MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Assignment-1

NAME : Palak Sunil Surana

ROLL NO : CS8-47

PRN : 202401120009

TOPIC : FIFA DATASET

Code for my dataset and problem statement :

EXPLORER

SAVE.
- hsi.py
- UI.PY
- yt.py

Welcome | hsi.py | UI.PY | yt.py

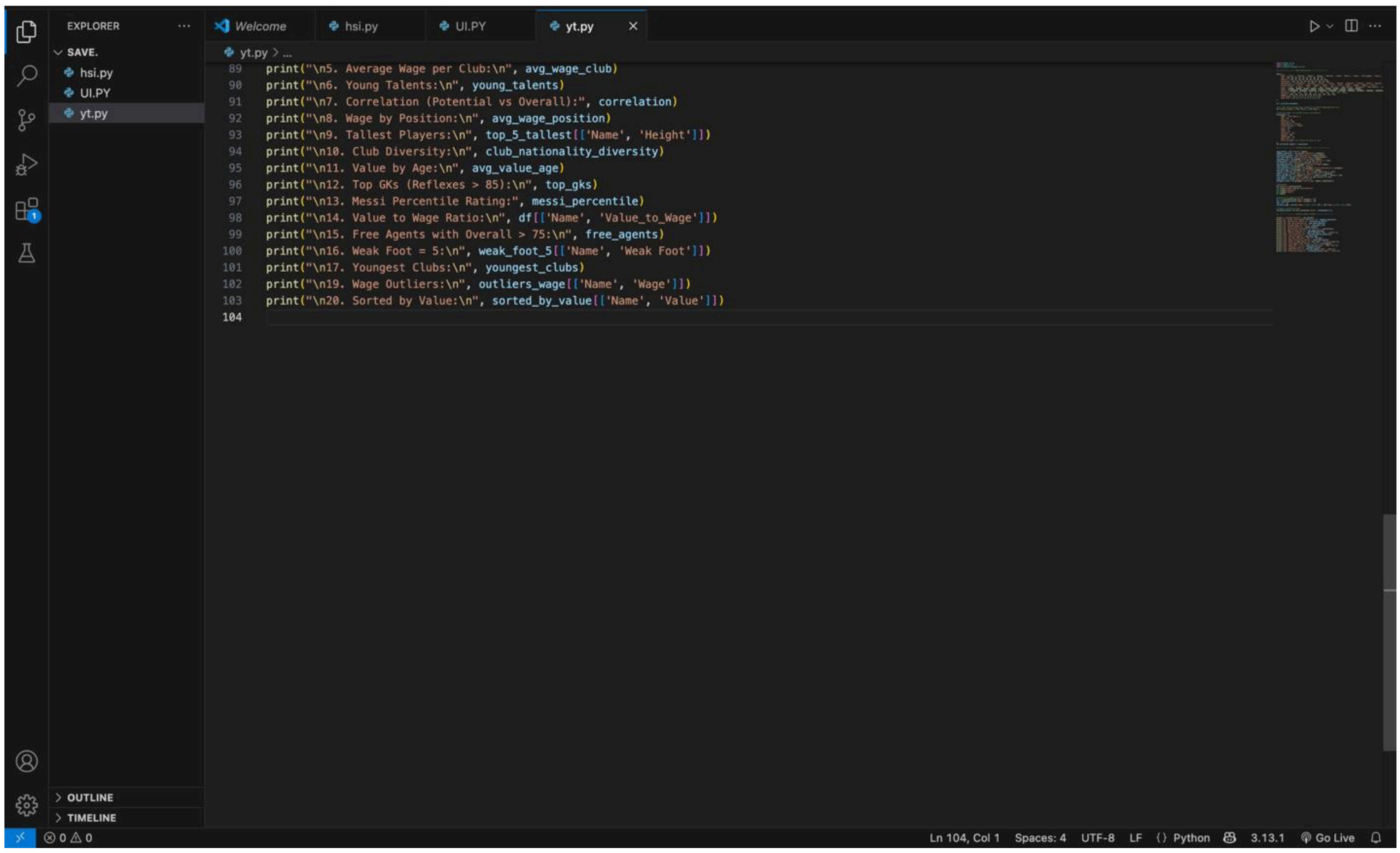
yt.py > ...

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# ------------------ Data Definitions ------------------

data = {
    'Name': ['Messi', 'Ronaldo', 'Neymar', 'Mbappe', 'Haaland', 'Foden', 'Modric', 'Pedri', 'Bellingham', 'Saka'],
    'Age': [34, 36, 29, 24, 22, 23, 37, 20, 19, 21],
    'Overall': [93, 91, 89, 91, 89, 86, 87, 84, 85, 86],
    'Potential': [93, 91, 89, 95, 94, 88, 87, 90, 91, 88],
    'Nationality': ['Argentina', 'Portugal', 'Brazil', 'France', 'Norway', 'England', 'Croatia', 'Spain', 'England', 'England'],
    'Preferred Foot': ['Left', 'Right', 'Right', 'Right', 'Left', 'Left', 'Right', 'Right', 'Right', 'Left'],
    'Club': ['PSG', 'Al Nassr', 'PSG', 'PSG', 'Man City', 'Man City', 'Real Madrid', 'Barcelona', 'Real Madrid', 'Arsenal'],
    'Wage': [560000, 300000, 350000, 400000, 350000, 150000, 180000, 120000, 110000, 100000],
    'Value': [100000000, 80000000, 95000000, 120000000, 110000000, 75000000, 20000000, 70000000, 75000000, 70000000],
    'Position': ['RW', 'ST', 'LW', 'ST', 'ST', 'CM', 'CM', 'CM', 'CM', 'RW'],
    'Height': [170, 187, 175, 178, 195, 171, 172, 174, 185, 176],
    'GKReflexes': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    'Weak Foot': [4, 4, 5, 5, 4, 4, 4, 4, 4, 4]
}

df = pd.DataFrame(data)

# 14. Add value-to-wage ratio (needs to be before appending new row)
df['Value_to_Wage'] = df['Value'] / df['Wage']

# Add Free Agent row (safely using a dictionary)
new_player = {
    'Name': 'Free Agent 1',
    'Age': 27,
    'Overall': 78,
    'Potential': 80,
    'Nationality': 'France',
    'Preferred Foot': 'Right',
    'Club': None,
    'Wage': 0,
    'Value': 0,
    'Position': 'CM',
    'Height': 180,
    'GKReflexes': 0,
    'Weak Foot': 4,
    'Value_to_Wage': 0  # Avoid division by zero
}
df.loc[len(df.index)] = new_player

# ------------------ Problem Solutions ------------------

avg_overall = df['Overall'].mean()
highest_potential = df.loc[df['Potential'].idxmax()]
nationality_counts = df['Nationality'].value_counts()
foot_distribution = df['Preferred Foot'].value_counts()
avg_wage_club = df.groupby('Club')['Wage'].mean()
young_talents = df[(df['Age'] < 21) & (df['Overall'] > 80)]
correlation = df['Potential'].corr(df['Overall'])
avg_wage_position = df.groupby('Position')['Wage'].mean()
top_5_tallest = df.nlargest(5, 'Height')
club_nationality_diversity = df.groupby('Club')['Nationality'].nunique()
avg_value_age = df.groupby('Age')['Value'].mean()
top_gks = df[(df['Position'] == 'GK') & (df['GKReflexes'] > 85)]
messi_rating = df[df['Name'] == 'Messi']['Overall'].values[0]
messi_percentile = np.sum(df['Overall'] < messi_rating) / len(df) * 100
free_agents = df[(df['Club'].isna()) & (df['Overall'] > 75)]
weak_foot_5 = df[df['Weak Foot'] == 5]
youngest_clubs = df.groupby('Club')['Age'].mean().nsmallest(5)

# Histogram
df['Overall'].hist(bins=10)
plt.title("Player Ratings Distribution")
plt.xlabel("Overall")
plt.ylabel("Count")
plt.show()

# Outliers in wages using IQR
Q1 = np.percentile(df['Wage'].dropna(), 25)
Q3 = np.percentile(df['Wage'].dropna(), 75)
IQR = Q3 - Q1
outliers_wage = df[(df['Wage'] < Q1 - 1.5 * IQR) | (df['Wage'] > Q3 + 1.5 * IQR)]

# Players sorted by value
sorted_by_value = df.sort_values(by='Value', ascending=False)

# ------------------ Display Sample Outputs ------------------

print("1. Average Overall:", avg_overall)
print("\n2. Player with Highest Potential:\n", highest_potential)
print("\n3. Nationality Counts:\n", nationality_counts)
print("\n4. Foot Distribution:\n", foot_distribution)
print("\n5. Average Wage per Club:\n", avg_wage_club)
print("\n6. Young Talents:\n", young_talents)
print("\n7. Correlation (Potential vs Overall):", correlation)
```

OUTLINE

TIMELINE

0 ⚠ 0 — Ln 104, Col 1 — Spaces: 4 — UTF-8 — LF — Python — 3.13.1 — Go Live



Problem Statement and their output :

1) Average:

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
```