Exploratory Data Analysis on Olympic Data

- Name: Prathamesh Rajesh Sonar.
- Project_name: Exploratory data analyis on Olympic_data (40mb).
- Libary_use: pandas,matplotlib.pyplot,plotly.express

In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv('Olympic_data.csv')
df
```

Out[2]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Seasor
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summe
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summe
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summe
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summe
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winte
...	...	...	...	...	...	...	...	...	...	...	.
271111	135569	Andrzej ya	M	29.0	179.0	89.0	Poland-1	POL	1976 Winter	1976	Winte
271112	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winte
271113	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winte
271114	135571	Tomasz Ireneusz ya	M	30.0	185.0	96.0	Poland	POL	1998 Winter	1998	Winte
271115	135571	Tomasz Ireneusz ya	M	34.0	185.0	96.0	Poland	POL	2002 Winter	2002	Winte

271116 rows × 15 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   ID      271116 non-null  int64  
 1   Name    271116 non-null  object 
 2   Sex     271116 non-null  object 
 3   Age     261642 non-null  float64
 4   Height  210945 non-null  float64
 5   Weight  208241 non-null  float64
 6   Team    271116 non-null  object 
 7   NOC     271116 non-null  object 
 8   Games   271116 non-null  object 
 9   Year    271116 non-null  int64  
 10  Season  271116 non-null  object 
 11  City    271116 non-null  object 
 12  Sport   271116 non-null  object 
 13  Event   271116 non-null  object 
 14  Medal   39783 non-null   object 
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB
```

In [4]:
```
df.describe()
```

Out[4]:

	ID	Age	Height	Weight	Year
count	271116.000000	261642.000000	210945.000000	208241.000000	271116.000000
mean	68248.954396	25.556898	175.338970	70.702393	1978.378480
std	39022.286345	6.393561	10.518462	14.348020	29.877632
min	1.000000	10.000000	127.000000	25.000000	1896.000000
25%	34643.000000	21.000000	168.000000	60.000000	1960.000000
50%	68205.000000	24.000000	175.000000	70.000000	1988.000000
75%	102097.250000	28.000000	183.000000	79.000000	2002.000000
max	135571.000000	97.000000	226.000000	214.000000	2016.000000

- The minimum age 10 Years and maximun age 97 Years is correct.

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
ID             0
Name           0
Sex            0
Age         9474
Height     60171
Weight     62875
Team           0
NOC            0
Games          0
Year           0
Season         0
City           0
Sport          0
Event          0
Medal     231333
dtype: int64
```

1. medal can be replace by 0.

2. age,height and weight can be replace with average values.

In [6]:
```
df['Medal']=df.Medal.fillna('no')
df['Age']=df.Age.fillna(25)
df['Height']=df.Height.fillna(175)
df['Weight']=df.Weight.fillna(70)
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
ID        0
Name      0
Sex       0
Age       0
Height    0
Weight    0
Team      0
NOC       0
Games     0
Year      0
Season    0
City      0
Sport     0
Event     0
Medal     0
dtype: int64
```

In [8]:
```
df = df.drop(['Name','Games','Event','ID','City'],axis=1)
df
```

Out[8]:

	Sex	Age	Height	Weight	Team	NOC	Year	Season	Sport	Medal
0	M	24.0	180.0	80.0	China	CHN	1992	Summer	Basketball	no
1	M	23.0	170.0	60.0	China	CHN	2012	Summer	Judo	no
2	M	24.0	175.0	70.0	Denmark	DEN	1920	Summer	Football	no
3	M	34.0	175.0	70.0	Denmark/Sweden	DEN	1900	Summer	Tug-Of-War	Gold
4	F	21.0	185.0	82.0	Netherlands	NED	1988	Winter	Speed Skating	no
...	...	...	...	...	...	...	...	...	...	...
271111	M	29.0	179.0	89.0	Poland-1	POL	1976	Winter	Luge	no
271112	M	27.0	176.0	59.0	Poland	POL	2014	Winter	Ski Jumping	no
271113	M	27.0	176.0	59.0	Poland	POL	2014	Winter	Ski Jumping	no
271114	M	30.0	185.0	96.0	Poland	POL	1998	Winter	Bobsleigh	no
271115	M	34.0	185.0	96.0	Poland	POL	2002	Winter	Bobsleigh	no

271116 rows × 10 columns

- we have Droped few columns which are not needed.

In [9]:
```
from sklearn.preprocessing import LabelEncoder
le1 = LabelEncoder()
le2 = LabelEncoder()
le3 = LabelEncoder()
df['le_sex'] =  le1.fit_transform(df.Sex)
df['le_medal']= le2.fit_transform(df.Medal)
df['le_season']= le3.fit_transform(df.Season)
df
```

Out[9]:

	Sex	Age	Height	Weight	Team	NOC	Year	Season	Sport	Medal	le_sex
0	M	24.0	180.0	80.0	China	CHN	1992	Summer	Basketball	no	1
1	M	23.0	170.0	60.0	China	CHN	2012	Summer	Judo	no	1
2	M	24.0	175.0	70.0	Denmark	DEN	1920	Summer	Football	no	1
3	M	34.0	175.0	70.0	Denmark/Sweden	DEN	1900	Summer	Tug-Of-War	Gold	1
4	F	21.0	185.0	82.0	Netherlands	NED	1988	Winter	Speed Skating	no	0
...	...	...	...	...	...	...	...	...	...	...	...
271111	M	29.0	179.0	89.0	Poland-1	POL	1976	Winter	Luge	no	1
271112	M	27.0	176.0	59.0	Poland	POL	2014	Winter	Ski Jumping	no	1
271113	M	27.0	176.0	59.0	Poland	POL	2014	Winter	Ski Jumping	no	1
271114	M	30.0	185.0	96.0	Poland	POL	1998	Winter	Bobsleigh	no	1
271115	M	34.0	185.0	96.0	Poland	POL	2002	Winter	Bobsleigh	no	1

271116 rows × 13 columns

- We have transform our string data into numeric data.

In [10]:
```
df.le_medal.unique()
```

Out[10]:
```
array([3, 1, 0, 2])
```

Medal :

- Gold = 1
- Silver = 2
- Bronze = 0
- Not get medal = 3

Sex :

- Male = 0
- Female = 1

Season :

- Summer = 0

- Winter = 1

In [11]:
```
df[df.duplicated()]
df.drop_duplicates(inplace=True)
df
```

Out[11]:

	Sex	Age	Height	Weight	Team	NOC	Year	Season	Sport	Medal	le_sex
0	M	24.0	180.0	80.0	China	CHN	1992	Summer	Basketball	no	1
1	M	23.0	170.0	60.0	China	CHN	2012	Summer	Judo	no	1
2	M	24.0	175.0	70.0	Denmark	DEN	1920	Summer	Football	no	1
3	M	34.0	175.0	70.0	Denmark/Sweden	DEN	1900	Summer	Tug-Of-War	Gold	1
4	F	21.0	185.0	82.0	Netherlands	NED	1988	Winter	Speed Skating	no	0
...	...	...	...	...	...	...	...	...	...	...	...
271110	F	33.0	171.0	69.0	Belarus	BLR	2016	Summer	Basketball	no	0
271111	M	29.0	179.0	89.0	Poland-1	POL	1976	Winter	Luge	no	1
271112	M	27.0	176.0	59.0	Poland	POL	2014	Winter	Ski Jumping	no	1
271114	M	30.0	185.0	96.0	Poland	POL	1998	Winter	Bobsleigh	no	1
271115	M	34.0	185.0	96.0	Poland	POL	2002	Winter	Bobsleigh	no	1

188894 rows × 13 columns

- Remove the dublicated

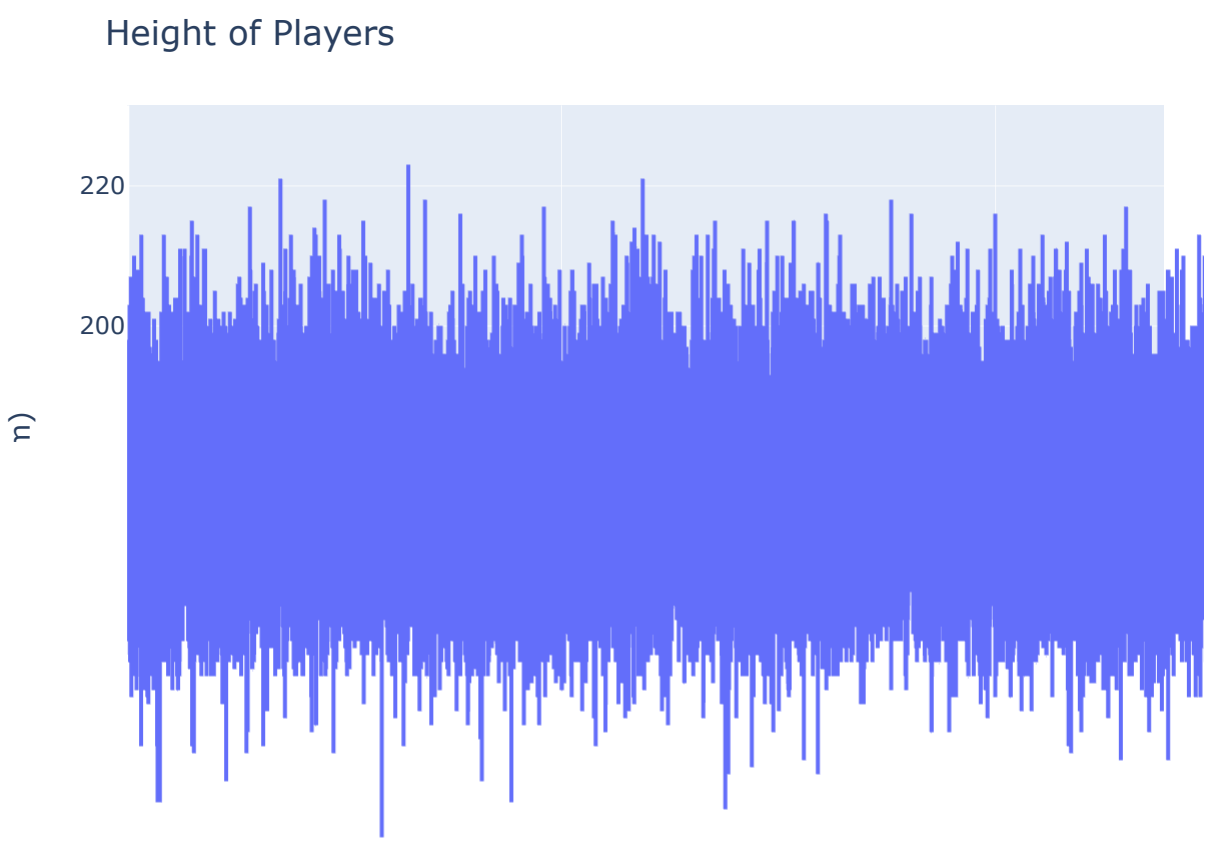
In [12]:
```
df.NOC.unique().shape
```

Out[12]:
```
(230,)
```

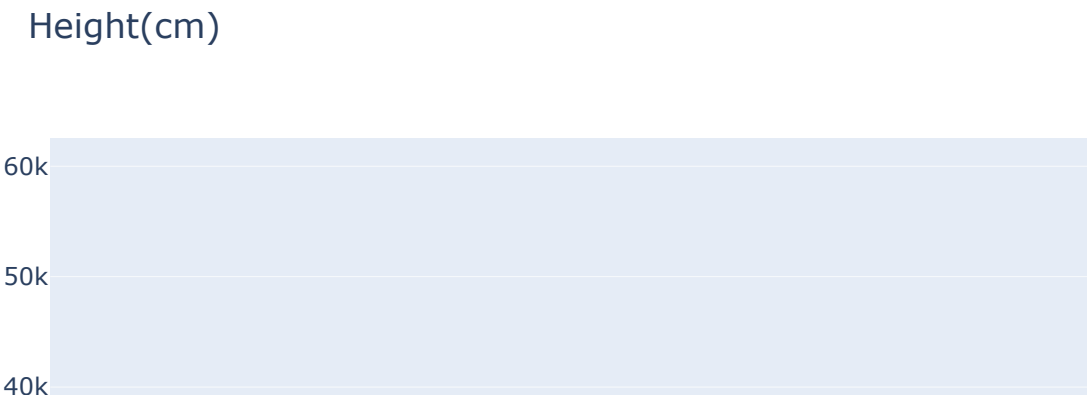
In [13]:
```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_style('darkgrid')
import plotly.express as px
```

In [14]:
```
import plotly.express as px
```

In [15]:
```
px.line(df.Height).update_layout(title="Height of Players",
    xaxis_title="count of players",
    yaxis_title="Height(cm)")
```

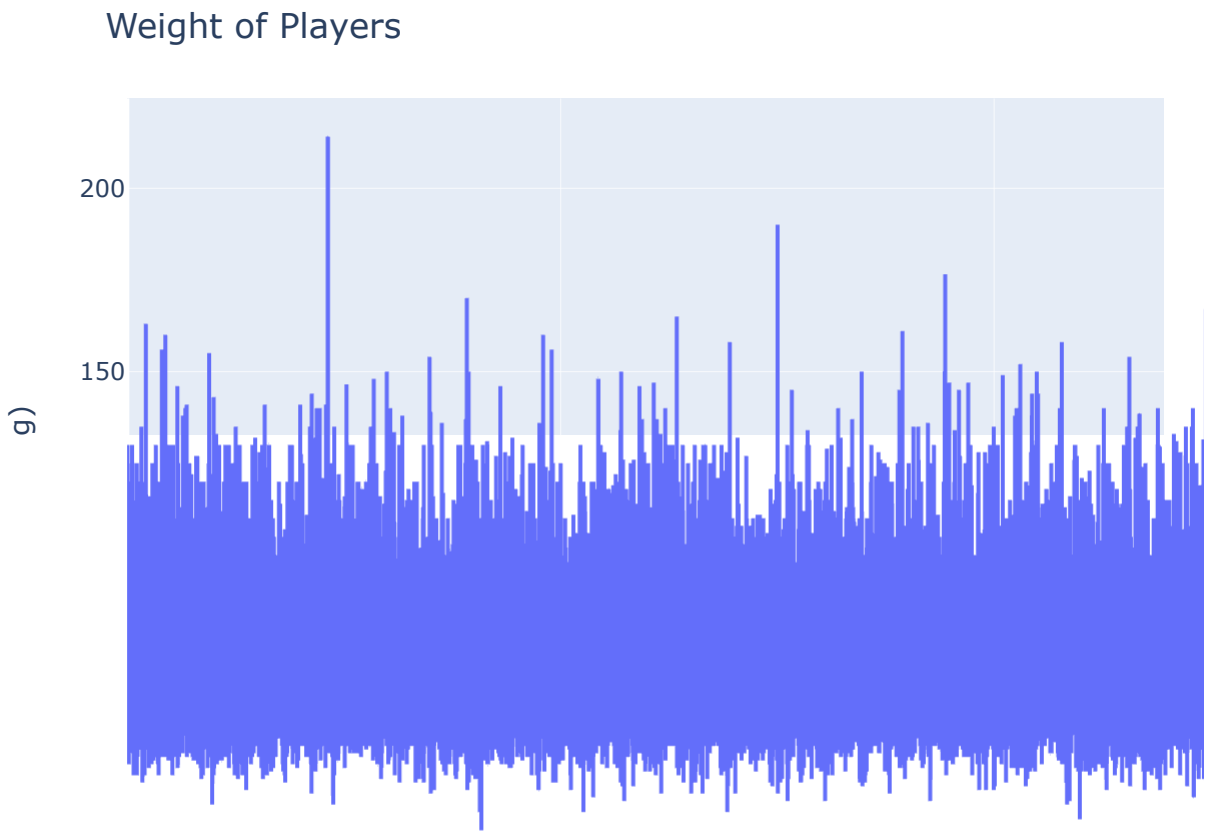


In [16]:
```
px.histogram(df.Height,nbins=30,title="Height(cm)").update_layout(
    xaxis_title="Height(cm)",
    yaxis_title="Count")
```

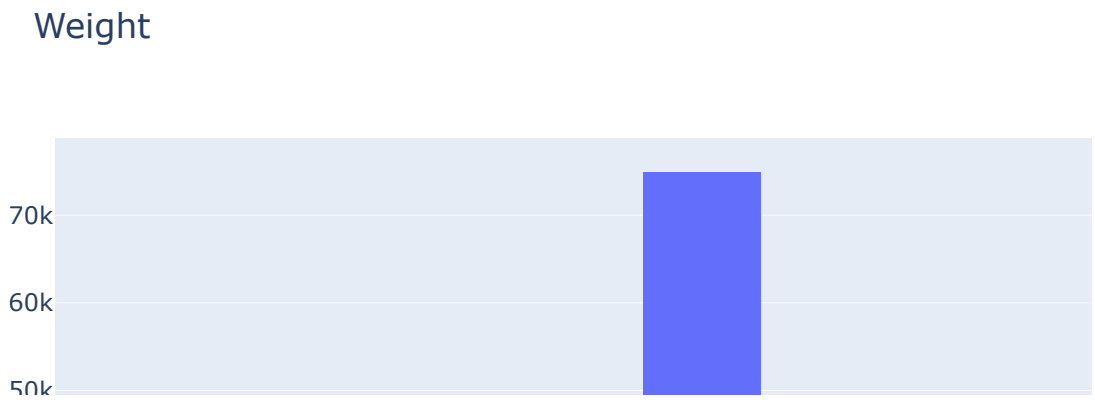


- The average Height of players is form 175 to 180 cms.

In [17]:
```
px.line(df.Weight).update_layout(title="Weight of Players",
    xaxis_title="Number of players",
    yaxis_title="Weight(kg)")
```

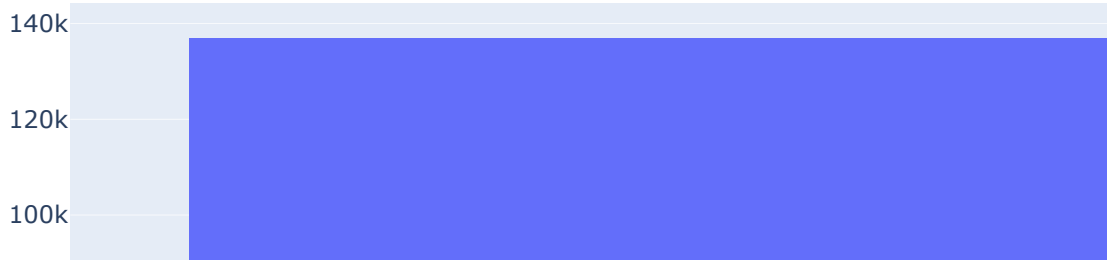


In [18]:
```
px.histogram(df.Weight,nbins=30,title="Weight").update_layout(
    xaxis_title="Weight(kg)",
    yaxis_title="Count")
```



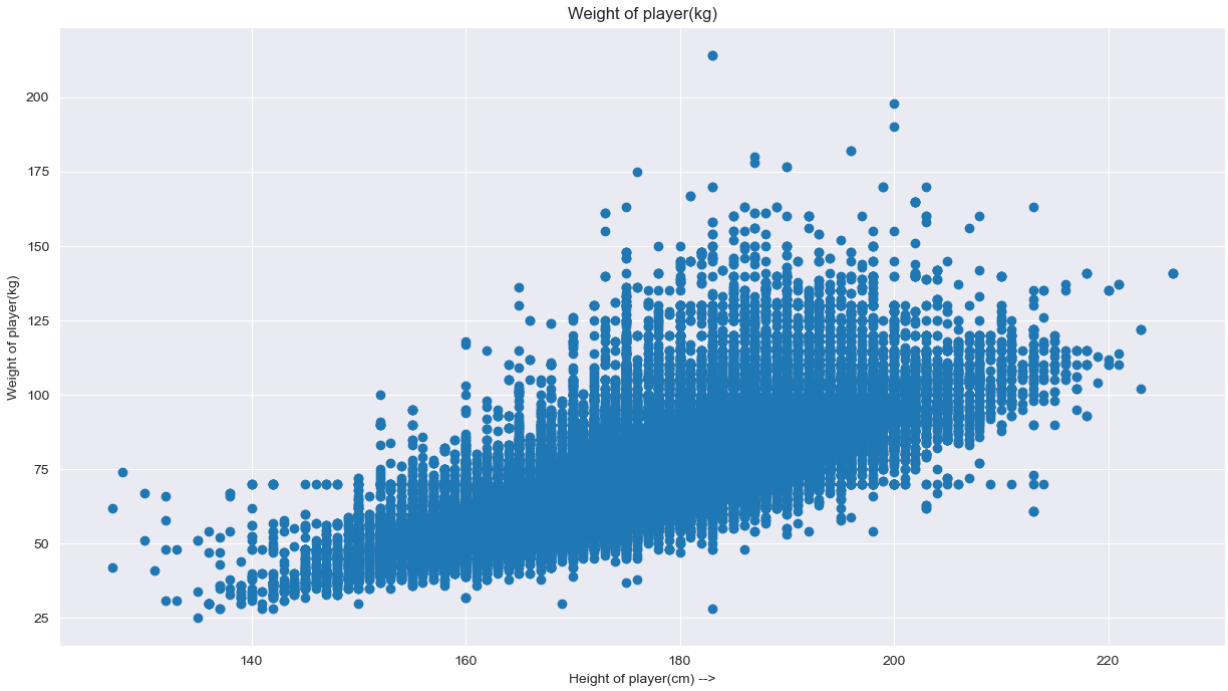
- The average Weight of players is form 70 to 80 kgs.

In [19]:
```
px.histogram(df.Sex,title="Gender (Male X Female)").update_layout(
    xaxis_title="Sex",
    yaxis_title="Count")
```

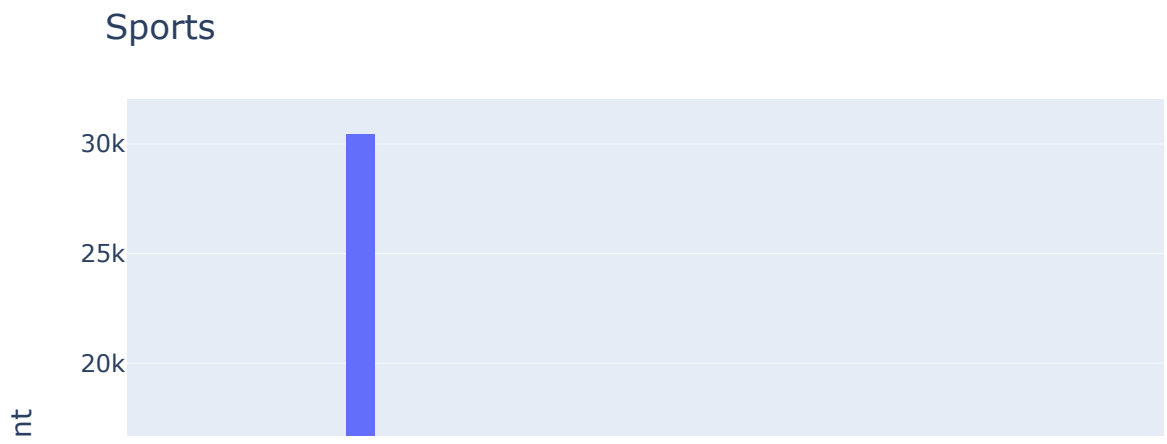


In [20]:
```
px.histogram(df.Medal,title="Medals").update_layout(
    xaxis_title="Medals",
    yaxis_title="Count")
```

In [21]:
```
plt.figure(figsize=(15,8))
plt.title("Weight of player(kg)")
plt.xlabel("Height of player(cm) -->")
plt.ylabel("Weight of player(kg)")
plt.scatter(df.Height,df.Weight);
```



In [22]:
```
px.histogram(df.Sport).update_layout(xaxis=dict(dtick=5)).update_layout(title='Sports'
    xaxis_title="Total sports in Olympics",
    yaxis_title="Count")
```



- Most participation is for Athetics, Swimming, Rowing, Cycling, Shooting...

- Now, by using this shorted data I will make a dashboard using PowerBI tool.

THANK YOU