

	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								
..
...								
125	0	0.00	...	125.64	2	0	0	0.00
0.00								
126	6814	75.78	...	42.85	0	0	0	0.00
0.00								
127	8051	87.58	...	131.88	67	569	23	24.74
7.02								
128	790	73.55	...	91.67	1	1783	65	27.43
7.75								
129	343	95.81	...	122.22	0	99	2	49.50
9.00								

	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
..
125	0.00		2010	50000	750000
126	0.00		2008	225000	225000
127	21.13		2011	400000	1800000
128	21.26		2008	200000	450000
129	33.00		2008	100000	110000

[130 rows x 26 columns]

Displaying first few records of the DataFrame

```
df.head(20)
```

	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								
5	6	Bailey, GJ	2	AUS	CSK	Batsman	0	
0								
6	7	Balaji, L	2	IND	CSK+	Bowler	51	
27								
7	8	Bollinger, DE	2	AUS	CSK	Bowler	54	
50								
8	9	Botha, J	2	SA	RR	Allrounder	83	
17								
9	10	Boucher, MV	2	SA	RCB+	W. Keeper	5515	
1								
10	11	Bravo, DJ	2	WI	MI+	Allrounder	2200	
86								
11	12	Chanderpaul, S	3	WI	RCB	Batsman	9918	
9								
12	13	Chawla, PP	1	IND	KXIP	Allrounder	5	
3								
13	14	de Villiers, AB	2	SA	DD+	W. Keeper	5457	
2								
14	15	Dhawan, S	2	IND	MI+	Batsman	0	
0								
15	16	Dhoni, MS	2	IND	CSK	W. Keeper	3509	
0								
16	17	Dilshan, TM	2	SL	DD+	Allrounder	4722	
32								
17	18	Dinda, AB	2	IND	KKR+	Bowler	0	
0								
18	19	Dravid, RS	3	IND	RCB+	Batsman	13288	
1								
19	20	Duminy, J-P	2	SA	MI+	Batsman	654	
11								
	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								
5	172	72.26	...	95.45	0	0	0	0.00

0.00								
6	120	78.94	...	72.22	1	1342	52	25.81
7.98								
7	50	92.59	...	165.88	1	693	37	18.73
7.22								
8	609	85.77	...	114.73	3	610	19	32.11
6.85								
9	4686	84.76	...	127.51	13	0	0	0.00
0.00								
10	2004	81.39	...	127.12	38	1338	47	28.47
8.12								
11	8778	70.74	...	80.64	0	0	0	0.00
0.00								
12	38	65.51	...	113.09	9	1819	73	126.30
38.11								
13	4998	93.19	...	128.53	42	0	0	0.00
0.00								
14	69	56.09	...	122.32	36	66	4	16.50
8.25								
15	6773	88.19	...	136.45	64	0	0	0.00
0.00								
16	6455	86.80	...	117.83	24	356	5	71.20
8.07								
17	18	60.00	...	33.33	0	926	36	25.72
7.29								
18	10889	71.24	...	116.88	23	0	0	0.00
0.00								
19	2536	84.00	...	119.27	35	377	10	37.70
7.11								

	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	0.00		2009		50000		50000
6	19.40		2011		100000		500000
7	15.57		2011		200000		700000
8	28.11		2011		200000		950000
9	0.00		2008		200000		450000
10	21.11		2011		200000		200000
11	0.00		2008		200000		200000
12	100.20		2008		125000		400000
13	0.00		2008		200000		300000
14	12.00		2011		100000		300000
15	0.00		2008		400000		1500000
16	53.00		2008		150000		250000
17	21.19		2011		100000		375000

18	0.00	2011	400000	500000
19	31.80	2009	300000	300000

[20 rows x 26 columns]

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
```

```
column = df.columns
```

```
column
```

```
Index(['Sl.NO.', 'PLAYER NAME', 'AGE', 'COUNTRY', 'TEAM', 'PLAYING  
ROLE',
```

```
      'T-RUNS', 'T-WKTS', 'ODI-RUNS-S', 'ODI-SR-B', 'ODI-WKTS', 'ODI-  
SR-BL',
```

```

    'CAPTAINCY EXP', 'RUNS-S', 'HS', 'AVE', 'SR-B', 'SIXERS',
    'RUNS-C',
    'WKTS', 'AVE-BL', 'ECON', 'SR-BL', 'AUCTION YEAR', 'BASE
    PRICE',
    'SOLD PRICE'],
    dtype='object')

```

```
df.shape
```

```
(130, 26)
```

```
print(df.nunique())
```

```

Sl.NO.      130
PLAYER NAME  130
AGE          3
COUNTRY      10
TEAM         17
PLAYING ROLE  4
T-RUNS      103
T-WKTS       60
ODI-RUNS-S   117
ODI-SR-B     118
ODI-WKTS     74
ODI-SR-BL    82
CAPTAINCY EXP  2
RUNS-S       115
HS           73
AVE          113
SR-B         125
SIXERS       48
RUNS-C       92
WKTS         51
AVE-BL       88
ECON         83
SR-BL        82
AUCTION YEAR  4
BASE PRICE   17
SOLD PRICE   53

```

```
dtype: int64
```

```
df.isnull().sum()
```

```

Sl.NO.      0
PLAYER NAME  0
AGE          0
COUNTRY      0
TEAM         0
PLAYING ROLE  0
T-RUNS       0
T-WKTS       0

```

```

ODI-RUNS-S      0
ODI-SR-B        0
ODI-WKTS        0
ODI-SR-BL       0
CAPTAINCY EXP   0
RUNS-S          0
HS              0
AVE             0
SR-B           0
SIXERS          0
RUNS-C          0
WKTS            0
AVE-BL          0
ECON            0
SR-BL           0
AUCTION YEAR    0
BASE PRICE      0
SOLD PRICE      0
dtype: int64

```

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
...

```

```

count      SR-B      SIXERS      RUNS-C      WKTS      AVE-BL \
mean      130.000000     130.000000     130.000000     130.000000     130.000000
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

```

```

count      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]
```

Slicing and Indexing a dataframe

```
df.iloc[:,0:16]
```

```

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							
..
...							
125	126	Yadav, AS	2	IND	DC	Batsman	0
0							
126	127	Younis Khan	2	PAK	RR	Batsman	6398
7							
127	128	Yuvraj Singh	2	IND	KXIP+	Batsman	1775
9							
128	129	Zaheer Khan	2	IND	MI+	Bowler	1114
288							
129	130	Zoysa, DNT	2	SL	DC	Bowler	288
64							
	ODI-RUNS-S	ODI-SR-B	ODI-WKTS	ODI-SR-BL	CAPTAINCY	EXP	RUNS-S
HS \							
0	0	0.00	0	0.0		0	0
0							
1	657	71.41	185	37.6		0	0
0							
2	1269	80.62	288	32.9		0	167
39							
3	241	84.56	51	36.8		0	58
11							
4	79	45.93	0	0.0		0	1317
71							
..
..							
125	0	0.00	0	0.0		0	49
16							
126	6814	75.78	3	86.6		1	3
3							
127	8051	87.58	109	44.3		1	1237
66							
128	790	73.55	278	35.4		0	99
23							
129	343	95.81	108	39.4		0	11
10							
	AVE						
0	0.00						

```
1      0.00
2     18.56
3      5.80
4     32.93
...
125    9.80
126    3.00
127   26.32
128    9.90
129   11.00
```

```
[130 rows x 16 columns]
```

Selecting Columns by Column Names (eg: PLAYER NAME)

```
df['PLAYER NAME']
```

```
0      Abdulla, YA
1     Abdur Razzak
2      Agarkar, AB
3      Ashwin, R
4     Badrinath, S
...
125     Yadav, AS
126    Younis Khan
127    Yuvraj Singh
128     Zaheer Khan
129     Zoysa, DNT
```

```
Name: PLAYER NAME, Length: 130, dtype: object
```

Finding Unique Occurrences of Values in Columns (eg: COUNTRY)

```
df['COUNTRY'].unique()
```

```
array(['SA', 'BAN', 'IND', 'AUS', 'WI', 'SL', 'NZ', 'ENG', 'PAK',
       'ZIM'],
      dtype=object)
```

```
df['COUNTRY'].value_counts()
```

```
COUNTRY
IND      53
AUS      22
SA        16
SL        12
PAK        9
NZ         7
WI         6
```

```
ENG      3
BAN      1
ZIM      1
Name: count, dtype: int64
```

Cross-tabulation between two columns (eg: AGE & PLAYING ROLE)

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

PLAYING ROLE	Allrounder	Batsman	Bowler	W. Keeper
AGE				
1	4	5	7	0
2	25	21	29	11
3	6	13	8	1

Sorting dataframe by column values (eg: SOLD PRICE)

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

SL.NO.	PLAYER NAME	AGE	COUNTRY	TEAM	PLAYING ROLE	T-RUNS		
9340	Sehwag, V	2	IND	DD	Batsman	8178		
1279	Yuvraj Singh	2	IND	KXIP+	Batsman	1775		
500	Kohli, V	1	IND	RCB	Batsman	491		
11145	Tendulkar, SR	3	IND	MI	Batsman	15470		
1130	Tiwary, SS	1	IND	MI+	Batsman	0		
...		
340	Henriques, MC	1	AUS	KKR+	Allrounder	0		
500	Bailey, GJ	2	AUS	CSK	Batsman	0		
00	Abdulla, YA	2	SA	KXIP	Allrounder	0		
460	Kamran Khan	1	IND	RR+	Bowler	0		
730	Noffke, AA	2	AUS	RCB	Allrounder	0		
	ODI-RUNS-S	ODI-SR-B	...	SR-B	SIXERS	RUNS-C	WKTS	AVE-BL

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]

Which player got the maximum premium on the base price?

```
df["Premium"] = df["SOLD PRICE"] - df["BASE PRICE"]
df["Premium"]
```

0	0
1	0
2	150000
3	750000
4	700000

```

125      ...
126      700000
127      0
128      1400000
129      250000
130      10000
Name: Premium, Length: 130, dtype: int64

High_premium_player_name = df.loc[df['Premium'].idxmax()]
High_premium_player_name

Sl.NO.      51
PLAYER NAME      Kohli, V
AGE      1
COUNTRY      IND
TEAM      RCB
PLAYING ROLE      Batsman
T-RUNS      491
T-WKTS      0
ODI-RUNS-S      3590
ODI-SR-B      86.31
ODI-WKTS      2
ODI-SR-BL      137.0
CAPTAINCY EXP      1
RUNS-S      1639
HS      73
AVE      28.26
SR-B      119.29
SIXERS      49
RUNS-C      345
WKTS      4
AVE-BL      86.25
ECON      8.84
SR-BL      58.5
AUCTION YEAR      2011
BASE PRICE      150000
SOLD PRICE      1800000
Premium      1650000
Name: 50, dtype: object

```

Which players got the maximum premium offering on their base price?

```

max_premium = df["Premium"].max()
max_premium

1650000

```

```
player_name = df[df["Premium"] == max_premium]
print(player_name['PLAYER NAME'])
```

```
50    Kohli, V
Name: PLAYER NAME, dtype: object
```

What is the average SOLD PRICE for each age category?

```
average_sold_price = df.groupby('AGE')['SOLD PRICE'].mean().reset_index()
average_sold_price.columns = ['AGE', 'Average SOLD PRICE']
print(average_sold_price)
```

	AGE	Average SOLD PRICE
0	1	720250.000000
1	2	484534.883721
2	3	520178.571429

Average SOLD PRICE for Different Playing Roles in Each Age Category?

```
average_sold_price = df.groupby(['AGE', 'PLAYING ROLE'])['SOLD PRICE'].mean().reset_index()
average_sold_price.columns = ['AGE', 'PLAYING ROLE', 'Average SOLD PRICE']
print(average_sold_price)
```

	AGE	PLAYING ROLE	Average SOLD PRICE
0	1	Allrounder	5.875000e+05
1	1	Batsman	1.110000e+06
2	1	Bowler	5.177143e+05
3	2	Allrounder	4.494000e+05
4	2	Batsman	6.547619e+05
5	2	Bowler	3.979310e+05
6	2	W. Keeper	4.677273e+05
7	3	Allrounder	7.666667e+05
8	3	Batsman	4.576923e+05
9	3	Bowler	4.143750e+05
10	3	W. Keeper	7.000000e+05