Auswertung

November 8, 2024

1 physics760 - Problem Set 3

1.0.1 Team:

- Sumin Alff-Kim : 3282322
- Lennart Voorgang: 3124372

1.0.2 Code

- Repository
- Subdirectory for this assignment
- CommitID: **cd6eb40feea346f8d477dcbc3edeedf2abf12548**

2 Dependencies / Setup

We use a python `venv` environment for the project:

- Create the environment with `python3 -m venv .venv`
- To active the environment run `source .venv/bin/active` (Linux)
- To Install the required dependencies run `pip install -r requirements.txt`
- Run `jupyter notebook`

[68]:
```
# Standard imports for computation physics
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import typing
import scienceplots

# Pretty styling for graphs
plt.style.use(['science', 'grid', 'notebook'])
plt.rcParams["figure.figsize"] = (12, 8)
```

3 3.1 The Third Cumulant

Q. The variance is the second central moment, $\sigma^2 = \langle (x-\mu)^2 \rangle$. Prove it is equal to $\langle x^2 \rangle - \langle x \rangle^2$.

$$
\begin{aligned}
\sigma^2 &= \langle (x-\mu)^2 \rangle & (1)\\
&= \langle x^2 - 2x\mu + \mu^2 \rangle & (2)\\
&= \langle x^2 \rangle - \langle 2x\mu \rangle + \langle \mu^2 \rangle & (3)\\
&= \langle x^2 \rangle - 2\langle x\langle x\rangle\rangle + \langle \langle x\rangle^2 \rangle & (4)\\
&= \langle x^2 \rangle - 2\langle x\rangle\langle x\rangle + \langle x\rangle^2 & (5)\\
&= \langle x^2 \rangle - \langle x\rangle^2 & (6)
\end{aligned}
$$

Q. To third order $G(k) = e^{ik\mu - \frac{k^2}{2}\sigma^2 - \frac{ik^3}{6}k_3 + \mathcal{O}(k^4)}$. Prove that this third cumulant k_3 is the third central moment $\langle (x-\mu)^3 \rangle$.

First observe that

$$
\begin{aligned}
\langle (x-\mu)^3 \rangle &= \langle x^3 - 3x^2\mu + 3x\mu^2 - \mu^3 \rangle & (7)\\
&= \langle x^3 - 3x^2\langle x\rangle + 3x\langle x\rangle^2 - \langle x\rangle^3 \rangle & (8)\\
&= \langle x^3 \rangle - \langle 3x^2\langle x\rangle\rangle + \langle 3x\langle x\rangle^2\rangle - \langle\langle x\rangle^3\rangle & (9)\\
&= \langle x^3 \rangle - 3\langle x^2\rangle\langle x\rangle + 3\langle x\rangle\langle x\rangle^2 - \langle x\rangle^3 & (10)\\
&= \langle x^3 \rangle - 3\langle x\rangle\langle x^2\rangle + 2\langle x\rangle^3 & (11)\\
& & (12)
\end{aligned}
$$

From the lecture we know that $G(k) = \sum_{n=0}^{\infty} \frac{i^n k^n}{n!}\langle x^n \rangle$. Expand this to order 3:

$$
\begin{aligned}
G(k) = \sum_{n=0}^{\infty} \frac{i^n k^n}{n!}\langle x^n \rangle &= 1 + ik - \frac{k^2}{2}\langle x^2 \rangle - \frac{ik^3}{6}\langle x^3 \rangle + \mathcal{O}(k^4) & (13)\\
&= 1 + ik - \frac{k^2}{2}\left(\langle x^2 \rangle - \langle x\rangle^2 + \langle x\rangle^2\right) - \frac{ik^3}{6}\langle x^3 \rangle + \mathcal{O}(k^4) & (14)\\
&= 1 + ik - \frac{k^2}{2}\left(\sigma^2 + \mu^2\right) - \frac{ik^3}{6}\left(\langle x^3 \rangle - 3\langle x\rangle^3 + 3\langle x\rangle^3\right) + \mathcal{O}(k^4) & (15)\\
&= 1 + ik - \frac{k^2}{2}\left(\sigma^2 + \mu^2\right) - \frac{ik^3}{6}\left(\langle x^3 \rangle - 3\langle x\rangle\langle x\rangle^2 + 3\langle x\rangle^3\right) + \mathcal{O}(k^4) & (16)\\
&= 1 + ik - \frac{k^2}{2}\left(\sigma^2 + \mu^2\right) - \frac{ik^3}{6}\left(\langle x^3 \rangle - 3\langle x\rangle(\langle x^2\rangle - \sigma^2) + 3\langle x\rangle^3\right) + \mathcal{O}(k^4) & (17)\\
&= 1 + ik - \frac{k^2}{2}\left(\sigma^2 + \mu^2\right) - \frac{ik^3}{6}\left(\langle x^3 \rangle - 3\langle x\rangle\langle x^2\rangle + 2\langle x\rangle^3 + 3\langle x\rangle\sigma^2 + \langle x\rangle^3\right) + \mathcal{O}(k^4) & (18)\\
&= 1 + ik - \frac{k^2}{2}\left(\sigma^2 + \mu^2\right) - \frac{ik^3}{6}\left(\langle (x-\mu)^3 \rangle + 3\langle x\rangle\sigma^2 + \langle x\rangle^3\right) + \mathcal{O}(k^4) & (19)
\end{aligned}
$$

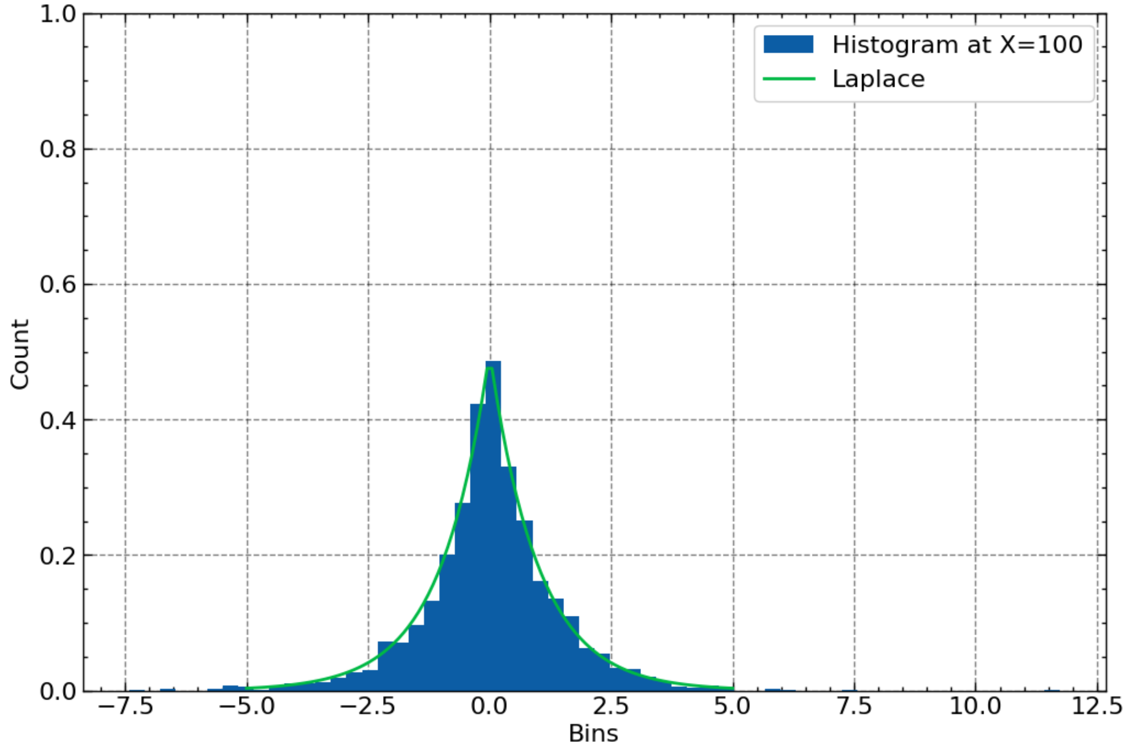
4 3.2 Evolution of the Sample Average

[69]:
```
def laplace(x: float) -> float:
    return np.exp(-np.abs(x)) / 2

rng = np.random.default_rng()
samples = rng.laplace(size=(1000, 2**11))
```

Q. For a single experiment make a histogram of all the samples x and compare it to $L_{0,1}$.

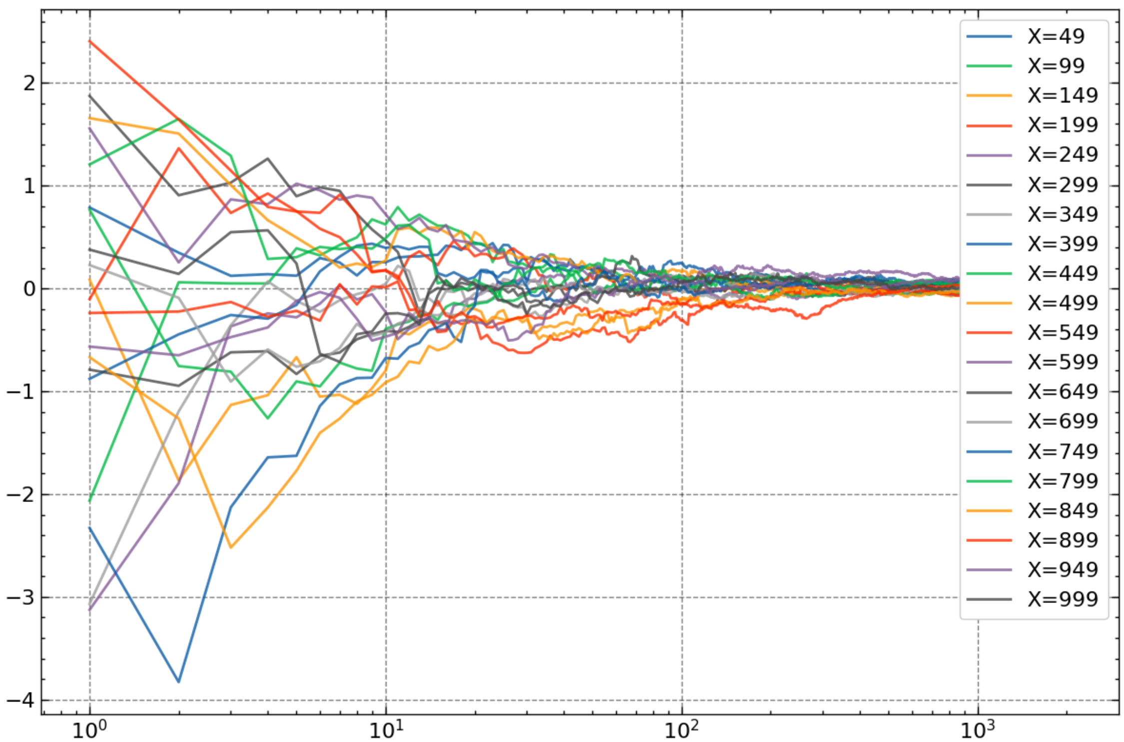
[71]:
```
laplace_sampling = np.linspace(-5, 5, num=100)

fig, ax = plt.subplots()
ax.hist(samples[100], bins=60, density=True, label=f'Histogram at X={100}')

ax.plot(laplace_sampling, laplace(laplace_sampling), label=f'Laplace')
ax.set_ylim([0, 1]);#cumulant third central moment characteristic function

ax.set_xlabel('Bins')
ax.set_ylabel('Count')
ax.legend()
```

[71]: <matplotlib.legend.Legend at 0x7efd7b65b440>



Q. For 20 of the experiments plot S_t as a function of time t. t runs from 1 (the beginning of the experiment where the sample average for any experiment is the one lonely sample in the experiment 'so far') all the way to T (the end of the experiment). You can plot them all on the same plot, or present them however you think is clear.

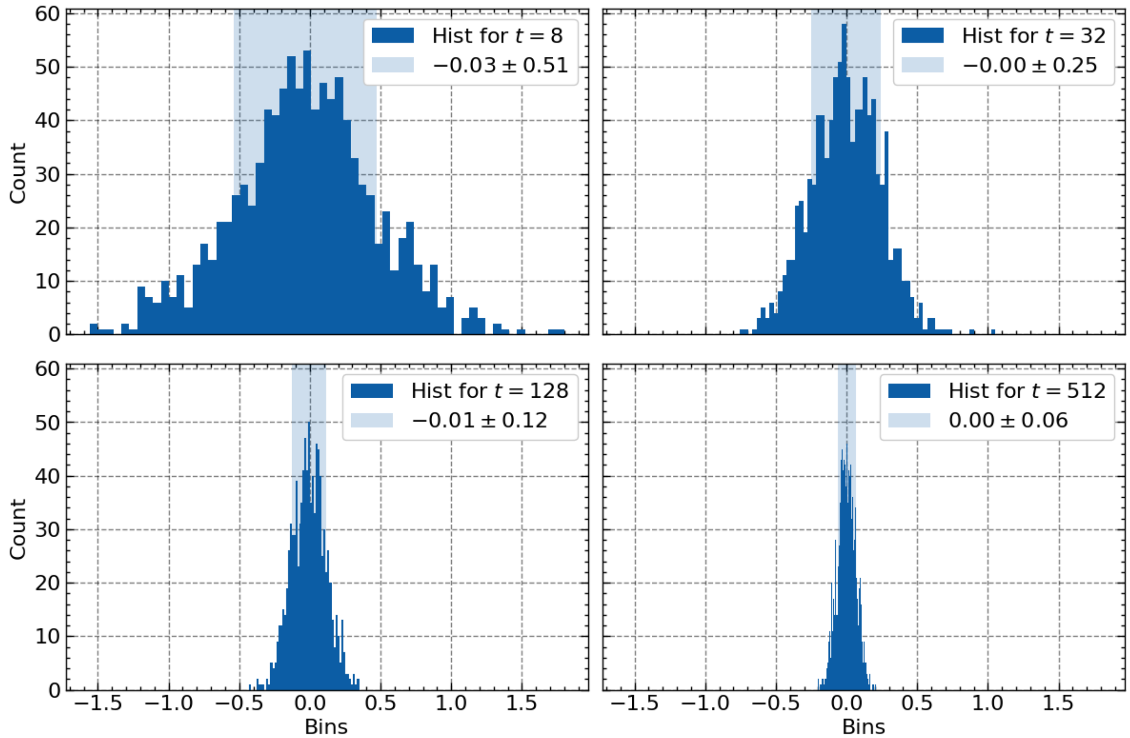
[ ]:
```
number_of_samples = np.arange(1, 2**11 + 1);
res_s = np.cumsum(samples, axis=1) / number_of_samples

fig, ax = plt.subplots()
for x in range(49, 1000, 50):
    ax.plot(number_of_samples, res_s[x], alpha=0.8, label=f'X={x}')

ax.set_xscale('log')
ax.legend(loc='upper right')

fig.tight_layout()
```



Q. Make a figure with normalized histograms of S_t for $t \in \{8, 32, 128, 512\}$. (Each histogramcontains $X = 1000$ running averages.).

[ ]:
```
fig, ax = plt.subplots(2, 2, sharex=True, sharey=True)
for idx, t in enumerate([8, 32, 128, 512]):
    mean = np.mean(res_s[:, t-1])
    std = np.std(res_s[:, t-1])
```

```
    ax[idx // 2, idx % 2].hist(res_s[:, t-1], bins=60, label=f'Hist for␣
↪$t={t}$')
    ax[idx // 2, idx % 2].axvspan(xmin=mean - std, xmax=mean + std, alpha=0.2,␣
↪label=f'${mean:.2f} \\pm {std:.2f}$')
    ax[idx // 2, idx % 2].legend()

    if idx // 2 == 1:
        ax[idx // 2, idx % 2].set_xlabel('Bins')
    if idx % 2 == 0:
        ax[idx // 2, idx % 2].set_ylabel('Count')

fig.tight_layout()
```

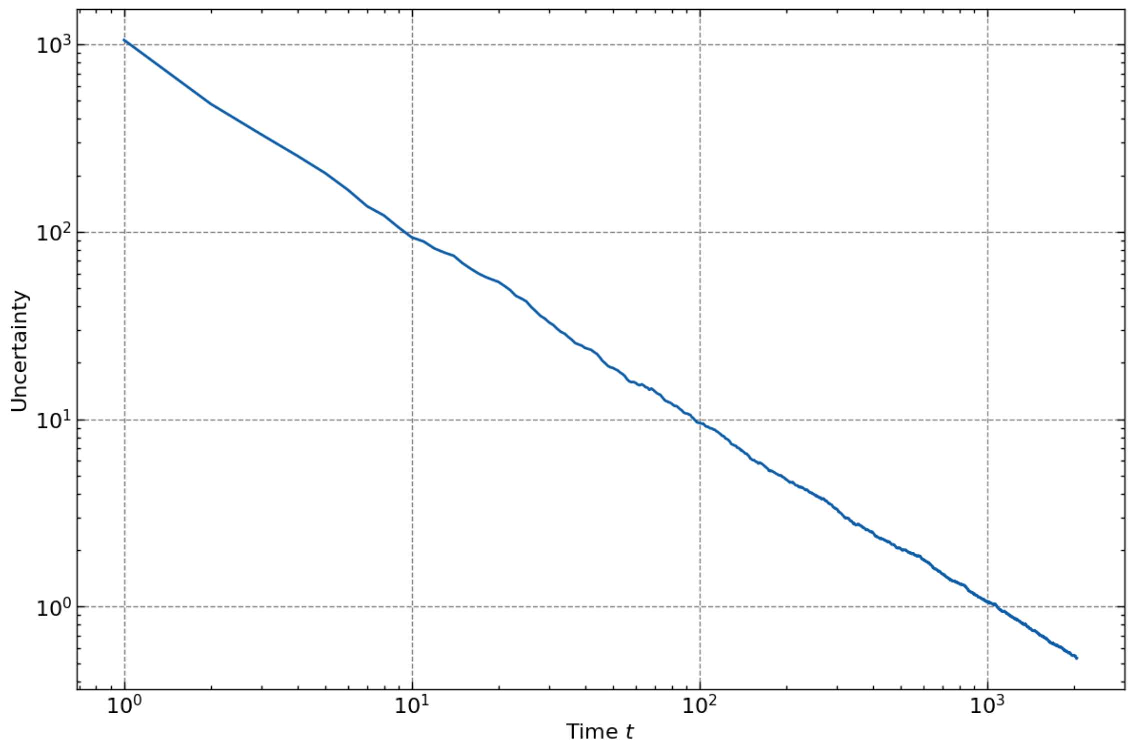


Q. Plot the estimated uncertainty of S_t as a function of t on a log-log plot. Describe what you see and explain how it is related to the CLT

In a log-log plot the Laplace distributed random numbers get linearized. Our rolling average is then just the rolling average of a uniform random number whose uncertainty, in accordance with the CLT and as seen on sheet 1, drops linearly with the number of samples.

[ ]:
```
final_s = res_s.mean()
uncertainties = np.cumsum((res_s - final_s)**2, axis=0).sum(axis=0) / 999
```

```
fig, ax = plt.subplots()
ax.loglog(number_of_samples, uncertainties)

ax.set_xlabel('Time $t$')
ax.set_ylabel('Uncertainty')

fig.tight_layout()
```

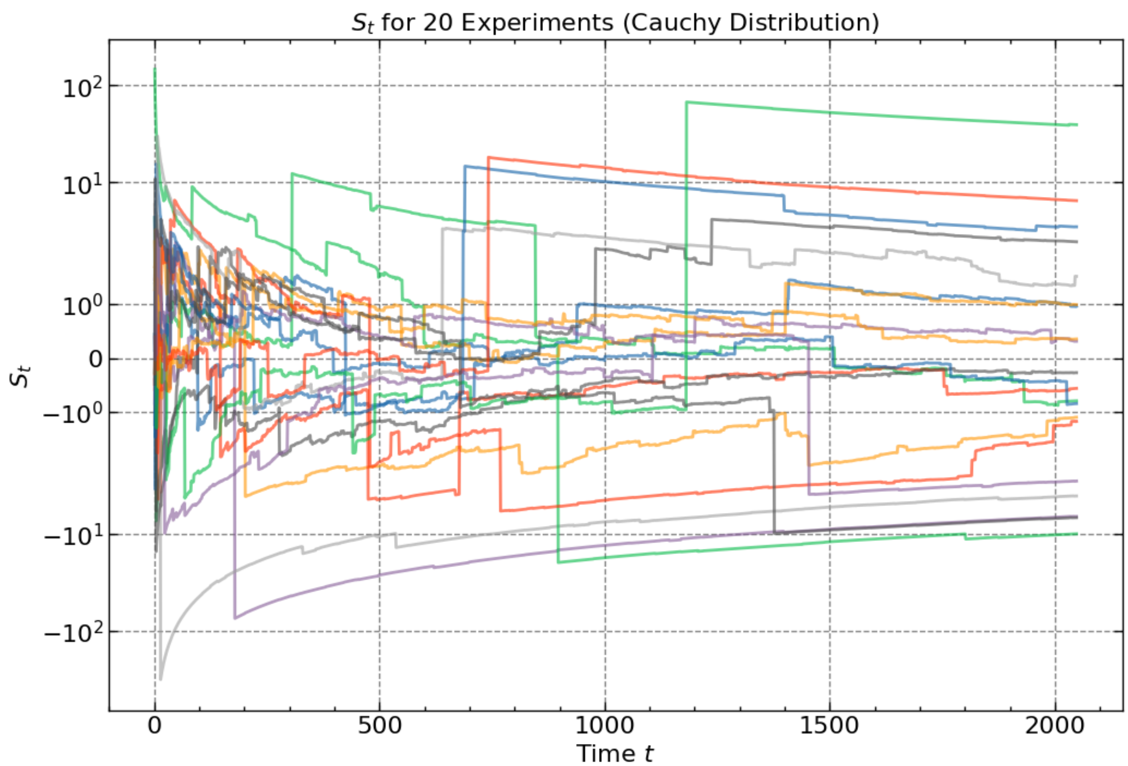


4.1 3.3 A Long-Tailed Distribution

Q. For 20 of the experiments plot S_t as a function of time t. t runs from 1 (the beginning of the experiment where the sample average for any experiment is the one lonely sample in the experiment 'so far') all the way to T (the end of the experiment).

[75]:
```
#gerate cauchy distribution of  given N experiments with iid, calculate S_t and␣
↪plot them for first 20 exp

X=1000 #N of Exp
T=2**11 #N of iid

#C_{0,1}
samples = rng.standard_cauchy(size=(X, T))
# print(len(samples))
```

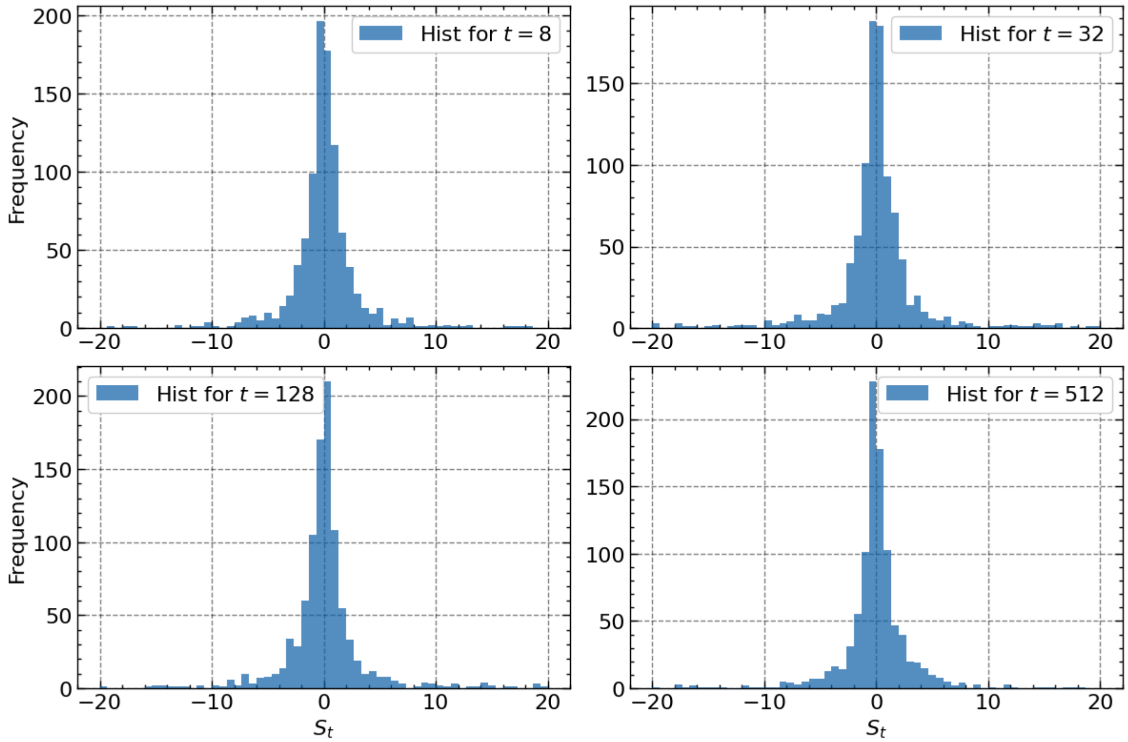
```
S_t_all = np.cumsum(samples, axis=1) / np.arange(1, T + 1)

plt.figure(figsize=(12, 8))
for i in range(20):
    plt.plot(np.arange(1, T + 1),S_t_all[i], alpha=0.6)
plt.title('$S_t$ for 20 Experiments (Cauchy Distribution)')
plt.xlabel('Time $t$')
plt.ylabel('$S_t$')
plt.yscale('symlog')  # Symmetrical log scale
plt.show()
```



Q. Make a figure with normalized histograms of St for t {8, 32, 128, 512}. (Each histogram contains X = 1000 running averages.) It might be beneficial to not show the entire histogram but to show some narrow window instead.

```
[ ]: #draw the normalised histogram of St for given t with window of t(-20, 20)
fig, ax = plt.subplots(2, 2)
for idx, t in enumerate([8, 32, 128, 512]):
    ax[idx // 2, idx % 2].hist(S_t_all[:, t-1], bins=60, range=(-20, 20),
 ↪alpha=0.7,label=f'Hist for $t={t}$')
    ax[idx // 2, idx % 2].legend()
    if idx // 2 == 1:
        ax[idx // 2, idx % 2].set_xlabel('$S_t$')
```

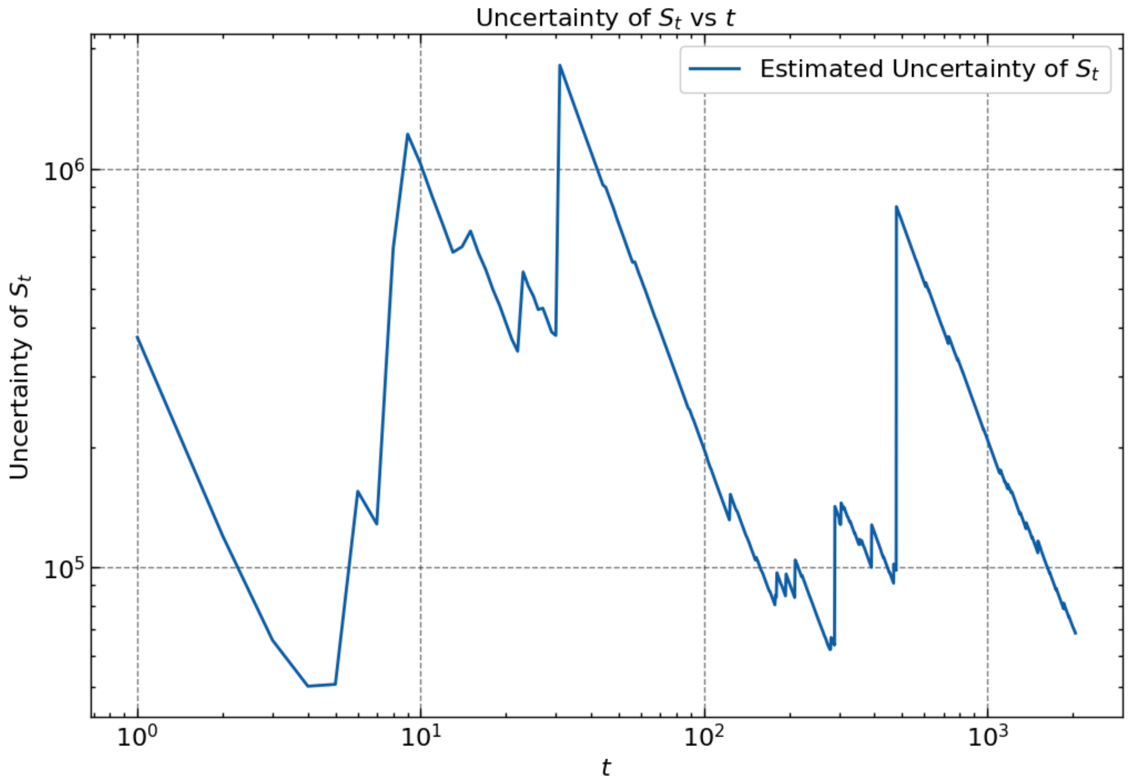
```
    if idx % 2 == 0:
        ax[idx // 2, idx % 2].set_ylabel('Frequency')
fig.tight_layout()
```



Q. Plot the estimated uncertainty of S_t as a function of t on a log-log plot. Describe what you see.

[ ]:
```
#calculate the uncertainty and plot
final_s = res_s.mean()
uncertainties = np.cumsum((S_t_all - final_s)**2, axis=0).sum(axis=0) / 999

plt.loglog(number_of_samples, uncertainties, label='Estimated Uncertainty of␣
↪$S_t$')
plt.xlabel('$t$')
plt.ylabel('Uncertainty of $S_t$')
plt.title('Uncertainty of $S_t$ vs $t$')
plt.legend()
```

[ ]: <matplotlib.legend.Legend at 0x7efd754dc5f0>



The estimated uncertainties fluctate over the time t. It shows various minimums and maximums.

Q.Why is this case so different from the Laplace case we previously examined? Provide some explanation as to why the central limit theorem fails.

The Sample average S_t of cauchy distiribution doesn't converge to certain point ($x_0 = 0$), unlike the laplace disitribution, which follows the central limit thoerem. This means that the mean μ of cauchy distribution is not defined and this leads to high uncertatinties. Also, the variance is infinity, which violates the central limit theorem.