

HW 2 - Types of Data (INTL)

February 4, 2025

1 HW 2: Types of Data

1.1 ##### CPE232 Data Models

2 Import Dependency

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

3 Part 1 : Basic Python

3.1 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.

```
[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: "))
```

```
[4]: # Display the data types
print("String_value is of type:", type(string_value))
print("Integer_value is of type:", type(integer_value))
print("Float_value is of type:", type(float_value))
```

```
String_value is of type: <class 'str'>
Integer_value is of type: <class 'int'>
Float_value is of type: <class 'float'>
```

```
[5]: # Convert integer to float and vice versa
integer_to_float = float(integer_value)
float_to_integer = int(float_value)

# Display the data types
print("Integer_to_float is of type:", type(integer_to_float))
print("Float_to_integer is of type:", type(float_to_integer))
```

Integer_to_float is of type: <class 'float'>
Float_to_integer is of type: <class 'int'>

```
[6]: # Show before and after convert float to int
# print the values
print(integer_value, "-->", integer_to_float)
print(float_value, "-->", float_to_integer)
```

12 --> 12.0
3.5 --> 3

3.2 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.

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

```
[8]: highest_grade = 0

# Calculate and print the average grade for each student
for student, grades in students_grades.items():
    average_grade = sum(grades) / len(grades)
    if average_grade > highest_grade:
        highest_grade = average_grade
        highest_grade_student = student
    print(student, "average grade:", average_grade)

# Print the student with the highest average grade
print("\nThe student with the highest average grade is:",
      highest_grade_student, "with an average grade of:", highest_grade)
```

John average grade: 84.33333333333333
Alice average grade: 86.66666666666667
Bob average grade: 77.33333333333333
Diana average grade: 93.0
Charlie average grade: 71.66666666666667

The student with the highest average grade is: Diana with an average grade of:
93.0

4 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.

```
[9]: # Load the CSV file
file_path = 'employee_data.csv'
df = pd.read_csv(file_path)
```

```
[10]: df.head()
```

```
[10]:
```

	Age	DailyRate	Department	EducationField	Gender	\
0	41	1102	Sales	Life Sciences	Female	
1	49	279	Research & Development	Life Sciences	Male	
2	37	1373	Research & Development	Other	Male	
3	33	1392	Research & Development	Life Sciences	Female	
4	27	591	Research & Development	Medical	Male	

	MaritalStatus	MonthlyRate	OverTime
0	Single	19479	Yes
1	Married	24907	No
2	Single	2396	Yes
3	Married	23159	Yes
4	Married	16632	No

```
[11]: # Add new column call 'Bonus'
# Calculate 10% of bonus into the 'Bonus' Column
df['Bonus'] = df['MonthlyRate'] * 0.1
```

```
[12]: # Print and show result
df
```

```
[12]:
```

	Age	DailyRate	Department	EducationField	Gender	\
0	41	1102	Sales	Life Sciences	Female	
1	49	279	Research & Development	Life Sciences	Male	
2	37	1373	Research & Development	Other	Male	
3	33	1392	Research & Development	Life Sciences	Female	
4	27	591	Research & Development	Medical	Male	
...	
1465	36	884	Research & Development	Medical	Male	
1466	39	613	Research & Development	Medical	Male	
1467	27	155	Research & Development	Life Sciences	Male	
1468	49	1023	Sales	Medical	Male	
1469	34	628	Research & Development	Medical	Male	

	MaritalStatus	MonthlyRate	OverTime	Bonus
0	Single	19479	Yes	1947.9
1	Married	24907	No	2490.7

2	Single	2396	Yes	239.6
3	Married	23159	Yes	2315.9
4	Married	16632	No	1663.2
...
1465	Married	12290	No	1229.0
1466	Married	21457	No	2145.7
1467	Married	5174	Yes	517.4
1468	Married	13243	No	1324.3
1469	Married	10228	No	1022.8

[1470 rows x 9 columns]

```
[13]: # Save the updated DataFrame back to a CSV file
df.to_csv(file_path, index=False)
```

5 Part 3 : Working with Matplotlib

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

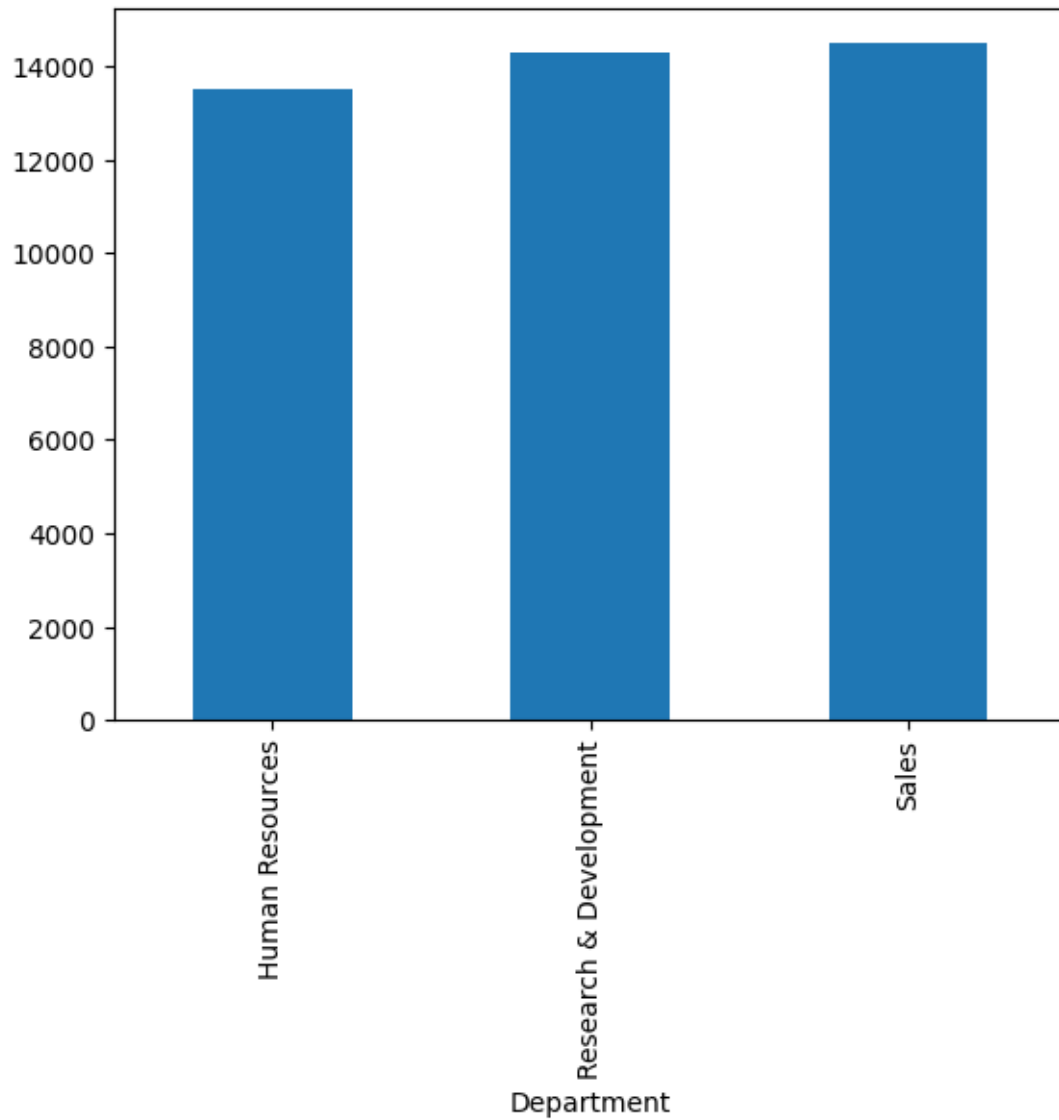
```
[14]: # Load CSV and create DataFrame
data = pd.read_csv('employee_data.csv')
df = pd.DataFrame(data)
```

```
[15]: # Calculate average MonthlyRate by Department
average_monthly_rate = df.groupby('Department')['MonthlyRate'].mean()
print(average_monthly_rate)
```

```
Department
Human Resources      13492.984127
Research & Development  14284.865765
Sales                14489.793722
Name: MonthlyRate, dtype: float64
```

```
[16]: # Plotting the bar chart
average_monthly_rate.plot(kind='bar')
```

```
[16]: <Axes: xlabel='Department'>
```



6 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 **e q u a l**.

```
[17]: # Implement code here
average_monthly_rate.gender = df.groupby('Gender')['MonthlyRate'].mean()
average_monthly_rate.education_field = df.
    ↳groupby('EducationField')['MonthlyRate'].mean()
average_monthly_rate.maritalStatus = df.groupby('MaritalStatus')['MonthlyRate'].
    ↳mean()

print(average_monthly_rate.gender, '\n')
print(average_monthly_rate.maritalStatus, '\n')
print(average_monthly_rate.education_field)
```

```
Gender
Female    14674.600340
Male      14072.105442
Name: MonthlyRate, dtype: float64
```

```
MaritalStatus
Divorced    14310.085627
Married     14044.502229
Single      14699.817021
Name: MonthlyRate, dtype: float64
```

```
EducationField
Human Resources    14810.740741
Life Sciences      14530.132013
Marketing          14076.943396
Medical            14295.056034
Other              13270.780488
Technical Degree   14210.363636
Name: MonthlyRate, dtype: float64
```

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!

```
[20]: # Plotting the bar chart
plt.figure(figsize=(15, 5))

# Plotting the bar chart for Gender
plt.subplot(1, 3, 1)
bars = average_monthly_rate.gender.sort_values(ascending=False).
    ↳plot(kind='bar', title='Average Monthly Rate by Gender', color='red')
plt.xticks(rotation=45)
for bar in bars.patches:
```

```

    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), round(bar.
↳get_height(), 2), ha='center', color='black')

# Plotting the bar chart for Marital Status
plt.subplot(1, 3, 2)
bars = average_monthly_rate.maritalStatus.sort_values(ascending=False).
↳plot(kind='bar', title='Average Monthly Rate by Marital Status',
↳color='green')
plt.xticks(rotation=45)
for bar in bars.patches:
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), round(bar.
↳get_height(), 2), ha='center', color='black')

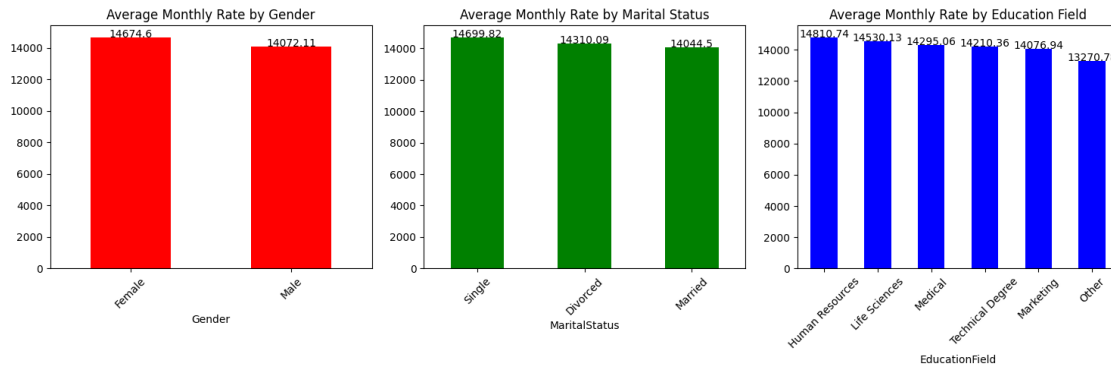
# Plotting the bar chart for Education Field
plt.subplot(1, 3, 3)
bars = average_monthly_rate.education_field.sort_values(ascending=False).
↳plot(kind='bar', title='Average Monthly Rate by Education Field',
↳color='blue')
plt.xticks(rotation=45)
for bar in bars.patches:
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), round(bar.
↳get_height(), 2), ha='center', color='black')

plt.tight_layout()
plt.show()

# Calculate percentage of different in gender and marital status and education
↳field
gender_diff = (average_monthly_rate.gender.max() - average_monthly_rate.gender.
↳min()) / average_monthly_rate.gender.min() * 100
marital_status_diff = (average_monthly_rate.maritalStatus.max() -
↳average_monthly_rate.maritalStatus.min()) / average_monthly_rate.
↳maritalStatus.min() * 100
education_field_diff = (average_monthly_rate.education_field.max() -
↳average_monthly_rate.education_field.min()) / average_monthly_rate.
↳education_field.min() * 100

print("Percentage difference in average income rate (min-max):")
print(f"by gender: {gender_diff:.2f}%")
print(f"by marital status: {marital_status_diff:.2f}%")
print(f"by education field: {education_field_diff:.2f}%")

```



Percentage difference in average income rate (min-max):

by gender: 4.28%

by marital status: 4.67%

by education field: 11.60%

Gender-based income differences stand at 4.28% which is lowest rate, female has more income. Marital status contributes to a 4.67% variation with single as highest follow by divorced and married respective. However, the most disparity is based on the field of education, with an 11.60% difference HR trend to have highest income follow by Life sciences, Medical, Technical Degree, Marketing, and other as lowest.