

Name:- Meghraj Suresh Tolnure

Div:- CS4

Roll No.:- CS4-19

PRN:- 202401040237

Problem Statment 1

Calculate the average rating of papers published in a specific year using Pandas.

Solution:

Import pandas as pd

Import numpy as np

Sample Paper Review dataset (simulated)

Data = {

 'paper_id': [1, 2, 3, 4, 5],

 'rating': [4.5, 3.8, 4.2, 3.5, 4.8],

 'year': [2020, 2020, 2021, 2021, 2020]

}

Df = pd.DataFrame(data)

Filter papers for a specific year (e.g., 2020)

Year_filter = 2020

Papers_in_year = df[df['year'] == year_filter]

```
# Calculate the average rating using Pandas
```

```
Average_rating = papers_in_year['rating'].mean()
```

```
Print(f"Average rating of papers published in {year_filter}:  
{average_rating}")
```

Output (Expected):

Average rating of papers published in 2020: 4.366666666666666

Problem Statement 2

Find the paper with the highest rating and its corresponding paper ID using NumPy.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
# Using the same sample Paper Review dataset
```

```
Data = {
```

```
    'paper_id': [1, 2, 3, 4, 5],
```

```
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],
```

```
    'year': [2020, 2020, 2021, 2021, 2020]
```

```
}
```

```
Df = pd.DataFrame(data)
```

```
# Convert ratings to a NumPy array
```

```
Ratings = np.array(df['rating'])
```

```
# Find the index of the highest rating
```

```
Max_rating_index = np.argmax(ratings)
# Get the corresponding paper ID and rating
Highest_rating = ratings[max_rating_index]
Paper_id_highest = df['paper_id'].iloc[max_rating_index]
Print(f"Paper with the highest rating: Paper ID {paper_id_highest}
with rating {highest_rating}")
```

Output (Expected):

Paper with the highest rating: Paper ID 5 with rating 4.8

Problem Statement 3:

Count the total number of papers reviewed in a specific year using Pandas.

Solution:

Import pandas as pd

Import numpy as np

Data = {

 'paper_id': [1, 2, 3, 4, 5],

 'rating': [4.5, 3.8, 4.2, 3.5, 4.8],

 'year': [2020, 2020, 2021, 2021, 2020]

}

Df = pd.DataFrame(data)

Year_filter = 2020

Total_papers = df[df['year'] == year_filter].shape[0]

```
Print(f"Total number of papers reviewed in {year_filter}:  
{total_papers}")
```

Output (Expected):

Total number of papers reviewed in 2020: 3

Problem Statement 4:

Find the average rating for papers in a specific category using Pandas.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {  
    'paper_id': [1, 2, 3, 4, 5],  
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],  
    'category': ['AI', 'AI', 'ML', 'ML', 'AI']  
}
```

```
Df = pd.DataFrame(data)
```

```
Category_filter = 'AI'
```

```
Avg_rating = df[df['category'] == category_filter]['rating'].mean()
```

```
Print(f"Average rating for papers in {category_filter}: {avg_rating}")
```

Output (Expected):

Average rating for papers in AI: 4.4

Problem Statement 5:

Identify the minimum rating in the dataset using NumPy.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {
```

```
    'paper_id': [1, 2, 3, 4, 5],
```

```
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8]
```

```
}
```

```
Df = pd.DataFrame(data)
```

```
Min_rating = np.min(df['rating'])
```

```
Print(f"Minimum rating in the dataset: {min_rating}")
```

Output (Expected):

Minimum rating in the dataset: 3.5

Problem Statement 6:

Calculate the total number of unique reviewers using Pandas.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {
```

```
    'paper_id': [1, 2, 3, 4, 5],
```

```
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],
```

```
    'reviewer': ['R1', 'R2', 'R1', 'R3', 'R2']
```

```
}
```

```
Df = pd.DataFrame(data)
```

```
Unique_reviewers = df['reviewer'].nunique()
```

```
Print(f"Total number of unique reviewers: {unique_reviewers}")
```

Output (Expected):

Total number of unique reviewers: 3

Problem Statenent 7:

Find the standard deviation of ratings across all papers using NumPy.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {
```

```
    'paper_id': [1, 2, 3, 4, 5],
```

```
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8]
```

```
}
```

```
Df = pd.DataFrame(data)
```

```
Rating_std = np.std(df['rating'])
```

```
Print(f"Standard deviation of ratings: {rating_std:.2f}")
```

Output (Expected):

Standard deviation of ratings: 0.46

Problem Statement 8:

Determine the number of papers reviewed by a specific reviewer using Pandas.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {  
    'paper_id': [1, 2, 3, 4, 5],  
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],  
    'reviewer': ['R1', 'R2', 'R1', 'R3', 'R2']  
}
```

```
Df = pd.DataFrame(data)
```

```
Reviewer_filter = 'R1'
```

```
Papers_by_reviewer = df[df['reviewer'] ==  
reviewer_filter].shape[0]
```

```
Print(f"Number of papers reviewed by {reviewer_filter}:  
{papers_by_reviewer}")
```

Output (Expected):

Number of papers reviewed by R1: 2

Problem Statement G:

Find the paper with the lowest rating in a specific category using Pandas.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {  
    'paper_id': [1, 2, 3, 4, 5],  
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],  
    'category': ['AI', 'AI', 'ML', 'ML', 'AI']  
}
```

```
Df = pd.DataFrame(data)
```

```
Category_filter = 'AI'
```

```
Lowest_rated_paper = df[df['category'] ==  
category_filter].nsmallest(1, 'rating')[['paper_id', 'rating']]
```

```
Print(f"Lowest rated paper in {category_filter}: Paper ID  
{lowest_rated_paper['paper_id'].values[0]} with rating  
{lowest_rated_paper['rating'].values[0]}")
```

Output (Expected):

Lowest rated paper in AI: Paper ID 2 with rating 3.8

Problem Statenent 10:

Calculate the median rating of papers published in a specific year using NumPy.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```



```

Data = {
    'paper_id': [1, 2, 3, 4, 5],
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],
    'year': [2020, 2020, 2021, 2021, 2020]
}
Df = pd.DataFrame(data)
Year_filter = 2020
Ratings_in_year = df[df['year'] == year_filter]['rating']
Median_rating = np.median(ratings_in_year)
Print(f"Median rating of papers published in {year_filter}:
{median_rating}")
Output (Expected):
Median rating of papers published in 2020: 4.5

```

Problem Statenent 11:

Count the number of papers in each category using Pandas.

Solution:

Import pandas as pd

Import numpy as np

```

Data = {
    'paper_id': [1, 2, 3, 4, 5],
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],
    'category': ['AI', 'AI', 'ML', 'ML', 'AI']
}

```

```

}
Df = pd.DataFrame(data)
Category_counts = df['category'].value_counts()
Print(f"Number of papers in each category:\n{category_counts}")
Output (Expected):
Number of papers in each category:
AI    3
ML    2
Name: category, dtype: int64

```

Problem Statement 12:

Find the variance of ratings across all papers using NumPy.

Solution:

```

Import pandas as pd
Import numpy as np
Data = {
    'paper_id': [1, 2, 3, 4, 5],
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8]
}
Df = pd.DataFrame(data)
Rating_variance = np.var(df['rating'])
Print(f"Variance of ratings: {rating_variance:.2f}")
Output (Expected):

```

Variance of ratings: 0.21

Problem Statement 13:

List all papers with a rating above a specific threshold using Pandas.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {
```

```
    'paper_id': [1, 2, 3, 4, 5],
```

```
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8]
```

```
}
```

```
Df = pd.DataFrame(data)
```

```
Threshold = 4.0
```

```
High_rated_papers = df[df['rating'] > threshold]['paper_id'].tolist()
```

```
Print(f"Papers with rating above {threshold}:  
{high_rated_papers}")
```

Output (Expected):

Papers with rating above 4.0: [1, 3, 5]

Problem Statement 14:

Find the average rating given by a specific reviewer using Pandas.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {  
    'paper_id': [1, 2, 3, 4, 5],  
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],  
    'reviewer': ['R1', 'R2', 'R1', 'R3', 'R2']  
}
```

```
Df = pd.DataFrame(data)
```

```
Reviewer_filter = 'R1'
```

```
Avg_rating_reviewer = df[df['reviewer'] ==  
reviewer_filter]['rating'].mean()
```

```
Print(f"Average rating by {reviewer_filter}: {avg_rating_reviewer}")
```

Output (Expected):

Average rating by R1: 4.35

Problem Statenent 15:

Calculate the sum of ratings for papers published in a specific year using Pandas.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {  
    'paper_id': [1, 2, 3, 4, 5],  
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],  
    'year': [2020, 2020, 2021, 2021, 2020]  
}  
Df = pd.DataFrame(data)
```

```
Year_filter = 2020
```

```
Total_rating = df[df['year'] == year_filter]['rating'].sum()
```

```
Print(f"Sum of ratings for papers published in {year_filter}:  
{total_rating}")
```

Output (Expected):

Sum of ratings for papers published in 2020: 13.1

Problem Statment 16:

Find the paper ID with the second-highest rating using NumPy.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {  
    'paper_id': [1, 2, 3, 4, 5],  
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8]
```

```
}  
Df = pd.DataFrame(data)  
Ratings = np.array(df['rating'])  
Second_highest_idx = np.argsort(ratings)[-2]  
Second_highest_paper = df.iloc[second_highest_idx]['paper_id']  
Second_highest_rating = ratings[second_highest_idx]  
Print(f"Paper with the second-highest rating: Paper ID  
{second_highest_paper} with rating {second_highest_rating}")  
Output (Expected):  
Paper with the second-highest rating: Paper ID 1 with rating 4.5
```

Problem Statement 17:

Count the number of papers published in each year using Pandas.

Solution:

Import pandas as pd

Import numpy as np

```
Data = {  
    'paper_id': [1, 2, 3, 4, 5],  
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],  
    'year': [2020, 2020, 2021, 2021, 2020]  
}  
Df = pd.DataFrame(data)
```

```
Year_counts = df['year'].value_counts()
Print(f"Number of papers published in each
year:\n{year_counts}")
```

Output (Expected):

Number of papers published in each year:

2020 3

2021 2

Name: year, dtype: int64

Problem Statement 18:

Find the range of ratings (max - min) in the dataset using NumPy.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {
```

```
    'paper_id': [1, 2, 3, 4, 5],
```

```
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8]
```

```
}
```

```
Df = pd.DataFrame(data)
```

```
Ratings = np.array(df['rating'])
```

```
Rating_range = np.max(ratings) - np.min(ratings)
```

```
Print(f"Range of ratings: {rating_range}")
```

Output (Expected):

Range of ratings: 1.3

Problem Statement 1G:

List all papers in a specific category with a rating below a threshold using Pandas.

Solution:

Import pandas as pd

Import numpy as np

```
Data = {  
    'paper_id': [1, 2, 3, 4, 5],  
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],  
    'category': ['AI', 'AI', 'ML', 'ML', 'AI']  
}
```

```
Df = pd.DataFrame(data)
```

```
Category_filter = 'AI'
```

```
Threshold = 4.0
```

```
Low_rated_papers = df[(df['category'] == category_filter) &  
(df['rating'] < threshold)][['paper_id']].tolist()
```

```
Print(f"Papers in {category_filter} with rating below {threshold}:  
{low_rated_papers}")
```

Output (Expected):

Papers in AI with rating below 4.0: [2]

Problem Statement 20:

Calculate the average rating for each year using Pandas.

Solution:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Data = {  
    'paper_id': [1, 2, 3, 4, 5],  
    'rating': [4.5, 3.8, 4.2, 3.5, 4.8],  
    'year': [2020, 2020, 2021, 2021, 2020]  
}
```

```
Df = pd.DataFrame(data)
```

```
Avg_rating_per_year = df.groupby('year')['rating'].mean()
```

```
Print(f"Average rating for each year:\n{avg_rating_per_year}")
```

Output (Expected):

Average rating for each year:

Year

2020 4.366667

2021 3.850000