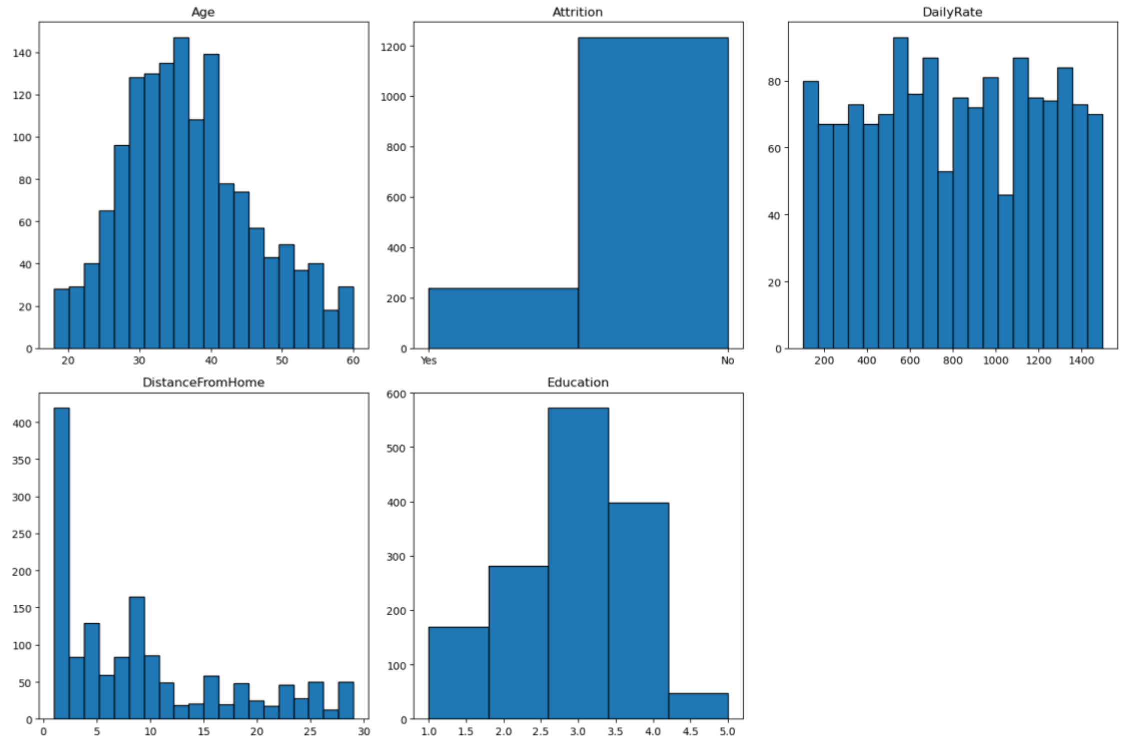
```
# Kelsey Barton
# DSC530-T304
# Final Assignment
# March 2, 2024

# The Dataset: WA_Fn-UseC_-HR-Employee-Attrition

# The Question: Can certain characteristics about a person predict whether
they will choose to remain employed within an organization or leave the
organization?

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load the CSV file into a DataFrame
df = pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")

# Display the first few rows of the DataFrame
print(df.head())

   Age Attrition     BusinessTravel  DailyRate              Department  \
0   41       Yes      Travel_Rarely       1102                   Sales
1   49        No  Travel_Frequently        279  Research & Development
2   37       Yes      Travel_Rarely       1373  Research & Development
3   33        No  Travel_Frequently       1392  Research & Development
4   27        No      Travel_Rarely        591  Research & Development

   DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber
\
0                 1          2  Life Sciences              1               1
1                 8          1  Life Sciences              1               2
2                 2          2          Other              1               4
3                 3          4  Life Sciences              1               5
4                 2          1        Medical              1               7

   ...  RelationshipSatisfaction StandardHours  StockOptionLevel  \
0  ...                         1            80                 0
1  ...                         4            80                 1
2  ...                         2            80                 0
3  ...                         3            80                 0
4  ...                         4            80                 1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany
\
0                  8                      0               1               6
1                 10                      3               3              10
2                  7                      3               3               0
3                  8                      3               3               8
```

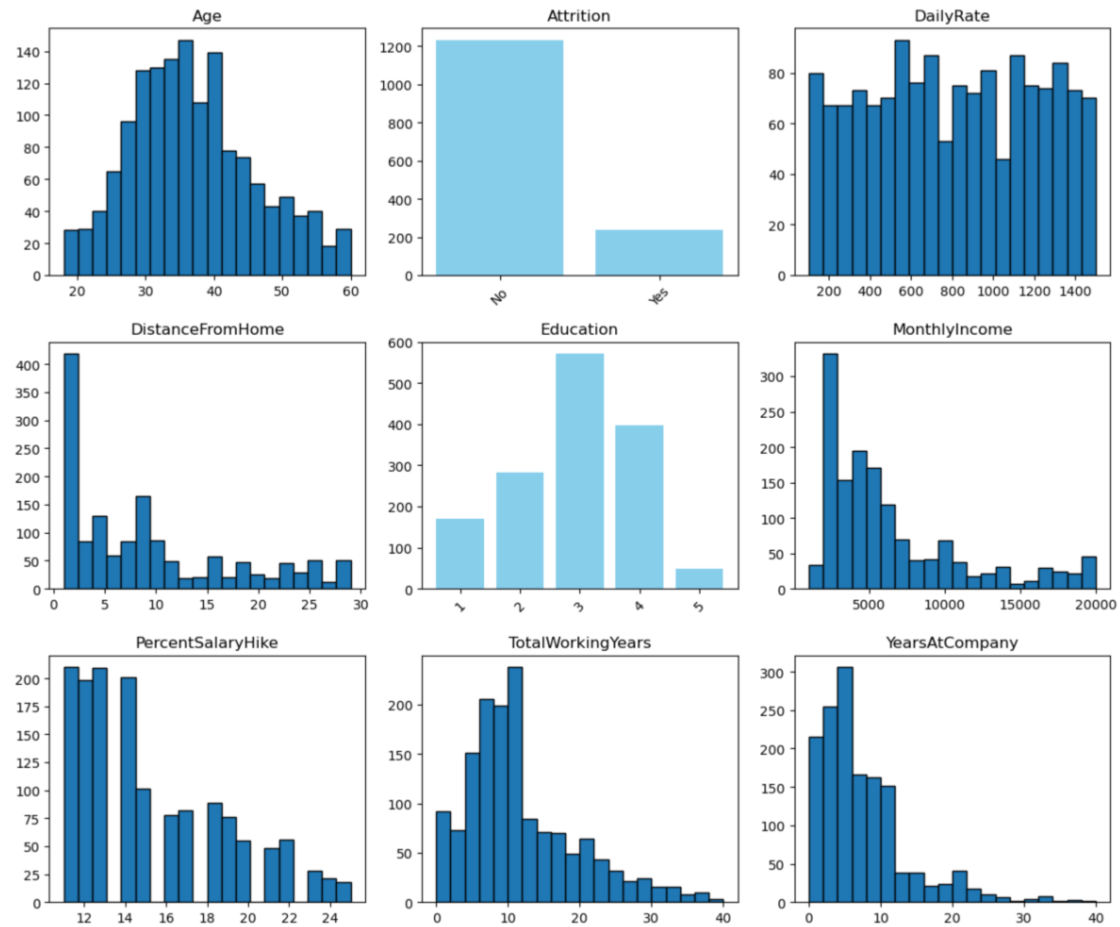
```
4                  6                      3                3                2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]

# Describe what the 5 variables mean in the dataset.

# This dataset includes many different variables. The most prevalent
variables for the experiments question are: age, attrition, rate of pay,
travel distance to work, and education. Age represents the age of the
employee. Attrition indicates whether the employee has left the company or
not. DailyRate, or rate of pay, denotes the daily rate of pay for the
employee. DistanceFromHome, or distance to work showcases the distance from
the employee's home to the workplace. Education provides the level of
education attained by the employee. These are the five main variables being
used in the provided code for analysis and visualization.

# Include a histogram of each of the 5 variables.

# Histograms
plt.figure(figsize=(15, 10))

# Age
plt.subplot(2, 3, 1)
plt.hist(df['Age'], bins=20, edgecolor='black')
plt.title('Age')

# Attrition
plt.subplot(2, 3, 2)
plt.hist(df['Attrition'], bins=2, edgecolor='black')
plt.title('Attrition')

# DailyRate
plt.subplot(2, 3, 3)
plt.hist(df['DailyRate'], bins=20, edgecolor='black')
plt.title('DailyRate')

# DistanceFromHome
plt.subplot(2, 3, 4)
plt.hist(df['DistanceFromHome'], bins=20, edgecolor='black')
plt.title('DistanceFromHome')

# Education
plt.subplot(2, 3, 5)
```

```
plt.hist(df['Education'], bins=5, edgecolor='black')
plt.title('Education')

plt.tight_layout()
plt.show()

plt.figure(figsize=(12, 10))

plt.subplot(3, 3, 1)
plt.hist(df['Age'], bins=20, edgecolor='black')
plt.title('Age')

plt.subplot(3, 3, 2)
attrition_counts = df['Attrition'].value_counts()
plt.bar(attrition_counts.index, attrition_counts.values, color='skyblue')
plt.title('Attrition')
plt.xticks(rotation=45)

plt.subplot(3, 3, 3)
plt.hist(df['DailyRate'], bins=20, edgecolor='black')
plt.title('DailyRate')

plt.subplot(3, 3, 4)
plt.hist(df['DistanceFromHome'], bins=20, edgecolor='black')
plt.title('DistanceFromHome')

plt.subplot(3, 3, 5)
education_counts = df['Education'].value_counts()
plt.bar(education_counts.index, education_counts.values, color='skyblue')
plt.title('Education')
plt.xticks(rotation=45)

plt.subplot(3, 3, 6)
plt.hist(df['MonthlyIncome'], bins=20, edgecolor='black')
plt.title('MonthlyIncome')

plt.subplot(3, 3, 7)
plt.hist(df['PercentSalaryHike'], bins=20, edgecolor='black')
plt.title('PercentSalaryHike')

plt.subplot(3, 3, 8)
plt.hist(df['TotalWorkingYears'], bins=20, edgecolor='black')
plt.title('TotalWorkingYears')

plt.subplot(3, 3, 9)
plt.hist(df['YearsAtCompany'], bins=20, edgecolor='black')
plt.title('YearsAtCompany')
```

```
plt.tight_layout()
plt.show()
```



png



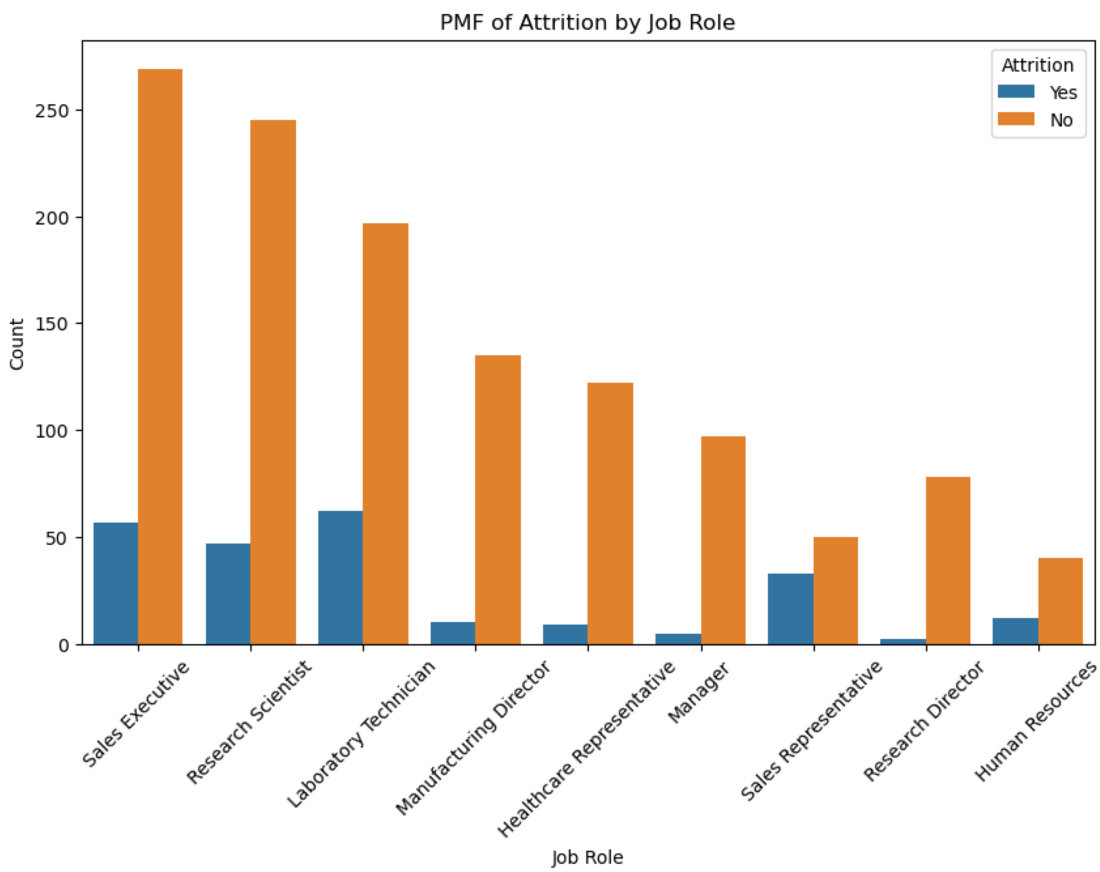
png

```
# Identify any outliers and explain the reasoning for them being outliers and how you believe they should be handled.

# The outliers for age are any data points that fall far from the bulk of the data, such as ages below 20 or above 60. Outliers in age might represent valid data points (e.g., experienced employees) or data entry errors.
# The outliers for daily rate are any extremely high or low daily rates that deviate significantly from the majority of the data. Similar to age, outliers in daily rate could be due to genuine differences in salaries or data entry errors. If they are legitimate data points, they should be kept in the analysis. Otherwise, they might be removed or adjusted after investigating the reasons behind them.
# The distance from home outliers are large distances from home that are not representative of the majority of employees. These outliers could indicate remote employees, employees who have relocated, or data entry errors. If they are valid data points, they should be retained; otherwise, they might be adjusted or removed.
# The education outliers consist of unusual education levels compared to the majority of employees. These outliers might indicate employees with advanced degrees or employees with less formal education than the majority. They might
```

```
be retained, grouped differently, or treated separately depending on how the
relate to the question.

# Include the other descriptive characteristics about the variables: mean,
mode, spread, and tails.

# Descriptive statistics
print("Descriptive Statistics:")
print(df.describe())

descriptive_stats = {
    'Variable': ['Age', 'Attrition', 'DailyRate', 'DistanceFromHome',
'Education'],
    'Mean': [df['Age'].mean(), None, df['DailyRate'].mean(),
df['DistanceFromHome'].mean(), df['Education'].mean()],
    'Mode': [df['Age'].mode().values[0], df['Attrition'].mode().values[0],
df['DailyRate'].mode().values[0], df['DistanceFromHome'].mode().values[0],
df['Education'].mode().values[0]],
    'Spread (Standard Deviation)': [df['Age'].std(), None,
df['DailyRate'].std(), df['DistanceFromHome'].std(), df['Education'].std()],
    'Tails': ['Two-tailed', None, 'Right-tailed', 'Right-tailed', None]
}

# Convert the dictionary to a DataFrame
descriptive_df = pd.DataFrame(descriptive_stats)

# Print descriptive statistics
print(descriptive_df)

Descriptive Statistics:
               Age    DailyRate  DistanceFromHome    Education  EmployeeCount
\
count  1470.000000  1470.000000       1470.000000  1470.000000         1470.0
mean     36.923810   802.485714          9.192517     2.912925            1.0
std       9.135373   403.509100          8.106864     1.024165            0.0
min      18.000000   102.000000          1.000000     1.000000            1.0
25%      30.000000   465.000000          2.000000     2.000000            1.0
50%      36.000000   802.000000          7.000000     3.000000            1.0
75%      43.000000  1157.000000         14.000000     4.000000            1.0
max      60.000000  1499.000000         29.000000     5.000000            1.0

       EmployeeNumber  EnvironmentSatisfaction   HourlyRate  JobInvolvement
\
count     1470.000000              1470.000000  1470.000000     1470.000000
mean      1024.865306                 2.721769    65.891156        2.729932
std        602.024335                 1.093082    20.329428        0.711561
min          1.000000                 1.000000    30.000000        1.000000
25%        491.250000                 2.000000    48.000000        2.000000
50%       1020.500000                 3.000000    66.000000        3.000000
75%       1555.750000                 4.000000    83.750000        3.000000
```

```
max       2068.000000                 4.000000   100.000000        4.000000

          JobLevel  ...  RelationshipSatisfaction  StandardHours  \
count  1470.000000  ...               1470.000000         1470.0
mean      2.063946  ...                  2.712245           80.0
std       1.106940  ...                  1.081209            0.0
min       1.000000  ...                  1.000000           80.0
25%       1.000000  ...                  2.000000           80.0
50%       2.000000  ...                  3.000000           80.0
75%       3.000000  ...                  4.000000           80.0
max       5.000000  ...                  4.000000           80.0

       StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
count       1470.000000        1470.000000            1470.000000
mean           0.793878          11.279592               2.799320
std            0.852077           7.780782               1.289271
min            0.000000           0.000000               0.000000
25%            0.000000           6.000000               2.000000
50%            1.000000          10.000000               3.000000
75%            1.000000          15.000000               3.000000
max            3.000000          40.000000               6.000000

       WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
count      1470.000000     1470.000000         1470.000000
mean          2.761224        7.008163            4.229252
std           0.706476        6.126525            3.623137
min           1.000000        0.000000            0.000000
25%           2.000000        3.000000            2.000000
50%           3.000000        5.000000            3.000000
75%           3.000000        9.000000            7.000000
max           4.000000       40.000000           18.000000

       YearsSinceLastPromotion  YearsWithCurrManager
count              1470.000000           1470.000000
mean                  2.187755              4.123129
std                   3.222430              3.568136
min                   0.000000              0.000000
25%                   0.000000              2.000000
50%                   1.000000              3.000000
75%                   3.000000              7.000000
max                  15.000000             17.000000

[8 rows x 26 columns]
           Variable        Mean Mode  Spread (Standard Deviation)  \
0               Age   36.923810   35                     9.135373
1         Attrition         NaN   No                          NaN
2         DailyRate  802.485714  691                   403.509100
3  DistanceFromHome    9.192517    2                     8.106864
4         Education    2.912925    3                     1.024165
```

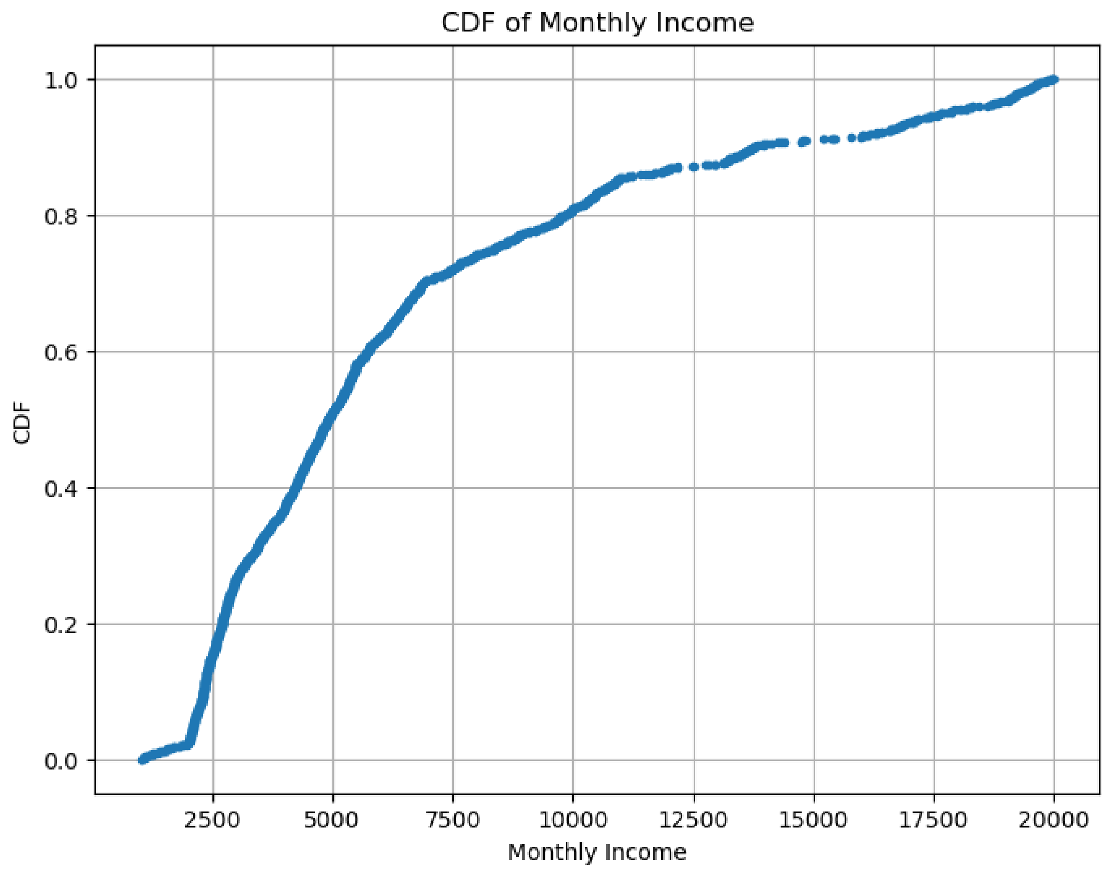
```
           Tails
0    Two-tailed
1          None
2  Right-tailed
3  Right-tailed
4          None

# Compare two scenarios in your data using a PMF.

# PMF comparing attrition for different job roles
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='JobRole', hue='Attrition')
plt.title('PMF of Attrition by Job Role')
plt.xticks(rotation=45)
plt.xlabel('Job Role')
plt.ylabel('Count')
plt.legend(title='Attrition')
plt.show()
```



png

```
# Create 1 CDF with one of your variables.
```

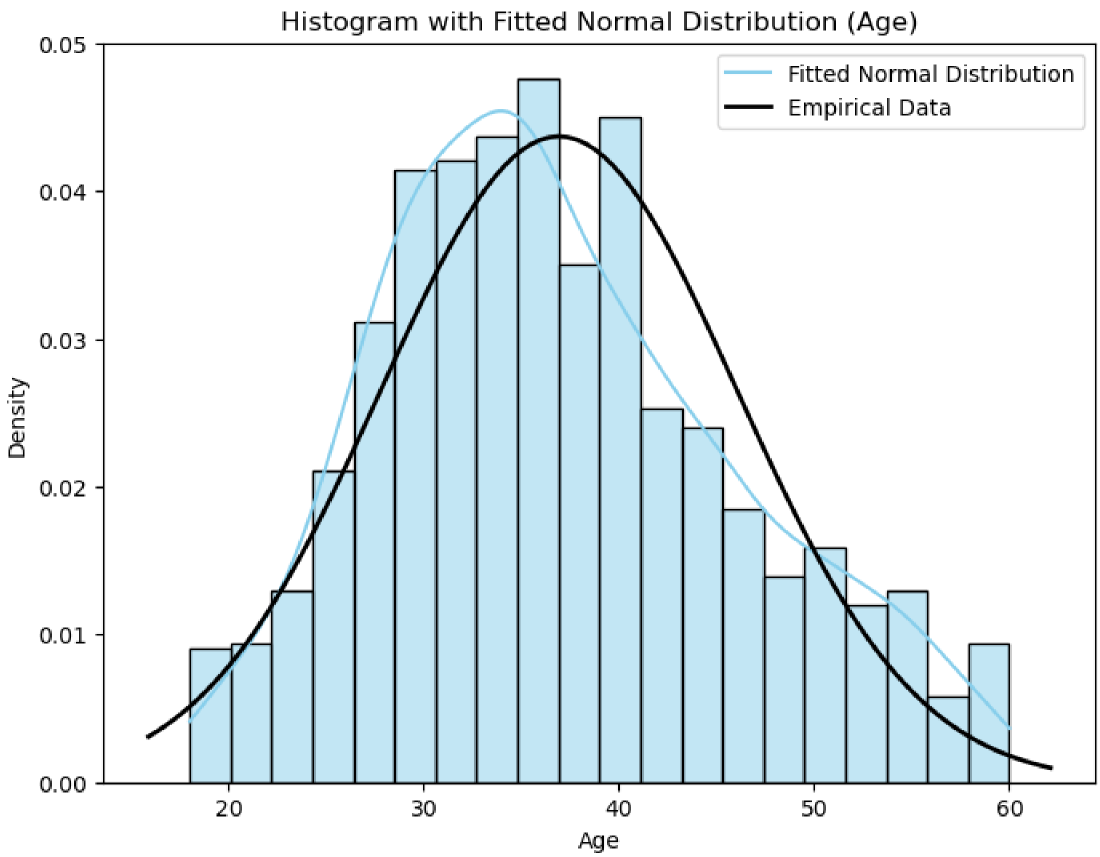
```python
# CDF of MonthlyIncome
sorted_income = df['MonthlyIncome'].sort_values()
cdf = np.arange(1, len(sorted_income) + 1) / len(sorted_income)

plt.figure(figsize=(8, 6))
plt.plot(sorted_income, cdf, marker='.', linestyle='none')
plt.xlabel('Monthly Income')
plt.ylabel('CDF')
plt.title('CDF of Monthly Income')
plt.grid(True)
plt.show()
```



png

```python
# What does this tell you about your variable and how does it address the
question you are trying to answer?

# The CTF of monthly income provides insight into the distribution of monthly
income within the organization. From left to right on the graph above, the
CDF curve rises, indicating an increase in the cumulative probability. The
corresponding Y value represents the proportion of individuals in the data
set with a monthly income less than or equal to that value. The graph is
providing information on the distribution of monthly income. The graph is
showing that less than 20% of the organization is making less than $2500 a
```

```
month. It also shows that about half of the organization is making $5000 a
month. There is a significant increase in the second half of the
organization. This increase is anywhere from over $5000 a month to over
$20,000 a month. It appears that the highest paid individuals within the
organization make anywhere from $10,000 a month to $20,000 a month. The top
10% within the organization have a $5000 differential of $15,000-$20,000.
This specific variable addresses the question of likelihood to stay within an
organization because people are driven by money. This is not always the
circumstance, but around 70% of this organization is making less than $6000 a
month. That is anywhere from $500-$1500 per week. The most highly paid person
within the organization is making $20,000. That is a $14,000 differential at
$5000 a week. If the employees within this organization know about this
differential and do not have a way of earning higher compensation, that could
be an incentive to leave or seek another job.

# Plot 1 analytical distribution.

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
from scipy import stats

# Load the CSV file into a DataFrame
df = pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")

# Select the 'Age' variable for analysis
data = df['Age']

# Fit a normal distribution to the data
mu, std = norm.fit(data)

# Plot the histogram of the data
plt.figure(figsize=(8, 6))
sns.histplot(data, kde=True, color='skyblue', stat='density', bins=20)

# Plot the fitted normal distribution
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2)

# Add labels and title
plt.xlabel('Age')
plt.ylabel('Density')
plt.title('Histogram with Fitted Normal Distribution (Age)')
plt.legend(['Fitted Normal Distribution', 'Empirical Data'])

plt.show()
```



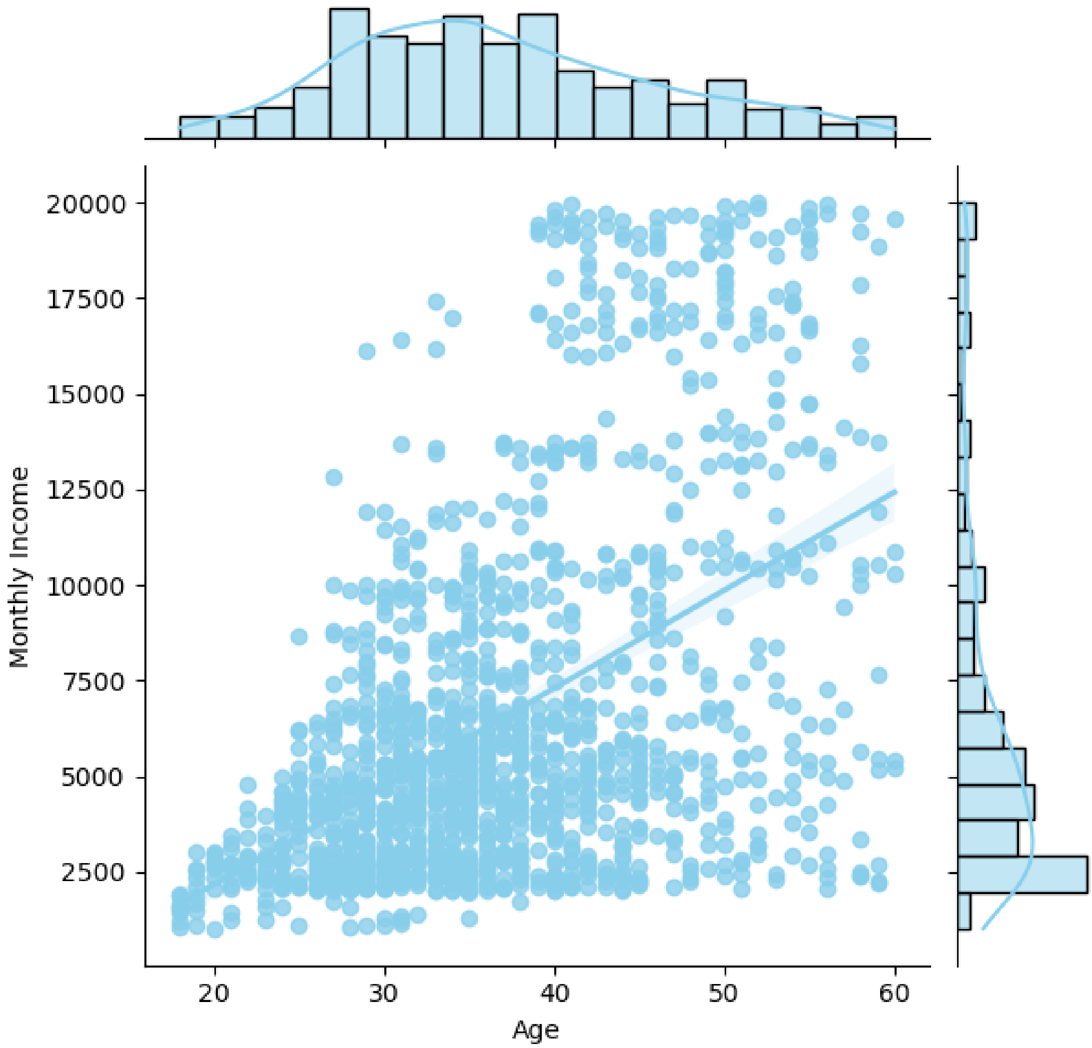
png

```
# Provide your analysis on how it applies to the dataset you have chosen.

# The age histogram shown above showcases the age range and density of age
within the organization. Within the histogram, it can be seen that the ages
range from 18 years old to 60 years old throughout the organization. The most
popular age for someone working within the organization is around 36 years of
age. The empirical data line shows that the median age is right around that
36 mark as well. The fitted normal distribution is skewed to the left showing
that most of the employees within the organization are 40 years or below. It
is much less common to see employees above the age of 40 and none at all
above the age of 60.

# Create two scatter plots comparing two variables and provide your analysis
on correlation and causation.

import seaborn as sns

# Selecting the variables
x = df['Age']
y = df['MonthlyIncome']

# Scatter plot
plt.figure(figsize=(8, 6))
```

```python
sns.scatterplot(x=x, y=y, color='skyblue')
plt.title('Scatter Plot of Age vs. Monthly Income')
plt.xlabel('Age')
plt.ylabel('Monthly Income')
plt.grid(True)
plt.show()

# Covariance
covariance = np.cov(x, y)[0][1]
print("Covariance:", covariance)

# Pearson's correlation coefficient
correlation = np.corrcoef(x, y)[0][1]
print("Pearson's correlation coefficient:", correlation)

# Non-linear Relationships

# Jointplot with regression line
sns.jointplot(x=x, y=y, kind="reg", color='skyblue')
plt.xlabel('Age')
plt.ylabel('Monthly Income')
plt.show()
```

png

```
Covariance: 21412.198982138805
Pearson's correlation coefficient: 0.4978545669265806
```



png

```
# Conduct a test on your hypothesis.

from scipy.stats import ttest_ind

# Split the data into two groups based on attrition
attrition_yes = df[df['Attrition'] == 'Yes']['MonthlyIncome']
attrition_no = df[df['Attrition'] == 'No']['MonthlyIncome']

# Perform the t-test
t_statistic, p_value = ttest_ind(attrition_yes, attrition_no)

# Print the results
print("T-Statistic:", t_statistic)
```

```
print("P-Value:", p_value)

# Interpretation
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis. There is a significant difference in
monthly income between employees who have left the company and those who are
still with the company.")
else:
    print("Fail to reject the null hypothesis. There is no significant
difference in monthly income between employees who have left the company and
those who are still with the company.")
```

```
T-Statistic: -6.203935765608938
P-Value: 7.147363985353381e-10
Reject the null hypothesis. There is a significant difference in monthly
income between employees who have left the company and those who are still
with the company.
```

```
# Conduct a regression analysis on either one dependent and one explanatory
variable, or multiple explanatory variables.

import statsmodels.api as sm

# Add a constant term to the explanatory variable
X = sm.add_constant(df['Age'])
y = df['MonthlyIncome']

# Fit the linear regression model
model = sm.OLS(y, X).fit()

# Print the regression summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:          MonthlyIncome   R-squared:
0.248
Model:                            OLS   Adj. R-squared:
0.247
Method:                 Least Squares   F-statistic:
483.8
Date:                Mon, 26 Feb 2024   Prob (F-statistic):           6.67e-
93
Time:                        23:52:07   Log-Likelihood:                   -
14308.
No. Observations:                1470   AIC:
2.862e+04
Df Residuals:                    1468   BIC:
```

```
2.863e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-
const      -2970.6712    443.702     -6.695      0.000   -3841.030   -
2100.313
Age          256.5716     11.665     21.995      0.000     233.689
279.454
==============================================================================
=
Omnibus:                      140.178   Durbin-Watson:
2.069
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
182.119
Skew:                           0.799   Prob(JB):                    2.84e-
40
Kurtosis:                       3.649   Cond. No.
159.
==============================================================================
=

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```