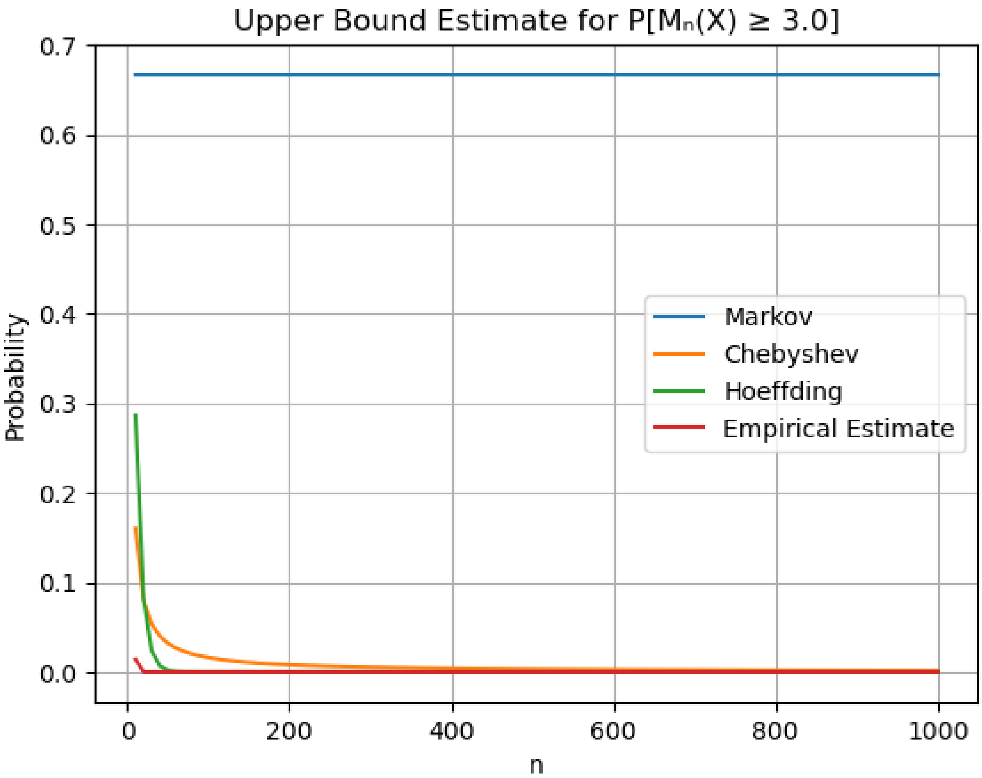
Upper Bound Estimate for $P[M_n(X) \geq 2.5]$

$c = 2.5$, $\theta = 0.6$

[156]:
```python
# c = 2.5
plt.plot(n_values, markov, label='Markov')
plt.plot(n_values, chebyshev, label='Chebyshev')
plt.plot(n_values, hoeffding, label='Hoeffding')
plt.plot(n_values, emp_estimates, label='Empirical Estimate')

plt.title(f'Upper Bound Estimate for P[Mₙ(X) ≥ {c}]')
plt.xlabel('n')
plt.ylabel('Probability')
plt.legend()
plt.grid(True)
```

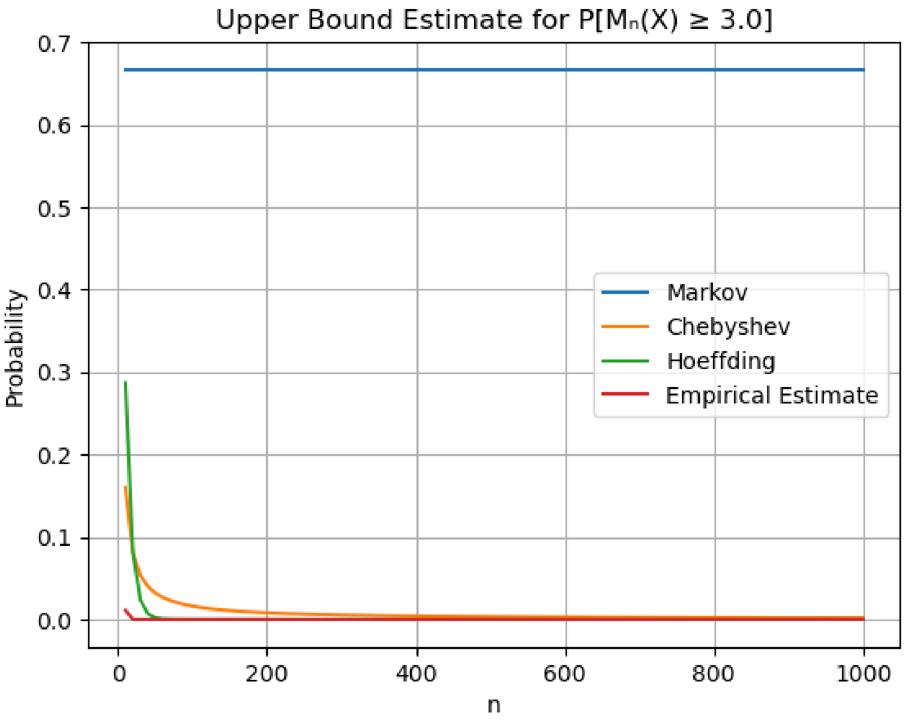


Upper Bound Estimate for $P[M_n(X) \geq 3.0]$

$c = 3.0$, $\theta = 0.6$

[103]:
```
plt.plot(n_values, markov, label='Markov')
plt.plot(n_values, chebyshev, label='Chebyshev')
plt.plot(n_values, hoeffding, label='Hoeffding')
plt.plot(n_values, emp_estimates, label='Empirical Estimate')

plt.title(f'Upper Bound Estimate for P[Mₙ(X) ≥ {c}]')
plt.xlabel('n')
plt.ylabel('Probability')
plt.legend()
plt.grid(True)
```

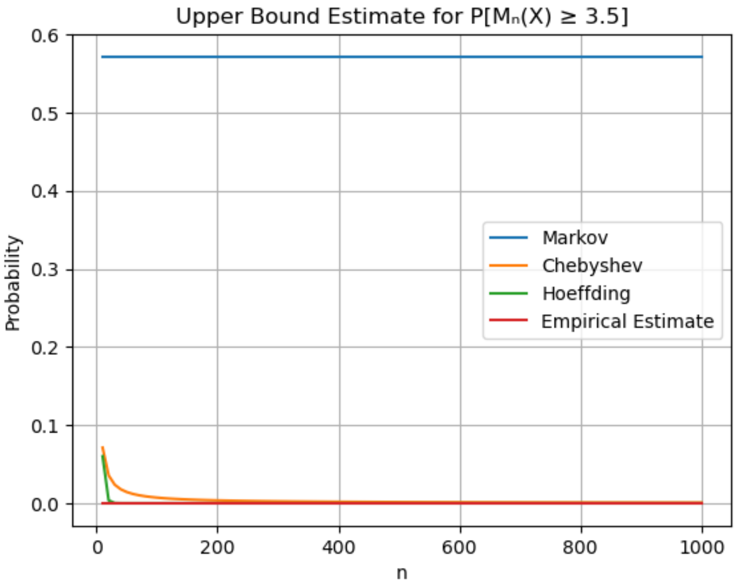


Upper Bound Estimate for P[M_n(X) ≥ 3.5]

c = 2.5, θ = 0.6

```
[107]: plt.plot(n_values, markov, label='Markov')
plt.plot(n_values, chebyshev, label='Chebyshev')
plt.plot(n_values, hoeffding, label='Hoeffding')
plt.plot(n_values, emp_estimates, label='Empirical Estimate')

plt.title(f'Upper Bound Estimate for P[Mₙ(X) ≥ {c}]')
plt.xlabel('n')
plt.ylabel('Probability')
plt.legend()
plt.grid(True)
```



Markov's bound: as c increases, the bound decreases from 0.8 (c=2.5) to below 0.6 (c=3.5).
Chebyshev's bound: As c increases, the Chebyshev bound becomes tighter and converges faster.
Hoeffding's bound: Similar to Chebyshev's bound, Hoeffding's bound also converges rapidly as c increases.
Empirical Estimate: Likewise, a larger c causes a very tight bound with Probability reaching almost 0 at c= 3.5 for the above instance.

- Markov's bound: as c increases, the bound decreases from 0.8 (c=2.5) to below 0.6 (c=3.5).
- Chebyshev's bound: As c increases, the Chebyshev bound becomes tighter and converges faster.
- Hoeffding's bound: Similar to Chebyshev's bound, Hoeffding's bound also converges rapidly as c increases.
- Empirical Estimate: Likewise, a larger c causes a very tight bound with Probability reaching almost 0 at c= 3.5 for the above instance.

[2]:
```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[15]:
```python
mu = 5.0
sigma = 1
c = mu + sigma
```

[33]:
```python
n_values = np.array([i for i in range(10,1001,10)])

chebyshev = (sigma**2) / ((c - mu)**2)
emp_estimates = np.zeros(len(n_values))
```

[34]:
```python
def calculate_probability(sample, c):
    return np.mean(sample >= c)
```

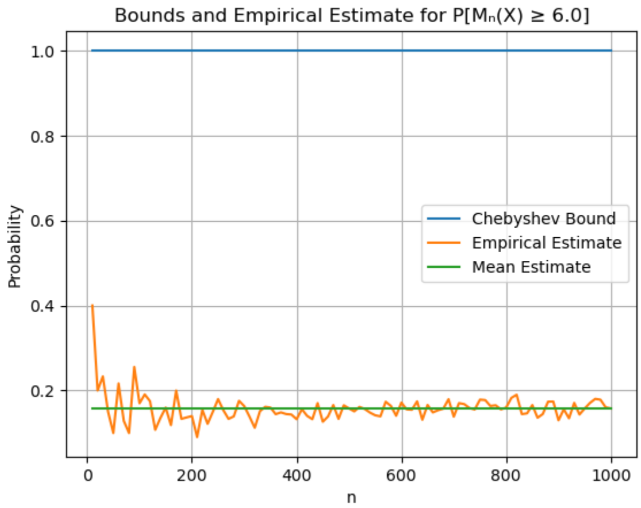
[36]:
```python
for i, n in enumerate(n_values):
    sample = np.random.normal(mu, sigma, size=n)
    emp_estimates[i] = calculate_probability(sample, c)
```

Upper Bound Estimate for $P[M_n(X) \geq 6.0]$

c = 6.0,

[66]:
```python
plt.plot(n_values, chebyshev * np.ones_like(n_values), label='Chebyshev Bound')
plt.plot(n_values, emp_estimates, label='Empirical Estimate')
plt.plot(n_values, np.mean(emp_estimates)*np.ones_like(n_values), label='Mean Estimate')

plt.xlabel('n')
plt.ylabel('Probability')
plt.title(f'Bounds and Empirical Estimate for P[Mₙ(X) ≥ {c}]')
plt.legend()
plt.grid(True)
```



Chebyshev Bound: It does not depend on size n. The bound is constant and depicts the worst case bound. This bound depicts the weak law of large numbers.
Empirical Estimate: It depends on size n, and the estimate becomes less noisy with increasing n.

- Chebyshev Bound: It does not depend on size n. The bound is constant and depicts the worst case bound. This bound depicts the weak law of large numbers.
- Empirical Estimate: It depends on size n, and the estimate becomes less noisy with increasing n.