CRICKET PLAYER PERFORMANCE

**import** numpy **as** np

**import** pandas **as** pd

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**import** warnings

warnings**.**filterwarnings('ignore')

**from** sklearn.linear\_model **import** LogisticRegression

**from** sklearn.naive\_bayes **import** GaussianNB

**from** sklearn.ensemble **import** RandomForestClassifier

**from** sklearn.model\_selection **import** train\_test\_split

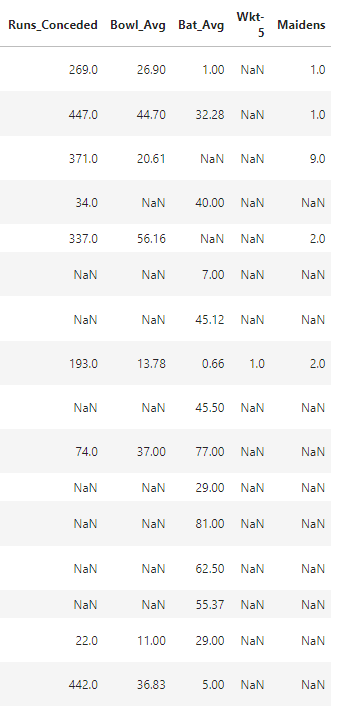
In [2]:

dataset**=**pd**.**read\_csv("C:\mahua\Projects\Internsavy\World\_Cup\_2019\_Player\_Stats.csv")

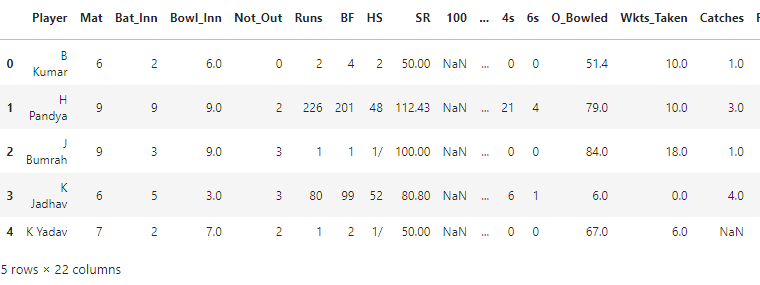
In [3]:

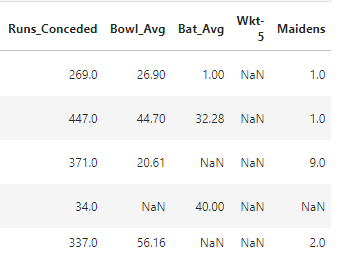
Dataset



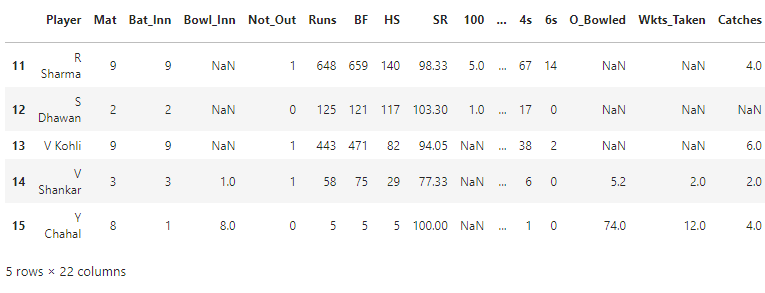


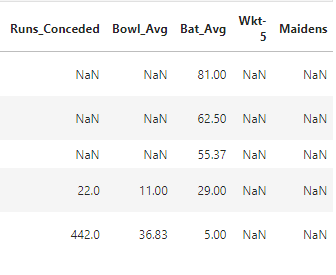
dataset**.**head()



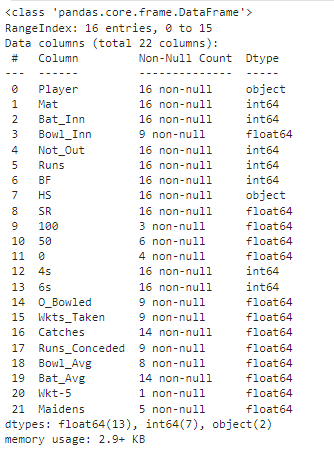


dataset**.**tail()





dataset**.**info()



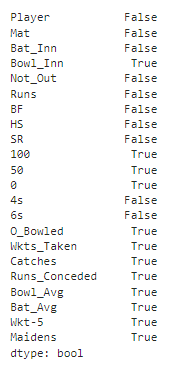
dataset**.**describe()



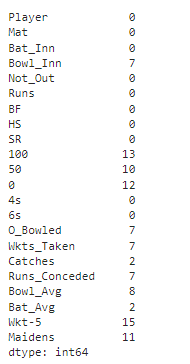
**Data Pre-processing**

In [8]:

dataset**.**isnull()**.**any()



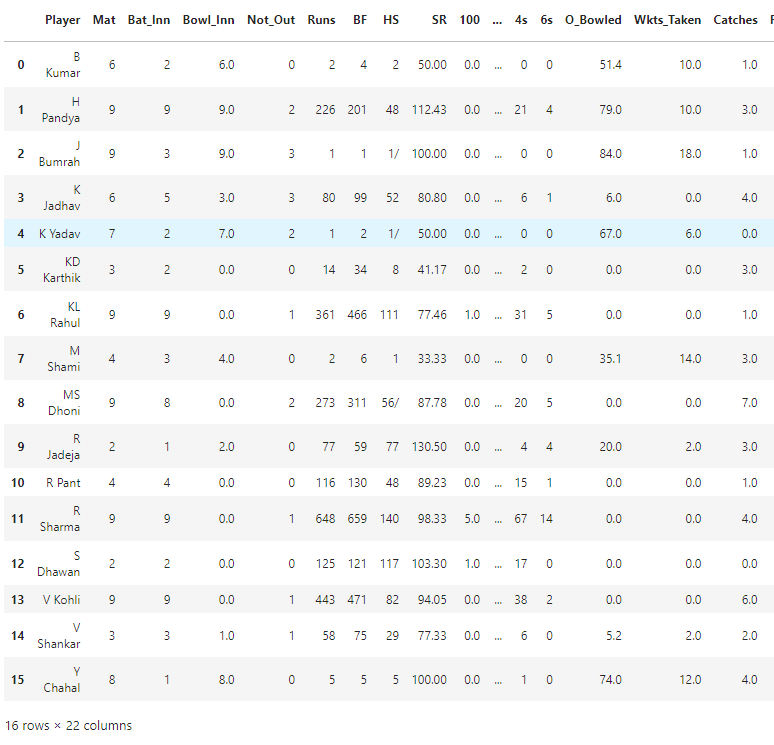
dataset**.**isnull()**.**sum()

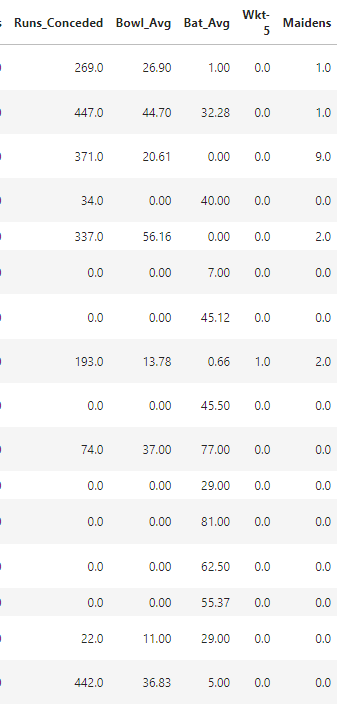


dataset**=**dataset**.**replace(np**.**nan,0)

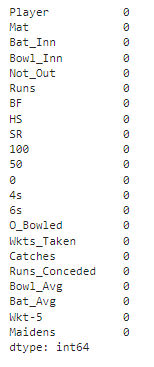
In [11]:

dataset





dataset**.**isnull()**.**sum()



**Data Visualisation**

***From the Batting perspective***

**Players with most NOT OUT's**

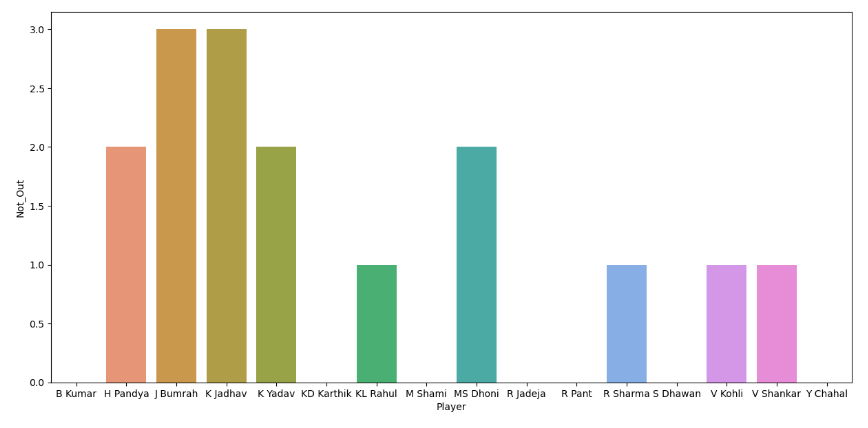
In [13]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(data**=**dataset,x**=**'Player',y**=**'Not\_Out')

Out[13]:

<Axes: xlabel='Player', ylabel='Not\_Out'>



**Players and Runs scored by them**

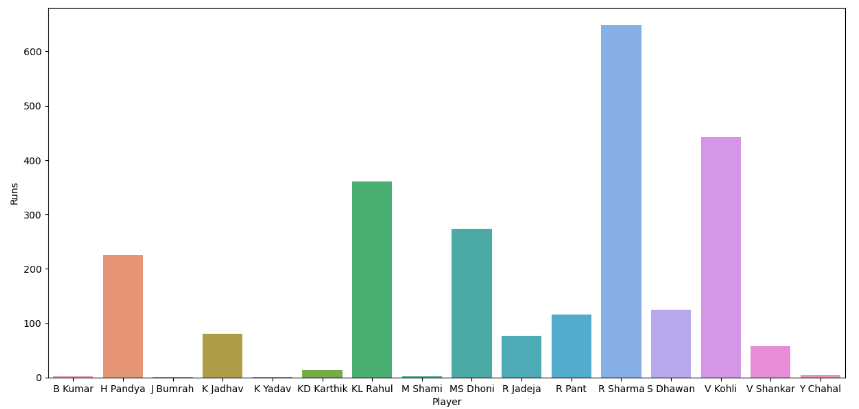
In [14]:

plt**.**figure(figsize**=**(15,7))

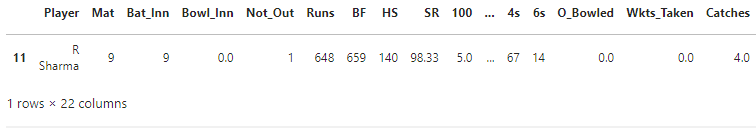
sns**.**barplot(data**=**dataset,x**=**'Player',y**=**'Runs')

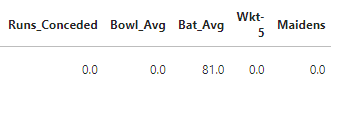
Out[14]:

<Axes: xlabel='Player', ylabel='Runs'>



dataset[dataset**.**Runs **==** dataset**.**Runs**.**max()]





**Players and their Strike Rate**

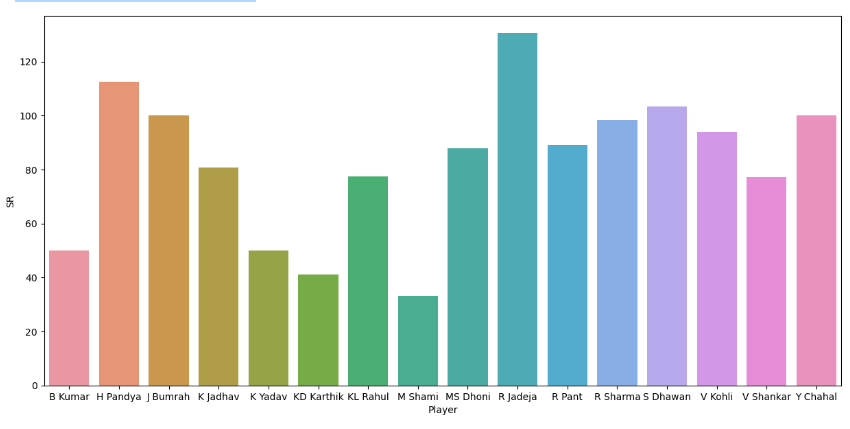
In [16]:

plt**.**figure(figsize**=**(15,7))

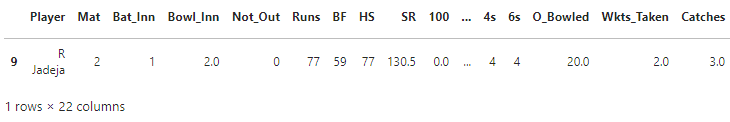
sns**.**barplot(data**=**dataset,x**=**'Player',y**=**'SR')

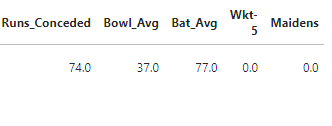
Out[16]:

<Axes: xlabel='Player', ylabel='SR'>

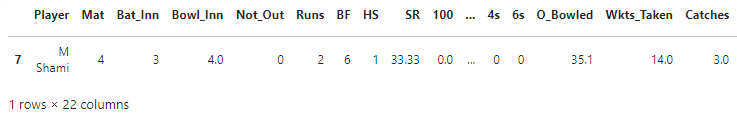


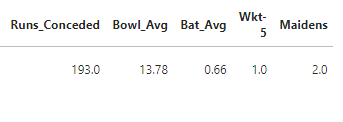
dataset[dataset**.**SR **==** dataset**.**SR**.**max()]





dataset[dataset**.**SR**==**dataset**.**SR**.**min()]





**Centuries scored**

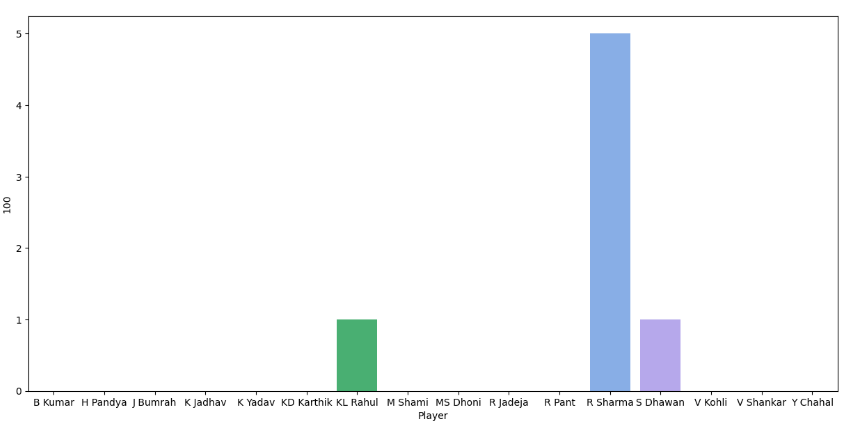
In [19]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'100',data**=**dataset)

Out[19]:

<Axes: xlabel='Player', ylabel='100'>



**Half Centuries scored**

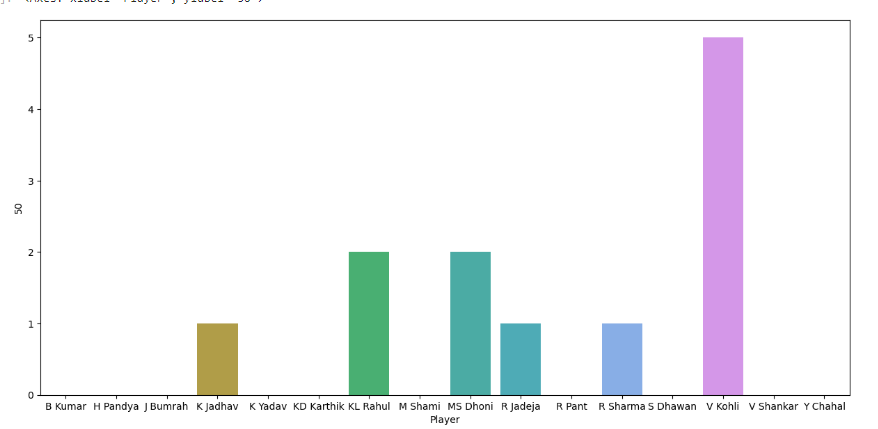
In [20]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'50',data**=**dataset)

Out[20]:

<Axes: xlabel='Player', ylabel='50'>



**Players and Boundaries**

In [21]:

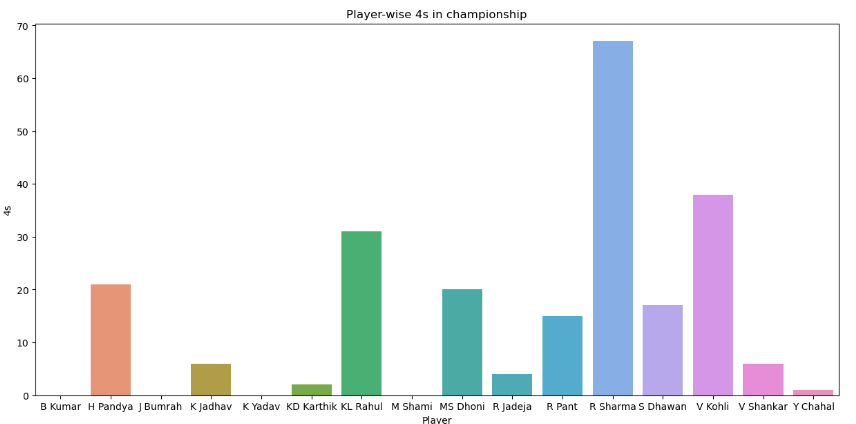
plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'4s',data**=**dataset)

plt**.**title("Player-wise 4s in championship")

Out[21]:

Text(0.5, 1.0, 'Player-wise 4s in championship')



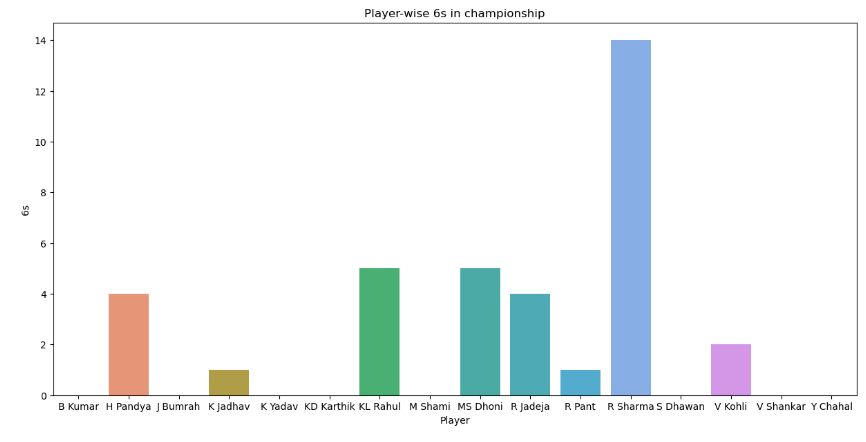
plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'6s',data**=**dataset)

plt**.**title("Player-wise 6s in championship")

Out[22]:

Text(0.5, 1.0, 'Player-wise 6s in championship')



**Players and their Batting Average**

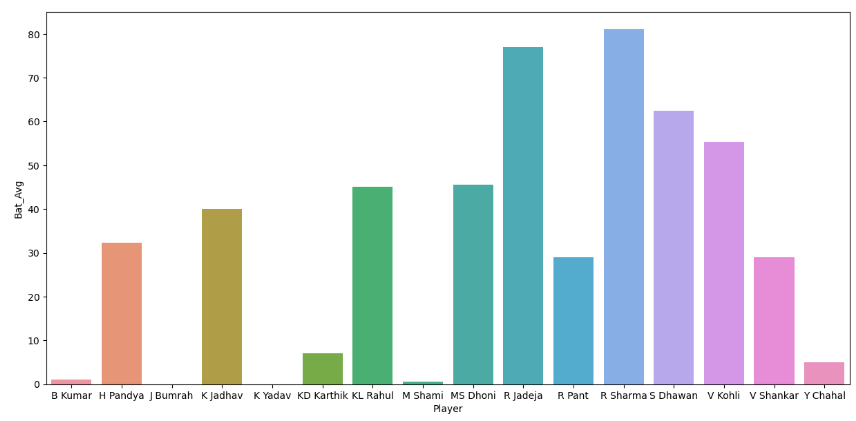
In [23]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'Bat\_Avg',data**=**dataset)

Out[23]:

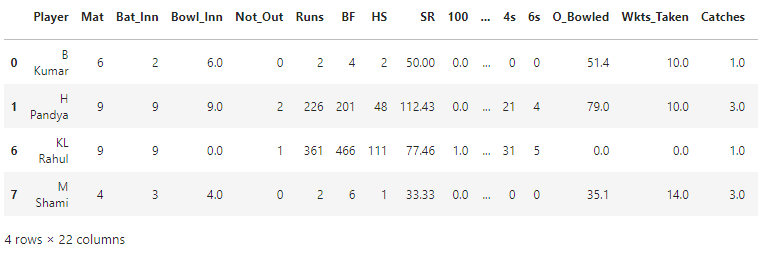
<Axes: xlabel='Player', ylabel='Bat\_Avg'>

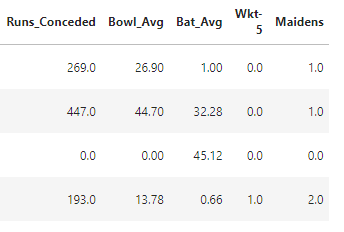


**Players with Zeros in World Cup 2019**

In [24]:

dataset[dataset['0']**==**dataset['0']**.**max()]





***From Bowling perspective***

**Wickets Taken by Players**

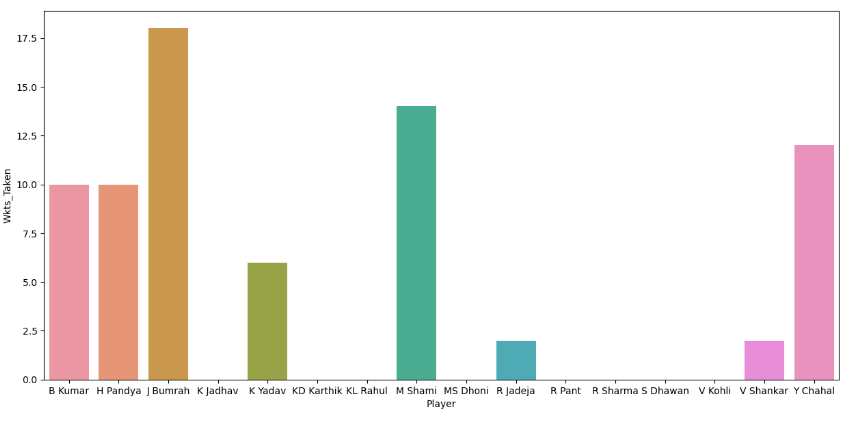
In [25]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'Wkts\_Taken',data**=**dataset)

Out[25]:

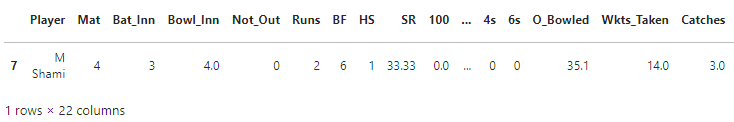
<Axes: xlabel='Player', ylabel='Wkts\_Taken'>

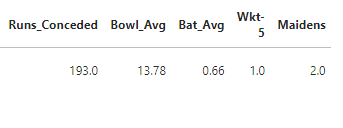


**Players with 5 wicket haul**

In [26]:

dataset[dataset['Wkt-5'] **==** dataset['Wkt-5']**.**max()]





**Player-wise Bowling Average**

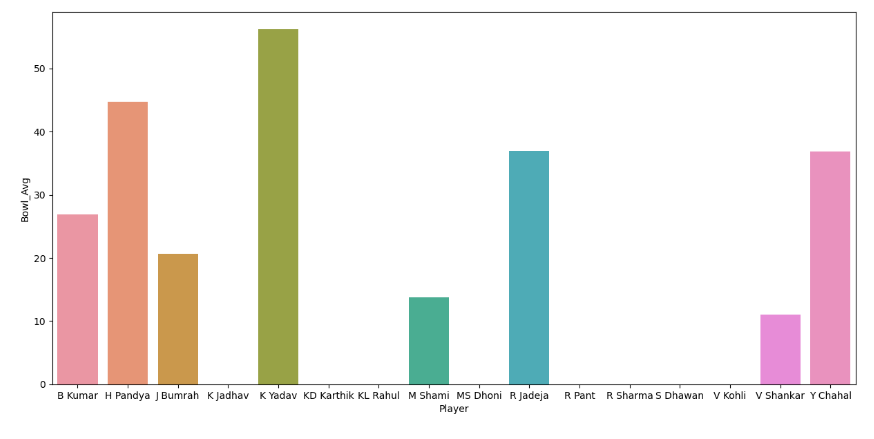
In [27]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'Bowl\_Avg',data**=**dataset)

Out[27]:

<Axes: xlabel='Player', ylabel='Bowl\_Avg'>



**Maidens by a bowler**

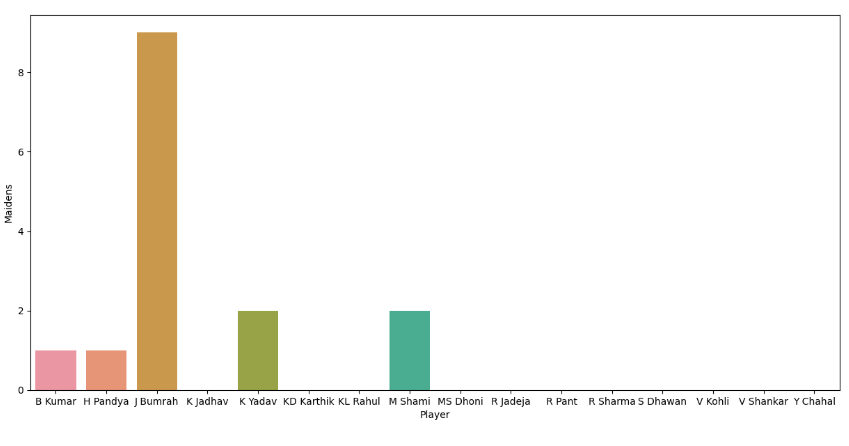
In [28]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'Maidens',data**=**dataset)

Out[28]:

<Axes: xlabel='Player', ylabel='Maidens'>



**Runs Conceded by a bowler**

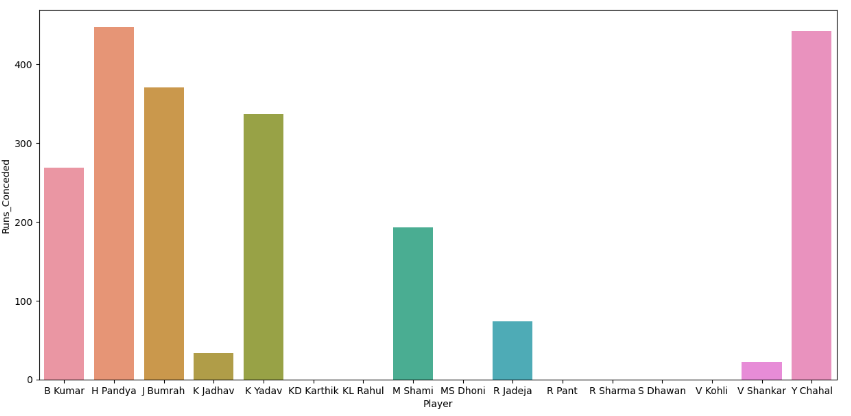
In [29]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'Runs\_Conceded',data**=**dataset)

Out[29]:

<Axes: xlabel='Player', ylabel='Runs\_Conceded'>



**Bowling Overs**

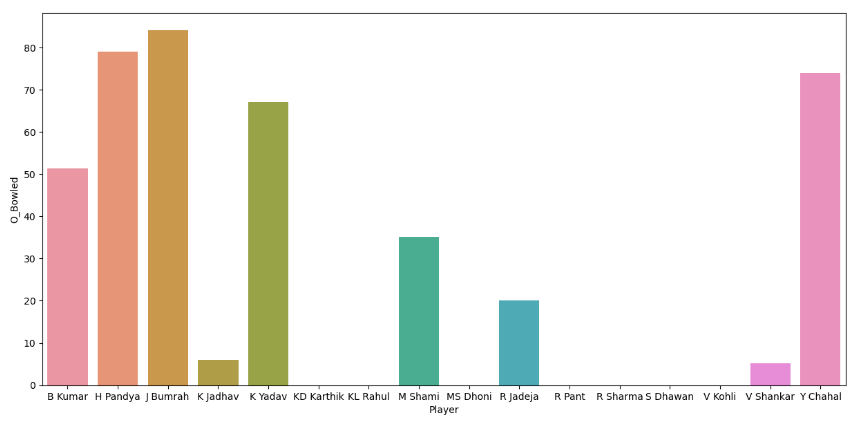
In [30]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'O\_Bowled',data**=**dataset)

Out[30]:

<Axes: xlabel='Player', ylabel='O\_Bowled'>



**Catches by Players**

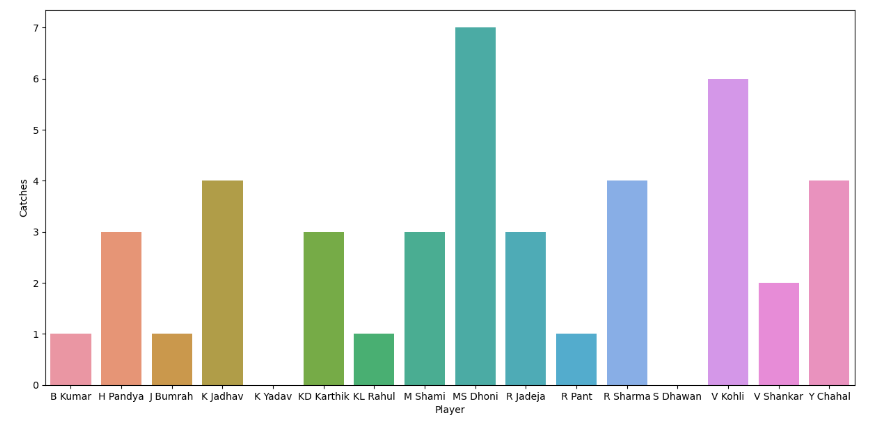
In [31]:

plt**.**figure(figsize**=**(15,7))

sns**.**barplot(x**=**'Player',y**=**'Catches',data**=**dataset)

Out[31]:

<Axes: xlabel='Player', ylabel='Catches'>



**Data Preparation for Model Building**

**Calculate Bowling Economy\_Rate for each player**

In [32]:

dataset['Economy\_Rate'] **=**round((dataset['Runs\_Conceded'] **/**dataset['O\_Bowled']),2)

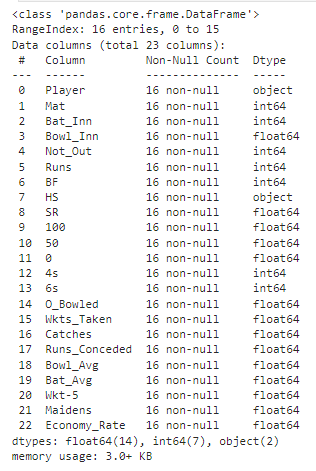
dataset**=**dataset**.**replace(np**.**nan,0)

In [33]:

dataset[['Player', 'Economy\_Rate']]



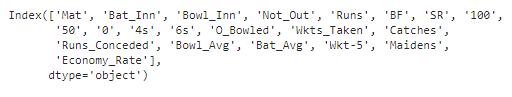
dataset**.**info()



dataset**.**drop(['Player','HS'],axis**=**1,inplace**=True**)

In [36]:

dataset**.**columns



**Model Building**

1. Logistic Regression Model

2. Random Forest Classifier

3. Gaussian Naive Bayes

*From Bowling Perspective*

Wickets taken by a player is an important criterion from the bowling perspective

In [37]:

x\_bowl**=**dataset**.**drop("Wkts\_Taken",axis**=**1)

y\_bowl**=**dataset['Wkts\_Taken']

In [38]:

x\_bowl\_train,x\_bowl\_test,y\_bowl\_train,y\_bowl\_test**=**train\_test\_split(x\_bowl,y\_bowl,test\_size**=**0.3,random\_state**=**45) *#random state=42*

**Logistic Regression Model**

In [39]:

lgm**=**LogisticRegression()

In [40]:

lgm**.**fit(x\_bowl\_train,y\_bowl\_train)

Out[40]:

LogisticRegression()

In [41]:

lgm**.**predict(x\_bowl\_test)

Out[41]:

array([ 2., 0., 0., 12., 0.])

In [42]:

score**=**round(lgm**.**score(x\_bowl\_train,y\_bowl\_train)**\***100,2)

print("Accuracy Score:",round(score,2,),"%")

Accuracy Score: 100.0 %

**Random Forest Classifier**

In [43]:

rfc**=**RandomForestClassifier()

In [44]:

rfc**.**fit(x\_bowl\_train,y\_bowl\_train)

Out[44]:

RandomForestClassifier()

In [45]:

rfc**.**predict(x\_bowl\_test)

Out[45]:

array([0., 0., 0., 6., 0.])

In [46]:

score**=**round(rfc**.**score(x\_bowl\_train,y\_bowl\_train)**\***100,2)

print("Accuracy Score:",round(score,2,),"%")

Accuracy Score: 100.0 %

**Gaussian Naive Bayes**

In [47]:

gnb**=**GaussianNB()

In [48]:

gnb**.**fit(x\_bowl\_train,y\_bowl\_train)

Out[48]:

GaussianNB()

In [49]:

gnb**.**predict(x\_bowl\_test)

Out[49]:

array([10., 0., 0., 10., 0.])

In [50]:

score**=**round(gnb**.**score(x\_bowl\_train,y\_bowl\_train)**\***100,2)

print("Accuracy Score:",round(score,2,),"%")

Accuracy Score: 100.0 %

***From Batting Perspective***

**Runs scored by player is an important criterion from the batting perspective**

In [51]:

x\_bat**=**dataset**.**drop("Runs",axis**=**1)

y\_bat**=**dataset['Runs']

In [52]:

x\_bat\_train,x\_bat\_test,y\_bat\_train,y\_bat\_test**=**train\_test\_split(x\_bat,y\_bat,test\_size**=**0.3,random\_state**=**45)

**Logistic Regression Model**

In [53]:

lgm**=**LogisticRegression()

In [54]:

lgm**.**fit(x\_bat\_train,y\_bat\_train)

Out[54]:

LogisticRegression()

In [55]:

lgm**.**predict(x\_bat\_test)

Out[55]:

array([ 58, 443, 443, 5, 14], dtype=int64)

In [56]:

score**=**round(lgm**.**score(x\_bat\_train,y\_bat\_train)**\***100,2)

print("Accuracy Score:",round(score,2,),"%")

Accuracy Score: 100.0 %

**Random Forest Classifier**

In [57]:

rfc**=**RandomForestClassifier()

In [58]:

rfc**.**fit(x\_bat\_train,y\_bat\_train)

Out[58]:

RandomForestClassifier()

In [59]:

rfc**.**predict(x\_bat\_test)

Out[59]:

array([ 58, 443, 443, 1, 125], dtype=int64)

In [60]:

score**=**round(rfc**.**score(x\_bat\_train,y\_bat\_train)**\***100,2)

print("Accuracy Score:",round(score,2,),"%")

Accuracy Score: 100.0 %

**Gaussian Naive Bayes**

In [61]:

gnb**=**GaussianNB()

In [62]:

gnb**.**fit(x\_bat\_train,y\_bat\_train)

Out[62]:

GaussianNB()

In [63]:

gnb**.**predict(x\_bat\_test)

Out[63]:

array([ 2, 2, 443, 2, 2], dtype=int64)

In [64]:

score**=**round(gnb**.**score(x\_bat\_train,y\_bat\_train)**\***100,2)

print("Accuracy Score:",round(score,2,),"%")

Accuracy Score: 100.0 %

**Conclusion:**

The Visualisation of dataset gave us clear idea about the player's performance in World Cup 2019.

Also, the above models gave highest accuracy, for both bowling as well as batting perspectives.