HW 2: Types of Data

CPE232 Data Models

Import Dependency

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

Part 1 : Basic Python

Task 1

Perform the following:

- Write a program to get input of different types from the user.
- Display type of each variable.
- Convert a variable of the type float to Integer.
- Demonstrate a comparison of before vs after the type conversion.

In [3]:
```
# Input values from the user
string_value = input("Enter a string: ")
integer_value = int(input("Enter an integer: "))
float_value = float(input("Enter a float: "))
```

In [4]:
```
# Display the data types
print(type(string_value))
print(type(integer_value))
print(type(float_value))
```

```
<class 'str'>
<class 'int'>
<class 'float'>
```

In [5]:
```
# Convert integer to float and vice versa
convert_int_to_float = float(integer_value)
convert_float_to_int = int(float_value)
```

In [6]:
```
# Show before and after convert float to int
print("Convert integer_value")
print("Before Convert: ",integer_value)
print("After Convert: ",convert_int_to_float)
print("Convert float_valie")
print("Before Convert: ",float_value)
print("After Convert: ",convert_float_to_int)
```

```
Convert integer_value
Before Convert:  12
After Convert:  12.0
Convert float_valie
Before Convert:  5.53
After Convert:  5
```

Task 2

Given a dictionary of students with their scores, find the average score of each student across all subjects and identify the student with the highest average score.

In [7]:

```
students_grades = {
    "John": [85, 90, 78],
    "Alice": [88, 92, 80],
    "Bob": [75, 85, 72],
    "Diana": [90, 95, 94],
    "Charlie": [70, 65, 80]
}
```

In [8]:

```
# Calculate and print the average grade for each student
for key in students_grades:
    print(key,sum(students_grades[key])/len(students_grades[key]))
```

```
John 84.33333333333333
Alice 86.66666666666667
Bob 77.33333333333333
Diana 93.0
Charlie 71.66666666666667
```

In [9]:

```
# Find the student with the highest average grade
mx = 0
mxSt = ""
for key in students_grades:
    avg = sum(students_grades[key])/len(students_grades[key])
    if avg > mx :
        mx = avg
        mxSt = key
print(mxSt)
```

```
Diana
```

Part 2 : Working with CSV!

Add a new column to this CSV file named "Bonus," and calculate the bonus for employees in the Sales department as 10% of their MonthlyRate.

In [10]:

```
# Load the CSV file
file_path = 'employee_data.csv'
```

```
df = pd.read_csv(file_path)
```

In [11]:
```
df.head()
```

Out[11]:

	Age	DailyRate	Department	EducationField	Gender	MaritalStatus	MonthlyRate	Ove
0	41	1102	Sales	Life Sciences	Female	Single	19479	
1	49	279	Research & Development	Life Sciences	Male	Married	24907	
2	37	1373	Research & Development	Other	Male	Single	2396	
3	33	1392	Research & Development	Life Sciences	Female	Married	23159	
4	27	591	Research & Development	Medical	Male	Married	16632	

In [12]:
```
# Add new column call 'Bonus'
df["Bonus"] = 0
df.head()
```

Out[12]:

	Age	DailyRate	Department	EducationField	Gender	MaritalStatus	MonthlyRate	Ove
0	41	1102	Sales	Life Sciences	Female	Single	19479	
1	49	279	Research & Development	Life Sciences	Male	Married	24907	
2	37	1373	Research & Development	Other	Male	Single	2396	
3	33	1392	Research & Development	Life Sciences	Female	Married	23159	
4	27	591	Research & Development	Medical	Male	Married	16632	

In [13]:
```
# Calculate 10% of bonus into the 'Bonus' Column
df["Bonus"] = df["MonthlyRate"]/10
```

In [14]:
```
# Print and show result
df
```

Out[14]:

	Age	DailyRate	Department	EducationField	Gender	MaritalStatus	MonthlyRate
0	41	1102	Sales	Life Sciences	Female	Single	19479
1	49	279	Research & Development	Life Sciences	Male	Married	24907
2	37	1373	Research & Development	Other	Male	Single	2396
3	33	1392	Research & Development	Life Sciences	Female	Married	23159
4	27	591	Research & Development	Medical	Male	Married	16632
...	...	...	...	...	...	...	...
1465	36	884	Research & Development	Medical	Male	Married	12290
1466	39	613	Research & Development	Medical	Male	Married	21457
1467	27	155	Research & Development	Life Sciences	Male	Married	5174
1468	49	1023	Sales	Medical	Male	Married	13243
1469	34	628	Research & Development	Medical	Male	Married	10228

1470 rows × 9 columns

In [15]:

```
# Save the updated DataFrame back to a CSV file
df.to_csv("employee_data_update.csv")
```

Part 3 : Working with Matplotlib

Plot a graph showing the average salary of employees in each department to compare the average salaries across departments.

In [16]:

```
# Load CSV and create DataFrame
data = pd.read_csv('employee_data.csv')
df = pd.DataFrame(data)
```

In [17]:

```
# Calculate average MonthlyRate by Department
chart_df = df.groupby("Department").agg({
    "MonthlyRate" : "mean"
})
chart_df
```

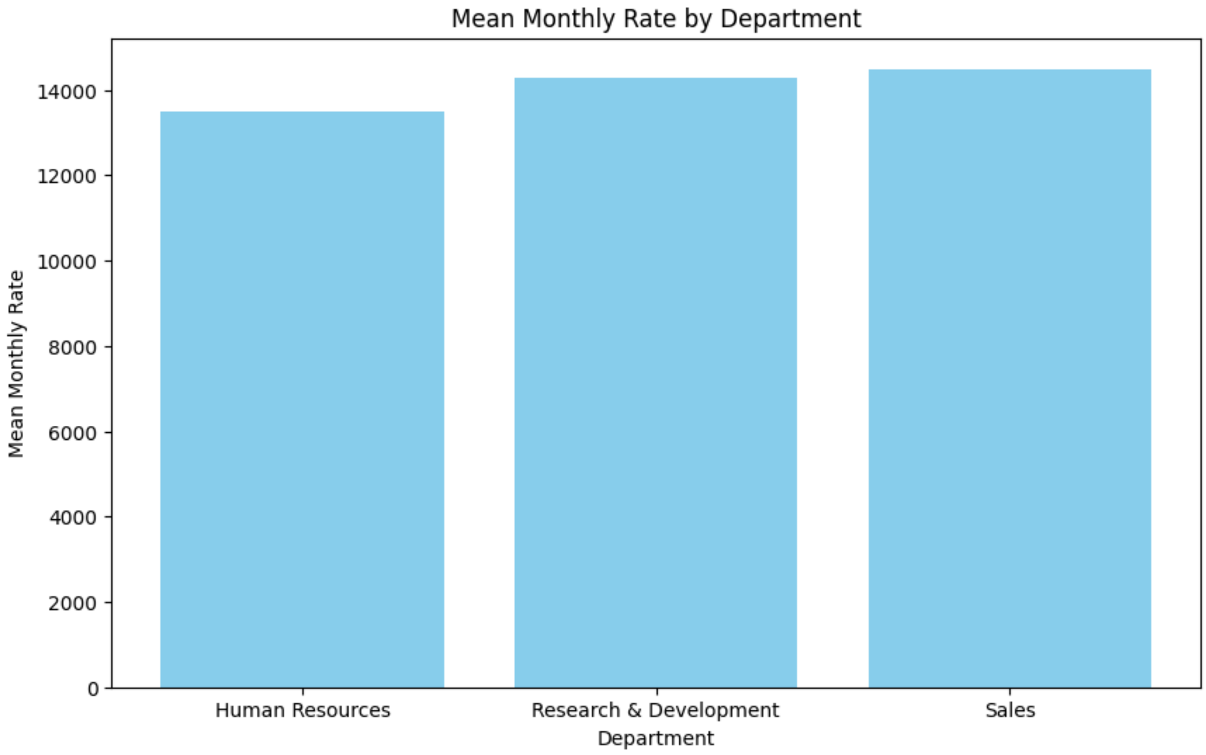
```
Out[17]:
```

	MonthlyRate
Department	
Human Resources	13492.984127
Research & Development	14284.865765
Sales	14489.793722

```
In [18]:
```

```python
# Plotting the bar chart
plt.figure(figsize=(10, 6))
plt.bar(chart_df.index, chart_df['MonthlyRate'], color='skyblue')
plt.xlabel('Department')
plt.ylabel('Mean Monthly Rate')
plt.title('Mean Monthly Rate by Department')
```

```
Out[18]: Text(0.5, 1.0, 'Mean Monthly Rate by Department')
```



Challenge!!! : Working with SATAN (Optional)

Great job, for not giving up on this subject!

Here’s the story: The CSV file that was given to you isn’t just an ordinary file -- it’s data from an organization where Alya-San works!

Alya feels that something suspicious is going on within the company. She suspects that there’s inequality in salaries between male and female employees.

She also wonders if the government might be intervening to encourage population growth by secretly increasing the salaries of married employees compared to those who are single.

Additionally, how much of a difference is there in salaries between employees who graduated from different fields of study?

Can you help Alya prove whether her suspicions are just in her head or if they’re actually true?

For Aria, everyone is truly **equal**.

In [19]:

```
# Implement code here
gender_rate_df = df.groupby("Gender").agg({
    "MonthlyRate" : "mean"
})
print("Average MonthlyRate by Gender")
gender_rate_df
```

```
Average MonthlyRate by Gender
```

Out[19]:

	MonthlyRate
Gender	
Female	14674.600340
Male	14072.105442

In [20]:

```
marriage_rate_df = df.groupby("MaritalStatus").agg({
    "MonthlyRate" : "mean"
})
print("Average MonthlyRate by Marital Status")
marriage_rate_df
```

```
Average MonthlyRate by Marital Status
```

Out[20]:

	MonthlyRate
MaritalStatus	
Divorced	14310.085627
Married	14044.502229
Single	14699.817021

In [21]:

```
study_rate_df = df.groupby("EducationField").agg({
    "MonthlyRate" : "mean"
})
print("Average MonthlyRate by EducationField")
study_rate_df
```

```
Average MonthlyRate by EducationField
```

Out[21]:

	MonthlyRate
EducationField	
Human Resources	14810.740741
Life Sciences	14530.132013
Marketing	14076.943396
Medical	14295.056034
Other	13270.780488
Technical Degree	14210.363636

In [24]:

```
rate_df = df.groupby(["Gender","MaritalStatus","EducationField"]).agg({
    "MonthlyRate" : "mean"
})
print("Average MonthlyRate each group")
rate_df
```

```
Average MonthlyRate each group
```

Out[24]:

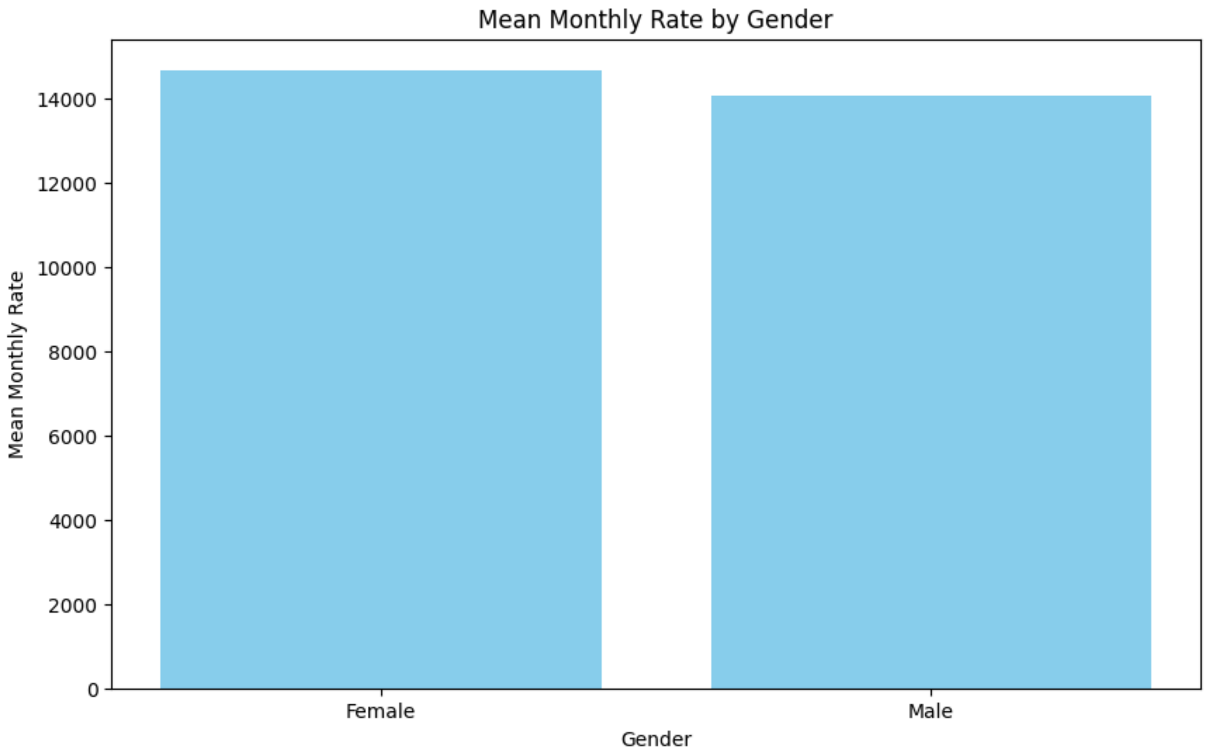
			MonthlyRate
Gender	**MaritalStatus**	**EducationField**	
Female	**Divorced**	**Human Resources**	10632.500000
		Life Sciences	14908.863636
		Marketing	10144.454545
		Medical	16488.840909
		Other	8053.200000
		Technical Degree	10555.181818
	Married	**Human Resources**	19303.000000
		Life Sciences	14285.870370
		Marketing	13711.078947
		Medical	14600.024390
		Other	15594.882353
		Technical Degree	15733.363636
	Single	**Human Resources**	13871.000000
		Life Sciences	15580.602273
		Marketing	14695.150000
		Medical	14193.046875
		Other	12789.857143
		Technical Degree	16438.368421
Male	**Divorced**	**Human Resources**	13089.000000
		Life Sciences	14571.855556
		Marketing	14712.217391
		Medical	14100.761905
		Other	11243.142857
		Technical Degree	16508.466667
	Married	**Human Resources**	15274.692308
		Life Sciences	13918.134969
		Marketing	14980.743590
		Medical	13145.515625
		Other	13531.789474

			MonthlyRate
Gender	**MaritalStatus**	**EducationField**	
		Technical Degree	12612.435897
	Single	**Human Resources**	4223.000000
		Life Sciences	14647.610619
		Marketing	13896.107143
		Medical	14829.710843
		Other	13939.400000
		Technical Degree	13911.000000

Since Kobayashi-san will be reviewing your code, please present the data in a way that is clear and accurate. Otherwise, Kobayashi-san might fire you from the company!

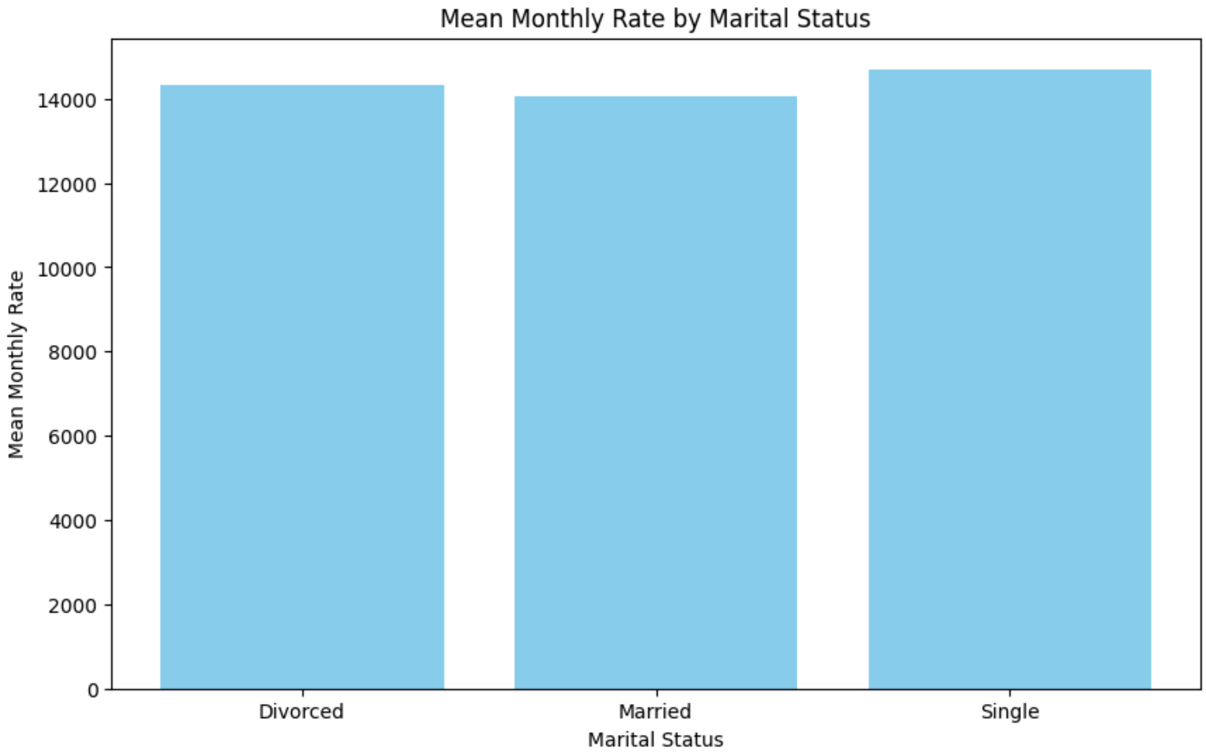
```
In [ ]: # Write your code here
        plt.figure(figsize=(10, 6))
        plt.bar(gender_rate_df.index, gender_rate_df['MonthlyRate'], color='skyblue')
        plt.xlabel('Gender')
        plt.ylabel('Mean Monthly Rate')
        plt.title('Mean Monthly Rate by Gender')
```

```
Out[ ]: Text(0.5, 1.0, 'Mean Monthly Rate by Gender')
```



```
In [ ]: plt.figure(figsize=(10, 6))
        plt.bar(marriage_rate_df.index, marriage_rate_df['MonthlyRate'], color='skyblue')
        plt.xlabel('Marital Status')
        plt.ylabel('Mean Monthly Rate')
        plt.title('Mean Monthly Rate by Marital Status')
```

```
Out[ ]: Text(0.5, 1.0, 'Mean Monthly Rate by Marital Status')
```

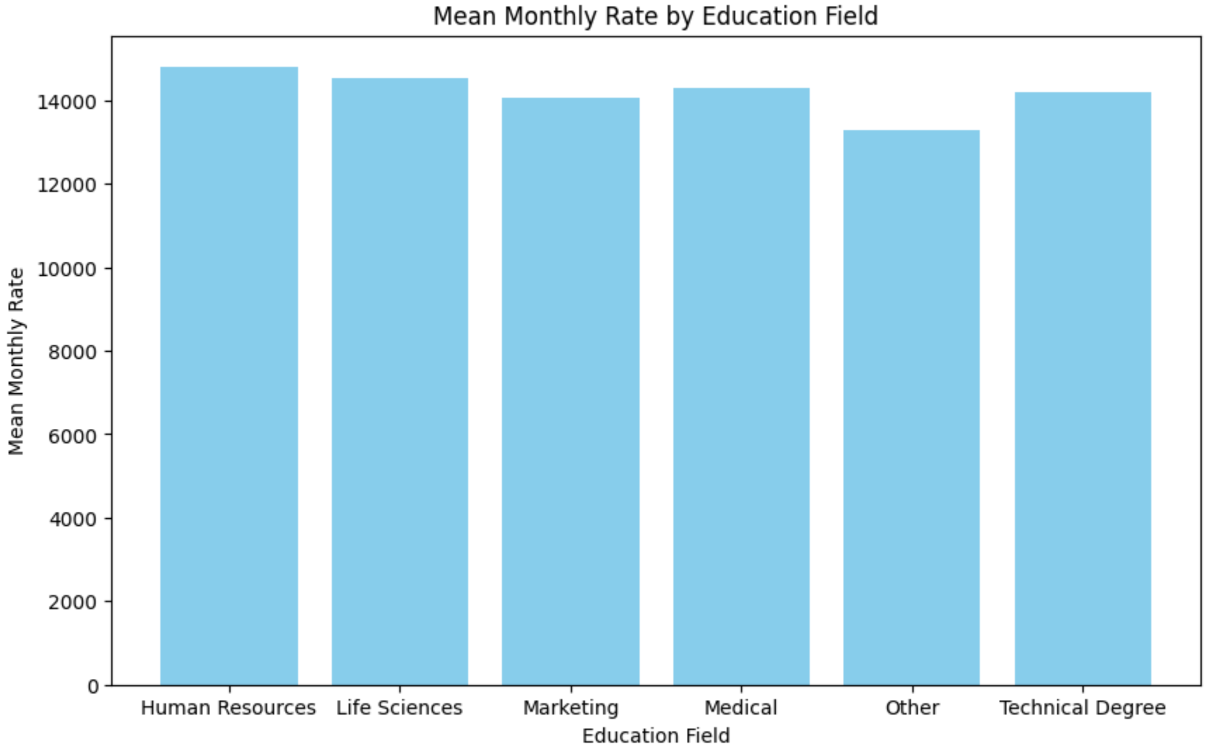


In [ ]:

```python
plt.figure(figsize=(10, 6))
plt.bar(study_rate_df.index, study_rate_df['MonthlyRate'], color='skyblue')
plt.xlabel('Education Field')
plt.ylabel('Mean Monthly Rate')
plt.title('Mean Monthly Rate by Education Field')
```

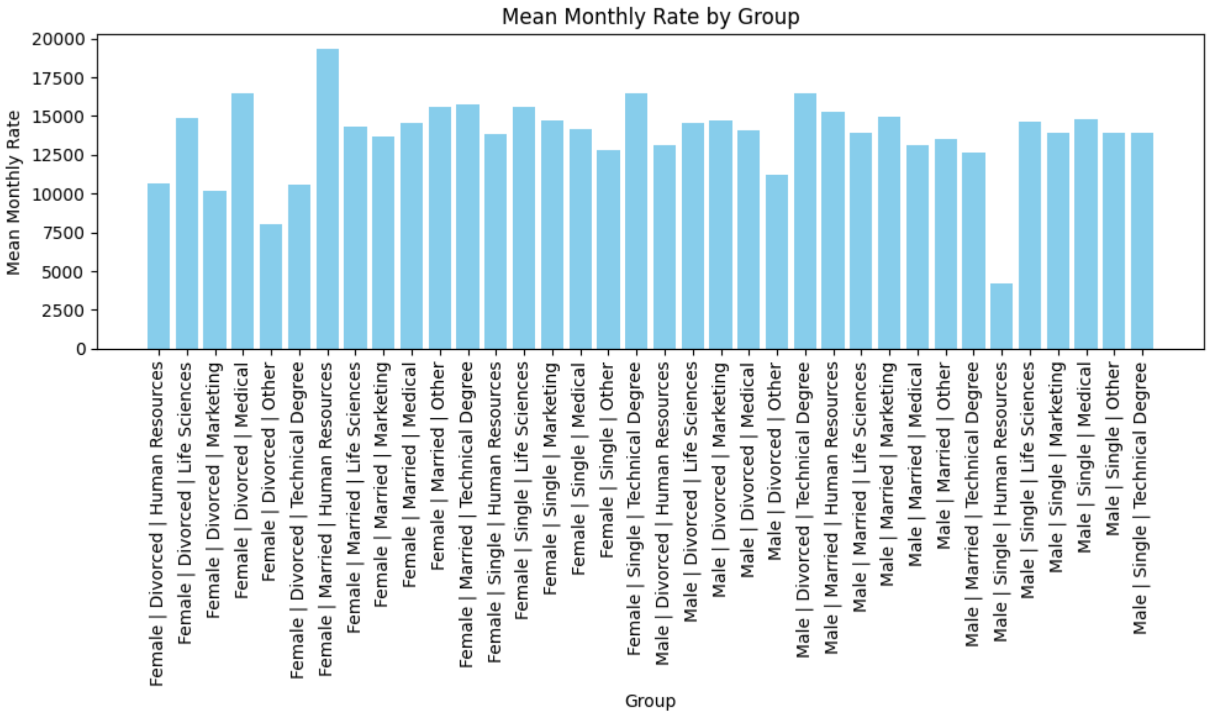
Out[ ]:

```
Text(0.5, 1.0, 'Mean Monthly Rate by Education Field')
```



In [26]:

```python
plt.figure(figsize=(10, 6))

labels = [" | ".join(idx) for idx in rate_df.index]

plt.bar(labels, rate_df['MonthlyRate'], color='skyblue')

plt.xlabel('Group')
plt.ylabel('Mean Monthly Rate')
plt.title('Mean Monthly Rate by Group')

plt.xticks(rotation=90)

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


Mean Monthly Rate by Group
Mean Monthly Rate
20000
17500
15000
12500
10000
7500
5000
2500
0
Female | Divorced | Human Resources
Female | Divorced | Life Sciences
Female | Divorced | Marketing
Female | Divorced | Medical
Female | Divorced | Other
Female | Divorced | Technical Degree
Female | Married | Human Resources
Female | Married | Life Sciences
Female | Married | Marketing
Female | Married | Medical
Female | Married | Other
Female | Married | Technical Degree
Female | Single | Human Resources
Female | Single | Life Sciences
Female | Single | Marketing
Female | Single | Medical
Female | Single | Other
Female | Single | Technical Degree
Male | Divorced | Human Resources
Male | Divorced | Life Sciences
Male | Divorced | Marketing
Male | Divorced | Medical
Male | Divorced | Other
Male | Divorced | Technical Degree
Male | Married | Human Resources
Male | Married | Life Sciences
Male | Married | Marketing
Male | Married | Medical
Male | Married | Other
Male | Married | Technical Degree
Male | Single | Human Resources
Male | Single | Life Sciences
Male | Single | Marketing
Male | Single | Medical
Male | Single | Other
Male | Single | Technical Degree
Group

In [ ]: