EXP NO: 6	EXPLORATORY DATA ANALYSIS
DATE: 11/9/25	

Aim:

The dataset was successfully cleaned and analyzed; various distributions (histogram, CDF, PDF, KDE) were visualized, revealing income patterns, gender-based differences, and statistical properties.

Program:

Step 1: Mount Google Drive and Extract Dataset

```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)

import zipfile, os

 zip_path = "/content/drive/MyDrive/adult.zip"
extract_path = "/content/drive/MyDrive/adult_data"
os.makedirs(extract_path,
exist_ok=True)
```

Step 2: Import Required Libraries

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

import    seaborn    as    sns
sns.set(style="whitegrid") Step
```

3: Load the Adult Dataset

```
files = os.listdir(extract_path)
candidates = [f for f in files if f.lower().startswith("adult")] file_path
= os.path.join(extract_path, candidates[0])


df = pd.read_csv(file_path, header=None, sep=r',\s*', engine='python',
na_values=['?'])
df.columns = ['age', 'type_employer', 'fnlwgt', 'education', 'education_num',
              'marital', 'occupation', 'relationship', 'race', 'sex',
              'capital_gain', 'capital_loss', 'hr_per_week', 'country', 'income']
df.head()
```

Step 4: Clean and Standardize Data

```
df = df.applymap(lambda x: x.strip() if isinstance(x, str) else x) num_cols
=
['age','fnlwgt','education_num','capital_gain','capital_loss','hr_per_week']

for c in num_cols:     df[c] = pd.to_numeric(df[c], errors='coerce')
```

```
df['income'] = df['income'].str.replace(r'\s+', '', regex=True)
```

Step 5: Explore and Summarize Data

```
df.info()
print("\nNull    values:\n",    df.isnull().sum())    print("\nTop
countries:\n", df['country'].value_counts().head(10))
```

Step 6: Split Data by Gender

```
ml = df[df['sex'] == 'Male'].copy() fm
= df[df['sex'] == 'Female'].copy()
print("Males:",       ml.shape)
print("Females:", fm.shape)
```

Step 7: Analyze Income Distribution

```
df_high = df[df['income'] == '>50K'].copy()
def pct(part,  whole):        return  0  if whole == 0 else
round(100 * part / whole, 2)
ml_high = ml[ml['income'] ==
'>50K'] fm_high = fm[fm['income'] ==
'>50K']

print("High income  overall:", pct(len(df_high), len(df)), "%")
print("Men:",  pct(len(ml_high), len(ml)), "%") print("Women:",
pct(len(fm_high),len(fm)), "%")
```

Step 8: Compute Mean, Variance, and Std.Dev.

```
print("Averageage(men):",                          ml['age'].mean())
print("Averageage(women):",                        fm['age'].mean())
print("Variance(men):",                             ml['age'].var())
print("Variance(women):", fm['age'].var()) print("StdDev(men):",
ml['age'].std())
print("StdDev(women):", fm['age'].std())
```

Step 9: Find Median Age and Hours per Week

```
print("Median age (men):", ml['age'].median()) print("Median age
(women):", fm['age'].median()) print("Median hours/week (men):",
ml['hr_per_week'].median()) print("Median hours/week (women):",
fm['hr_per_week'].median())
```

Step 10: Plot Age Histograms

```
plt.figure(figsize=(8,4))
ml['age'].hist(edgecolor="red", bins=20)
plt.title("Histogram - Male Age")
plt.show()
plt.figure(figsize=(8,4))
fm['age'].hist(edgecolor="blue", bins=20) plt.title("Histogram
- Female Age") plt.show()
```

Step 11: Plot Overlapping Histograms

```
plt.figure(figsize=(8,5))
fm['age'].hist(alpha=.5, bins=20, label='Female')
ml['age'].hist(alpha=.5, bins=20, label='Male') plt.legend()
plt.title("Overlapping Age Histograms") plt.show()
```

Step 12: Plot CDF for Age

```
plt.figure(figsize=(8,4))
ml['age'].hist(density=True, histtype='step', cumulative=True, linewidth=2.5,
label='Male')
fm['age'].hist(density=True, histtype='step', cumulative=True, linewidth=2.5,
label='Female') plt.legend()
plt.title("CDF - Male vs Female Age") plt.show()
```

Step 13: Remove Outliers Based on Age

```
median_age = df['age'].median()
low_thresh = median_age - 15 high_thresh
= median_age + 35


drop_idx = df.index[(df['income'] == '>50K') & ((df['age'] < low_thresh) |
(df['age'] > high_thresh))]
df2 = df.drop(drop_idx).reset_index(drop=True)
 print("Original   shape:",   df.shape)
print("After   cleaning:",   df2.shape)
```

Step 14: Compare Before and After Cleaning

```
plt.figure(figsize=(13,5))             df.loc[df['income']             ==
'>50K','age'].plot(color='blue',  label='Before')  df2.loc[df2['income']  ==
'>50K','age'].plot(color='red',        label='After')         plt.legend()
plt.title("High-Income Ages: Before vs After Outlier Removal") plt.show()
```

2116231801132

Step 15: Compute and Plot Density Differences

```
countx, divx = np.histogram(ml_high['age'], bins=10, density=True)
county, divy = np.histogram(fm_high['age'], bins=10, density=True)
midpoints  =  [(divx[i]+divx[i+1])/2  for  i  in  range(len(divx)-1)]
plt.plot(midpoints,    countx-county,    'o-')    plt.title("Density
Difference (Male - Female)") plt.show()
```

Step 16: Calculate Skewness

```
def skewness(s): return ((s - s.mean())**3).sum() / (len(s)*s.std()**3)
print("Skewness (men):", skewness(ml['age']))
print("Skewness (women):", skewness(fm['age']))
```

Step 17: Plot Exponential and Gaussian Distributions

```
# Exponential x = np.arange(0,
10, 0.1) lam = 3 plt.plot(x,
lam*np.exp(-lam*x))
plt.title("Exponential PDF")
plt.show()


# Gaussian u, s = 6, 2 x
= np.arange(0, 15, 0.1)
y = (1/(np.sqrt(2*np.pi*s*s))) * np.exp(-((x-u)**2)/(2*s*s))
plt.plot(x, y) plt.title("Gaussian PDF") plt.show()
```

Step 18: Demonstrate Central Limit Theorem

```
fig, ax = plt.subplots(1, 4, figsize=(16,4))
x_plot = np.linspace(0,1,100) for i in
range(4):

    n = i + 1
    f = np.mean(np.random.random((10000, n)), axis=1)
m, s = np.mean(f), np.std(f, ddof=1)
    fn = (1/(s*np.sqrt(2*np.pi))) * np.exp(-(x_plot-m)**2/(2*s**2))
ax[i].hist(f, bins=40, density=True, alpha=0.6)
ax[i].plot(x_plot, fn, 'r-', linewidth=2) plt.suptitle("Central
Limit Theorem Demonstration") plt.show()
```

Step 19: Kernel Density Estimation

```
from scipy.stats import norm, gaussian_kde
x1 = np.random.normal(-1, 2, 15) x2 =
np.random.normal(6, 3, 10) y = np.r_[x1,
x2]
x_grid = np.linspace(min(y), max(y), 200)


# Manual KDE
plt.plot(x_grid, [norm.pdf(x_grid, yi, 0.4) for yi in y], 'k:', alpha=0.3)
plt.plot(x_grid, norm.pdf(x_grid, 0, 3), 'r-') plt.title("Manual Kernel
Density Estimation") plt.show()
# Scipy KDE density =
gaussian_kde(y)
plt.hist(y, bins=20, density=True, alpha=0.5)
plt.plot(x_grid, density(x_grid), 'r-')
plt.title("Gaussian KDE (scipy)") plt.show()
```

Step 20: Plot Bimodal Distribution

```
x1 = np.random.normal(-1, 0.5, 15)
x2 = np.random.normal(6, 1.0, 10) x
= np.r_[x1, x2]
density = gaussian_kde(x)
xgrid = np.linspace(x.min(), x.max(), 200)


plt.hist(x, bins=18, density=True, alpha=0.6)
plt.plot(xgrid, density(xgrid), 'r-')
plt.title("Bimodal Distribution with KDE") plt.show()
```

Output:

```
Extracted files: ['Index', 'adult.data', 'adult.names', 'adult.test', 'old.adult.names']
Using file: /content/drive/MyDrive/adult_data/adult.data
```

	age	type_employer	fnlwgt	education	education_num	marital	occupation	relationship	race	sex	capital_gain	capital_loss	hr_per_week	country	income
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

```
/tmp/ipython-input-2670513830.py:2: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map instead.
  df = df.applymap(lambda x: x.strip() if isinstance(x, str) else x)
Shape: (32561, 15)
Income unique values: ['<=50K' '>50K']
Sex unique values: ['Male' 'Female']
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   type_employer   30725 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education_num   32561 non-null  int64
 5   marital         32561 non-null  object
 6   occupation      30718 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital_gain    32561 non-null  int64
 11  capital_loss    32561 non-null  int64
 12  hr_per_week     32561 non-null  int64
 13  country         31978 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
None

Null counts per column:
 age                0
type_employer    1836
fnlwgt              0
education           0
education_num       0
marital             0
occupation       1843
relationship        0
race                0
sex                 0
capital_gain        0
capital_loss        0
hr_per_week         0
country           583
income              0
dtype: int64
```

	age	type_employer	fnlwgt	education	education_num	marital	occupation	relationship	race	sex	capital_gain	capital_loss	hr_per_week	country	income
8444	39	State-gov	114055	Bachelors	13	Never-married	Exec-managerial	Not-in-family	White	Female	0	0	40	United-States	<=50K
27565	40	Self-emp-inc	475322	Bachelors	13	Separated	Craft-repair	Own-child	White	Male	0	0	50	United-States	<=50K
14299	36	Private	186376	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Asian-Pac-Islander	Male	0	0	50	United-States	>50K
10946	29	Private	135296	Assoc-voc	11	Never-married	Adm-clerical	Own-child	White	Female	0	0	40	United-States	<=50K
1855	25	Local-gov	190057	Bachelors	13	Never-married	Prof-specialty	Own-child	White	Female	0	0	40	United-States	<=50K

```
Top countries:
 country
United-States         29170
Mexico                  643
Philippines             198
Germany                 137
Canada                  121
Puerto-Rico             114
El-Salvador             106
India                   100
Cuba                     95
England                  90
Jamaica                  81
South                    80
China                    75
Italy                    73
Dominican-Republic       70
Vietnam                  67
Guatemala                64
Japan                    62
Poland                   60
Columbia                 59
Name: count, dtype: int64

Age counts (first 20):
 age
17    395
18    550
19    712
20    753
21    720
22    765
23    877
24    798
25    841
26    785
27    835
28    867
29    813
30    861
31    888
32    828
33    875
34    886
35    876
36    898
dtype: int64
```

```
Males: (21790, 15)
Females: (10771, 15)
```

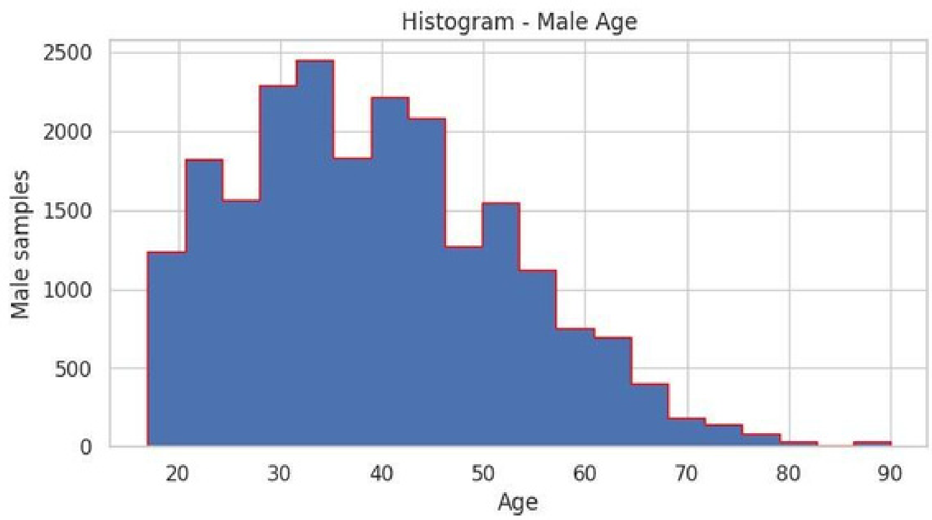
```
Income unique (re-check): ['<=50K' '>50K']
High income overall: (7841, 15)
Rate of high income people: 24.08 %
Rate of high income men: 30.57 %
Rate of high income women: 10.95 %
```

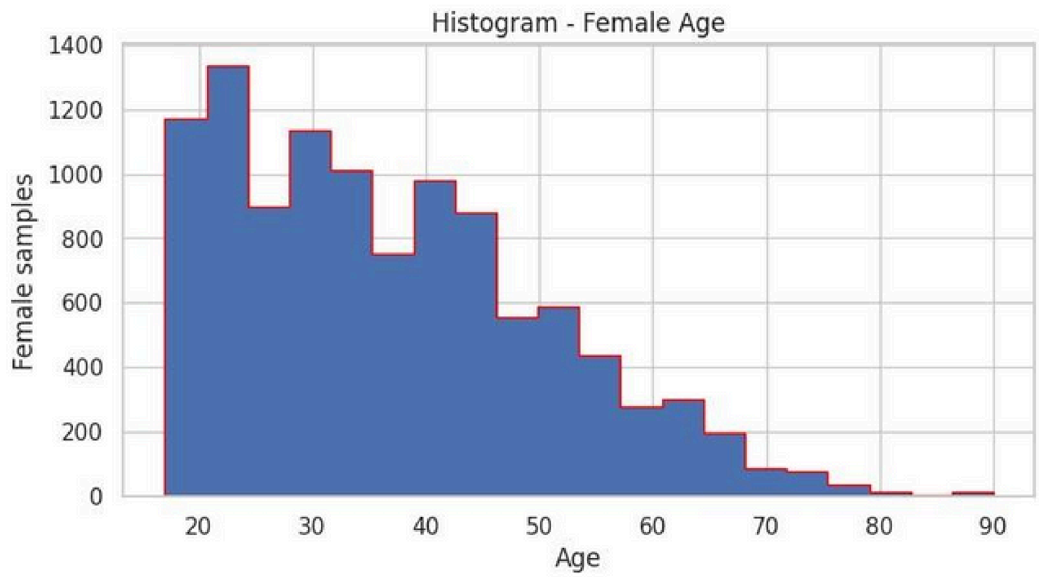
```
Average age (men): 39.43354749885268
Average age (women): 36.85823043357163
Average age (high-income men): 44.62578805163614
Average age (high-income women): 42.125530110262936

Age variance (men): 178.77375174530096  std: 13.37063019252649
Age variance (women): 196.3837063948037  std: 14.01369709943824

Hours/week mean (men): 42.42808627810923
Hours/week mean (women): 36.410361154953115
Hours/week std (men): 12.119755243874367
Hours/week std (women): 11.81129954748725
```

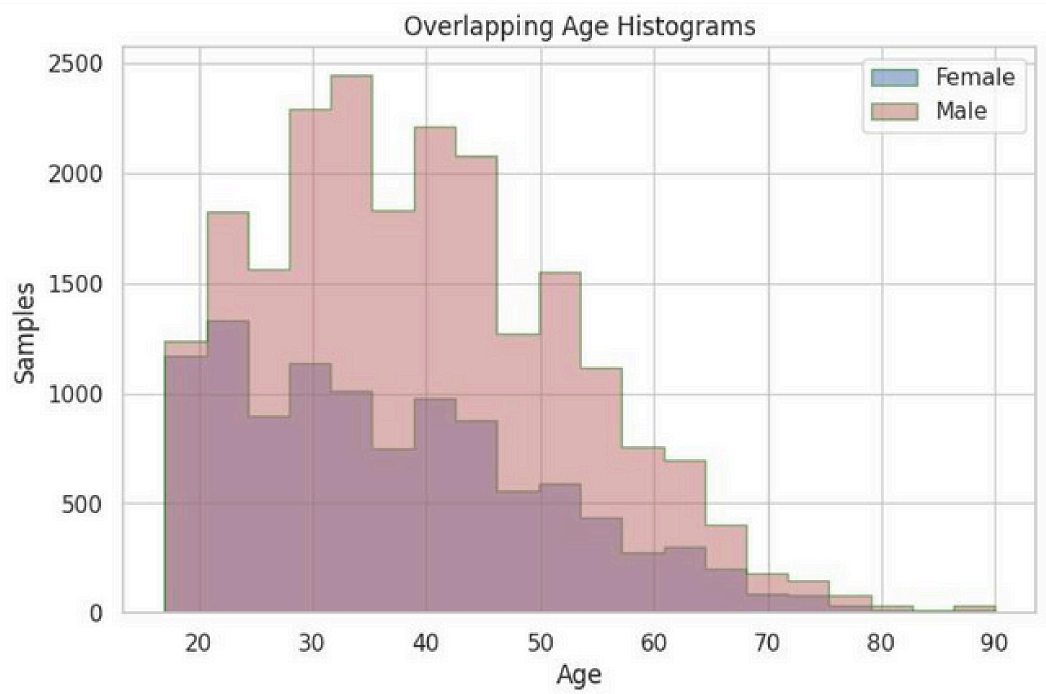
```
Average age men: 39.43354749885268
Average age women: 36.85823043357163
Average age high-income men: 44.62578805163614
Average age high-income women: 42.125530110262936
Variance men: 178.77375174530096
Variance women: 196.3837063948037
Std Dev men: 13.37063019252649
Std Dev women: 14.01369709943824
```

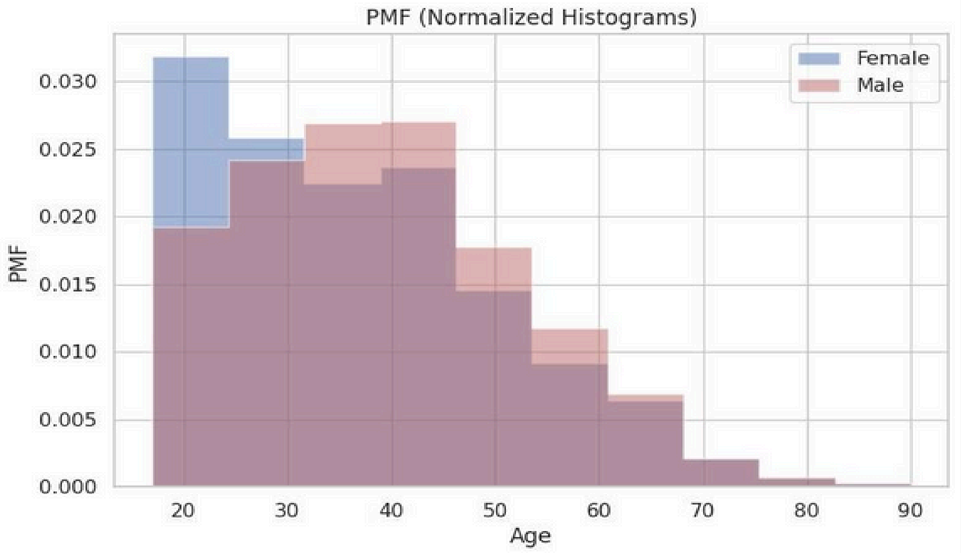


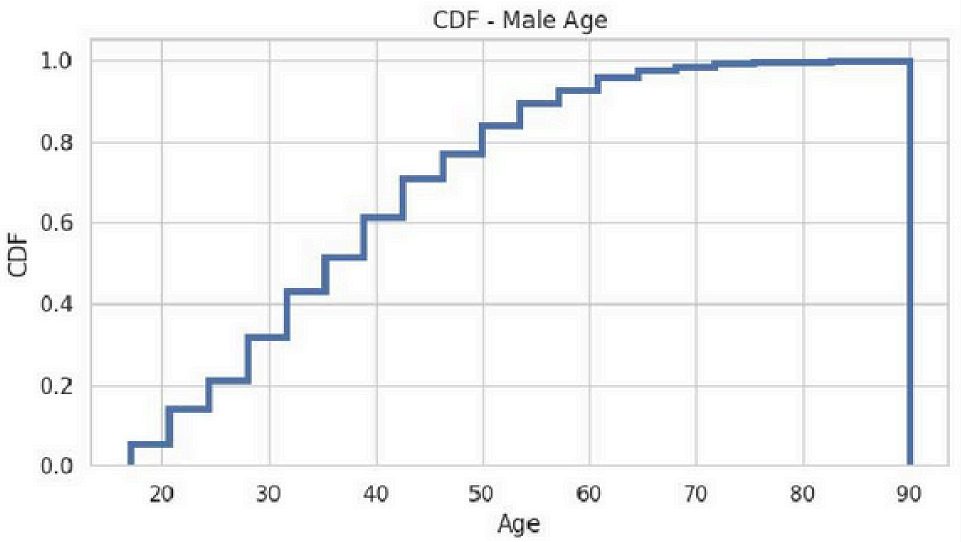
```
Median age (men): 38.0
Median age (women): 35.0
Median age (high-income men): 44.0
Median age (high-income women): 41.0
Median hours/week (men): 40.0
Median hours/week (women): 40.0
```

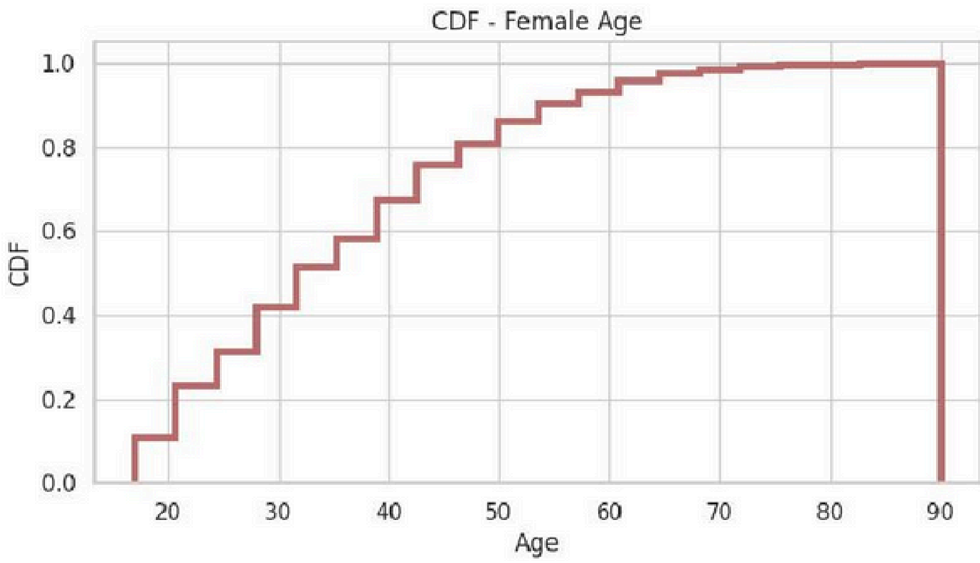


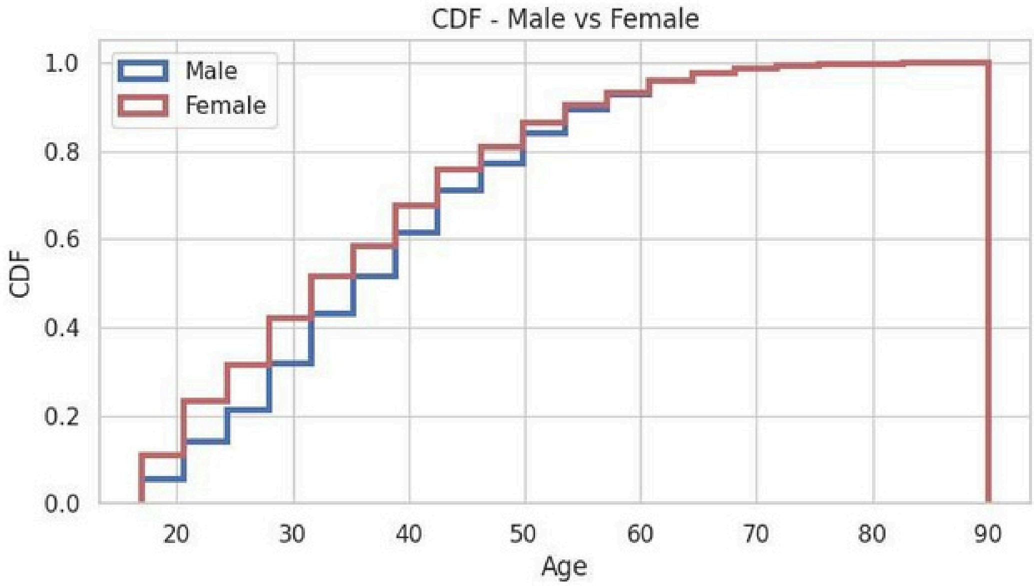

Histogram - Female Age
Female samples
0
200
400
600
800
1000
1200
1400
20
30
40
50
60
70
80
90
Age


Overlapping Age Histograms
Female
Male
Samples
0
500
1000
1500
2000
2500
20
30
40
50
60
70
80
90
Age


PMF (Normalized Histograms)
Female
Male
PMF
0.030
0.025
0.020
0.015
0.010
0.005
0.000
20
30
40
50
60
70
80
90
Age


CDF - Male Age
CDF
1.0
0.8
0.6
0.4
0.2
0.0
20
30
40
50
60
70
80
90
Age


CDF - Female Age
CDF
1.0
0.8
0.6
0.4
0.2
0.0
20
30
40
50
60
70
80
90
Age



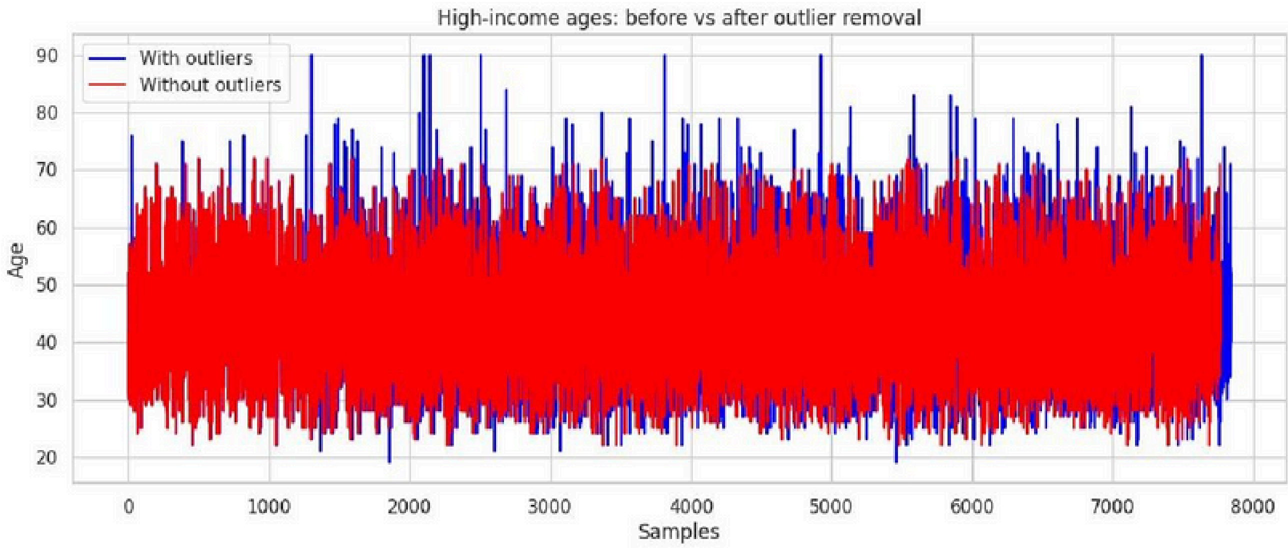

```
Mean age difference (Male - Female): 2.57532
```

```
Overall median age: 37.0
High-income & age < median-15: 5
High-income & age > median+35: 69
```

```
Original shape: (32561, 15)
After dropping outliers shape: (32487, 15)
```

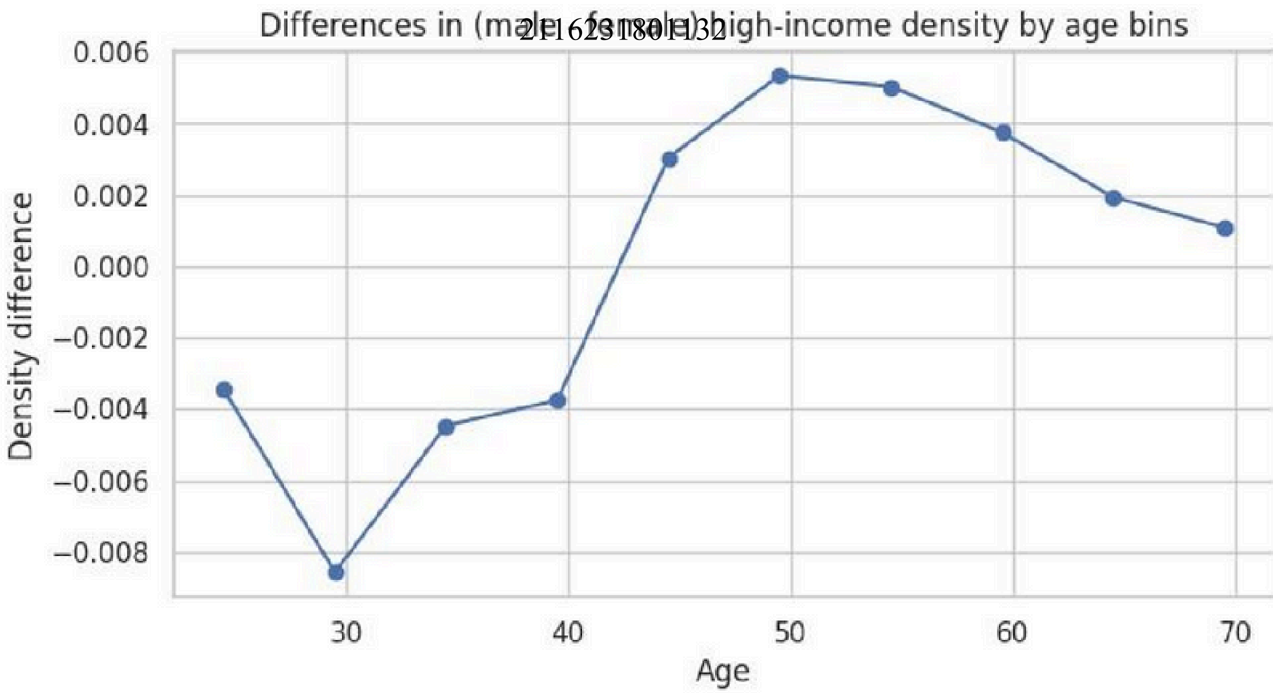
```
ml2_age shape: (6599,)
fm2_age shape: (1168,)
Men stats after drop - Mean: 44.325352326110014 Std: 10.012302742491952 Median: 44.0 Min: 22 Max: 72
Women stats after drop - Mean: 41.93236301369863 Std: 9.989525648849213 Median: 41.0 Min: 22 Max: 72
```

```
Originally:  39.433547 36.85823 2.575317
High-income:  44.625788 42.12553 2.500258
After cleaning:  44.325352 41.932363 2.392989

Medians originally:  38.0 35.0 3.0
High-income medians:  44.0 41.0 3.0
After cleaning medians:  44.0 41.0 3.0
```

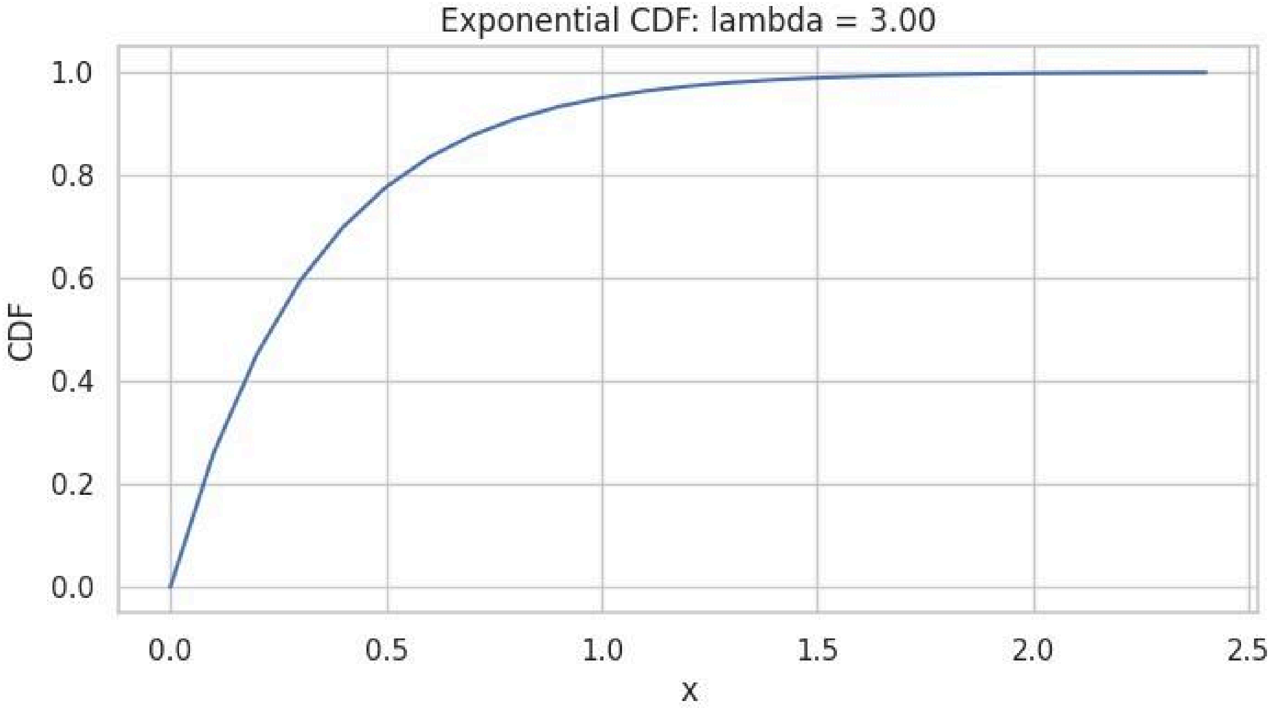



```
Bin midpoints: [np.float64(24.5), np.float64(29.5), np.float64(34.5), np.float64(39.5), np.float64(44.5), np.float64(49.5), np.float64(54.5), np.float64(59.5), np.float64(64.5), np.float64(69.5)]
Male density bins: [0.00390968 0.01642673 0.02773147 0.03545992 0.03661161 0.03206546
 0.02182149 0.01557812 0.00691014 0.00348538]
Female density bins: [0.00736301 0.025      0.03219178 0.03921233 0.03356164 0.02671233
 0.01678082 0.01181507 0.00496575 0.00239726]
```

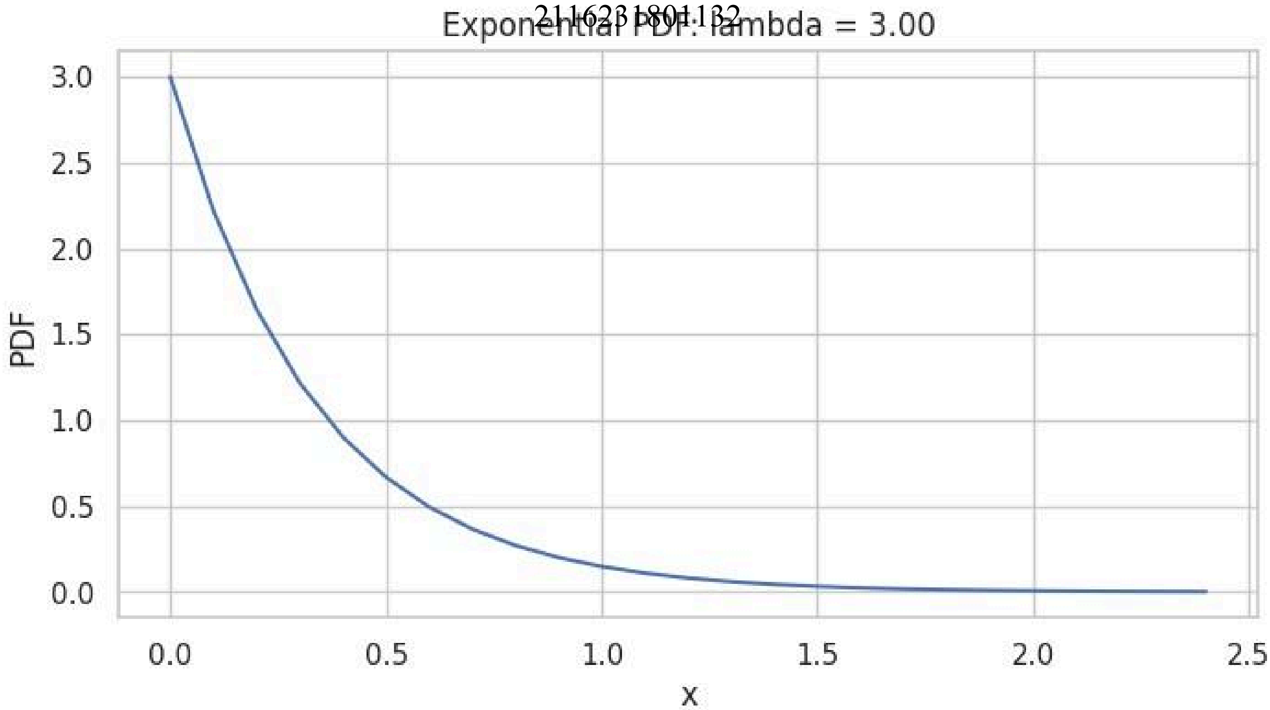
2116231801132

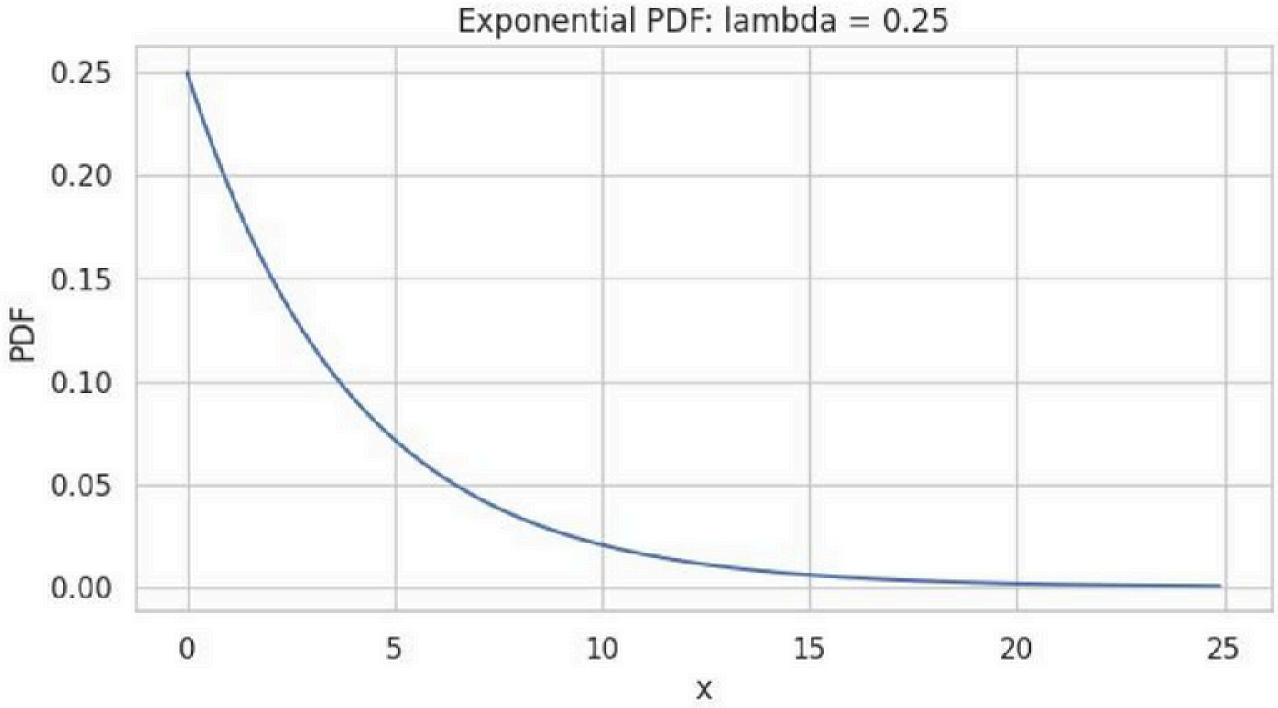



```
Skewness (ml2_age): 0.26927674749980657
Skewness (fm2_age): 0.4021179824911571
Pearson (ml2_age): 0.09748576360837374
Pearson (fm2_age): 0.28000218823384304
```

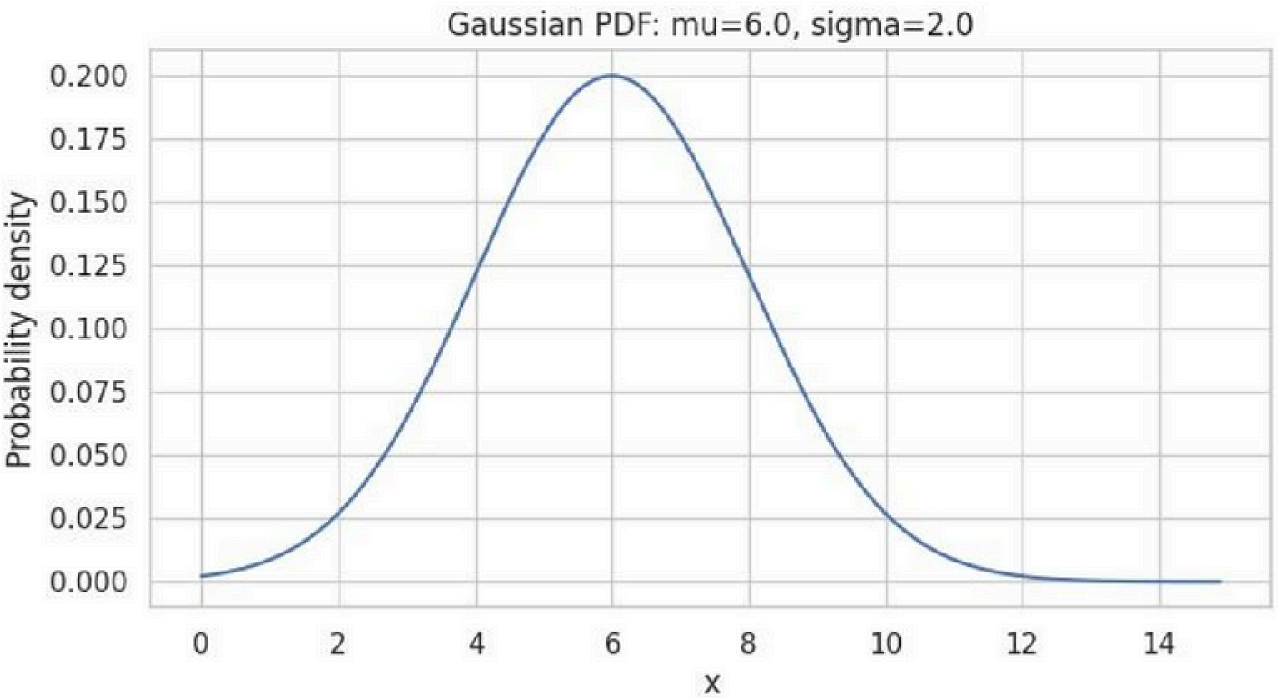


2116231801132

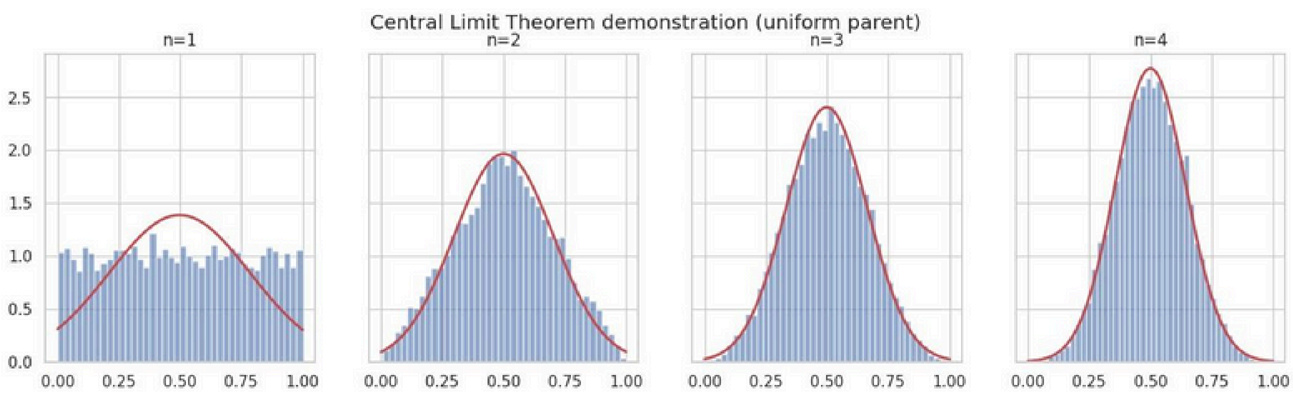




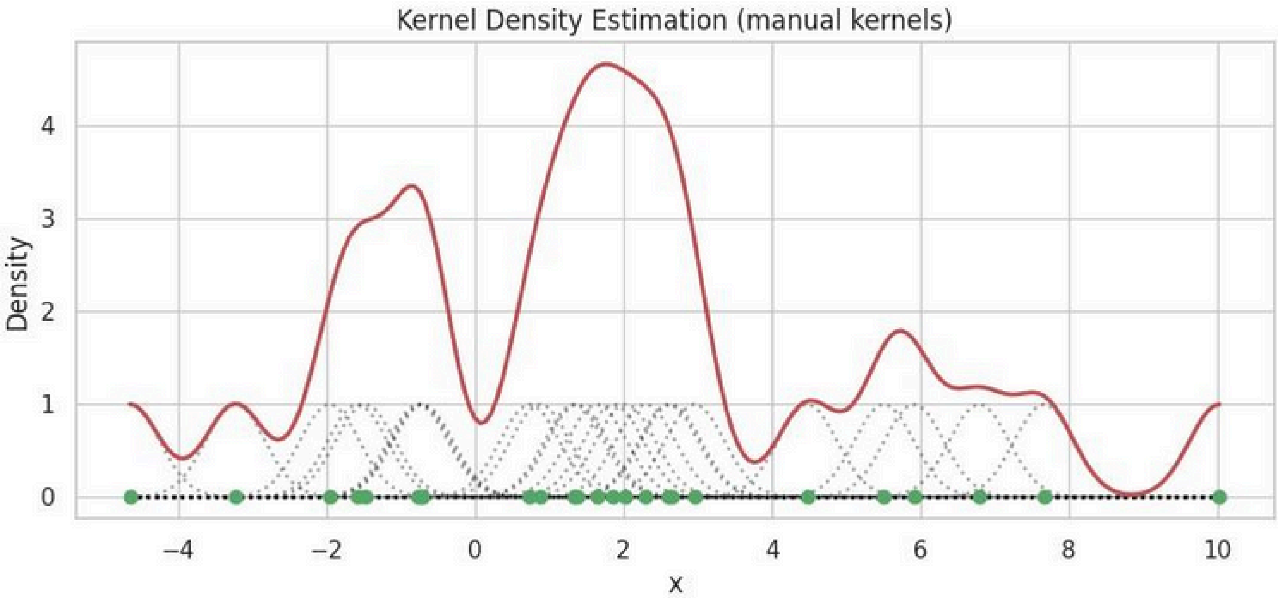
Exponential PDF: lambda = 0.25
PDF
x

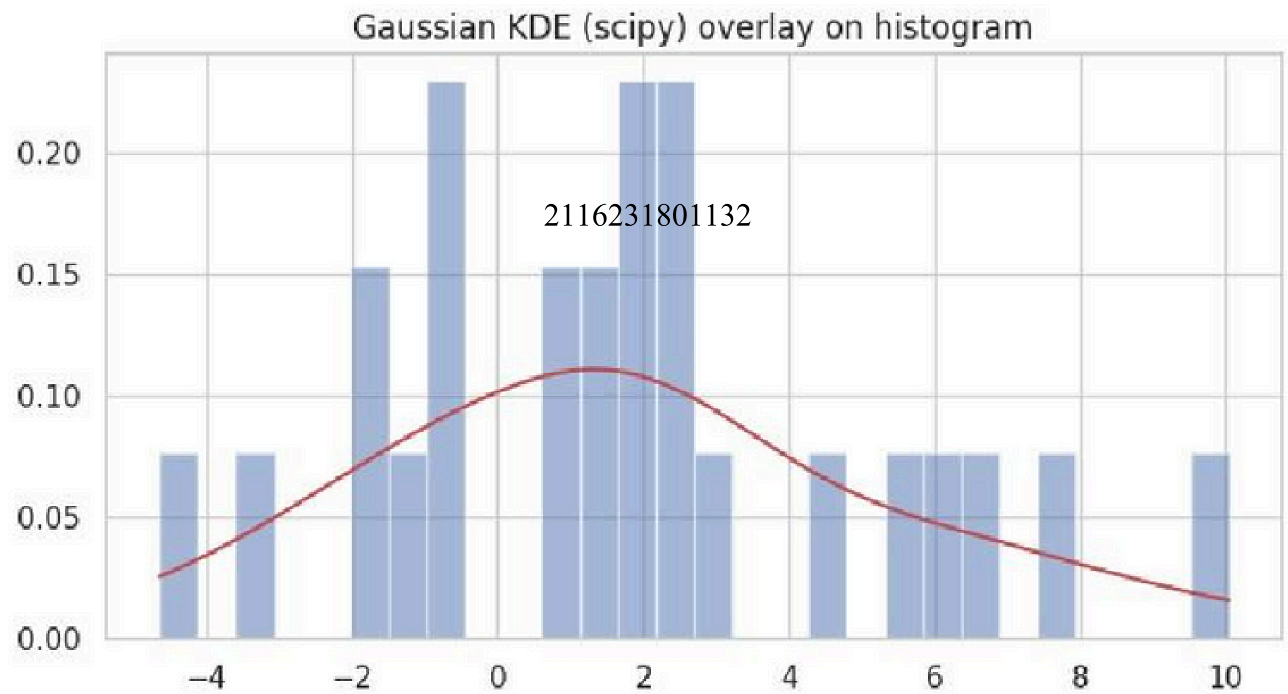


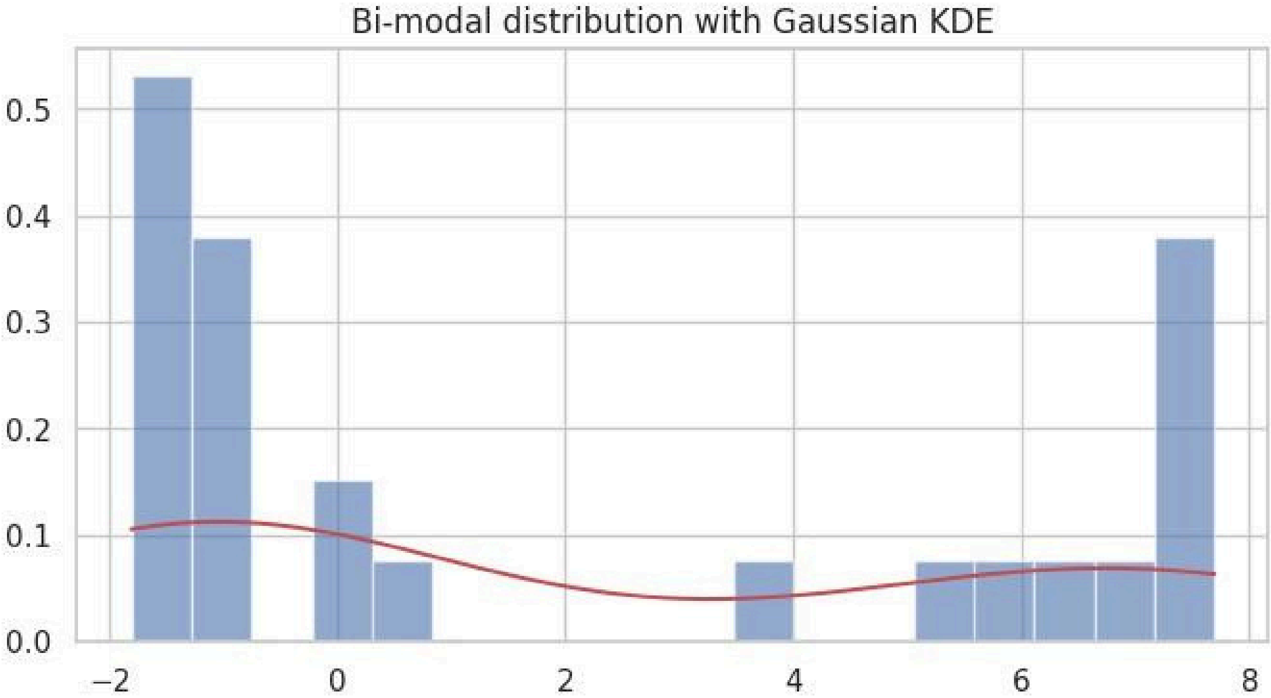
Gaussian PDF: mu=6.0, sigma=2.0
Probability density
x



Central Limit Theorem demonstration (uniform parent)
n=1
n=2
n=3
n=4


Kernel Density Estimation (manual kernels)
Density
x
0
1
2
3
4
−4
−2
0
2
4
6
8
10


Gaussian KDE (scipy) overlay on histogram
0.00
0.05
0.10
0.15
0.20
−4
−2
0
2
4
6
8
10



Result:

The dataset was successfully cleaned and analysed; various distributions (histogram, CDF, PDF, KDE) were visualized, revealing income patterns, gender-based differences, and statistical properties.