```
import pandas as pd
import numpy as np

#1. Loading the dataset onto a DataFrame

df = pd.read_csv('IPL IMB381IPL2013.csv')

#2. Displaying the first few records of the DataFrame

df.head()
```

```
   Sl.NO.   PLAYER NAME  AGE COUNTRY  TEAM PLAYING ROLE  T-RUNS  T-
WKTS  \
0       1   Abdulla, YA    2      SA  KXIP   Allrounder       0
0
1       2  Abdur Razzak    2     BAN   RCB       Bowler     214
18
2       3   Agarkar, AB    2     IND   KKR       Bowler     571
58
3       4     Ashwin, R    1     IND   CSK       Bowler     284
31
4       5  Badrinath, S    2     IND   CSK      Batsman      63
0

   ODI-RUNS-S  ODI-SR-B  ...    SR-B  SIXERS  RUNS-C  WKTS  AVE-BL
ECON  \
0           0      0.00  ...    0.00       0     307    15   20.47
8.90
1         657     71.41  ...    0.00       0      29     0    0.00
14.50
2        1269     80.62  ...  121.01       5    1059    29   36.52
8.81
3         241     84.56  ...   76.32       0    1125    49   22.96
6.23
4          79     45.93  ...  120.71      28       0     0    0.00
0.00

   SR-BL  AUCTION YEAR  BASE PRICE  SOLD PRICE
0  13.93          2009       50000       50000
1   0.00          2008       50000       50000
2  24.90          2008      200000      350000
3  22.14          2011      100000      850000
4   0.00          2011      100000      800000

[5 rows x 26 columns]
```

```
#3. Finding metadata of the DataFrame

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 130 entries, 0 to 129
Data columns (total 26 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Sl.NO.          130 non-null    int64
 1   PLAYER NAME     130 non-null    object
 2   AGE             130 non-null    int64
 3   COUNTRY         130 non-null    object
 4   TEAM            130 non-null    object
 5   PLAYING ROLE    130 non-null    object
 6   T-RUNS          130 non-null    int64
 7   T-WKTS          130 non-null    int64
 8   ODI-RUNS-S      130 non-null    int64
 9   ODI-SR-B        130 non-null    float64
 10  ODI-WKTS        130 non-null    int64
 11  ODI-SR-BL       130 non-null    float64
 12  CAPTAINCY EXP   130 non-null    int64
 13  RUNS-S          130 non-null    int64
 14  HS              130 non-null    int64
 15  AVE             130 non-null    float64
 16  SR-B            130 non-null    float64
 17  SIXERS          130 non-null    int64
 18  RUNS-C          130 non-null    int64
 19  WKTS            130 non-null    int64
 20  AVE-BL          130 non-null    float64
 21  ECON            130 non-null    float64
 22  SR-BL           130 non-null    float64
 23  AUCTION YEAR    130 non-null    int64
 24  BASE PRICE      130 non-null    int64
 25  SOLD PRICE      130 non-null    int64
dtypes: float64(7), int64(15), object(4)
memory usage: 26.5+ KB
```

#4. Finding Summary of the DataFrame

```
df.describe()
```

```
           Sl.NO.         AGE        T-RUNS      T-WKTS    ODI-RUNS-S
\
count  130.000000  130.000000    130.000000  130.000000    130.000000

mean    65.500000    2.092308   2166.715385   66.530769   2508.738462

std     37.671829    0.576627   3305.646757  142.676855   3582.205625

min      1.000000    1.000000      0.000000    0.000000      0.000000

25%     33.250000    2.000000     25.500000    0.000000     73.250000
```

```
50%     65.500000    2.000000    542.500000    7.000000    835.000000

75%     97.750000    2.000000   3002.250000   47.500000   3523.500000

max    130.000000    3.000000  15470.000000  800.000000  18426.000000


          ODI-SR-B    ODI-WKTS   ODI-SR-BL  CAPTAINCY EXP       RUNS-S
...  \
count  130.000000  130.000000  130.000000     130.000000   130.000000
...
mean    71.164385   76.076923   34.033846       0.315385   514.246154
...
std     25.898440  111.205070   26.751749       0.466466   615.226335
...
min      0.000000    0.000000    0.000000       0.000000     0.000000
...
25%     65.650000    0.000000    0.000000       0.000000    39.000000
...
50%     78.225000   18.500000   36.600000       0.000000   172.000000
...
75%     86.790000  106.000000   45.325000       1.000000   925.250000
...
max    116.660000  534.000000  150.000000       1.000000  2254.000000
...

             SR-B      SIXERS       RUNS-C        WKTS      AVE-BL  \
count  130.000000  130.000000   130.000000  130.000000  130.000000
mean   111.053462   17.692308   475.523077   17.169231   23.110231
std     35.928907   23.828146   558.314049   21.816763   20.802057
min      0.000000    0.000000     0.000000    0.000000    0.000000
25%     98.237500    1.000000     0.000000    0.000000    0.000000
50%    118.510000    6.000000   297.000000    8.500000   24.785000
75%    129.102500   29.750000   689.250000   23.750000   35.580000
max    235.490000  129.000000  1975.000000   83.000000  126.300000

             ECON       SR-BL  AUCTION YEAR    BASE PRICE    SOLD
PRICE
count  130.000000  130.000000    130.000000  1.300000e+02
1.300000e+02
mean     6.204462   17.382615   2009.092308  1.922308e+05
5.212231e+05
std      4.941531   15.273422      1.377821  1.530973e+05
4.068074e+05
min      0.000000    0.000000   2008.000000  2.000000e+04
2.000000e+04
25%      0.000000    0.000000   2008.000000  1.000000e+05
2.250000e+05
50%      7.380000   19.935000   2008.000000  2.000000e+05
4.375000e+05
```

```
75%      8.247500   26.212500   2011.000000  2.250000e+05
7.000000e+05
max     38.110000  100.200000   2011.000000  1.350000e+06
1.800000e+06

[8 rows x 22 columns]
```

```
#5. Slicing and Indexing a DataFrame

df.loc[0:4, ['PLAYER NAME', 'AGE']]
```

```
    PLAYER NAME  AGE
0   Abdulla, YA    2
1  Abdur Razzak    2
2   Agarkar, AB    2
3     Ashwin, R    1
4  Badrinath, S    2
```

```
#6. Selecting Columns by Column Names

df[['PLAYER NAME', 'AGE', 'COUNTRY']]
```

```
      PLAYER NAME  AGE COUNTRY
0     Abdulla, YA    2      SA
1    Abdur Razzak    2     BAN
2     Agarkar, AB    2     IND
3       Ashwin, R    1     IND
4    Badrinath, S    2     IND
..            ...  ...     ...
125     Yadav, AS    2     IND
126   Younis Khan    2     PAK
127  Yuvraj Singh    2     IND
128   Zaheer Khan    2     IND
129    Zoysa, DNT    2      SL

[130 rows x 3 columns]
```

```
#7. Finding Unique Occurrences of Values in Columns

df['COUNTRY'].unique()

df['COUNTRY'].nunique()

10
```

```
#8. Cross-tabulation between two columns

pd.crosstab(df['AGE'], df['PLAYING ROLE'])
```

```
PLAYING ROLE  Allrounder  Batsman  Bowler  W. Keeper
AGE
1                      4        5       7          0
```

```
2                    25       21      29          11
3                     6       13       8           1
```

```
#9. Sorting DataFrame by Column Values

df.sort_values(by='SOLD PRICE', ascending=False)
```

```
     Sl.NO.    PLAYER NAME  AGE COUNTRY   TEAM PLAYING ROLE  T-RUNS
T-WKTS  \
93       94      Sehwag, V    2     IND     DD       Batsman    8178
40
127     128   Yuvraj Singh    2     IND  KXIP+       Batsman    1775
9
50       51       Kohli, V    1     IND    RCB       Batsman     491
0
111     112  Tendulkar, SR    3     IND     MI       Batsman   15470
45
113     114     Tiwary, SS    1     IND    MI+       Batsman       0
0
..      ...            ...  ...     ...    ...           ...     ...
...
34       35  Henriques, MC    1     AUS   KKR+   Allrounder       0
0
5         6     Bailey, GJ    2     AUS    CSK       Batsman       0
0
0         1    Abdulla, YA    2      SA   KXIP   Allrounder       0
0
46       47    Kamran Khan    1     IND    RR+       Bowler       0
0
73       74     Noffke, AA    2     AUS    RCB   Allrounder       0
0

     ODI-RUNS-S  ODI-SR-B  ...    SR-B  SIXERS  RUNS-C  WKTS  AVE-BL
ECON  \
93         8090    104.68  ...  167.32      79     226     6   37.67
10.56
127        8051     87.58  ...  131.88      67     569    23   24.74
7.02
50         3590     86.31  ...  119.29      49     345     4   86.25
8.84
111       18426     86.23  ...  119.22      24      58     0    0.00
9.67
113          49     87.50  ...  119.60      32       0     0    0.00
0.00
..          ...       ...  ...     ...     ...     ...   ...     ...
...
34           18     60.00  ...  108.89       1     142     3   47.33
8.82
5           172     72.26  ...   95.45       0       0     0    0.00
0.00
```

```
0              0      0.00  ...    0.00       0     307    15   20.47
8.90
46             0      0.00  ...   60.00       0     224     9   24.89
8.48
73             0      0.00  ...   90.00       0      40     1   40.00
10.00

     SR-BL  AUCTION YEAR  BASE PRICE  SOLD PRICE
93   21.67          2011      400000     1800000
127  21.13          2011      400000     1800000
50   58.50          2011      150000     1800000
111   0.00          2011      400000     1800000
113   0.00          2011      100000     1600000
..     ...           ...         ...         ...
34   32.33          2011       50000       50000
5     0.00          2009       50000       50000
0    13.93          2009       50000       50000
46   17.78          2009       20000       24000
73   24.00          2010       20000       20000

[130 rows x 26 columns]
```

#10. Which Player Got the Maximum Premium on the Base Price?

```
df['PREMIUM'] = df['SOLD PRICE'] - df['BASE PRICE']

df.loc[df['PREMIUM'].idxmax(), 'PLAYER NAME']
```

```
'Kohli, V'
```

#11. Which Players Got the Maximum Premium Offering on Their Base Price?

```
max_premium = df['PREMIUM'].max()
df[df['PREMIUM'] == max_premium]
```

```
    Sl.NO. PLAYER NAME  AGE COUNTRY TEAM PLAYING ROLE  T-RUNS  T-WKTS
\
50      51    Kohli, V    1     IND  RCB      Batsman     491       0


    ODI-RUNS-S  ODI-SR-B  ...  SIXERS  RUNS-C  WKTS  AVE-BL  ECON  SR-
BL  \
50        3590     86.31  ...      49     345     4   86.25  8.84
58.5

    AUCTION YEAR  BASE PRICE  SOLD PRICE  PREMIUM
50          2011      150000     1800000  1650000

[1 rows x 27 columns]
```

```
#12. What is the Average SOLD PRICE for Each Age Category?

df.groupby('AGE')['SOLD PRICE'].mean()
```

```
AGE
1    720250.000000
2    484534.883721
3    520178.571429
Name: SOLD PRICE, dtype: float64
```

```
#13. Average SOLD PRICE for Different Playing Roles in Each Age Category

df.groupby(['AGE', 'PLAYING ROLE'])['SOLD PRICE'].mean()
```

```
AGE  PLAYING ROLE
1    Allrounder      5.875000e+05
     Batsman         1.110000e+06
     Bowler          5.177143e+05
2    Allrounder      4.494000e+05
     Batsman         6.547619e+05
     Bowler          3.979310e+05
     W. Keeper       4.677273e+05
3    Allrounder      7.666667e+05
     Batsman         4.576923e+05
     Bowler          4.143750e+05
     W. Keeper       7.000000e+05
Name: SOLD PRICE, dtype: float64
```