Section A Coding Practices

```
# Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Import necessary library to ignore all futurewarning dialogs
import warnings
warnings.filterwarnings("ignore")


# Load the dataset and create a dataframe object "df"
df = pd.read_csv("staff_dataset.csv")


df
```

	Age	BusinessTravel	MonthlyIncome	JobSatisfaction	Bonus	Department	DistanceFromHome	Educa
0	41	Travel_Rarely	5993	4	17979	Sales	1	
1	49	Travel_Frequently	5130	2	20520	Research & Development	8	
2	37	Travel_Rarely	2090	3	6270	Research & Development	2	
3	33	Travel_Frequently	2909	3	8727	Research & Development	3	
4	27	Travel_Rarely	3468	2	10404	Research & Development	2	
...	...	...	...	...	...	...	...	
1465	36	Travel_Frequently	2571	4	7713	Research & Development	23	
1466	39	Travel_Rarely	9991	1	29973	Research & Development	6	
1467	27	Travel_Rarely	6142	2	24568	Research & Development	4	
1468	49	Travel_Frequently	5390	2	16170	Sales	2	
1469	34	Travel_Rarely	4404	3	13212	Research & Development	8	

1470 rows × 24 columns

1. Determine the total number of attributes (columns)

```
# Determine the total number of attributes
print("Total number of attributes (columns):", len(df.columns))
```

```
Total number of attributes (columns): 24
```

2. assess the dataset's dimensions by identifying both the number of rows and columns

```
# Get the information of the dataset's dimensions (rows, columns)
df.shape
```

```
(1470, 24)
```

3. Calculate the average values for key attributes: 'Age', 'Monthly Income', and 'Years at Company'

```
ave_age = np.average(df["Age"]) # Average value for "Age" attribute
ave_monthly_income = np.average(df["MonthlyIncome"]) # Average value for "Monthly Income" attribute
ave_years_at_company = np.average(df["YearsAtCompany"]) # Average value for "Years at Company" attribute

# Rounding these average values to two decimal places to ensure precision in the analysis
print("Average age:", round(ave_age, 2))
print("Average Monthly Income:", round(ave_monthly_income, 2))
print("Average Years at Company:", round(ave_years_at_company, 2))
```

```
Average age: 36.92
Average Monthly Income: 6502.93
Average Years at Company: 7.01
```

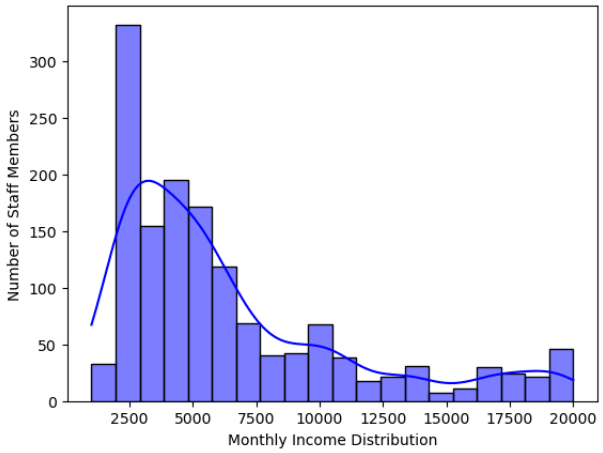
4. Finding both the minimum and maximum values in the "MonthlyIncome" attribute

```
print("Maximum Monthly Income:", max(df["MonthlyIncome"])) # Maximum value in the "MonthlyIncome" attribute
print("Minimum Monthly Income:", min(df["MonthlyIncome"])) # Minimun value in the "MonthlyIncome" attribute
```

```
Maximum Monthly Income: 19999
Minimum Monthly Income: 1009
```

5. Create a histogram plotting "MonthlyIncome" against the number of staff members

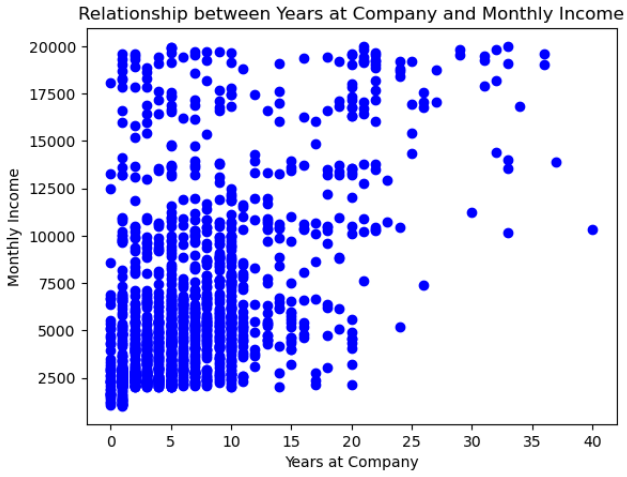
```
# Create a histogram plotting "MonthlyIncome" against the number of staff members
sns.histplot(df["MonthlyIncome"], kde = True, color = "blue")
plt.xlabel("Monthly Income Distribution") # label x as "Monthly Income Distribution"
plt.ylabel("Number of Staff Members") # label y as "Number of Staff Members"
plt.show() # display the histogram
```



The right-skewed histogram plotting “MonthlyIncome” is generated, which illustrates the most frequent monthly income range is approximately the income range around 2500.

6. Create a scatter plot to examine the relationship between "Years at Company" and "Monthly Income"

```
# Use scatter plot, "YearsAtCompany" as x, and "MonthlyIncome" as y
plt.scatter(df["YearsAtCompany"], df["MonthlyIncome"], color = "blue")
plt.xlabel("Years at Company") # label x as "Years at Company"
plt.ylabel("Monthly Income") # label y as "Monthly Income"
plt.title("Relationship between Years at Company and Monthly Income") # label the title of this scatter plot
plt.show() # display the scatter plot
```



Most employees are concentrated between 0 to 10 years.
→ Higher incomes do not significantly increase with tenure beyond a certain point

The pattern of income distribution is not rigorously linear, hinting at the influence of other factors beyond tenure that could influence monthly income levels.

7. Calculate the correlation coefficient between Years at Company' and 'Monthly Income'

```
corr_coef = df[["MonthlyIncome", "YearsAtCompany"]].corr()
corr_coef
```

	MonthlyIncome	YearsAtCompany
MonthlyIncome	1.000000	0.514285
YearsAtCompany	0.514285	1.000000

Correlation coefficient of 0.514285 (Moderate positive relationship)

8. Further discussions

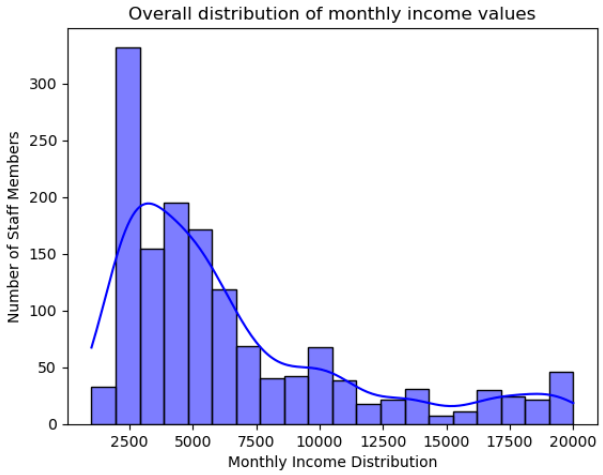
```
# a. The range of monthly income at Company A
income_range = max(df["MonthlyIncome"]) - min(df["MonthlyIncome"])
print("The range of monthly income at Company A:", income_range)

max_income = max(df["MonthlyIncome"])
min_income = min(df["MonthlyIncome"])
print(f"Monthly Income ranges from {min_income} to {max_income}.")
```

```
The range of monthly income at Company A: 18990
Monthly Income ranges from 1009 to 19999.
```

```
# b. The most and least frequent monthly income values
income_freqs = df["MonthlyIncome"].value_counts()
print("The most frequent monthly income value:", income_freqs.idxmax())
print("The least frequent monthly income value:", income_freqs.idxmin())
```

```
The most frequent monthly income value: 2342
The least frequent monthly income value: 3423
```

```
# c. the overall distribution of monthly income values
sns.histplot(df["MonthlyIncome"], kde = True, color = "blue")
plt.xlabel("Monthly Income Distribution")
plt.ylabel("Number of Staff Members")
plt.title("Overall distribution of monthly income values")
plt.show()
```



This histogram of staff members' monthly income illustrates that employees' earnings are highly concentrated between approximately 2,000 and 3,000, which represents the mode of this monthly income distribution. This distribution illustration is right-skewed, impling that only a few staff members have much higher salaries. This also indicates the existence of outliers or a few individuals with extraordinarily larger monthly incomes. However, monthly income values are widely distributed, showing considerable variation among staff members' earnings.

Due to the skewness, the mean of this income distribution is likely higher than the median, representing that the average income is pulled up by higher earners. The density curve overlaid on the histogram smooths out the distribution, confirming the skewness and variability in staff incomes. In general, this suggests a diverse income range within the organization with most employees earning at the lower end of the spectrum but with some exceptional higher earners.

```
# d. whether there appears to be a linear relationship between an employee's years at the company and their monthly income

from scipy.stats import linregress
from sklearn.metrics import r2_score


slope, intercept, r_value, p_value, std_err = linregress(df["YearsAtCompany"], df["MonthlyIncome"])
print(f"Slope is: {slope}")
print(f"Intercept is: {intercept}", "\n")

r2 = round(r_value ** 2, 4)
print(f"R-squared value is: {r2}")
```

```
Slope is: 395.20456923368243
Intercept is: 3733.2731481323835

R-squared value is: 0.2645
```

Since the valid slope and intercept are provided, there is the linear relationship between an employee's years at the company and their monthly income. On the other hand, because the R-square value is 0.2645, we could say the relationship is extremely weak and not very trustworthy.