I have a two variables A and B. I have mean_A, std_A, mean_B, std_B and also correlation rho of A and B. Now I want to generate synthetic data for A and B using a bivariate normal model and I want 1000 samples. How to do so

You can do this with a multivariate normal using NumPy. The key is to build the **covariance matrix** from the standard deviations and correlation.

- Let
  - `μ_A = mean_A`, `σ_A = std_A`
  - `μ_B = mean_B`, `σ_B = std_B`
  - correlation `ρ = rho`

Then

$$\text{cov}(A, B) = \rho \cdot \sigma_A \cdot \sigma_B$$

and the covariance matrix is

$$\Sigma = \begin{bmatrix} \sigma_A^2 & \rho\sigma_A\sigma_B \\ \rho\sigma_A\sigma_B & \sigma_B^2 \end{bmatrix}$$

Python code

python Copy code

```python
import numpy as np

# --- Given parameters ---
mean_A = 10.0
std_A  = 2.0
mean_B = 20.0
std_B  = 5.0
rho    = 0.6   # correlation between A and B

n_samples = 1000

# 1. Mean vector
mean = np.array([mean_A, mean_B])
```

```python
# 2. Covariance matrix
cov_AB = rho * std_A * std_B
cov = np.array([
    [std_A**2,  cov_AB],
    [cov_AB,    std_B**2]
])

# 3. Generate samples from bivariate normal
rng = np.random.default_rng(seed=42)  # optional seed for reproducibility
samples = rng.multivariate_normal(mean, cov, size=n_samples)

# 4. Split back into A and B
A_synth = samples[:, 0]
B_synth = samples[:, 1]

print(A_synth.shape, B_synth.shape)  # (1000,) (1000,)
```

(Optional) Check that it worked

python Copy code

```python
print("Empirical mean A, B:", A_synth.mean(), B_synth.mean())
print("Empirical std  A, B:", A_synth.std(ddof=1), B_synth.std(ddof=1))
print("Empirical corr(A,B):", np.corrcoef(A_synth, B_synth)[0, 1])
```

As `n_samples` grows, these empirical values should be close to your target `mean_A`, `std_A`, `mean_B`, `std_B`, and `rho`.