In [1]:
```python
#LIb importing

import numpy as np # linear algebra
import pandas as pd # data processing,
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
df = pd.read_csv('C:/Users/Admin/Downloads/fifa_eda.csv') ## file reading
```

In [3]:
```python
df.head
```

Out[3]:
```
<bound method NDFrame.head of            ID               Name  Age Nationality  Overall  Potential  \
0      158023           L. Messi   31   Argentina       94         94
1       20801  Cristiano Ronaldo   33    Portugal       94         94
2      190871          Neymar Jr   26      Brazil       92         93
3      193080             De Gea   27       Spain       91         93
4      192985       K. De Bruyne   27     Belgium       91         92
...       ...                ...  ...         ...      ...        ...
18202  238813       J. Lundstram   19     England       47         65
18203  243165  N. Christoffersson   19      Sweden       47         63
18204  241638          B. Worman   16     England       47         67
18205  246268     D. Walker-Rice   17     England       47         66
18206  246269          G. Nugent   16     England       46         66

                      Club     Value   Wage Preferred Foot  \
0             FC Barcelona  110500.0  565.0           Left
1                 Juventus   77000.0  405.0          Right
2      Paris Saint-Germain  118500.0  290.0          Right
3        Manchester United   72000.0  260.0          Right
4          Manchester City  102000.0  355.0          Right
...                    ...       ...    ...            ...
18202       Crewe Alexandra      60.0    1.0          Right
18203        Trelleborgs FF      60.0    1.0          Right
18204      Cambridge United      60.0    1.0          Right
18205       Tranmere Rovers      60.0    1.0          Right
18206       Tranmere Rovers      60.0    1.0          Right

       International Reputation  Skill Moves Position  Joined  \
0                           5.0          4.0       RF    2004
1                           5.0          5.0       ST    2018
2                           5.0          5.0       LW    2017
3                           4.0          1.0       GK    2011
4                           4.0          4.0      RCM    2015
...                         ...          ...      ...     ...
18202                       1.0          2.0       CM    2017
18203                       1.0          2.0       ST    2018
18204                       1.0          2.0       ST    2017
18205                       1.0          2.0       RW    2018
18206                       1.0          2.0       CM    2018

      Contract Valid Until    Height  Weight  Release Clause
0               2021-01-01  5.583333   159.0        226500.0
1               2022-01-01  6.166667   183.0        127100.0
2               2022-01-01  5.750000   150.0        228100.0
3               2020-01-01  6.333333   168.0        138600.0
4               2023-01-01  5.916667   154.0        196400.0
...                    ...       ...     ...             ...
18202           2019-01-01  5.750000   134.0           143.0
18203           2020-01-01  6.250000   170.0           113.0
18204           2021-01-01  5.666667   148.0           165.0
18205           2019-01-01  5.833333   154.0           143.0
18206           2019-01-01  5.833333   176.0           165.0

[18207 rows x 18 columns]>
```

In [4]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18207 entries, 0 to 18206
Data columns (total 18 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   ID                        18207 non-null  int64  
 1   Name                      18207 non-null  object 
 2   Age                       18207 non-null  int64  
 3   Nationality               18207 non-null  object 
 4   Overall                   18207 non-null  int64  
 5   Potential                 18207 non-null  int64  
 6   Club                      17966 non-null  object 
 7   Value                     17955 non-null  float64
 8   Wage                      18207 non-null  float64
 9   Preferred Foot            18207 non-null  object 
 10  International Reputation  18159 non-null  float64
 11  Skill Moves               18159 non-null  float64
 12  Position                  18207 non-null  object 
 13  Joined                    18207 non-null  int64  
 14  Contract Valid Until      17918 non-null  object 
 15  Height                    18207 non-null  float64
 16  Weight                    18207 non-null  float64
 17  Release Clause            18207 non-null  float64
dtypes: float64(7), int64(5), object(6)
memory usage: 2.5+ MB
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['ID', 'Name', 'Age', 'Nationality', 'Overall', 'Potential', 'Club',
       'Value', 'Wage', 'Preferred Foot', 'International Reputation',
       'Skill Moves', 'Position', 'Joined', 'Contract Valid Until', 'Height',
       'Weight', 'Release Clause'],
      dtype='object')
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
ID                            0
Name                          0
Age                           0
Nationality                   0
Overall                       0
Potential                     0
Club                        241
Value                       252
Wage                          0
Preferred Foot                0
International Reputation     48
Skill Moves                  48
Position                      0
Joined                        0
Contract Valid Until        289
Height                        0
Weight                        0
Release Clause                0
dtype: int64
```

In [7]:
```
df.dropna(how='all',inplace=True)
```

In [8]:
```
# filling null values

df['Club'].fillna(0, inplace=True)
df['Value'].fillna(0, inplace=True)
df['International Reputation'].fillna(-1, inplace=True)
df['Skill Moves'].fillna(-1, inplace=True)
df['Contract Valid Until'].fillna(0, inplace=True)
```

In [9]:
```
df.isnull().sum()
```

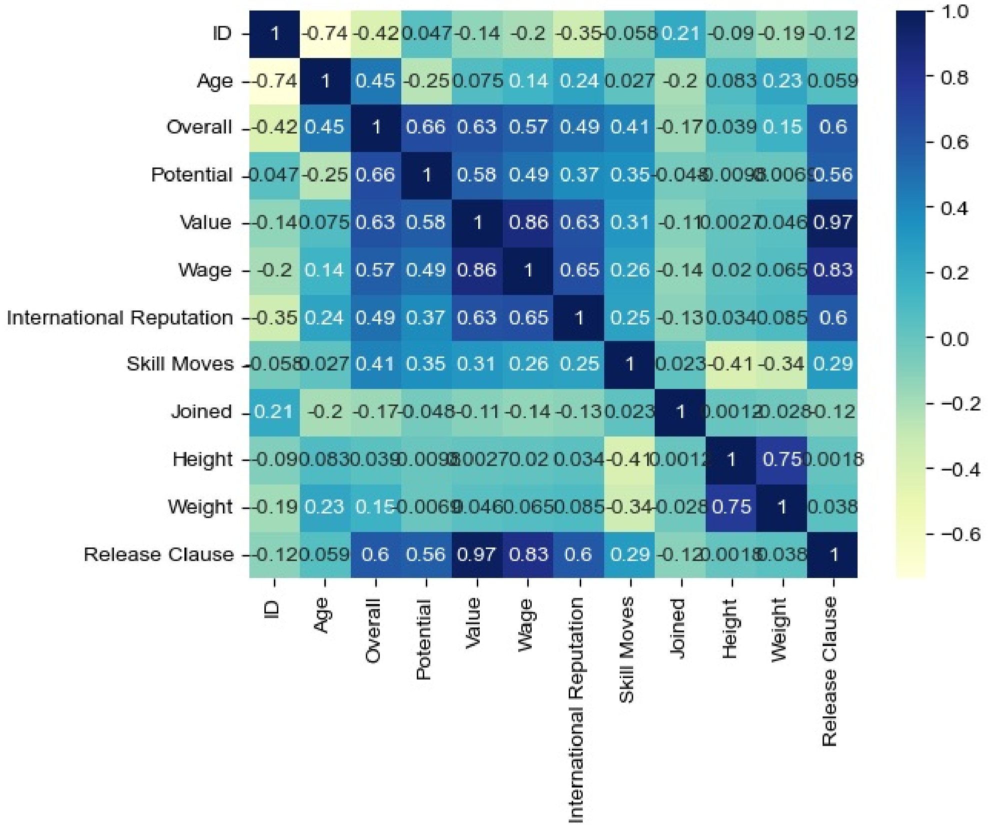
Out[9]:
```
ID                          0
Name                        0
Age                         0
Nationality                 0
Overall                     0
Potential                   0
Club                        0
Value                       0
Wage                        0
Preferred Foot              0
International Reputation    0
Skill Moves                 0
Position                    0
Joined                      0
Contract Valid Until        0
Height                      0
Weight                      0
Release Clause              0
dtype: int64
```

In [10]:
```
df.describe()
```

Out[10]:

	ID	Age	Overall	Potential	Value	Wage	International Reputation	Skill Moves	Joined	
count	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18
mean	214298.338606	25.122206	66.238699	71.307299	2410.695886	9.731312	1.107651	2.352447	2016.420607	
std	29965.244204	4.669943	6.908930	6.136496	5594.932671	21.999290	0.408159	0.774588	2.018194	
min	16.000000	16.000000	46.000000	48.000000	0.000000	0.000000	-1.000000	-1.000000	1991.000000	
25%	200315.500000	21.000000	62.000000	67.000000	300.000000	1.000000	1.000000	2.000000	2016.000000	
50%	221759.000000	25.000000	66.000000	71.000000	675.000000	3.000000	1.000000	2.000000	2017.000000	
75%	236529.500000	28.000000	71.000000	75.000000	2000.000000	9.000000	1.000000	3.000000	2018.000000	
max	246620.000000	45.000000	94.000000	95.000000	118500.000000	565.000000	5.000000	5.000000	2018.000000	

In [11]:
```python
#Finding Correlation between all the columns with each other

df = sns.heatmap(df.corr(), cmap="YlGnBu", annot=True)

# displaying heatmap
sns.set(rc = {'figure.figsize':(20,8)})
plt.show()

#There is strong correlation between Internation Reputation and Wage.
#Skill Moves has NEGATIVE correlation with HEIGHT and WEIGHT. It means more heighted
# or more weighted the player is lesser SKILL MOVES he will have.
#AGE has little bit positive correlation with WAGE.
#Whereas AGE has strong NEGATIVE CORRELATION with POTENTIAL but POSTIVE CORRELATION with OVERALL rating.
#OVERALL RATING has strong POSITIVE CORRELATION with WAGE and RELEASE CLAUSE>
```

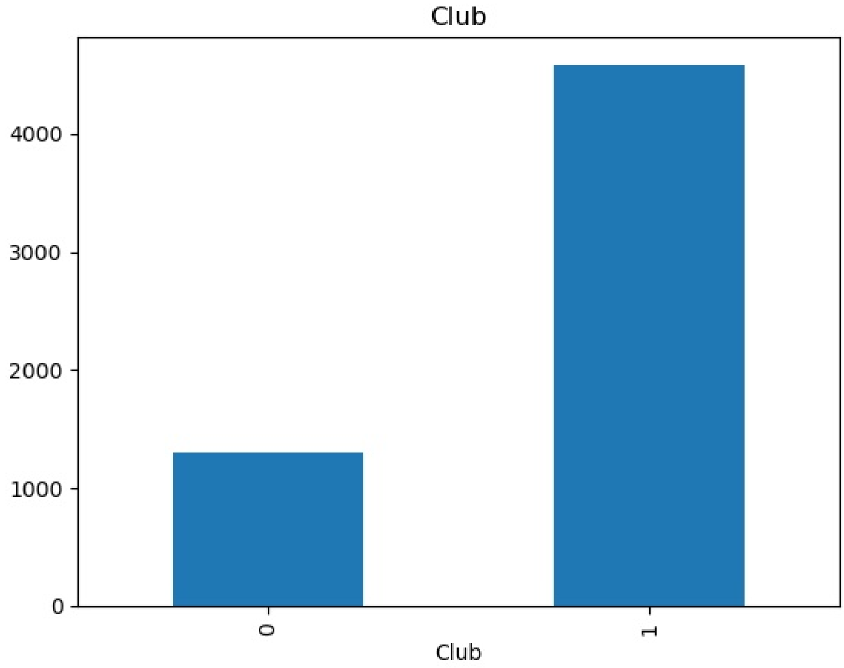


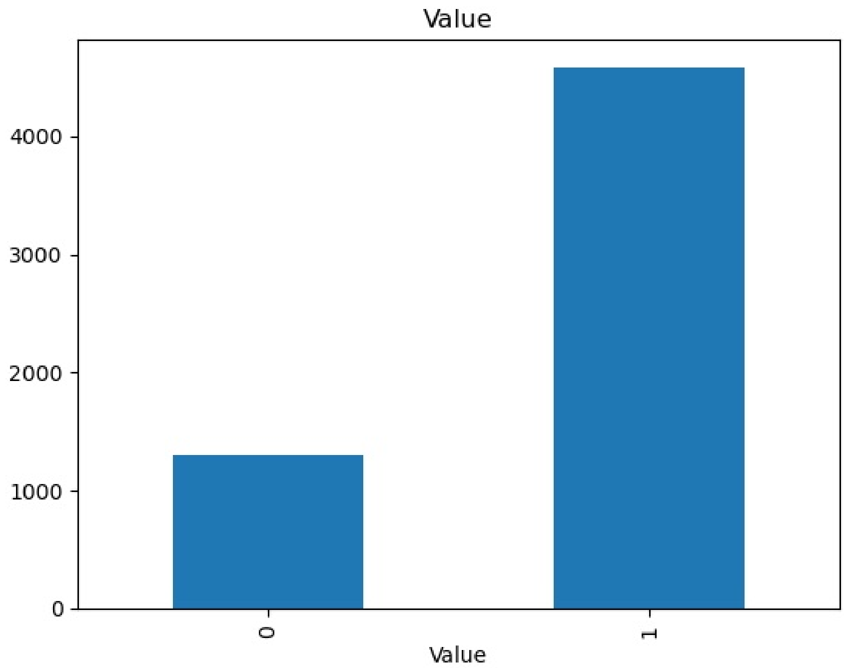
In [7]:
```python
# Eliminate The features contains Null values:

features_with_na = [features for features in df.columns if df[features].isnull().sum()>1]
for features in features_with_na:
    print(features , np.round(df[features].isnull().mean(),2),'%missing values')
```

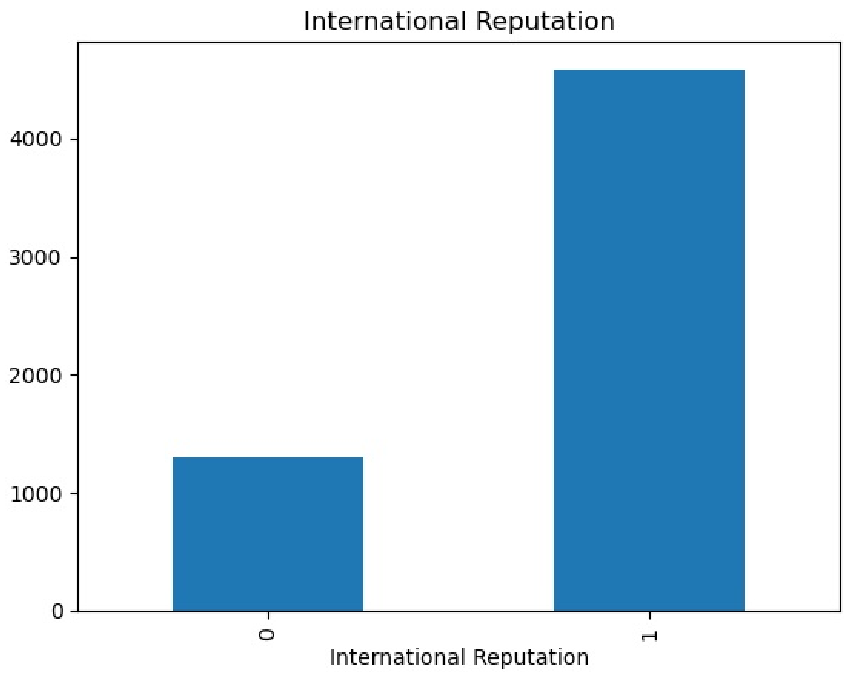
```
Club 0.01 %missing values
Value 0.01 %missing values
International Reputation 0.0 %missing values
Skill Moves 0.0 %missing values
Contract Valid Until 0.02 %missing values
```

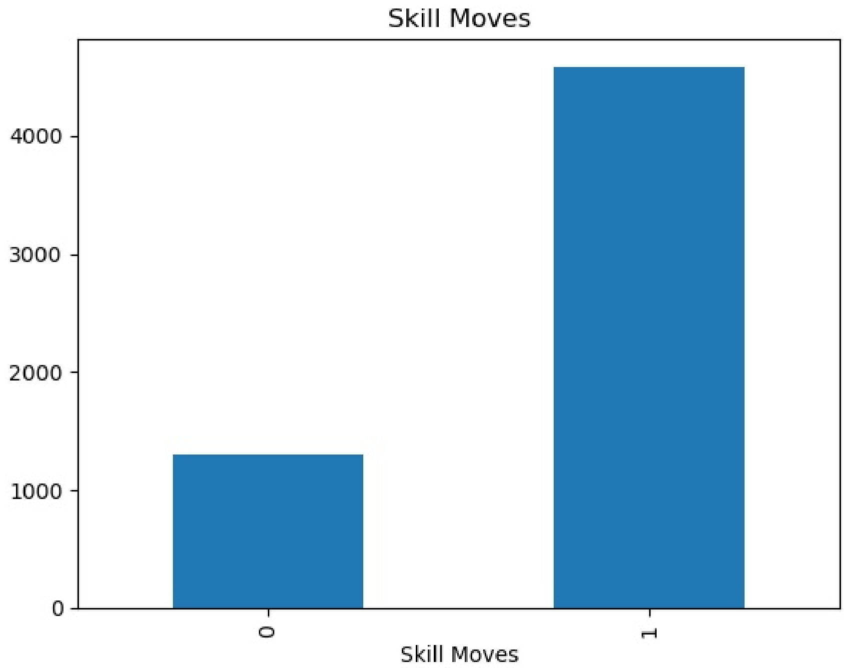
In [8]:
```python
#Comprehend the relation between features and target variables (Release Clasue)
for features in features_with_na:
    df11=df.copy()
    df[features]=np.where(df[features].isnull(),1,0)

    df.groupby(features)['Release Clause'].median().plot.bar()
    plt.title(features)
    plt.show()
```

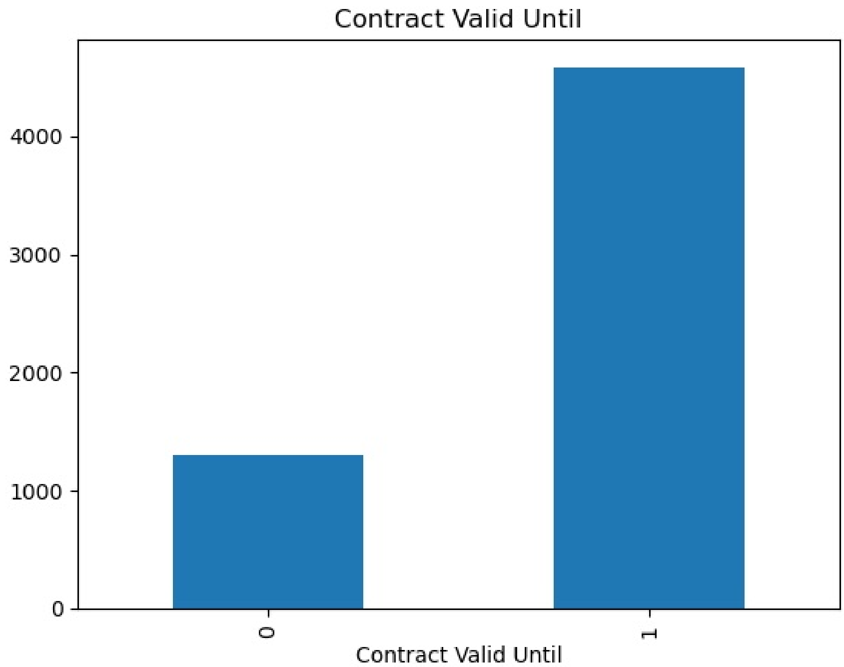
```
#From the  graph it is clearly visible that faetures and target variables have a logarithimic relationship
```











In [9]:
```
#Eliminate the Numerical features from the data
numerical_features=[features for features in df.columns if df[features].dtypes !='O']
print('No of numerical_features :',len(numerical_features))
df[numerical_features].head()
```

```
No of numerical_features : 14
```

Out[9]:

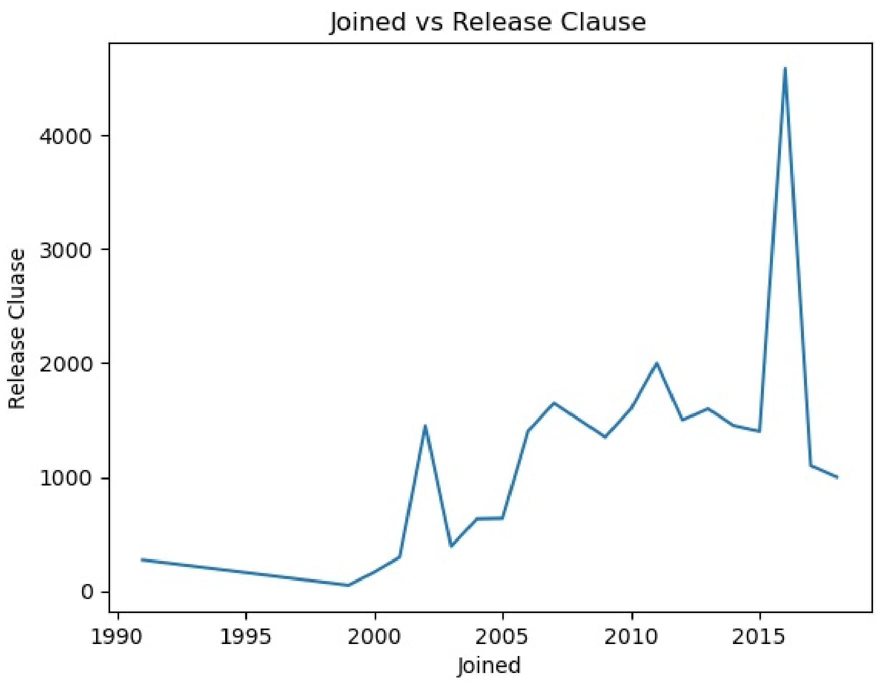
	ID	Age	Overall	Potential	Club	Value	Wage	International Reputation	Skill Moves	Joined	Contract Valid Until	Height	Weight	Release Clause
0	158023	31	94	94	0	0	565.0	0	0	2004	0	5.583333	159.0	226500.0
1	20801	33	94	94	0	0	405.0	0	0	2018	0	6.166667	183.0	127100.0
2	190871	26	92	93	0	0	290.0	0	0	2017	0	5.750000	150.0	228100.0
3	193080	27	91	93	0	0	260.0	0	0	2011	0	6.333333	168.0	138600.0
4	192985	27	91	92	0	0	355.0	0	0	2015	0	5.916667	154.0	196400.0

In [10]:
```
#Eliminate the year features from the numerical features :
year_features=[features for features in numerical_features if 'Joined' in features  ]

print(df[year_features])


for features in year_features:
    print(features,df[features].unique())
```

```
        Joined
0         2004
1         2018
2         2017
3         2011
4         2015
...        ...
18202     2017
18203     2018
18204     2017
18205     2018
18206     2018

[18207 rows x 1 columns]
Joined [2004 2018 2017 2011 2015 2012 2014 2005 2010 2016 2008 2013 2007 2009
 2002 2003 2006 2001 1991 1998 2000 1999]
```

In [11]:

```
##Comprehend The relation between year features and target varibales
df.groupby('Joined')['Release Clause'].median().plot()
plt.xlabel('Joined')
plt.ylabel('Release Cluase')
plt.title('Joined vs Release Clause')
plt.show()
##from the above graphs its is comprehend that players joined the year on 2015 have more Realase Clause
#numerical variable are of two type descrete and continues, so we eliminate these variables
## like wise for better analysis
```



In [12]:

```
#Eliminate Discrete features from numerical features

discrete_features=[features for features in numerical_features if len(df[features].unique())<10 and features no
len(discrete_features)

df[discrete_features]
```

Out[12]:

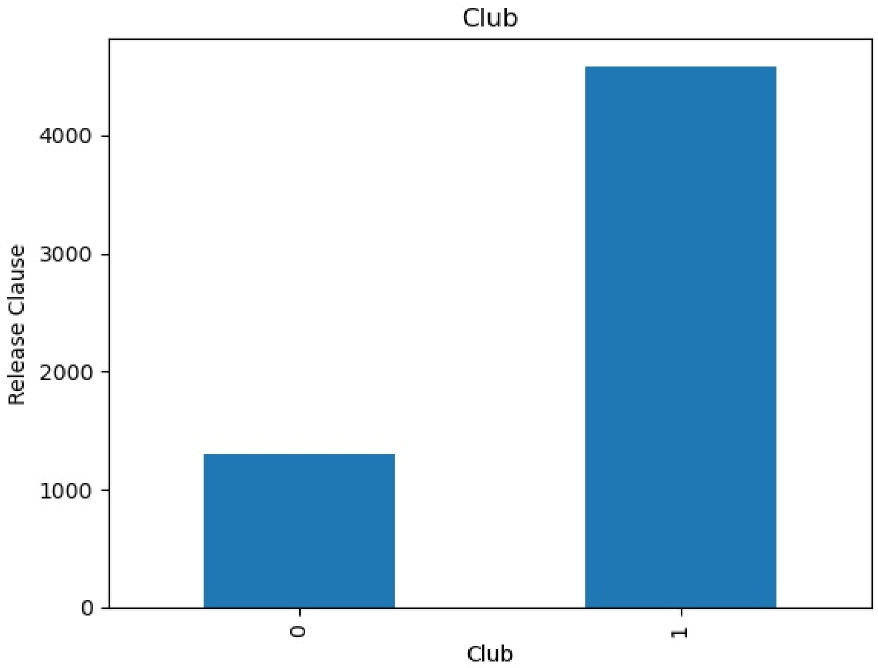
	Club	Value	International Reputation	Skill Moves	Contract Valid Until
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
...	...	...	...	...	...
18202	0	0	0	0	0
18203	0	0	0	0	0
18204	0	0	0	0	0
18205	0	0	0	0	0
18206	0	0	0	0	0

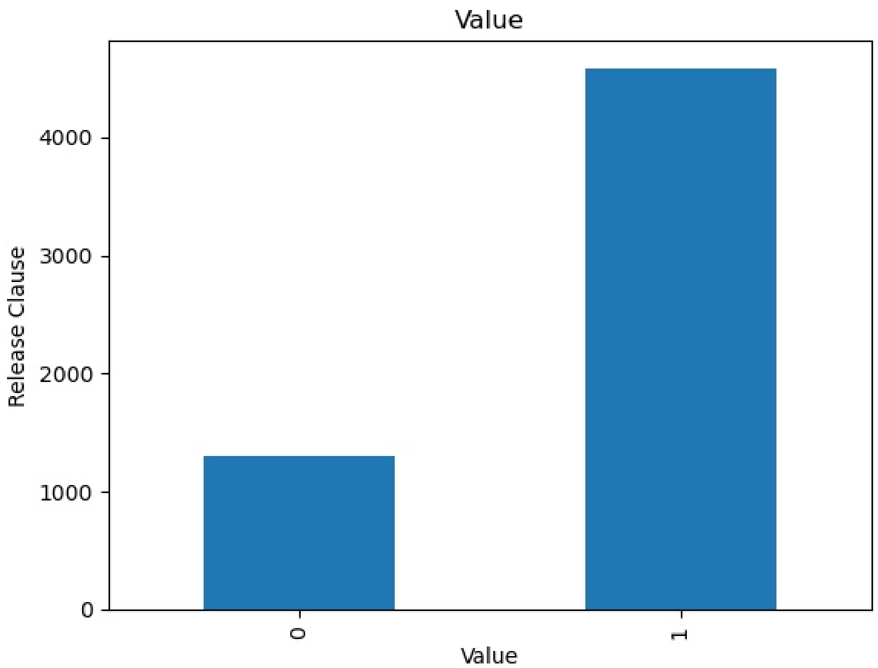
18207 rows × 5 columns

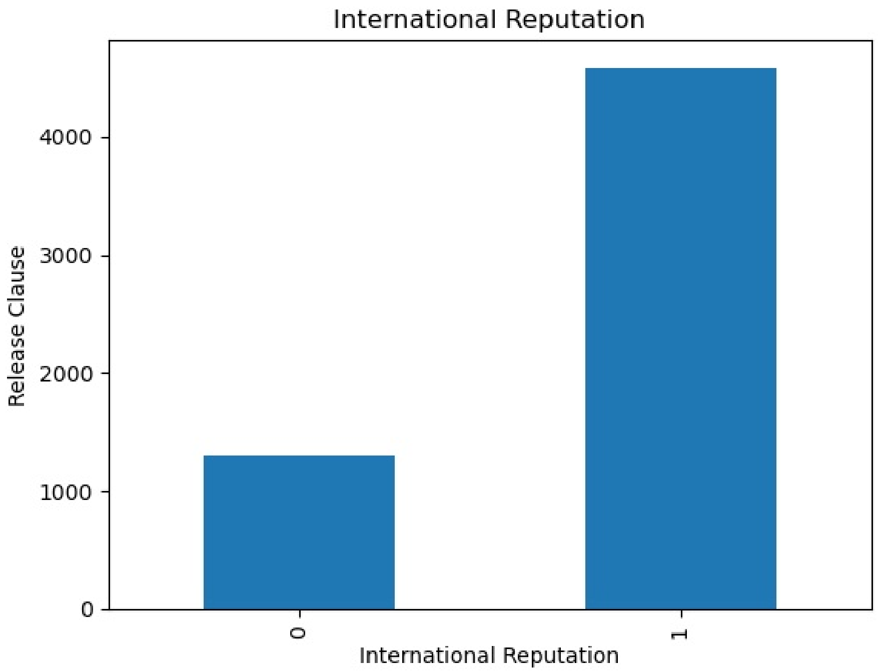
In [13]:

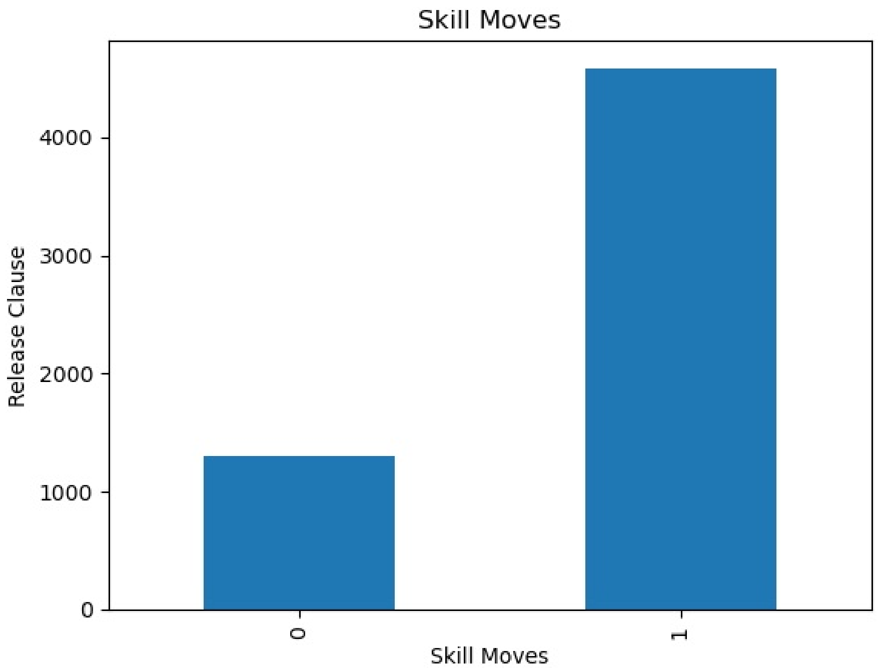
```
#Comprehend the relation between discrete features and target varibales
for features in discrete_features:
```

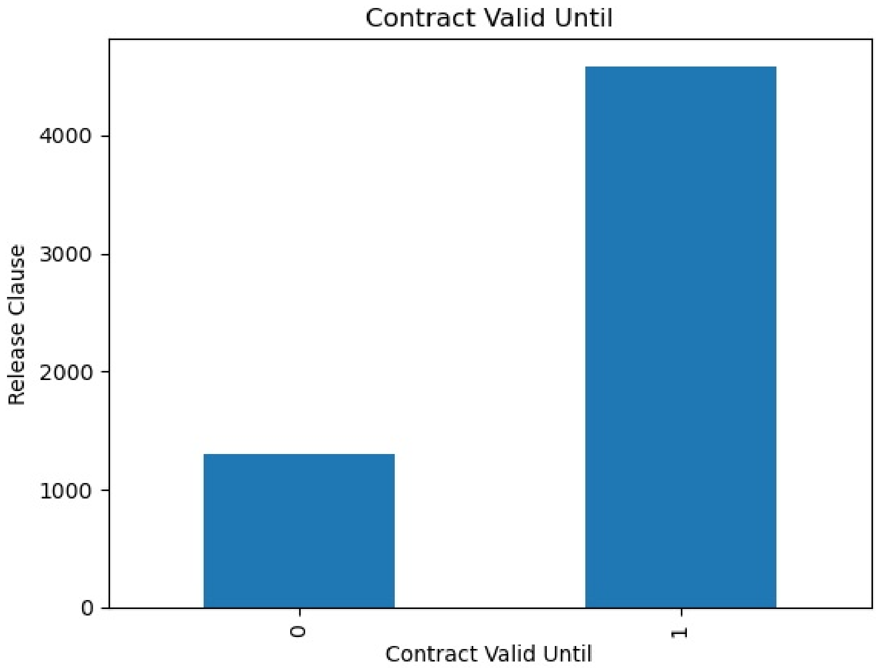
```python
    df=df.copy()
    df.groupby(features)['Release Clause'].median().plot.bar()
    plt.xlabel(features)
    plt.ylabel('Release Clause')
    plt.title(features)
    plt.show()
#insight  it is clearly visible that faetures and target variables have a logarithimic relationship
```






International Reputation
Release Clause
0
1000
2000
3000
4000
0
1
International Reputation


Skill Moves
Release Clause
0
1000
2000
3000
4000
0
1
Skill Moves



In [14]:

```
#Eliminate continues features from numerical features

contineous_features=[features for features in numerical_features if features not in discrete_features+year_feat
len(contineous_features)

df[contineous_features]
```

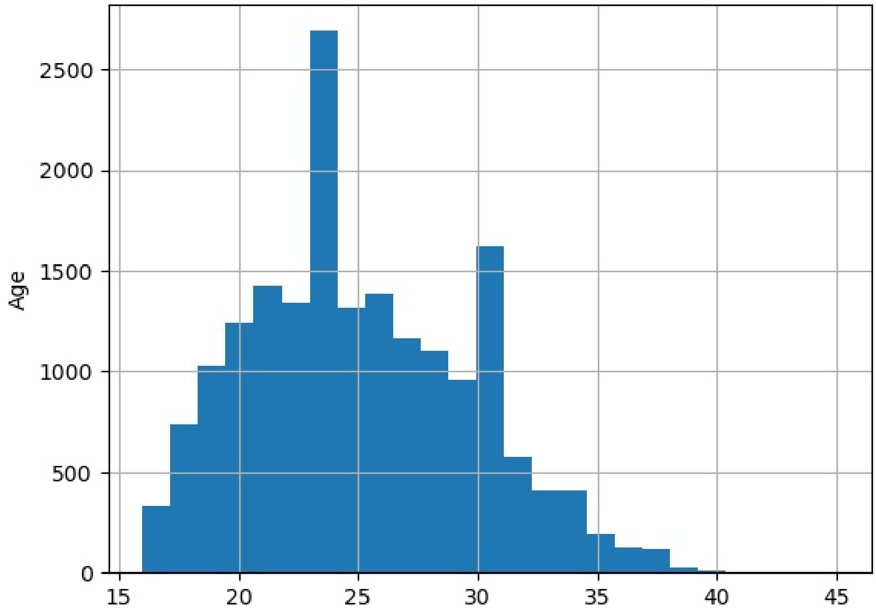
Out[14]:

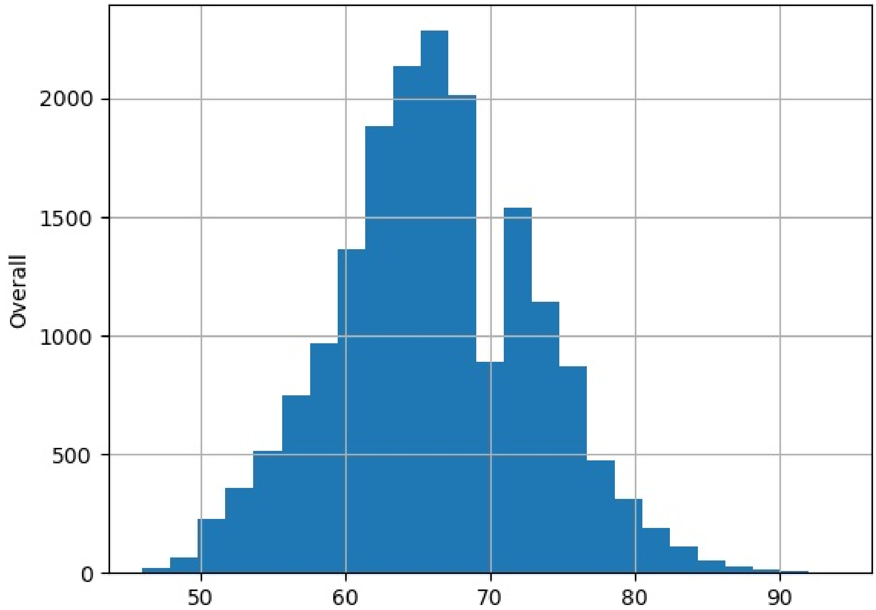
	Age	Overall	Potential	Wage	Height	Weight	Release Clause
0	31	94	94	565.0	5.583333	159.0	226500.0
1	33	94	94	405.0	6.166667	183.0	127100.0
2	26	92	93	290.0	5.750000	150.0	228100.0
3	27	91	93	260.0	6.333333	168.0	138600.0
4	27	91	92	355.0	5.916667	154.0	196400.0
...	...	...	...	...	...	...	...
18202	19	47	65	1.0	5.750000	134.0	143.0
18203	19	47	63	1.0	6.250000	170.0	113.0
18204	16	47	67	1.0	5.666667	148.0	165.0
18205	17	47	66	1.0	5.833333	154.0	143.0
18206	16	46	66	1.0	5.833333	176.0	165.0

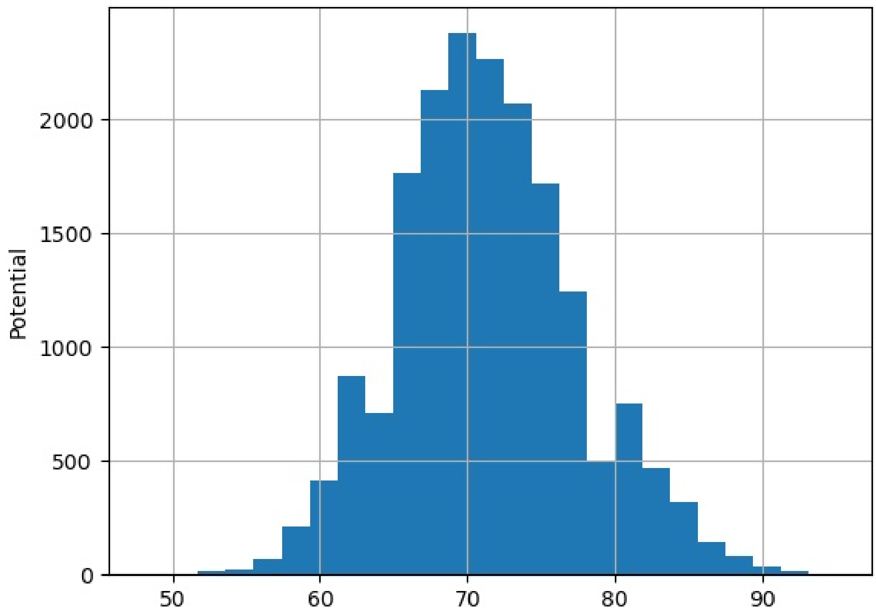
18207 rows × 7 columns

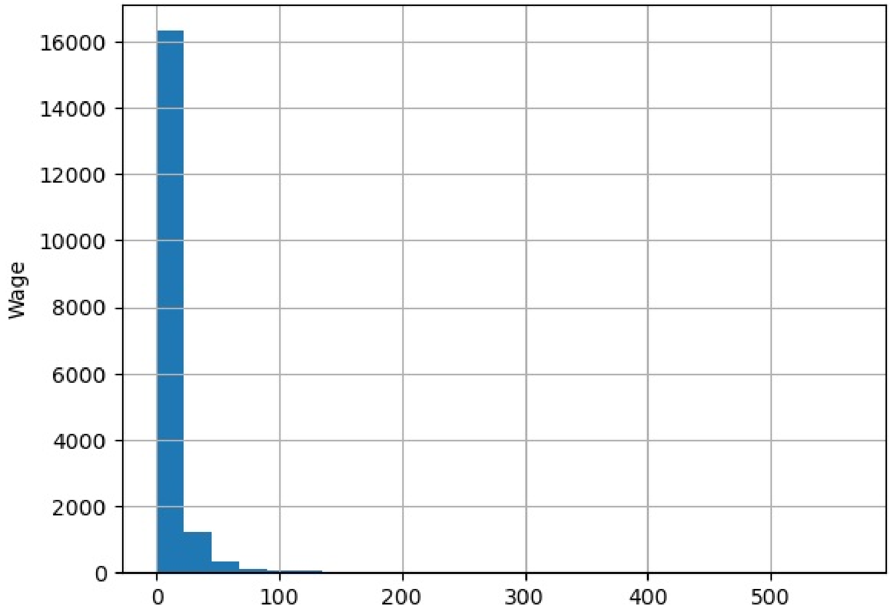
In [15]:

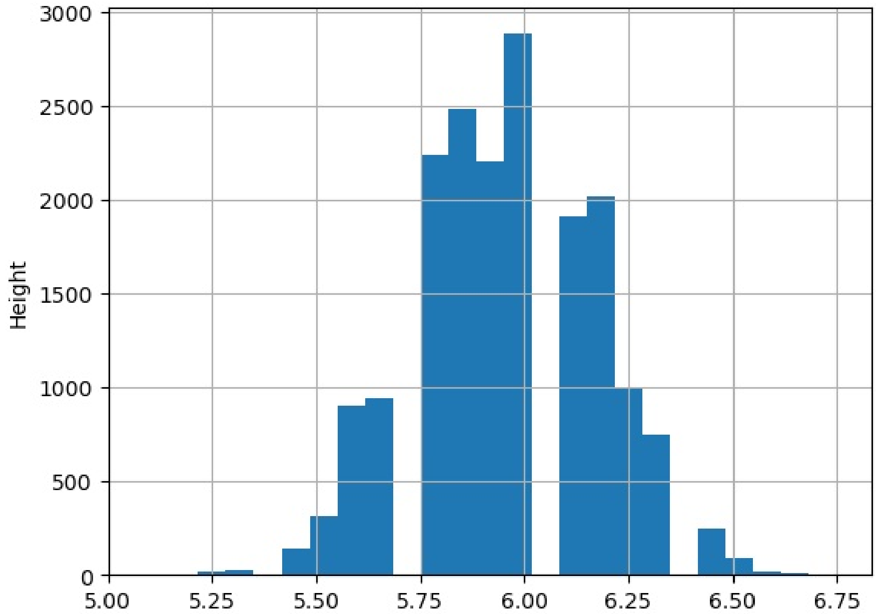
```
#comprehend the relation betwen contineuos features and target variables
for features in contineous_features:
    df1=df.copy()
    df1[features].hist(bins=25)
    plt.ylabel(features)
    plt.show()
```

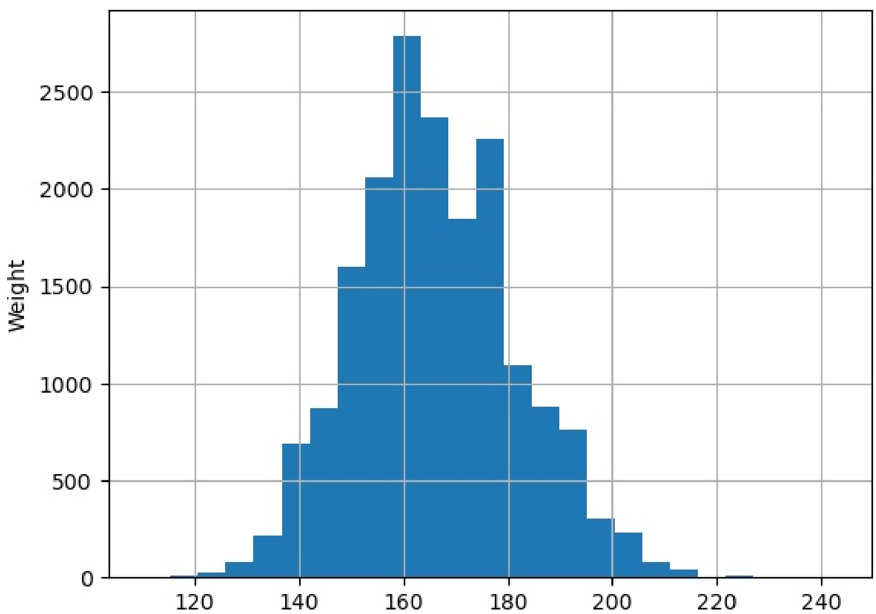

Age
0
500
1000
1500
2000
2500
15
20
25
30
35
40
45

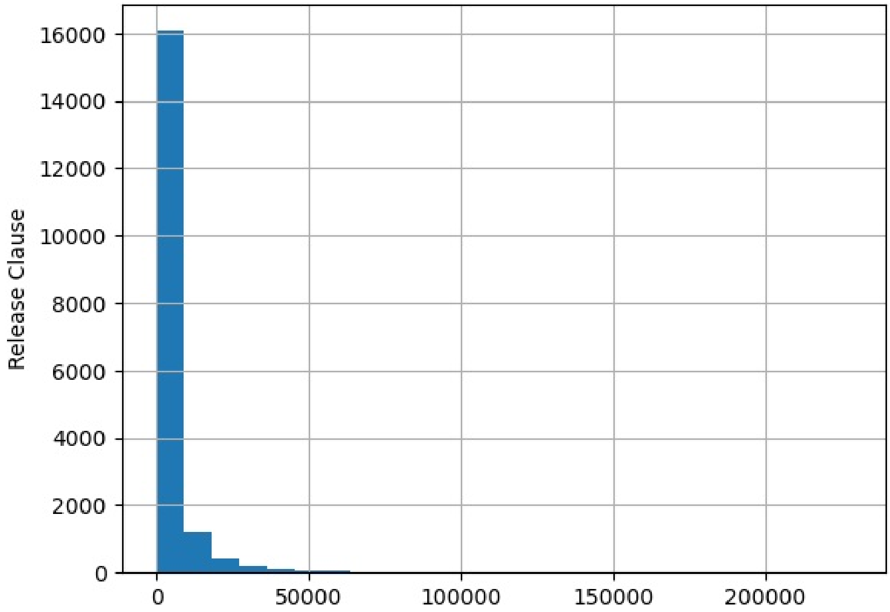

Overall
0
500
1000
1500
2000
50
60
70
80
90


Potential
0
500
1000
1500
2000
50
60
70
80
90


Wage
0
2000
4000
6000
8000
10000
12000
14000
16000
100
200
300
400
500

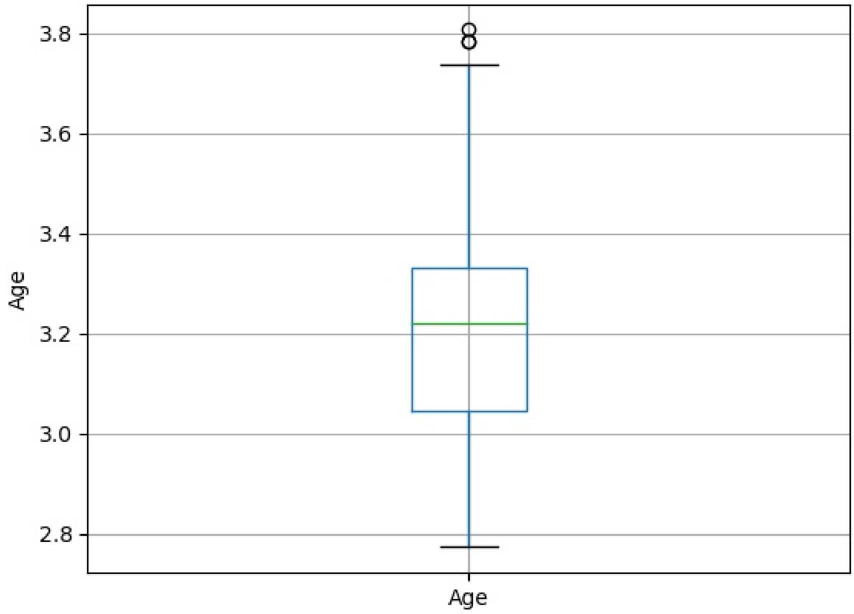

Height
0
500
1000
1500
2000
2500
3000
5.00
5.25
5.50
5.75
6.00
6.25
6.50
6.75

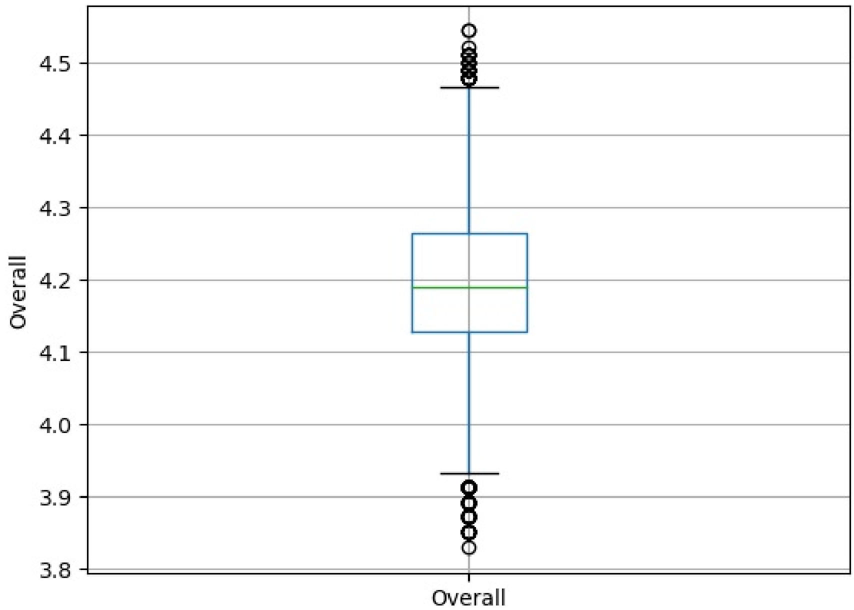

Weight
0
500
1000
1500
2000
2500
120
140
160
180
200
220
240

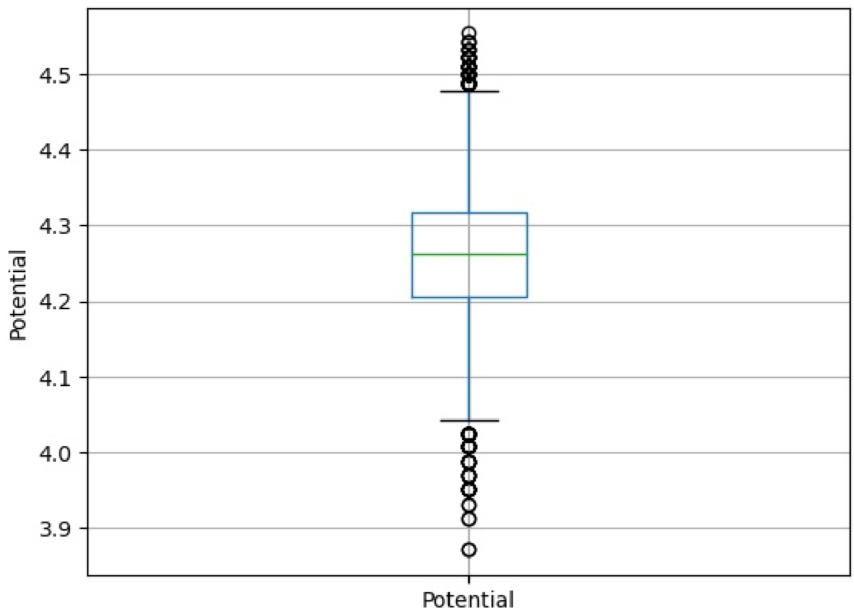


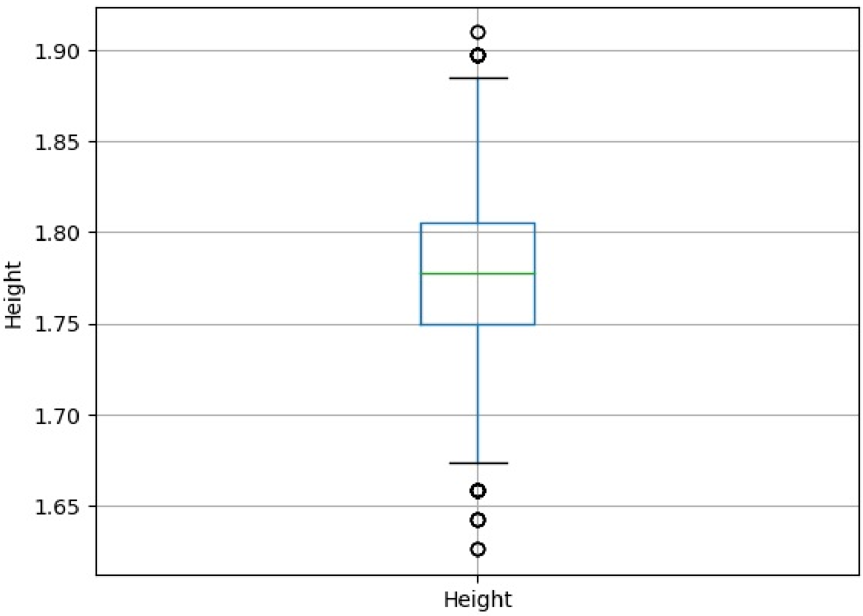
In [16]:

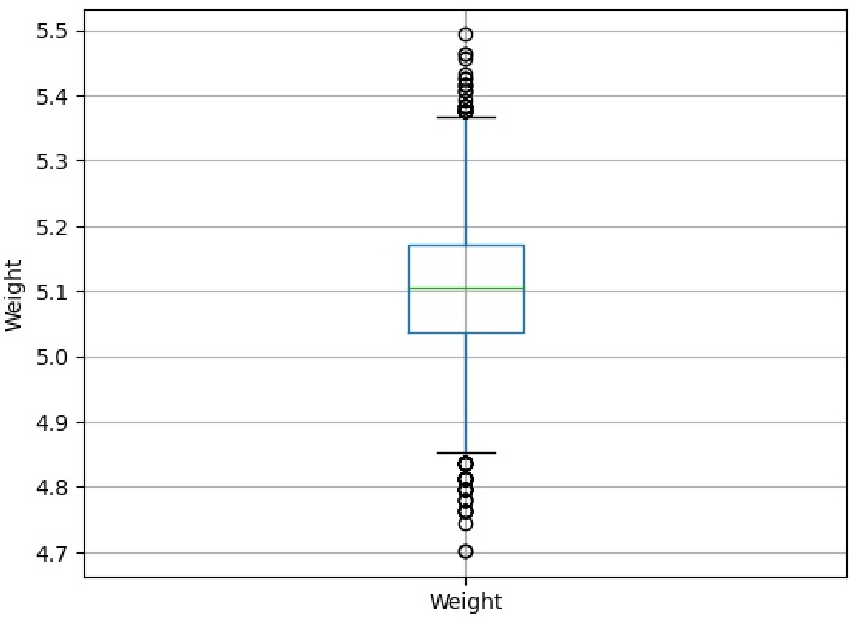
```
# eliminataeing outer layers
for features in contineous_features:
    df=df.copy()
    if 0 in df[features].unique():
        pass
    else:
        df[features]=np.log(df[features])
        df.boxplot(column=features)
        plt.ylabel(features)
        plt.show()
```

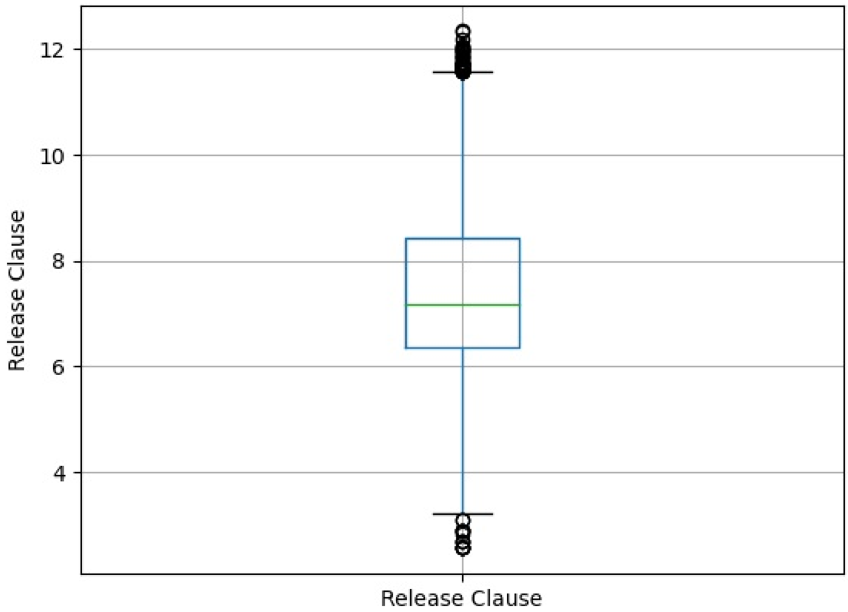



4.5
4.4
4.3
4.2
4.1
4.0
3.9
3.8
Overall
Overall


4.5
4.4
4.3
4.2
4.1
4.0
3.9
Potential
Potential

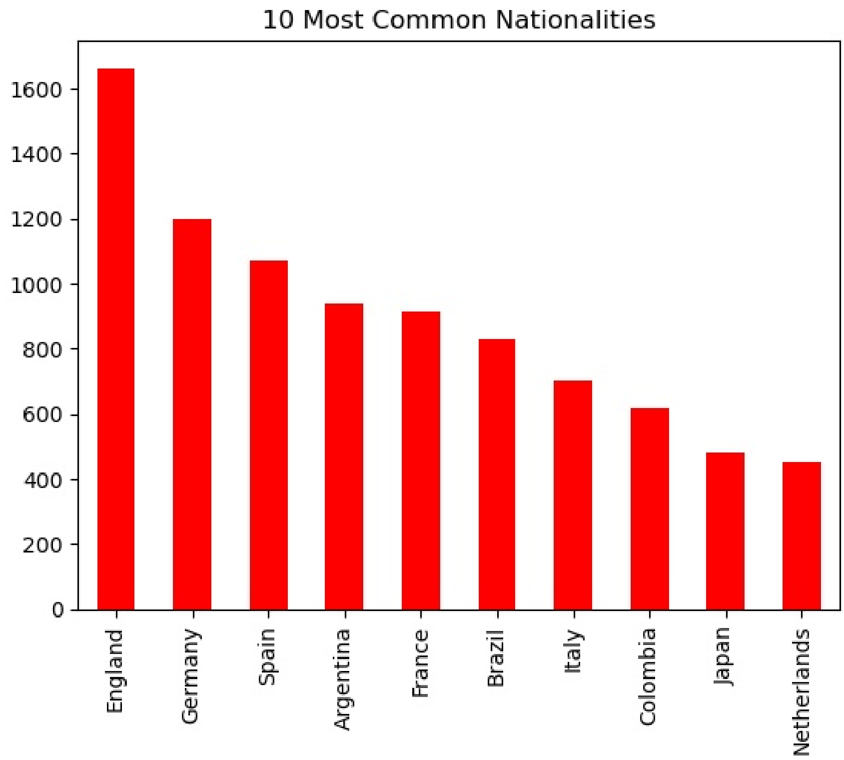

1.90
1.85
1.80
1.75
1.70
1.65
Height
Height


5.5
5.4
5.3
5.2
5.1
5.0
4.9
4.8
4.7
Weight
Weight



In [17]:
```
#10 Most Common Nationalities


national = df['Nationality'].value_counts()[:10]
national.plot.bar(cmap ='prism')
plt.title('10 Most Common Nationalities')
```

Out[17]:
```
Text(0.5, 1.0, '10 Most Common Nationalities')
```

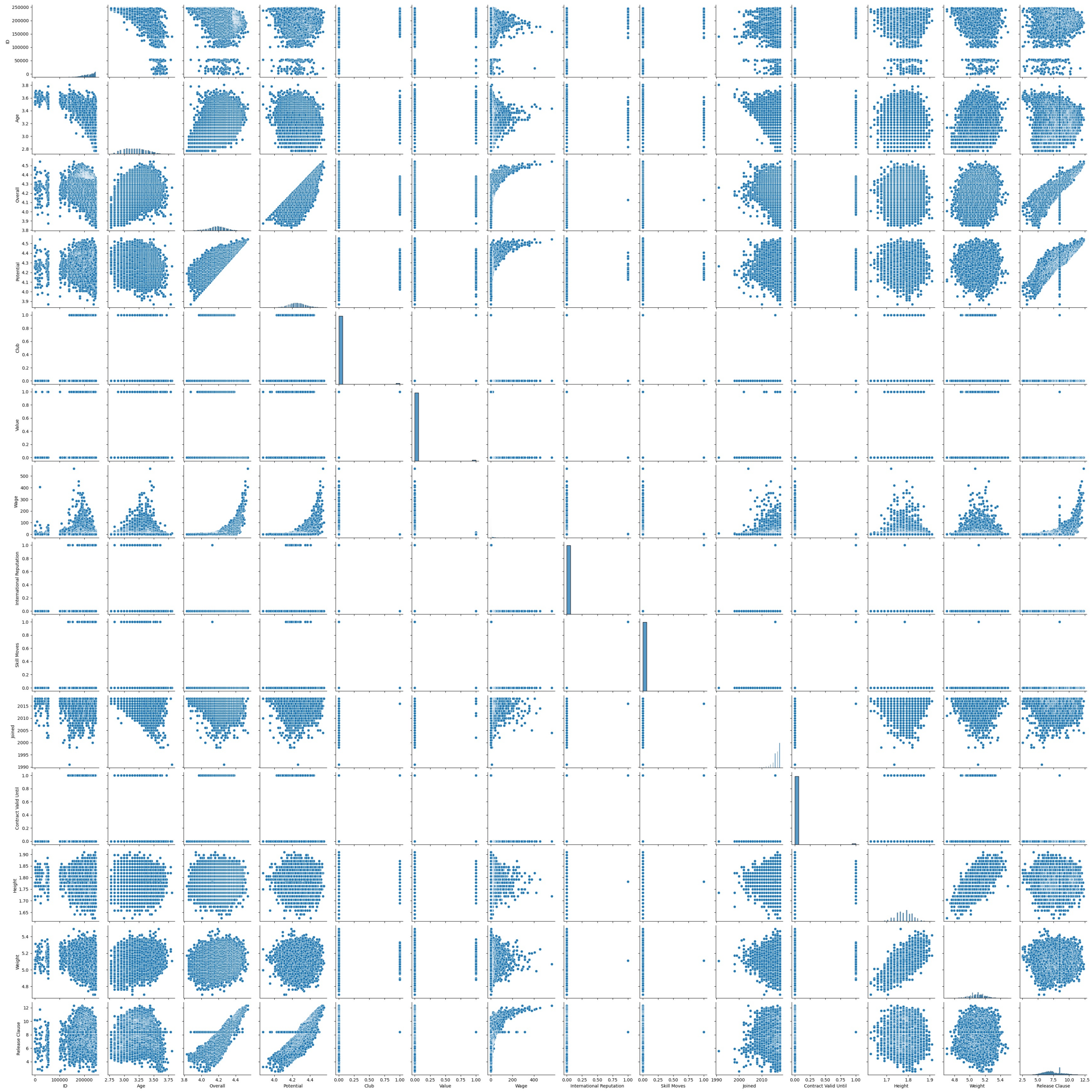


In [29]:

```
#England player who has potential greater than 90

eng = df[(df.Nationality == 'England') & (df.Potential >= 90)]
sns.pairplot(df)
```

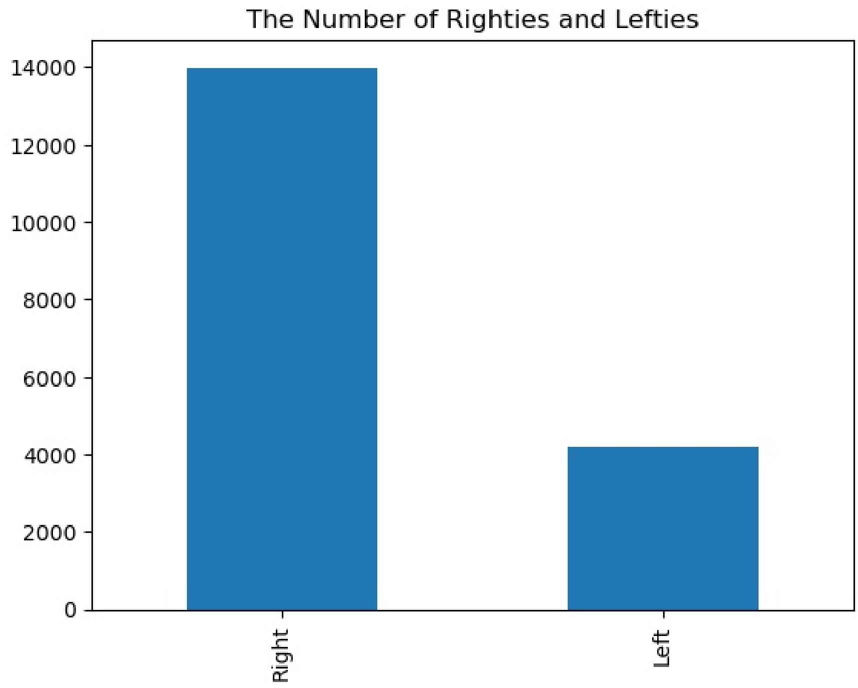
Out[29]:

```
<seaborn.axisgrid.PairGrid at 0x2c6af07a5e0>
```



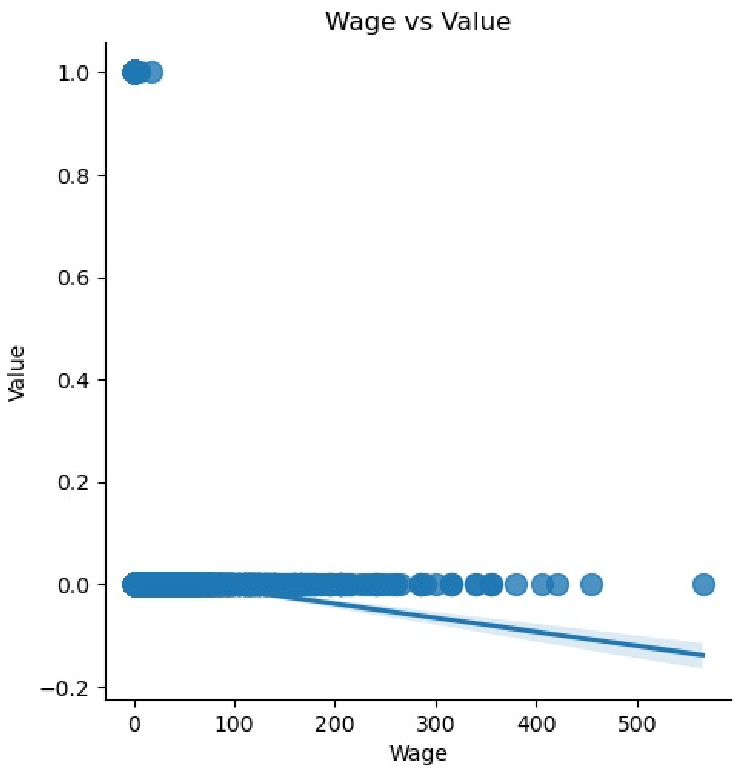
In [20]:
```python
#The Number of Righties and Lefties
pf = df['Preferred Foot'].value_counts()
pf.plot.bar()
plt.title('The Number of Righties and Lefties')
```

Out[20]:
```
Text(0.5, 1.0, 'The Number of Righties and Lefties')
```

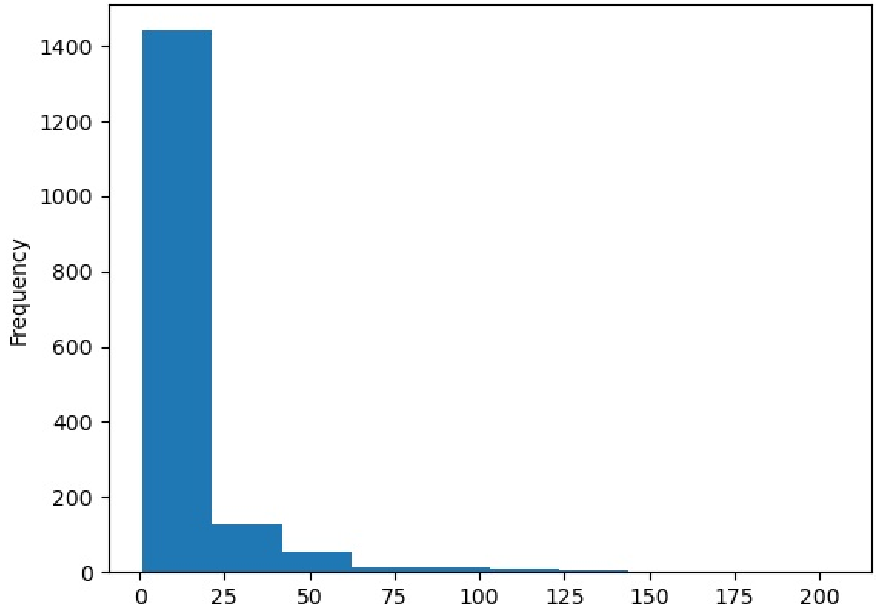


In [83]:
```
#The List of Player Who Has Potential Greater Than 90

potential = df[df['Potential']>20]
potential["Name"]
```
Out[83]: `Series([], Name: Name, dtype: object)`

In [58]:
```
## wages vs salary
sns.lmplot(x='Wage',y='Value',data=df,scatter_kws={'s':100})
plt.title('Wage vs Value')
```
Out[58]: `Text(0.5, 1.0, 'Wage vs Value')`



In [36]:
```
#British Player Salary in k

bsalary = df.loc[df.Nationality=='England','Wage']
eng['Name']
bsalary.plot.hist()
```
Out[36]: `<AxesSubplot:ylabel='Frequency'>`



In [79]:
```
#British Player with single International Reputation
engIR = df[(df['Nationality']=='England') & (df['International Reputation']==1)]
engIR['Name']
```

Out[79]:
```
13238      J. Stead
13240    R. Bingham
13243     M. Feeney
13256     R. Deacon
13265    D. Gardner
Name: Name, dtype: object
```

In [ ]:

In [84]:
```
potential = df[df['Potential']>90]
potential["Name"]
```

Out[84]:
```
Series([], Name: Name, dtype: object)
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js