In [1]:
```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import linear_model
import matplotlib.pyplot as plt
```

In [2]:
```python
import seaborn as sns
```

In [3]:
```python
df = pd.read_csv('Virat.csv', encoding='latin-1')
df.head()
```

Out[3]:

	Number	Format	Inning	Position	Score	Balls	Strike Rate	Against	Venue	Host Nation	Series	Year	Team Total	Wickets lost	Not Out	MOTM	Win	Captain
0	1	ODI	2	4	107	114	93.859649	Sri Lanka	Kolkata	India	Bilateral	2009	316	3	No	No	Yes	No
1	2	ODI	2	3	102	95	107.368421	Bangladesh	Mirpur	Bangladesh	Tri-Series	2010	249	4	Yes	Yes	Yes	No
2	3	ODI	2	3	118	121	97.520661	Australia	Vizag	India	Bilateral	2010	292	5	No	Yes	Yes	No
3	4	ODI	1	3	105	104	100.961539	New Zealand	Guwahati	India	Bilateral	2010	276	10	No	Yes	Yes	No
4	5	ODI	1	4	100	83	120.481928	Bangladesh	Mirpur	Bangladesh	World Cup	2011	370	4	Yes	No	Yes	No

In [4]:
```python
df.columns
```

Out[4]:
```
Index(['Number', 'Format', 'Inning', 'Position', 'Score', 'Balls',
       'Strike Rate', 'Against', 'Venue', 'Host Nation', 'Series', 'Year',
       'Team Total', 'Wickets lost', 'Not Out', 'MOTM', 'Win', 'Captain'],
      dtype='object')
```

In [5]:
```python
df.describe()
```

Out[5]:

	Number	Inning	Score	Balls	Strike Rate	Year	Team Total	Wickets lost
count	82.000000	82.000000	82.000000	82.000000	82.000000	82.000000	82.000000	82.000000
mean	41.500000	1.670732	130.378049	147.731707	103.825286	2015.865854	335.317073	6.134146
std	23.815261	0.667682	35.198078	77.370210	39.533027	3.365629	117.733102	2.818643
min	1.000000	1.000000	100.000000	50.000000	34.915254	2009.000000	166.000000	1.000000
25%	21.250000	1.000000	107.000000	96.500000	75.630252	2013.000000	268.500000	4.000000
50%	41.500000	2.000000	116.500000	119.500000	97.160331	2016.000000	305.500000	6.000000
75%	61.750000	2.000000	139.000000	192.500000	121.399552	2018.000000	360.500000	9.000000
max	82.000000	4.000000	254.000000	366.000000	226.000000	2023.000000	687.000000	10.000000

In [6]:
```python
df.dtypes
```

Out[6]:
```
Number            int64
Format           object
Inning            int64
Position         object
Score             int64
Balls             int64
Strike Rate     float64
Against          object
Venue            object
Host Nation      object
Series           object
Year              int64
Team Total        int64
Wickets lost      int64
Not Out          object
MOTM             object
Win              object
Captain          object
dtype: object
```

In [7]:
```python
df.rename(columns = {'Strike Rate' : 'stk', 'Position' :'pos'}, inplace = True)
```

In [8]:
```python
df.dtypes
```

Out[8]:
```
Number            int64
Format           object
Inning            int64
pos              object
Score             int64
Balls             int64
stk             float64
Against          object
Venue            object
Host Nation      object
Series           object
Year              int64
Team Total        int64
Wickets lost      int64
Not Out          object
MOTM             object
Win              object
Captain          object
dtype: object
```

In [9]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 82 entries, 0 to 81
Data columns (total 18 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Number        82 non-null     int64  
 1   Format        82 non-null     object 
 2   Inning        82 non-null     int64  
 3   pos           82 non-null     object 
 4   Score         82 non-null     int64  
 5   Balls         82 non-null     int64  
 6   stk           82 non-null     float64
 7   Against       82 non-null     object 
 8   Venue         82 non-null     object 
 9   Host Nation   82 non-null     object 
 10  Series        82 non-null     object 
 11  Year          82 non-null     int64  
 12  Team Total    82 non-null     int64  
 13  Wickets lost  82 non-null     int64  
 14  Not Out       82 non-null     object 
 15  MOTM          82 non-null     object 
 16  Win           82 non-null     object 
 17  Captain       82 non-null     object 
dtypes: float64(1), int64(7), object(10)
memory usage: 11.7+ KB
```

In [10]:
```python
df.isnull().sum()
```

Out[10]:
```
Number          0
Format          0
Inning          0
pos             0
Score           0
Balls           0
stk             0
Against         0
Venue           0
Host Nation     0
Series          0
Year            0
Team Total      0
Wickets lost    0
Not Out         0
MOTM            0
Win             0
Captain         0
dtype: int64
```

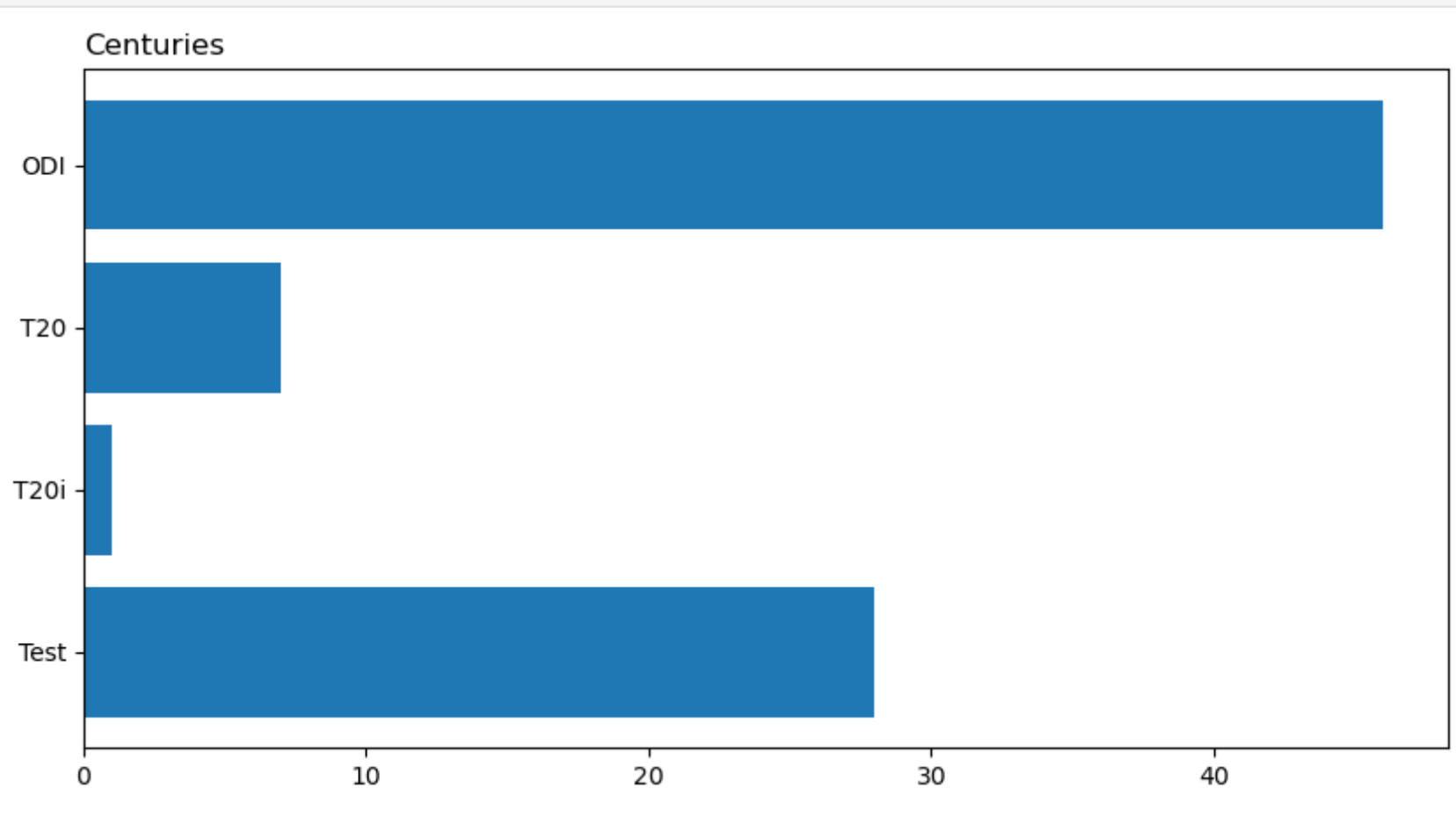
In [11]:
```python
df1 = pd.get_dummies(df,columns = ['Format', 'Against','Venue','Captain', 'MOTM','Win'])
df.drop(['Host Nation', 'Series'], axis = 1)
```

Out[11]:

	Number	Format	Inning	pos	Score	Balls	stk	Against	Venue	Year	Team Total	Wickets lost	Not Out	MOTM	Win	Captain
0	1	ODI	2	4	107	114	93.859649	Sri Lanka	Kolkata	2009	316	3	No	No	Yes	No
1	2	ODI	2	3	102	95	107.368421	Bangladesh	Mirpur	2010	249	4	Yes	Yes	Yes	No
2	3	ODI	2	3	118	121	97.520661	Australia	Vizag	2010	292	5	No	Yes	Yes	No
3	4	ODI	1	3	105	104	100.961539	New Zealand	Guwahati	2010	276	10	No	Yes	Yes	No
4	5	ODI	1	4	100	83	120.481928	Bangladesh	Mirpur	2011	370	4	Yes	No	Yes	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
77	78	ODI	1	3	113	87	129.885057	Sri Lanka	Guwahati	2023	373	7	No	Yes	Yes	No
78	79	ODI	1	3	166	110	150.909091	Sri Lanka	Thiruvananthapuram	2023	390	5	Yes	Yes	Yes	No
79	80	Test	2	4	186	364	51.098901	Australia	Ahmedabad	2023	571	9	No	Yes	Drawn	No
80	81	T20	2	Open	100	63	158.730159	Sunrisers Hyderabad	Hyderabad	2023	187	2	No	Yes	Yes	No
81	82	T20	1	Open	101	61	165.573770	Gujarat Titans	Bengaluru	2023	197	5	Yes	No	No	No

82 rows × 16 columns

In [12]:
```python
centuries = df.groupby('Format').size()
fig, ax = plt.subplots(figsize = (10,5))
ax.barh(centuries.index, centuries.values)
ax.invert_yaxis()
ax.set_title('Centuries', loc ='left')
plt.show()
```



In [13]:
```python
df.groupby(['Format', 'Inning']).count()[['Number']]
```

Out[13]:

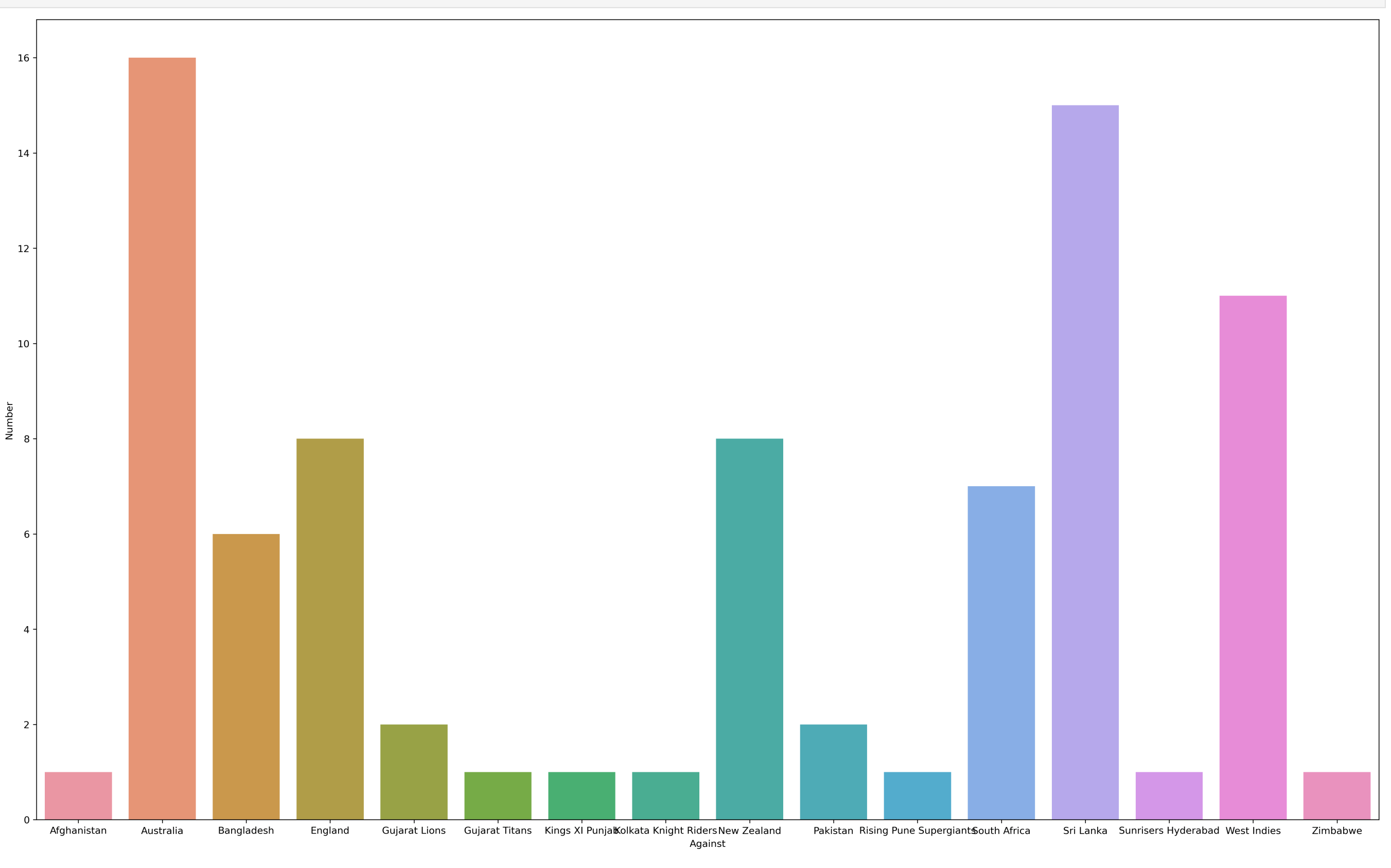
Format	Inning	Number
ODI	**1**	20
	2	26
T20	**1**	5
	2	2
T20i	**1**	1
Test	**1**	8
	2	15
	3	3
	4	2

In [14]:
```python
df.groupby(['Against']).count()[['Number']]
```

Out[14]:

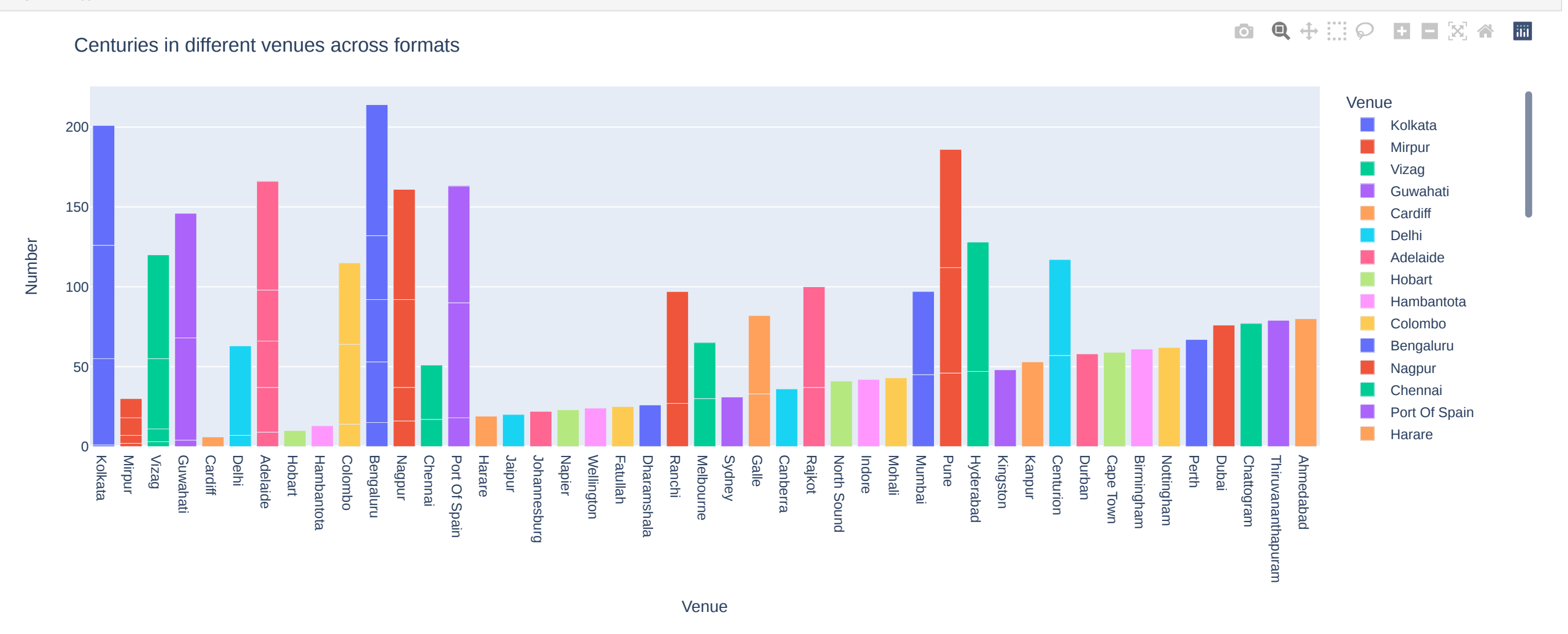
Against	Number
Afghanistan	1
Australia	16
Bangladesh	6
England	8
Gujarat Lions	2
Gujarat Titans	1
Kings XI Punjab	1
Kolkata Knight Riders	1
New Zealand	8
Pakistan	2
Rising Pune Supergiants	1
South Africa	7
Sri Lanka	15
Sunrisers Hyderabad	1
West Indies	11
Zimbabwe	1

In [15]:
```python
centuries=df.groupby(['Against']).count()[['Number']].reset_index()
plt.figure(figsize=(25,15),dpi=300)
sns.barplot(x="Against",y="Number",data=centuries)
plt.show()
```



In [16]:
```python
import plotly.express as px
fig = px.bar(df, x='Format', y='stk', color='Against')
fig.update_layout(height=700, width=700, title_text='Comparison of strike rate in different formats against nations')
fig.show()
```

In [17]:
```python
ven_cent = df.groupby(['Format', 'Venue','Against']).count()[['Number']].sort_values('Venue').reset_index()
fig = px.bar(df, x = 'Venue', y = 'Number', color = 'Venue')
fig.update_layout(title_text='Centuries in different venues across formats')
fig.show()
```



In [18]:
```python
print('Average team score in test when Virat scores a double century',df.query('Score>200')['Team Total'].mean())
print('Average team total when Virat scores a century in ODI',df[df['Format']=='ODI']['Team Total'].mean())
print('Average team total when Virat scores a century in T20',df[df['Format']=='T20']['Team Total'].mean())
```

```
Average team score in test when Virat scores a double century 603.6666666666666
Average team total when Virat scores a century in ODI 299.82608695652175
Average team total when Virat scores a century in T20 204.42857142857142
```

In [19]:
```python
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
for column in ('Format', 'Against', 'Venue', 'Captain'):
    label_encoder.fit(df[column])
    df[column + '_encoded'] = label_encoder.transform(df[column])
```

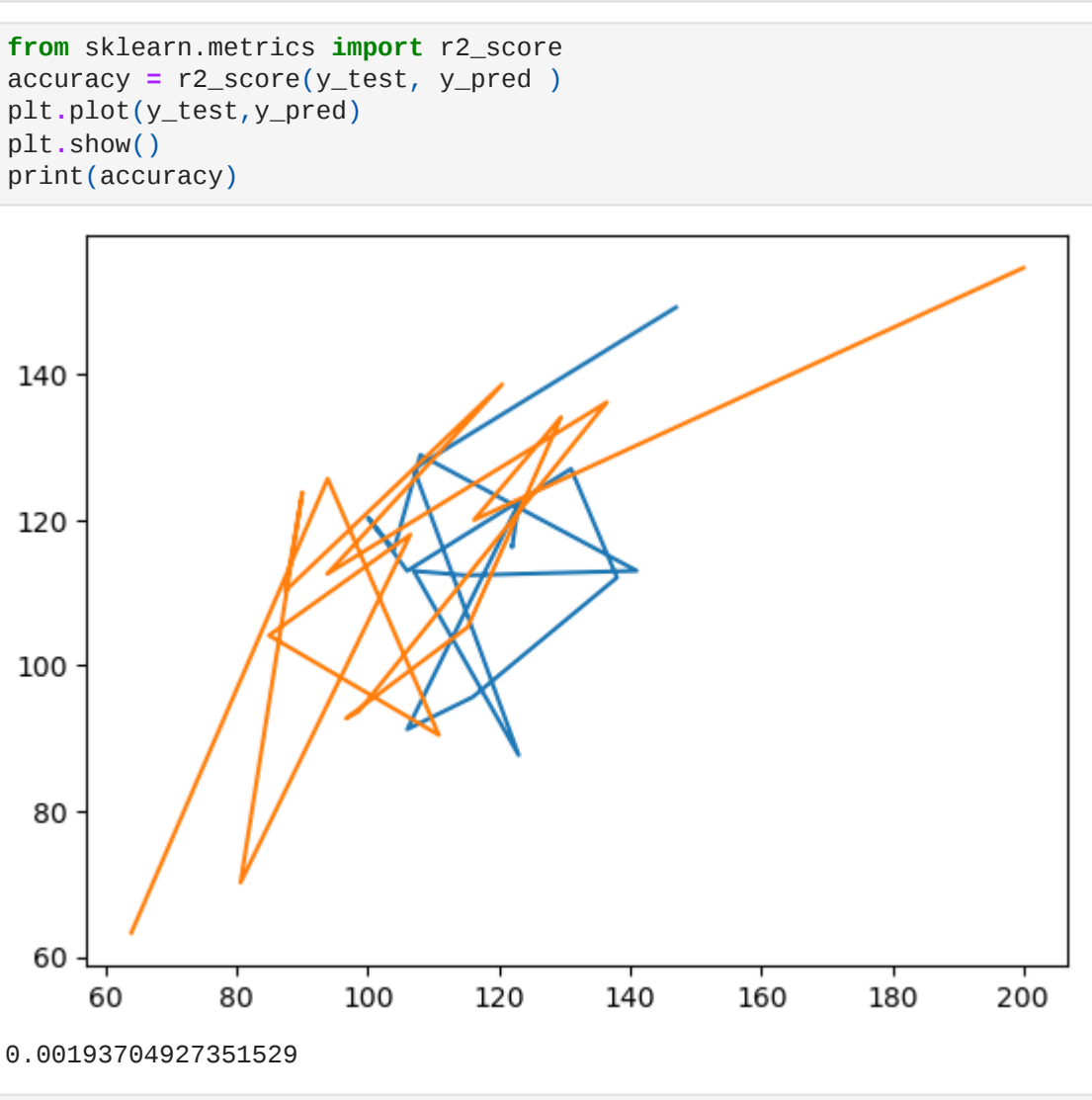
In [20]:
```python
X = df[['Number', 'Balls', 'Team Total', 'Wickets lost']]
Y = df[['Score','stk']]
```

In [21]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state = 42)
```

In [22]:
```python
regression = linear_model.LinearRegression()
regression.fit(X_train, y_train)

y_pred = regression.predict(X_test)
y_test = np.array(y_test)
```

In [23]:
```python
from sklearn.metrics import r2_score
accuracy = r2_score(y_test, y_pred )
plt.plot(y_test,y_pred)
plt.show()
print(accuracy)
```



```
0.00193704927351529
```

In [24]:
```python
print(X_test)
```

```
    Number  Balls  Team Total  Wickets lost
30      31    230         475            10
0        1    114         316             3
22      23    111         268            10
31      32    126         300             7
18      19    108         230             4
28      29    175         315            10
10      11    120         304             3
53      54    119         352             8
4        5     83         370             4
12      13    113         314             6
49      50     96         375             5
33      34    140         299             8
68      69    120         250            10
35      36     92         323            10
69      70     95         313             5
45      46    105         356             7
75      76     61         212             2
```

In [25]:
```python
y_pred_ = regression.predict([[1, 114, 316, 3]])
```

```
/Users/shrinivass/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarning:

X does not have valid feature names, but LinearRegression was fitted with feature names
```

In [26]:
```python
y_pred_
```

Out[26]:
```
array([[127.03145898, 125.6570205 ]])
```

SVM

In [27]:
```python
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler
X = df[['Number', 'Balls', 'Team Total', 'Wickets lost']]
Y = df[['Score']]
```

In [28]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 43)
```

In [29]:
```python
svm_classifier = SVC(kernel = 'linear')
svm_classifier.fit(X_train, y_train)
y_pred = svm_classifier.predict(X_test)
accuracy = accuracy_score(y_pred, y_test)
matrix = confusion_matrix(y_pred, y_test)
matrix
accuracy
```

```
/Users/shrinivass/opt/anaconda3/lib/python3.9/site-packages/sklearn/utils/validation.py:993: DataConversionWarning:

A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
```

Out[29]:
```
0.0
```

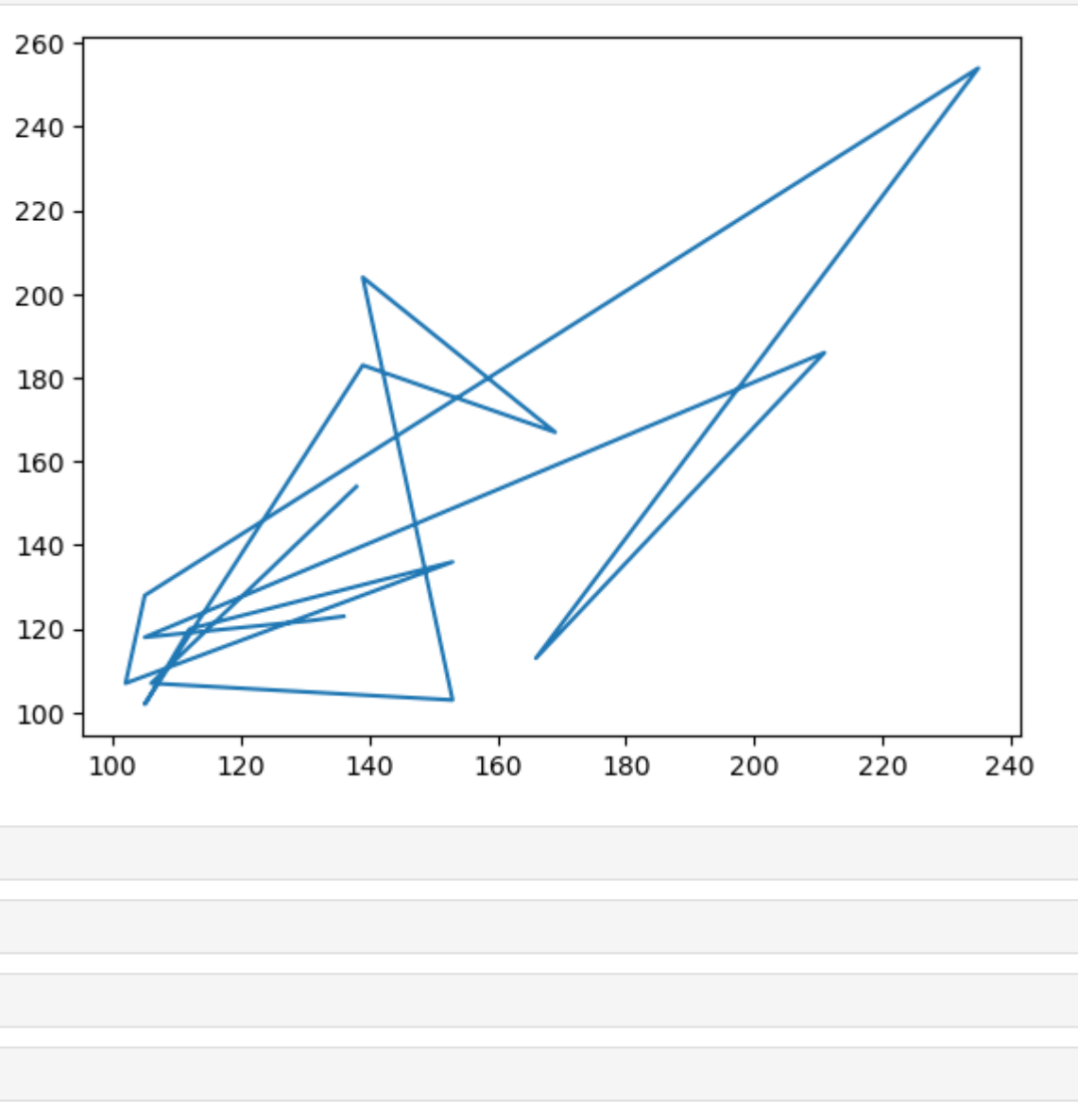
In [30]:
```python
y_pred_ = svm_classifier.predict([[1, 114, 316, 3]])
y_pred_
```

```
/Users/shrinivass/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarning:

X does not have valid feature names, but SVC was fitted with feature names
```

Out[30]:
```
array([106])
```

In [31]:
```python
plt.plot(y_pred, y_test)
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: