STATS 531 Homework 1 Solutions

Kunyang He

2026-01-18

Table of contents

1 Question 1.1: Variance of Sample Mean for Covariance Stationary Time Series

1.1 Problem Statement

Let $Y_{1:N}$ be a covariance stationary time series with autocovariance function γ_h and constant mean $\mu_n = \mu$. We want to derive the variance of the sample mean estimator:

$$\hat{\mu}(y_{1:N}) = \frac{1}{N}\sum_{n=1}^{N} y_n$$

1.2 Solution

We need to show that:

$$\text{Var}(\hat{\mu}(Y_{1:N})) = \frac{1}{N}\gamma_0 + \frac{2}{N^2}\sum_{h=1}^{N-1}(N-h)\gamma_h$$

1.2.1 Step 1: Express the Variance Using Covariance Properties

Starting with the definition of the sample mean:

$$\hat{\mu}(Y_{1:N}) = \frac{1}{N}\sum_{n=1}^{N} Y_n$$

The variance can be written as:

$$\text{Var}(\hat{\mu}(Y_{1:N})) = \text{Var}\left(\frac{1}{N}\sum_{n=1}^{N} Y_n\right)$$

1.2.2 Step 2: Apply Property P1 (Variance as Covariance)

Using P1: $\text{Cov}(Y, Y) = \text{Var}(Y)$, we have:

$$\text{Var}\left(\frac{1}{N}\sum_{n=1}^{N} Y_n\right) = \text{Cov}\left(\frac{1}{N}\sum_{n=1}^{N} Y_n, \frac{1}{N}\sum_{n=1}^{N} Y_n\right)$$

1.2.3 Step 3: Apply Property P3 (Scalar Multiplication)

Using P3: $\text{Cov}(aX, bY) = ab\,\text{Cov}(X, Y)$:

$$= \frac{1}{N^2}\text{Cov}\left(\sum_{n=1}^{N} Y_n, \sum_{n=1}^{N} Y_n\right)$$

1.2.4 Step 4: Apply Property P4 (Sum of Covariances)

Using P4: $\text{Cov}\left(\sum_{m=1}^{M} Y_m, \sum_{n=1}^{N} Y_n\right) = \sum_{m=1}^{M}\sum_{n=1}^{N}\text{Cov}(Y_m, Y_n)$:

$$= \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\text{Cov}(Y_m, Y_n)$$

1.2.5 Step 5: Use Covariance Stationarity

For a covariance stationary process, $\text{Cov}(Y_m, Y_n) = \gamma_{|m-n|}$. Therefore:

$$\text{Var}(\hat{\mu}(Y_{1:N})) = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\gamma_{|m-n|}$$

1.2.6 Step 6: Reorganize the Double Sum

Let's count how many pairs (m, n) give each lag $h = |m - n|$:

- For $h = 0$: There are N pairs where $m = n$
- For $h = 1, 2, \ldots, N-1$: There are $2(N - h)$ pairs (both $m - n = h$ and $n - m = h$)

Thus:

$$\sum_{m=1}^{N}\sum_{n=1}^{N}\gamma_{|m-n|} = N\gamma_0 + \sum_{h=1}^{N-1} 2(N-h)\gamma_h$$

1.2.7 Step 7: Final Result

Substituting back:

$$\mathrm{Var}(\hat{\mu}(Y_{1:N})) = \frac{1}{N^2}\left[N\gamma_0 + 2\sum_{h=1}^{N-1}(N-h)\gamma_h\right]$$

$$\boxed{\mathrm{Var}(\hat{\mu}(Y_{1:N})) = \frac{1}{N}\gamma_0 + \frac{2}{N^2}\sum_{h=1}^{N-1}(N-h)\gamma_h}$$

This completes the derivation. □

2 Question 1.2: Sample Autocorrelation Function Analysis

2.1 Part A: Mean and Standard Deviation of the Simplified ACF Estimator

2.1.1 Problem Setup

We analyze the simplified autocorrelation estimator (assuming known zero mean):

$$\widehat{\rho}_h(Y_{1:N}) = \frac{\frac{1}{N}\sum_{n=1}^{N-h} Y_n Y_{n+h}}{\frac{1}{N}\sum_{n=1}^{N} Y_n^2}$$

where $Y_1, \ldots, Y_N$ are IID with zero mean and finite variance σ^2.

2.1.2 Define the Components

Let:

$$U = \widehat{\gamma}_h(Y_{1:N}) = \frac{1}{N}\sum_{n=1}^{N-h} Y_n Y_{n+h}$$

$$V = \widehat{\gamma}_0(Y_{1:N}) = \frac{1}{N}\sum_{n=1}^{N} Y_n^2$$

Then $\widehat{\rho}_h = g(U, V) = \frac{U}{V}$.

2.1.3 Step 1: Find $\mathbb{E}[U]$ and $\mathbb{E}[V]$

For V:

$$\mathbb{E}[V] = \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N} Y_n^2\right] = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}[Y_n^2] = \frac{1}{N} \cdot N \cdot \sigma^2 = \sigma^2$$

For U (when $h > 0$): Since Y_n and Y_{n+h} are independent for $h > 0$:

$$\mathbb{E}[U] = \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N-h} Y_n Y_{n+h}\right] = \frac{1}{N}\sum_{n=1}^{N-h} \mathbb{E}[Y_n]\mathbb{E}[Y_{n+h}] = 0$$

2.1.4 Step 2: Find $\mathrm{Var}(U)$ and $\mathrm{Var}(V)$

For $\mathrm{Var}(V)$:

Let $\mu_4 = \mathbb{E}[Y_1^4]$ (the fourth moment). Since the Y_i^2 are IID:

$$\mathrm{Var}(V) = \mathrm{Var}\left(\frac{1}{N}\sum_{n=1}^{N} Y_n^2\right) = \frac{1}{N^2} \cdot N \cdot \mathrm{Var}(Y_1^2) = \frac{1}{N}(\mu_4 - \sigma^4)$$

So $\mathrm{Var}(V) = O(1/N)$.

For $\mathrm{Var}(U)$ (when $h > 0$):

$$\mathrm{Var}(U) = \mathrm{Var}\left(\frac{1}{N}\sum_{n=1}^{N-h} Y_n Y_{n+h}\right)$$

Since $Y_n Y_{n+h}$ are independent across n (for $h > 0$, the pairs don't overlap):

$$\mathrm{Var}(U) = \frac{1}{N^2}\sum_{n=1}^{N-h} \mathrm{Var}(Y_n Y_{n+h})$$

For independent Y_n and Y_{n+h}:

$$\mathrm{Var}(Y_n Y_{n+h}) = \mathbb{E}[Y_n^2 Y_{n+h}^2] - (\mathbb{E}[Y_n Y_{n+h}])^2 = \mathbb{E}[Y_n^2]\mathbb{E}[Y_{n+h}^2] - 0 = \sigma^4$$

Therefore:

$$\mathrm{Var}(U) = \frac{1}{N^2} \cdot (N-h) \cdot \sigma^4 \approx \frac{\sigma^4}{N} \text{ for large } N$$

2.1.5 Step 3: Apply the Delta Method

Using the two-dimensional delta method with $g(u, v) = u/v$:

The partial derivatives are:

$$\frac{\partial g}{\partial u} = \frac{1}{v}, \quad \frac{\partial g}{\partial v} = -\frac{u}{v^2}$$

At $(\mu_U, \mu_V) = (0, \sigma^2)$:

$$\left.\frac{\partial g}{\partial u}\right|_{(0,\sigma^2)} = \frac{1}{\sigma^2}, \quad \left.\frac{\partial g}{\partial v}\right|_{(0,\sigma^2)} = 0$$

The Taylor expansion gives:

$$\hat{\rho}_h \approx g(0, \sigma^2) + (U - 0) \cdot \frac{1}{\sigma^2} + (V - \sigma^2) \cdot 0 = \frac{U}{\sigma^2}$$

2.1.6 Step 4: Compute Mean and Variance of $\hat{\rho}_h$

Mean:

$$\mathbb{E}[\hat{\rho}_h] \approx \mathbb{E}\left[\frac{U}{\sigma^2}\right] = \frac{\mathbb{E}[U]}{\sigma^2} = \frac{0}{\sigma^2} = 0$$

Variance:

$$\mathrm{Var}(\hat{\rho}_h) \approx \mathrm{Var}\left(\frac{U}{\sigma^2}\right) = \frac{\mathrm{Var}(U)}{\sigma^4} = \frac{\sigma^4/N}{\sigma^4} = \frac{1}{N}$$

2.1.7 Conclusion for Part A

For large N:

- **Mean:** $\mathbb{E}[\hat{\rho}_h] \approx 0$
- **Standard Deviation:** $\mathrm{SD}(\hat{\rho}_h) \approx \boxed{\frac{1}{\sqrt{N}}}$

This confirms the approximation used in `statsmodels` for the standard error of the sample autocorrelation under the null hypothesis of white noise.

2.2 Part B: Is the Shaded Region a Confidence Interval?

2.2.1 Definition of a Confidence Interval

A **confidence interval** is a random interval $[L(Y_{1:N}), U(Y_{1:N})]$ such that for a given confidence level $(1-\alpha)$:

$$\mathbb{P}(\theta \in [L(Y_{1:N}), U(Y_{1:N})]) \geq 1-\alpha$$

where θ is a fixed (unknown) parameter.

2.2.2 Analysis of the ACF Plot Shaded Region

The shaded region in `plot_acf` is computed as:

$$\hat{\rho}_h \pm z_{\alpha/2} \cdot \frac{1}{\sqrt{N}}$$

where $z_{\alpha/2} \approx 1.96$ for a 95% level (when `alpha=0.05`).

2.2.3 Critical Observation

The shaded region does NOT construct a proper confidence interval. Here's why:

1. **The region is testing the null hypothesis:** The interval is centered at 0 (the null hypothesis value $\rho_h = 0$), not at the observed estimate $\hat{\rho}_h$.
2. **Interpretation as a significance test:** The horizontal dashed lines at $\pm 1.96/\sqrt{N}$ represent rejection regions for testing:
   - $H_0 : \rho_h = 0$ (the true autocorrelation is zero)
   - $H_1 : \rho_h \neq 0$

   If $|\hat{\rho}_h| > 1.96/\sqrt{N}$, we reject H_0 at the 5% significance level.
3. **A proper confidence interval would be:**

   $$\left[\hat{\rho}_h - z_{\alpha/2} \cdot \frac{1}{\sqrt{N}}, \quad \hat{\rho}_h + z_{\alpha/2} \cdot \frac{1}{\sqrt{N}}\right]$$

   This interval is centered at the **estimated value** $\hat{\rho}_h$, not at zero.

2.2.4 Correct Interpretation

The shaded region should be interpreted as follows:

> If the true autocorrelations are all zero (i.e., the data is white noise), then approximately 95% of the sample autocorrelation estimates should fall within the shaded region.

This is a **pointwise significance test**, not a confidence interval. The region helps assess whether observed autocorrelations are significantly different from zero.

2.2.5 Additional Issues

1. **Multiple comparisons problem:** When examining many lags simultaneously, even for true white noise, we expect about 5% of lags to exceed the bounds by chance.
2. **Assumption validity:** The $1/\sqrt{N}$ formula assumes:
   - The null hypothesis that the series is IID (white noise)
   - Large sample sizes for the normal approximation
   - The Bartlett formula (`bartlett_confint=True`) provides a more sophisticated estimate when the null hypothesis is not white noise

2.2.6 Conclusion for Part B

The documentation's use of "confidence interval" is **technically incorrect**. The shaded region represents a **critical region for hypothesis testing** at a specified significance level. The probability statement is:

$$\mathbb{P}\left(|\hat{\rho}_h| < \frac{1.96}{\sqrt{N}} \,\middle|\, H_0 : \rho_h = 0\right) \approx 0.95$$

This is a statement about Type I error control, not about interval coverage of an unknown parameter.

3 Numerical Verification with Python

3.1 Simulation Study for Question 1.2A

We verify our theoretical results through Monte Carlo simulation:

```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf
import statsmodels.api as sm

# Set random seed for reproducibility
np.random.seed(531)

# Parameters
N = 100  # Sample size
n_simulations = 10000  # Number of Monte Carlo replications
h = 1  # Lag to examine

# Storage for sample autocorrelations
rho_hat_samples = []

for _ in range(n_simulations):
    # Generate IID N(0,1) data
    Y = np.random.normal(0, 1, N)

    # Compute simplified autocorrelation estimator (known mean = 0)
    numerator = np.mean(Y[:N-h] * Y[h:])
    denominator = np.mean(Y**2)
    rho_hat = numerator / denominator

    rho_hat_samples.append(rho_hat)

rho_hat_samples = np.array(rho_hat_samples)

# Compute empirical statistics
empirical_mean = np.mean(rho_hat_samples)
empirical_std = np.std(rho_hat_samples)
theoretical_std = 1 / np.sqrt(N)

print("="*50)
print("Monte Carlo Simulation Results (lag h = 1)")
```

```
print("="*50)
print(f"Sample size N = {N}")
print(f"Number of simulations = {n_simulations}")
print("-"*50)
print(f"Empirical Mean:      {empirical_mean:.6f}")
print(f"Theoretical Mean:    0.000000")
print("-"*50)
print(f"Empirical Std Dev:   {empirical_std:.6f}")
print(f"Theoretical Std Dev: {theoretical_std:.6f}")
print(f"Ratio (Empirical/Theoretical): {empirical_std/theoretical_std:.4f}")
print("="*50)
```

```
==================================================
Monte Carlo Simulation Results (lag h = 1)
==================================================
Sample size N = 100
Number of simulations = 10000
--------------------------------------------------
Empirical Mean:      0.000793
Theoretical Mean:    0.000000
--------------------------------------------------
Empirical Std Dev:   0.098778
Theoretical Std Dev: 0.100000
Ratio (Empirical/Theoretical): 0.9878
==================================================
```

```
# Plot histogram of sample autocorrelations
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Histogram
axes[0].hist(rho_hat_samples, bins=50, density=True, alpha=0.7,
             edgecolor='black', label='Empirical')
x = np.linspace(-0.4, 0.4, 100)
axes[0].plot(x, 1/(theoretical_std*np.sqrt(2*np.pi)) *
             np.exp(-x**2/(2*theoretical_std**2)),
             'r-', lw=2, label=f'N(0, 1/√N)')
axes[0].axvline(x=0, color='k', linestyle='--', alpha=0.5)
axes[0].set_xlabel(r'$\hat{\rho}_1$')
axes[0].set_ylabel('Density')
axes[0].set_title(f'Distribution of Sample Autocorrelation (N={N})')
axes[0].legend()
```

```
# Q-Q plot
from scipy import stats
stats.probplot(rho_hat_samples, dist="norm", plot=axes[1])
axes[1].set_title('Q-Q Plot (Normal)')

plt.tight_layout()
plt.savefig('acf_distribution.png', dpi=150, bbox_inches='tight')
plt.show()
```

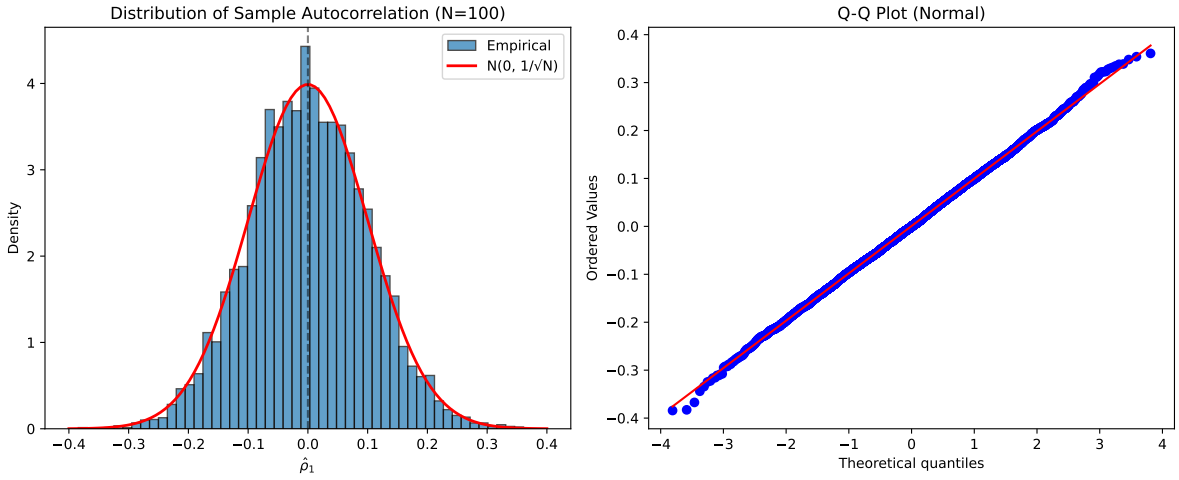

Figure 1: Distribution of sample autocorrelation estimates under H0

3.2 Example: Sample ACF Plot with Confidence Bounds

```
# Generate a single realization of white noise
np.random.seed(42)
N = 200
white_noise = np.random.normal(0, 1, N)

# Create ACF plot
fig, axes = plt.subplots(1, 2, figsize=(14, 5))

# Plot 1: The data
axes[0].plot(white_noise, 'b-', alpha=0.7, linewidth=0.8)
axes[0].axhline(y=0, color='k', linestyle='-', alpha=0.3)
axes[0].set_xlabel('Time')
```

```
axes[0].set_ylabel('Value')
axes[0].set_title(f'White Noise Realization (N={N})')

# Plot 2: ACF
plot_acf(white_noise, ax=axes[1], lags=30, alpha=0.05)
axes[1].set_xlabel('Lag')
axes[1].set_ylabel('Autocorrelation')
axes[1].set_title('Sample ACF with 95% Bounds')

# Add annotations
bound = 1.96 / np.sqrt(N)
axes[1].annotate(f'Upper bound: +{bound:.3f}', xy=(25, bound),
                 fontsize=10, color='blue')
axes[1].annotate(f'Lower bound: -{bound:.3f}', xy=(25, -bound),
                 fontsize=10, color='blue')

plt.tight_layout()
plt.savefig('acf_example.png', dpi=150, bbox_inches='tight')
plt.show()

print(f"\nTheoretical 95% bounds: ±{bound:.4f}")
print(f"Formula: ±1.96/√N = ±1.96/√{N} = ±{1.96/np.sqrt(N):.4f}")
```

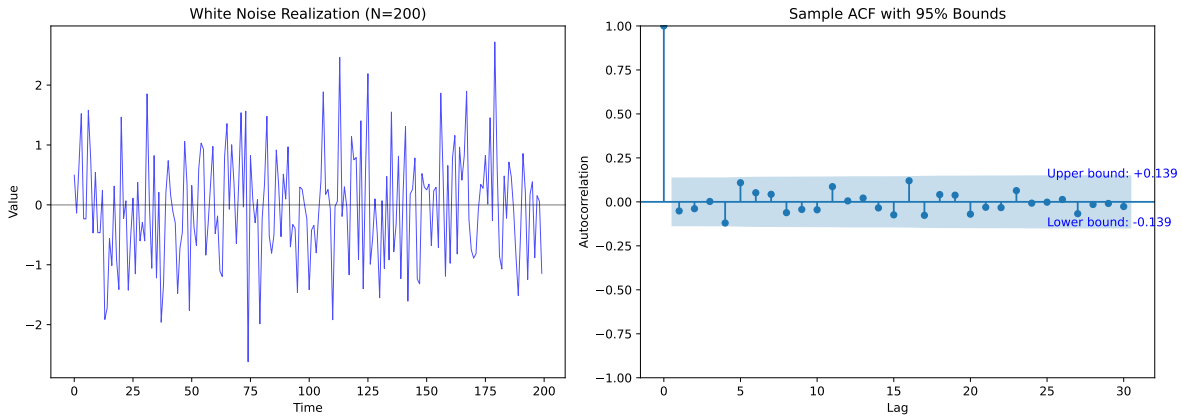

Figure 2: Sample ACF for white noise data showing the shaded confidence region

```
Theoretical 95% bounds: ±0.1386
Formula: ±1.96/√N = ±1.96/√200 = ±0.1386
```

3.3 Demonstration: Multiple Comparisons Issue

```
# Demonstrate multiple comparisons problem
np.random.seed(123)
N = 100
max_lag = 40

# Count how many lags exceed bounds
n_trials = 1000
exceedance_counts = []

for _ in range(n_trials):
    Y = np.random.normal(0, 1, N)
    acf_vals = sm.tsa.acf(Y, nlags=max_lag)
    bound = 1.96 / np.sqrt(N)
    # Count exceedances (excluding lag 0)
    exceedances = np.sum(np.abs(acf_vals[1:]) > bound)
    exceedance_counts.append(exceedances)

exceedance_counts = np.array(exceedance_counts)

print("Multiple Comparisons Analysis")
print("="*50)
print(f"Sample size: N = {N}")
print(f"Number of lags tested: {max_lag}")
print(f"Number of simulations: {n_trials}")
print("-"*50)
print(f"Expected exceedances per realization: {0.05 * max_lag:.1f}")
print(f"Observed mean exceedances: {np.mean(exceedance_counts):.2f}")
print(f"Proportion with at least one exceedance: "
      f"{np.mean(exceedance_counts > 0):.3f}")
print("="*50)

# Plot histogram of exceedances
plt.figure(figsize=(8, 5))
plt.hist(exceedance_counts, bins=range(0, 12), density=True,
         alpha=0.7, edgecolor='black', align='left')
plt.axvline(x=np.mean(exceedance_counts), color='r', linestyle='--',
            label=f'Mean = {np.mean(exceedance_counts):.2f}')
plt.axvline(x=0.05*max_lag, color='g', linestyle=':',
            label=f'Expected = {0.05*max_lag:.1f}')
plt.xlabel('Number of lags exceeding bounds')
```

```
plt.ylabel('Proportion')
plt.title('Exceedances Under True White Noise')
plt.legend()
plt.savefig('multiple_comparisons.png', dpi=150, bbox_inches='tight')
plt.show()
```

```
Multiple Comparisons Analysis
==================================================
Sample size: N = 100
Number of lags tested: 40
Number of simulations: 1000
--------------------------------------------------
Expected exceedances per realization: 2.0
Observed mean exceedances: 1.09
Proportion with at least one exceedance: 0.631
==================================================
```

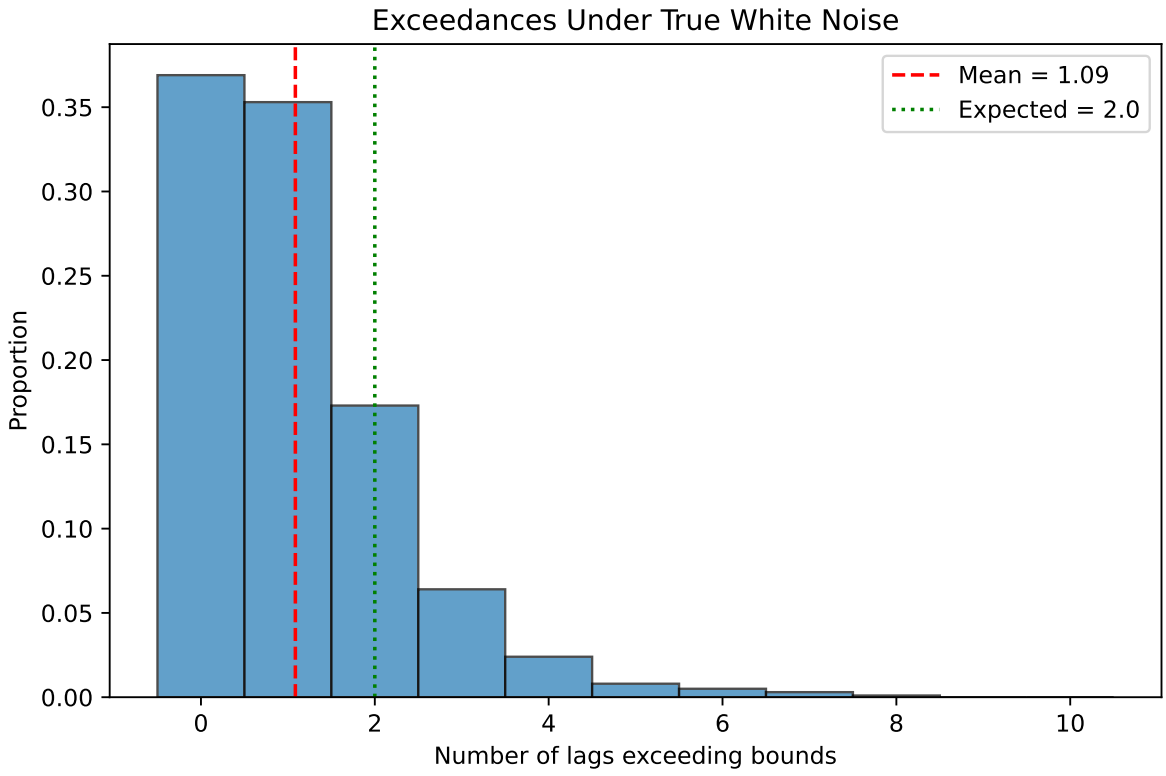

Figure 3: Under true white noise, some lags exceed bounds by chance

4 References

1. Shumway, R.H. and Stoffer, D.S. (2017). *Time Series Analysis and Its Applications: With R Examples* (4th ed.). Springer.
2. Rice, J.A. (2007). *Mathematical Statistics and Data Analysis* (3rd ed.). Cengage Learning.
3. Wikipedia contributors. (2024). Delta method. In *Wikipedia, The Free Encyclopedia*. Retrieved from https://en.wikipedia.org/wiki/Delta_method
4. statsmodels Documentation. *statsmodels.graphics.tsaplots.plot_acf*. Retrieved from https://www.statsmodels.org/stable/generated/statsmodels.graphics.tsaplots.plot_acf.html
5. statsmodels Source Code. GitHub Repository. https://github.com/statsmodels/statsmodels/blob/main/st