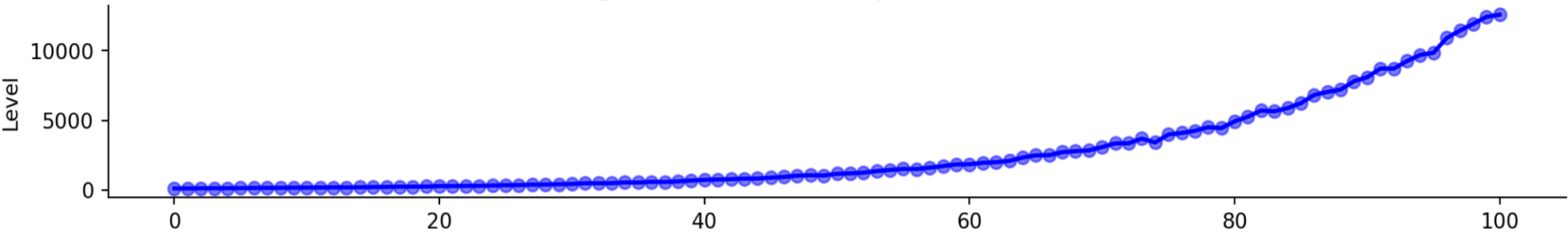
```
1 # Create first differences variables
2 data['gdp_diff'] = df['gdp'].diff()
3 df['unemployment_diff'] = df['unemployment'].diff()
4
5 # Drop the first row which has NaN due to differencing
6 data = data.dropna()
7
8 # Fit the differences model
9 model3 = smf.ols('gdp_diff ~ unemployment_diff', data=data).fit()
10 print(model3.summary())
```

> β_1 now represents the short-term impact of changes in X on changes in Y

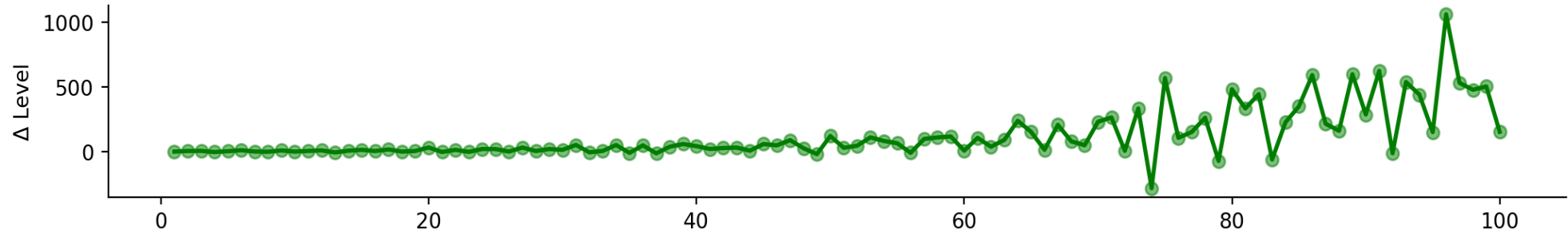
Model 4: Growth Rates

Proportional changes provide interpretable coefficients: $g_Y = \frac{Y_t - Y_{t-1}}{Y_{t-1}} = \frac{\Delta Y_t}{Y_{t-1}}$

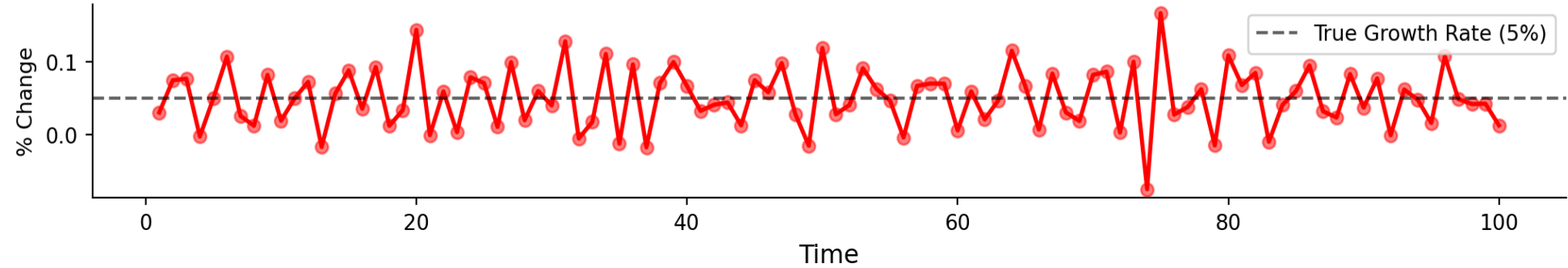
Original Series with Exponential Growth



Absolute Differences (Growing Over Time)

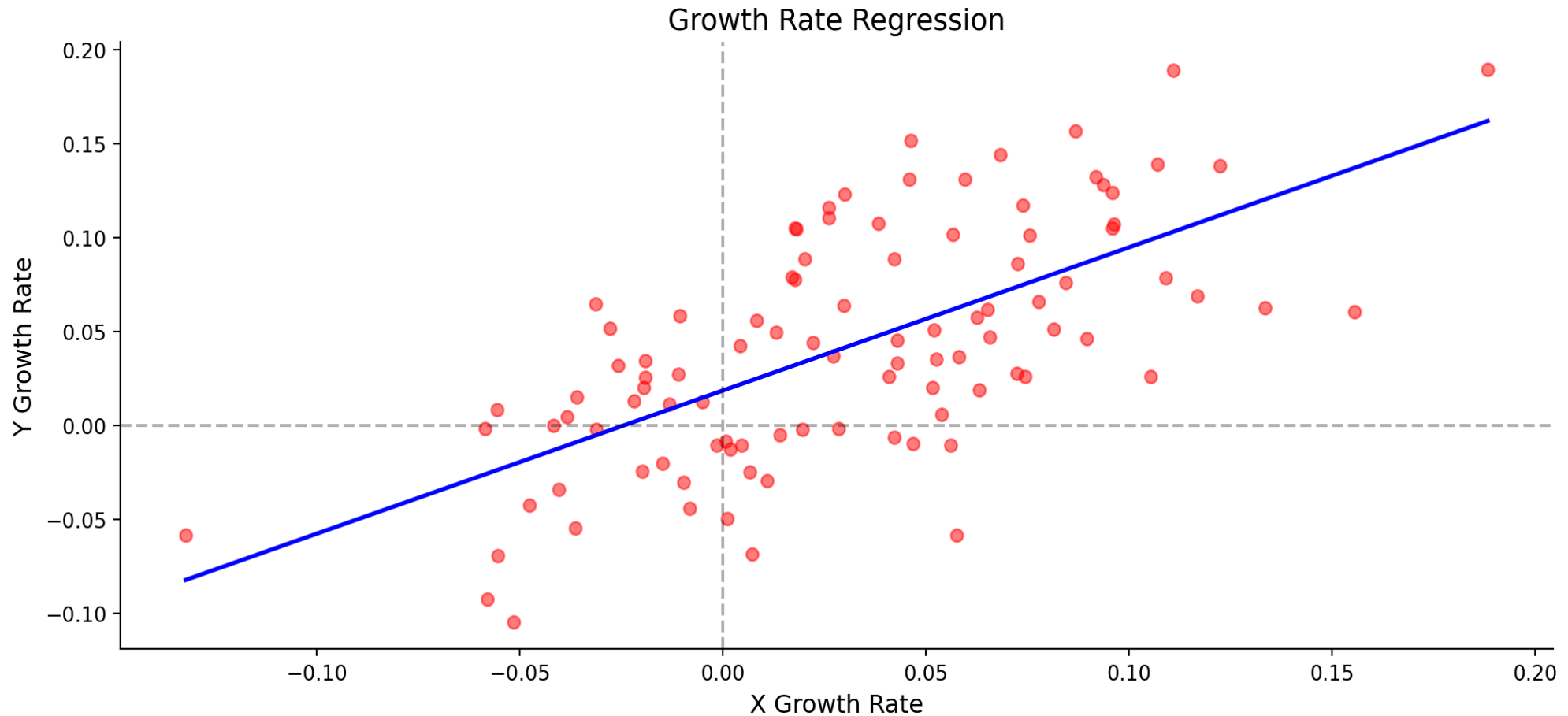


Percentage Changes (Stationary Around Growth Rate)



Model 4: Growth Rate Regression

Relating growth in X to growth in Y .



$$g_Y = \beta_0 + \beta_1 \times g_X + \varepsilon_t$$

Model 4: Growth Rate Regression

Relating growth in X to growth in Y .

$$g_Y = \beta_0 + \beta_1 \times g_X + \varepsilon_t$$

- > advantages of first differences plus better scale properties*
- > natural for variables with exponential growth*
- > β_0 is Y 's baseline growth rate*
- > β_1 is how Y 's growth responds to a 1 percentage point increase in X 's growth*

Model 4: The Code

Implementing a growth rates regression

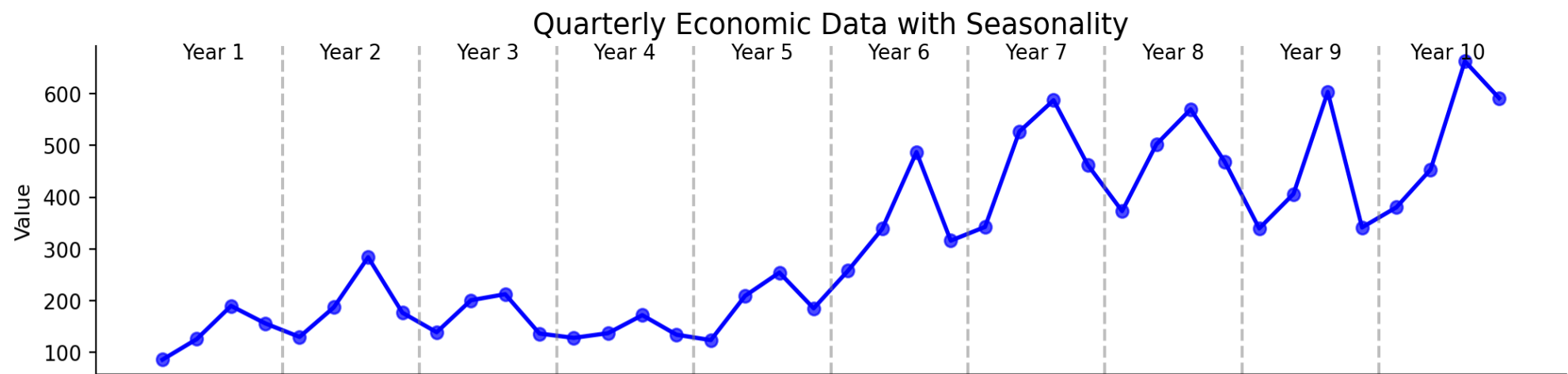
```
1 # Calculate growth rates (percentage changes)
2 data['gdp_growth'] = data['gdp'].pct_change() # in percentage points
3 data['unemployment_growth'] = data['unemployment'].pct_change()
4
5 # Drop rows with NaN values
6 data = data.dropna()
7
8 # Fit the growth rate model
9 model4 = smf.ols('gdp_growth ~ unemployment_growth', data=data).fit()
10 print(model4.summary())
```

> β_1 is now expressed in percentage point terms

> easier to interpret for policy-relevant questions

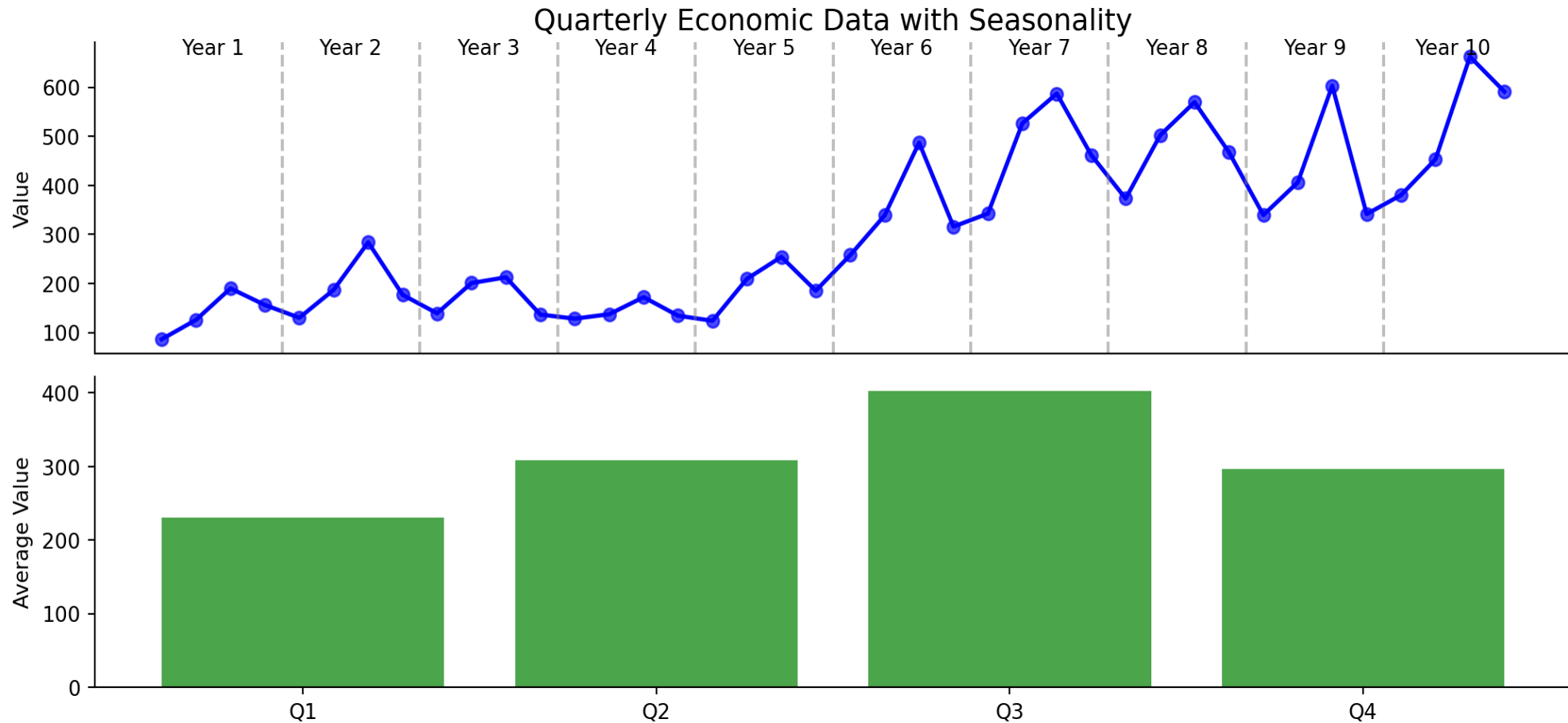
Seasonal Patterns in Economic Data

Many economic variables follow seasonal patterns.



Seasonal Patterns in Economic Data

Many economic variables follow seasonal patterns.

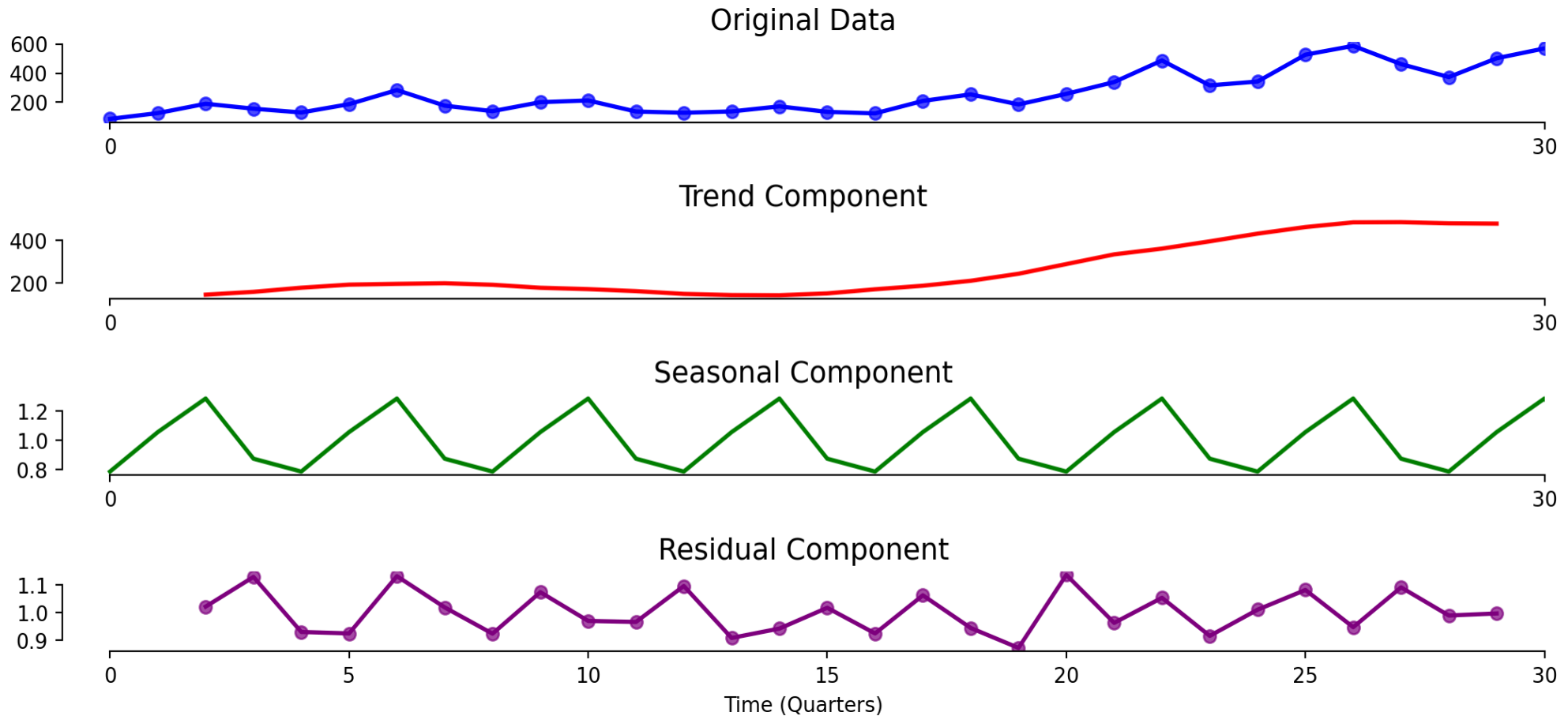


> *there are regular spikes in Q3!*

> *but there is also an increase over time*

Model 5: Deseasonalization

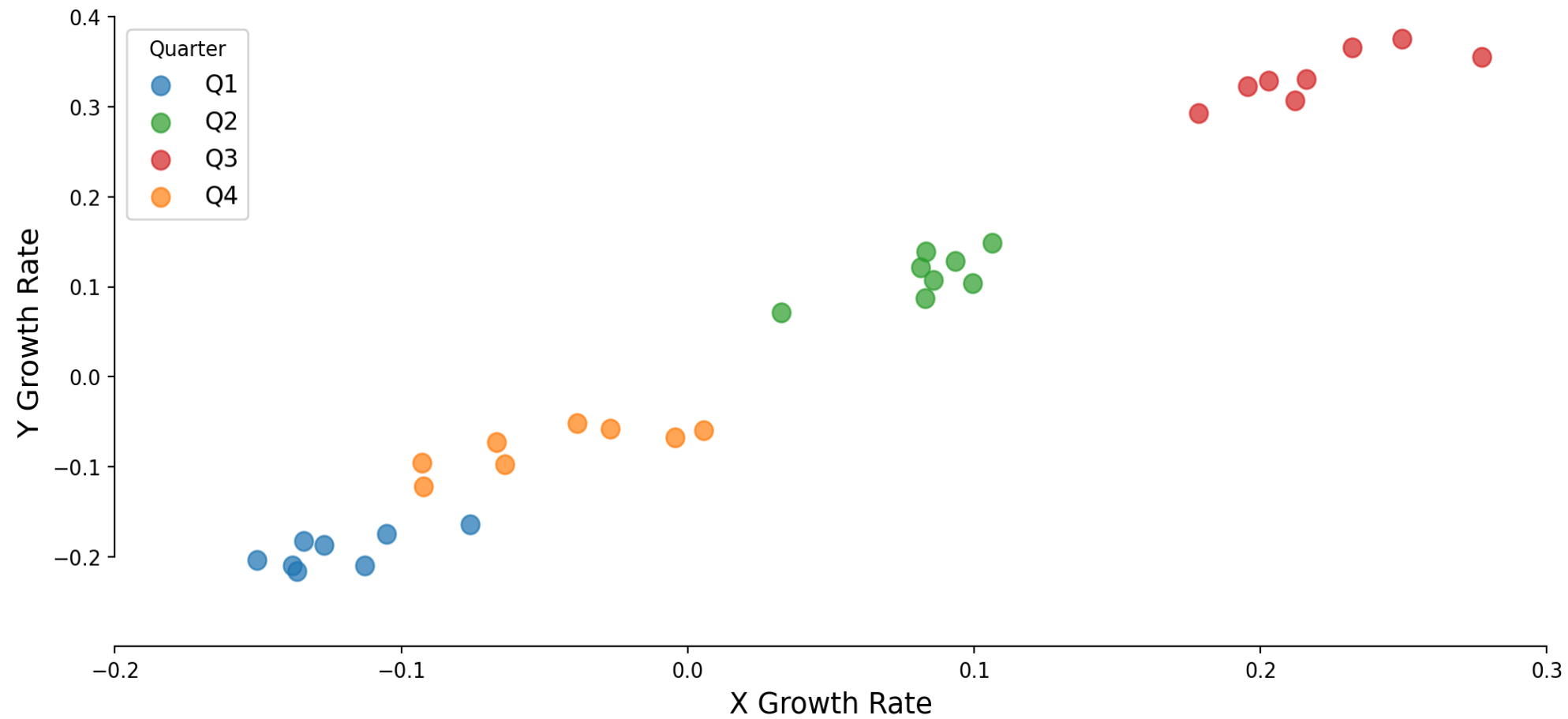
We can remove seasonal patterns to see the trend more clearly.



> *seasonal decomposition separates trend, seasonal, and residual components*

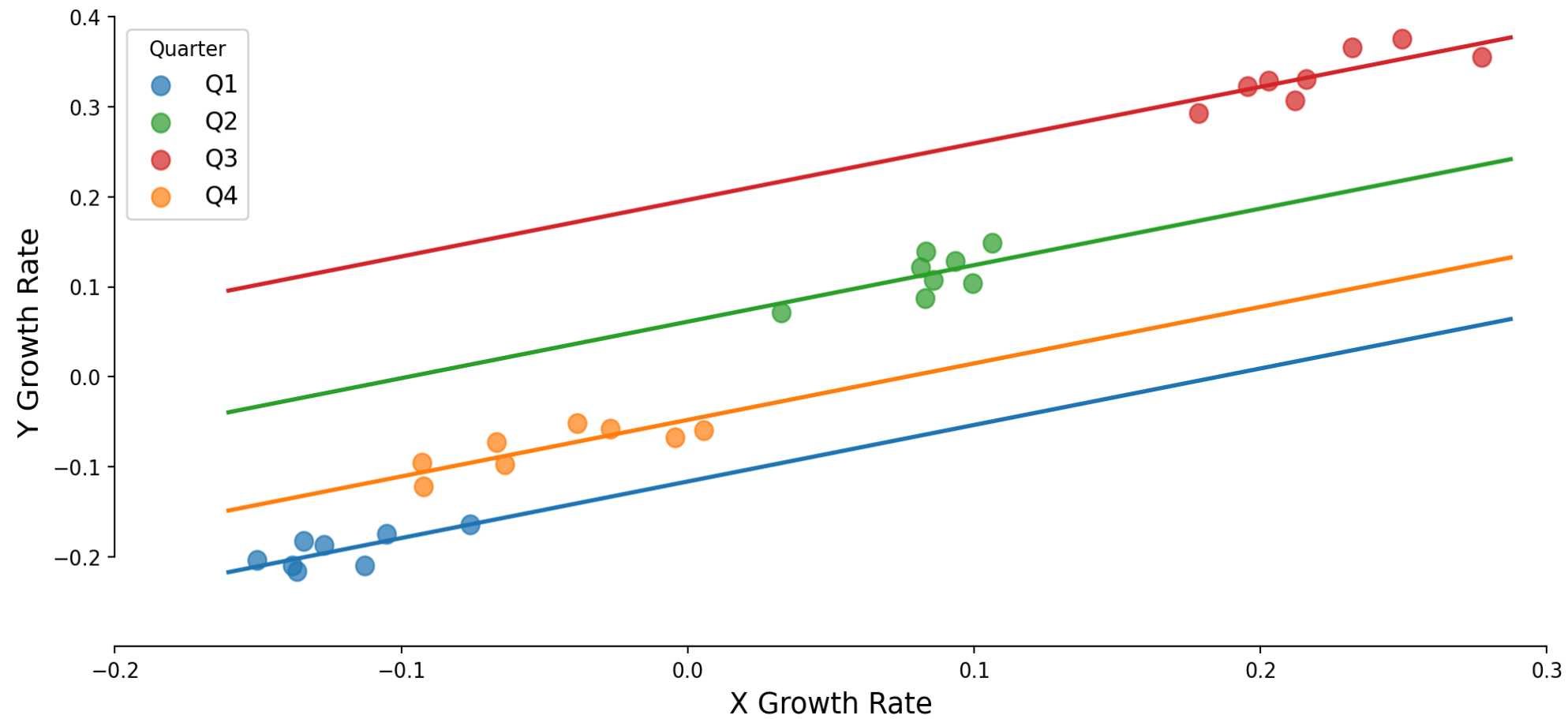
Model 5: Deseasonalization with Fixed Effects

Using seasonal dummies to adjust for quarterly patterns.



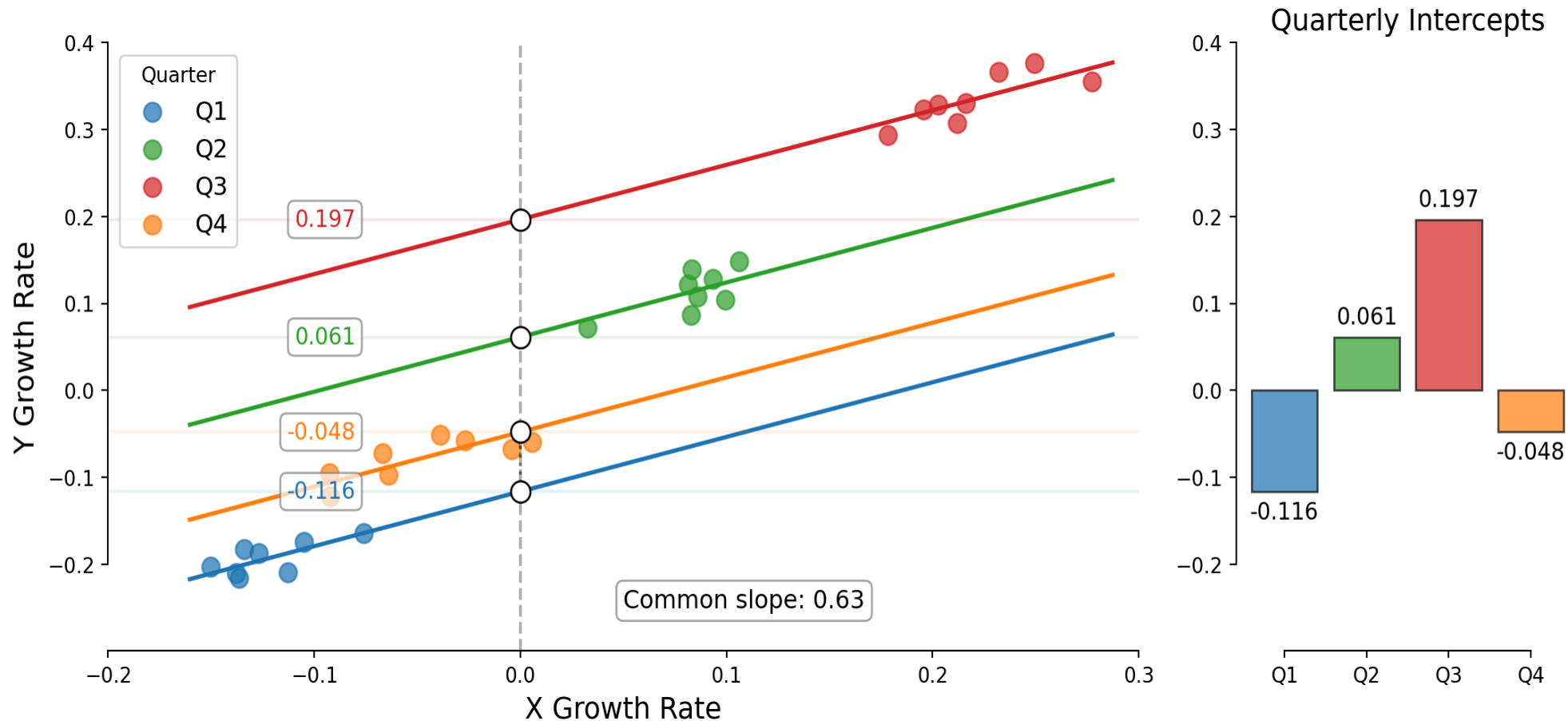
Model 5: Deseasonalization with Fixed Effects

Using seasonal dummies to adjust for quarterly patterns.



Model 5: Deseasonalization with Fixed Effects

Using seasonal dummies to adjust for quarterly patterns.



> *using these fixed effects deseasonalizes the data*

> *the slope captures the trend consistent across quarters*

Model 5: Implementing Seasonal Fixed Effects

Deseasonalizing data through regression.

```
1 # Run regression with seasonal dummies using C() notation  
2 model5 = smf.ols('gdp_growth ~ unemployment_growth + C(quarter)', data=data).fit()
```

- > *deseasonalized data removes the average effect of each season*
- > *relationship between variables is now clearer without seasonal distortions*

Key Takeaways

Best practices for time series analysis in economics.

1. Use differences or growth rates

- *Reduces serial correlation and removes trends*
- *First differences focus on period-to-period changes*

2. Address seasonality explicitly

- *Seasonal dummies, year-over-year comparisons*

3. Combine methods when appropriate

- *Growth rates of seasonally adjusted data*

> time series analysis requires special care but yields valuable economic insights