

# Quiz 4 (in-class)

- Due Dec 1 at 12:20pm
- Points 10
- Questions 1
- Available Dec 1 at 11:45am - Dec 1 at 12:25pm 40 minutes
- Time Limit 25 Minutes

## Instructions

Due on Dec. 1.

This quiz was locked Dec 1 at 12:25pm.

## Attempt History

	Attempt	Time	Score
LATEST	<a href="#">Attempt 1</a>	20 minutes	8 out of 10

Score for this quiz: 8 out of 10

Submitted Dec 1 at 12:04pm

This attempt took 20 minutes.



Question 1

8 / 10 pts

# Learning Objective

- Gain proficiency in using Pandas for data manipulation and analysis.
- Understand and apply advanced Pandas operations including `apply`, `groupby`, and `lambda`.
- Work with DataFrame transformations, such as splitting, shuffling, inserting, and deleting columns.

## Problem Description

You are given a Pandas DataFrame named **employees**, which contains information about employees in a company. The schema of the **employees** table is as follows:

Column Name	Type
<code>employee_id</code>	<code>int</code>
<code>name</code>	<code>string</code>
<code>age</code>	<code>int</code>
<code>department</code>	<code>string</code>
<code>salary</code>	<code>int</code>

- **employee\_id**: A unique identifier for each employee.
- **name**: The name of the employee.
- **age**: The age of the employee.
- **department**: The department the employee works in.
- **salary**: The annual salary of the employee.

Your task is to perform the following operations step-by-step:

1. **Shuffle:** Randomly shuffle the rows of the DataFrame.
2. **Split:** Add a new column called `salary_level`, which classifies each employee's salary into three categories:
  - **Low:** Salaries below \$50,000.
  - **Medium:** Salaries between \$50,000 and \$100,000 (inclusive).
  - **High:** Salaries above \$100,000.
3. **Groupby and Summarize:** Group the data by `department` and calculate the following:
  - The total salary for each department.
  - The average age of employees in each department.
4. **Transform:** Insert a new column called `bonus`, which is calculated as 10% of each employee's salary, and round the result to the nearest integer.
5. **Delete:** Remove the `age` column from the DataFrame.

Finally, return two outputs:

- The modified DataFrame after performing all transformations.
- A summary DataFrame showing the department-level statistics (total salary and average age).

## Input

A Pandas DataFrame `employees` with the following columns:

- `employee_id` (int): The unique ID for each employee.
- `name` (string): The name of the employee.
- `age` (int): The age of the employee.
- `department` (string): The department of the employee.
- `salary` (int): The annual salary of the employee.

## Output

- A modified DataFrame after all transformations.
- A summary DataFrame containing:
  - `department` (string): The name of the department.
  - `total_salary` (int): The total salary of all employees in the department.
  - `average_age` (float): The average age of employees in the department.

# Examples

## Example 1

Input:

employees =	employee_id	name	age	department	salary
	1	Alice	25	HR	45000
	2	Bob	30	Engineering	95000
	3	Carol	28	Marketing	120000
	4	David	35	HR	60000
	5	Eve	40	Engineering	75000

Output: Modified DataFrame:

employee_id	name	department	salary	salary_level	bonus
3	Carol	Marketing	120000	High	12000
1	Alice	HR	45000	Low	4500
4	David	HR	60000	Medium	6000
5	Eve	Engineering	75000	Medium	7500
2	Bob	Engineering	95000	Medium	9500

Output: Summary DataFrame:

department	total_salary	average_age
Engineering	170000	35.0
HR	105000	30.0
Marketing	120000	28.0

## Requirements

1. The solution to this problem must be implemented using [Python and Pandas only](#). No other programming languages or libraries are allowed.
2. You must implement all steps described in the problem using appropriate Pandas functions and methods.
3. Respect the honor code: [you are not allowed to use the internet, including ChatGPT or any other AI tools, to assist in the implementation of this problem. Otherwise, points will be deducted.](#)
4. Do not change the framework provided to you. Only complete the function `process_employees`.

Partial codes:

```
import pandas as pd
```

```
def process_employees(employees: pd.DataFrame) -> (pd.DataFrame, pd.DataFrame):
```

```
    """
```

Processes the employees DataFrame as per the described steps.

Args:

employees (pd.DataFrame): The input DataFrame with employee details.

Returns:

tuple: A tuple containing:

- A modified DataFrame after all transformations.
- A summary DataFrame with department-level statistics.

"""

# Your code starts here

# Your code ends here

Hints:

- Use the `sample(frac=1)` function to shuffle the rows of a DataFrame randomly.
- Apply a lambda function to classify salaries into the salary level column.
- Use `groupby` and `agg` to calculate the total salary and average age for each department.
- Use the `insert()` method to add new columns, and `drop()` to remove existing columns.

Your Answer:

```
import pandas as pd
```

```
def process_employees(employees: pd.DataFrame) -> (pd.DataFrame, pd.DataFrame):
```

```
    """
```

Processes the employees DataFrame as per the described steps.

Args:

`employees (pd.DataFrame)`: The input DataFrame with employee details.

Returns:

tuple: A tuple containing:

- A modified DataFrame after all transformations.
- A summary DataFrame with department-level statistics.

"""

```
    modified = (  
        employees.copy()
```

```
.sort_values(by=["department", "salary"], ascending=[False, True])  
.reset_index(drop=True)  
)
```

```
salary_levels = modified["salary"].apply(  
    lambda value: "Low" if value < 50_000 else (  
        "Medium" if value <= 100_000 else "High")  
    )  
modified.insert(len(modified.columns), "salary_level", salary_levels)
```

```
summary = (  
    modified.groupby("department")  
    .agg(total_salary=("salary", "sum"), average_age=("age", "mean"))  
    .reset_index()  
    )
```

```
bonuses = modified["salary"].mul(0.1).round().astype(int)  
modified.insert(len(modified.columns), "bonus", bonuses)  
modified = modified.drop(columns=["age"])
```

```
return modified, summary
```

Quiz Score: 8 out of 10