

EDS Theory Activity 01

NAME : SHUBHAM NIKWADE

ROLL NO : CS8-48

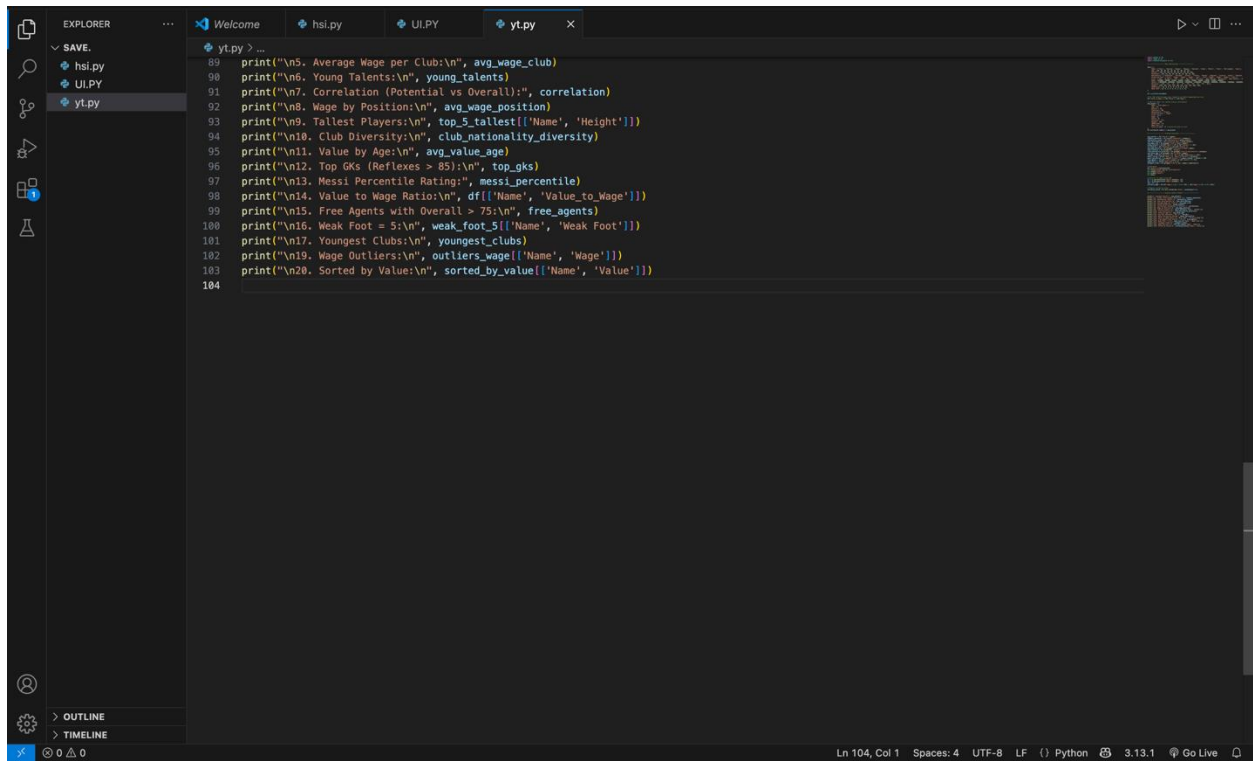
PRN : 202401120033

TOPIC : FIFA DATASET

Code for my dataset and problem statement :

```
EXPLORER  Welcome  hsi.py  UL.PY  yt.py  X
SAVE
  hsi.py
  UL.PY
  yt.py
yt.py > ...
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # ----- Data Definitions -----
6
7 data = {
8     'Name': ['Messi', 'Ronaldo', 'Neymar', 'Mbappe', 'Haaland', 'Foden', 'Modric', 'Pedri', 'Bellingham', 'Saka'],
9     'Age': [34, 36, 29, 24, 22, 23, 37, 20, 19, 21],
10    'Overall': [93, 91, 89, 91, 89, 86, 87, 84, 85, 86],
11    'Potential': [93, 91, 89, 95, 94, 88, 87, 90, 91, 88],
12    'Nationality': ['Argentina', 'Portugal', 'Brazil', 'France', 'Norway', 'England', 'Croatia', 'Spain', 'England', 'England'],
13    'Preferred Foot': ['Left', 'Right', 'Right', 'Right', 'Left', 'Left', 'Right', 'Right', 'Right', 'Left'],
14    'Club': ['PSG', 'Al Nassr', 'PSG', 'PSG', 'Man City', 'Man City', 'Real Madrid', 'Barcelona', 'Real Madrid', 'Arsenal'],
15    'Wage': [560000, 300000, 350000, 400000, 350000, 150000, 180000, 120000, 110000, 100000],
16    'Value': [100000000, 80000000, 95000000, 120000000, 110000000, 75000000, 20000000, 70000000, 75000000, 70000000],
17    'Position': ['RW', 'ST', 'LW', 'ST', 'ST', 'CM', 'CM', 'CM', 'CM', 'RW'],
18    'Height': [170, 187, 175, 178, 195, 171, 172, 174, 185, 176],
19    'GKReflexes': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
20    'Weak Foot': [4, 4, 5, 5, 4, 4, 4, 4, 4, 4]
21 }
22
23 df = pd.DataFrame(data)
24
25 # 14. Add value-to-wage ratio (needs to be before appending new row)
26 df['Value_to_Wage'] = df['Value'] / df['Wage']
27
28 # Add Free Agent row (safely using a dictionary)
29 new_player = {
30     'Name': 'Free Agent 1',
31     'Age': 27,
32     'Overall': 78,
33     'Potential': 80,
34     'Nationality': 'France',
35     'Preferred Foot': 'Right',
36     'Club': None,
37     'Wage': 0,
38     'Value': 0,
39     'Position': 'CM',
40     'Height': 180,
41     'GKReflexes': 0,
42     'Weak Foot': 4,
43     'Value_to_Wage': 0 # Avoid division by zero
44 }
45 df.loc[len(df.index)] = new_player
46
```

```
EXPLORER  Welcome  hsi.py  UL.PY  yt.py  X
yt.py > ...
46
47 # ----- Problem Solutions -----
48
49 avg_overall = df['Overall'].mean()
50 highest_potential = df.loc[df['Potential'].idxmax()]
51 nationality_counts = df['Nationality'].value_counts()
52 foot_distribution = df['Preferred Foot'].value_counts()
53 avg_wage_club = df.groupby('Club')['Wage'].mean()
54 young_talents = df[(df['Age'] < 21) & (df['Overall'] > 80)]
55 correlation = df['Potential'].corr(df['Overall'])
56 avg_wage_position = df.groupby('Position')['Wage'].mean()
57 top_5_tallest = df.nlargest(5, 'Height')
58 club_nationality_diversity = df.groupby('Club')['Nationality'].nunique()
59 avg_value_age = df.groupby('Age')['Value'].mean()
60 top_gks = df[(df['Position'] == 'GK') & (df['GKReflexes'] > 85)]
61 messi_rating = df[df['Name'] == 'Messi']['Overall'].values[0]
62 messi_percentile = np.sum(df['Overall'] < messi_rating) / len(df) * 100
63 free_agents = df[(df['Club'].isna()) & (df['Overall'] > 75)]
64 weak_foot_5 = df[df['Weak Foot'] == 5]
65 youngest_clubs = df.groupby('Club')['Age'].mean().nsmallest(5)
66
67 # Histogram
68 df['Overall'].hist(bins=10)
69 plt.title("Player Ratings Distribution")
70 plt.xlabel("Overall")
71 plt.ylabel("Count")
72 plt.show()
73
74 # Outliers in wages using IQR
75 Q1 = np.percentile(df['Wage'], 25)
76 Q3 = np.percentile(df['Wage'], 75)
77 IQR = Q3 - Q1
78 outliers_wage = df[(df['Wage'] < Q1 - 1.5 * IQR) | (df['Wage'] > Q3 + 1.5 * IQR)]
79
80 # Players sorted by value
81 sorted_by_value = df.sort_values(by='Value', ascending=False)
82
83 # ----- Display Sample Outputs -----
84
85 print(f"1. Average Overall: {avg_overall}")
86 print(f"\n2. Player with Highest Potential:\n", highest_potential)
87 print(f"\n3. Nationality Counts:\n", nationality_counts)
88 print(f"\n4. Foot Distribution:\n", foot_distribution)
89 print(f"\n5. Average Wage per Club:\n", avg_wage_club)
90 print(f"\n6. Young Talents:\n", young_talents)
91 print(f"\n7. Correlation (Potential vs Overall):", correlation)
```



Problem Statement and their output :

1)Average:

```
(base) shubham@shubhams-MacBook-Pro: ~ % python  
1. Average Overall: 87.181818181819
```

2)Player with Highest :

```
2. Player with Highest Potential:  
Name           Mbappe  
Age             24  
Overall         91  
Potential       95  
Nationality     France  
Preferred Foot  Right  
Club            PSG  
Wage            400000  
Value           120000000  
Position       ST  
Height          178  
GKReflexes      0  
Weak Foot       5  
Value_to_Wage   300.0  
Name: 3, dtype: object
```

3)Nationality Counts :

```
3. Nationality Counts:  
Nationality  
England      3  
France       2  
Argentina    1  
Portugal     1  
Brazil       1  
Norway       1  
Croatia      1  
Spain        1  
Name: count, dtype: int64
```

4) Foot Distribution :

```
4. Foot Distribution:
Preferred Foot
Right    7
Left     4
Name: count, dtype: int64
```

5) Average Wage per Club:

```
5. Average Wage per Club:
Club
Al Nassr      300000.000000
Arsenal       100000.000000
Barcelona     120000.000000
Man City      250000.000000
PSG           436666.666667
Real Madrid   145000.000000
Name: Wage, dtype: float64
```

6) Young Talents :

```
6. Young Talents:
   Name  Age  Overall  Potential  Nationality Preferred Foot   Club   Wage   Value  Position  Height  GKReflexes  Weak Foot  Value_to_Wage
7  Pedri   20    84      90        Spain         Right  Barcelona  120000  70000000    CM    174         0         4      583.333333
8 Bellingham 19    85      91        England        Right   Real Madrid  110000  75000000    CM    185         0         4      681.818182
```

7) Correlation :

```
7. Correlation (Potential vs Overall): 0.8273991996471627
```

8) Wage by Position:

```
8. Wage by Position:
Position
CM      112000.0
LW      350000.0
RW      330000.0
ST      350000.0
Name: Wage, dtype: float64
```

9) Tallest Players:

```
9. Tallest Players:
      Name  Height
4      Haaland    195
1      Ronaldo    187
8      Bellingham  185
10     Free Agent 1  180
3      Mbappe     178
```

10) Club Diversity:

```
10. Club Diversity:
      Club
Al Nassr      1
Arsenal       1
Barcelona     1
Man City      2
PSG           3
Real Madrid   2
Name: Nationality, dtype: int64
```

11) Value by Age :

```
11. Value by Age:
      Age
19      75000000.0
20      70000000.0
21      70000000.0
22     110000000.0
23      75000000.0
24     120000000.0
27           0.0
29      95000000.0
34     100000000.0
36      80000000.0
37      20000000.0
Name: Value, dtype: float64
```

12) Top GKs (Reflexes > 85):

```
12. Top GKs (Reflexes > 85):  
Empty DataFrame  
Columns: [Name, Age, Overall, Potential, Nationality, Preferred Foot, Club, Wage, Value, Position, Height, GKReflexes, Weak Foot, Value_to_Wage]  
Index: []
```

13) Messi Percentile Rating :

```
13. Messi Percentile Rating: 90.9090909090909
```

14) Value to Wage Ratio :

```
14. Value to Wage Ratio:  
      Name  Value_to_Wage  
0      Messi      178.571429  
1     Ronaldo      266.666667  
2     Neymar      271.428571  
3      Mbappe      300.000000  
4     Haaland      314.285714  
5       Foden      500.000000  
6      Modric      111.111111  
7       Pedri      583.333333  
8  Bellingham      681.818182  
9        Saka      700.000000  
10 Free Agent 1         0.000000
```

15) Free Agents with Overall > 75:

```
15. Free Agents with Overall > 75:  
      Name  Age  Overall  Potential  Nationality  Preferred Foot  Club  Wage  Value  Position  Height  GKReflexes  Weak Foot  Value_to_Wage  
10 Free Agent 1  27    78      80      France      Right  None      0      0      CM      180      0      4      0.0
```

16) Weak Foot = 5:

```
16. Weak Foot = 5:  
      Name  Weak Foot  
2  Neymar      5  
3  Mbappe      5
```

17) Youngest Clubs:

```
17. Youngest Clubs:
Club
Barcelona      20.0
Arsenal        21.0
Man City       22.5
Real Madrid    28.0
PSG            29.0
Name: Age, dtype: float64
```

18) Wage Outliers :

```
19. Wage Outliers:
Empty DataFrame
Columns: [Name, Wage]
Index: []
```

19) Sorted by Value:

```
20. Sorted by Value:
      Name      Value
3      Mbappe 120000000
4      Haaland 110000000
0      Messi  100000000
2      Neymar  95000000
1      Ronaldo 80000000
5      Foden  75000000
8      Bellingham 75000000
7      Pedri  70000000
9      Saka   70000000
6      Modric 20000000
10 Free Agent 1      0
```