data_generation

November 24, 2025

1 Data Generation

Create a script that generates random data from different distributions. Compare: (a) a normal or Gaussian distribution for different values of the variance and mean, (b) a uniformly random distribution, (c) the beta distribution.

```python
[1]: # Generate the data required by problems 1, 2 and 3
     # Ouput is a csv file with one column for each combination of distribution and
      ↪corresponding variations

     import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt

     # General parameters

     N = 100000               # Number of samples
     Seed = 100013059         # Seed for reproducibility
     output_file_path = f"./data_{N}_{Seed}.csv"
     plot = True

     rng = np.random.default_rng(seed=Seed)
     df = pd.DataFrame()
     df.index.name = "sample"

     def plot_columns(df, ncols, nrows, title, figsize=(12, 9), max_height=1):
         fig, axes = plt.subplots(nrows, ncols, figsize=figsize)
         fig.suptitle(title)
         for i in range(nrows):
             for j in range(ncols):
                 index = i*ncols + j
                 data = df[df.columns[index]]
                 subtitle = df.columns[index]
                 ax = axes[i, j]
                 ax.set_title(subtitle)
                 ax.set_xlabel("x")
                 # split the x range in 50 bins
                 counts, bin_edges = np.histogram(data, bins=50, density=True)
```

```
        # use the center of the bins for representation
        x = (bin_edges[:-1] + bin_edges[1:]) / 2
        ax.plot(x,counts,label="PDF")
        ax.set_ylim(0, max_height)
        ax.legend()
    plt.tight_layout()
    plt.show()
```

1.1 Normal/Gaussian distribution

A continuous gaussian distribution with mean μ and variance σ^2 has the following probability density function:

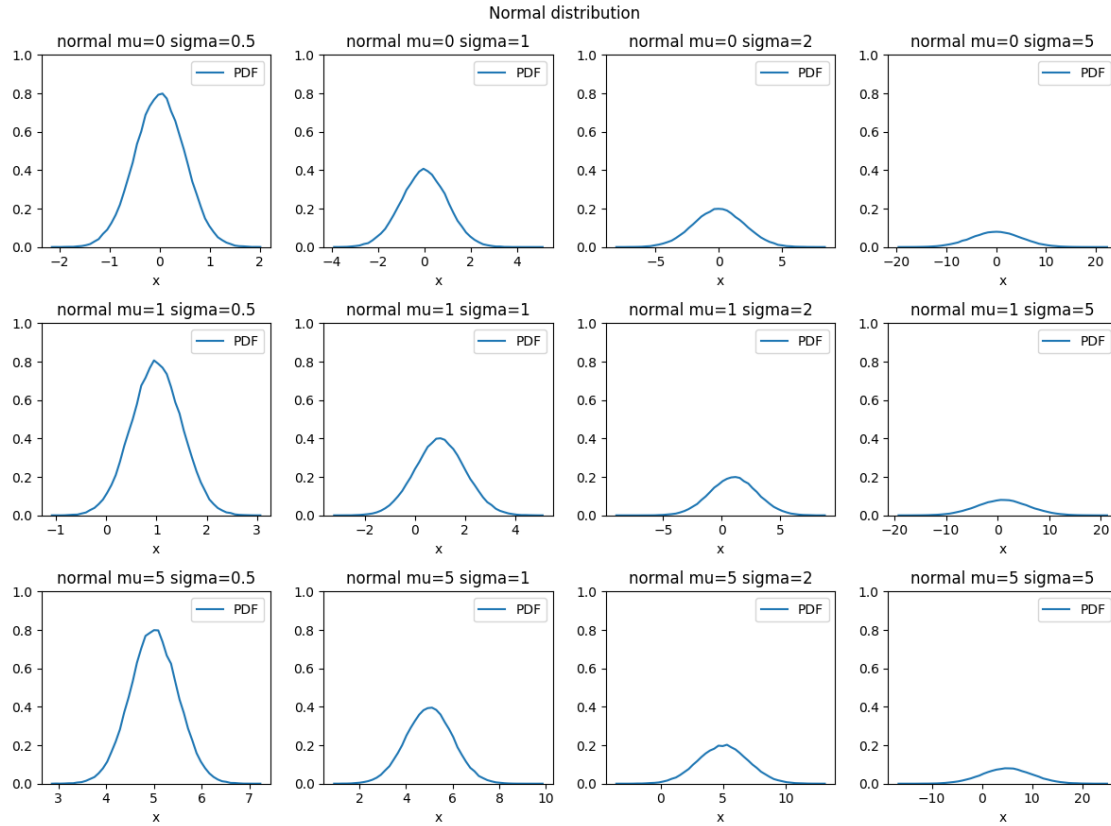$$f(x|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp{-\frac{(x-\mu)^2}{2\sigma}}$$

We can generate a sequence of values following normal distribution with given mean and variance with numpy.random.normal.

[2]:
```
# Normal/Gaussian distribution

normal_mean_values = [0, 1, 5]
normal_sigma_values = [0.5, 1, 2, 5]
for mu in normal_mean_values:
    for sigma in normal_sigma_values:
        df[f"normal mu={mu} sigma={sigma}"] = rng.normal(mu, sigma, size=N)
plot_columns(df.filter(like="normal", axis=1),4,3,"Normal distribution")
```

The probability density function of gaussian distribution is a bell-shaped function centered on the mean value. Its height is inversely proportional to the standard deviation σ. The larger the variance the wider the bell.

1.2 Uniform distribution

A continuous uniform distribution within interval [a,b] has the following probability density function

$$f(x) = \frac{1}{b - a}$$

for $a \leq x < b$ and 0 otherwise

The mean is:

$$\mu = \frac{a + b}{2}$$

And the variance is

$$\sigma^2 = \frac{(b - a)^2}{12}$$

Ref: Continuous uniform distribution

For a given pair of mean and variance values we can choose an interval that satisfies the above relations
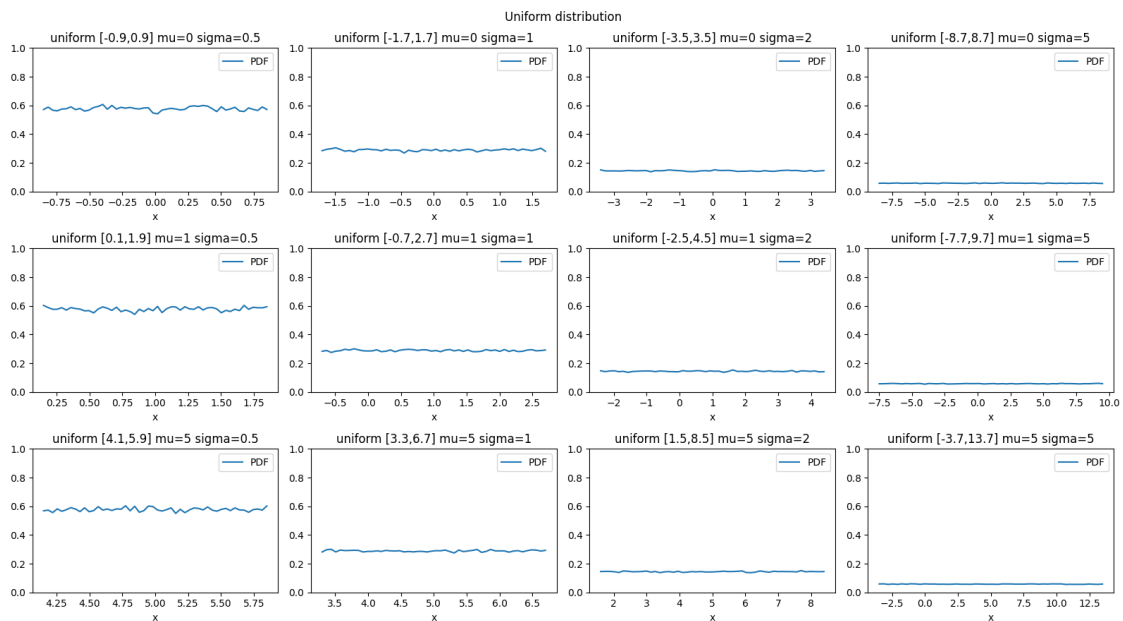
$$a = \mu - \sqrt{3}\sigma$$
$$b = \mu + \sqrt{3}\sigma$$

We can generate a sequence of values following uniform distribution within interval [a,b) with numpy.random.uniform.

[3]:
```python
# Uniformly random distribution

uniform_mean_values = [0, 1, 5]
uniform_sigma_values = [0.5, 1, 2, 5]
for mu in uniform_mean_values:
    for sigma in uniform_sigma_values:
        a = mu-np.sqrt(3)*sigma
        b = mu+np.sqrt(3)*sigma
        df[f"uniform [{a:.1f},{b:.1f}] mu={mu} sigma={sigma}"] = rng.uniform(a,↵
    ↪b, size=N)
plot_columns(df.filter(like="uniform", axis=1),4,3,"Uniform↵
    ↪distribution",figsize=(16, 9))
```


The probability density function of the uniform dstribution is a horizontal segment between the limits of the interval [a,b], centered horizontally in the middle point of the interval. Its height is inversely proportional to the interval width (b-a). Its standard deviation is directly proportional to the inerval width.

1.3 Beta distribution

A continuous beta distribution has the following probability distribution function:

$$f_X(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

for $0 < x < 1$ with beta function defined as:

$$B(\alpha, \beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}\, dx$$

Mean:

$$\mu = \frac{\alpha}{\alpha + \beta}$$

Variance:

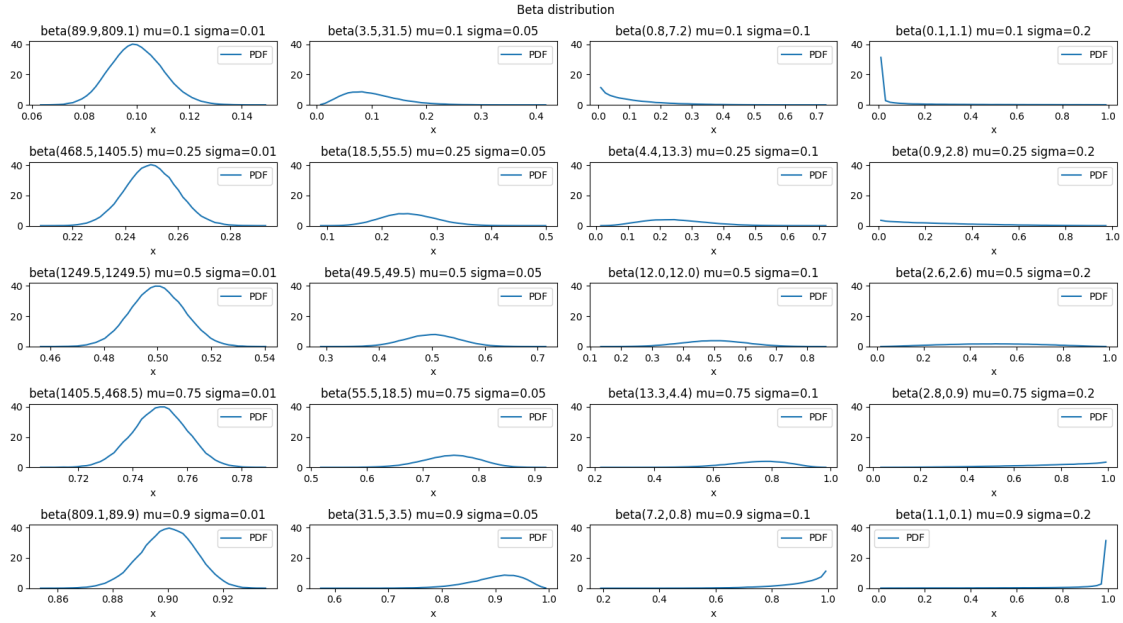$$\sigma^2 = \frac{\alpha\beta}{(\alpha + \beta + 1)(\alpha + \beta)^2}$$

For a given pair of mean and variance values we can choose a pair of α and β that satisfies the above relations

$$\alpha = \mu\left(\frac{\mu(1-\mu)}{\sigma^2} - 1\right)$$

$$\beta = (1-\mu)\left(\frac{\mu(1-\mu)}{\sigma^2} - 1\right))$$

We can generate a sequence of values following uniform distribution within interval [a,b) with numpy.random.beta.

```
[4]:  # Beta distribution

      beta_mean_values = [0.1, 0.25, 0.5, 0.75, 0.9]
      beta_sigma_values = [0.01, 0.05, 0.1, 0.2]
      for mu in beta_mean_values:
          for sigma in beta_sigma_values:
              alpha = mu*(mu*(1-mu)/(sigma*sigma)-1)
              beta = alpha*(1-mu)/mu
              df[f"beta({alpha:.1f},{beta:.1f}) mu={mu} sigma={sigma}"] = rng.
        ↪beta(alpha, beta, size=N)
      plot_columns(df.filter(like="beta", axis=1),4,5,"Beta distribution",␣
        ↪figsize=(16, 9), max_height=42)
```

Beta distribution

The probability density function of a beta distribution is defined in the [0, 1] interval. The center of mass is determined by the mean. It is symmetric with respect to the mean when $\alpha = \beta$. The larger the variance the flatter the shape

1.4 Save to file

```
[5]: df.to_csv(output_file_path)
```