Homework 3 Fundamentals of Econometrics

Samuel Fraley Daniel Campos Elvis Casco

2026-05-10

1 Question 1

Consider assumptions [A1]-[A4] presented in class slide 1(3). If we wanted to present these same assumptions for random samples, that is, when $\{(x_i, y_i)\}$ are i.i.d (independently and identically distributed across observations), how would you write these 4 assumptions? Provide the expressions and briefly justify.

Original Assumptions (fixed-X sampling):

$$[A1]: \quad y_i = \beta_1 + \beta_2 x_{i2} + \cdots + \beta_K x_{iK} + \epsilon_i = x_i'\beta + \epsilon_i$$

$$[A2]: \quad E(\epsilon_i|X) = 0$$

$$[A3]: \quad \text{Var}(\epsilon_i|X) = \sigma^2$$

$$[A4]: \quad \text{Cov}(\epsilon_i, \epsilon_j|X) = 0 \quad \forall i \neq j$$

Answer

To adapt assumptions [A1]–[A4] for the case where observations $\{(x_i, y_i)\}$ are independently and identically distributed (*i.i.d.*), we shift from conditioning on the full design matrix X to conditioning on the individual regressor vector x_i. Here's how each assumption would be rewritten:

Assumption [A1]: Linearity

Original (fixed-X sampling):

$$y_i = x_i'\beta + \epsilon_i$$

i.i.d. Version:

$$y_i = x_i'\beta + \epsilon_i$$

Justification: This assumption remains unchanged. It states that the outcome y_i is a linear function of the covariates x_i plus an error term ϵ_i. Linearity is a structural assumption and does not depend on whether the data are *i.i.d.*

Assumption [A2]: Zero Conditional Mean (Strict Exogeneity)

Original (fixed-X sampling):

$$E(\epsilon_i|X) = 0 \quad \forall i = 1, \dots, n$$

i.i.d. Version:

$$E(\epsilon_i|x_i) = 0 \quad \forall i = 1, \dots, n$$

Justification: Under *i.i.d.* sampling, each observation (x_i, y_i) is independent of the others. Therefore, we condition only on the individual covariates x_i, not the entire matrix X. This assumption ensures that the regressors are exogenous and that the error term has no systematic relationship with the predictors.

Assumption [A3]: Conditional Homoskedasticity

Original (fixed-X sampling):

$$\mathrm{Var}(\epsilon_i|X) = \sigma^2 \quad \forall i = 1, \dots, n$$

i.i.d. Version:

$$\mathrm{Var}(\epsilon_i|x_i) = \sigma^2 \quad \forall i = 1, \dots, n$$

Justification: Again, we condition on x_i rather than the full matrix X. This assumption implies that the variance of the error term is constant across observations, regardless of the values of the covariates.

Assumption [A4]: No Autocorrelation

Original (fixed-X sampling):

$$\mathrm{Cov}(\epsilon_i, \epsilon_j|X) = 0 \quad \forall i \neq j$$

i.i.d. Version:

$$\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0 \quad \forall i \neq j$$

Justification: Under *i.i.d.* sampling, the errors are automatically uncorrelated across observations. Since the observations are independent, we do not need to condition on X to assert that the errors are uncorrelated. In fact, under *i.i.d.* sampling, the errors are independent (a stronger condition than uncorrelated), which implies $\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0$ for all $i \neq j$.

2 Question 2

Under linearity and strict exogeneity assumptions we saw that OLS estimator of the parameters of a linear regression model, $\hat{\beta}$, is conditionally unbiased:

$$(A): \quad E\left(\hat{\beta}_k \mid X\right) = \beta_k \quad \forall k = 1, \ldots, K$$

Additionally, under Gauss-Markov assumptions, we saw that the OLS estimator is the one with the smallest conditional variance among the class of all linear unbiased estimators:

$$(B): \quad \text{Var}(\hat{\beta}_k \mid X) \leq \text{Var}(\tilde{\beta}_k \mid X) \quad \forall k = 1, \ldots, K$$

where $\tilde{\beta}_k$ is any other linear unbiased estimator of regression parameter β_k.

Part (a)

(i) Departing from result (A), prove that OLS is also unconditionally unbiased.

To prove that the OLS estimator $\hat{\beta}_k$ is unconditionally unbiased, we start from the conditional unbiasedness result (A):

$$E(\hat{\beta}_k \mid X) = \beta_k \quad \forall k = 1, \ldots, K$$

We want to show that:

$$E(\hat{\beta}_k) = \beta_k$$

Proof:

By the law of iterated expectations:

$$E(\hat{\beta}_k) = E\left[E(\hat{\beta}_k \mid X)\right]$$

Substituting the conditional expectation from result (A):

$$E(\hat{\beta}_k) = E[\beta_k]$$

Since β_k is a fixed parameter (not a random variable), we have:

$$E[\beta_k] = \beta_k$$

Therefore:

$$E(\hat{\beta}_k) = \beta_k$$

which establishes that the OLS estimator is unconditionally unbiased.

(ii) In a sentence describe what this property is telling us.*

This property tells us that, on average across all possible samples, the OLS estimator correctly targets the true value of the parameter—it does not systematically overestimate or underestimate β_k.

Part (b)

Departing from result (B), prove that this inequality also holds for the unconditional version. That is, prove:

$$\text{Var}(\hat{\beta}_k) \leq \text{Var}(\tilde{\beta}_k) \quad \forall k$$

Proof:

We use the law of total variance, which states that for any random variables Z and W:

$$\text{Var}(Z) = \text{Var}\left[E(Z \mid W)\right] + E\left[\text{Var}(Z \mid W)\right]$$

Step 1: Apply the law of total variance to $\hat{\beta}_k$, conditioning on X:

$$\text{Var}(\hat{\beta}_k) = \text{Var}\left[E(\hat{\beta}_k \mid X)\right] + E\left[\text{Var}(\hat{\beta}_k \mid X)\right]$$

From result (A), we know that $E(\hat{\beta}_k \mid X) = \beta_k$, which is a constant. Therefore:

$$\text{Var}\left[E(\hat{\beta}_k \mid X)\right] = \text{Var}(\beta_k) = 0$$

Thus:

$$\text{Var}(\hat{\beta}_k) = E\left[\text{Var}(\hat{\beta}_k \mid X)\right]$$

Step 2: Apply the law of total variance to any other linear unbiased estimator $\tilde{\beta}_k$:

$$\text{Var}(\tilde{\beta}_k) = \text{Var}\left[E(\tilde{\beta}_k \mid X)\right] + E\left[\text{Var}(\tilde{\beta}_k \mid X)\right]$$

Since $\tilde{\beta}_k$ is also conditionally unbiased (by definition of a linear unbiased estimator):

$$E(\tilde{\beta}_k \mid X) = \beta_k$$

Therefore:

$$\text{Var}\left[E(\tilde{\beta}_k \mid X)\right] = 0$$

Thus:

$$\text{Var}(\tilde{\beta}_k) = E\left[\text{Var}(\tilde{\beta}_k \mid X)\right]$$

Step 3: Apply the Gauss-Markov result (B):

From result (B), we know that:

$$\text{Var}(\hat{\beta}_k \mid X) \leq \text{Var}(\tilde{\beta}_k \mid X) \quad \forall X$$

Taking expectations on both sides:

$$E\left[\text{Var}(\hat{\beta}_k \mid X)\right] \leq E\left[\text{Var}(\tilde{\beta}_k \mid X)\right]$$

Combining the results from Steps 1 and 2:

$$\text{Var}(\hat{\beta}_k) \leq \text{Var}(\tilde{\beta}_k) \quad \forall k$$

This establishes that OLS has the smallest unconditional variance among all linear unbiased estimators, completing the proof.

3 Question 3

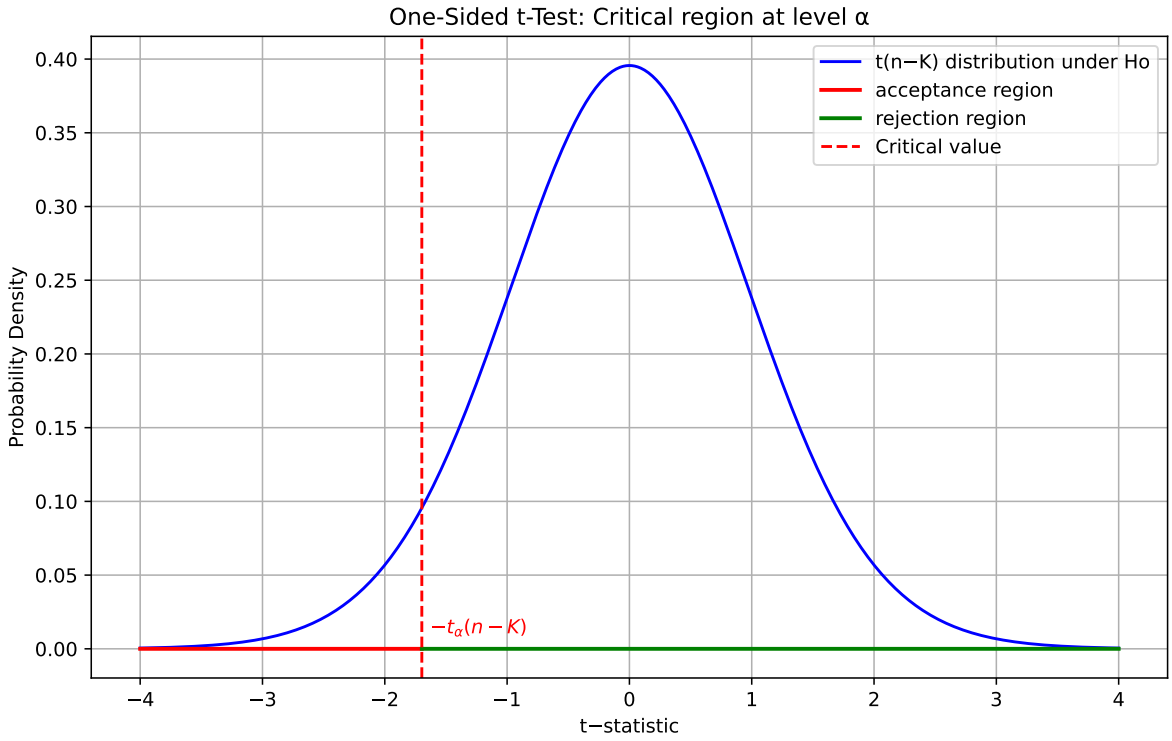
Consider performing the following test:

$$H_0 : \beta_2 = 0 \quad \text{versus} H_1 : \beta_2 < 0,$$

using the t-test statistic. This test is called one-sided test since its rejection region is all in one of the tails of the distribution.

3 (a)

Draw the acceptance and rejection regions for this test using $\alpha = 5$.



3 (b)

Explain the intuition behind the location of the critical region.

Answer:

The t-test at significance level α (one-side version):

- Test: $H_0 : \beta_k = r$ vs $H_1 : \beta_k \neq r$

Test statistic: $t \equiv \dfrac{\widehat{\beta}_k - r}{se(\widehat{\beta}_k)} \underset{under H_0}{\sim} t(n-K)$

Critical value at significance level α, from $t(n-K)$ distribution tables:

$-t_\alpha(n-K)$

We should use the one-sided critical values only when the parameter space is known to satisfy a one-sided restriction such as $\beta \leq 0$. See Hansen, pg. 225.

The intuition behind the location of the critical region in a one-sided t-test like:

$H_0 : \beta_2 = 0 \quad \text{vs.} \quad H_1 : \beta_2 < 0$

must be in the direction of the alternative hypothesis.

The alternative hypothesis $H_1 : \beta_2 < 0$ suggests we're testing whether the parameter is significantly less than zero. The t-statistic measures how far the estimated coefficient $\widehat{\beta}_2$ is from zero, in standard error units. If $\widehat{\beta}_2$ is much smaller than zero, the t-statistic will be strongly negative.

4 Question 4

Consider the code included in file Assig3Q4.R or file Assig3Q4.py. Both codes are reproduced at the end of the assignment, after the instructions.

4 (a)

Provide the expression of the dgp behind any generated sample. Be careful with the notation.

Answer:

The data generating process (DGP) behind any simulated sample is given by the following linear model:

$y_i = \beta_1 + \beta_2 x_{i2} + \cdots + \beta_k x_{iK} + \epsilon_i$

For this particular case, we have that the functional relationship between y_i, x_i and the disturbance ϵ_i is:

$y_i = \beta_1 + \beta_2 x_{i2} + \epsilon_i$

where:

- $\beta_1 = 10$
- $\beta_2 = 5$
- $x_{i2} \sim \text{Uniform}(0, 20)$, 50 observations
- $\epsilon_i \sim \mathcal{N}(0, 6^2)$, independently across $i = 1, \ldots, 50$

So the full expression becomes:

$y_i = 10 + 5x_{i2} + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2 = 36)$

This DGP is used to simulate 100 independent samples of size 50, each time generating new y_i values based on the same x_{i2} vector and fresh random noise ϵ_i.

In the end we are simulating dgp under Gauss-Markow assumptions plus normality:

- for observation i:

$y_i = 10 + 5x_{i2} + \epsilon_i, \quad \epsilon_i | X \sim ii\mathcal{N}(0, \sigma^2 = 36)$

4 (b) (i)

How many samples does the code file generate?

Answer:

The code simulates a DGP for 100 samples. This is controlled by the line:

```
M = 100
```

Each iteration of the for loop simulates one sample of size 50, fits a linear regression model, and computes a confidence interval for β_2. So in total, it produces 100 confidence intervals -one for each simulated sample.

```
Number of simulations:
100

Sample size:
50
```

4 (b) (ii)

What do all the samples have in common?

Answer:

All the samples in the code share the same underlying data generating process (DGP) and same predictor values. Specifically:

- They all use the same vector of covariates: $x_{i2} \sim \text{Uniform}(0, 20)$
- This vector is generated once and reused across all 100 samples.
- They all follow the same linear model:

$y_i = 10 + 5x_{i2} + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, 36)$ is independently drawn for each sample. - Each sample has the same size: 50 observations.

So while the noise term ϵ_i varies across samples (introducing randomness), the structure of the model and the covariates remain constant, allowing us to assess how often the confidence intervals correctly capture the true value $\beta_2 = 5$.

Then all the samples are generated with a dgp under Gauss-Markow assumptions plus normality; that is, for observation i:

$y_i = \beta_1 + \beta_2 x_{i2} + \epsilon_i$

$\epsilon_i | X \sim i.i.N(0, \sigma^2)$

4 (b) (iii)

Describe what the R code from line 1 to line 16 (or from line 6 to 30 in Python) does. Be specific.

Answer:

This code performs a simulation study to construct 100 confidence intervals for the slope coefficient (the intercept of the regression is a constant = 10) in a simple linear regression model.

- `np.random.seed(1010)`: Ensures reproducibility of random numbers.
- `M = 100`: Sets the number of simulated samples to 100.
- `lower` and `upper`: Initialize vectors to store the lower and upper bounds of confidence intervals for each sample.
- `x2`: Generates a fixed vector of 50 random values from a uniform distribution between 0 and 20.
- `for i in range(M):` begins the iteration. The loop runs 100 times, simulating 100 samples.

In each iteration:

- `y = 10 + 5 * x2 + np.random.normal(0, 6, 50)`: Generates a new response vector y using the DGP: $y_i = 10 + 5x_{i2} + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, 6^2)$
- `X = sm.add_constant(x2)`: creates the matrix X
- `model = sm.OLS(y, X).fit()`: Fits a linear regression model
- `b2 = model.params[1]`: Extracts the estimated slope coefficient
- `varb2 = model.cov_params()[1, 1]`: Computes the variance error of `b2`
- `se2 = np.sqrt(varb2)`: Computes the standard error of `b2`
- `t_val = t.ppf(0.975, 48)`: Computes the critical t-value for a 95% confidence interval with $n - 2 = 48$ degrees of freedom.
- `lower[i] = b2 - t_val * se2` and `upper[i] = b2 + t_val * se2` Calculates and stores the lower and upper bounds of the confidence interval for β_2.

- `CIs = np.column_stack((lower, upper))`: Combines the lower and upper bounds into a single array CIs of shape (100, 2), where each row is a confidence interval

At the end of the process:

- You have 100 confidence intervals stored in lower and upper.
- These intervals reflect the uncertainty around the estimate of $\beta_2 = 5$ across repeated samples.

To build the confidence interval we use the test statistic:

$$t \equiv \frac{\hat{\beta}_k - r}{se(\hat{\beta}_k)} \sim_{under H_0} t(n-K)$$

where $se(\hat{\beta}_k) \equiv \sqrt{\widehat{var}(\hat{\beta}_k|X)}$

A confidence interval for β_k at $100(1-\alpha)\%$ is given by all values r such that:

$$r \in [\hat{\beta}_k - t_{\frac{\alpha}{2}}(n-K) \cdot se(\hat{\beta}_k), \hat{\beta}_k + t_{\frac{\alpha}{2}}(n-K) \cdot se(\hat{\beta}_k)]$$

4 (c)

Describe what does R code from line 18 to line 23 (or from line 32 to 39 in Python) do? Be specific.

Answer:

This code evaluates how many of the confidence intervals constructed in your simulation actually contain the true slope value of 5, and calculates the coverage rate — the percentage of intervals that successfully capture the true parameter.

- `IDg = np.where((lower <= 5) & (upper >= 5))[0]`: Finds the indices of intervals where 5 (the true slope value) lies between the lower and upper bounds; `length_IDg = len(IDg)` Counts how many intervals contain the true value
- `IDb = np.where(~((lower <= 5) & (upper >= 5)))[0]`: Finds the indices of intervals where the lower and upper bounds do not contain 5 (the true slope value); `length_IDb = len(IDb)` Counts how many intervals do not contain the true value
- `ratiog = (length_IDg / M) * 100`: Computes the proportion of intervals that contain the true value; `print("Ratio", ratiog)` prints this ratio

This block of code checks how often the simulated confidence intervals actually include the true parameter value ($\beta_2 = 5$). It's a practical way to assess the reliability of the interval estimation procedure.

From this case, we have

```
t_val:  2.010634757624232
se2:  0.16023406437362797
t_val * se2:  0.32217217918501506
```

$se(\hat{\beta}_k) \equiv \sqrt{\widehat{var}(\hat{\beta}_k|X)}$

$5 \in [\hat{\beta}_k - t_{0.025}(48) \cdot se(\hat{\beta}_k), \hat{\beta}_k + t_{0.025}(48) \cdot se(\hat{\beta}_k)]$

$5 \in [\hat{\beta}_k - 2.0106 \cdot se(\hat{\beta}_k), \hat{\beta}_k + 2.0106 \cdot se(\hat{\beta}_k)]$

4 (d)

Describe what does R code from line 25 to line 40 (or from line 41 to 56 in Python) do? Be specific.

Answer:

This code creates a visualization of the 100 confidence intervals for β_2, highlighting which intervals contain the true value (5) and which do not.

- `plt.figure()`: Initializes an empty plot with: `plt.xlim([4, 6])` x-axis from 4 to 6 (range of confidence intervals); `plt.ylim([0, 100])` y-axis from 1 to 100 (one row per sample); `plt.xlabel(r'$\beta_2$')` and `plt.ylabel('Samples')` labels the x- and y-axes and adds a title.
- `plt.axvline(x=5, color='black', linestyle='--')`: Draws a vertical dashed line at $\beta_2 = 5$, the true value, to visually assess which intervals include it.

To create a vector of colors for each interval:

- `colors = ['gray'] * 100` - Initializes all interval colors as gray and `colors = np.array(colors)` converts to NumPy array for indexing
- `colors[IDb[IDb < 100]] = 'red'`: changes to color red for intervals that miss the true value (those in IDb).

```
for j in range(100):
    plt.plot([CIs[j, 0], CIs[j, 1]], [j, j], color=colors[j], lw=2)
```

- Loops through all 100 intervals
- Plots a horizontal line for each interval from its lower to upper bound
- Each line is placed at a different vertical position (j) to separate them visually
- The lines highlight missed intervals in red

This plot provides a visual summary of:

- How many intervals captured the true $\beta_2 = 5$ (gray lines)
- Which intervals failed (lines in red)

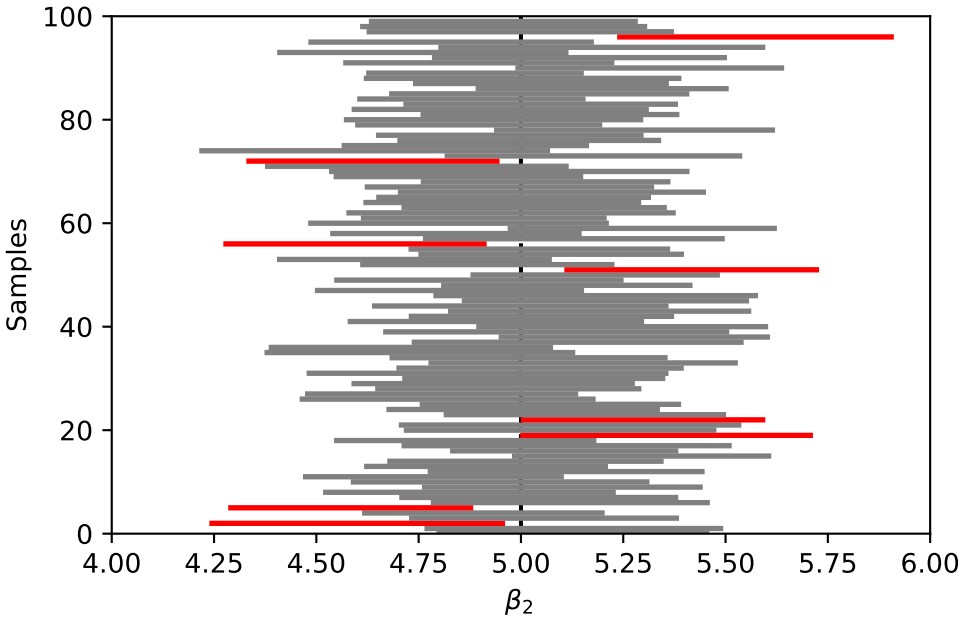
- The coverage performance of the confidence interval procedure

4 (e)

Run the entire code file (your choice, ideally try both). Include the value of the variable 'ratiog' and the plot. Surprised by the value of 'ratiog'? Rigorously comment.

Answer:

```
Ratio 92.0
```



Surprised? Not really:

- The confidence level is 95%, meaning we expect on average 95 out of 100 intervals to contain the true parameter.
- The observed coverage of 92% is well within the expected statistical fluctuation due to random sampling.
- This is a classic demonstration of the frequentist interpretation of confidence intervals: over many repetitions, about 95% of intervals should contain the true value.

This code is useful to observe:

- Statistical Validity: The simulation confirms that the confidence interval procedure is well-calibrated. A 92% coverage rate is consistent with the nominal 95% level.
- Random Variation: The slight deviation from 95% is due to sampling variability.

- Model Assumptions: The linear model assumes homoscedasticity and normal errors. The simulation uses `rnorm` with `sd = 6`, which is reasonable and supports valid inference.

4 (f)

What does the plot illustrate? Rigorously comment.

Answer:

The plot illustrates the empirical coverage performance of 95% confidence intervals for the slope parameter β_2 in a simple linear regression model, based on repeated sampling; calculates 100 confidence intervals for β_2 using a simulated sample with 50 observations.

- Each horizontal line represents a confidence interval for β_2 from one of 100 simulated samples.
- The dashed vertical line at $\beta_2 = 5$ marks the true value used in the data generating process.
- Gray lines indicate intervals that successfully contain the true value.
- Red lines indicate intervals that fail to contain the true value.
- The plot visually confirms that most intervals include the true value, consistent with the nominal 95% confidence level.

```
Intervals containing the true value:  92

Intervals missing the true value:  8
```

- The presence of 8 red intervals (out of 100) aligns with the theoretical expectation that, under correct model assumptions, approximately 5% of intervals will miss the true value purely due to sampling variability. This difference (8 vs 5) is due to the number of iterations, as will see in 4g.
- This validates the frequentist interpretation of confidence intervals: If we repeated the experiment many times, about 95% of the intervals would contain the true parameter.
- If we increase the sample to higher values, the

The reliability of the intervals -and the accuracy of the coverage rate—depends on the following assumptions being satisfied in the simulation:

- Linearity: The model $y_i = 10 + 5x_{i2} + \epsilon_i$ is correctly specified.
- Independence: Observations are independent across samples.
- Homoskedasticity: Constant variance of errors ($\epsilon_i \sim \mathcal{N}(0, 36)$).
- Normality: Errors are normally distributed, justifying the use of the t-distribution for interval construction.

With 100 simulations, you get a rough estimate of the true coverage probability. Due to random variation, you might see values like 94%, 97%, or even 92% — all within the expected range for a 95% CI.

4 (g)

If you increased the value of M at the tope of the code file, say from 100 to 10,000, how would you expect the value of 'ratiog' to change? Rigorously argue.

Answer:

```
Number of simulations:
10000

Sample size:
50
```

If you increase M from 100 to 10,000, you're dramatically increasing the number of simulated samples — and here's what that means for ratiog, both intuitively and rigorously:

What Is ratiog?

ratiog is the percentage of confidence intervals (CIs) that contain the true slope value $\beta_2 = 5$ across M simulations. Each simulation fits a linear model and computes a 95% CI for the slope.

So:

```
ratiog = (number of intervals containing 5 / M) × 100
```

Now you're running 10,000 simulations, which means:

- Law of Large Numbers kicks in: the empirical coverage (ratiog) will converge to the theoretical coverage of 95%.
- Random fluctuations shrink: the standard error of the estimate decreases.

The standard error (SE) of a proportion is:

$SE = \sqrt{\frac{p(1-p)}{M}}$

So with M = 10,000, you'd expect ratiog to fall within a very tight band around 95%, say between 94.5% and 95.5%.

Rigorously Interpreted

- Theoretical expectation: The confidence interval procedure is designed to capture the true parameter 95% of the time.
- Empirical convergence: As M increases, the observed proportion (ratiog) converges to this theoretical value.
- Statistical consistency: The simulation becomes more reliable and less sensitive to random noise.

```
Intervals containing the true value:  9490

Intervals missing the true value:  510

Ratio 94.89999999999999
```

4 (h)

Finally, for M = 10, 000, if we replaced value 0.025 in R code line 13 and 14 by 0.005 (or replace value 0.975 by 0.995 in line 26 in Python code file), what would you expect the value of 'Ratio' to be? Rigorously argue. And the plot to change?

Answer:

In the first code, the confidence interval is constructed as:

```
Number of simulations:
10000

Sample size:
50
```

$b2 \pm qt(0.025, df = 48, lower.tail = FALSE) \times se2$

That is:

```
t_val:  2.010634757624232
```

$5 \in [\hat{\beta}_k - 2.0106 \cdot se(\hat{\beta}_k), \hat{\beta}_k + 2.0106 \cdot se(\hat{\beta}_k)]$

This uses the 97.5th percentile of the t-distribution to create a 95% confidence interval (since 2.5% is in each tail).

If you replace with 0.995, you're now using the 99.5th percentile, which gives you a 99% confidence interval. That is:

```
t_val:  2.6822040269502136
```

$5 \in [\hat{\beta}_k - 2.6822 \cdot se(\hat{\beta}_k), \hat{\beta}_k + 2.6822 \cdot se(\hat{\beta}_k)]$

Let's define:

- "ratiog = 95" = proportion of intervals that contain the true value when using 95% confidence intervals
- "ratiog = 99"= same, but for 99% confidence intervals

Theoretical Expectation

- By definition, a 99% confidence interval is wider than a 95% interval.
- Therefore, it is more likely to contain the true parameter.

So with M = 10,000

- At 95% confidence, you expect ~9,500 intervals to contain the true value.
- At 99% confidence, you expect ~9,900 intervals to contain the true value.
- So should increase from ~95% to ~99%.

This is a direct consequence of the confidence level: higher confidence → wider intervals → higher coverage.

How Will the Plot Change?

The plot shows horizontal confidence intervals for each simulation, with:

- Gray lines: intervals that contain the true value ($\beta_2 = 5$)
- Red lines: intervals that miss the true value

With 99% Confidence:

- The intervals will be visibly wider.
- Fewer red lines: because more intervals will now include 5.
- The vertical dashed line at 5 will intersect more intervals.

So visually, the plot will look more "forgiving" - more intervals will span the true value, and the red lines will nearly disappear.

4 (i)

Finally, what do you think the purpose of this simulation exercise is? That is, what is it trying to illustrate? Be specific.

Answer:

This simulation exercise is designed to visually and empirically demonstrate the frequentist interpretation of confidence intervals — specifically, how often a confidence interval constructed at a given confidence level (e.g. 95%) actually contains the true parameter value when the experiment is repeated many times.

The simulation illustrates the following key statistical concepts:

1. Frequentist Coverage Probability

- A 95% confidence interval does not mean there's a 95% chance the true parameter lies within a single interval.
- Instead, it means that if we repeated the experiment many times, about 95% of the constructed intervals would contain the true parameter.
- The simulation makes this abstract idea concrete by repeating the experiment M times and calculating the proportion of intervals that actually include the true slope ($\beta_2 = 5$).

2. Sampling Variability

- Even with a fixed model and known true parameter, the estimated slope and its confidence interval vary from sample to sample due to random noise.
- This variability is visualized in the plot, where some intervals miss the true value (red lines), and most include it (gray lines).

3. Effect of Confidence Level

- By adjusting the quantile (e.g., from 0.025 to 0.005), you can see how increasing the confidence level (from 95% to 99%) leads to: Wider intervals, Higher coverage, Fewer misses (red lines)
- This helps students understand the trade-off between precision and confidence.

4. Law of Large Numbers in Simulation

- As M increases (e.g., from 100 to 10,000), the empirical coverage (ratiog) converges to the theoretical confidence level.
- This reinforces the idea that confidence intervals are long-run properties, not guarantees for individual samples.

By simulating and visualizing the process, the exercise:

- Clarifies the correct interpretation
- Builds trust in statistical inference methods
- Reveals the role of randomness in estimation

5 Question 5.

Ray Fair (https://fairmodel.econ.yale.edu/) work on US presidential elections, includes the following regression model for the Democratic share of the two party presidential vote:

$$VP_t = \beta_1 + \beta_2 I_t + \beta_3 DPER_t + \beta_4 DUR_t + \beta_5 WAR_t + \beta_6(G_t \cdot I_t) + \beta_7(P_t \cdot I_t) + \beta_8(Z_t \cdot I_t) + \epsilon_t$$

where:

$VP \equiv$ Democratic share of the two party presidential vote in election year t,

$I \equiv 1$ if party in White House in election year is Democrat and -1 if its Republican;

$DPER \equiv 1$ if Democratic president in office is running for reelection; -1 is a Republican president is running again, 0 otherwise;

$DUR \equiv 0$ if party in White House has been in office for only one term, $1[-1]$ if the Democratic [Republican] party has been in the White house for 2 consecutive terms, $1.25[-1.25]$ if the Democratic [Republican] party has been in the White house for 3 consecutive terms, $1.5[-1.5]$ if the Democratic [Republican] party has been in the White house for 4 consecutive terms and so on;

$WAR \equiv 1$ for election years 1918, 1920, 1942, 1944, 1946, 1948, 0 otherwise;

$G \equiv$ growth rate of real GDP in the first 3 quarters of the on-term election year (annual rate);

$P \equiv$ absolute value of growth rate of GDP deflator in the first 15 quarters of the administration (annual rate) except for 1920, 1944, and 1948, where values set to 0;

$Z \equiv$ number of quarters in the first 15 quarters of the administration in which the growth rate of real per capita GDP exceeds 3.2 % at an annual rate except for 1920, 1944, and 1948, where the values are zero.

Fair argues that there are 4 conditions that affect voting patterns in US presidential elections:

(i) Incumbent presidents running again for reelection: Voters tend to favor presidents running again.

(ii) How long a party has controlled the White House. Voters like change. When a party has been in power for two or more consecutive terms, this has a negative effect on votes for that party's candidate.

(iii) There is a slight, but persistent, bias in favor of the Republican Party.

(iv) The state of the economy. A good economy at the time of the election has a positive effect on votes for the incumbent party candidate.

(a) Each of the four Fair's conditions translate in terms of the sign of the parameters of the regression above. Taking the conditions one by one, identify the relevant parameter that captures that condition and briefly argue what the expected sign for the parameter is according to Fair.

(b) With the help of Stata, estimate the model above using data on US elections from 1916 to 2020 included in file USelections.csv (Fair's data). Include the Stata output in your answer. Are the signs of the OLS estimates of the parameters as expected given Fair's 4 conditions? Briefly argue.

(c) Stata output includes, among others, the default calculations of the following statistics:

i. $se(\hat{\beta}_6)$
ii. t−value and p−value associated with the significance test of regressor I.

Provide the specific expression defining each of these three statistics, and then using R/Python calculate each these statistics by using the expression that defines them, and verify you get the same values as the ones reproduced in the Stata output.

(d) Test $H_0 : \beta_7 = 0$ versus $H_0 : \beta_7 \neq 0$ using exact t − test statistic and 5% significance level, basing your decision rule on the comparison of t − value with the criticalvalue. Draw the associated acceptance and rejection regions properly labelling both axes. What did you conclude?

(e) Draw the p-value associated to the test performed in question 5d. Properly label the axes. What is the minimum significance level that would make this regressor be significant?

(f) We want to test whether the famous phrase "it's the economy, stupid!" " has empirical evidence. With the help of Stata, test whether the economic variables $G \cdot I, P \cdot I$ and $Z \cdot I$ are jointly significant. Show all the steps. Include the Stata output as the answer.

(g)

(h) Given that the latest estimate of the democratic share of the two-party vote for November 2024 elections is 49.25, what is the prediction error we would get if we used R. Fair values as for $G = 1.7, P = 4.54$ and $Z = 4$ before the election? Justify.

(ii) What can you say about the method used by Fair, and reproduced in this exercise, to predict VP? Briefly justify.

Exercise 5

5(a)

i.

β_3, and its sign should be positive. Therefore, if $DPER_t = 1$, meaning that the current Democratic president runs for reelection, the average association between VP_t and $DPER_t$ controlling for every other variable in the model is positive, correctly reflecting that voters

tend to favor presidents running again. Clearly, if it is the Republican president the one that is running for reelection ($DPER_t = -1$), said relationship is negative.

ii.

β_4, and in this case, its sign should also be positive. According to Fair, when a party has been in power for two or more consecutive terms, a negative effect on votes for that party's candidate appears. Thus, and given that DUR_t takes negative values if the Republican Party has been in the White House for at least 2 consecutive terms and that we are estimating the Democratic share of the vote, the sign of the parameter has to be negative so as to obtain a positive VP_t estimation when holding every other variable in the model constant.

iii.

β_2, and we should expect its sign to be negative. Suppose that we control for every variable in the model except for I_t. When $I_t = 1$, that is, when a Democrat is in power, its relationship with VP_t is negative, which can be interpreted as a systematic bias in favor of the Republicans. Expressed in other words, the Democratic vote share is lower when it is the Democratic Party itself in the White House. However, if I_t is not in the model or presents issues (multicollinearity or statistical insignificance, for instance), the intercept term β_1 can represent said bias by taking a negative value.

iv.

β_6 and β_8, and its sign should be positive. This case is particularly interesting, as we are looking into the parameters associated with two interactions of variables, $G_t \cdot I_t$ and $Z_t \cdot I_t$. In the favorable economic scenario portrayed by Fair, both G_t and Z_t take positive values, and $I_t = 1$ means that the Democrats are in office. Thus, if both estimators are positive and we hold every other in the model constant, the relationship between a healthy economy and the Democratic share of the vote given that the president is a Democrat is positive, which is in line with Fair's condition.

5(b)

Source	SS	df	MS	Number of obs	=	27
				F(8, 18)	=	14.47
Model	989.23852	8	123.654815	Prob > F	=	0.0000
Residual	153.785735	18	8.54365196	R-squared	=	0.8655
				Adj R-squared	=	0.8057
Total	1143.02425	26	43.9624713	Root MSE	=	2.923

vp	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
vc	.2954658	.2013048	1.47	0.159	-.1274599	.7183915
i	.5726985	2.40617	0.24	0.815	-4.482476	5.627873
dper	1.030307	1.697542	0.61	0.551	-2.536097	4.596711
dur	-3.434541	1.34685	-2.55	0.020	-6.264168	-.6049144
war	3.986597	2.693987	1.48	0.156	-1.673259	9.646453
gi	.6380737	.1336845	4.77	0.000	.3572131	.9189343
pi	-.4075756	.3414378	-1.19	0.248	-1.12491	.3097586
zi	.6205397	.3050234	2.03	0.057	-.0202907	1.26137
_cons	33.25177	10.22114	3.25	0.004	11.77795	54.72559

The OLS estimates $\hat{\beta}_3$, $\hat{\beta}_4$, $\hat{\beta}_6$ and $\hat{\beta}_8$ present the expected signs. $\hat{\beta}_2$, however, is positive. This could be explained by the sample size being too small to back up a statement as bold as the one associated with our intuition, considering that there are only 27 observations in this dataset. There might also be multicollinearity problems between different variables, such as I_t and $DPER_t$, since both take the value 1 or -1, depending on the party, if the current president is running for reelection. It is also important to note that neither $\hat{\beta}_1$ nor its true value β_1 are negative, which could have been the case given the potential multicollinearity problems or the low statistical significance of I_t.

5(c) (missing R output)

$$se(\hat{\beta}_6) \equiv \sqrt{\widehat{var}(\hat{\beta}_6/X)} = \sqrt{\hat{\sigma}^2 (X'X)^{-1}_{66}}$$

$$\mathbf{t} \equiv \frac{\hat{\beta}_2}{se(\hat{\beta}_2)} \overset{\text{under } H_0}{\sim} t(n-K) \Longrightarrow t\text{-value} \equiv \mathbf{t}(data)$$

$$p\text{-value} \equiv \mathbb{P}\left(|\mathbf{t}| \geqslant |t\text{-value}| \,\middle|\, \text{under } H_0\right)$$

5(d)

The observed t-value ≈ -0.4076 does not fall inside the rejection region, since $|t\text{-value}| < |\text{critical value}|$. Hence, we do not reject H_0, implying that the true value of the parameter β_7 is equal to 0.

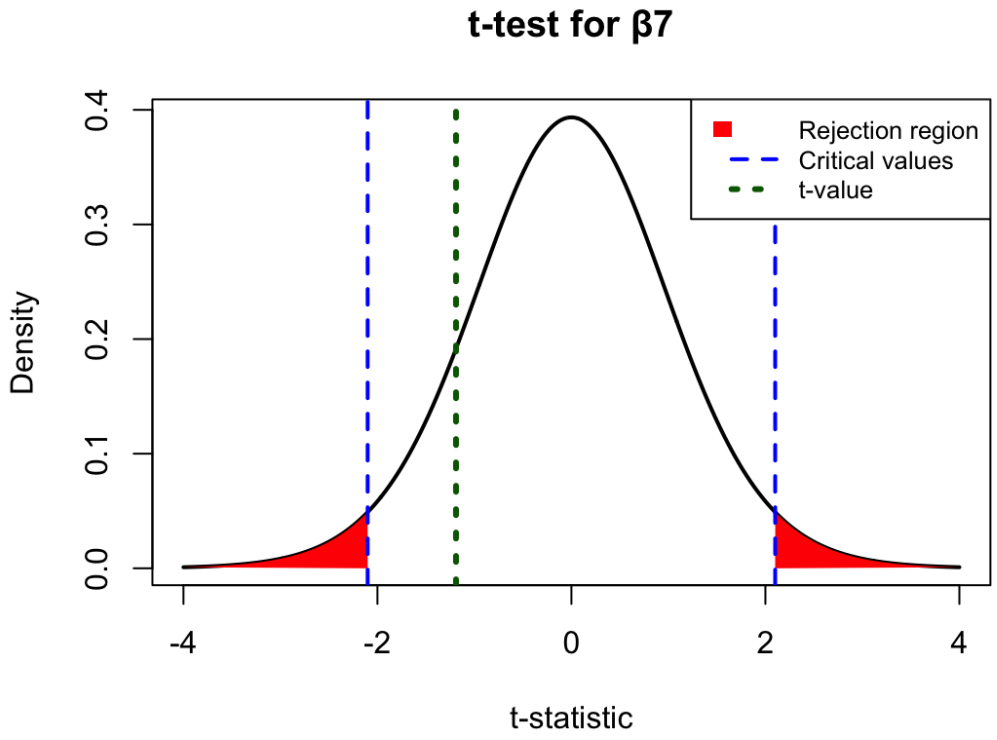


Figure 1: The rejection region could have also been drawn as an interval of values of t-statistic.

5(e)

The p-value must be lower than the significance level for us to reject H_0 and thus conclude that the true value of β_7 is not equal to 0, deeming it statistically significant. Hence, in this case, the regressor $P_t I_t$ would be significant if α was equal to or above $0.248 = 24.8\%$.

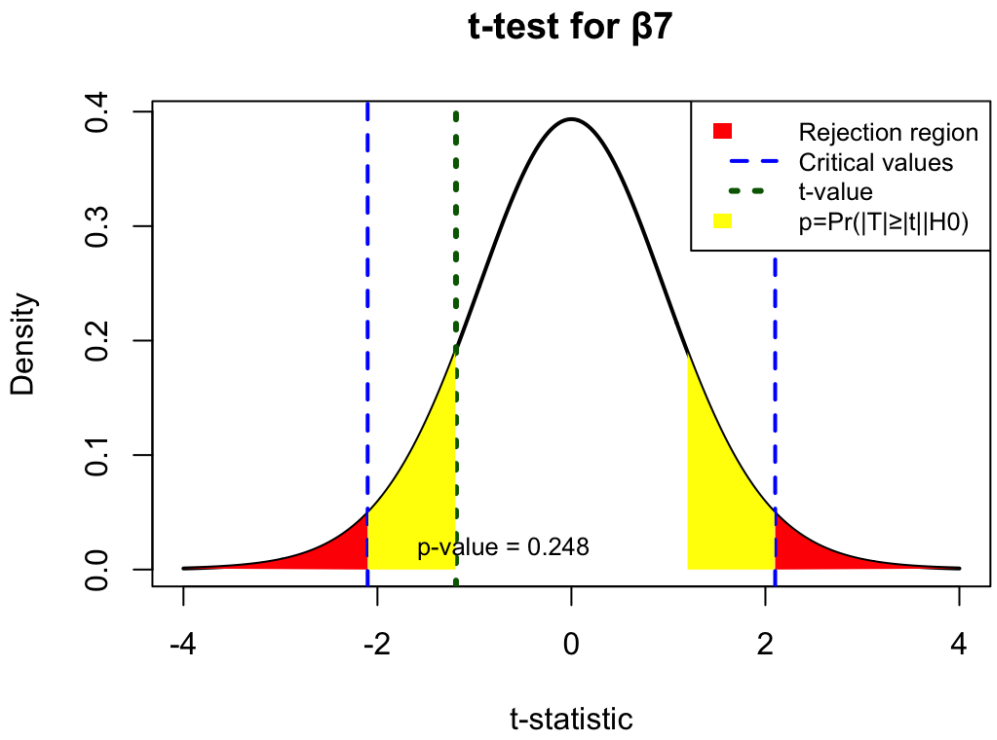


Figure 2: The rejection region could have also been drawn as an interval of values of t-statistic.

5(f)

First, the null and alternative hypotheses must be stated:

$$H_0 : \beta_6 = \beta_7 = \beta_8 = 0 \quad \text{vs} \quad H_1 : \text{not } H_0$$

The full model needs to be estimated so as to obtain $SSE_{unrestricted}$, but since such estimation was already computed in a previous section, we can directly conclude that $SSE_{unrestricted} = 153.785735$

Let us now build the restricted model, that is, the full model without including the interactions G_tI_t, P_tI_t and Z_tI_t:

$$VP_t = \beta_1 + \beta_2 I_t + \beta_3 DPER_t + \beta_4 DUR_t + \beta_5 WAR_t + \epsilon_t$$

Once the model has been estimated, we obtain that $SSE_{restricted} = 432.789743$.

At this point, we are set to compute the F-value:

$$F = \frac{(SSE_{restricted} - SSE_{unrestricted})/q}{SSE_{unrestricted}/(n-K)} = \frac{432.789743 - 153.785735)/3}{153.785735/(26-8)} =$$