# GEOMETRIC BROWNIAN MOTION

## Definition (Geometric Brownian Motion)

In continuous-time finance one of the most important SDEs is Geometric Brownian Motion (GBM). It is defined by the following Stochastic Differential equation:

$$S_t = S_0 + \int_0^t \mu S_u \, du + \int_0^t \sigma S_u \, dB_u$$

where S0=s0>0, and Wt is a standard Brownian motion

We will attempt to solve this stochastic differential equation:

**Solution**

Let

$dY(t) = \mu Y(t)dt + \sigma Y(t)dZ(t)$ (1)

be our Geometric Brownian Motion (GBM). Now rewrite the above equation as $dY(t) = a(Y(t), t)dt + b(Y(t), t)dZ(t)$ (2)

where $a = \mu Y(t)$, $b = \sigma Y(t)$. Both are functions of $Y(t)$ and $\boldsymbol{t}$ (albeit simple ones). Now also let $f = ln(Y(t))$. We can now apply Ito's lemma to equation (2) under the function $f = ln(Y(t))$. This leads to:

$$df \; = \; d(ln(Y(t)) = \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial Y}\, a + \frac{1}{2}\frac{\partial^2 f}{\partial Y^2}\right)dt + b\,\frac{\partial f}{\partial Y}\, dZ(t) \quad (3)$$

Now we substitute all the derivatives in (3) and the functions a and b. Note that
$\frac{\partial f}{\partial t} = \frac{\partial \ln{(Y(t))}}{\partial t} = \; 0$ (partial derivative with respect to a function Y is 0.
$\frac{\partial f}{\partial Y} = \frac{\partial \ln{(Y(t))}}{\partial Y} = \frac{1}{Y}$ and $\frac{\partial^2 f}{\partial Y^2} = \frac{-1}{Y^2}$ We finally have that:

$$(3) = \left(0 + \frac{1}{Y}\, Y\, \mu + \frac{1}{2}\, \frac{-1}{Y^2}\, \sigma^2\, Y^2\right)dt + \sigma\, Y\, \frac{1}{Y}\, dZ(t) = \left(\mu - \frac{\sigma^2}{2}\right)dt + \sigma\, dZ(t)$$

i.e.

$$d(ln(Y(t)) = \left(\mu - \frac{\sigma^2}{2}\right)dt + \sigma\, dZ(t)$$

Integrating this from 0 to t gives:

$$ln(Y(t)) - ln(Y(0)) = \left(\mu - \frac{\sigma^2}{2}\right)t + \sigma\,(Z(t) - Z(0))$$

The integral $\int_0^t dZ_u$ is, by the definition of the Ito integral, equal to $Z(t) - Z(0)$ as we are integrating the simple constant process 1 with respect to the Brownian motion.

If we rearrange and note that $Z(t) \sim N(0, t), Z(0) \sim N(0,0) = 0$ and are independent, we finally get:

$$Y(t) = Y(0) * \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma\, Z(t)\right)$$

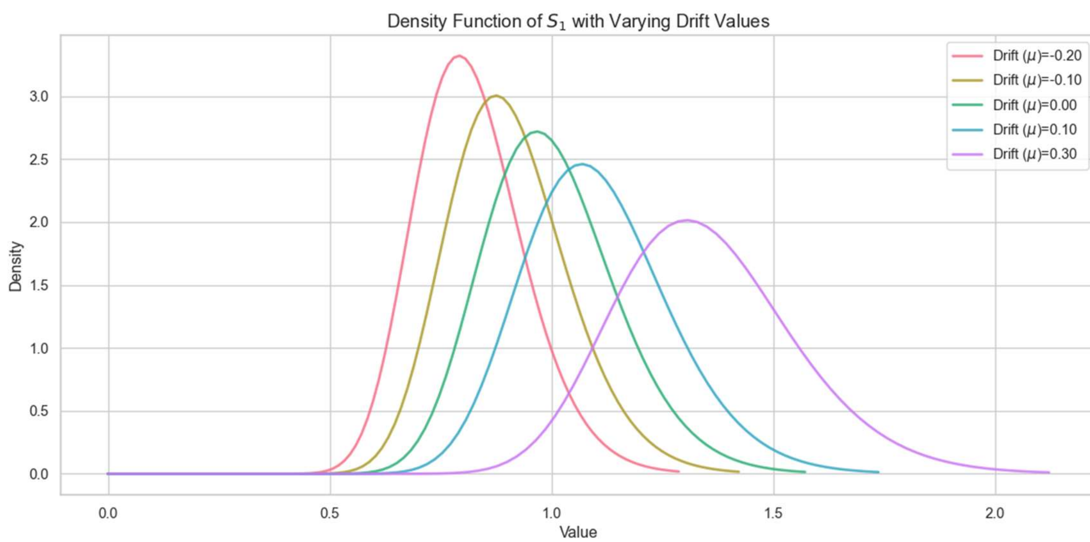
## VISUALIZATION OF THE GBM

We will now attempt to visualize the density function of a Geometric Brownian Motion all while varying parameters.

We will first attempt to vary $\mu$:

```
from scipy.stats import lognorm
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# Set a random seed for reproducibility

np.random.seed(42)

# Set style for the plots

sns.set(style="whitegrid")

# Function to generate geometric Brownian motion distribution

def geometric_brownian_motion(initial_value, drift, volatility, time_period):

    expected_log = np.log(initial_value) + (drift - 0.5 * volatility**2) * time
_period

    adjusted_volatility = volatility * np.sqrt(time_period)

    gbm_distribution = lognorm(s=adjusted_volatility, scale=np.exp(expected_log
))

    return gbm_distribution

# Plotting for different drift values

fig, ax = plt.subplots(figsize=(12, 6))

drift_values = [-0.2, -0.1, 0, 0.1, 0.3]

colors = sns.color_palette("husl", len(drift_values))

for drift, color in zip(drift_values, colors):

    distribution = geometric_brownian_motion(initial_value=1.0, drift=drift, vo
latility=0.15, time_period=1)

    x_values = np.linspace(0, distribution.ppf(0.999), 100)
```

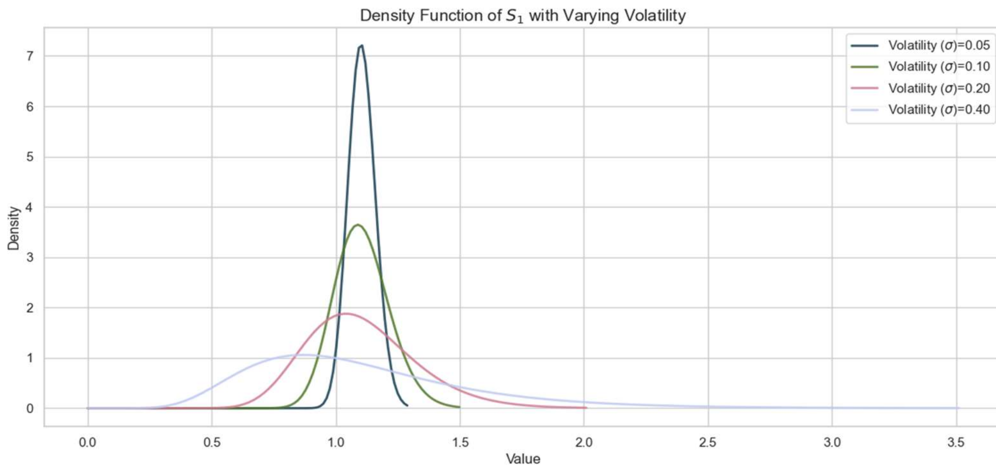
```
    ax.plot(x_values, distribution.pdf(x_values), color=color, lw=2, alpha=0.85
, label=f'Drift ($\mu$)={drift:.2f}')

ax.set_title('Density Function of $S_1$ with Varying Drift Values', fontsize=14
)

ax.set_xlabel('Value', fontsize=12)

ax.set_ylabel('Density', fontsize=12)

ax.legend()

plt.tight_layout()

plt.show()
```




Now we will attempt to do it while varying $\sigma$

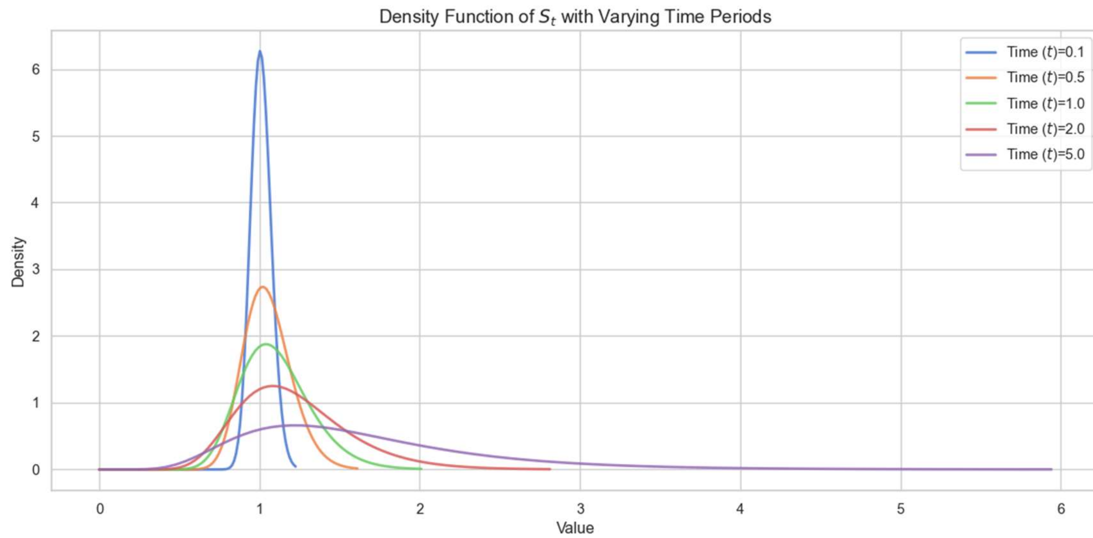
```
# Plotting for different volatility values

fig, ax = plt.subplots(figsize=(12, 6))

volatility_values = [0.05, 0.1, 0.2, 0.4]

colors = sns.color_palette("cubehelix", len(volatility_values))

for volatility, color in zip(volatility_values, colors):

    distribution = geometric_brownian_motion(initial_value=1.0, drift=0.1, vola
tility=volatility, time_period=1)

    x_values = np.linspace(0, distribution.ppf(0.999), 100)

    ax.plot(x_values, distribution.pdf(x_values), color=color, lw=2, alpha=0.85
, label=f'Volatility ($\sigma$)={volatility:.2f}')

ax.set_title('Density Function of $S_1$ with Varying Volatility', fontsize=14)

ax.set_xlabel('Value', fontsize=12)

ax.set_ylabel('Density', fontsize=12)

ax.legend()
```

```
plt.tight_layout()
plt.show()
```




Finally, we will be varying the time interval, and see how it affects our density function for the GBM:

```
# Plotting for different time periods
fig, ax = plt.subplots(figsize=(12, 6))
time_periods = [0.1, 0.5, 1, 2, 5]
colors = sns.color_palette("muted", len(time_periods))
for time, color in zip(time_periods, colors):
    distribution = geometric_brownian_motion(initial_value=1.0, drift=0.1, vola
tility=0.2, time_period=time)
    x_values = np.linspace(0, distribution.ppf(0.999), 100)
    ax.plot(x_values, distribution.pdf(x_values), color=color, lw=2, alpha=0.85
, label=f'Time ($t$)={time:.1f}')
ax.set_title('Density Function of $S_t$ with Varying Time Periods', fontsize=14
)
ax.set_xlabel('Value', fontsize=12)
ax.set_ylabel('Density', fontsize=12)
ax.legend()
plt.tight_layout()
plt.show()
```




## Expectation

For the behavior of the expectation and variance we need to use the fact that $Nt = exp(\sigma\, B_t - \frac{\sigma^2}{2}\, t\,)$ is an exponential martingale.

A rough sketch of its proof is :

$$E[N_t|F_s] = E\left[exp\left(\sigma\, B_t - \frac{\sigma^2}{2}\, t\,\right)\middle|F_s\right] = \exp\left(-\frac{\sigma^2}{2}\, t\,\right) * E[exp(\sigma\,(B_t - Bs + Bs)|F_s]$$

$E[N_t\,|F_s] = \exp\left(-\frac{\sigma^2}{2}\, t\,\right) * \exp(\sigma B_s) * \exp\left(\frac{\sigma^2}{2}\,(t-s)\right)$ We have used the moment generating function of a normal Random variable property.
Hence
$E[N_t\,|F_s]$= Ns the adaptedness is trivial since $B_t$ and $t$ are $F_t$-measurable, as for the integrability property, it is easy to show it using the moment generating function once again.

Hence for the expectation of $S_t$, a GBM:
$$E[S_t] = E\left[S_0 \exp\left\{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma\, Z(t)\right\}\right] = S_0 \exp\,(\mu * t)$$

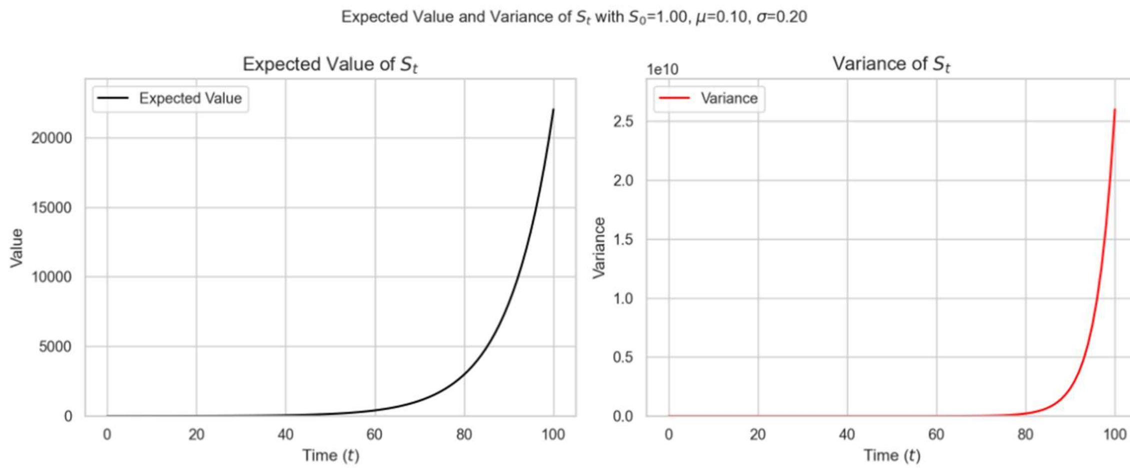
Similarly, using $Nt = exp(2\sigma\, B_t - 2\sigma^2\, t\,)$ being an exponential martingale we get
$Var[S_t] = E\left[{S_t}^2\right] - E[S_t]^2 = {S_0}^2 * \exp(\,2\mu t) * (\exp(\,\sigma^2 t) - 1)$

## Long-Time Behavior

$$\lim_{t\to\infty} E\left[S_t\right] = \lim_{t\to\infty} s_0 \exp\left\{\mu t\right\} = \begin{cases} \infty, \text{ if } \mu > 0 \\ s_0, \text{ if } \mu = 0 \\ 0, \text{ if } \mu < 0, \end{cases}$$

$$\lim_{t\to\infty} Var[S_t] = \lim_{t\to\infty} s_0^2 \exp\left\{2\mu t\right\}\left(\exp\left\{\sigma^2 t\right\} - 1\right) = \begin{cases} \infty, \text{ if } 2\mu + \sigma^2 > 0 \\ s_0^2, \text{ if } 2\mu + \sigma^2 = 0. \\ 0, \text{ if } 2\mu + \sigma^2 < 0. \end{cases}$$

```python
# Function to draw mean and variance of the modified GBM
def draw_gbm_mean_variance(initial_price, drift_rate, volatility, time_steps=100):
    fig, (mean_ax, variance_ax) = plt.subplots(1, 2, figsize=(12, 5))
    time_values = np.linspace(0, time_steps, time_steps)


    # Plotting Expected Value
    mean_ax.plot(time_values, np.exp(drift_rate * time_values), lw=1.5, color='black', label='Expected Value')
    mean_ax.set_title('Expected Value of $S_t$', fontsize=14)
    mean_ax.set_xlabel('Time ($t$)', fontsize=12)
    mean_ax.set_ylabel('Value', fontsize=12)
    mean_ax.legend()
    mean_ax.set_ylim(0, np.exp(drift_rate * time_steps) * 1.1)  # Adjusting y-limits for clarity
    # Plotting Variance
    variance_ax.plot(time_values, (initial_price**2) * np.exp(2 * drift_rate * time_values) * (np.exp(time_values * volatility**2) - 1), lw=1.5, color='red', label='Variance')
    variance_ax.set_title('Variance of $S_t$', fontsize=14)
    variance_ax.set_xlabel('Time ($t$)', fontsize=12)
    variance_ax.set_ylabel('Variance', fontsize=12)
    variance_ax.legend()
    variance_ax.set_ylim(0, (initial_price**2) * np.exp(2 * drift_rate * time_steps) * (np.exp(time_steps * volatility**2) - 1) * 1.1)  # Adjusting y-limits for clarity
    fig.suptitle(f'Expected Value and Variance of $S_t$ with $S_0$={initial_price:.2f}, $\mu$={drift_rate:.2f}, $\sigma$={volatility:.2f}', fontsize=12)
    plt.tight_layout()
    plt.show()
# Example 1
draw_gbm_mean_variance(initial_price=1.0, drift_rate=0.1, volatility=0.2, time_steps=100)
```




## SIMULATION

```
from scipy.stats import norm


def generate_time_series(start=0.0, end=1.0, num_steps=30):
    """Generate a series of time points from start to end."""
    time_interval = (end - start) / num_steps
    time_points = np.arange(start, end + time_interval, time_interval)
    return time_points


def simulate_brownian_motion(time_points, initial_value=0):
    """Simulate a Brownian motion path."""
    num_points = len(time_points)
    time_delta = (time_points[-1] - time_points[0]) / num_points
    increments = norm.rvs(loc=0, scale=np.sqrt(time_delta), size=num_points - 1
)
    increments = np.insert(increments, 0, initial_value)
    brownian_path = increments.cumsum()


    return brownian_path


# Generate time series
time_series = generate_time_series(start=1, end=10, num_steps=100)
brownian_motion = simulate_brownian_motion(time_series)
```

```python
# Parameters for the Geometric Brownian Motion

initial_price = 1

drift = 0.2

volatility = 0.25


# Calculate the Geometric Brownian Motion path

gbm_path = initial_price * np.exp((drift - 0.5 * volatility**2) * time_series +
volatility * brownian_motion)


# Plotting the Geometric Brownian Motion path

plt.figure(figsize=(10, 6))

plt.plot(time_series, gbm_path, '-', lw=1.5, label='Geometric Brownian Motion')

plt.title('Geometric Brownian Motion Path', fontsize=16)

plt.xlabel('Time', fontsize=12)

plt.ylabel('Price', fontsize=12)

plt.legend()

plt.grid()

plt.tight_layout()

plt.show()
```