DS201/DSL253: Statistical Programming assignment 2 report

Shashank yadav 12342010

Introduction

This report is divided into 3 parts for 3 different questions.

Part 1:

we analyze two distinct processes that generate random numbers between 0 and 1. By uniform and exponential , random variable Y is defined as Y = FX(x), where FX(x) represents the cumulative distribution function (CDF) of X. This report explores the probability distribution function (PDF) of Y.

Part 2:

Analyzed an old forgotten text file to extract meaningful patterns from the content. With the top 30 most frequent words, we will use the Cumulative Distribution Function (CDF) to transform word frequencies to a new scale.

Part 3:

We analyzed the transformation of a random number U and drawn from a uniform distribution on the interval [0, 1], using the inverse cumulative distribution function of another random variable X to generate random variable Y and determine distribution/

Data

We have used the inbuilt fn and python libraries to generate random number of variable length from the distribution given in each question and tex_file.txt for question 2

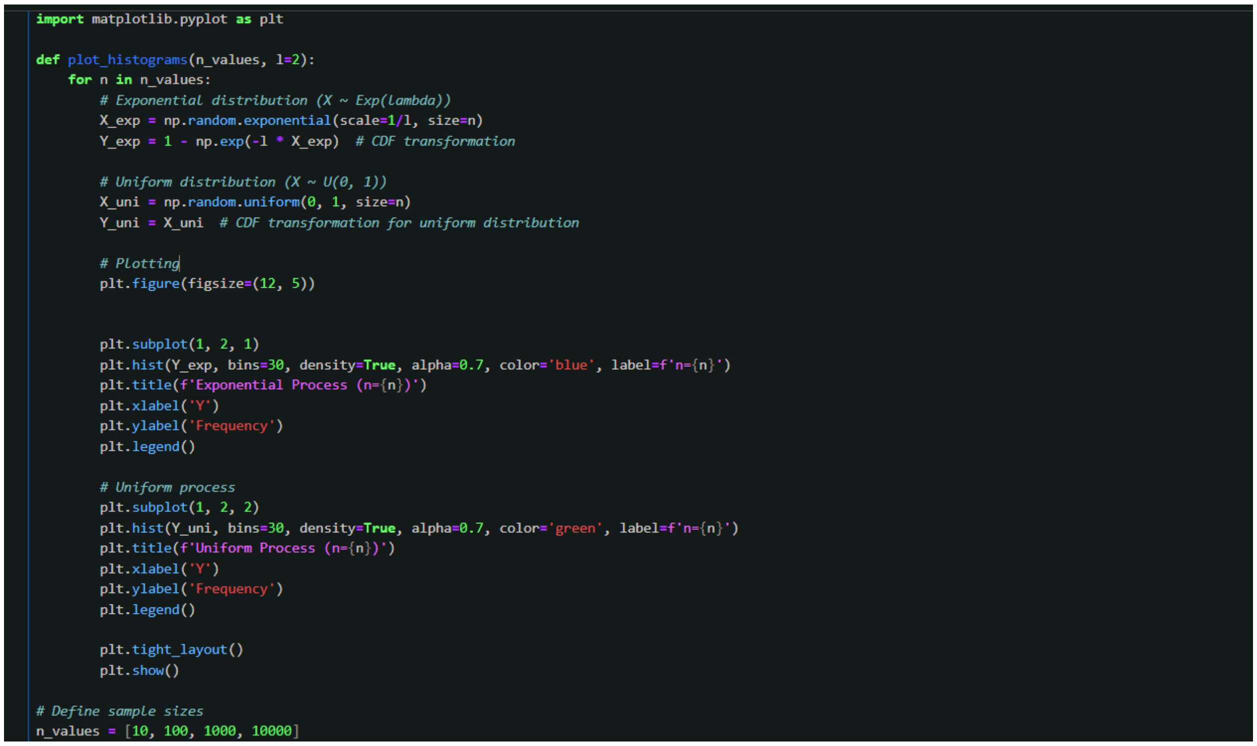
Methodology

Part 1: Exponential Distribution (λ = 2) •

- The PDF of X is given by $f_X(x) = \lambda e^{-\lambda x}, x \geq 0$. •

- The cdf distribution will be 1-e-λx
- For the uniform distribution pdf = 1 for x[0,1]
- Cdf will be safe for uniform

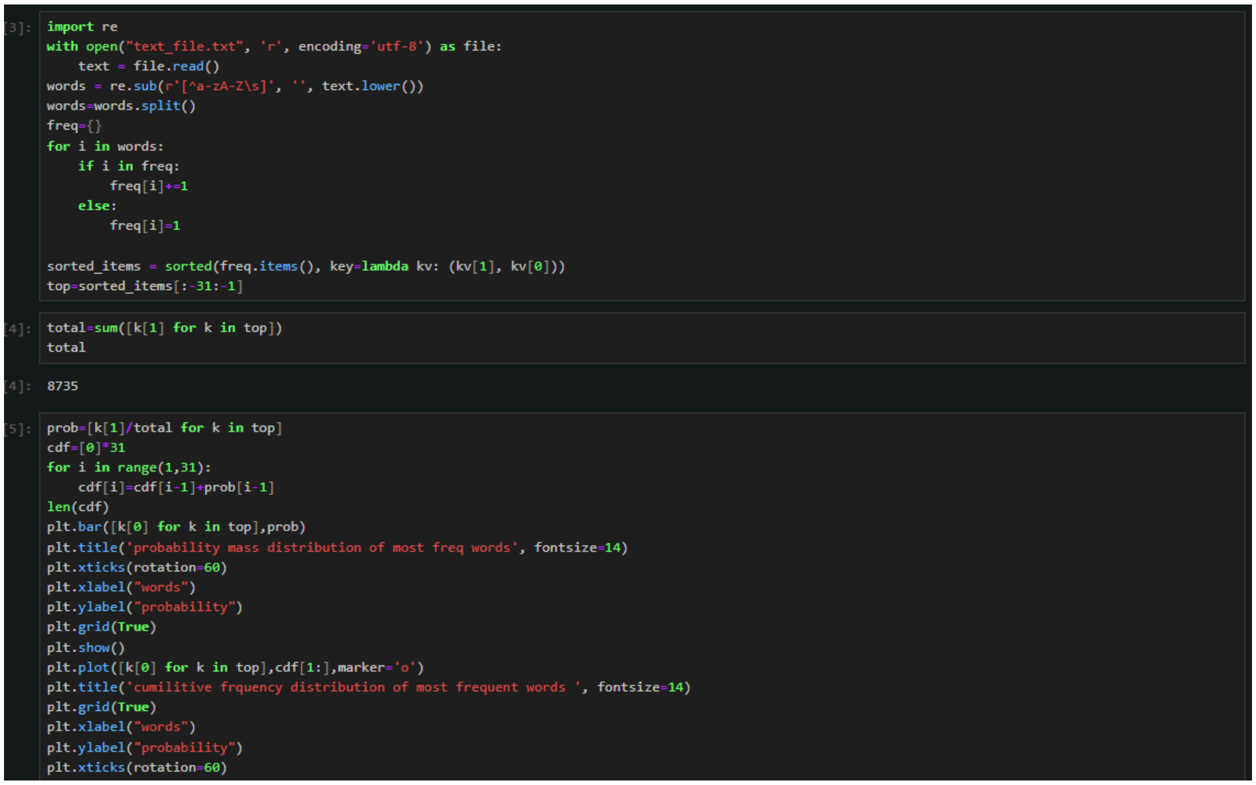
We have drawn n samples from each distribution transformed it using the cdf function then plot the histogram



```python
import matplotlib.pyplot as plt

def plot_histograms(n_values, l=2):
    for n in n_values:
        # Exponential distribution (X ~ Exp(lambda))
        X_exp = np.random.exponential(scale=1/l, size=n)
        Y_exp = 1 - np.exp(-l * X_exp)  # CDF transformation

        # Uniform distribution (X ~ U(0, 1))
        X_uni = np.random.uniform(0, 1, size=n)
        Y_uni = X_uni  # CDF transformation for uniform distribution

        # Plotting
        plt.figure(figsize=(12, 5))


        plt.subplot(1, 2, 1)
        plt.hist(Y_exp, bins=30, density=True, alpha=0.7, color='blue', label=f'n={n}')
        plt.title(f'Exponential Process (n={n})')
        plt.xlabel('Y')
        plt.ylabel('Frequency')
        plt.legend()

        # Uniform process
        plt.subplot(1, 2, 2)
        plt.hist(Y_uni, bins=30, density=True, alpha=0.7, color='green', label=f'n={n}')
        plt.title(f'Uniform Process (n={n})')
        plt.xlabel('Y')
        plt.ylabel('Frequency')
        plt.legend()

        plt.tight_layout()
        plt.show()

# Define sample sizes
n_values = [10, 100, 1000, 10000]
```

Part 2:

- Read the file and remove the unwanted characters and found the top 30 words and their freq by making a dictionary
- Finding the their cumulative freq by summing their probability in each iteration
- Plotting the pdf and cdf of the top 30 words



```
[3]: import re
     with open("text_file.txt", 'r', encoding='utf-8') as file:
         text = file.read()
     words = re.sub(r'[^a-zA-Z\s]', '', text.lower())
     words=words.split()
     freq={}
     for i in words:
         if i in freq:
             freq[i]+=1
         else:
             freq[i]=1

     sorted_items = sorted(freq.items(), key=lambda kv: (kv[1], kv[0]))
     top=sorted_items[:-31:-1]
```

```
[4]: total=sum([k[1] for k in top])
     total
```

```
[4]: 8735
```

```
[5]: prob=[k[1]/total for k in top]
     cdf=[0]*31
     for i in range(1,31):
         cdf[i]=cdf[i-1]+prob[i-1]
     len(cdf)
     plt.bar([k[0] for k in top],prob)
     plt.title('probability mass distribution of most freq words', fontsize=14)
     plt.xticks(rotation=60)
     plt.xlabel("words")
     plt.ylabel("probability")
     plt.grid(True)
     plt.show()
     plt.plot([k[0] for k in top],cdf[1:],marker='o')
     plt.title('cumilitive frquency distribution of most frequent words ', fontsize=14)
     plt.grid(True)
     plt.xlabel("words")
     plt.ylabel("probability")
     plt.xticks(rotation=60)
```

Part 3:

(a) Exponential Distribution

- **PDF:** λe−λx
- **CDF:**1-e-λx
- **Inverse CDF:**   -log(1 - x) /λ
- **Parameters:** λ = 2

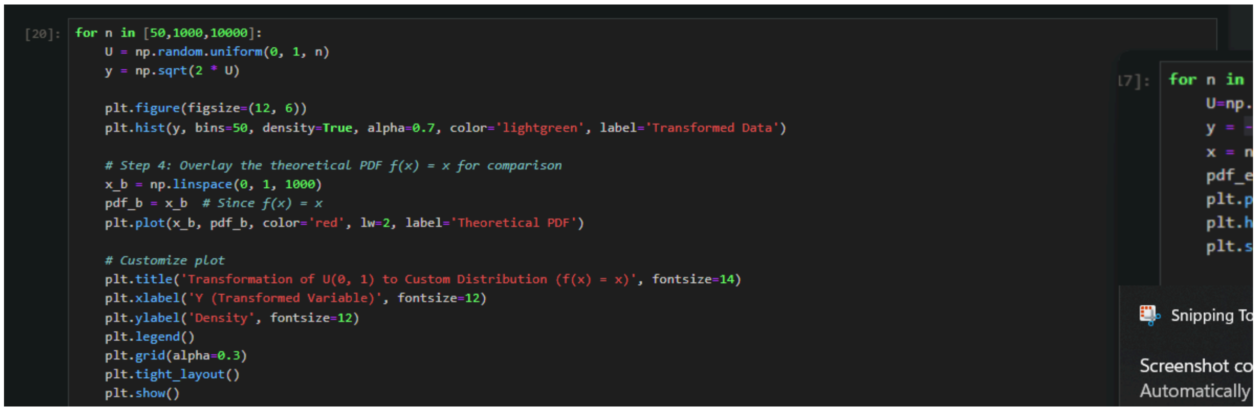
(b) Custom Distribution

- **PDF: x**
- **CDF: x2/2**
- **Inverse CDF: root(2x)**

Generate Uniform Data: Random samples are drawn from the uniform distribution.

Apply Inverse CDF**:** The inverse CDF of the desired distribution is applied to to create the transformed random variable.

Visualization: Histograms of the transformed data are plotted and compared to the theoretical PDF.

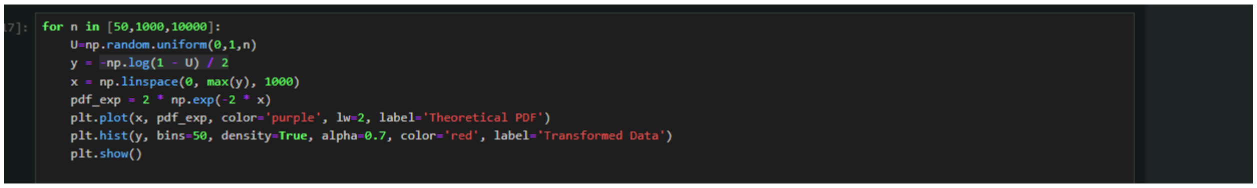


```
[20]: for n in [50,1000,10000]:
          U = np.random.uniform(0, 1, n)
          y = np.sqrt(2 * U)

          plt.figure(figsize=(12, 6))
          plt.hist(y, bins=50, density=True, alpha=0.7, color='lightgreen', label='Transformed Data')

          # Step 4: Overlay the theoretical PDF f(x) = x for comparison
          x_b = np.linspace(0, 1, 1000)
          pdf_b = x_b  # Since f(x) = x
          plt.plot(x_b, pdf_b, color='red', lw=2, label='Theoretical PDF')

          # Customize plot
          plt.title('Transformation of U(0, 1) to Custom Distribution (f(x) = x)', fontsize=14)
          plt.xlabel('Y (Transformed Variable)', fontsize=12)
          plt.ylabel('Density', fontsize=12)
          plt.legend()
          plt.grid(alpha=0.3)
          plt.tight_layout()
          plt.show()
```

```
17]: for n in
      U=np.
      y = -
      x = n
      pdf_e
      plt.p
      plt.h
      plt.s
```

Snipping To

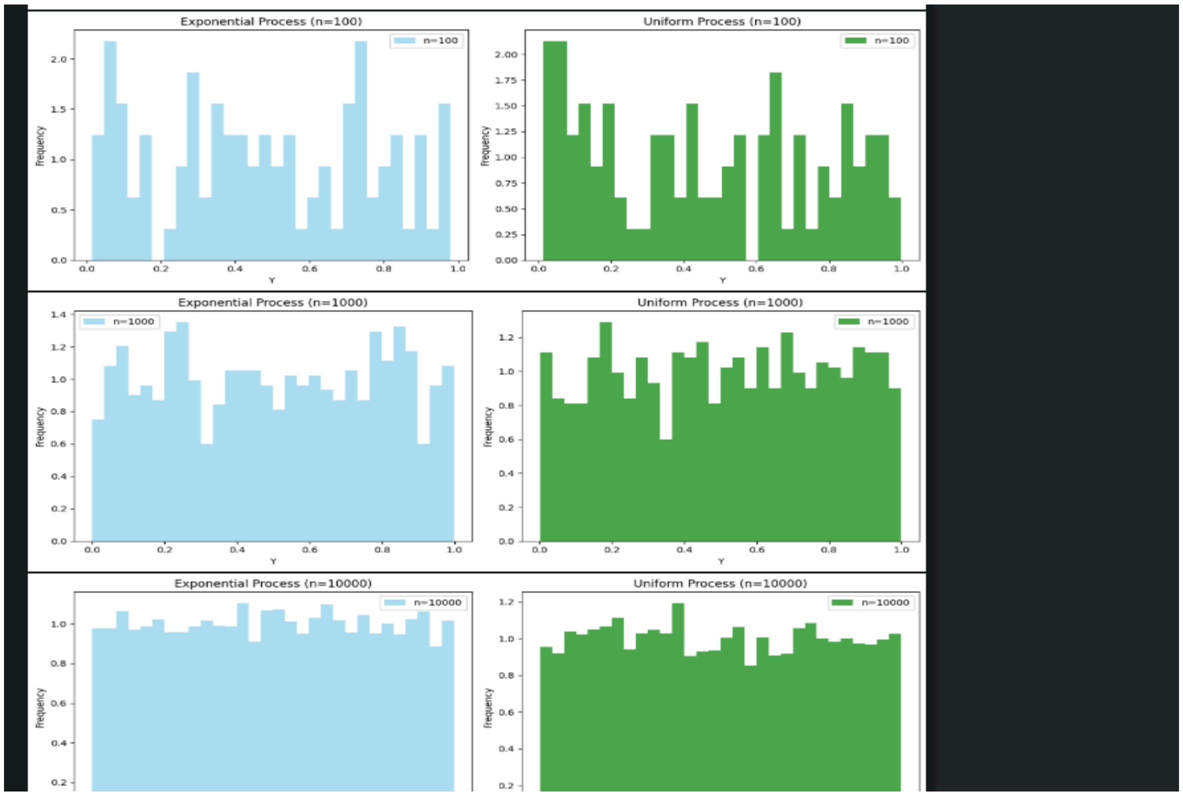
Screenshot co
Automatically



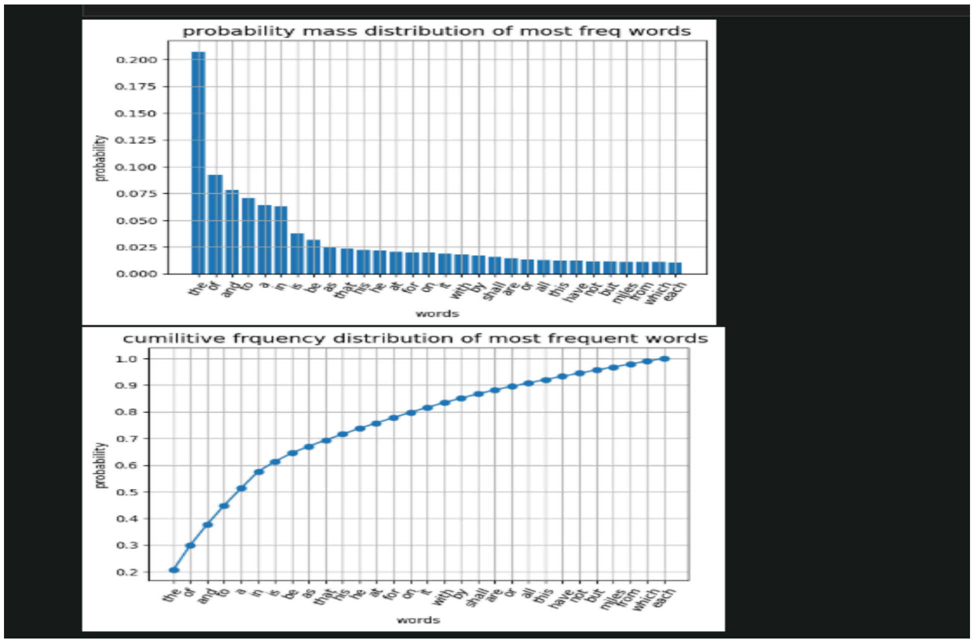
```
7]: for n in [50,1000,10000]:
        U=np.random.uniform(0,1,n)
        y = -np.log(1 - U) / 2
        x = np.linspace(0, max(y), 1000)
        pdf_exp = 2 * np.exp(-2 * x)
        plt.plot(x, pdf_exp, color='purple', lw=2, label='Theoretical PDF')
        plt.hist(y, bins=50, density=True, alpha=0.7, color='red', label='Transformed Data')
        plt.show()
```

4. Results

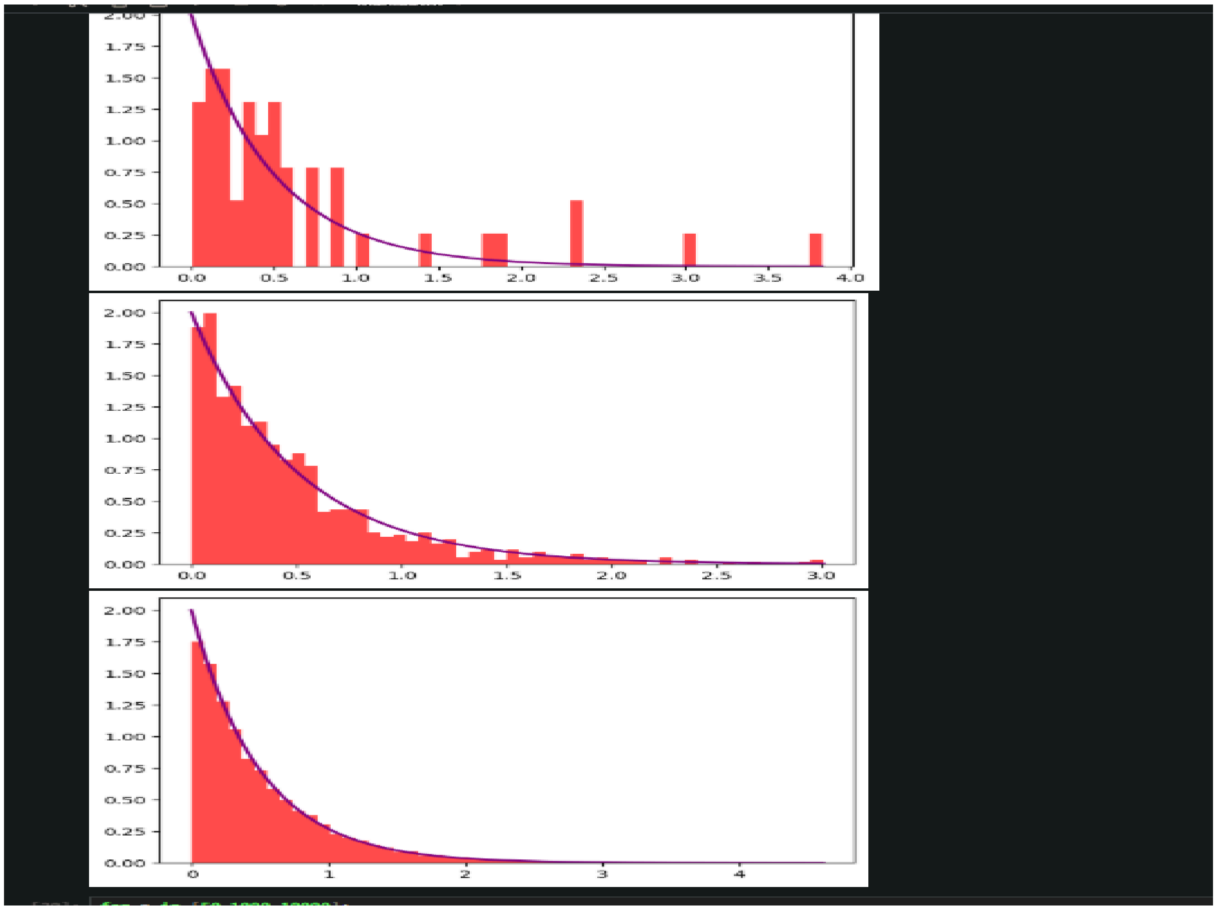
Part 1:



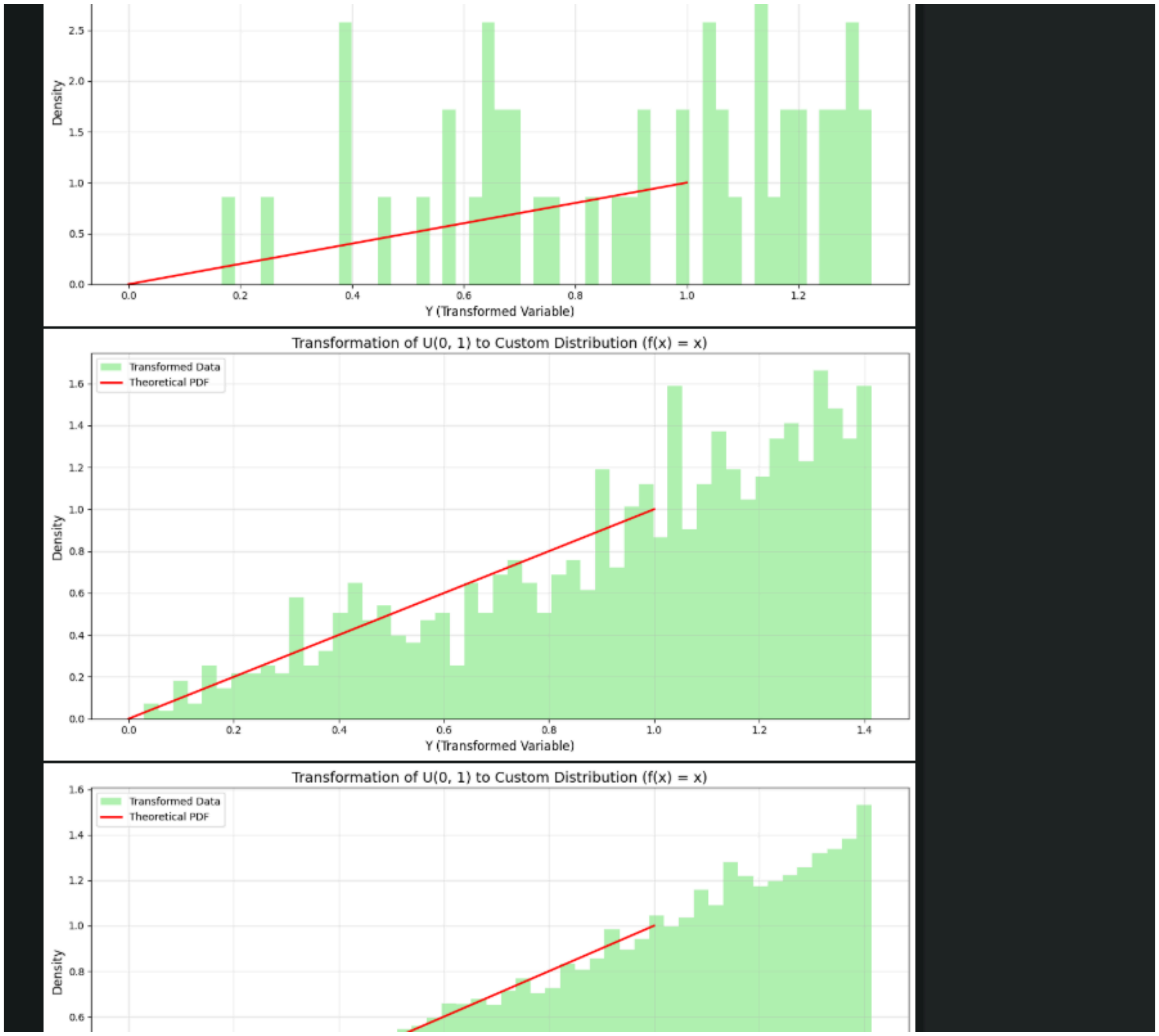
Part 2:


probability mass distribution of most freq words
probability
words
the
of
and
to
a
in
is
be
as
that
his
he
at
for
on
it
with
by
shall
are
or
all
this
have
not
but
miles
from
which
each
cumilitive frquency distribution of most frequent words
probability
words

Part 3



:



6. Observation/Conclusion

Part 1:

Exponential Process **:**

For small N, the histogram of nY appears noisy due to limited samples.

- As n increases, the histogram becomes smoother and converges to a uniform distribution, consistent with the theoretical PDF (fY(y)=1, 0≤y≤1.

Uniform Process:

- For all n, the histogram of y remains uniform as Y=XY = XY=X, and the distribution does not transform under the CDF.

Part 2 :

The histogram demonstrates that a few words, like "the," dominate the frequency distribution, representing the classic "long tail" seen in natural language usage.

CDF Transformation Insights:

- Using the CDF as a transformation reveals how quickly most word frequency is concentrated in the top few words. For example, over 50% of the total frequency is accounted for by just the top 6 words.
- After applying the CDF, the word frequency transformation can highlight rare words in the remaining dataset, which might provide insights into unique topics or contexts in the text.

Part 3:

The histogram closely matches the theoretical exponential PDF, confirming that the transformation via the inverse CDF worked correctly.

This analysis highlights the utility of the inverse CDF method in transforming uniformly distributed random variables into other distributions. The experiments with exponential and custom distributions demonstrate that:

- The method is reliable for generating random samples that conform to theoretical PDFs.
- The transformed variables provide insights into the behavior of the target distributions, which can be applied in simulation and modeling tasks.