# Multiple Linear Regression - II

## Sourcing the data

In [33]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [34]:
```
ipl = pd.read_csv("https://raw.githubusercontent.com/Foridur3210/IPL-Dataset-Player-price-prediction/master/IPL%20IMB381IPL2013.c
```

In [35]:
```
ipl
```

Out[35]:

| | Sl.NO. | PLAYER NAME | AGE | COUNTRY | TEAM | PLAYING ROLE | T-RUNS | T-WKTS | ODI-RUNS-S | ODI-SR-B | ... | SR-B | SIXERS | RUNS-C | WKTS | AVE-BL | ECON | SR-BL | AUCTION YEAR | I P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | Abdulla, YA | 2 | SA | KXIP | Allrounder | 0 | 0 | 0 | 0.00 | ... | 0.00 | 0 | 307 | 15 | 20.47 | 8.90 | 13.93 | 2009 | 5 |
| **1** | 2 | Abdur Razzak | 2 | BAN | RCB | Bowler | 214 | 18 | 657 | 71.41 | ... | 0.00 | 0 | 29 | 0 | 0.00 | 14.50 | 0.00 | 2008 | 5 |
| **2** | 3 | Agarkar, AB | 2 | IND | KKR | Bowler | 571 | 58 | 1269 | 80.62 | ... | 121.01 | 5 | 1059 | 29 | 36.52 | 8.81 | 24.90 | 2008 | 2( |
| **3** | 4 | Ashwin, R | 1 | IND | CSK | Bowler | 284 | 31 | 241 | 84.56 | ... | 76.32 | 0 | 1125 | 49 | 22.96 | 6.23 | 22.14 | 2011 | 1( |
| **4** | 5 | Badrinath, S | 2 | IND | CSK | Batsman | 63 | 0 | 79 | 45.93 | ... | 120.71 | 28 | 0 | 0 | 0.00 | 0.00 | 0.00 | 2011 | 1( |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **125** | 126 | Yadav, AS | 2 | IND | DC | Batsman | 0 | 0 | 0 | 0.00 | ... | 125.64 | 2 | 0 | 0 | 0.00 | 0.00 | 0.00 | 2010 | 5 |
| **126** | 127 | Younis Khan | 2 | PAK | RR | Batsman | 6398 | 7 | 6814 | 75.78 | ... | 42.85 | 0 | 0 | 0 | 0.00 | 0.00 | 0.00 | 2008 | 2: |
| **127** | 128 | Yuvraj Singh | 2 | IND | KXIP+ | Batsman | 1775 | 9 | 8051 | 87.58 | ... | 131.88 | 67 | 569 | 23 | 24.74 | 7.02 | 21.13 | 2011 | 4( |
| **128** | 129 | Zaheer Khan | 2 | IND | MI+ | Bowler | 1114 | 288 | 790 | 73.55 | ... | 91.67 | 1 | 1783 | 65 | 27.43 | 7.75 | 21.26 | 2008 | 2( |
| **129** | 130 | Zoysa, DNT | 2 | SL | DC | Bowler | 288 | 64 | 343 | 95.81 | ... | 122.22 | 0 | 99 | 2 | 49.50 | 9.00 | 33.00 | 2008 | 1( |

130 rows × 26 columns

In [36]:
```
ipl.iloc[0:10, 15:]
```

Out[36]:

| | AVE | SR-B | SIXERS | RUNS-C | WKTS | AVE-BL | ECON | SR-BL | AUCTION YEAR | BASE PRICE | SOLD PRICE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.00 | 0.00 | 0 | 307 | 15 | 20.47 | 8.90 | 13.93 | 2009 | 50000 | 50000 |
| **1** | 0.00 | 0.00 | 0 | 29 | 0 | 0.00 | 14.50 | 0.00 | 2008 | 50000 | 50000 |
| **2** | 18.56 | 121.01 | 5 | 1059 | 29 | 36.52 | 8.81 | 24.90 | 2008 | 200000 | 350000 |
| **3** | 5.80 | 76.32 | 0 | 1125 | 49 | 22.96 | 6.23 | 22.14 | 2011 | 100000 | 850000 |
| **4** | 32.93 | 120.71 | 28 | 0 | 0 | 0.00 | 0.00 | 0.00 | 2011 | 100000 | 800000 |
| **5** | 21.00 | 95.45 | 0 | 0 | 0 | 0.00 | 0.00 | 0.00 | 2009 | 50000 | 50000 |
| **6** | 4.33 | 72.22 | 1 | 1342 | 52 | 25.81 | 7.98 | 19.40 | 2011 | 100000 | 500000 |
| **7** | 21.00 | 165.88 | 1 | 693 | 37 | 18.73 | 7.22 | 15.57 | 2011 | 200000 | 700000 |
| **8** | 30.45 | 114.73 | 3 | 610 | 19 | 32.11 | 6.85 | 28.11 | 2011 | 200000 | 950000 |
| **9** | 28.14 | 127.51 | 13 | 0 | 0 | 0.00 | 0.00 | 0.00 | 2008 | 200000 | 450000 |

## Preprocessing the data

In [37]:
```
# Target

y = ipl['SOLD PRICE']
y
```

Out[37]:
```
0        50000
1        50000
2       350000
3       850000
4       800000
        ...
125     750000
126     225000
127    1800000
128     450000
129     110000
Name: SOLD PRICE, Length: 130, dtype: int64
```

In [38]:
```
# Features

X = ipl.drop(['SOLD PRICE'], axis = 1)
X
```

Out[38]:

| | Sl.NO. | PLAYER NAME | AGE | COUNTRY | TEAM | PLAYING ROLE | T-RUNS | T-WKTS | ODI-RUNS-S | ODI-SR-B | ... | AVE | SR-B | SIXERS | RUNS-C | WKTS | AVE-BL | ECON | SR-BL | AUCT YE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | Abdulla, YA | 2 | SA | KXIP | Allrounder | 0 | 0 | 0 | 0.00 | ... | 0.00 | 0.00 | 0 | 307 | 15 | 20.47 | 8.90 | 13.93 | 2 |
| **1** | 2 | Abdur Razzak | 2 | BAN | RCB | Bowler | 214 | 18 | 657 | 71.41 | ... | 0.00 | 0.00 | 0 | 29 | 0 | 0.00 | 14.50 | 0.00 | 2 |
| **2** | 3 | Agarkar, AB | 2 | IND | KKR | Bowler | 571 | 58 | 1269 | 80.62 | ... | 18.56 | 121.01 | 5 | 1059 | 29 | 36.52 | 8.81 | 24.90 | 2 |
| **3** | 4 | Ashwin, R | 1 | IND | CSK | Bowler | 284 | 31 | 241 | 84.56 | ... | 5.80 | 76.32 | 0 | 1125 | 49 | 22.96 | 6.23 | 22.14 | 2 |
| **4** | 5 | Badrinath, S | 2 | IND | CSK | Batsman | 63 | 0 | 79 | 45.93 | ... | 32.93 | 120.71 | 28 | 0 | 0 | 0.00 | 0.00 | 0.00 | 2 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **125** | 126 | Yadav, AS | 2 | IND | DC | Batsman | 0 | 0 | 0 | 0.00 | ... | 9.80 | 125.64 | 2 | 0 | 0 | 0.00 | 0.00 | 0.00 | 2 |
| **126** | 127 | Younis Khan | 2 | PAK | RR | Batsman | 6398 | 7 | 6814 | 75.78 | ... | 3.00 | 42.85 | 0 | 0 | 0 | 0.00 | 0.00 | 0.00 | 2 |
| **127** | 128 | Yuvraj Singh | 2 | IND | KXIP+ | Batsman | 1775 | 9 | 8051 | 87.58 | ... | 26.32 | 131.88 | 67 | 569 | 23 | 24.74 | 7.02 | 21.13 | 2 |
| **128** | 129 | Zaheer Khan | 2 | IND | MI+ | Bowler | 1114 | 288 | 790 | 73.55 | ... | 9.90 | 91.67 | 1 | 1783 | 65 | 27.43 | 7.75 | 21.26 | 2 |
| **129** | 130 | Zoysa, DNT | 2 | SL | DC | Bowler | 288 | 64 | 343 | 95.81 | ... | 11.00 | 122.22 | 0 | 99 | 2 | 49.50 | 9.00 | 33.00 | 2 |

130 rows × 25 columns

In [39]:
```
# Dropping irrelevant features

X_1 = X.drop(['Sl.NO.','PLAYER NAME','TEAM'], axis = 1)
```

In [40]:
```
X_1.head()
```

Out[40]:

| | AGE | COUNTRY | PLAYING ROLE | T-RUNS | T-WKTS | ODI-RUNS-S | ODI-SR-B | ODI-WKTS | ODI-SR-BL | CAPTAINCY EXP | ... | AVE | SR-B | SIXERS | RUNS-C | WKTS | AVE-BL | ECON | SR-BL | AUCTIO YEA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2 | SA | Allrounder | 0 | 0 | 0 | 0.00 | 0 | 0.0 | 0 | ... | 0.00 | 0.00 | 0 | 307 | 15 | 20.47 | 8.90 | 13.93 | 20 |
| **1** | 2 | BAN | Bowler | 214 | 18 | 657 | 71.41 | 185 | 37.6 | 0 | ... | 0.00 | 0.00 | 0 | 29 | 0 | 0.00 | 14.50 | 0.00 | 20 |
| **2** | 2 | IND | Bowler | 571 | 58 | 1269 | 80.62 | 288 | 32.9 | 0 | ... | 18.56 | 121.01 | 5 | 1059 | 29 | 36.52 | 8.81 | 24.90 | 20 |
| **3** | 1 | IND | Bowler | 284 | 31 | 241 | 84.56 | 51 | 36.8 | 0 | ... | 5.80 | 76.32 | 0 | 1125 | 49 | 22.96 | 6.23 | 22.14 | 20 |
| **4** | 2 | IND | Batsman | 63 | 0 | 79 | 45.93 | 0 | 0.0 | 0 | ... | 32.93 | 120.71 | 28 | 0 | 0 | 0.00 | 0.00 | 0.00 | 20 |

5 rows × 22 columns

In [41]:
```
# Converting categorical features to numeric

X_1 = pd.get_dummies(X_1, columns = ['AGE', 'COUNTRY', 'PLAYING ROLE', 'CAPTAINCY EXP'])
X_1.shape
```

Out[41]: (130, 37)

In [42]:
```
X_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 130 entries, 0 to 129
Data columns (total 37 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   T-RUNS                   130 non-null    int64
 1   T-WKTS                   130 non-null    int64
 2   ODI-RUNS-S               130 non-null    int64
 3   ODI-SR-B                 130 non-null    float64
 4   ODI-WKTS                 130 non-null    int64
 5   ODI-SR-BL                130 non-null    float64
 6   RUNS-S                   130 non-null    int64
 7   HS                       130 non-null    int64
 8   AVE                      130 non-null    float64
 9   SR-B                     130 non-null    float64
 10  SIXERS                   130 non-null    int64
 11  RUNS-C                   130 non-null    int64
 12  WKTS                     130 non-null    int64
 13  AVE-BL                   130 non-null    float64
 14  ECON                     130 non-null    float64
 15  SR-BL                    130 non-null    float64
 16  AUCTION YEAR             130 non-null    int64
 17  BASE PRICE               130 non-null    int64
 18  AGE_1                    130 non-null    uint8
 19  AGE_2                    130 non-null    uint8
 20  AGE_3                    130 non-null    uint8
 21  COUNTRY_AUS              130 non-null    uint8
 22  COUNTRY_BAN              130 non-null    uint8
 23  COUNTRY_ENG              130 non-null    uint8
 24  COUNTRY_IND              130 non-null    uint8
 25  COUNTRY_NZ               130 non-null    uint8
 26  COUNTRY_PAK              130 non-null    uint8
 27  COUNTRY_SA               130 non-null    uint8
 28  COUNTRY_SL               130 non-null    uint8
 29  COUNTRY_WI               130 non-null    uint8
 30  COUNTRY_ZIM              130 non-null    uint8
 31  PLAYING ROLE_Allrounder  130 non-null    uint8
 32  PLAYING ROLE_Batsman     130 non-null    uint8
 33  PLAYING ROLE_Bowler      130 non-null    uint8
 34  PLAYING ROLE_W. Keeper   130 non-null    uint8
 35  CAPTAINCY EXP_0          130 non-null    uint8
 36  CAPTAINCY EXP_1          130 non-null    uint8
dtypes: float64(7), int64(11), uint8(19)
memory usage: 20.8 KB
```

All features are converted to numeric

In [43]:
```
# Adding constant 1 to each row

import statsmodels.api as sm
```

In [44]:
```
X_1 = sm.add_constant(X_1)
X_1
```

```
C:\Users\Urvi Sharma\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:142: FutureWarning: In a future version of pandas
all arguments of concat except for the argument 'objs' will be keyword-only
  x = pd.concat(x[::order], 1)
```

Out[44]:

| | const | T-RUNS | T-WKTS | ODI-RUNS-S | ODI-SR-B | ODI-WKTS | ODI-SR-BL | RUNS-S | HS | AVE | ... | COUNTRY_SA | COUNTRY_SL | COUNTRY_WI | COUNTRY_ZIM | PLAYING ROLE_Allrounder | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1.0 | 0 | 0 | 0 | 0.00 | 0 | 0.0 | 0 | 0 | 0.00 | ... | 1 | 0 | 0 | 0 | 1 | |
| **1** | 1.0 | 214 | 18 | 657 | 71.41 | 185 | 37.6 | 0 | 0 | 0.00 | ... | 0 | 0 | 0 | 0 | 0 | |
| **2** | 1.0 | 571 | 58 | 1269 | 80.62 | 288 | 32.9 | 167 | 39 | 18.56 | ... | 0 | 0 | 0 | 0 | 0 | |
| **3** | 1.0 | 284 | 31 | 241 | 84.56 | 51 | 36.8 | 58 | 11 | 5.80 | ... | 0 | 0 | 0 | 0 | 0 | |
| **4** | 1.0 | 63 | 0 | 79 | 45.93 | 0 | 0.0 | 1317 | 71 | 32.93 | ... | 0 | 0 | 0 | 0 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **125** | 1.0 | 0 | 0 | 0 | 0.00 | 0 | 0.0 | 49 | 16 | 9.80 | ... | 0 | 0 | 0 | 0 | 0 | |
| **126** | 1.0 | 6398 | 7 | 6814 | 75.78 | 3 | 86.6 | 3 | 3 | 3.00 | ... | 0 | 0 | 0 | 0 | 0 | |
| **127** | 1.0 | 1775 | 9 | 8051 | 87.58 | 109 | 44.3 | 1237 | 66 | 26.32 | ... | 0 | 0 | 0 | 0 | 0 | |
| **128** | 1.0 | 1114 | 288 | 790 | 73.55 | 278 | 35.4 | 99 | 23 | 9.90 | ... | 0 | 0 | 0 | 0 | 0 | |
| **129** | 1.0 | 288 | 64 | 343 | 95.81 | 108 | 39.4 | 11 | 10 | 11.00 | ... | 0 | 1 | 0 | 0 | 0 | |

130 rows × 38 columns

In [45]:
```
X_1.shape
```

Out[45]: (130, 38)

## Splitting data to train and test

In [46]:
```
from sklearn.model_selection import train_test_split
```

In [47]:
```
X_train_1, X_test_1, y_train, y_test = train_test_split(X_1, y, test_size = 0.2, random_state = 10)
```

In [48]:
```
X_train_1.shape, X_test_1.shape, y_train.shape, y_test.shape
```

Out[48]: ((104, 38), (26, 38), (104,), (26,))

In [49]:
```
X_train_1
```

Out[49]:

| | const | T-RUNS | T-WKTS | ODI-RUNS-S | ODI-SR-B | ODI-WKTS | ODI-SR-BL | RUNS-S | HS | AVE | ... | COUNTRY_SA | COUNTRY_SL | COUNTRY_WI | COUNTRY_ZIM | PLAYING ROLE_Allrounder | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **19** | 1.0 | 654 | 11 | 2536 | 84.00 | 25 | 47.6 | 978 | 74 | 36.22 | ... | 1 | 0 | 0 | 0 | 0 | |
| **14** | 1.0 | 0 | 0 | 69 | 56.09 | 0 | 0.0 | 1540 | 95 | 31.43 | ... | 0 | 0 | 0 | 0 | 0 | |
| **91** | 1.0 | 9382 | 0 | 10472 | 75.75 | 0 | 0.0 | 1567 | 94 | 27.98 | ... | 0 | 1 | 0 | 0 | 0 | |
| **35** | 1.0 | 503 | 0 | 575 | 87.51 | 1 | 66.0 | 1006 | 73 | 31.44 | ... | 0 | 0 | 0 | 0 | 0 | |
| **20** | 1.0 | 380 | 157 | 73 | 45.62 | 60 | 35.6 | 4 | 3 | 4.00 | ... | 0 | 0 | 1 | 0 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **64** | 1.0 | 392 | 43 | 5 | 27.77 | 19 | 40.1 | 186 | 31 | 10.94 | ... | 0 | 0 | 0 | 0 | 0 | |
| **15** | 1.0 | 3509 | 0 | 6773 | 88.19 | 1 | 12.0 | 1782 | 70 | 37.13 | ... | 0 | 0 | 0 | 0 | 0 | |
| **100** | 1.0 | 537 | 1 | 1587 | 70.40 | 1 | 42.0 | 40 | 23 | 20.00 | ... | 0 | 1 | 0 | 0 | 0 | |
| **125** | 1.0 | 0 | 0 | 0 | 0.00 | 0 | 0.0 | 49 | 16 | 9.80 | ... | 0 | 0 | 0 | 0 | 0 | |
| **9** | 1.0 | 5515 | 1 | 4686 | 84.76 | 0 | 0.0 | 394 | 50 | 28.14 | ... | 1 | 0 | 0 | 0 | 0 | |

104 rows × 38 columns

In [50]:
```
X_test_1
```

Out[50]:

| | const | T-RUNS | T-WKTS | ODI-RUNS-S | ODI-SR-B | ODI-WKTS | ODI-SR-BL | RUNS-S | HS | AVE | ... | COUNTRY_SA | COUNTRY_SL | COUNTRY_WI | COUNTRY_ZIM | PLAYING ROLE_Allrounder |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **105** | 1.0 | 281 | 87 | 44 | 36.36 | 75 | 33.0 | 33 | 15 | 11.00 | ... | 0 | 0 | 0 | 0 | 0 |
| **96** | 1.0 | 0 | 0 | 35 | 116.66 | 1 | 150.0 | 36 | 16 | 9.00 | ... | 0 | 0 | 0 | 0 | 1 |
| **41** | 1.0 | 6973 | 98 | 13430 | 91.21 | 323 | 46.0 | 768 | 114 | 27.43 | ... | 0 | 1 | 0 | 0 | 1 |
| **128** | 1.0 | 1114 | 288 | 790 | 73.55 | 278 | 35.4 | 99 | 23 | 9.90 | ... | 0 | 0 | 0 | 0 | 0 |
| **116** | 1.0 | 0 | 0 | 786 | 91.92 | 0 | 0.0 | 1538 | 69 | 26.98 | ... | 0 | 0 | 0 | 0 | 0 |
| **59** | 1.0 | 1219 | 7 | 1447 | 82.59 | 42 | 43.0 | 376 | 65 | 25.07 | ... | 0 | 1 | 0 | 0 | 1 |
| **53** | 1.0 | 16 | 16 | 73 | 58.87 | 100 | 34.8 | 8 | 8 | 4.00 | ... | 1 | 0 | 0 | 0 | 0 |
| **63** | 1.0 | 2173 | 0 | 2763 | 75.10 | 0 | 0.0 | 117 | 47 | 16.71 | ... | 0 | 0 | 0 | 0 | 0 |
| **45** | 1.0 | 2648 | 0 | 2924 | 84.31 | 0 | 0.0 | 128 | 53 | 25.60 | ... | 0 | 0 | 0 | 0 | 0 |
| **39** | 1.0 | 0 | 0 | 860 | 78.61 | 57 | 46.2 | 904 | 48 | 23.18 | ... | 0 | 0 | 0 | 0 | 1 |
| **2** | 1.0 | 571 | 58 | 1269 | 80.62 | 288 | 32.9 | 167 | 39 | 18.56 | ... | 0 | 0 | 0 | 0 | 0 |
| **47** | 1.0 | 1000 | 0 | 1008 | 74.50 | 0 | 0.0 | 1231 | 69 | 24.14 | ... | 0 | 0 | 0 | 0 | 0 |
| **121** | 1.0 | 11 | 1 | 43 | 43.87 | 28 | 35.3 | 217 | 25 | 9.43 | ... | 0 | 0 | 0 | 0 | 0 |
| **52** | 1.0 | 2506 | 619 | 938 | 61.06 | 337 | 43.0 | 35 | 8 | 11.67 | ... | 0 | 0 | 0 | 0 | 0 |
| **38** | 1.0 | 5708 | 7 | 5262 | 86.97 | 2 | 117.0 | 958 | 116 | 39.92 | ... | 0 | 0 | 0 | 0 | 0 |
| **1** | 1.0 | 214 | 18 | 657 | 71.41 | 185 | 37.6 | 0 | 0 | 0.00 | ... | 0 | 0 | 0 | 0 | 0 |
| **84** | 1.0 | 3781 | 421 | 3519 | 86.69 | 393 | 39.9 | 147 | 33 | 18.37 | ... | 1 | 0 | 0 | 0 | 1 |
| **56** | 1.0 | 556 | 25 | 1042 | 84.44 | 133 | 33.3 | 177 | 39 | 17.70 | ... | 0 | 1 | 0 | 0 | 1 |
| **10** | 1.0 | 2200 | 86 | 2004 | 81.39 | 142 | 34.1 | 839 | 70 | 27.97 | ... | 0 | 0 | 1 | 0 | 1 |
| **58** | 1.0 | 0 | 0 | 245 | 95.33 | 13 | 63.2 | 74 | 27 | 8.22 | ... | 0 | 0 | 0 | 0 | 1 |
| **83** | 1.0 | 6654 | 5 | 4184 | 86.76 | 7 | 57.1 | 634 | 103 | 42.27 | ... | 0 | 0 | 0 | 0 | 0 |
| **102** | 1.0 | 320 | 7 | 925 | 97.26 | 56 | 44.8 | 439 | 87 | 25.82 | ... | 0 | 0 | 1 | 0 | 1 |
| **43** | 1.0 | 624 | 0 | 2753 | 72.03 | 0 | 0.0 | 259 | 34 | 14.39 | ... | 0 | 0 | 0 | 0 | 0 |
| **70** | 1.0 | 0 | 0 | 1 | 50.00 | 1 | 42.0 | 4 | 3 | 4.00 | ... | 0 | 0 | 0 | 0 | 0 |
| **26** | 1.0 | 6373 | 72 | 8087 | 83.95 | 156 | 44.4 | 1804 | 128 | 50.11 | ... | 0 | 0 | 1 | 0 | 1 |
| **48** | 1.0 | 88 | 24 | 126 | 70.78 | 37 | 51.5 | 111 | 21 | 18.50 | ... | 0 | 0 | 0 | 0 | 0 |

26 rows × 38 columns

## Building the model

In [51]:
```
mlr_1 = sm.OLS(y_train, X_train_1) #estimating coefficients of linear regression equations which describe the relationship betwee
```

In [52]:
```
# Training the model

mlr_1= mlr_1.fit()
```

In [53]:
```
mlr_1.params
```

Out[53]:
```
const                     -4.030794e+07
T-RUNS                    -3.642910e+01
T-WKTS                    -7.924946e+02
ODI-RUNS-S                 1.524333e+01
ODI-SR-B                  -1.061064e+03
ODI-WKTS                   1.649076e+03
ODI-SR-BL                 -1.044786e+03
RUNS-S                     1.805545e+02
HS                        -2.881482e+03
AVE                        5.848201e+03
SR-B                      -6.373365e+01
SIXERS                     3.016505e+03
RUNS-C                     1.745518e+02
WKTS                      -1.364873e+03
AVE-BL                     1.169297e+04
ECON                      -3.327271e+03
SR-BL                     -1.669414e+04
AUCTION YEAR               4.406899e+04
BASE PRICE                 1.888119e+00
AGE_1                     -1.329211e+07
AGE_2                     -1.347979e+07
AGE_3                     -1.353603e+07
COUNTRY_AUS               -4.453859e+06
COUNTRY_BAN                9.993025e-08
COUNTRY_ENG               -4.916729e+06
COUNTRY_IND               -4.303342e+06
COUNTRY_NZ                -4.374145e+06
COUNTRY_PAK               -4.496219e+06
COUNTRY_SA                -4.395567e+06
COUNTRY_SL                -4.486193e+06
COUNTRY_WI                -4.386283e+06
COUNTRY_ZIM               -4.495603e+06
PLAYING ROLE_Allrounder   -1.005057e+07
PLAYING ROLE_Batsman      -1.001859e+07
PLAYING ROLE_Bowler       -1.009737e+07
PLAYING ROLE_W. Keeper    -1.014141e+07
CAPTAINCY EXP_0           -2.023362e+07
CAPTAINCY EXP_1           -2.007432e+07
dtype: float64
```

## Diagnosing the model

In [54]:
```
mlr_1.summary2()
```

Out[54]:

| | | | |
|---|---|---|---|
| Model: | OLS | Adj. R-squared: | 0.503 |
| Dependent Variable: | SOLD PRICE | AIC: | 2941.3368 |
| Date: | 2022-10-12 10:55 | BIC: | 3028.6017 |
| No. Observations: | 104 | Log-Likelihood: | -1437.7 |
| Df Model: | 32 | F-statistic: | 4.257 |
| Df Residuals: | 71 | Prob (F-statistic): | 1.92e-07 |
| R-squared: | 0.657 | Scale: | 8.7185e+10 |

| | Coef. | Std.Err. | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | -40307940.3172 | 24745537.1103 | -1.6289 | 0.1078 | -89649139.9119 | 9033259.2775 |
| **T-RUNS** | -36.4291 | 26.8420 | -1.3572 | 0.1790 | -89.9505 | 17.0923 |
| **T-WKTS** | -792.4946 | 566.6974 | -1.3984 | 0.1663 | -1922.4571 | 337.4678 |
| **ODI-RUNS-S** | 15.2433 | 28.6606 | 0.5319 | 0.5965 | -41.9042 | 72.3909 |
| **ODI-SR-B** | -1061.0636 | 1450.3045 | -0.7316 | 0.4668 | -3952.8887 | 1830.7615 |
| **ODI-WKTS** | 1649.0764 | 742.0599 | 2.2223 | 0.0295 | 169.4510 | 3128.7018 |
| **ODI-SR-BL** | -1044.7855 | 1686.6499 | -0.6194 | 0.5376 | -4407.8700 | 2318.2989 |
| **RUNS-S** | 180.5545 | 163.9192 | 1.1015 | 0.2744 | -146.2911 | 507.4002 |
| **HS** | -2881.4824 | 2458.9831 | -1.1718 | 0.2452 | -7784.5555 | 2021.5907 |
| **AVE** | 5848.2011 | 7729.2484 | 0.7566 | 0.4518 | -9563.4826 | 21259.8847 |
| **SR-B** | -63.7337 | 1172.4877 | -0.0544 | 0.9568 | -2401.6078 | 2274.1405 |
| **SIXERS** | 3016.5046 | 3549.7471 | 0.8498 | 0.3983 | -4061.4900 | 10094.4993 |
| **RUNS-C** | 174.5518 | 249.3836 | 0.6999 | 0.4863 | -322.7049 | 671.8085 |
| **WKTS** | -1364.8732 | 6016.5118 | -0.2269 | 0.8212 | -13361.4571 | 10631.7107 |
| **AVE-BL** | 11692.9681 | 9725.7685 | 1.2023 | 0.2333 | -7699.6634 | 31085.5997 |
| **ECON** | -3327.2705 | 9459.2777 | -0.3517 | 0.7261 | -22188.5345 | 15533.9935 |
| **SR-BL** | -16694.1377 | 13373.3174 | -1.2483 | 0.2160 | -43359.7753 | 9971.4998 |
| **AUCTION YEAR** | 44068.9943 | 27027.4095 | 1.6305 | 0.1074 | -9822.1297 | 97960.1183 |
| **BASE PRICE** | 1.8881 | 0.5338 | 3.5374 | 0.0007 | 0.8238 | 2.9524 |
| **AGE_1** | -13292113.4053 | 8254614.0242 | -1.6103 | 0.1118 | -29751346.2895 | 3167119.4789 |
| **AGE_2** | -13479794.7546 | 8248893.0973 | -1.6341 | 0.1067 | -29927620.4346 | 2968030.9254 |
| **AGE_3** | -13536032.1574 | 8242925.8285 | -1.6421 | 0.1050 | -29971959.4414 | 2899895.1266 |
| **COUNTRY_AUS** | -4453859.3284 | 2771303.9943 | -1.6071 | 0.1125 | -9979682.5470 | 1071963.8902 |
| **COUNTRY_BAN** | 0.0000 | 0.0000 | 1.6302 | 0.1075 | -0.0000 | 0.0000 |
| **COUNTRY_ENG** | -4916729.2323 | 2814659.3130 | -1.7468 | 0.0850 | -10529000.5010 | 695542.0365 |
| **COUNTRY_IND** | -4303342.2864 | 2779116.7278 | -1.5485 | 0.1260 | -9844743.6532 | 1238059.0803 |
| **COUNTRY_NZ** | -4374144.5913 | 2745138.4899 | -1.5934 | 0.1155 | -9847795.2760 | 1099506.0933 |
| **COUNTRY_PAK** | -4496219.2405 | 2721433.5914 | -1.6522 | 0.1029 | -9922603.7001 | 930165.2191 |
| **COUNTRY_SA** | -4395566.8905 | 2763070.5940 | -1.5908 | 0.1161 | -9904973.1751 | 1113839.3941 |
| **COUNTRY_SL** | -4486192.5787 | 2731080.1550 | -1.6426 | 0.1049 | -9931811.7397 | 959426.5823 |
| **COUNTRY_WI** | -4386283.1747 | 2747032.4467 | -1.5967 | 0.1148 | -9863710.3019 | 1091143.9526 |
| **COUNTRY_ZIM** | -4495602.9945 | 2750548.5599 | -1.6344 | 0.1066 | -9980041.0523 | 988835.0633 |
| **PLAYING ROLE_Allrounder** | -10050574.2972 | 6186416.1335 | -1.6246 | 0.1087 | -22385937.7147 | 2284789.1204 |
| **PLAYING ROLE_Batsman** | -10018590.2432 | 6186804.6479 | -1.6193 | 0.1098 | -22354728.3364 | 2317547.8501 |
| **PLAYING ROLE_Bowler** | -10097368.6031 | 6199035.9371 | -1.6289 | 0.1078 | -22457895.1943 | 2263157.9882 |
| **PLAYING ROLE_W. Keeper** | -10141407.1739 | 6176128.9522 | -1.6420 | 0.1050 | -22456258.5345 | 2173444.1867 |
| **CAPTAINCY EXP_0** | -20233622.9568 | 12374362.3734 | -1.6351 | 0.1064 | -44907400.7375 | 4440154.8238 |
| **CAPTAINCY EXP_1** | -20074317.3605 | 12371417.5760 | -1.6226 | 0.1091 | -44742223.3819 | 4593588.6610 |

| | | | |
|---|---|---|---|
| Omnibus: | 11.448 | Durbin-Watson: | 2.154 |
| Prob(Omnibus): | 0.003 | Jarque-Bera (JB): | 12.071 |
| Skew: | 0.705 | Prob(JB): | 0.002 |
| Kurtosis: | 3.893 | Condition No.: | 11985550242486654 |

We look at features whose P value is less than 0.05 since that feature will be influencing the target variable

## Multicollinearity

Multicollinearity - One feature dependent on another feature

Address multicollinearity using variance inflation factor

If Variance Inflation Factor > 4

Steps:

1. Check whether multicollinearity is present using VIF
2. Checking correlation of the variable with all other variables
3. Decide

In [55]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [56]:
```
def var_inf_factor(data):
    vif = pd.DataFrame()
    vif['Feature'] = data.columns
    vif['VIF_Value'] = [variance_inflation_factor(data.values, i) for i in range(data.shape[1])]
    print(vif)
```

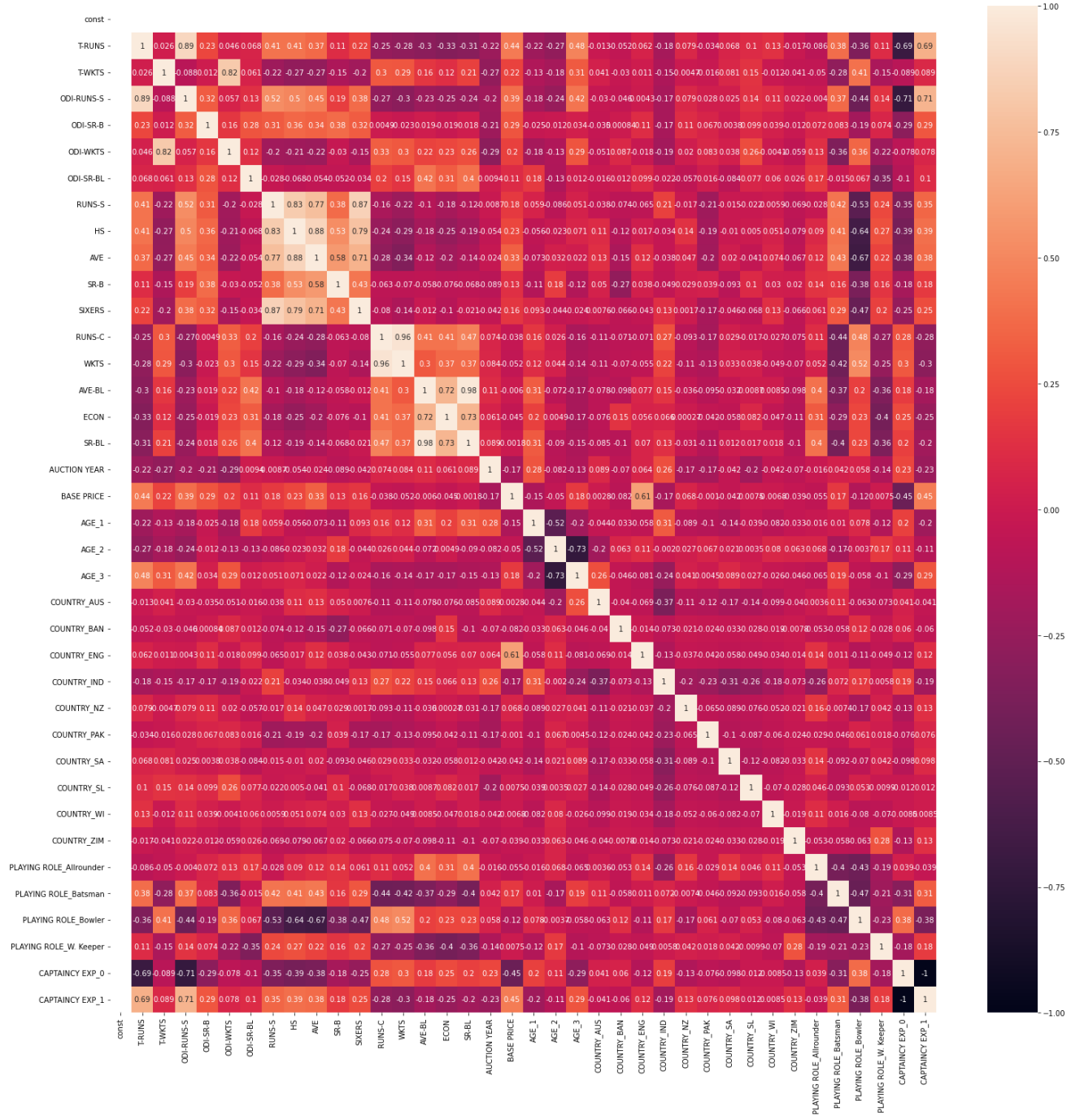
In [57]:
```
# Calling the fn

var_inf_factor(X_1)
```

```
                   Feature  VIF_Value
0                    const   0.000000
1                   T-RUNS   9.233542
2                   T-WKTS   6.522453
3                ODI-RUNS-S  11.067128
4                  ODI-SR-B   1.703841
5                  ODI-WKTS   7.048664
6                 ODI-SR-BL   1.707550
7                    RUNS-S   9.948044
8                        HS   8.602278
9                       AVE   7.467939
10                    SR-B   2.293498
11                  SIXERS   6.425581
12                  RUNS-C  22.310115
13                    WKTS  20.896087
14                  AVE-BL  45.182628
15                    ECON   2.981483
16                   SR-BL  45.596075
17             AUCTION YEAR   1.508571
18               BASE PRICE   3.347050
19                    AGE_1        inf
20                    AGE_2        inf
21                    AGE_3        inf
22              COUNTRY_AUS        inf
23              COUNTRY_BAN        inf
24              COUNTRY_ENG        inf
25              COUNTRY_IND        inf
26               COUNTRY_NZ        inf
27              COUNTRY_PAK        inf
28               COUNTRY_SA        inf
29               COUNTRY_SL        inf
30               COUNTRY_WI        inf
31              COUNTRY_ZIM        inf
32  PLAYING ROLE_Allrounder        inf
33     PLAYING ROLE_Batsman        inf
34      PLAYING ROLE_Bowler        inf
35   PLAYING ROLE_W. Keeper        inf
36          CAPTAINCY EXP_0        inf
37          CAPTAINCY EXP_1        inf
C:\Users\Urvi Sharma\anaconda3\lib\site-packages\statsmodels\regression\linear_model.py:1715: RuntimeWarning: divide by zero en
countered in double_scalars
  return 1 - self.ssr/self.centered_tss
C:\Users\Urvi Sharma\anaconda3\lib\site-packages\statsmodels\stats\outliers_influence.py:193: RuntimeWarning: divide by zero en
countered in double_scalars
  vif = 1. / (1. - r_squared_i)
```

Identify features with VIF>4

In [58]:
```
plt.figure(figsize = (25, 25))
sns.heatmap(X_1.corr(), annot = True)
```

Out[58]: <AxesSubplot:>




Look for features with corr>0.7 (high correlation)

## Features to be dropped

T-RUNS <==> ODI-RUNS-S

T-WKTS <==> ODI-WKTS

ODI-RUNS-S <==> T-RUNS, CAPTAINCY-EXP_1

RUNS-S, HS, AVE, SIXERS (3 to be dropped)

RUNS-C <==> WKTS

AVG-BL, ECON, SR-BL (2 to be dropped)

In [59]:
```
features_to_drop_1 = ['ODI-RUNS-S', 'ODI-WKTS', 'HS', 'AVE', 'SIXERS', 'RUNS-C', 'ECON', 'SR-BL']
```

In [60]:
```
features_2 = list(set(X_1.columns)-set(features_to_drop_1))
```

In [61]:
```
features_2
```

Out[61]:
```
['COUNTRY_AUS',
 'AGE_1',
 'COUNTRY_IND',
 'PLAYING ROLE_Bowler',
 'COUNTRY_NZ',
 'RUNS-S',
 'AUCTION YEAR',
 'CAPTAINCY EXP_0',
 'AGE_3',
 'COUNTRY_ENG',
 'CAPTAINCY EXP_1',
 'COUNTRY_PAK',
 'const',
 'AVE-BL',
 'AGE_2',
 'WKTS',
 'T-RUNS',
 'ODI-SR-B',
 'ODI-SR-BL',
 'COUNTRY_ZIM',
 'COUNTRY_BAN',
 'PLAYING ROLE_W. Keeper',
 'COUNTRY_WI',
 'SR-B',
 'PLAYING ROLE_Batsman',
 'PLAYING ROLE_Allrounder',
 'COUNTRY_SL',
 'BASE PRICE',
 'COUNTRY_SA',
 'T-WKTS']
```

In [62]:
```
len(features_2) #30
```

Out[62]:
```
30
```

## The new feature set

In [63]:
```
X_2 = X_1[features_2]
```

In [64]:
```
X_2
```

Out[64]:

| | COUNTRY_AUS | AGE_1 | COUNTRY_IND | PLAYING ROLE_Bowler | COUNTRY_NZ | RUNS-S | AUCTION YEAR | CAPTAINCY EXP_0 | AGE_3 | COUNTRY_ENG | ... | COUNTRY_BAN | PLAYI ROLE_ Kee |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 2009 | 1 | 0 | 0 | ... | 0 | |
| **1** | 0 | 0 | 0 | 1 | 0 | 0 | 2008 | 1 | 0 | 0 | ... | 1 | |
| **2** | 0 | 0 | 1 | 1 | 0 | 167 | 2008 | 1 | 0 | 0 | ... | 0 | |
| **3** | 0 | 1 | 1 | 1 | 0 | 58 | 2011 | 1 | 0 | 0 | ... | 0 | |
| **4** | 0 | 0 | 1 | 0 | 0 | 1317 | 2011 | 1 | 0 | 0 | ... | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **125** | 0 | 0 | 1 | 0 | 0 | 49 | 2010 | 1 | 0 | 0 | ... | 0 | |
| **126** | 0 | 0 | 0 | 0 | 0 | 3 | 2008 | 0 | 0 | 0 | ... | 0 | |
| **127** | 0 | 0 | 1 | 0 | 0 | 1237 | 2011 | 0 | 0 | 0 | ... | 0 | |
| **128** | 0 | 0 | 1 | 1 | 0 | 99 | 2008 | 1 | 0 | 0 | ... | 0 | |
| **129** | 0 | 0 | 0 | 1 | 0 | 11 | 2008 | 1 | 0 | 0 | ... | 0 | |

130 rows × 30 columns

In [75]:
```
# Splitting X_2 to train and test

X_train_2, X_test_2, y_train_2, y_test_2 = train_test_split(X_2, y, test_size = 0.2, random_state=10)
```

In [76]:
```
X_train_2.shape, X_test_2.shape, y_train_2.shape, y_test_2.shape
```

Out[76]:
```
((104, 30), (26, 30), (104,), (26,))
```

In [77]:
```
X_train_2
```

Out[77]:

| | COUNTRY_AUS | AGE_1 | COUNTRY_IND | PLAYING ROLE_Bowler | COUNTRY_NZ | RUNS-S | AUCTION YEAR | CAPTAINCY EXP_0 | AGE_3 | COUNTRY_ENG | ... | COUNTRY_BAN | PLAYI ROLE_ Kee |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **19** | 0 | 0 | 0 | 0 | 0 | 978 | 2009 | 1 | 0 | 0 | ... | 0 | |
| **14** | 0 | 0 | 1 | 0 | 0 | 1540 | 2011 | 1 | 0 | 0 | ... | 0 | |
| **91** | 0 | 0 | 0 | 0 | 0 | 1567 | 2008 | 0 | 0 | 0 | ... | 0 | |
| **35** | 1 | 0 | 0 | 0 | 0 | 1006 | 2011 | 1 | 1 | 0 | ... | 0 | |
| **20** | 0 | 0 | 0 | 1 | 0 | 4 | 2009 | 1 | 0 | 0 | ... | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **64** | 0 | 0 | 1 | 1 | 0 | 186 | 2011 | 1 | 0 | 0 | ... | 0 | |
| **15** | 0 | 0 | 1 | 0 | 0 | 1782 | 2008 | 0 | 0 | 0 | ... | 0 | |
| **100** | 0 | 0 | 0 | 0 | 0 | 40 | 2008 | 1 | 0 | 0 | ... | 0 | |
| **125** | 0 | 0 | 1 | 0 | 0 | 49 | 2010 | 1 | 0 | 0 | ... | 0 | |
| **9** | 0 | 0 | 0 | 0 | 0 | 394 | 2008 | 0 | 0 | 0 | ... | 0 | |

104 rows × 30 columns

In [78]:
```
# Building the model

mlr_2=sm.OLS(y_train_2,X_train_2)
```

In [79]:
```
# Fit

mlr_2=mlr_2.fit()
```

In [80]:
```
# Diagnosis

mlr_2.summary2()
```

Out[80]:

| | | | |
|---|---|---|---|
| Model: | OLS | Adj. R-squared: | 0.481 |
| Dependent Variable: | SOLD PRICE | AIC: | 2940.9738 |
| Date: | 2022-10-12 10:59 | BIC: | 3007.0836 |
| No. Observations: | 104 | Log-Likelihood: | -1445.5 |
| Df Model: | 24 | F-statistic: | 4.975 |
| Df Residuals: | 79 | Prob (F-statistic): | 3.07e-08 |
| R-squared: | 0.602 | Scale: | 9.1070e+10 |

| | Coef. | Std.Err. | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **COUNTRY_AUS** | -4857039.4771 | 2740138.0351 | -1.7726 | 0.0802 | -10311147.8044 | 597068.8503 |
| **AGE_1** | -14513247.2654 | 8158901.1914 | -1.7788 | 0.0791 | -30753133.8551 | 1726639.3243 |
| **COUNTRY_IND** | -4638299.6036 | 2746195.0086 | -1.6890 | 0.0952 | -10104464.0351 | 827864.8279 |
| **PLAYING ROLE_Bowler** | -10975670.9382 | 6130071.8878 | -1.7905 | 0.0772 | -23177273.8146 | 1225931.9381 |

In [81]:
```
# VIF

var_inf_factor(X_2)
```

```
                  Feature  VIF_Value
0             COUNTRY_AUS        inf
1                   AGE_1        inf
2             COUNTRY_IND        inf
3      PLAYING ROLE_Bowler        inf
4              COUNTRY_NZ        inf
5                  RUNS-S   2.502698
6            AUCTION YEAR   1.439253
7         CAPTAINCY EXP_0        inf
8                   AGE_3        inf
9             COUNTRY_ENG        inf
10        CAPTAINCY EXP_1        inf
11            COUNTRY_PAK        inf
12                  const   0.000000
13                 AVE-BL   2.314276
14                  AGE_2        inf
15                   WKTS   1.954828
16                 T-RUNS   3.300162
17                ODI-SR-B   1.614147
```

In [82]:
```
# features with p<0.05
features_3=['RUNS-S','BASE PRICE','COUNTRY_ENG']
```

In [83]:
```
X_3=X_2[features_3]
X_3
```

Out[83]:

| | RUNS-S | BASE PRICE | COUNTRY_ENG |
|---|---|---|---|
| **0** | 0 | 50000 | 0 |
| **1** | 0 | 50000 | 0 |
| **2** | 167 | 200000 | 0 |
| **3** | 58 | 100000 | 0 |
| **4** | 1317 | 100000 | 0 |
| **...** | ... | ... | ... |
| **125** | 49 | 50000 | 0 |
| **126** | 3 | 225000 | 0 |
| **127** | 1237 | 400000 | 0 |
| **128** | 99 | 200000 | 0 |
| **129** | 11 | 100000 | 0 |

130 rows × 3 columns

In [84]:
```
# Splitting

X_train_3,X_test_3,y_train_3,y_test_3=train_test_split(X_3,y,train_size=0.8,random_state=10)
```

In [85]:
```
X_train_3.shape,X_test_3.shape,y_train_3.shape,y_test_3.shape
```

Out[85]:
```
((104, 3), (26, 3), (104,), (26,))
```

In [86]:
```
# Building the model
mlr_3=sm.OLS(y_train_3,X_train_3)
```

In [87]:
```
mlr_3=mlr_3.fit()
```

In [88]:
```
mlr_3.summary2()
```

Out[88]:

| | | | |
|---|---|---|---|
| Model: | OLS | Adj. R-squared (uncentered): | 0.761 |
| Dependent Variable: | SOLD PRICE | AIC: | 2940.4262 |
| Date: | 2022-10-12 11:01 | BIC: | 2948.3594 |
| No. Observations: | 104 | Log-Likelihood: | -1467.2 |
| Df Model: | 3 | F-statistic: | 111.5 |
| Df Residuals: | 101 | Prob (F-statistic): | 6.25e-32 |
| R-squared (uncentered): | 0.768 | Scale: | 1.0818e+11 |

| | Coef. | Std.Err. | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **RUNS-S** | 275.5983 | 53.1568 | 5.1846 | 0.0000 | 170.1494 | 381.0472 |
| **BASE PRICE** | 1.9146 | 0.2184 | 8.7681 | 0.0000 | 1.4815 | 2.3478 |
| **COUNTRY_ENG** | -285999.0077 | 387666.4080 | -0.7377 | 0.4624 | -1055024.8800 | 483026.8645 |

| | | | |
|---|---|---|---|
| Omnibus: | 16.668 | Durbin-Watson: | 1.931 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 20.268 |
| Skew: | 0.868 | Prob(JB): | 0.000 |
| Kurtosis: | 4.290 | Condition No.: | 2680494 |

In [89]:
```
var_inf_factor(X_3)
```

```
       Feature  VIF_Value
0       RUNS-S   1.736843
1   BASE PRICE   2.295873
2  COUNTRY_ENG   1.504594
```

In [90]:
```
features_4=['RUNS-S','BASE PRICE']
```

In [91]:
```
X_4=X_3[features_4]
X_4
```

Out[91]:

| | RUNS-S | BASE PRICE |
|---|---|---|
| **0** | 0 | 50000 |
| **1** | 0 | 50000 |
| **2** | 167 | 200000 |
| **3** | 58 | 100000 |
| **4** | 1317 | 100000 |
| **...** | ... | ... |
| **125** | 49 | 50000 |
| **126** | 3 | 225000 |
| **127** | 1237 | 400000 |
| **128** | 99 | 200000 |
| **129** | 11 | 100000 |

130 rows × 2 columns

In [92]:
```
# Splitting

X_train_4,X_test_4,y_train_4,y_test_4=train_test_split(X_4,y,train_size=0.8,random_state=10)
```

In [93]:
```
X_train_4.shape,X_test_4.shape,y_train_4.shape,y_test_4.shape
```

Out[93]:
```
((104, 2), (26, 2), (104,), (26,))
```

In [94]:
```
# Building the model
mlr_4=sm.OLS(y_train_4,X_train_4)
```

In [95]:
```
mlr_4=mlr_4.fit()
```

In [96]:
```
mlr_4.summary2()
```

Out[96]:

| | | | |
|---|---|---|---|
| Model: | OLS | Adj. R-squared (uncentered): | 0.762 |
| Dependent Variable: | SOLD PRICE | AIC: | 2938.9851 |
| Date: | 2022-10-12 11:03 | BIC: | 2944.2739 |
| No. Observations: | 104 | Log-Likelihood: | -1467.5 |
| Df Model: | 2 | F-statistic: | 167.8 |
| Df Residuals: | 102 | Prob (F-statistic): | 5.57e-33 |
| R-squared (uncentered): | 0.767 | Scale: | 1.0769e+11 |

| | Coef. | Std.Err. | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **RUNS-S** | 289.6304 | 49.5265 | 5.8480 | 0.0000 | 191.3949 | 387.8659 |
| **BASE PRICE** | 1.8294 | 0.1849 | 9.8963 | 0.0000 | 1.4627 | 2.1960 |

| | | | |
|---|---|---|---|
| Omnibus: | 16.006 | Durbin-Watson: | 1.928 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 18.853 |
| Skew: | 0.862 | Prob(JB): | 0.000 |
| Kurtosis: | 4.175 | Condition No.: | 343 |

In [97]:
```
var_inf_factor(X_4)
```

```
      Feature  VIF_Value
0      RUNS-S   1.529542
1  BASE PRICE   1.529542
```

In [98]:
```
mlr_4.params
```

Out[98]:
```
RUNS-S        289.630397
BASE PRICE      1.829376
dtype: float64
```

## NOTE :-

**VIF SHOULD BE < 4 and p < 0.05 to stop the iterations**

The model is:

SOLD PRICE = RUNS-S * 289.630397 + BASE PRICE * 1.829376

## Residual Analysis

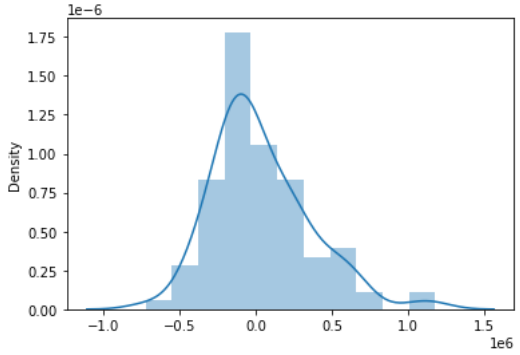
### Normality

In [101]:
```
mlr_4.resid
```

Out[101]:
```
19    -532071.340056
14    -328968.415292
91    -211194.841913
35    -232243.387237
20    -125564.927490
          ...
64      63191.142226
15     252128.216833
100    -94522.819814
125    644339.308580
9      -29989.584282
Length: 104, dtype: float64
```

In [102]:
```
sns.distplot(mlr_4.resid);
```

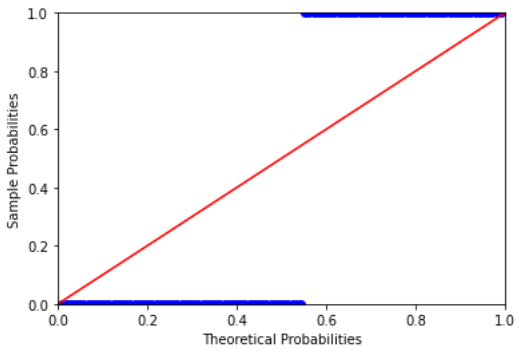
```
C:\Users\Urvi Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated funct
ion and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with simil
ar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```




In [103]:
```
## Prob prob plot

def prob_prob_plot(model):
    probplot = sm.ProbPlot(model.resid)
    probplot.ppplot(line = '45')
    plt.show();
```

In [104]:
```
prob_prob_plot(mlr_4)
```

```
C:\Users\Urvi Sharma\anaconda3\lib\site-packages\statsmodels\graphics\gofplots.py:993: UserWarning: marker is redundantly defin
ed by the 'marker' keyword argument and the fmt string "bo" (-> marker='o'). The keyword argument will take precedence.
  ax.plot(x, y, fmt, **plot_style)
```



In [ ]:
```
X_train_4
```

The residuals are not fully normally distributed

In [ ]:
```
## Homoscedasticity
```

In [ ]:
```
def standard(data):
    return (data-data.mean())/data.std()
```

In [ ]:
```
# Plotting residual plot

def residual_plot(model):
    plt.scatter(standard(model.fittedvalues),
                standard(model.resid))
    plt.xlabel(' Normalized Fitted Values')
    plt.ylabel(' Normalized Residue Values')
    plt.title(' Residual Plot')
    plt.show();
```

In [ ]:
```
# Calling the function

residual_plot(mlr_4)
```

Some what like a funnel shape.

So, not exactly following homoscedasticity

## Checking the outliers

Zscore

Cooks distance

Leverage Distance

In [ ]:
```
# zscore

from scipy.stats import zscore

z_score=zscore(X_4)
```

In [ ]:
```
z_score[z_score>3].count()
```

In [ ]:
```
z_score[z_score <-3].count()
```

There are 2 points (|zscore|>3), which can be outliers.

## Cooks distance

In [113]:
```
def cooks_dist(model):
    model_influence = model.get_influence()
    (c, ) = model_influence.cooks_distance
    plt.stem(np.arange(len(X_train_4)), c)
    plt.xlabel('Observation No')
    plt.ylabel('Cooks distance')
    plt.title('Cooks plot')
    plt.show();
```

In [114]:
```
cooks_dist(mlr_4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
C:\Users\URVISH~1\AppData\Local\Temp/ipykernel_3328/3481482799.py in <module>
----> 1 cooks_dist(mlr_4)

C:\Users\URVISH~1\AppData\Local\Temp/ipykernel_3328/3890932848.py in cooks_dist(model)
      1 def cooks_dist(model):
      2     model_influence = model.get_influence()
----> 3     (c, ) = model_influence.cooks_distance
      4     plt.stem(np.arange(len(X_train_4)), c)
      5     plt.xlabel('Observation No')

ValueError: too many values to unpack (expected 1)
```

The is no observation with cooks distance>1

So, the measure says there is no outlier

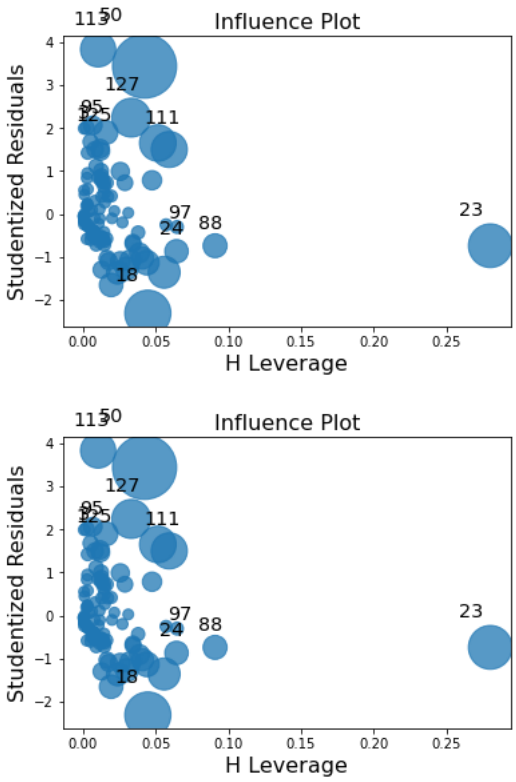
## Leverage distance

In [115]:
```
n = 104 # No. of training data

k = 2 # No. of features in the model mlr_4

lev_cutoff = (3*(k+1))/n

print('The leverage cutoff:', lev_cutoff)
```

```
The leverage cutoff: 0.08653846153846154
```

In [116]:
```
from statsmodels.graphics.regressionplots import influence_plot
influence_plot(mlr_4)
```

Out[116]:







Two outliers: 88 and 23

In [117]:
```
X_train_4.iloc[88]
```

Out[117]:
```
RUNS-S            0
BASE PRICE    50000
Name: 0, dtype: int64
```

In [118]:
```
X_train_4['RUNS-S'].min(), X_train_4['RUNS-S'].max()
```

Out[118]: (0, 2254)

In [119]:
```
X_train_4['BASE PRICE'].min(), X_train_4['BASE PRICE'].max()
```

Out[119]: (20000, 950000)

In [120]:
```
X_5 = X_4.drop([88, 23])
```

In [122]:
```
X_5.shape
```

Out[122]: (128, 2)

In [124]:
```
y_5 = y.drop([88, 23])
```

In [125]:
```
y_5.shape
```

Out[125]: (128,)

In [126]:
```
# Splitting into train and test

X_train_5,X_test_5,y_train_5,y_test_5=train_test_split(X_5,y_5,train_size=0.8,random_state=10)
```

In [127]:
```
X_train_5.shape,X_test_5.shape,y_train_5.shape,y_test_5.shape
```

Out[127]: ((102, 2), (26, 2), (102,), (26,))

In [128]:
```
mlr_5 = sm.OLS(y_train_5, X_train_5)
```

In [129]:
```
mlr_5 = mlr_5.fit()
```

In [130]:
```
mlr_5.summary2()
```

Out[130]:

| | | | |
|---|---|---|---|
| Model: | OLS | Adj. R-squared (uncentered): | 0.753 |
| Dependent Variable: | SOLD PRICE | AIC: | 2886.6249 |
| Date: | 2022-10-12 11:27 | BIC: | 2891.8749 |
| No. Observations: | 102 | Log-Likelihood: | -1441.3 |
| Df Model: | 2 | F-statistic: | 156.1 |
| Df Residuals: | 100 | Prob (F-statistic): | 1.77e-31 |
| R-squared (uncentered): | 0.757 | Scale: | 1.1213e+11 |

| | Coef. | Std.Err. | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **RUNS-S** | 340.0613 | 54.4170 | 6.2492 | 0.0000 | 232.0996 | 448.0230 |
| **BASE PRICE** | 1.5606 | 0.1776 | 8.7851 | 0.0000 | 1.2082 | 1.9131 |

| | | | |
|---|---|---|---|
| Omnibus: | 8.520 | Durbin-Watson: | 2.077 |
| Prob(Omnibus): | 0.014 | Jarque-Bera (JB): | 8.906 |
| Skew: | 0.534 | Prob(JB): | 0.012 |
| Kurtosis: | 3.977 | Condition No.: | 397 |

## Predicting the value

In [ ]:
```
mse_5 = mean_squared_error(mlr_5.predict(X_test_5), y_test_5)
```

In [131]:
```
y_pred = mlr_5.predict(X_test_5)
```

In [132]:
```
y_pred
```

Out[132]:
```
106    2.578986e+05
97     9.057153e+05
42     8.903873e+05
125    9.469443e+04
117    3.396707e+05
60     6.924056e+05
54     3.299916e+05
64     2.193143e+05
46     3.223275e+04
40     2.783022e+05
2      3.689159e+05
48     3.498725e+05
122    7.696150e+05
53     1.587833e+05
39     4.634783e+05
1      7.803143e+04
85     1.141875e+05
57     3.338896e+05
10     5.974371e+05
59     5.960516e+05
84     3.621147e+05
101    3.298089e+05
44     1.019362e+06
71     2.694859e+05
27     6.914514e+05
49     3.940805e+05
dtype: float64
```

## Performance of the model

In [133]:
```
from sklearn.metrics import r2_score, mean_squared_error
```

In [134]:
```
r2 = r2_score(y_test_5, y_pred)
print('R2:', r2)

mse = mean_squared_error(y_test_5, y_pred)

rmse = np.sqrt(mse)

print('RMSE:', rmse)
```

```
R2: 0.10139730948481451
RMSE: 276546.0338651926
```

## Tranform the target

In [137]:
```
y.min(), y.max()
```

Out[137]:
```
(20000, 1800000)
```

Huge difference between min and max, hence we need to compress this

In [139]:
```
y_sq = np.sqrt(y)
```

In [140]:
```
y_sq.min(), y_sq.max()
```

Out[140]:
```
(141.4213562373095, 1341.640786499874)
```

In [141]:
```
# log

y_log = np.log(y)
```

In [142]:
```
y_log.min(), y_log.max()
```

Out[142]:
```
(9.903487552536127, 14.403297222866392)
```

In [ ]:
```

```