In [ ]:
```python
#Loading the libraries

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sqlite3



pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

In [1]:
```python
#Creating 1d numpy Array

arr = np.array([1, 2, 3, 4, 5])
print(arr)
print(type(arr))
```

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

In [2]:
```python
#Creating 2d numpy Array

x = np.array([[25, 95, 33, 29, 28],[19, 91, 17, 40, 30],[38, 87, 13, 87, 18],[65, 97, 20, 16, 28], [8, 89, 70,
print(x)
print(type(x))
```

```
[[25 95 33 29 28]
 [19 91 17 40 30]
 [38 87 13 87 18]
 [65 97 20 16 28]
 [ 8 89 70 25 99]]
<class 'numpy.ndarray'>
```

Numpy

In [3]:
```python
#How many numbers are greater than 12?

c = 0
for i in range(len(x)):
    for j in range(len(x[i])):
        if x[i][j] > 12:
            c += 1
        else:
            continue
print(c)
```

```
24
```

In [4]:
```python
#How many numbers are less than 10?

c = 0
for i in range(len(x)):
    for j in range(len(x[i])):
        if x[i][j] < 10:
            c += 1
        else:
            continue
print(c)
```

```
1
```

In [5]:
```python
#Add 5 to the array

print(x + 5)
```

```
[[ 30 100  38  34  33]
 [ 24  96  22  45  35]
 [ 43  92  18  92  23]
 [ 70 102  25  21  33]
 [ 13  94  75  30 104]]
```

In [6]:
```python
#Subtract 2 from the array

print(x - 2)
```

```
[[23 93 31 27 26]
 [17 89 15 38 28]
 [36 85 11 85 16]
 [63 95 18 14 26]
 [ 6 87 68 23 97]]
```

In [7]:
```python
#Multiple 5 to the array

print(x*5)
```

```
[[125 475 165 145 140]
 [ 95 455  85 200 150]
 [190 435  65 435  90]
 [325 485 100  80 140]
 [ 40 445 350 125 495]]
```

In [8]:
```python
#Divide the array by 3

print(x//3)
```

```
[[ 8 31 11  9  9]
 [ 6 30  5 13 10]
 [12 29  4 29  6]
 [21 32  6  5  9]
 [ 2 29 23  8 33]]
```

Now let's look at manipulating the arrays more:

In [10]:
```python
# array for reference
print(x)
```

```
[[25 95 33 29 28]
 [19 91 17 40 30]
 [38 87 13 87 18]
 [65 97 20 16 28]
 [ 8 89 70 25 99]]
```

In [9]:
```python
print(x[1:3, 1:3])
```

```
[[91 17]
 [87 13]]
```

In [11]:
```python
print(x[2:3, 1:])
```

```
[[87 13 87 18]]
```

In [12]:
```python
print(x[3:4, 2:])
```

```
[[20 16 28]]
```

In [13]:
```python
print(x[1:4, 2:])
```

```
[[17 40 30]
 [13 87 18]
 [20 16 28]]
```

Let's play with some statistics:

In [14]:
```python
#What is the smallest number in the 3rd column?

x[:,2:3].min()
```

Out[14]: 13

In [15]:
```python
#What is the average of the second row?

x[1:2,:].mean()
```

Out[15]: 39.4

In [16]:
```python
#What is the median of the 5th row?

np.median(x[4:,:])
```

Out[16]: 70.0

In [86]:
```python
# What is the standard deviation of the 4th row starting at the 3rd element?

print(x[3:4,1:].std())
```

Out[86]: 33.04826016600571

Pandas

In [30]:
```python
#Loafing the database into dataframe
c = sqlite3.connect('database.sqlite')

# List all tables in the database
data = pd.read_sql_query("SELECT * FROM sqlite_master WHERE type='table'",c)

# Output dataframe
data
```

Out[30]:

	type	name	tbl_name	rootpage	sql
0	table	sqlite_sequence	sqlite_sequence	4	CREATE TABLE sqlite_sequence(name,seq)
1	table	Player_Attributes	Player_Attributes	11	CREATE TABLE "Player_Attributes" (\n\t\tIN...
2	table	Player	Player	14	CREATE TABLE $Player$ (\n\t\tINTEGER PRIMA...
3	table	Match	Match	18	CREATE TABLE $Match$ (\n\t\tINTEGER PRIMAR...
4	table	League	League	24	CREATE TABLE $League$ (\n\t\tINTEGER PRIMA...
5	table	Country	Country	26	CREATE TABLE $Country$ (\n\t\tINTEGER PRIM...
6	table	Team	Team	29	CREATE TABLE "Team" (\n\t\tINTEGER PRIMARY...
7	table	Team_Attributes	Team_Attributes	2	CREATE TABLE $Team_Aributes$ (\n\t\tINTE...

Player Tables

In [31]:
```python
players = pd.read_sql_query("SELECT * FROM Player",c)
```

In [82]:
```python
#Print all rows

players
```

Out[82]:

	id	player_api_id	player_name	player_fifa_api_id	birthday	height	weight
0	1	505942	Aaron Appindangoye	218353	1992-02-29 00:00:00	182.88	187
1	2	155782	Aaron Cresswell	189615	1989-12-15 00:00:00	170.18	146
2	3	162549	Aaron Doran	186170	1991-05-13 00:00:00	170.18	163
3	4	30572	Aaron Galindo	140161	1982-05-08 00:00:00	182.88	198
4	5	23780	Aaron Hughes	17725	1979-11-08 00:00:00	182.88	154
...	...	...	...	...	...	...	...
11055	11071	26357	Zoumana Camara	2488	1979-04-03 00:00:00	182.88	168
11056	11072	111182	Zsolt Laczko	164680	1986-12-18 00:00:00	182.88	176
11057	11073	36491	Zsolt Low	111191	1979-04-29 00:00:00	180.34	154
11058	11074	35506	Zurab Khizanishvili	47058	1981-10-06 00:00:00	185.42	172
11059	11075	39902	Zvjezdan Misimovic	102359	1982-06-05 00:00:00	180.34	176

11060 rows × 7 columns

In [81]:
```python
print(players)
```

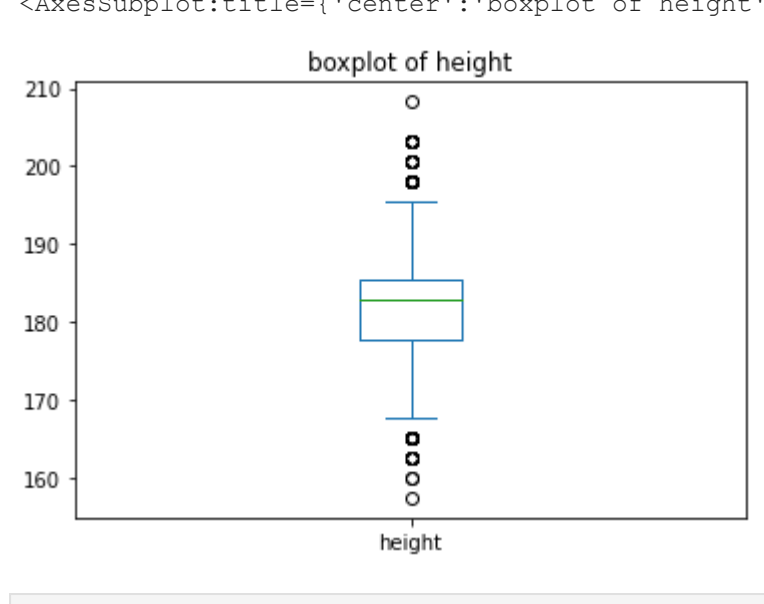
```
          id  player_api_id          player_name  player_fifa_api_id             birthday  height  weight
0          1         505942   Aaron Appindangoye              218353  1992-02-29 00:00:00  182.88     187
1          2         155782      Aaron Cresswell              189615  1989-12-15 00:00:00  170.18     146
2          3         162549          Aaron Doran              186170  1991-05-13 00:00:00  170.18     163
3          4          30572        Aaron Galindo              140161  1982-05-08 00:00:00  182.88     198
4          5          23780         Aaron Hughes               17725  1979-11-08 00:00:00  182.88     154
...      ...            ...                  ...                 ...                  ...     ...     ...
11055  11071          26357       Zoumana Camara                2488  1979-04-03 00:00:00  182.88     168
11056  11072         111182         Zsolt Laczko              164680  1986-12-18 00:00:00  182.88     176
11057  11073          36491            Zsolt Low              111191  1979-04-29 00:00:00  180.34     154
11058  11074          35506  Zurab Khizanishvili               47058  1981-10-06 00:00:00  185.42     172
11059  11075          39902   Zvjezdan Misimovic              102359  1982-06-05 00:00:00  180.34     176

[11060 rows x 7 columns]
```

In [33]:
```python
#Print the dataframe schema

players.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11060 entries, 0 to 11059
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id                  11060 non-null  int64
 1   player_api_id       11060 non-null  int64
 2   player_name         11060 non-null  object
 3   player_fifa_api_id  11060 non-null  int64
 4   birthday            11060 non-null  object
 5   height              11060 non-null  float64
 6   weight              11060 non-null  int64
dtypes: float64(1), int64(4), object(2)
memory usage: 605.0+ KB
```

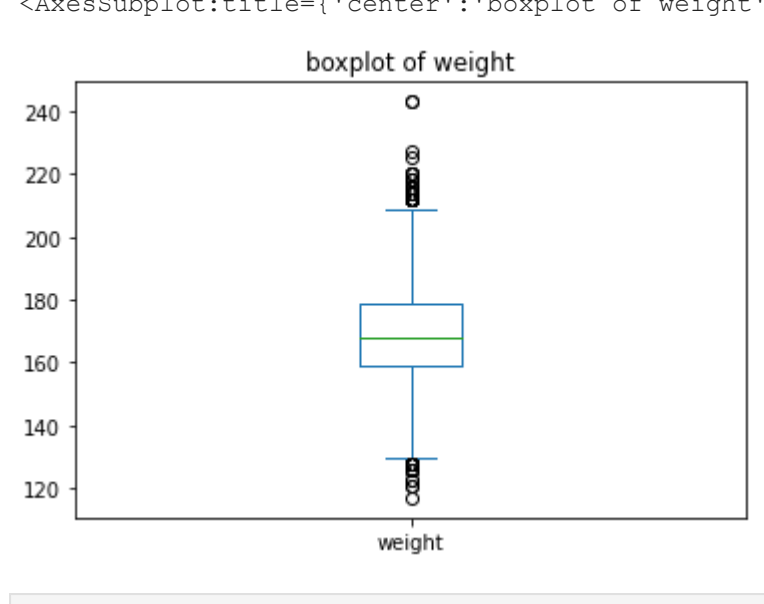
In [34]:
```python
# Create a Boxplot  all players heights
players['height'].plot(kind= 'box', title= 'boxplot of height')
```

Out[34]: <AxesSubplot:title={'center':'boxplot of height'}>



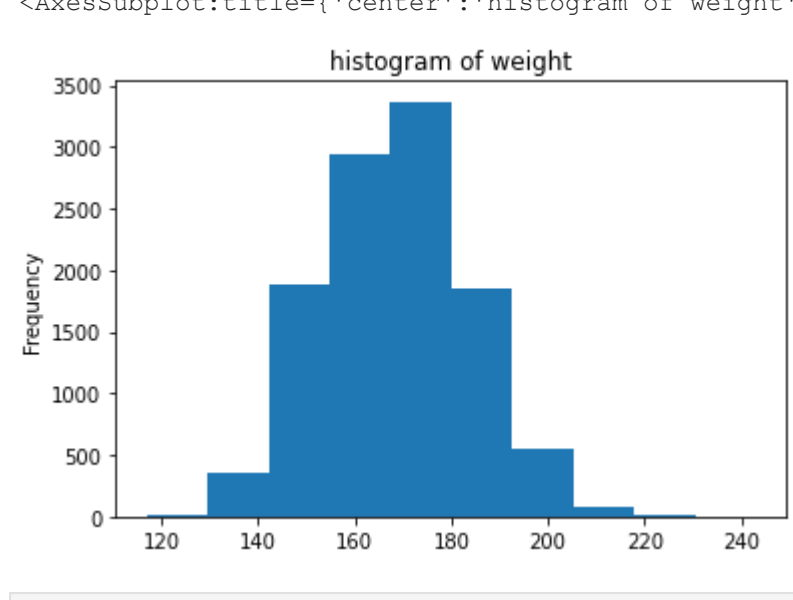
In [35]:
```python
# Create a Boxplot  all players weights

players['weight'].plot(kind= 'box', title= 'boxplot of weight')
```

Out[35]: <AxesSubplot:title={'center':'boxplot of weight'}>



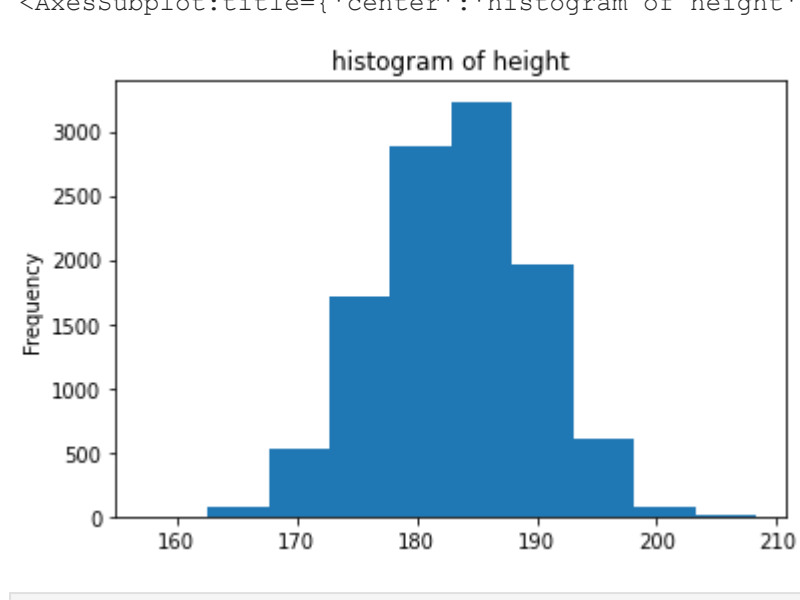
In [36]:
```python
## Create a Histogram  all players weights

players['weight'].plot(kind= 'hist', title= 'histogram of weight')
```

Out[36]: <AxesSubplot:title={'center':'histogram of weight'}, ylabel='Frequency'>



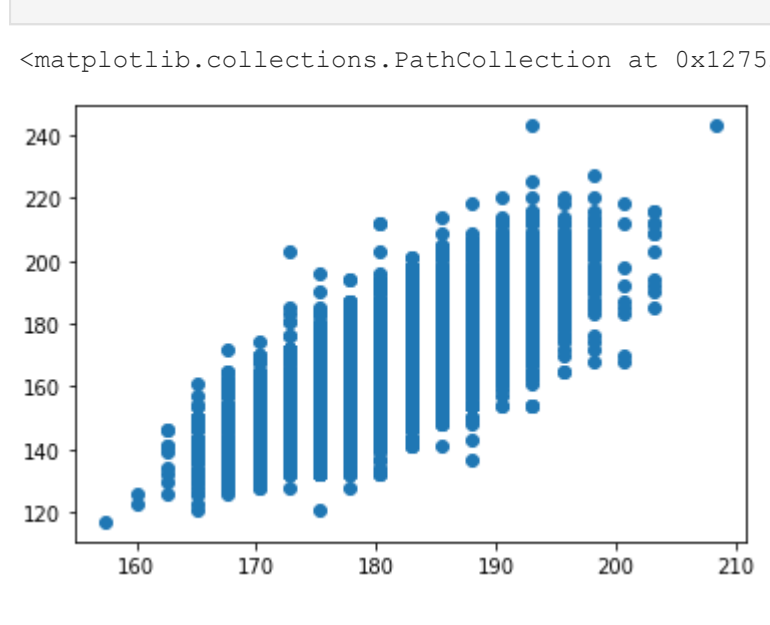
In [37]:
```python
### Create a Histogram  all players heights

players['height'].plot(kind= 'hist', title= 'histogram of height')
```

Out[37]: <AxesSubplot:title={'center':'histogram of height'}, ylabel='Frequency'>



In [38]:
```python
plt.scatter(x=players['height'], y=players['weight'])
```

Out[38]: <matplotlib.collections.PathCollection at 0x127520b80>



Player Attributes Table

In [39]:
```python
#Loading the data

player_a = pd.read_sql_query("SELECT * FROM Player_Attributes",c)
```

In [40]:
```python
# Print the first 10 rows

player_a.head(10)
```

Out[40]:

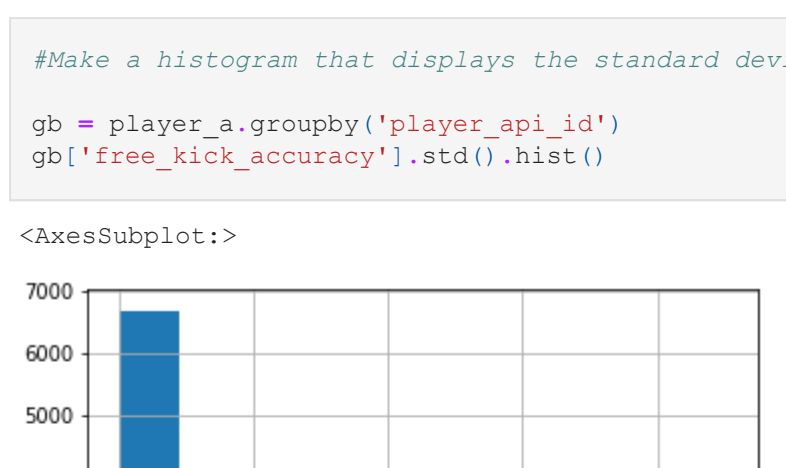
	id	player_fifa_api_id	player_api_id	date	overall_rating	potential	preferred_foot	attacking_work_rate	defensive_work_rate
0	1	218353	505942	2016-02-18 00:00:00	67.0	71.0	right	medium	medium
1	2	218353	505942	2015-11-19 00:00:00	67.0	71.0	right	medium	medium
2	3	218353	505942	2015-09-21 00:00:00	62.0	66.0	right	medium	medium
3	4	218353	505942	2015-03-20 00:00:00	61.0	65.0	right	medium	medium
4	5	218353	505942	2007-02-22 00:00:00	61.0	65.0	right	medium	medium
5	6	189615	155782	2016-04-21 00:00:00	74.0	76.0	left	high	medium
6	7	189615	155782	2016-04-07 00:00:00	74.0	76.0	left	high	medium
7	8	189615	155782	2016-01-07 00:00:00	73.0	75.0	left	high	medium
8	9	189615	155782	2015-12-24 00:00:00	73.0	75.0	left	high	medium
9	10	189615	155782	2015-12-17 00:00:00	73.0	75.0	left	high	medium

In [41]:
```python
# Print the schema

player_a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 183978 entries, 0 to 183977
Data columns (total 42 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   id                   183978 non-null  int64
 1   player_fifa_api_id   183978 non-null  int64
 2   player_api_id        183978 non-null  int64
 3   date                 183978 non-null  object
 4   overall_rating       183142 non-null  float64
 5   potential            183142 non-null  float64
 6   preferred_foot       183142 non-null  object
 7   attacking_work_rate  180748 non-null  object
 8   defensive_work_rate  183142 non-null  object
 9   crossing             183142 non-null  float64
 10  finishing            183142 non-null  float64
 11  heading_accuracy     183142 non-null  float64
 12  short_passing        183142 non-null  float64
 13  volleys              181265 non-null  float64
 14  dribbling            183142 non-null  float64
 15  curve                181265 non-null  float64
 16  free_kick_accuracy   183142 non-null  float64
 17  long_passing         183142 non-null  float64
 18  ball_control         183142 non-null  float64
 19  acceleration         183142 non-null  float64
 20  sprint_speed         183142 non-null  float64
 21  agility              181265 non-null  float64
 22  reactions            183142 non-null  float64
 23  balance              181265 non-null  float64
 24  shot_power           183142 non-null  float64
 25  jumping              181265 non-null  float64
 26  stamina              183142 non-null  float64
 27  strength             183142 non-null  float64
 28  long_shots           183142 non-null  float64
 29  aggression           183142 non-null  float64
 30  interceptions        183142 non-null  float64
 31  positioning          183142 non-null  float64
 32  vision               181265 non-null  float64
 33  penalties            183142 non-null  float64
 34  marking              183142 non-null  float64
 35  standing_tackle      183142 non-null  float64
 36  sliding_tackle       181265 non-null  float64
 37  gk_diving            183142 non-null  float64
 38  gk_handling          183142 non-null  float64
 39  gk_kicking           183142 non-null  float64
 40  gk_positioning       183142 non-null  float64
 41  gk_reflexes          183142 non-null  float64
dtypes: float64(35), int64(3), object(4)
memory usage: 59.0+ MB
```

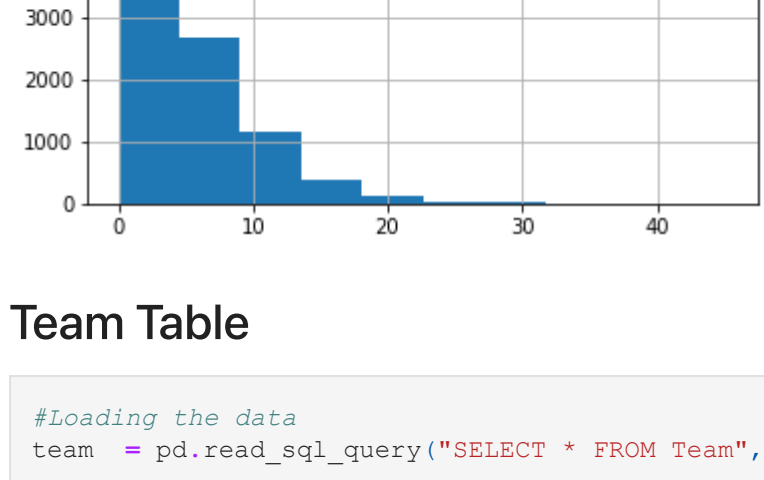
In [42]:
```python
#Make a histogram visualizing the penalties of all players

player_a.hist(column= 'penalties')
```

Out[42]: array([[<AxesSubplot:title={'center':'penalties'}>]], dtype=object)



In [71]:
```python
#Make a histogram that displays the standard deviation of free_kick_accuracy grouped by player_api_id .

gb = player_a.groupby('player_api_id')
gb['free_kick_accuracy'].std().hist()
```

Out[71]: <AxesSubplot:>



Team Table

In [48]:
```python
#Loading the data
team  = pd.read_sql_query("SELECT * FROM Team",c)
```

In [49]:
```python
#Print top 5 rows

team.head(5)
```

Out[49]:

	id	team_api_id	team_fifa_api_id	team_long_name	team_short_name
0	1	9987	673.0	KRC Genk	GEN
1	2	9993	675.0	Beerschot AC	BAC
2	3	10000	15005.0	SV Zulte-Waregem	ZUL
3	4	9994	2007.0	Sporting Lokeren	LOK
4	5	9984	1750.0	KSV Cercle Brugge	CEB

In [50]:
```python
#Print the schema

team.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   id                299 non-null    int64
 1   team_api_id       299 non-null    int64
 2   team_fifa_api_id  288 non-null    float64
 3   team_long_name    299 non-null    object
 4   team_short_name   299 non-null    object
dtypes: float64(1), int64(2), object(2)
memory usage: 11.8+ KB
```

In [53]:
```python
#Function to find the missing values in data Frame

def missing_values_table(df):
    mis_val = df.isnull().sum()
    mis_val_percent = 100 * df.isnull().sum() / len(df)
    mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)
    mis_val_table_ren_columns = mis_val_table.rename(
    columns = {0 : 'Missing Values', 1 : '% of Total Values'})
    mis_val_table_ren_columns = mis_val_table_ren_columns[
        mis_val_table_ren_columns.iloc[:,1] != 0].sort_values(
    '% of Total Values', ascending=False).round(1)
    print ("Your selected dataframe has " + str(df.shape[1]) + " columns.\n"
        "There are " + str(mis_val_table_ren_columns.shape[0]) +
          " columns that have missing values.")
    return mis_val_table_ren_columns
```

In [54]:
```python
missing_values_table(team)
```

```
Your selected dataframe has 5 columns.
There are 1 columns that have missing values.
```

Out[54]:

	Missing Values	% of Total Values
team_fifa_api_id	11	3.7

In [55]:
```python
# Replace all null values with zeros (0s) in the team_api_id and team_fifa_api_id columns.

team['team_api_id'] = team['team_api_id'].fillna(0)
team['team_fifa_api_id'] = team['team_fifa_api_id'].fillna(0)
```

In [57]:
```python
missing_values_table(team)
```

```
Your selected dataframe has 5 columns.
There are 0 columns that have missing values.
```

Out[57]:

Missing Values	% of Total Values

Team Attributes Table

In [64]:
```python
#Loading the data

team_a  = pd.read_sql_query("SELECT * FROM Team_Attributes",c)
```

In [65]:
```python
#Print the top 10 rows

team_a.head(10)
```

Out[65]:

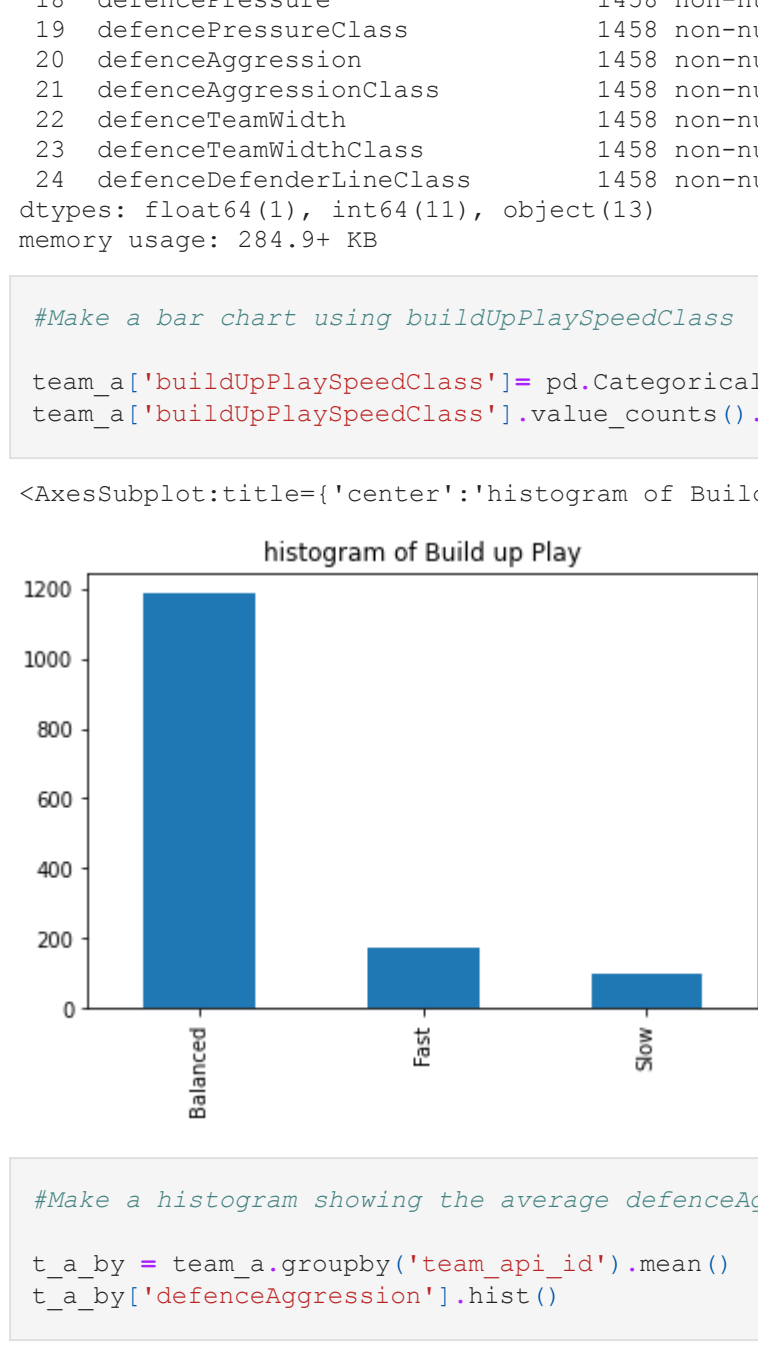
	id	team_fifa_api_id	team_api_id	date	buildUpPlaySpeed	buildUpPlaySpeedClass	buildUpPlayDribbling	buildUpPlayDribblingC
0	1	434	9930	2010-02-22 00:00:00	60	Balanced	NaN	
1	2	434	9930	2014-09-19 00:00:00	52	Balanced	48.0	No
2	3	434	9930	2015-09-10 00:00:00	47	Balanced	41.0	No
3	4	77	8485	2010-02-22 00:00:00	70	Fast	NaN	
4	5	77	8485	2011-02-22 00:00:00	47	Balanced	NaN	
5	6	77	8485	2012-02-22 00:00:00	58	Balanced	NaN	
6	7	77	8485	2013-09-20 00:00:00	62	Balanced	NaN	
7	8	77	8485	2014-09-19 00:00:00	58	Balanced	64.0	No
8	9	77	8485	2015-09-10 00:00:00	59	Balanced	64.0	No
9	10	614	8576	2010-02-22 00:00:00	60	Balanced	NaN	

In [61]:
```python
#Print the schema

team_a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1458 entries, 0 to 1457
Data columns (total 25 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              1458 non-null   int64
 1   team_fifa_api_id                1458 non-null   int64
 2   team_api_id                     1458 non-null   int64
 3   date                            1458 non-null   object
 4   buildUpPlaySpeed                1458 non-null   int64
 5   buildUpPlaySpeedClass           1458 non-null   object
 6   buildUpPlayDribbling            489 non-null    float64
 7   buildUpPlayDribblingClass       1458 non-null   object
 8   buildUpPlayPassing              1458 non-null   int64
 9   buildUpPlayPassingClass         1458 non-null   object
 10  buildUpPlayPositioningClass     1458 non-null   object
 11  chanceCreationPassing           1458 non-null   int64
 12  chanceCreationPassingClass      1458 non-null   object
 13  chanceCreationCrossing          1458 non-null   int64
 14  chanceCreationCrossingClass     1458 non-null   object
 15  chanceCreationShooting          1458 non-null   int64
 16  chanceCreationShootingClass     1458 non-null   object
 17  chanceCreationPositioningClass  1458 non-null   object
 18  defencePressure                 1458 non-null   int64
 19  defencePressureClass            1458 non-null   object
 20  defenceAggression               1458 non-null   int64
 21  defenceAggressionClass          1458 non-null   object
 22  defenceTeamWidth                1458 non-null   int64
 23  defenceTeamWidthClass           1458 non-null   object
 24  defenceDefenderLineClass        1458 non-null   object
dtypes: float64(1), int64(11), object(13)
memory usage: 284.9+ KB
```

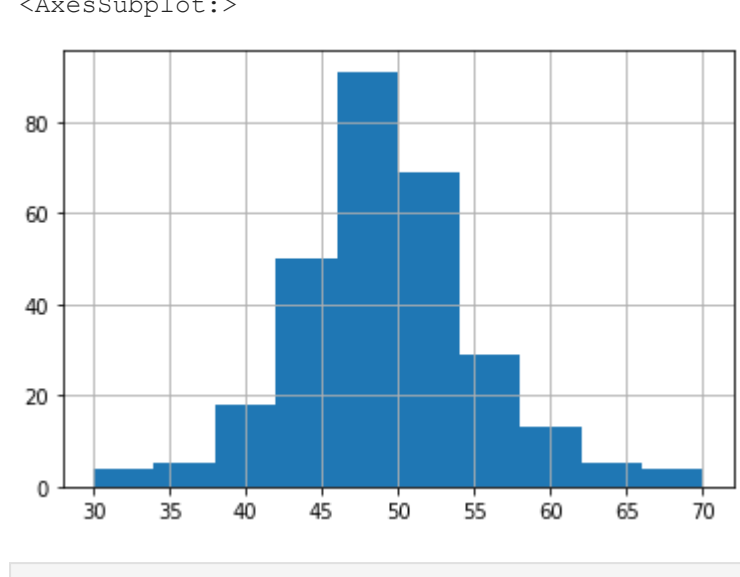
In [67]:
```python
#Make a bar chart using buildUpPlaySpeedClass

team_a['buildUpPlaySpeedClass']= pd.Categorical(team_a['buildUpPlaySpeedClass'])
team_a['buildUpPlaySpeedClass'].value_counts().plot(kind= 'bar', title= 'histogram of Build up Play')
```

Out[67]: <AxesSubplot:title={'center':'histogram of Build up Play'}>



In [80]:
```python
#Make a histogram showing the average defenceAggression grouped by team-api_id.

t_a_by = team_a.groupby('team_api_id').mean()
t_a_by['defenceAggression'].hist()
```

Out[80]: <AxesSubplot:>



In [ ]:
```python
!jupyter-nbconvert --to PDFviaHTML Assignment2_Sharanbasav.ipynb
```