Individual Assignment

1) Summary of Uniform and Bernoulli distribution

2) Formulas, basic plots in python, examples(What can be distributed as each).

3) Intuitive explanation of pareto distribution.

4) Formulas of pareto distribution.

5) Explanation of the parameters.

6) Density and cumulative distribution plots in python.

7) Examples of application.

Discrete Uniform Distribution- probability distribution where values are eaqual likely and they are finite values.

Continuous Uniform Distribution- probability distribution where each value within a certain range is equally likely to occur and values outside the range never occur. There is an infinite number of points that can exist.

Formula:

$$p(x;a,b)=\begin{cases}0 & \text{si } x\leq a\\ \frac{x-a}{b-a} & \text{si } a\leq x\leq b\\ 1 & \text{si } b\leq x\end{cases}$$

```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
import numpy as np
```

```
fig, ax = plt.subplots(1, 1)
x = np.linspace(stats.uniform.ppf(0.01),
                stats.uniform.ppf(0.99), 100)
ax.plot(x, stats.uniform.pdf(x),'r-', lw=5, alpha=0.6, label='uniform pdf')
r = stats.uniform.rvs(size=1000)
ax.hist(r, density=True, histtype='stepfilled', alpha=0.2)
ax.legend(loc='best', frameon=False)
plt.show()
```

```
uniform_data = stats.uniform.rvs(size=100000, loc = 10, scale=20)
#pd.DataFrame(uniform_data).plot()
pd.DataFrame(uniform_data).plot(kind='density')
```

EXAMPLE:

Discrete- Rolling a fair dice. The possible values area 1, 2, 3, 4, 5, 6 and they have the same probability to appear. 1/6

Continuous- Random number generator.

Bernoulli Distribution- Descrete probability distribution of a random variable which takes a binary, boolean output. It has two possible outcomes 1 with probability p and 0 with probability (1-p)

```
p =  0.5 # parametro de forma
bernoulli = stats.bernoulli(p)
x = np.arange(-1, 3)
fmp = bernoulli.pmf(x) # Función de Masa de Probabilidad
fmp
fig, ax = plt.subplots()
ax.plot(x, fmp, 'bo')
ax.vlines(x, 0, fmp, colors='b', lw=5, alpha=0.5)
ax.set_yticks([0., 0.2, 0.4, 0.6])
plt.title('Bernoulli Distribution')
plt.ylabel('probability')
plt.xlabel('values')
plt.show()
```

Example: Coint toss.

Pareto-

- Continuous porbability distribution
- Called the 80-20 rule
- Used to model distribution of incomes. 80% of the wealth is owned by 20% of the people.
- Used to describe real-world problems (social, scientific, geographycal, actuarial)

Haz doble clic (o pulsa Intro) para editar

PDF Formula: $$P(x) = \frac{a\, b^a}{x^{a+1}}$$

CDF Formula: $$D(x) = 1 - \left(\frac{b}{x}\right)^a$$

Parameters:

- a= location parameter. Sets the lower limit of time
- b= shape parameter. Affects the shape of the distribution
- x= quantiles

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "none"
from scipy.stats import pareto
import matplotlib.pyplot as plt
```

Plot Pareto probability density function

```
x_m = 1 #scale- smallest value that random variable can take
alpha = [1, 2, 3] #list of values of shape parameters
samples = np.linspace(start=0, stop=5, num=1000)

for a in alpha:
    #Probability density function
    output = np.array([pareto.pdf(x=samples, b=a, loc=0, scale=x_m)])#quantiles, shape, locat
    plt.plot(samples, output.T, label='alpha {0}' .format(a))

plt.xlabel('samples', fontsize=15)
plt.ylabel('PDF', fontsize=15)
plt.title('PARETO Probability Density function', fontsize=15)
plt.grid(b=True, color='grey', alpha=0.3, linestyle='-.', linewidth=2)
plt.rcParams["figure.figsize"] = [15, 7]
plt.legend(loc='best')
plt.show()
```

Plot Pareto comulative distribution function

```
x_m = 1 #scale- smallest value that random variable can take
alpha = [1, 2, 3] #list of values of shape parameters
samples = np.linspace(start=0, stop=5, num=1000)

for a in alpha:
  #Comulative distribution function
    output = np.array([pareto.cdf(x=samples, b=a, loc=0, scale=x_m)])#quantiles, shape, locat
    plt.plot(samples, output.T, label='alpha {0}' .format(a))

plt.xlabel('samples', fontsize=15)
plt.ylabel('PDF', fontsize=15)
plt.title('PARETO Probability Density function', fontsize=15)
plt.grid(b=True, color='grey', alpha=0.3, linestyle='-.', linewidth=2)
plt.rcParams["figure.figsize"] = [15, 7]
plt.legend(loc='best')
plt.show()
```

Examples:

- 20% of comapnies products represents 80% of sales
- 20% of the employees are responsible for 80% of the results
- 20% of drivers cause 80% of all traffic accidents