```
"""
Registration : xxxx
Description  : Generating Random Numbers & Histogram
Author       : AKB
"""

import numpy as np
from scipy.special import gamma, beta, comb, factorial # combinatorics in Binomial Distro.
from collections import Counter                        # Counter counts how many times a
number appears in Histogram
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")

if(0):
  # ==================================== #
  print ('Generation of Random Numbers')
  # ==================================== #

  # Uniform distribution within half-open interval [0.0,1.0) {include 0, exclude 1}
  a = np.random.rand(10000,5); plt.hist(a); plt.show(); print ([np.max(a), np.min(a)])
  a = np.random.random(size=10000); plt.hist(a); plt.show(); print ([np.max(a),
np.min(a)])
  a = np.random.ranf(size=10000); plt.hist(a); plt.show(); print ([np.max(a), np.min(a)])
  a = np.random.sample(size=10000); plt.hist(a); plt.show(); print ([np.max(a),
np.min(a)])

  a = np.random.randn(200000,1); plt.hist(a); plt.show(); print ([np.max(a), np.min(a)])
# Normal distribution (mean 0 variance 1)
  a = np.random.randint(1,10,size=(10000,2)); plt.hist(a); plt.show(); print ([np.max(a),
np.min(a)]) # Integers from Uniform distribution within [1,10]

  print ('Random Permutations')
  a = np.arange(10); np.random.shuffle(a); print (a);
  a = np.random.permutation(10);           print (a);

if(1):
  # ====================================== #
  print ('Plotting Various Distributions')
  # ====================================== #

  # ========= Chi Square Distribution ======== #
  df, npts, nbin = 20, 6000, 40  # Independent Normal Distributed (mu=0,sig=1) Random
Variables, Number of Points and bin
  N = np.random.chisquare(df, npts);
  plt.figure(1);
  plt.subplot(4,3,1)
  nn, bins, patches = plt.hist(N, nbin, density=True, color='darkorange',
edgecolor='teal', label=r'$N=$'+str(N.size));
  plt.plot(bins, bins**(df/2.0 - 1)*np.exp(-bins/2)/(2**(df/2.0)*gamma(df/2.0)),
linewidth=2, color='k',
           label=r'$P(x) = \frac{x^{\frac{k}{2} - 1}e^{-\frac{x}{2}}}{2^{\frac{x}{2}}
\Gamma(\frac{x}{2})}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y', alpha=0.75)
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{\chi^2}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Gamma Distribution ========= #
  k, theta, npts, nbin = 2.0, 2.0, 6000, 50
  N = np.random.gamma(k, theta, npts);
  plt.subplot(4,3,2)
  count, bins, ignored = plt.hist(N, nbin, density=True, color='skyblue',
edgecolor='blue', label=r'$N=$'+str(N.size))
  plt.plot(bins, bins**(k-1)*(np.exp(-bins/theta)/(gamma(k)*theta**k)), linewidth=2,
color='k',
```

```
         label=r'$P(x) = x^{k-1} \frac{e^{-x/\theta}}{\theta^k \Gamma(k)}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y', alpha=0.75)
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{\Gamma}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Poisson Distribution ======== #
  lam, N = 5.0, 6000
  N = np.random.poisson(lam, N);
  plt.subplot(4,3,3)
  nn, bins, patches = plt.hist(N, 'sturges', density=True, color='lightseagreen',
edgecolor='cyan', label=r'$N=$'+str(N.size));
  plt.plot(bins, lam**bins*np.exp(-lam)/factorial(bins), linewidth=2, color='k',
label=r'$P(x) = \frac{\lambda^xe^{-\lambda}}{x!}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y')
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Poisson}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Exponential Distribution ======== #
  scale, size, nbin = 1.0, 6000, 20  # Scale Parameter (inverse of Rate Parameter), size
and how many bins
  N = np.random.exponential(scale, size);
  plt.subplot(4,3,4)
  n, bins, patches = plt.hist(N, nbin, density=True, color='gold', edgecolor='maroon',
label=r'$N=$'+str(N.size));
  plt.plot(bins, 1/scale*np.exp(-bins/scale), linewidth=2, color='k', label=r'$P(x) =
\frac{1}{\beta}e^{-x/\beta}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y', alpha=0.75)
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Exp}(x)$', size=12); plt.yticks(size=10) #plt.show()

  # ========= Binomial Distribution (Discrete) ======== #
  n, p = 20, 0.6  # Total Number of Trials & Success Probability
  npts = 6000
  N = np.random.binomial(n, p, npts);
  plt.subplot(4,3,5)
  nn, bins, patches = plt.hist(N, 'sturges', density=True, color='seagreen',
edgecolor='slategray', label=r'$N=$'+str(N.size));
  plt.plot(bins, comb(n,bins)*pow(p,bins)*pow(1-p,n-bins), linewidth=2, color='k',
label=r'$P(x) = \frac{n!}{k!(n-k)!}p^k (1-p)^{n-k}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y', alpha=0.75)
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Binom}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Gaussian Distribution ======== #
  npts, nbin = 6000, 40; # Total Number of Points & Bins
  mu, sigma = 0.0, 10;  # Mean & Standard Deviation of PDF
  x = np.random.normal(mu, sigma, npts);
  plt.subplot(4,3,6)
  n, bins, patches = plt.hist(x, nbin, density=True, color='tan', edgecolor='brown',
label=r'$N=$'+str(npts))
  plt.plot(bins, 1/(sigma*np.sqrt(2*np.pi))*np.exp(-(bins-mu)**2/(2*sigma**2)),
         linewidth=2, color='k',label=r'$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-
\mu)^2/{2\sigma^2}}$')
  plt.legend(loc='best',prop={'size':11})
  plt.title(r'$\mu='+str(mu)+',\ \sigma='+str(sigma)+'$', size=10)
  plt.grid(axis='y', alpha=0.75)
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Gauss}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Weibull Distribution (One parameter) ======== #
  shape, size, nbin = 1.5, 6000, 20
  N = np.random.weibull(shape, size);
```

```
  plt.subplot(4,3,7)
  n, bins, patches = plt.hist(N, nbin, density=True, color='chocolate',
edgecolor='firebrick', label=r'$N=$'+str(N.size));
  plt.plot(bins, shape*(bins)**(shape-1)*np.exp(-(bins**shape)), linewidth=2, color='k',
label=r'$P(x) = kx^{k-1}e^{-x^k}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y')
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Weibull}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Rayleigh Distribution ======== #
  sigma, npts = 3.0, 6000
  N = np.random.rayleigh(sigma, npts);
  plt.subplot(4,3,8)
  count, bins, ignored = plt.hist(N, 'fd', density=True, color='orchid',
edgecolor='crimson', label=r'$N=$'+str(N.size))
  plt.plot(bins, bins*np.exp(-(bins**2)/(2*sigma**2))/(sigma**2), linewidth=2, color='k',
           label=r'$P(x) = \frac{x}{\sigma^2}e^{\frac{-x^2}{2\sigma^2}}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y')
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Rayleigh}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Beta Distribution ======== #
  npts, nbin = 6000, 30;
  alpha, Beta = 2.0,2.0
  x = np.random.beta(alpha, Beta, npts);
  plt.subplot(4,3,9)
  n, bins, patches = plt.hist(x, nbin, density=True, color='olive', edgecolor='yellow',
label=r'$N=$'+str(npts))
  plt.plot(bins, (bins**(alpha-1))*((1-bins)**(Beta-1))/beta(alpha,Beta), linewidth=2,
color='k',
           label=r'$P(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y')
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Beta}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Laplace Distribution ======== #
  npts, nbin = 6000, 40;
  mu, b = 0.0, 4.0;
  x = np.random.laplace(mu, b, npts);
  plt.subplot(4,3,10)
  n, bins, patches = plt.hist(x, nbin, density=True, color='indigo', edgecolor='azure',
label=r'$N=$'+str(npts))
  plt.plot(bins, np.exp(-np.abs(bins-mu)/b)/(2*b), linewidth=2, color='k', label=r'$P(x) =
\frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y')
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Laplace}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Geometric Distribution ======== #
  npts, p = 6000, 0.15;
  x = np.random.geometric(p, npts);
  plt.subplot(4,3,11)
  n, bins, patches = plt.hist(x, 'sturges', density=True, color='cyan',
edgecolor='purple', label=r'$N=$'+str(npts))
  plt.plot(bins, p*(1-p)**(bins-1), linewidth=2, color='k', label=r'$P(x) = p(1-
p)^{x-1}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y')
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Geometric}(x)$', size=12); plt.yticks(size=10); #plt.show()

  # ========= Wald Distribution ======== #
```

```
  scale, size, mean = 3.0, 6000, 1.0  # Scale Parameter (inverse of Rate Parameter), size
and how many bins
  N = np.random.wald(mean, scale, size);
  plt.subplot(4,3,12)
  n, bins, patches = plt.hist(N, 'sturges', density=True, color='purple',
edgecolor='gold', label=r'$N=$'+str(N.size));
  plt.plot(bins, np.sqrt(scale/(2.0*np.pi*bins**3))*np.exp(-scale*(bins-mean)**2/
(2*mean**2*bins)),
           linewidth=2, color='k', label=r'$P(x) = \sqrt{\frac{\lambda}{2\pi x^3}}e^{-
\frac{\lambda(x-\mu)^2}{2 \mu^2 x}}$')
  plt.legend(loc='best',prop={'size':11})
  plt.grid(axis='y', alpha=0.75)
  plt.xlim([min(bins), max(bins)]); plt.xlabel('x', size=12); plt.xticks(size=12)
  plt.ylabel('$P_{Wald}(x)$', size=12); plt.yticks(size=10); plt.show()
```