Intuição matemática

September 11, 2023

0.1 Teoria do valor esperado

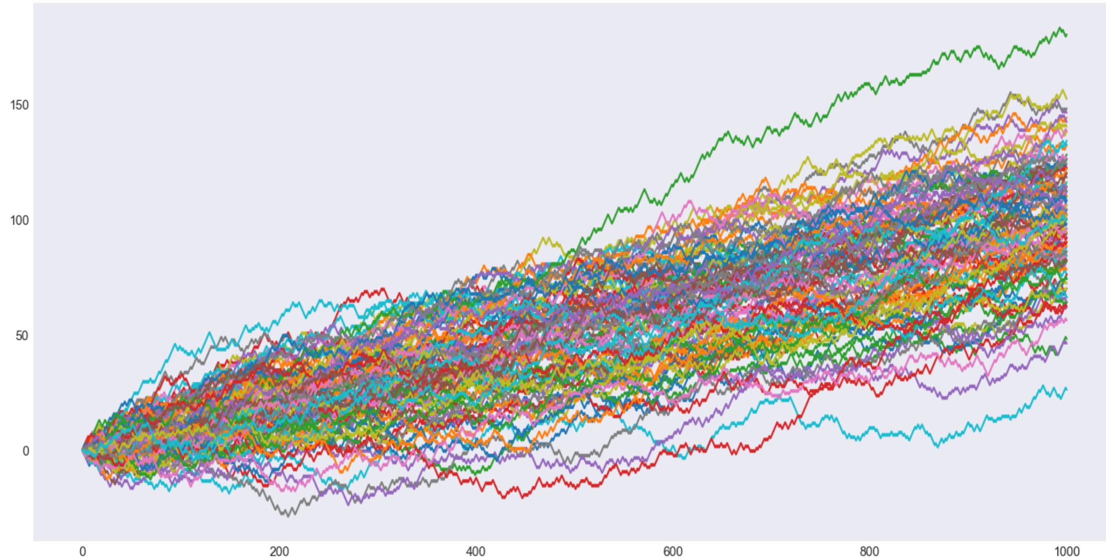
[4]:
```
import random
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
plt.style.use("seaborn-dark")
```

```
/var/folders/vq/zccc3xt90bg0bg15g24q6brh0000gp/T/ipykernel_2700/3662181781.py:5:
MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are
deprecated since 3.6, as they no longer correspond to the styles shipped by
seaborn. However, they will remain available as 'seaborn-v0_8-<style>'.
Alternatively, directly use the seaborn API instead.
  plt.style.use("seaborn-dark")
```

0.1.1 Teoria do valor esperado - Simulação de Monte Carlo

[5]:
```
capital = 100
bet_size = 1

prob_vit = 0.55
win_reward = 1

prob_loss = (1 - prob_vit)
loss_reward = -1
```

[6]:
```
fig, ax = plt.subplots(figsize=(16, 8))

for j in range(100):
    equity_curve = [0]
    for i in range(1000):
        if prob_loss < random.random():
            equity_curve += [equity_curve[-1] + win_reward * bet_size]
        else:
            equity_curve += [equity_curve[-1] + loss_reward * bet_size]
    ax.plot(equity_curve)

# ax.plot(curves.transpose())
```

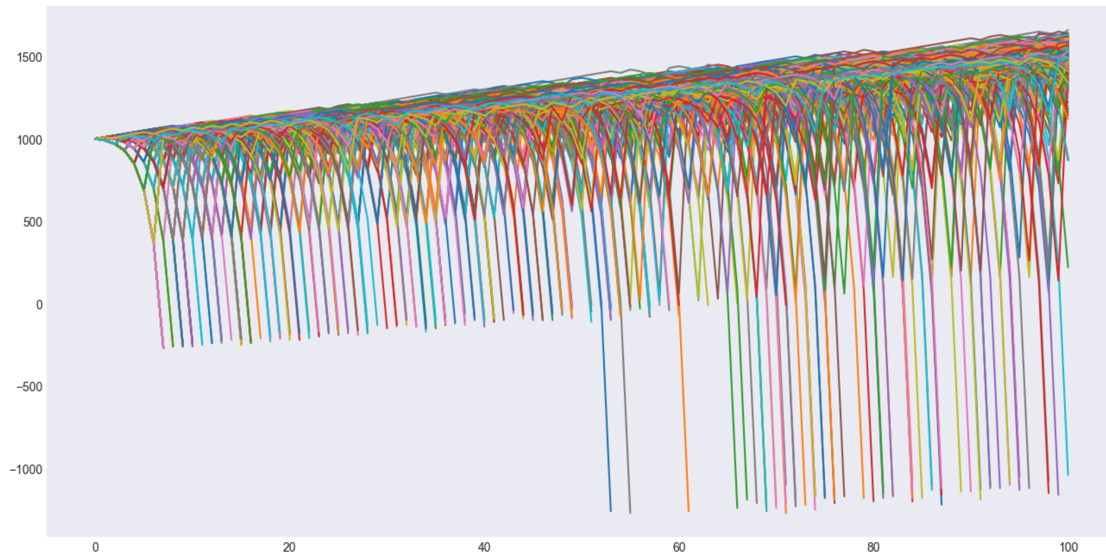


0.1.2 Simulando o efeito de um Martingale

[5]:
```
capital = 1000
bet_size = 10
bet_size_start = 10

# =========================
prob_vit = 0.5
prob_loss = (1 - prob_vit)
win_reward = 1
loss_reward = -1
```

[8]:
```
fig, ax = plt.subplots(figsize=(16, 8))

return_curve = []
curves = np.array([])

for j in range(1000):
    bet_size = 10
    equity_curve = [capital]

    for i in range(100):
        if prob_vit < random.random():
            equity_curve += [equity_curve[-1] + win_reward * bet_size]
            return_curve += [win_reward * bet_size]
            bet_size = bet_size_start

        else:
```

```
            equity_curve += [equity_curve[-1] + loss_reward * bet_size]
            return_curve += [loss_reward * bet_size]
            bet_size = bet_size * 2

            if equity_curve[-1] <= 0:
                break
    ax.plot(equity_curve)

# plt.figure(figsize=(10, 6))
# plt.plot(equity_curve)
```




```
[ ]:
```