SDA ASSIGNMENT-2

Members:

Sayam Kumar S20180010158

Hrishabh Pandey S20180010064

Raahul Singh S20180010141

Ram Nad S20180010145

1.Chi-Square Distribution:

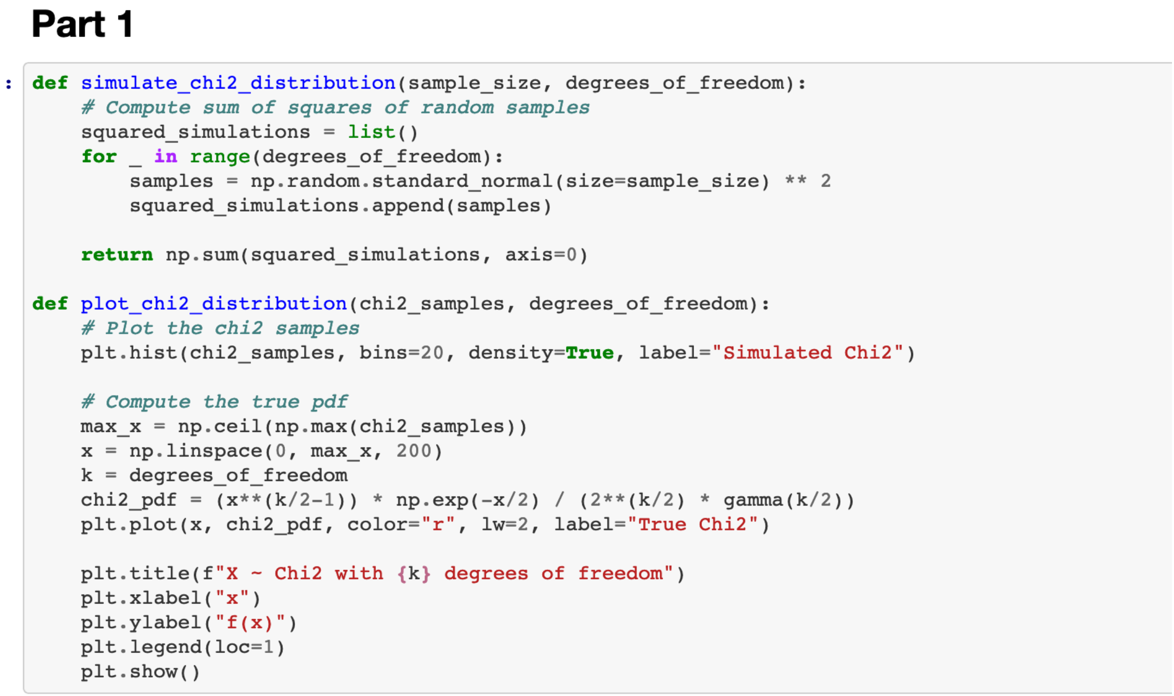
Chi-square distribution with k degrees of freedom is the sum of squares of k standard normal variables.

PDF $= \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} * e^{-x/2}$ k=degrees of freedom

- In terms of standard normal variables,consider st1,st2,st3….stk be k standard normal variables then

  $st1^2 + st2^2 + st3^2 +$ ----------- $+ stk^2 = \chi^2 (k)$

- The following piece of code generates 3 samples of size 1000 and add the squares of each of the sample and compare it to χ^2 (3) distribution.The following plots depict the similarity between them.



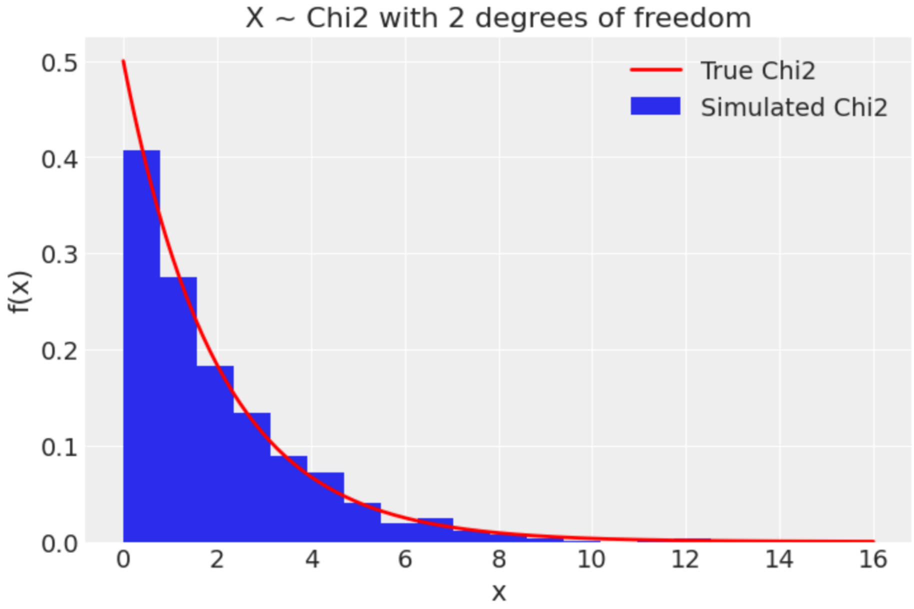
Part 1

```
def simulate_chi2_distribution(sample_size, degrees_of_freedom):
    # Compute sum of squares of random samples
    squared_simulations = list()
    for _ in range(degrees_of_freedom):
        samples = np.random.standard_normal(size=sample_size) ** 2
        squared_simulations.append(samples)

    return np.sum(squared_simulations, axis=0)

def plot_chi2_distribution(chi2_samples, degrees_of_freedom):
    # Plot the chi2 samples
    plt.hist(chi2_samples, bins=20, density=True, label="Simulated Chi2")

    # Compute the true pdf
    max_x = np.ceil(np.max(chi2_samples))
    x = np.linspace(0, max_x, 200)
    k = degrees_of_freedom
    chi2_pdf = (x**(k/2-1)) * np.exp(-x/2) / (2**(k/2) * gamma(k/2))
    plt.plot(x, chi2_pdf, color="r", lw=2, label="True Chi2")

    plt.title(f"X ~ Chi2 with {k} degrees of freedom")
    plt.xlabel("x")
    plt.ylabel("f(x)")
    plt.legend(loc=1)
    plt.show()
```

Plots:

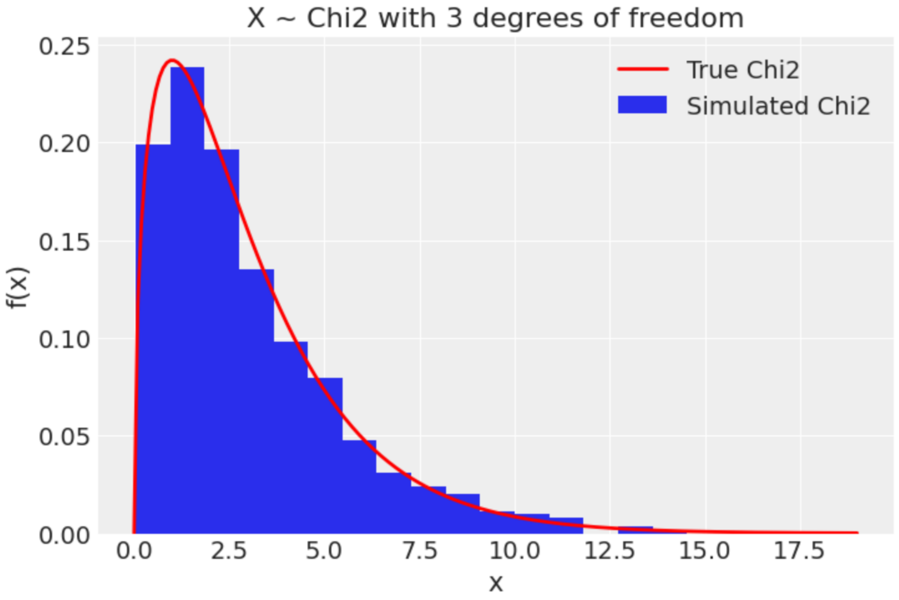
]:

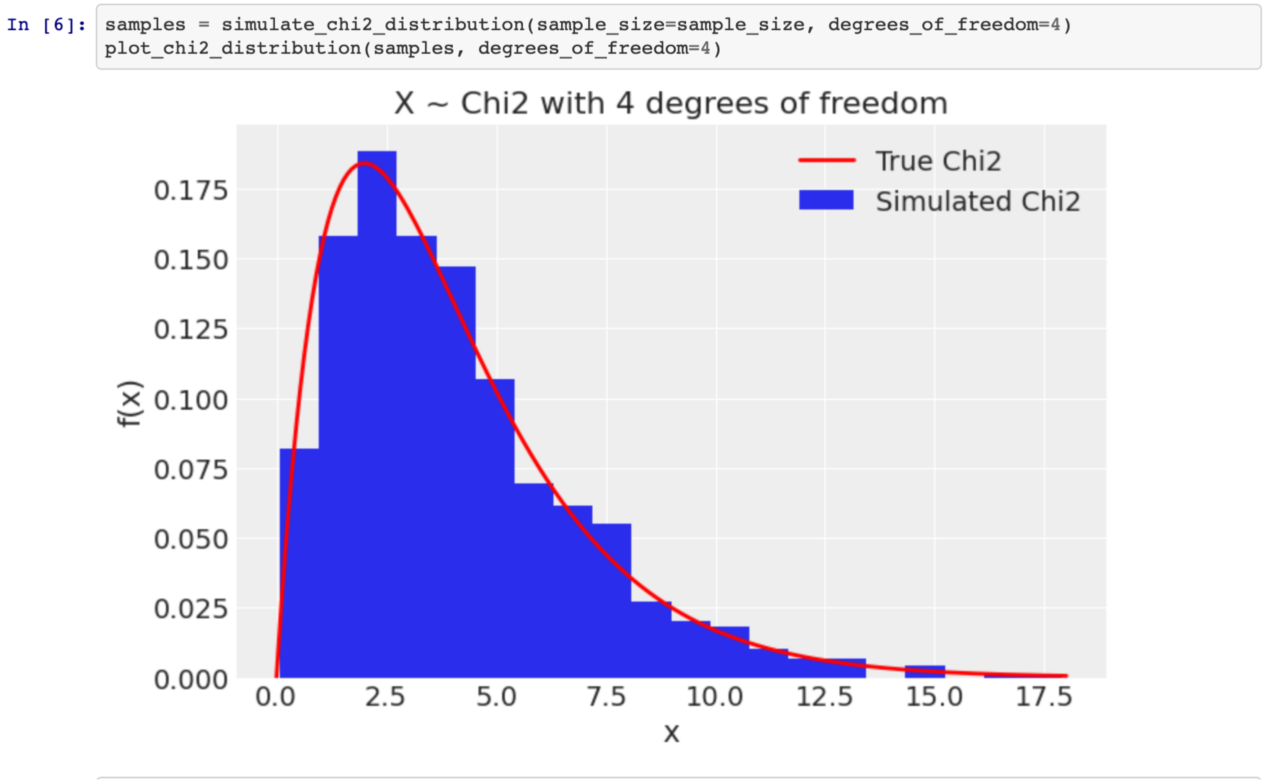
```
samples = simulate_chi2_distribution(sample_size=sample_size, degrees_of_freedom=2)
plot_chi2_distribution(samples, degrees_of_freedom=2)
```



5]:

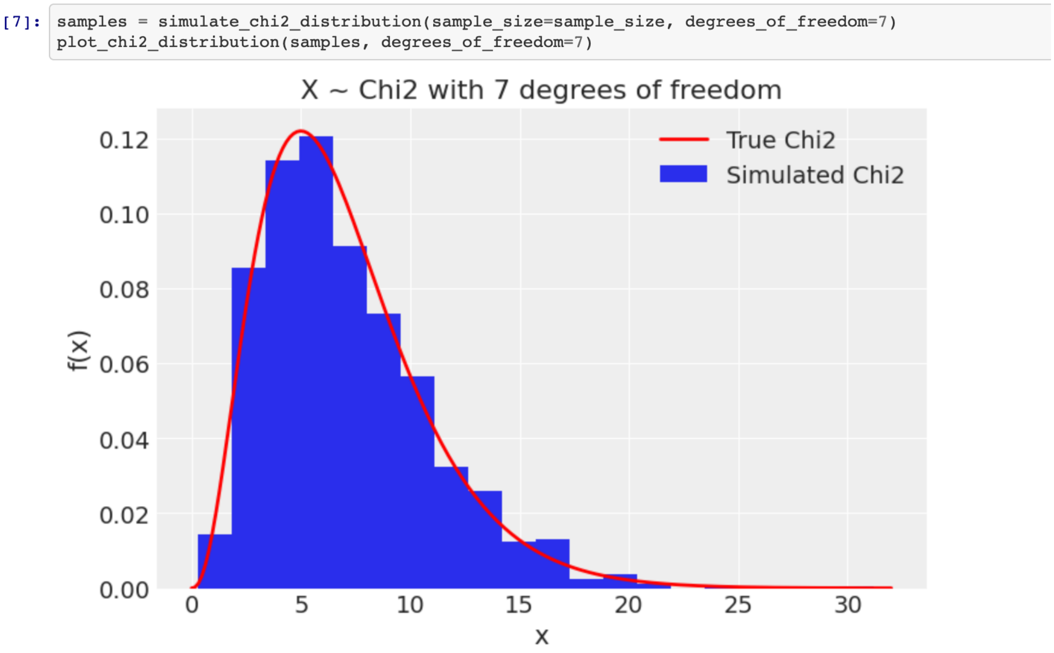
```
samples = simulate_chi2_distribution(sample_size=sample_size, degrees_of_freedom=3)
plot_chi2_distribution(samples, degrees_of_freedom=3)
```





```
In [6]: samples = simulate_chi2_distribution(sample_size=sample_size, degrees_of_freedom=4)
        plot_chi2_distribution(samples, degrees_of_freedom=4)
```

x

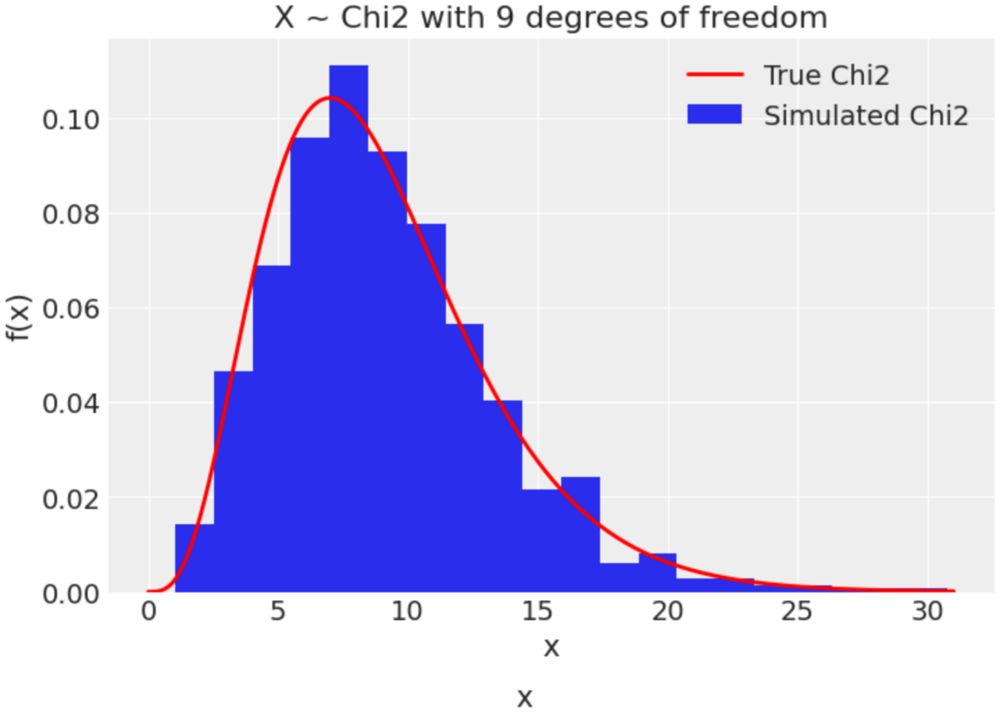


```
[7]: samples = simulate_chi2_distribution(sample_size=sample_size, degrees_of_freedom=7)
     plot_chi2_distribution(samples, degrees_of_freedom=7)
```

Caption

In [8]:
```
samples = simulate_chi2_distribution(sample_size=sample_size, degrees_of_freedom=9)
plot_chi2_distribution(samples, degrees_of_freedom=9)
```

In



In the above plots blue colour depicts the histogram of sum of squares of standard normal variables and the red line depicts the plot of chi-square distribution.

Conclusion:

From the above plots we can clearly see that the histograms of sum of squares of standard normal variables(blue) approximately overlap with the theoretical chi-square plot(red).Hence we can say that sum of squares of standard normal variables follow chi-square distribution with degrees of freedom being the number of standard normal variables considered.

2 . t-Distribution:

T distribution is a continuous probability distribution of the z-score when the estimated standard deviation is used in the denominator rather than the true standard deviation. T-tests are used in statistics to estimate significance.

$$t = \frac{Z}{\frac{\chi^2(v)}{v}}$$

$Z(0, 1)$ - Standard Normal Distribution variables with $= 0$, $^2 = 0$

$^2(v)$ - chi-square values

v - degrees of freedom

- In T distribution we compute the above equation by generating samples according to the simulations.
- Here, 1000 standard normal variables are generated using scipy.stats, $^2(v)$ value is obtained by squaring and adding the s.n.v 's for degrees of freedom times.
- For v = [3,5,10,15] t is simulated and histograms are plotted. Those histograms are verified by plotting the chi-square curve on the same plot.
- Plotted Graphs and Code are :

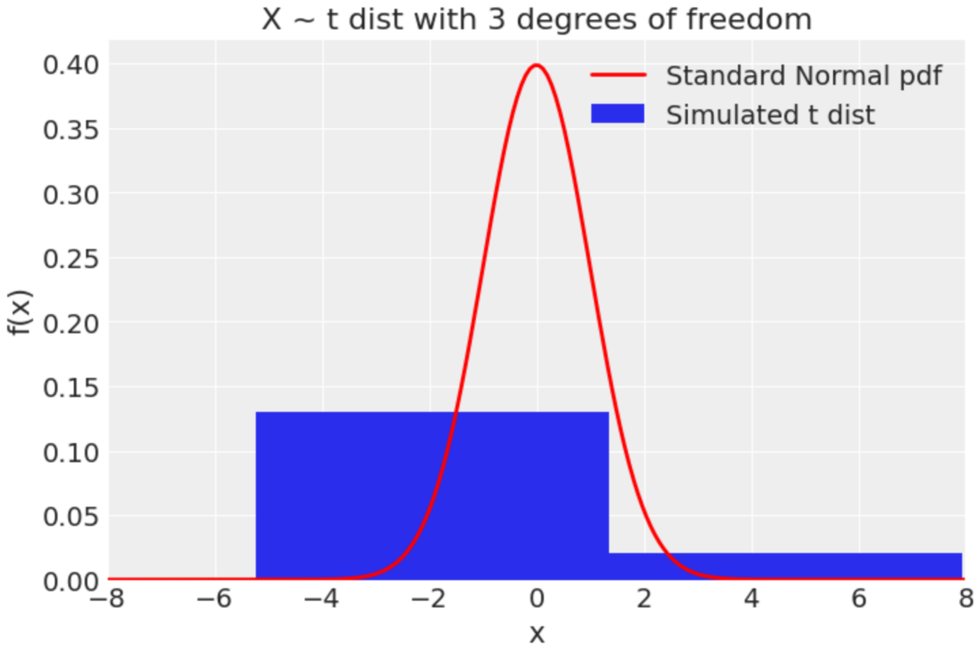
Part 2

9]:

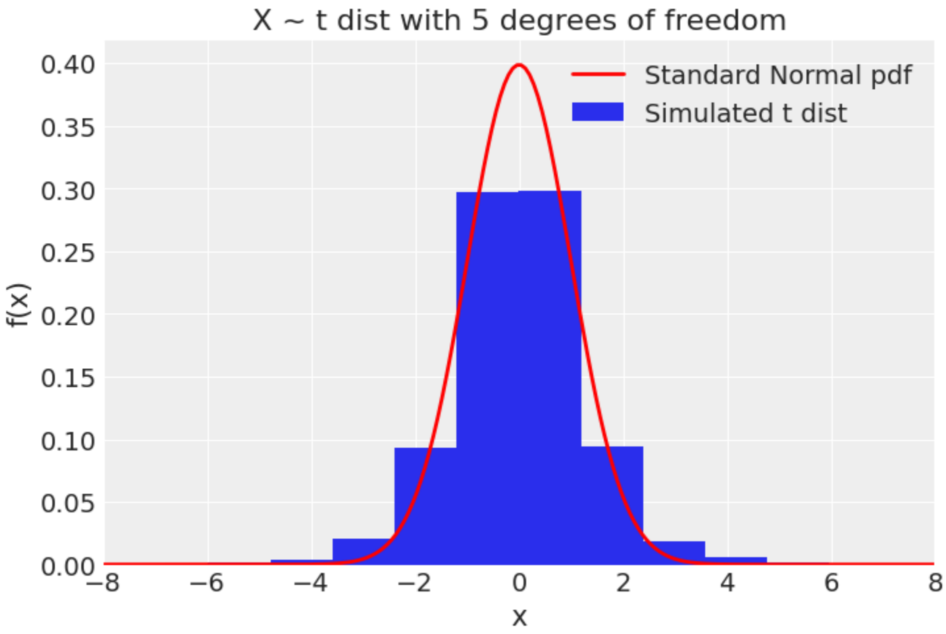
```
def simulate_t_distribution(sample_size, degrees_of_freedom):
    # Compute t distribution samples
    k = degrees_of_freedom
    Z_samples = np.random.standard_normal(size=sample_size)
    chi2_samples = simulate_chi2_distribution(sample_size, degrees_of_freedom=k)
    t_samples = Z_samples / np.sqrt(chi2_samples/k)

    # Plot
    plt.hist(t_samples, bins=20, density=True, label="Simulated t dist")
    x = np.linspace(-8, 8, 200)
    z_pdf = np.exp(-(x**2)/2) / np.sqrt(2*np.pi)
    plt.plot(x, z_pdf, color="r", lw=2, label="Standard Normal pdf")

    plt.title(f"X ~ t dist with {k} degrees of freedom")
    plt.xlabel("x")
    plt.ylabel("f(x)")
    plt.legend(loc=1)
    plt.xlim((-8, 8))
    plt.show()
```

In [10]:

```
simulate_t_distribution(sample_size=10_000, degrees_of_freedom=3)
```

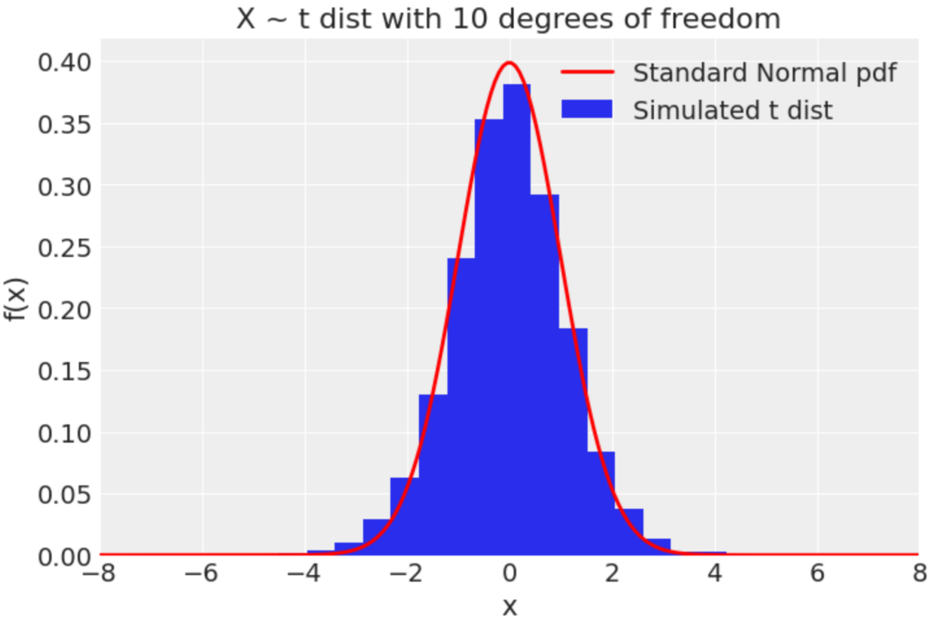


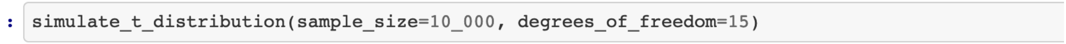
```
[11]: simulate_t_distribution(sample_size=10_000, degrees_of_freedom=5)
```



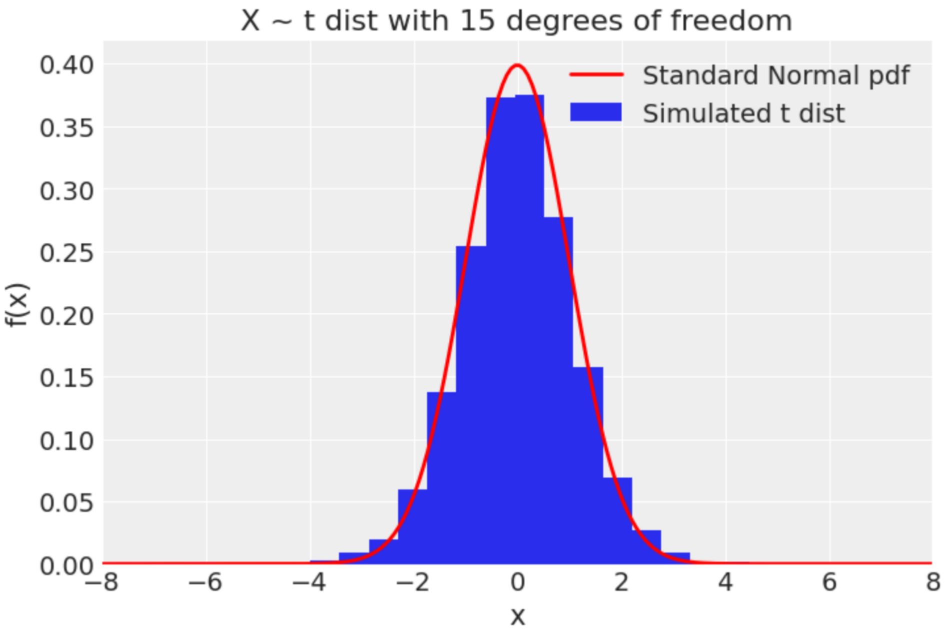
x

```
2]: simulate_t_distribution(sample_size=10_000, degrees_of_freedom=10)
```





```
: simulate_t_distribution(sample_size=10_000, degrees_of_freedom=15)
```



3. F-distribution:

A random variate of the *F*-distribution with parameters n1 and n2 is the ratio of two appropriately scaled chi-squared random variates.

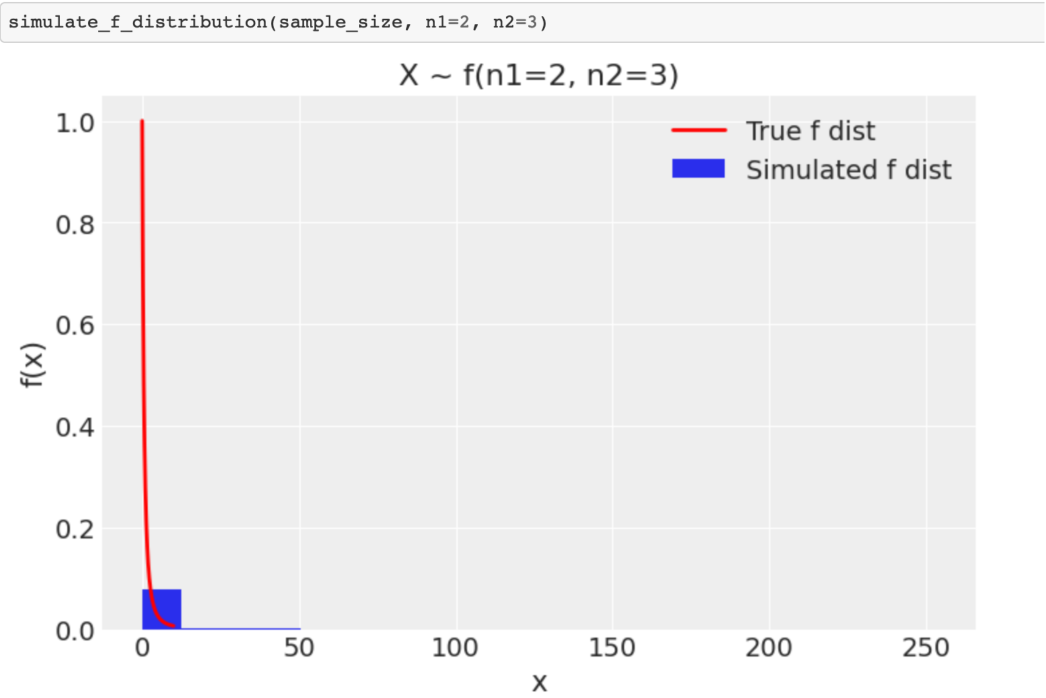
$$X = (U1/n1)/(U2/n22)$$

Where U1,U2 are independent chi-squared distributions $X^2(d1), X^2(d2)$ degrees of freedom respectively.

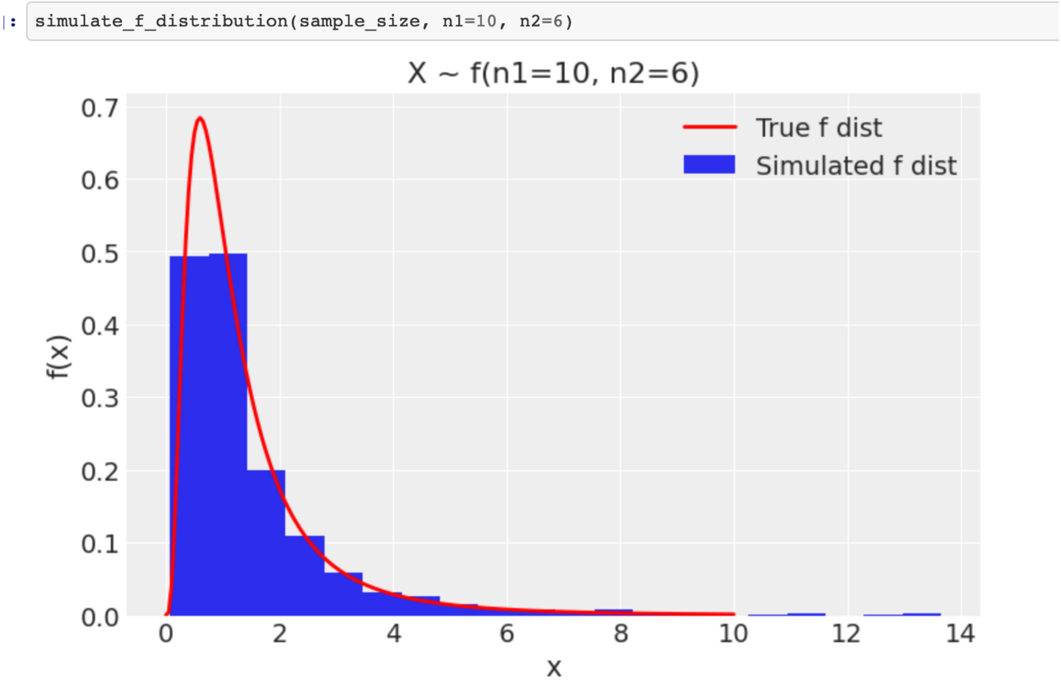
Code:

Part 3

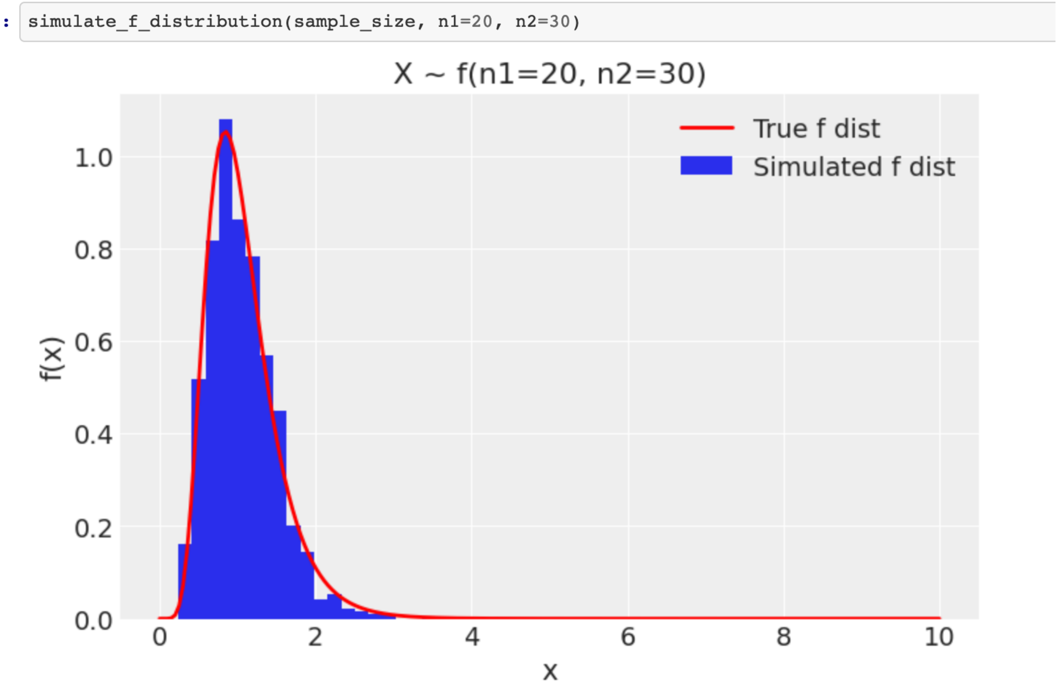
```
[14]: def simulate_f_distribution(sample_size, n1, n2):

          u1_samples = simulate_chi2_distribution(sample_size, degrees_of_freedom=n1)
          u2_samples = simulate_chi2_distribution(sample_size, degrees_of_freedom=n2)
          f_samples = (u1_samples/n1) / (u2_samples/n2)

          # Plot
          plt.hist(f_samples, bins=20, density=True, label="Simulated f dist")
          x = np.linspace(0, 10, 200)
          f_pdf = (n1**(n1/2)) * (n2**(n2/2)) * (x**((n1/2)-1)) / ((n1*x + n2)**((n1+n2)/2) * beta(n1/2
      n2/2))
          plt.plot(x, f_pdf, color="r", lw=2, label="True f dist")

          plt.title(f"X ~ f(n1={n1}, n2={n2})")
          plt.xlabel("x")
          plt.ylabel("f(x)")
          plt.legend(loc=1)
          plt.show()
```



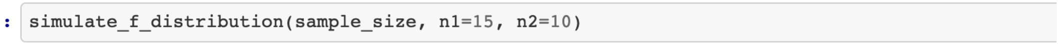
```
simulate_f_distribution(sample_size, n1=2, n2=3)
```



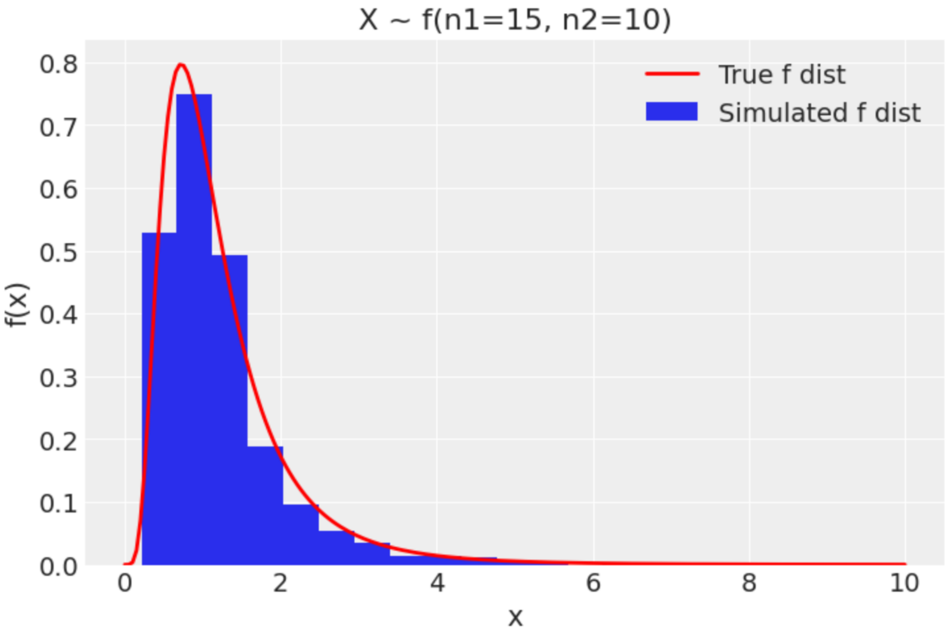
```
simulate_f_distribution(sample_size, n1=10, n2=6)
```



```
simulate_f_distribution(sample_size, n1=20, n2=30)
```



```
: simulate_f_distribution(sample_size, n1=15, n2=10)
```



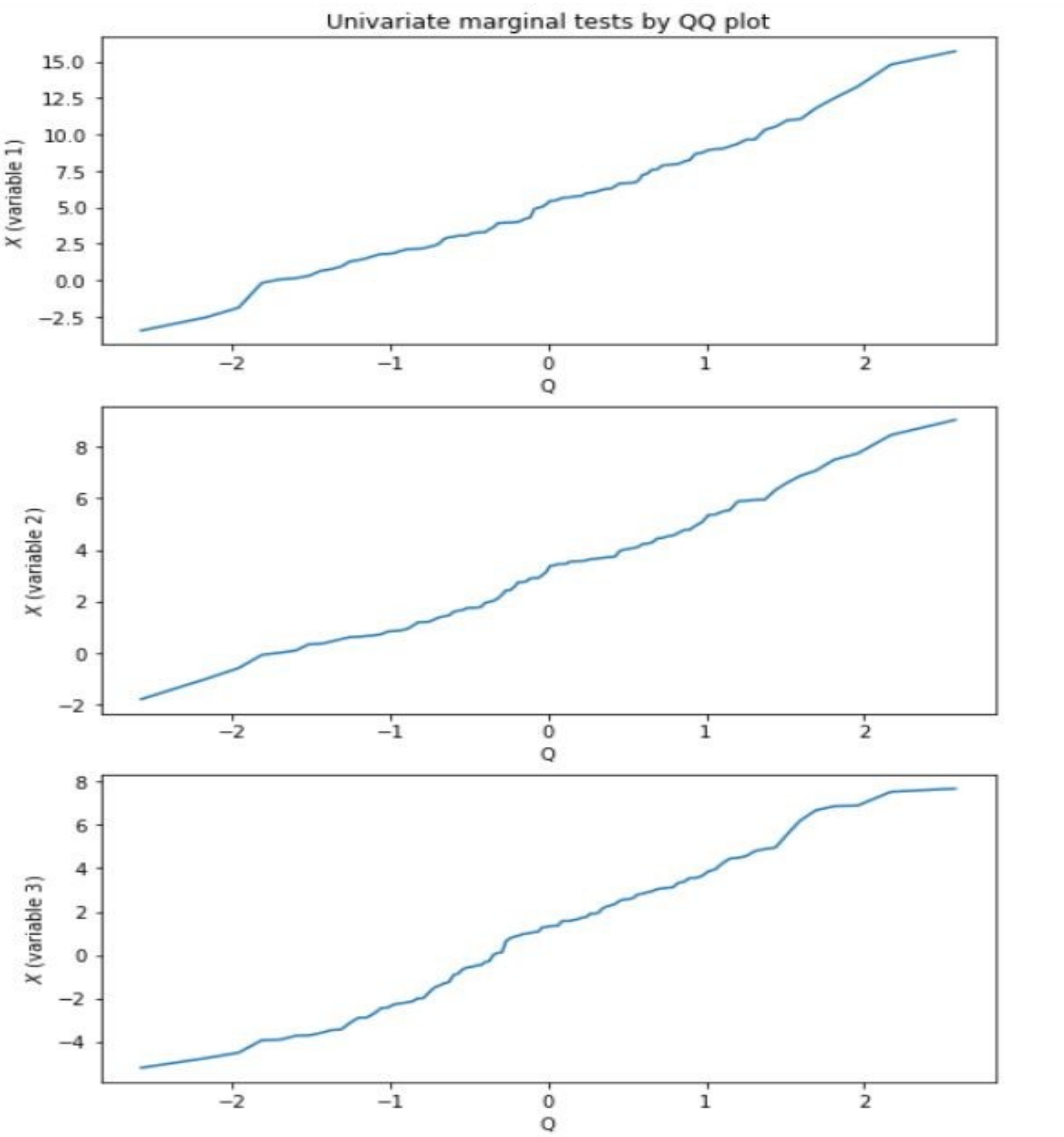
4. Multivariate distribution :

A multivariate normal distribution is a vector in multiple normally distributed variables, such that any linear combination of the variables is also normally distributed. It is mostly useful in extending the central limit theorem to multiple variables

Univariate Marginal Test:

The univariate normal distribution is given by :

$$p(x;\mu,\sigma^2)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$



In the univariate Marginal test by Quantile- Quantile plot, we observed that all three variables are approximately linear. By this, we can infer that all three variables are normally distributed.

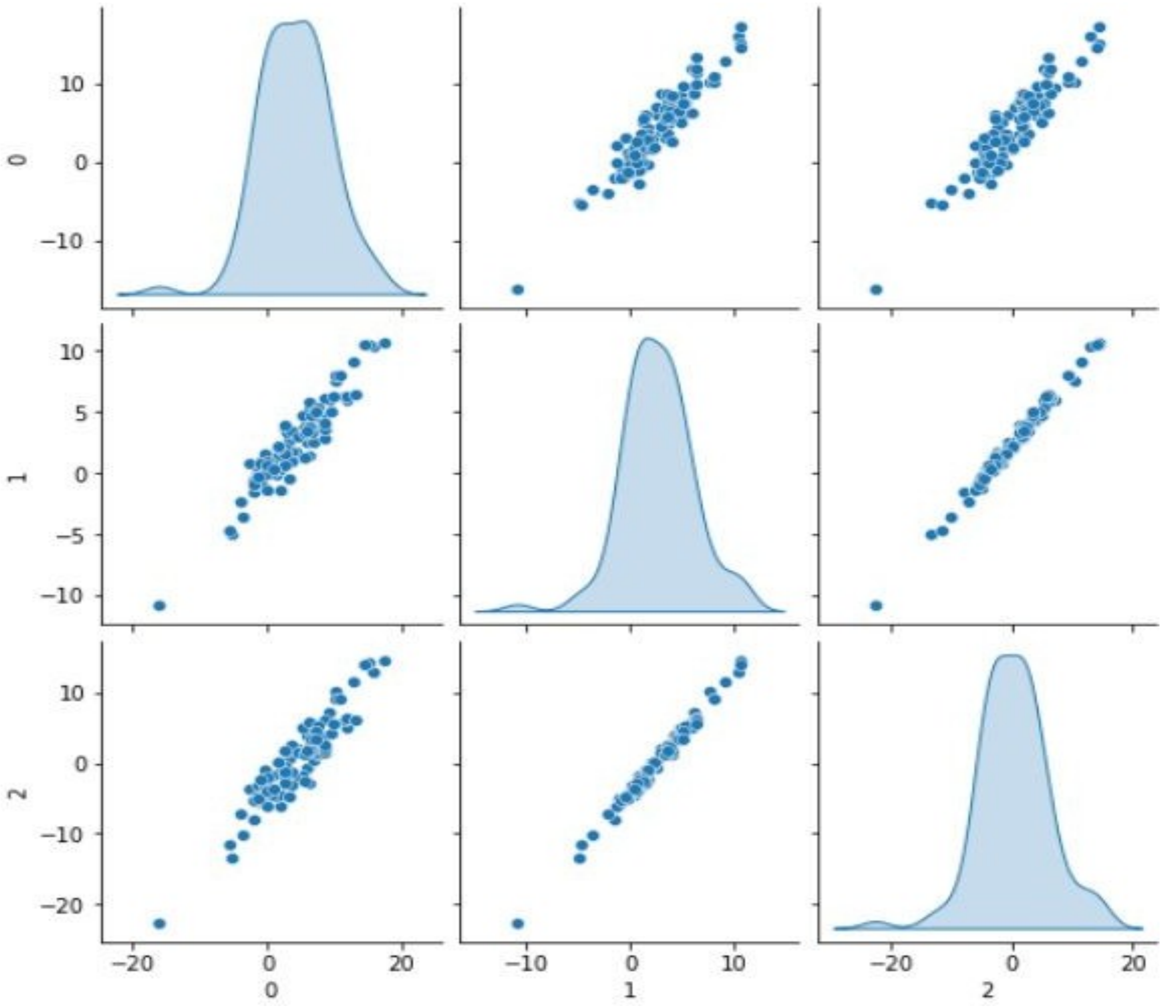
Bivariate Normality Test:

The Bivariate normal distribution is given by :

$$P(x_1, x_2) = \frac{1}{2\pi\sigma_1\,\sigma_2\sqrt{1-\rho^2}}\exp\left[-\frac{z}{2(1-\rho^2)}\right],$$

Where ρ = correlation(x1, x2), and

$$z \equiv \frac{(x_1-\mu_1)^2}{\sigma_1^2} - \frac{2\rho(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\,\sigma_2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2},$$

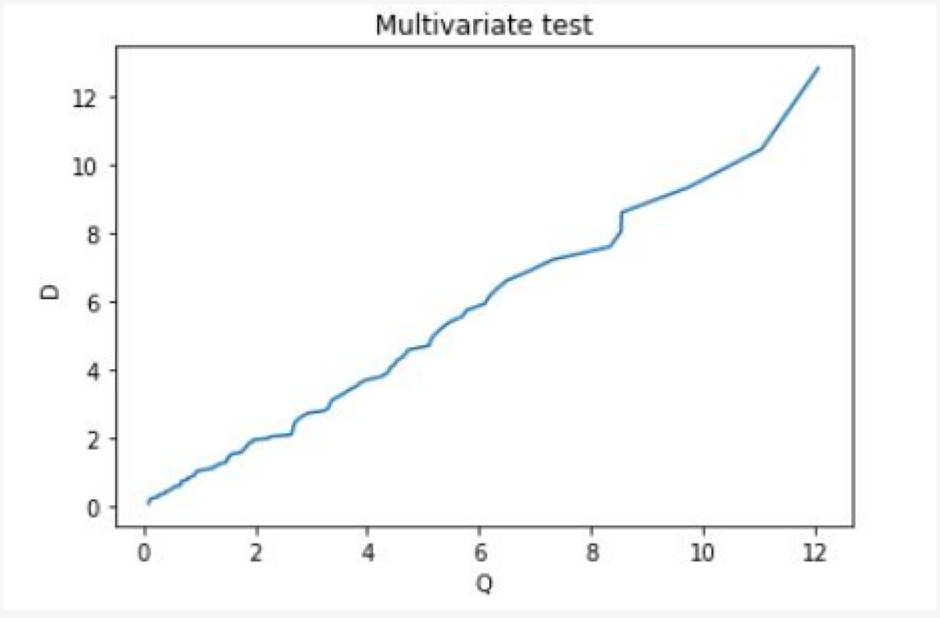


As we can see in the plot in the bivariate normality test, every combination of variables is linear except the diagonals. So it shows that these variables follow the multivariate distribution.

Multivariate Normality Test:

The multivariate normal distribution is given by :

$$p\,(\mathbf{x};\mu, \Sigma) = (1/[(2pi)^{n/2}\,|\Sigma|^{1/2}] * \exp(-\tfrac{1}{2}(x - \mu)^{T}\,\Sigma^{-1}\,(x - \mu))$$



Multivariate Normality test is satisfied because the graph is linear for Multivariates plot. Which shows that it is Normally distributed.