

1 Simulated SEM Results

```
import numpy as np import pandas as pd import matplotlib.pyplot as plt from pathlib  
import Path  
np.random.seed(42)
```

2 Carpetas de salida

```
out_fig = Path("results/figures") out_tab = Path("results/tables") out_fig.mkdir(parents=True, exist_ok=True) out_tab.mkdir(parents=True, exist_ok=True)
```

3 Parámetros del SEM

```
n_products = 200 n_attributes = 3 n_process_vars = 5  
beta = np.random.normal(0, 0.8, n_process_vars) lambda_k = np.array([0.9, 1.1, 0.8])  
sigma_zeta = 0.5 sigma_eps = np.array([0.4, 0.3, 0.5])
```

4 Datos simulados

```
X = np.random.normal(0, 1, (n_products, n_process_vars)) zeta = np.random.normal(0,  
sigma_zeta, n_products) eta = X @ beta + zeta  
Y = np.zeros((n_products, n_attributes)) for k in range(n_attributes): eps = np.random.normal(0, sigma_eps[k], n_products) Y[:, k] = lambda_k[k] * eta + eps  
dfY = pd.DataFrame(Y, columns=["Durability", "Appearance", "Comfort"]) corr = dfY.corr()
```

5 Guardar tabla

```
corr.to_csv(out_tab / "corr_attributes.csv", index=True)
```

6 Figuras

```
plt.figure() plt.hist(eta, bins=30) plt.title("Simulated latent quality ") plt.tight_layout()  
plt.savefig(out_fig / "eta_hist.png", dpi=200) plt.show()  
plt.figure() plt.hist(dfY["Durability"], bins=30) plt.title("Observed attribute: Durability")  
plt.tight_layout() plt.savefig(out_fig / "durability_hist.png", dpi=200) plt.show()
```

corr