In [1]:
```python
#importing necessary libraries
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import plotly.express as px
import math
```

In [2]:
```python
pd.set_option('display.max_columns', None)
```

In [3]:
```python
rgs=pd.read_csv("RGS/RGS_odi.csv")
rgs
```

Out[3]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary	Innings
0		NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.0
1	8	9.0	0.0	0.0	88.88	7.0	bowled	NaN	JH Kallis	Pacer	NaN	NaN	NaN	1.0
2	1	4.0	0.0	0.0	25.00	7.0	caught	B Lee	JR Hopes	NaN	NaN	NaN	NaN	2.0
3	52	61.0	3.0	1.0	85.24	5.0	caught	Yasir Hameed	Shoaib Malik	Spin	NaN	NaN	NaN	2.0
4	29	43.0	5.0	0.0	67.44	4.0	caught	AC Gilchrist	B Lee	Pacer	NaN	NaN	NaN	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
219	159	138.0	17.0	5.0	115.21	1.0	caught	SD Hope	SS Cottrell	Pacer	NaN	NaN	NaN	1.0
220	63	63.0	8.0	1.0	100.00	1.0	caught	SD Hope	JO Holder	Pacer	NaN	NaN	NaN	2.0
221	10	15.0	2.0	0.0	66.66	1.0	caught	DA Warner	MA Starc	Pacer	NaN	NaN	NaN	1.0
222	42	44.0	6.0	0.0	95.45	1.0	lbw	NaN	A Zampa	Spin	NaN	NaN	NaN	1.0
223	119	128.0	8.0	6.0	92.96	1.0	caught	MA Starc	A Zampa	Spin	NaN	NaN	NaN	2.0

224 rows × 22 columns

In [4]:
```python
#reading data
vk=pd.read_csv("VK/VK_odi.csv",encoding='latin1')
vk
```

Out[4]:

	Match No	Runs	Balls Faced	No of Innings	Not outs	No of Times Dissmissed	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary
0	1	12.0	22.0	1	0	1	1.0	0.0	54.54	2.0	lbw	NaN	KMDN Kulasekara	Pacer	NaN
1	2	37.0	67.0	2	0	2	6.0	0.0	55.22	2.0	caught	CK Kapugedera	T Thushara	Pacer	NaN
2	3	25.0	38.0	3	0	3	4.0	0.0	65.78	1.0	run out	NaN	NaN	NaN	NaN
3	4	54.0	66.0	4	0	4	7.0	0.0	81.81	1.0	bowled	NaN	T Thushara	Pacer	NaN
4	5	31.0	46.0	5	0	5	3.0	1.0	67.39	1.0	lbw	NaN	KMDN Kulasekara	Pacer	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
243	244	78.0	76.0	235	39	196	6.0	0.0	102.63	3.0	caught	MA Starc	A Zampa	Spin	Zampa to Kohli OUT ,He's gone Zampa's go Koh...
244	245	89.0	91.0	236	39	197	8.0	0.0	97.80	3.0	bowled	NaN	JR Hazlewood	Pacer	Hazlewood to Kohli OUT ,He's gone through him!...
245	246	51.0	63.0	237	39	198	6.0	0.0	80.95	3.0	bowled	NaN	IS Sodhi	Spin	Sodhi to Kohli OUT,Through the gate! This is a...
246	247	15.0	25.0	238	39	199	1.0	0.0	60.00	3.0	bowled	NaN	TG Southee	Pacer	Southee to Kohli, OUT Now, it's Southee's tur...
247	248	9.0	12.0	239	39	200	0.0	1.0	75.00	3.0	caught	KA Jamieson	HK Bennett	Pacer	Bennett to Kohli OUT Caught at third man Tha...

248 rows × 27 columns

In [5]:
```python
ss=pd.read_csv("SS/SS_odi.csv")
ss
```

Out[5]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary
0								NaN	NaN	NaN	NaN	NaN	NaN
1	41	53	3	0	77.35	7	caught	PD Collingwood	SCJ Broad	Pacer	NaN	NaN	NaN
2	20	21	1	0	95.23	7	lbw	NaN	JM Anderson	Pacer	NaN	NaN	NaN
3	18	8	4	0	225	7	not out	NaN	NaN	NaN	NaN	NaN	NaN
4	15	13	1	0	115.38	7	caught	JM Anderson	SCJ Broad	Pacer	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
120	131	132	14	1	99.24	3	caught	SS Iyer	Mohammed Shami	Pacer	NaN	NaN	NaN
121	76	94	3	0	80.85	3	lbw	NaN	A Nortje	Pacer	NaN	NaN	NaN
122	13	17	1	0	76.47	3	caught	JT Smuts	L Ngidi	Pacer	NaN	NaN	NaN
123	20	26	3	0	76.92	3	lbw	NaN	AL Phehlukwayo	Pacer	NaN	NaN	NaN
124	14	17	0	1	82.35	3	bowled	NaN	MJ Santner	Spin	NaN	NaN	NaN

125 rows × 22 columns

In [6]:
```python
kw=pd.read_csv("KW/kw_odi.csv")
kw
```

Out[6]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary
0	0.0	9.0	0.0	0.0	0.00	4.0	bowled	NaN	P Kumar	Pacer	NaN	NaN	NaN
1	0.0	2.0	0.0	0.0	0.00	4.0	caught	TT Samaraweera	AD Mathews	Pacer	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	13.0	27.0	1.0	0.0	48.14	6.0	bowled	NaN	I Sharma	Pacer	NaN	NaN	NaN
4	108.0	132.0	6.0	2.0	81.81	5.0	caught	Raqibul Hasan	Shafiul Islam	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
146	27.0	40.0	3.0	0.0	67.50	3.0	run out	NaN	NaN	NaN	NaN	NaN	NaN
147	67.0	95.0	6.0	0.0	70.52	3.0	caught	RA Jadeja	YS Chahal	Spin	NaN	NaN	NaN
148	30.0	53.0	2.0	0.0	56.60	3.0	caught	JC Buttler	LE Plunkett	Pacer	NaN	NaN	NaN
149	22.0	31.0	2.0	0.0	70.96	3.0	caught	MA Agarwal	YS Chahal	Spin	NaN	NaN	NaN
150	19.0	26.0	2.0	0.0	73.07	3.0	bowled	NaN	A Zampa	Spin	NaN	NaN	NaN

151 rows × 22 columns

In [7]:
```python
jr=pd.read_csv("JR/jr_odi.csv")
jr
```

Out[7]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary	Innings
0								NaN	NaN	NaN	NaN	NaN	NaN	1
1	36	50	3	0	72	4	bowled	NaN	RA Jadeja	NaN	NaN	NaN	NaN	2
2	39	57	4	0	68.42	4	caught	MS Dhoni	I Sharma	NaN	NaN	NaN	NaN	1
3	57	45	6	1	126.66	6	not out	NaN	NaN	NaN	NaN	NaN	NaN	1
4	31	49	4	0	63.26	4	bowled	NaN	RA Jadeja	NaN	NaN	NaN	NaN	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
142	7	30	0	0	23.33	3	caught	TWM Latham	C de Grandhomme	NaN	NaN	NaN	NaN	2
143	17	21	0	0	80.95	3	run out	NaN	NaN	NaN	NaN	NaN	NaN	1
144								NaN	NaN	NaN	NaN	NaN	NaN	
145	49	52	3	0	94.23	3	caught	T Bavuma	T Shamsi	NaN	NaN	NaN	NaN	2
146								NaN	NaN	NaN	NaN	NaN	NaN	

147 rows × 22 columns

In [ ]:

In [8]:
```python
#changing position of columns
#cols = rgs.columns.tolist()
#len(cols)
```

In [9]:
```python
#removing null values
rgs.replace(' ', np.NaN,inplace=True)
rgs
```

Out[9]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary	Innings
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.0
1	8	9.0	0.0	0.0	88.88	7.0	bowled	NaN	JH Kallis	Pacer	NaN	NaN	NaN	1.0
2	1	4.0	0.0	0.0	25.00	7.0	caught	B Lee	JR Hopes	NaN	NaN	NaN	NaN	2.0
3	52	61.0	3.0	1.0	85.24	5.0	caught	Yasir Hameed	Shoaib Malik	Spin	NaN	NaN	NaN	2.0
4	29	43.0	5.0	0.0	67.44	4.0	caught	AC Gilchrist	B Lee	Pacer	NaN	NaN	NaN	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
219	159	138.0	17.0	5.0	115.21	1.0	caught	SD Hope	SS Cottrell	Pacer	NaN	NaN	NaN	1.0
220	63	63.0	8.0	1.0	100.00	1.0	caught	SD Hope	JO Holder	Pacer	NaN	NaN	NaN	2.0
221	10	15.0	2.0	0.0	66.66	1.0	caught	DA Warner	MA Starc	Pacer	NaN	NaN	NaN	1.0
222	42	44.0	6.0	0.0	95.45	1.0	lbw	NaN	A Zampa	Spin	NaN	NaN	NaN	1.0
223	119	128.0	8.0	6.0	92.96	1.0	caught	MA Starc	A Zampa	Spin	NaN	NaN	NaN	2.0

224 rows × 22 columns

In [10]:
```python
vk.replace(' ', np.NaN,inplace=True)
vk
```

Out[10]:

	Match No	Runs	Balls Faced	No of Innings	Not outs	No of Times Dissmissed	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary
0	1	12.0	22.0	1	0	1	1.0	0.0	54.54	2.0	lbw	NaN	KMDN Kulasekara	Pacer	NaN
1	2	37.0	67.0	2	0	2	6.0	0.0	55.22	2.0	caught	CK Kapugedera	T Thushara	Pacer	NaN
2	3	25.0	38.0	3	0	3	4.0	0.0	65.78	1.0	run out	NaN	NaN	NaN	NaN
3	4	54.0	66.0	4	0	4	7.0	0.0	81.81	1.0	bowled	NaN	T Thushara	Pacer	NaN
4	5	31.0	46.0	5	0	5	3.0	1.0	67.39	1.0	lbw	NaN	KMDN Kulasekara	Pacer	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
243	244	78.0	76.0	235	39	196	6.0	0.0	102.63	3.0	caught	MA Starc	A Zampa	Spin	Zampa to Kohli OUT ,He's gone Zampa's go Koh...
244	245	89.0	91.0	236	39	197	8.0	0.0	97.80	3.0	bowled	NaN	JR Hazlewood	Pacer	Hazlewood to Kohli OUT ,He's gone through him!...
245	246	51.0	63.0	237	39	198	6.0	0.0	80.95	3.0	bowled	NaN	IS Sodhi	Spin	Sodhi to Kohli OUT,Through the gate! This is a...
246	247	15.0	25.0	238	39	199	1.0	0.0	60.00	3.0	bowled	NaN	TG Southee	Pacer	Southee to Kohli, OUT Now, it's Southee's tur...
247	248	9.0	12.0	239	39	200	0.0	1.0	75.00	3.0	caught	KA Jamieson	HK Bennett	Pacer	Bennett to Kohli OUT Caught at third man Tha...

248 rows × 27 columns

In [11]:
```python
ss.replace(' ', np.NaN,inplace=True)
ss
```

Out[11]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	41	53	3	0	77.35	7	caught	PD Collingwood	SCJ Broad	Pacer	NaN	NaN	NaN
2	20	21	1	0	95.23	7	lbw	NaN	JM Anderson	Pacer	NaN	NaN	NaN
3	18	8	4	0	225	7	not out	NaN	NaN	NaN	NaN	NaN	NaN
4	15	13	1	0	115.38	7	caught	JM Anderson	SCJ Broad	Pacer	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
120	131	132	14	1	99.24	3	caught	SS Iyer	Mohammed Shami	Pacer	NaN	NaN	NaN
121	76	94	3	0	80.85	3	lbw	NaN	A Nortje	Pacer	NaN	NaN	NaN
122	13	17	1	0	76.47	3	caught	JT Smuts	L Ngidi	Pacer	NaN	NaN	NaN
123	20	26	3	0	76.92	3	lbw	NaN	AL Phehlukwayo	Pacer	NaN	NaN	NaN
124	14	17	0	1	82.35	3	bowled	NaN	MJ Santner	Spin	NaN	NaN	NaN

125 rows × 22 columns

In [12]:
```python
kw.replace(' ', np.NaN,inplace=True)
kw
```

Out[12]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary
0	0.0	9.0	0.0	0.0	0.00	4.0	bowled	NaN	P Kumar	Pacer	NaN	NaN	NaN
1	0.0	2.0	0.0	0.0	0.00	4.0	caught	TT Samaraweera	AD Mathews	Pacer	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	13.0	27.0	1.0	0.0	48.14	6.0	bowled	NaN	I Sharma	Pacer	NaN	NaN	NaN
4	108.0	132.0	6.0	2.0	81.81	5.0	caught	Raqibul Hasan	Shafiul Islam	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
146	27.0	40.0	3.0	0.0	67.50	3.0	run out	NaN	NaN	NaN	NaN	NaN	NaN
147	67.0	95.0	6.0	0.0	70.52	3.0	caught	RA Jadeja	YS Chahal	Spin	NaN	NaN	NaN
148	30.0	53.0	2.0	0.0	56.60	3.0	caught	JC Buttler	LE Plunkett	Pacer	NaN	NaN	NaN
149	22.0	31.0	2.0	0.0	70.96	3.0	caught	MA Agarwal	YS Chahal	Spin	NaN	NaN	NaN
150	19.0	26.0	2.0	0.0	73.07	3.0	bowled	NaN	A Zampa	Spin	NaN	NaN	NaN

151 rows × 22 columns

In [13]:
```python
jr.replace(' ', np.NaN,inplace=True)
jr
```

Out[13]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	36	50	3	0	72	4	bowled	NaN	RA Jadeja	NaN	NaN	NaN	NaN
2	39	57	4	0	68.42	4	caught	MS Dhoni	I Sharma	NaN	NaN	NaN	NaN
3	57	45	6	1	126.66	6	not out	NaN	NaN	NaN	NaN	NaN	NaN
4	31	49	4	0	63.26	4	bowled	NaN	RA Jadeja	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
142	7	30	0	0	23.33	3	caught	TWM Latham	C de Grandhomme	NaN	NaN	NaN	NaN
143	17	21	0	0	80.95	3	run out	NaN	NaN	NaN	NaN	NaN	NaN
144	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145	49	52	3	0	94.23	3	caught	T Bavuma	T Shamsi	NaN	NaN	NaN	NaN
146	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

147 rows × 22 columns

In [ ]:

In [14]:
```python
#conversion of data type to float
rgs['Runs'] = rgs['Runs'].astype(float)
rgs
```

Out[14]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary	Innings
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.0
1	8.0	9.0	0.0	0.0	88.88	7.0	bowled	NaN	JH Kallis	Pacer	NaN	NaN	NaN	1.0
2	1.0	4.0	0.0	0.0	25.00	7.0	caught	B Lee	JR Hopes	NaN	NaN	NaN	NaN	2.0
3	52.0	61.0	3.0	1.0	85.24	5.0	caught	Yasir Hameed	Shoaib Malik	Spin	NaN	NaN	NaN	2.0
4	29.0	43.0	5.0	0.0	67.44	4.0	caught	AC Gilchrist	B Lee	Pacer	NaN	NaN	NaN	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
219	159.0	138.0	17.0	5.0	115.21	1.0	caught	SD Hope	SS Cottrell	Pacer	NaN	NaN	NaN	1.0
220	63.0	63.0	8.0	1.0	100.00	1.0	caught	SD Hope	JO Holder	Pacer	NaN	NaN	NaN	2.0
221	10.0	15.0	2.0	0.0	66.66	1.0	caught	DA Warner	MA Starc	Pacer	NaN	NaN	NaN	1.0
222	42.0	44.0	6.0	0.0	95.45	1.0	lbw	NaN	A Zampa	Spin	NaN	NaN	NaN	1.0
223	119.0	128.0	8.0	6.0	92.96	1.0	caught	MA Starc	A Zampa	Spin	NaN	NaN	NaN	2.0

224 rows × 22 columns

In [15]:
```python
vk['Runs'] = vk['Runs'].astype(float)
vk
```

Out[15]:

	Match No	Runs	Balls Faced	No of Innings	Not outs	No of Times Dissmissed	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary
0	1	12.0	22.0	1	0	1	1.0	0.0	54.54	2.0	lbw	NaN	KMDN Kulasekara	Pacer	NaN
1	2	37.0	67.0	2	0	2	6.0	0.0	55.22	2.0	caught	CK Kapugedera	T Thushara	Pacer	NaN
2	3	25.0	38.0	3	0	3	4.0	0.0	65.78	1.0	run out	NaN	NaN	NaN	NaN
3	4	54.0	66.0	4	0	4	7.0	0.0	81.81	1.0	bowled	NaN	T Thushara	Pacer	NaN
4	5	31.0	46.0	5	0	5	3.0	1.0	67.39	1.0	lbw	NaN	KMDN Kulasekara	Pacer	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
243	244	78.0	76.0	235	39	196	6.0	0.0	102.63	3.0	caught	MA Starc	A Zampa	Spin	Zampa to Kohli OUT ,He's gone Zampa's go Koh...
244	245	89.0	91.0	236	39	197	8.0	0.0	97.80	3.0	bowled	NaN	JR Hazlewood	Pacer	Hazlewood to Kohli OUT ,He's gone through him!...
245	246	51.0	63.0	237	39	198	6.0	0.0	80.95	3.0	bowled	NaN	IS Sodhi	Spin	Sodhi to Kohli OUT,Through the gate! This is a...
246	247	15.0	25.0	238	39	199	1.0	0.0	60.00	3.0	bowled	NaN	TG Southee	Pacer	Southee to Kohli, OUT Now, it's Southee's tur...
247	248	9.0	12.0	239	39	200	0.0	1.0	75.00	3.0	caught	KA Jamieson	HK Bennett	Pacer	Bennett to Kohli OUT Caught at third man Tha...

248 rows × 27 columns

In [16]:
```python
ss['Runs'] = ss['Runs'].astype(float)
ss
```

Out[16]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	41.0	53	3	0	77.35	7	caught	PD Collingwood	SCJ Broad	Pacer	NaN	NaN	NaN
2	20.0	21	1	0	95.23	7	lbw	NaN	JM Anderson	Pacer	NaN	NaN	NaN
3	18.0	8	4	0	225	7	not out	NaN	NaN	NaN	NaN	NaN	NaN
4	15.0	13	1	0	115.38	7	caught	JM Anderson	SCJ Broad	Pacer	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
120	131.0	132	14	1	99.24	3	caught	SS Iyer	Mohammed Shami	Pacer	NaN	NaN	NaN
121	76.0	94	3	0	80.85	3	lbw	NaN	A Nortje	Pacer	NaN	NaN	NaN
122	13.0	17	1	0	76.47	3	caught	JT Smuts	L Ngidi	Pacer	NaN	NaN	NaN
123	20.0	26	3	0	76.92	3	lbw	NaN	AL Phehlukwayo	Pacer	NaN	NaN	NaN
124	14.0	17	0	1	82.35	3	bowled	NaN	MJ Santner	Spin	NaN	NaN	NaN

125 rows × 22 columns

In [17]:
```python
kw['Runs'] = kw['Runs'].astype(float)
kw
```

Out[17]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary
0	0.0	9.0	0.0	0.0	0.00	4.0	bowled	NaN	P Kumar	Pacer	NaN	NaN	NaN
1	0.0	2.0	0.0	0.0	0.00	4.0	caught	TT Samaraweera	AD Mathews	Pacer	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	13.0	27.0	1.0	0.0	48.14	6.0	bowled	NaN	I Sharma	Pacer	NaN	NaN	NaN
4	108.0	132.0	6.0	2.0	81.81	5.0	caught	Raqibul Hasan	Shafiul Islam	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
146	27.0	40.0	3.0	0.0	67.50	3.0	run out	NaN	NaN	NaN	NaN	NaN	NaN
147	67.0	95.0	6.0	0.0	70.52	3.0	caught	RA Jadeja	YS Chahal	Spin	NaN	NaN	NaN
148	30.0	53.0	2.0	0.0	56.60	3.0	caught	JC Buttler	LE Plunkett	Pacer	NaN	NaN	NaN
149	22.0	31.0	2.0	0.0	70.96	3.0	caught	MA Agarwal	YS Chahal	Spin	NaN	NaN	NaN
150	19.0	26.0	2.0	0.0	73.07	3.0	bowled	NaN	A Zampa	Spin	NaN	NaN	NaN

151 rows × 22 columns

In [18]:
```python
jr['Runs'] = jr['Runs'].astype(float)
jr
```

Out[18]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	36.0	50	3	0	72	4	bowled	NaN	RA Jadeja	NaN	NaN	NaN	NaN
2	39.0	57	4	0	68.42	4	caught	MS Dhoni	I Sharma	NaN	NaN	NaN	NaN
3	57.0	45	6	1	126.66	6	not out	NaN	NaN	NaN	NaN	NaN	NaN
4	31.0	49	4	0	63.26	4	bowled	NaN	RA Jadeja	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
142	7.0	30	0	0	23.33	3	caught	TWM Latham	C de Grandhomme	NaN	NaN	NaN	NaN
143	17.0	21	0	0	80.95	3	run out	NaN	NaN	NaN	NaN	NaN	NaN
144	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
145	49.0	52	3	0	94.23	3	caught	T Bavuma	T Shamsi	NaN	NaN	NaN	NaN
146	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

147 rows × 22 columns

In [ ]:

In [19]:
```python
#total runs in all formats
rgs_total_runs=rgs["Runs"].sum()
int(rgs_total_runs)
```

Out[19]: 9115

In [20]:
```python
vk_total_runs=vk['Runs'].sum()
int(vk_total_runs)
```

Out[20]: 11867

In [21]:
```python
ss_total_runs=ss['Runs'].sum()
int(ss_total_runs)
```

Out[21]: 4162

In [22]:
```python
kw_total_runs=kw['Runs'].sum()
int(kw_total_runs)
```

Out[22]: 6173

In [23]:
```python
jr_total_runs=jr['Runs'].sum()
int(jr_total_runs)
```

Out[23]: 5922

In [ ]:

In [24]:
```python
rgs_total_matches=len(rgs.axes[0])
rgs_total_matches
```

Out[24]: 224

In [25]:
```python
vk_total_matches=len(vk.axes[0])
vk_total_matches
```

Out[25]: 248

In [26]:
```python
ss_total_matches=len(ss.axes[0])
ss_total_matches
```

Out[26]: 125

In [27]:
```python
kw_total_matches=len(kw.axes[0])
kw_total_matches
```

Out[27]: 151

In [28]:
```python
jr_total_matches=len(jr.axes[0])
jr_total_matches
```

Out[28]: 147

In [ ]:

In [29]:
```python
rgs_innings_played=0
for i in rgs['Runs']:
    if i >=0:
        rgs_innings_played+=1
        #print(i)
print(rgs_innings_played)
```

217

In [30]:
```python
vk_innings_played=0
for i in vk['Runs']:
    if i >=0:
        vk_innings_played+=1
        #print(i)
print(vk_innings_played)
```

239

In [31]:
```python
ss_innings_played=0
for i in ss['Runs']:
    if i >=0:
        ss_innings_played+=1
        #print(i)
print(ss_innings_played)
```

110

In [32]:
```python
kw_innings_played=0
for i in kw['Runs']:
    if i >=0:
        kw_innings_played+=1
        #print(i)
print(kw_innings_played)
```

144

In [33]:
```python
jr_innings_played=0
for i in jr['Runs']:
    if i >=0:
        jr_innings_played+=1
        #print(i)
print(jr_innings_played)
```

137

In [ ]:

In [34]:
```python
#df[(df['Format']=='ODI') & (df['Dismissal']=='not out')]
rgs_Not_outs=len(rgs[(rgs['Format']=='ODI') & (rgs['Dismissal']=='not out')])
print(rgs_Not_outs)
```

32

In [35]:
```python
rgs['Dismissal'].value_counts()
```

Out[35]:
```
caught      122
not out      32
bowled       29
lbw          18
run out      13
stumped       3
Name: Dismissal, dtype: int64
```

In [36]:
```python
#df[(df['Format']=='ODI') & (df['Dismissal']=='not out')]
vk_Not_outs=len(vk[(vk['Format']=='ODI') & (vk['Dismissal']=='not out')])
print(vk_Not_outs)
```

39

In [37]:
```python
vk['Dismissal'].value_counts()
```

Out[37]:
```
caught        142
not out        39
bowled         27
lbw            14
run out        12
stumped         4
hit wicket      1
Name: Dismissal, dtype: int64
```

In [38]:
```python
#df[(df['Format']=='ODI') & (df['Dismissal']=='not out')]
ss_Not_outs=len(ss[(ss['Format']=='ODI') & (ss['Dismissal']=='not out')])
print(ss_Not_outs)
```

12

In [39]:
```python
ss['Dismissal'].value_counts()
```

Out[39]:
```
caught      58
lbw         16
bowled      16
not out     12
run out      5
stumped      3
Name: Dismissal, dtype: int64
```

In [40]:
```python
#df[(df['Format']=='ODI') & (df['Dismissal']=='not out')]
kw_Not_outs=len(kw[(kw['Format']=='ODI') & (kw['Dismissal']=='not out')])
print(kw_Not_outs)
```

```
14
```

In [41]:
```python
kw['Dismissal'].value_counts()
```

Out[41]:
```
caught      80
bowled      25
not out     14
lbw         13
run out      9
stumped      3
Name: Dismissal, dtype: int64
```

In [42]:
```python
#df[(df['Format']=='ODI') & (df['Dismissal']=='not out')]
jr_Not_outs=len(jr[(jr['Format']=='ODI') & (jr['Dismissal']=='not out')])
print(jr_Not_outs)
```

```
21
```

In [43]:
```python
jr['Dismissal'].value_counts()
```

Out[43]:
```
caught      67
bowled      26
not out     21
lbw         14
run out      6
stumped      3
Name: Dismissal, dtype: int64
```

In [ ]:

In [44]:
```python
#Current ODI Average
rgs_total_runs/(rgs_innings_played-rgs_Not_outs)
```

Out[44]: 49.270270270270274

In [45]:
```python
#Current ODI Average
vk_total_runs/(vk_innings_played-vk_Not_outs)
```

Out[45]: 59.335

In [46]:
```python
#Current ODI Average
ss_total_runs/(ss_innings_played-ss_Not_outs)
```

Out[46]: 42.46938775510204

In [47]:
```python
#Current ODI Average
kw_total_runs/(kw_innings_played-kw_Not_outs)
```

Out[47]: 47.48461538461538

In [48]:
```python
#Current ODI Average
jr_total_runs/(jr_innings_played-jr_Not_outs)
```

Out[48]: 51.05172413793103

In [ ]:

In [49]:
```python
#Here outs include some of the matches which he did'nt get to bat also
rgs_out=0
rgs_notout=0
for i in rgs['Dismissal']:
    if i=='not out':
        rgs_notout+=1
    else:
        rgs_out+=1
rgs_matches_missed=rgs_total_matches-rgs_innings_played
print("Total matches played:",rgs_total_matches)
print("Matches he did not bat:",rgs_matches_missed)
print("Matches he got batting (total innings):",rgs_innings_played)
print("Dismissed:",rgs_out-rgs_matches_missed)
print("Total Notouts:",rgs_notout)
print("current average:",rgs_total_runs/(rgs_out-rgs_matches_missed))#runs/dismissals
```

```
Total matches played: 224
Matches he did not bat: 7
Matches he got batting (total innings): 217
Dismissed: 185
Total Notouts: 32
current average: 49.270270270270274
```

In [50]:
```python
#Here outs include some of the matches which he did'nt get to bat also
vk_out=0
vk_notout=0
for i in vk['Dismissal']:
    if i=='not out':
        vk_notout+=1
    else:
        vk_out+=1
vk_matches_missed=vk_total_matches-vk_innings_played
print("Total matches played:",vk_total_matches)
print("Matches he did not bat:",vk_matches_missed)
print("Matches he got batting (total innings):",vk_innings_played)
print("Dismissed:",vk_out-vk_matches_missed)
print("Total Notouts:",vk_notout)
print("current average:",vk_total_runs/(vk_out-vk_matches_missed)) #runs/dismissals
```

```
Total matches played: 248
Matches he did not bat: 9
Matches he got batting (total innings): 239
Dismissed: 200
Total Notouts: 39
current average: 59.335
```

In [51]:
```python
#Here outs include some of the matches which he did'nt get to bat also
ss_out=0
ss_notout=0
for i in ss['Dismissal']:
    if i=='not out':
        ss_notout+=1
    else:
        ss_out+=1
ss_matches_missed=ss_total_matches-ss_innings_played
print("Total matches played:",ss_total_matches)
print("Matches he did not bat:",ss_matches_missed)
print("Matches he got batting (total innings):",ss_innings_played)
print("Dismissed:",ss_out-ss_matches_missed)
print("Total Notouts:",ss_notout)
print("current average:",ss_total_runs/(ss_out-ss_matches_missed)) #runs/dismissals
```

```
Total matches played: 125
Matches he did not bat: 15
Matches he got batting (total innings): 110
Dismissed: 98
Total Notouts: 12
current average: 42.46938775510204
```

In [52]:
```python
#Here outs include some of the matches which he did'nt get to bat also
kw_out=0
kw_notout=0
for i in kw['Dismissal']:
    if i=='not out':
        kw_notout+=1
    else:
        kw_out+=1
kw_matches_missed=kw_total_matches-kw_innings_played
print("Total matches played:",kw_total_matches)
print("Matches he did not bat:",kw_matches_missed)
print("Matches he got batting (total innings):",kw_innings_played)
print("Dismissed:",kw_out-kw_matches_missed)
print("Total Notouts:",kw_notout)
print("current average:",kw_total_runs/(kw_out-kw_matches_missed)) #runs/dismissals
```

```
Total matches played: 151
Matches he did not bat: 7
Matches he got batting (total innings): 144
Dismissed: 130
Total Notouts: 14
current average: 47.48461538461538
```

In [53]:
```python
#Here outs include some of the matches which he did'nt get to bat also
jr_out=0
jr_notout=0
for i in jr['Dismissal']:
    if i=='not out':
        jr_notout+=1
    else:
        jr_out+=1
jr_matches_missed=jr_total_matches-jr_innings_played
print("Total matches played:",jr_total_matches)
print("Matches he did not bat:",jr_matches_missed)
print("Matches he got batting (total innings):",jr_innings_played)
print("Dismissed:",jr_out-jr_matches_missed)
print("Total Notouts:",jr_notout)
print("current average:",jr_total_runs/(jr_out-jr_matches_missed)) #runs/dismissals
```

```
Total matches played: 147
Matches he did not bat: 10
Matches he got batting (total innings): 137
Dismissed: 116
Total Notouts: 21
current average: 51.05172413793103
```

In [ ]:

In [54]:
```python
rgs_runs_list=list(rgs['Runs'])
print(len(rgs_runs_list))
print(rgs_runs_list)
```

```
224
[nan, 8.0, 1.0, 52.0, 29.0, 0.0, 39.0, 70.0, 1.0, 24.0, 1.0, 3.0, 66.0, 2.0, 9.0, 26.0, 24.0, 11.0,
0.0, 22.0, 58.0, 22.0, 3.0, 19.0, 0.0, 32.0, 18.0, 3.0, 11.0, 3.0, 28.0, 8.0, 25.0, 4.0, 15.0, nan, 4
3.0, 4.0, 0.0, 11.0, nan, 48.0, 114.0, 101.0, 13.0, 32.0, 0.0, 22.0, 69.0, 41.0, 4.0, 0.0, 11.0, 5.0,
44.0, nan, 11.0, 9.0, 23.0, 1.0, 5.0, 68.0, 7.0, 86.0, 39.0, 57.0, 0.0, 72.0, 90.0, 95.0, 27.0, 21.0,
21.0, 10.0, 33.0, 15.0, 0.0, nan, 4.0, 68.0, 5.0, 0.0, 0.0, 4.0, 4.0, 4.0, 83.0, 4.0, 65.0, 52.0, 18.
0, 33.0, 9.0, 60.0, 5.0, 46.0, 48.0, 58.0, 20.0, 1.0, 14.0, 64.0, 42.0, 141.0, 11.0, 9.0, 79.0, 209.
0, 72.0, 12.0, 4.0, 18.0, 19.0, nan, 3.0, 20.0, 39.0, 79.0, 4.0, 21.0, 13.0, 56.0, 18.0, 52.0, 264.0,
9.0, 138.0, 15.0, 0.0, 57.0, 7.0, 64.0, 16.0, 137.0, 34.0, 63.0, 0.0, 29.0, 150.0, 3.0, 65.0, 21.0, 1
6.0, 171.0, 124.0, 6.0, 41.0, 99.0, 14.0, 15.0, 13.0, 11.0, 70.0, 91.0, 78.0, 12.0, 123.0, 0.0, 4.0,
54.0, 124.0, 104.0, 16.0, 28.0, 7.0, 71.0, 65.0, 125.0, 20.0, 7.0, 147.0, 2.0, 208.0, 7.0, 20.0, 15.
0, 0.0, 5.0, 115.0, 15.0, 137.0, 15.0, 2.0, 23.0, 52.0, 83.0, 111.0, 48.0, 152.0, 4.0, 8.0, 162.0, 6
3.0, 133.0, 43.0, 9.0, 11.0, 87.0, 62.0, 7.0, 2.0, 37.0, 0.0, 14.0, 95.0, 56.0, 122.0, 57.0, 140.0,
1.0, 18.0, 102.0, 104.0, 103.0, 1.0, nan, 18.0, 10.0, 36.0, 159.0, 63.0, 10.0, 42.0, 119.0]
```

In [55]:
```python
vk_runs_list=list(vk['Runs'])
print(len(vk_runs_list))
print(vk_runs_list)
```

```
248
[12.0, 37.0, 25.0, 54.0, 31.0, 2.0, 16.0, nan, 79.0, 30.0, 10.0, 27.0, 54.0, 107.0, nan, 9.0, 91.0, 7
1.0, 102.0, 2.0, 31.0, nan, 57.0, 0.0, 82.0, 18.0, 68.0, 11.0, 18.0, 10.0, 28.0, 0.0, 8.0, 37.0, 118.
0, 105.0, 64.0, 63.0, 0.0, 2.0, 54.0, 22.0, 28.0, 87.0, 2.0, 100.0, 8.0, 34.0, 12.0, 1.0, 59.0, 24.0,
9.0, 35.0, 2.0, 81.0, 0.0, 22.0, 94.0, 55.0, 9.0, 7.0, 16.0, 107.0, 37.0, 112.0, 35.0, 86.0, 0.0, 3.
0, 117.0, 20.0, 23.0, 80.0, 31.0, 77.0, 18.0, 15.0, 12.0, 66.0, 21.0, 133.0, 108.0, 66.0, 183.0, 106.
0, 1.0, 38.0, 128.0, 23.0, 0.0, 6.0, 7.0, 15.0, 37.0, 77.0, 26.0, 0.0, 31.0, 22.0, 22.0, 58.0, 43.0,
11.0, 2.0, 102.0, 31.0, 2.0, 115.0, 14.0, 68.0, nan, nan, 61.0, 100.0, 68.0, nan, 115.0, 0.0, 86.0, 9
9.0, 19.0, 31.0, 0.0, nan, 123.0, 78.0, 6.0, 2.0, 82.0, 136.0, 48.0, 5.0, nan, 0.0, 40.0, 1.0, 13.0,
2.0, 62.0, 127.0, 22.0, 49.0, 53.0, 66.0, 139.0, 9.0, 4.0, 3.0, 8.0, 107.0, 46.0, 33.0, 33.0, 44.0, 3
8.0, 3.0, 1.0, 1.0, 23.0, 25.0, 11.0, 12.0, 77.0, 138.0, 7.0, 91.0, 59.0, 117.0, 106.0, 8.0, 85.0, 9.
0, 154.0, 45.0, 65.0, 122.0, 8.0, 55.0, 81.0, 0.0, 76.0, 96.0, 5.0, 32.0, 87.0, 11.0, 3.0, 111.0, 82.
0, 4.0, 3.0, 131.0, 110.0, 0.0, 92.0, 28.0, 21.0, 39.0, 121.0, 29.0, 113.0, 112.0, 46.0, 160.0, 75.0,
36.0, 129.0, 75.0, 45.0, 71.0, 140.0, 157.0, 107.0, 16.0, 33.0, 3.0, 104.0, 46.0, 45.0, 43.0, 60.0, 4
4.0, 116.0, 123.0, 7.0, 20.0, 18.0, 82.0, 77.0, 67.0, 72.0, 66.0, 26.0, 34.0, 1.0, nan, 120.0, 114.0,
4.0, 0.0, 85.0, 16.0, 78.0, 89.0, 51.0, 15.0, 9.0]
```

In [56]:
```python
ss_runs_list=list(ss['Runs'])
print(len(ss_runs_list))
print(ss_runs_list)
```

```
125
[nan, 41.0, 20.0, 18.0, 15.0, nan, 12.0, 33.0, nan, 5.0, 0.0, 26.0, 46.0, 24.0, 0.0, 11.0, nan, nan,
17.0, nan, 25.0, 1.0, nan, 5.0, nan, nan, 18.0, nan, 26.0, nan, 8.0, 21.0, 8.0, 19.0, 1.0, 31.0, 36.
0, 10.0, 101.0, 12.0, 77.0, 10.0, 73.0, 104.0, 67.0, 37.0, 47.0, 102.0, nan, 40.0, 5.0, 4.0, 95.0, 7
2.0, nan, 65.0, 105.0, 56.0, 21.0, 44.0, 70.0, 25.0, 5.0, 12.0, 149.0, 46.0, 41.0, 51.0, 28.0, 18.0,
2.0, 21.0, 6.0, 8.0, 52.0, 74.0, nan, 78.0, 46.0, 58.0, 30.0, 59.0, 8.0, 14.0, 108.0, 21.0, 0.0, 164.
0, 72.0, 0.0, 0.0, 60.0, 108.0, 49.0, 4.0, 8.0, 22.0, 56.0, 1.0, 59.0, 63.0, 3.0, 16.0, 23.0, 18.0, 4
5.0, 4.0, 12.0, 18.0, 73.0, 69.0, 10.0, 73.0, 1.0, 38.0, 5.0, 7.0, 85.0, nan, 98.0, 131.0, 76.0, 13.
0, 20.0, 14.0]
```

In [57]:
```python
kw_runs_list=list(kw['Runs'])
print(len(kw_runs_list))
print(kw_runs_list)
```

```
151
[0.0, 0.0, nan, 13.0, 108.0, 0.0, 25.0, 29.0, 21.0, 42.0, 15.0, 34.0, 5.0, 38.0, 22.0, nan, 14.0, 10
0.0, 35.0, 4.0, 38.0, 55.0, 13.0, 22.0, 24.0, 58.0, 9.0, 3.0, 69.0, nan, 5.0, 21.0, nan, 5.0, 145.0,
6.0, 74.0, 33.0, 7.0, 0.0, 55.0, 19.0, 16.0, 18.0, 67.0, nan, 8.0, 47.0, 16.0, 71.0, 77.0, 65.0, 60.
0, 88.0, 10.0, 70.0, 46.0, 123.0, 97.0, 15.0, 103.0, 26.0, 97.0, 54.0, 112.0, 57.0, 38.0, 9.0, 45.0,
33.0, 1.0, 33.0, 6.0, 12.0, 45.0, 93.0, 118.0, 90.0, 50.0, 97.0, nan, 90.0, 47.0, 7.0, 39.0, 59.0, 1
2.0, 61.0, 10.0, 84.0, 0.0, 60.0, 18.0, 3.0, 118.0, 22.0, 41.0, 27.0, 9.0, 81.0, 13.0, 31.0, 14.0, 9
5.0, 24.0, 36.0, 59.0, 69.0, 23.0, 21.0, 9.0, 100.0, 87.0, 57.0, 6.0, 3.0, 64.0, 38.0, 115.0, 19.0, 7
3.0, 32.0, 22.0, 8.0, 112.0, 45.0, 14.0, 27.0, 1.0, 76.0, 1.0, 55.0, 64.0, 20.0, 28.0, 11.0, 39.0, 1
1.0, 65.0, nan, 40.0, 79.0, 106.0, 148.0, 41.0, 40.0, 27.0, 67.0, 30.0, 22.0, 19.0]
```

In [58]:
```python
jr_runs_list=list(jr['Runs'])
print(len(jr_runs_list))
print(jr_runs_list)
```

```
147
[nan, 36.0, 39.0, 57.0, 31.0, 56.0, 79.0, 28.0, 30.0, 28.0, 33.0, 12.0, 68.0, 38.0, 48.0, 7.0, 3.0, 1
2.0, 0.0, 21.0, 3.0, 2.0, 55.0, 37.0, 23.0, 107.0, 17.0, 45.0, 0.0, nan, 43.0, 10.0, 4.0, 2.0, 44.0,
113.0, 2.0, 42.0, 48.0, 36.0, 104.0, 55.0, 80.0, 5.0, nan, 69.0, 3.0, 25.0, 5.0, 46.0, 1.0, 121.0, 2
9.0, nan, 104.0, 6.0, 54.0, 106.0, 4.0, 0.0, 63.0, 11.0, 71.0, 52.0, 38.0, 125.0, 109.0, 27.0, 2.0, n
an, 11.0, 65.0, 93.0, 61.0, 89.0, 85.0, 30.0, 9.0, 78.0, 54.0, 4.0, 90.0, 101.0, 49.0, 73.0, 37.0, 3
9.0, 2.0, 133.0, 64.0, 15.0, 46.0, 54.0, nan, 84.0, 14.0, 46.0, 91.0, 46.0, 27.0, 0.0, 62.0, 71.0, 9.
0, 20.0, 102.0, 23.0, 29.0, 50.0, 22.0, 4.0, 27.0, 1.0, 3.0, 113.0, 100.0, 25.0, 71.0, 8.0, 32.0, 10.
0, 102.0, 36.0, nan, 5.0, 1.0, 7.0, nan, 40.0, 43.0, 36.0, 84.0, 51.0, 107.0, 21.0, 100.0, 88.0, 57.
0, 8.0, 44.0, 24.0, 49.0, 7.0, 17.0, nan, 49.0, nan]
```

In [ ]:

In [59]:
```python
#to replace all of the types
b=['caught','not out','bowled','lbw','run out','stumped','hit wicket']
z=[1,2,3,4,5,6,7]
```

In [60]:
```python
rgs.replace(b,z,inplace=True)
rgs['Dismissal'] = rgs['Dismissal'].astype(float) #converting datatype to float
#rgs
```

In [61]:
```python
rgs_dismissals_list=list(rgs['Dismissal'])
print(len(rgs_dismissals_list))
print(rgs_dismissals_list)
```

```
224
[nan, 3.0, 1.0, 1.0, 1.0, 1.0, 2.0, 2.0, 1.0, 1.0, 1.0, 2.0, 3.0, 1.0, 3.0, 1.0, 1.0, 5.0, 2.0, 1.0,
1.0, 2.0, 4.0, 1.0, 4.0, 1.0, 1.0, 4.0, 2.0, 1.0, 1.0, 2.0, 2.0, 2.0, 1.0, nan, 2.0, 1.0, 1.0, 1.0, n
an, 1.0, 1.0, 2.0, 5.0, 3.0, 4.0, 4.0, 5.0, 1.0, 1.0, 4.0, 4.0, 6.0, 1.0, nan, 1.0, 4.0, 3.0, 1.0, 3.
0, 2.0, 2.0, 2.0, 1.0, 3.0, 2.0, 3.0, 2.0, 5.0, 3.0, 3.0, 1.0, 1.0, 1.0, 5.0, 1.0, nan, 5.0, 1.0, 3.
0, 3.0, 4.0, 4.0, 3.0, 1.0, 4.0, 1.0, 1.0, 1.0, 1.0, 3.0, 3.0, 1.0, 1.0, 1.0, 2.0, 3.0, 1.0, 1.0, 1.
0, 2.0, 1.0, 2.0, 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0, 5.0, 1.0, nan, 1.0, 1.0, 1.0, 1.0, 1.0, 3.0, 4.
0, 1.0, 2.0, 1.0, 1.0, 3.0, 1.0, 1.0, 5.0, 2.0, 1.0, 3.0, 1.0, 3.0, 3.0, 1.0, 1.0, 1.0, 1.0, 3.0, 1.
0, 1.0, 1.0, 2.0, 5.0, 1.0, 1.0, 1.0, 4.0, 1.0, 4.0, 1.0, 1.0, 5.0, 1.0, 1.0, 2.0, 4.0, 5.0, 4.0, 2.
0, 1.0, 1.0, 1.0, 1.0, 1.0, 5.0, 1.0, 3.0, 1.0, 1.0, 1.0, 2.0, 3.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.
0, 3.0, 1.0, 1.0, 3.0, 2.0, 2.0, 1.0, 2.0, 1.0, 3.0, 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0, 6.0, 1.0, 3.
0, 1.0, 1.0, 4.0, 1.0, 6.0, 2.0, 1.0, 1.0, 3.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, nan, 1.0, 5.0, 1.0, 1.
0, 1.0, 4.0, 1.0]
```

In [62]:
```python
vk.replace(b,z,inplace=True)
#vk
```

In [63]:
```python
vk_dismissals_list=list(vk['Dismissal'])
print(len(vk_dismissals_list))
print(vk_dismissals_list)
```

```
248
[4.0, 1.0, 5.0, 3.0, 4.0, 2.0, 1.0, nan, 2.0, 1.0, 1.0, 3.0, 4.0, 1.0, nan, 1.0, 1.0, 2.0, 2.0, 1.0,
1.0, nan, 1.0, 5.0, 1.0, 1.0, 1.0, 6.0, 3.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0,
1.0, 5.0, 1.0, 2.0, 1.0, 2.0, 3.0, 5.0, 3.0, 1.0, 3.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 4.0, 6.0, 5.0, 3.0,
1.0, 1.0, 1.0, 7.0, 1.0, 2.0, 4.0, 2.0, 3.0, 3.0, 1.0, 4.0, 2.0, 1.0, 1.0, 5.0, 1.0, 4.0, 1.0, 1.0,
1.0, 2.0, 1.0, 3.0, 1.0, 1.0, 1.0, 1.0, 2.0, 4.0, 3.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 1.0, 3.0,
2.0, 2.0, 1.0, 1.0, 1.0, 1.0, 4.0, 1.0, 1.0, 1.0, 2.0, nan, nan, 4.0, 2.0, 1.0, nan, 2.0, 5.0, 1.0,
1.0, 1.0, 1.0, 1.0, nan, 1.0, 1.0, 1.0, 1.0, 1.0, 3.0, 3.0, 1.0, nan, 1.0, 1.0, 2.0, 1.0, 1.0, 1.0,
5.0, 1.0, 1.0, 1.0, 5.0, 2.0, 1.0, 1.0, 2.0, 1.0, 1.0, 1.0, 2.0, 1.0, 2.0, 3.0, 1.0, 1.0, 1.0, 4.0,
3.0, 1.0, 5.0, 1.0, 1.0, 1.0, 1.0, 5.0, 1.0, 1.0, 1.0, 2.0, 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0,
1.0, 2.0, 2.0, 1.0, 2.0, 1.0, 1.0, 1.0, 2.0, 2.0, 3.0, 1.0, 1.0, 2.0, 1.0, 1.0, 3.0, 1.0, 3.0, 1.0, 1.0,
1.0, 1.0, 1.0, 2.0, 2.0, 1.0, 5.0, 2.0, 6.0, 4.0, 3.0, 6.0, 2.0, 3.0, 1.0, 2.0, 2.0, 1.0, 1.0, 1.0,
1.0, 1.0, 4.0, 1.0, 3.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 4.0, nan, 1.0, 2.0, 3.0,
1.0, 3.0, 1.0, 1.0, 3.0, 3.0, 3.0, 1.0]
```

In [64]:
```python
ss.replace(b,z,inplace=True)
ss['Dismissal'] = ss['Dismissal'].astype(float)
#ss
```

In [65]:
```python
ss_dismissals_list=list(ss['Dismissal'])
print(len(ss_dismissals_list))
print(ss_dismissals_list)
```

```
125
[nan, 1.0, 4.0, 2.0, 1.0, nan, 1.0, 6.0, nan, 1.0, 3.0, 1.0, 2.0, 2.0, 5.0, 1.0, nan, nan, 2.0, nan,
3.0, 3.0, nan, 1.0, nan, nan, 5.0, nan, 1.0, nan, 1.0, 1.0, 1.0, 1.0, 5.0, 3.0, 1.0, 1.0, 1.0, 1.0,
3.0, 1.0, 2.0, 3.0, 1.0, 3.0, 1.0, 2.0, nan, 6.0, 3.0, 1.0, 1.0, 1.0, nan, 4.0, 1.0, 2.0, 1.0, 1.0,
1.0, 1.0, 4.0, 1.0, 1.0, 3.0, 1.0, 1.0, 1.0, 3.0, 1.0, 4.0, 4.0, 4.0, 2.0, 1.0, nan, 5.0, 1.0, 1.0,
1.0, 2.0, 4.0, 1.0, 3.0, 4.0, 3.0, 1.0, 1.0, 1.0, 1.0, 3.0, 2.0, 4.0, 1.0, 2.0, 2.0, 1.0, 1.0, 1.0,
1.0, 1.0, 4.0, 1.0, 4.0, 1.0, 1.0, 6.0, 1.0, 1.0, 4.0, 1.0, 3.0, 4.0, 1.0, 1.0, 4.0, 5.0, nan, 3.0,
1.0, 4.0, 1.0, 4.0, 3.0]
```

In [66]:
```python
kw.replace(b,z,inplace=True)
kw['Dismissal'] = kw['Dismissal'].astype(float)
#kw
```

In [67]:
```python
kw_dismissals_list=list(kw['Dismissal'])
print(len(kw_dismissals_list))
print(kw_dismissals_list)
```

```
151
[3.0, 1.0, nan, 3.0, 1.0, 1.0, 1.0, 3.0, 4.0, 3.0, 1.0, 2.0, 6.0, 2.0, 4.0, nan, 2.0, 2.0, 1.0, 1.0,
4.0, 1.0, 1.0, 5.0, 1.0, 4.0, 3.0, 4.0, 1.0, nan, 1.0, 3.0, nan, 1.0, 2.0, 1.0, 5.0, 3.0, 1.0, 1.0,
3.0, 4.0, 4.0, 2.0, 1.0, nan, 1.0, 1.0, 4.0, 1.0, 6.0, 3.0, 5.0, 1.0, 1.0, 2.0, 1.0, 3.0, 1.0, 1.0,
3.0, 1.0, 5.0, 1.0, 1.0, 1.0, 1.0, 2.0, 2.0, 1.0, 1.0, 1.0, 3.0, 1.0, 1.0, 1.0, 1.0, 1.0, 3.0, 3.0, n
an, 1.0, 1.0, 3.0, 3.0, 1.0, 1.0, 1.0, 3.0, 6.0, 1.0, 1.0, 3.0, 1.0, 1.0, 4.0, 1.0, 1.0, 1.0, 1.0, 4.
0, 1.0, 1.0, 2.0, 1.0, 1.0, 3.0, 1.0, 3.0, 4.0, 5.0, 5.0, 1.0, 5.0, 1.0, 4.0, 1.0, 3.0, 1.0, 1.0, 1.
0, 1.0, 1.0, 1.0, 2.0, 1.0, 3.0, 1.0, 5.0, 3.0, 1.0, 1.0, 1.0, 3.0, 1.0, 1.0, 1.0, 4.0, 2.0, nan, 1.
0, 2.0, 2.0, 1.0, 1.0, 1.0, 5.0, 1.0, 1.0, 1.0, 3.0]
```

In [68]:
```python
jr.replace(b,z,inplace=True)
jr['Dismissal'] = jr['Dismissal'].astype(float)
#jr
```

In [69]:
```python
jr_dismissals_list=list(jr['Dismissal'])
print(len(jr_dismissals_list))
print(jr_dismissals_list)
```

```
147
[nan, 3.0, 1.0, 2.0, 3.0, 3.0, 2.0, 2.0, 3.0, 1.0, 5.0, 1.0, 1.0, 1.0, 3.0, 1.0, 3.0, 1.0, 1.0, 3.0,
4.0, 4.0, 1.0, 1.0, 1.0, 1.0, 1.0, 4.0, 3.0, nan, 1.0, 1.0, 3.0, 6.0, 1.0, 1.0, 1.0, 4.0, 2.0, 3.0,
2.0, 3.0, 4.0, 1.0, nan, 1.0, 1.0, 4.0, 1.0, 1.0, 1.0, 4.0, 1.0, nan, 1.0, 1.0, 3.0, 2.0, 6.0, 4.0,
3.0, 1.0, 6.0, 3.0, 3.0, 5.0, 1.0, 4.0, 3.0, nan, 2.0, 1.0, 3.0, 5.0, 1.0, 1.0, 1.0, 3.0, 1.0, 1.0,
3.0, 2.0, 1.0, 2.0, 1.0, 1.0, 5.0, 4.0, 2.0, 3.0, 1.0, 1.0, 3.0, nan, 4.0, 1.0, 2.0, 2.0, 2.0, 3.0,
1.0, 1.0, 3.0, 1.0, 1.0, 1.0, 2.0, 5.0, 1.0, 1.0, 2.0, 3.0, 1.0, 4.0, 2.0, 2.0, 2.0, 1.0, 3.0, 2.0,
1.0, 1.0, 1.0, nan, 1.0, 1.0, 4.0, nan, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 3.0, 2.0, 1.0, 1.0, 4.0, 1.0,
1.0, 2.0, 1.0, 5.0, nan, 1.0, nan]
```

In [ ]:

In [70]:
```python
#for i in range(len(rgs_dismissals_list)):
    #if (math.isnan(rgs_dismissals_list[i])):
        #print(rgs_dismissals_list[i])
```

In [71]:
```python
#for i in range(len(vk_dismissals_list)):
    #if (math.isnan(vk_dismissals_list[i])):
        #print(vk_dismissals_list[i])
```

In [72]:
```python
#rgs['Dismissal'].value_counts()
```

In [73]:
```python
rgs_without_nan_dis=[]
for i in range(len(rgs_dismissals_list)):
    if (math.isnan(rgs_dismissals_list[i])):
        continue
    rgs_without_nan_dis.append(int(rgs_dismissals_list[i]))
    #print(vk_dismissals_list[i])
print(rgs_without_nan_dis)
print(len(rgs_without_nan_dis))
```

```
[3, 1, 1, 1, 1, 2, 2, 1, 1, 1, 2, 3, 1, 3, 1, 1, 5, 2, 1, 1, 2, 4, 1, 4, 1, 1, 4, 2, 1, 1, 2, 2, 2,
1, 2, 1, 1, 1, 1, 1, 2, 5, 3, 4, 4, 5, 1, 1, 4, 4, 6, 1, 1, 4, 3, 1, 3, 2, 2, 2, 1, 3, 2, 3, 2, 5, 3,
3, 1, 1, 1, 5, 1, 5, 1, 3, 3, 4, 4, 3, 1, 4, 1, 1, 1, 1, 3, 3, 1, 1, 1, 2, 3, 1, 1, 1, 2, 1, 2, 1, 2,
1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 1, 3, 4, 1, 2, 1, 1, 3, 1, 1, 5, 2, 1, 3, 1, 3, 3, 1, 1, 1, 1, 3, 1,
1, 1, 2, 5, 1, 1, 1, 4, 1, 4, 1, 1, 5, 1, 1, 2, 4, 5, 4, 2, 1, 1, 1, 1, 1, 5, 1, 3, 1, 1, 1, 2, 3, 1,
1, 1, 1, 1, 1, 2, 3, 1, 1, 3, 2, 2, 1, 2, 1, 3, 1, 2, 1, 1, 1, 1, 1, 6, 1, 3, 1, 1, 4, 1, 6, 2, 1, 1,
3, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 4, 1]
217
```

In [74]:
```python
vk_without_nan_dis=[]
for i in range(len(vk_dismissals_list)):
    if (math.isnan(vk_dismissals_list[i])):
        continue
    vk_without_nan_dis.append(int(vk_dismissals_list[i]))
    #print(vk_dismissals_list[i])
print(vk_without_nan_dis)
print(len(vk_without_nan_dis))
```

```
[4, 1, 5, 3, 4, 2, 1, 2, 1, 1, 3, 4, 1, 1, 1, 2, 2, 1, 1, 1, 5, 1, 1, 1, 6, 3, 1, 1, 1, 1, 1, 1, 1,
1, 2, 1, 1, 1, 5, 1, 2, 1, 2, 3, 5, 3, 1, 3, 1, 1, 1, 1, 1, 4, 6, 5, 3, 1, 1, 1, 7, 1, 2, 4, 2, 3, 3,
1, 4, 2, 1, 1, 5, 1, 4, 1, 1, 1, 2, 1, 3, 1, 1, 1, 1, 2, 4, 3, 1, 1, 1, 1, 2, 1, 1, 1, 3, 2, 2, 1, 1,
1, 1, 4, 1, 1, 1, 2, 4, 2, 1, 2, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 1, 1, 2, 1, 1, 1, 5, 1, 1,
1, 5, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 3, 1, 1, 1, 4, 3, 1, 5, 1, 1, 1, 1, 5, 1, 1, 1, 2, 1, 2, 1, 1, 1,
1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 2, 3, 1, 1, 2, 1, 3, 1, 3, 1, 1, 1, 1, 1, 2, 2, 1, 5, 2, 6, 4, 3,
6, 2, 3, 1, 2, 1, 1, 1, 1, 1, 4, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 4, 1, 2, 3, 1, 3, 1, 1, 3, 3,
3, 1]
239
```

In [75]:
```python
ss_without_nan_dis=[]
for i in range(len(ss_dismissals_list)):
    if (math.isnan(ss_dismissals_list[i])):
        continue
    ss_without_nan_dis.append(int(ss_dismissals_list[i]))
    #print(ss_dismissals_list[i])
print(ss_without_nan_dis)
print(len(ss_without_nan_dis))
```

```
[1, 4, 2, 1, 1, 6, 1, 3, 1, 2, 2, 5, 1, 2, 3, 3, 1, 5, 1, 1, 1, 1, 1, 5, 3, 1, 1, 1, 1, 3, 1, 2, 3,
1, 3, 1, 2, 6, 3, 1, 1, 1, 4, 1, 2, 1, 1, 1, 1, 4, 1, 1, 3, 1, 1, 1, 3, 1, 4, 4, 4, 2, 1, 5, 1, 1, 1,
2, 4, 1, 3, 4, 3, 1, 1, 1, 1, 3, 2, 4, 1, 2, 2, 1, 1, 1, 1, 1, 4, 1, 4, 1, 1, 6, 1, 1, 4, 1, 3, 4, 1,
1, 4, 5, 3, 1, 4, 1, 4, 3]
110
```

In [76]:
```python
kw_without_nan_dis=[]
for i in range(len(kw_dismissals_list)):
    if (math.isnan(kw_dismissals_list[i])):
        continue
    kw_without_nan_dis.append(int(kw_dismissals_list[i]))
    #print(kw_dismissals_list[i])
print(kw_without_nan_dis)
print(len(kw_without_nan_dis))
```

```
[3, 1, 3, 1, 1, 1, 3, 4, 3, 1, 2, 6, 2, 4, 2, 2, 1, 1, 4, 1, 1, 5, 1, 4, 3, 4, 1, 1, 3, 1, 2, 1, 5,
3, 1, 1, 3, 4, 4, 2, 1, 1, 1, 4, 1, 6, 3, 5, 1, 1, 2, 1, 3, 1, 1, 3, 1, 5, 1, 1, 1, 1, 2, 2, 1, 1, 1,
3, 1, 1, 1, 1, 1, 3, 3, 1, 1, 3, 3, 1, 1, 1, 3, 6, 1, 1, 3, 1, 1, 4, 1, 1, 1, 1, 4, 1, 1, 2, 1, 1, 3,
1, 3, 4, 5, 5, 1, 5, 1, 4, 1, 3, 1, 1, 1, 1, 1, 1, 2, 1, 3, 1, 5, 3, 1, 1, 1, 3, 1, 1, 1, 4, 2, 1, 2,
2, 1, 1, 1, 5, 1, 1, 1, 3]
144
```

In [77]:
```python
jr_without_nan_dis=[]
for i in range(len(jr_dismissals_list)):
    if (math.isnan(jr_dismissals_list[i])):
        continue
    jr_without_nan_dis.append(int(jr_dismissals_list[i]))
    #print(jr_dismissals_list[i])
print(jr_without_nan_dis)
print(len(jr_without_nan_dis))
```

```
[3, 1, 2, 3, 3, 2, 2, 3, 1, 5, 1, 1, 1, 3, 1, 3, 1, 1, 3, 4, 4, 1, 1, 1, 1, 1, 4, 3, 1, 1, 3, 6, 1,
1, 1, 4, 2, 3, 2, 3, 4, 1, 1, 1, 4, 1, 1, 1, 4, 1, 1, 1, 3, 2, 6, 4, 3, 1, 6, 3, 3, 5, 1, 4, 3, 2, 1,
3, 5, 1, 1, 1, 3, 1, 1, 3, 2, 1, 2, 1, 1, 5, 4, 2, 3, 1, 1, 3, 4, 1, 2, 2, 2, 3, 1, 1, 3, 1, 1, 1, 2,
5, 1, 1, 2, 3, 1, 4, 2, 2, 2, 1, 3, 2, 1, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1, 3, 2, 1, 1, 4, 1, 1, 2, 1,
5, 1]
137
```

In [ ]:

In [78]:
```python
rgs_without_nan_runs=[]
for i in range(len(rgs_runs_list)):
    if (math.isnan(rgs_runs_list[i])):
        continue
    rgs_without_nan_runs.append(int(rgs_runs_list[i]))
    #print(c[i])
print(rgs_without_nan_runs)
print(len(rgs_without_nan_runs))
```

```
[8, 1, 52, 29, 0, 39, 70, 1, 24, 1, 3, 66, 2, 9, 26, 24, 11, 0, 22, 58, 22, 3, 19, 0, 32, 18, 3, 11,
3, 28, 8, 25, 4, 15, 43, 4, 0, 11, 48, 114, 101, 13, 32, 0, 22, 69, 41, 4, 0, 11, 5, 44, 11, 9, 23,
1, 5, 68, 7, 86, 39, 57, 0, 72, 90, 95, 27, 21, 21, 10, 33, 15, 0, 4, 68, 5, 0, 0, 4, 4, 4, 83, 4, 6
5, 52, 18, 33, 9, 60, 5, 46, 48, 58, 20, 1, 14, 64, 42, 141, 11, 9, 79, 209, 72, 12, 4, 18, 19, 3, 2
0, 39, 79, 4, 21, 13, 56, 18, 52, 264, 9, 138, 15, 0, 57, 7, 64, 16, 137, 34, 63, 0, 29, 150, 3, 65,
21, 16, 171, 124, 6, 41, 99, 14, 15, 13, 11, 70, 91, 78, 12, 123, 0, 4, 54, 124, 104, 16, 28, 7, 71,
65, 125, 20, 7, 147, 2, 208, 7, 20, 15, 0, 5, 115, 15, 137, 15, 2, 23, 52, 83, 111, 48, 152, 4, 8, 16
2, 63, 133, 43, 9, 11, 87, 62, 7, 2, 37, 0, 14, 95, 56, 122, 57, 140, 1, 18, 102, 104, 103, 1, 18, 1
0, 36, 159, 63, 10, 42, 119]
217
```

In [79]:
```python
vk_without_nan_runs=[]
for i in range(len(vk_runs_list)):
    if (math.isnan(vk_runs_list[i])):
        continue
    vk_without_nan_runs.append(int(vk_runs_list[i]))
    #print(c[i])
print(vk_without_nan_runs)
print(len(vk_without_nan_runs))
```

```
[12, 37, 25, 54, 31, 2, 16, 79, 30, 10, 27, 54, 107, 9, 91, 71, 102, 2, 31, 57, 0, 82, 18, 68, 11, 1
8, 10, 28, 0, 8, 37, 118, 105, 64, 63, 0, 2, 54, 22, 28, 87, 2, 100, 8, 34, 12, 1, 59, 24, 9, 35, 2,
81, 0, 22, 94, 55, 9, 7, 16, 107, 37, 112, 35, 86, 0, 3, 117, 20, 23, 80, 31, 77, 18, 15, 12, 66, 21,
133, 108, 66, 183, 106, 1, 38, 128, 23, 0, 6, 7, 15, 37, 77, 26, 0, 31, 22, 22, 58, 43, 11, 2, 102, 3
1, 2, 115, 14, 68, 61, 100, 68, 115, 0, 86, 99, 19, 31, 0, 123, 78, 6, 2, 82, 136, 48, 5, 0, 40, 1, 1
3, 2, 62, 127, 22, 49, 53, 66, 139, 9, 4, 3, 8, 107, 46, 33, 33, 44, 38, 3, 1, 1, 23, 25, 11, 12, 77,
138, 7, 91, 59, 117, 106, 8, 85, 9, 154, 45, 65, 122, 8, 55, 81, 0, 76, 96, 5, 32, 87, 11, 3, 111, 8
2, 4, 3, 131, 110, 0, 92, 28, 21, 39, 121, 29, 113, 112, 46, 160, 75, 36, 129, 75, 45, 71, 140, 157,
107, 16, 33, 3, 104, 46, 45, 43, 60, 44, 116, 123, 7, 20, 18, 82, 77, 67, 72, 66, 26, 34, 1, 120, 11
4, 4, 0, 85, 16, 78, 89, 51, 15, 9]
239
```

In [80]:
```python
ss_without_nan_runs=[]
for i in range(len(ss_runs_list)):
    if (math.isnan(ss_runs_list[i])):
        continue
    ss_without_nan_runs.append(int(ss_runs_list[i]))
    #print(c[i])
print(ss_without_nan_runs)
print(len(ss_without_nan_runs))
```

```
[41, 20, 18, 15, 12, 33, 5, 0, 26, 46, 24, 0, 11, 17, 25, 1, 5, 18, 26, 8, 21, 8, 19, 1, 31, 36, 10,
101, 12, 77, 10, 73, 104, 67, 37, 47, 102, 40, 5, 4, 95, 72, 65, 105, 56, 21, 44, 70, 25, 5, 12, 149,
46, 41, 51, 28, 18, 2, 21, 6, 8, 52, 74, 78, 46, 58, 30, 59, 8, 14, 108, 21, 0, 164, 72, 0, 0, 60, 10
8, 49, 4, 8, 22, 56, 1, 59, 63, 3, 16, 23, 18, 45, 4, 12, 18, 73, 69, 10, 73, 1, 38, 5, 7, 85, 98, 13
1, 76, 13, 20, 14]
110
```

In [81]:
```python
kw_without_nan_runs=[]
for i in range(len(kw_runs_list)):
    if (math.isnan(kw_runs_list[i])):
        continue
    kw_without_nan_runs.append(int(kw_runs_list[i]))
    #print(c[i])
print(kw_without_nan_runs)
print(len(kw_without_nan_runs))
```

```
[0, 0, 13, 108, 0, 25, 29, 21, 42, 15, 34, 5, 38, 22, 14, 100, 35, 4, 38, 55, 13, 22, 24, 58, 9, 3, 6
9, 5, 21, 5, 145, 6, 74, 33, 7, 0, 55, 19, 16, 18, 67, 8, 47, 16, 71, 77, 65, 60, 88, 10, 70, 46, 12
3, 97, 15, 103, 26, 97, 54, 112, 57, 38, 9, 45, 33, 1, 33, 6, 12, 45, 93, 118, 90, 50, 97, 90, 47, 7,
39, 59, 12, 61, 10, 84, 0, 60, 18, 3, 118, 22, 41, 27, 9, 81, 13, 31, 14, 95, 24, 36, 59, 69, 23, 21,
9, 100, 87, 57, 6, 3, 64, 38, 115, 19, 73, 32, 22, 8, 112, 45, 14, 27, 1, 76, 1, 55, 64, 20, 28, 11,
39, 11, 65, 40, 79, 106, 148, 41, 40, 27, 67, 30, 22, 19]
144
```

In [82]:
```python
jr_without_nan_runs=[]
for i in range(len(jr_runs_list)):
    if (math.isnan(jr_runs_list[i])):
        continue
    jr_without_nan_runs.append(int(jr_runs_list[i]))
    #print(c[i])
print(jr_without_nan_runs)
print(len(jr_without_nan_runs))
```

```
[36, 39, 57, 31, 56, 79, 28, 30, 28, 33, 12, 68, 38, 48, 7, 3, 12, 0, 21, 3, 2, 55, 37, 23, 107, 17,
45, 0, 43, 10, 4, 2, 44, 113, 2, 42, 48, 36, 104, 55, 80, 5, 69, 3, 25, 5, 46, 1, 121, 29, 104, 6, 5
4, 106, 4, 0, 63, 11, 71, 52, 38, 125, 109, 27, 2, 11, 65, 93, 61, 89, 85, 30, 9, 78, 54, 4, 90, 101,
49, 73, 37, 39, 2, 133, 64, 15, 46, 54, 84, 14, 46, 91, 46, 27, 0, 62, 71, 9, 20, 102, 23, 29, 50, 2
2, 4, 27, 1, 3, 113, 100, 25, 71, 8, 32, 10, 102, 36, 5, 1, 7, 40, 43, 36, 84, 51, 107, 21, 100, 88,
57, 8, 44, 24, 49, 7, 17, 49]
137
```

In [ ]:

In [83]:
```python
count=0
rgs_out_counter=[]
for i in range(len(rgs_without_nan_dis)):
    if rgs_without_nan_dis[i]==2:
        count-=1
    count+=1
    rgs_out_counter.append(count)
print(len(rgs_out_counter))
print(rgs_out_counter)
```

```
217
[1, 2, 3, 4, 5, 5, 5, 6, 7, 8, 8, 9, 10, 11, 12, 13, 14, 14, 15, 16, 16, 17, 18, 19, 20, 21, 22, 22,
23, 24, 24, 24, 24, 25, 25, 26, 27, 28, 29, 30, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 4
3, 44, 45, 46, 46, 46, 46, 47, 48, 48, 49, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 6
3, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 75, 76, 77, 78, 79, 79, 80, 80, 81, 81, 82, 83, 8
4, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 96, 97, 98, 99, 100, 101, 102, 102, 103, 104, 105,
106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 1
25, 126, 127, 127, 128, 129, 130, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 141, 14
2, 143, 144, 145, 146, 147, 148, 148, 149, 150, 151, 152, 152, 152, 153, 153, 154, 155, 156, 156, 15
7, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 169, 170, 171, 172, 173, 174, 175, 17
6, 177, 178, 179, 180, 181, 182, 183, 184, 185]
```

In [84]:
```python
count=0
vk_out_counter=[]
for i in range(len(vk_without_nan_dis)):
    if vk_without_nan_dis[i]==2:
        count-=1
    count+=1
    vk_out_counter.append(count)
print(len(vk_out_counter))
print(vk_out_counter)
```

```
239
[1, 2, 3, 4, 5, 5, 6, 6, 7, 8, 9, 10, 11, 12, 13, 13, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27, 28, 29, 30, 30, 31, 32, 33, 34, 35, 35, 36, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 4
7, 48, 49, 50, 51, 52, 53, 54, 55, 56, 56, 57, 58, 59, 60, 60, 61, 62, 63, 64, 65, 66, 67, 68, 6
8, 69, 70, 71, 72, 73, 74, 74, 75, 76, 77, 78, 79, 80, 80, 81, 82, 83, 84, 84, 84, 85, 86, 87, 88, 8
9, 90, 91, 92, 92, 93, 93, 94, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 1
09, 110, 110, 111, 112, 113, 114, 115, 116, 117, 118, 118, 119, 120, 120, 121, 122, 123, 123, 124, 12
4, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 140, 141, 141, 14
2, 143, 144, 145, 146, 146, 147, 147, 147, 148, 148, 149, 150, 151, 151, 151, 152, 153, 154, 154, 15
5, 156, 157, 158, 159, 160, 161, 162, 163, 163, 163, 164, 165, 165, 166, 167, 168, 169, 169, 170, 17
1, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 18
9, 190, 191, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200]
```

In [85]:
```python
count=0
ss_out_counter=[]
for i in range(len(ss_without_nan_dis)):
    if ss_without_nan_dis[i]==2:
        count-=1
    count+=1
    ss_out_counter.append(count)
print(len(ss_out_counter))
print(ss_out_counter)
```

```
110
[1, 2, 2, 3, 4, 5, 6, 7, 8, 8, 8, 9, 10, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27, 27, 28, 29, 30, 31, 31, 32, 33, 34, 35, 36, 37, 38, 38, 39, 40, 41, 42, 43, 44, 45, 46, 4
7, 48, 49, 50, 51, 52, 53, 54, 54, 55, 56, 57, 58, 59, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6
9, 70, 71, 71, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 9
2, 93, 94, 95, 96, 97, 98]
```

In [86]:
```python
count=0
kw_out_counter=[]
for i in range(len(kw_without_nan_dis)):
    if kw_without_nan_dis[i]==2:
        count-=1
    count+=1
    kw_out_counter.append(count)
print(len(kw_out_counter))
print(kw_out_counter)
```

```
144
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 11, 11, 12, 12, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 2
4, 25, 26, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 44, 45, 4
6, 47, 48, 49, 50, 51, 52, 53, 54, 55, 55, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 6
9, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 88, 89, 90, 91, 92, 9
3, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 108, 109, 110, 111, 112, 113,
114, 115, 116, 117, 118, 119, 120, 121, 121, 122, 122, 122, 123, 124, 125, 126, 127, 128, 129, 130]
```

In [87]:
```python
count=0
jr_out_counter=[]
for i in range(len(jr_without_nan_dis)):
    if jr_without_nan_dis[i]==2:
        count-=1
    count+=1
    jr_out_counter.append(count)
print(len(jr_out_counter))
print(jr_out_counter)
```

```
137
[1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
26, 27, 28, 29, 30, 31, 32, 33, 33, 34, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 4
8, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 69, 70, 7
0, 71, 72, 73, 74, 74, 75, 76, 77, 78, 79, 80, 80, 80, 81, 82, 83, 84, 85, 86, 87, 87, 88, 89, 9
0, 90, 91, 92, 93, 93, 93, 93, 94, 95, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 10
8, 108, 109, 110, 111, 112, 113, 113, 114, 115, 116]
```

In [ ]:

In [88]:
```python
rgs_run_counter=[]
for i in range(len(rgs_without_nan_runs)):
    if i==0:
        rgs_run_counter.append(rgs_without_nan_runs[0])
    else:
        rgs_run_counter.append(rgs_without_nan_runs[i]+rgs_run_counter[i-1])
print(rgs_run_counter)
print(len(rgs_run_counter))
```

```
[8, 9, 61, 90, 90, 129, 199, 200, 224, 225, 228, 294, 296, 305, 331, 355, 366, 366, 388, 446, 468, 47
1, 490, 490, 522, 540, 543, 554, 557, 585, 593, 618, 622, 637, 680, 684, 684, 695, 743, 857, 958, 97
1, 1003, 1003, 1025, 1094, 1135, 1139, 1139, 1150, 1155, 1199, 1210, 1219, 1242, 1243, 1248, 1316, 13
23, 1409, 1448, 1505, 1505, 1577, 1667, 1762, 1789, 1810, 1831, 1841, 1874, 1889, 1889, 1893, 1961, 1
966, 1966, 1966, 1970, 1974, 1978, 2061, 2065, 2130, 2182, 2200, 2233, 2242, 2302, 2307, 2353, 2401,
2459, 2479, 2480, 2494, 2558, 2600, 2741, 2752, 2761, 2840, 3049, 3121, 3133, 3137, 3155, 3174, 3177,
3197, 3236, 3315, 3319, 3340, 3353, 3409, 3427, 3479, 3743, 3752, 3890, 3905, 3905, 3962, 3969, 4033,
4049, 4186, 4220, 4283, 4283, 4312, 4462, 4465, 4530, 4551, 4567, 4738, 4862, 4868, 4909, 5008, 5022,
5037, 5050, 5061, 5131, 5222, 5300, 5312, 5435, 5435, 5439, 5493, 5617, 5721, 5737, 5765, 5772, 5843,
5908, 6033, 6053, 6060, 6207, 6209, 6417, 6424, 6444, 6459, 6459, 6464, 6579, 6594, 6731, 6746, 6748,
6771, 6823, 6906, 7017, 7065, 7217, 7221, 7229, 7391, 7454, 7587, 7630, 7639, 7650, 7737, 7799, 7806,
7808, 7845, 7845, 7859, 7954, 8010, 8132, 8189, 8329, 8330, 8348, 8450, 8554, 8657, 8658, 8676, 8686,
8722, 8881, 8944, 8954, 8996, 9115]
217
```

In [89]:
```python
vk_run_counter=[]
for i in range(len(vk_without_nan_runs)):
    if i==0:
        vk_run_counter.append(vk_without_nan_runs[0])
    else:
        vk_run_counter.append(vk_without_nan_runs[i]+vk_run_counter[i-1])
print(vk_run_counter)
print(len(vk_run_counter))
```

```
[12, 49, 74, 128, 159, 161, 177, 256, 286, 296, 323, 377, 484, 493, 584, 655, 757, 759, 790, 847, 84
7, 929, 947, 1015, 1026, 1044, 1054, 1082, 1082, 1090, 1127, 1245, 1350, 1414, 1477, 1477, 1479, 153
3, 1555, 1583, 1670, 1672, 1772, 1780, 1814, 1826, 1827, 1886, 1910, 1919, 1954, 1956, 2037, 2037, 20
59, 2153, 2208, 2217, 2224, 2240, 2347, 2384, 2496, 2531, 2617, 2617, 2620, 2737, 2757, 2780, 2860, 2
891, 2968, 2986, 3001, 3013, 3079, 3100, 3233, 3341, 3407, 3590, 3696, 3697, 3735, 3863, 3886, 3886,
3892, 3899, 3914, 3951, 4028, 4054, 4054, 4085, 4107, 4129, 4187, 4230, 4241, 4243, 4345, 4376, 4378,
4493, 4507, 4575, 4636, 4736, 4804, 4919, 4919, 5005, 5104, 5123, 5154, 5154, 5277, 5355, 5361, 5363,
5445, 5581, 5629, 5634, 5634, 5674, 5675, 5688, 5690, 5752, 5879, 5901, 5950, 6003, 6069, 6208, 6217,
6221, 6224, 6232, 6339, 6385, 6418, 6451, 6495, 6533, 6536, 6537, 6538, 6561, 6586, 6597, 6609, 6686,
6824, 6831, 6922, 6981, 7098, 7204, 7212, 7297, 7306, 7460, 7505, 7570, 7692, 7700, 7755, 7836, 7836,
7912, 8008, 8013, 8045, 8132, 8143, 8146, 8257, 8339, 8343, 8346, 8477, 8587, 8587, 8679, 8707, 8728,
8767, 8888, 8917, 9030, 9142, 9188, 9348, 9423, 9459, 9588, 9663, 9708, 9779, 9919, 10076, 10183, 101
99, 10232, 10235, 10339, 10385, 10430, 10473, 10533, 10577, 10693, 10816, 10823, 10843, 10861, 10943,
11020, 11087, 11159, 11225, 11251, 11285, 11286, 11406, 11520, 11524, 11524, 11609, 11625, 11703, 117
92, 11843, 11858, 11867]
239
```

In [90]:
```python
ss_run_counter=[]
for i in range(len(ss_without_nan_runs)):
    if i==0:
        ss_run_counter.append(ss_without_nan_runs[0])
    else:
        ss_run_counter.append(ss_without_nan_runs[i]+ss_run_counter[i-1])
print(ss_run_counter)
print(len(ss_run_counter))
```

```
[41, 61, 79, 94, 106, 139, 144, 144, 170, 216, 240, 240, 251, 268, 293, 294, 299, 317, 343, 351, 372,
380, 399, 400, 431, 467, 477, 578, 590, 667, 677, 750, 854, 921, 958, 1005, 1107, 1147, 1152, 1156, 1
251, 1323, 1388, 1493, 1549, 1570, 1614, 1684, 1709, 1714, 1726, 1875, 1921, 1962, 2013, 2041, 2059,
2061, 2082, 2088, 2096, 2148, 2222, 2300, 2346, 2404, 2434, 2493, 2501, 2515, 2623, 2644, 2644, 2808,
2880, 2880, 2880, 2940, 3048, 3097, 3101, 3109, 3131, 3187, 3188, 3247, 3310, 3313, 3329, 3352, 3370,
3415, 3419, 3431, 3449, 3522, 3591, 3601, 3674, 3675, 3713, 3718, 3725, 3810, 3908, 4039, 4115, 4128,
4148, 4162]
110
```

In [91]:
```python
kw_run_counter=[]
for i in range(len(kw_without_nan_runs)):
    if i==0:
        kw_run_counter.append(kw_without_nan_runs[0])
    else:
        kw_run_counter.append(kw_without_nan_runs[i]+kw_run_counter[i-1])
print(kw_run_counter)
print(len(kw_run_counter))
```

```
[0, 0, 13, 121, 121, 146, 175, 196, 238, 253, 287, 292, 330, 352, 366, 466, 501, 505, 543, 598, 611,
633, 657, 715, 724, 727, 796, 801, 822, 827, 972, 978, 1052, 1085, 1092, 1092, 1147, 1166, 1182, 120
0, 1267, 1275, 1322, 1338, 1409, 1486, 1551, 1611, 1699, 1709, 1779, 1825, 1948, 2045, 2060, 2163, 21
89, 2286, 2340, 2452, 2509, 2547, 2556, 2601, 2634, 2635, 2668, 2674, 2686, 2731, 2824, 2942, 3032, 3
082, 3179, 3269, 3316, 3323, 3362, 3421, 3433, 3494, 3504, 3588, 3588, 3648, 3666, 3669, 3787, 3809,
3850, 3877, 3886, 3967, 3980, 4011, 4025, 4120, 4144, 4180, 4239, 4308, 4331, 4352, 4361, 4461, 4548,
4605, 4611, 4614, 4678, 4716, 4831, 4850, 4923, 4955, 4977, 4985, 5097, 5142, 5156, 5183, 5184, 5260,
5261, 5316, 5380, 5400, 5428, 5439, 5478, 5489, 5554, 5594, 5673, 5779, 5927, 5968, 6008, 6035, 6102,
6132, 6154, 6173]
144
```

In [92]:
```python
jr_run_counter=[]
for i in range(len(jr_without_nan_runs)):
    if i==0:
        jr_run_counter.append(jr_without_nan_runs[0])
    else:
        jr_run_counter.append(jr_without_nan_runs[i]+jr_run_counter[i-1])
print(jr_run_counter)
print(len(jr_run_counter))
```

```
[36, 75, 132, 163, 219, 298, 326, 356, 384, 417, 429, 497, 535, 583, 590, 593, 605, 605, 626, 629, 63
1, 686, 723, 746, 853, 870, 915, 915, 958, 968, 972, 974, 1018, 1131, 1133, 1175, 1223, 1259, 1363, 1
418, 1498, 1503, 1572, 1575, 1600, 1605, 1651, 1652, 1773, 1802, 1906, 1912, 1966, 2072, 2076, 2076,
2139, 2150, 2221, 2273, 2311, 2436, 2545, 2572, 2574, 2585, 2650, 2743, 2804, 2893, 2978, 3008, 3017,
3095, 3149, 3153, 3243, 3344, 3393, 3466, 3503, 3542, 3544, 3677, 3741, 3756, 3802, 3856, 3940, 3954,
4000, 4091, 4137, 4164, 4164, 4226, 4297, 4306, 4326, 4428, 4451, 4480, 4530, 4552, 4556, 4583, 4584,
4587, 4700, 4800, 4825, 4896, 4904, 4936, 4946, 5048, 5084, 5089, 5090, 5097, 5137, 5180, 5216, 5300,
5351, 5458, 5479, 5579, 5667, 5724, 5732, 5776, 5800, 5849, 5856, 5873, 5922]
137
```

In [ ]:

In [93]:
```python
rgs_average_list=[]
for i in range(len(rgs_run_counter)):
    rgs_average_list.append(round((rgs_run_counter[i]/rgs_out_counter[i]),2))
print(rgs_average_list)
print(len(rgs_average_list))
```

```
[8.0, 4.5, 20.33, 22.5, 18.0, 25.8, 39.8, 33.33, 32.0, 28.12, 28.5, 32.67, 29.6, 27.73, 27.58, 27.31,
26.14, 26.14, 25.87, 27.88, 29.25, 27.71, 27.22, 25.79, 26.1, 25.71, 24.68, 25.18, 24.22, 24.38, 24.7
1, 25.75, 25.92, 25.48, 27.2, 26.31, 25.33, 24.82, 25.62, 28.57, 31.93, 31.32, 31.34, 30.39, 30.15, 3
1.26, 31.53, 30.78, 29.97, 29.49, 28.88, 29.24, 28.81, 28.35, 28.23, 27.62, 27.13, 28.61, 28.76, 30.6
3, 30.81, 31.35, 31.35, 32.18, 34.02, 35.24, 35.08, 34.81, 34.55, 34.09, 34.07, 33.73, 33.14, 32.64,
33.24, 32.77, 32.23, 31.71, 31.27, 30.84, 30.43, 31.23, 30.82, 31.32, 31.62, 31.43, 31.45, 31.14, 31.
53, 31.18, 31.37, 32.01, 32.36, 32.19, 31.79, 31.57, 32.38, 32.5, 34.26, 33.98, 34.09, 34.63, 36.73,
37.15, 36.86, 36.48, 36.26, 36.07, 35.7, 35.52, 35.56, 36.03, 35.69, 35.53, 35.29, 35.51, 35.7, 35.8
7, 38.19, 37.9, 38.9, 38.66, 38.28, 38.84, 38.53, 38.78, 38.56, 39.49, 39.44, 39.66, 39.29, 39.2, 40.
2, 39.87, 40.09, 39.92, 39.71, 41.2, 41.91, 41.61, 41.6, 42.08, 41.85, 41.63, 41.39, 41.15, 41.38, 4
1.78, 42.06, 41.83, 42.8, 42.46, 42.16, 42.25, 43.21, 43.67, 43.46, 43.35, 43.07, 43.28, 43.44, 44.0
4, 43.86, 43.6, 44.34, 44.04, 45.51, 45.24, 45.06, 44.85, 44.54, 44.27, 44.76, 44.55, 45.48, 45.28, 4
4.99, 44.84, 44.89, 45.43, 46.16, 46.18, 47.17, 46.89, 46.64, 47.38, 47.78, 48.32, 48.29, 48.04, 47.8
1, 48.06, 48.14, 47.89, 47.61, 47.55, 47.26, 47.06, 47.35, 47.4, 48.12, 48.17, 48.71, 48.43, 48.25, 4
8.56, 48.88, 49.19, 48.92, 48.74, 48.53, 48.46, 49.07, 49.14, 48.93, 48.89, 49.27]
217
```

In [94]:
```python
vk_average_list=[]
for i in range(len(vk_run_counter)):
    vk_average_list.append(round((vk_run_counter[i]/vk_out_counter[i]),2))
print(vk_average_list)
print(len(vk_average_list))
```

```
[12.0, 24.5, 24.67, 32.0, 31.8, 32.2, 29.5, 42.67, 40.86, 37.0, 35.89, 37.7, 44.0, 41.08, 44.92, 50.3
8, 58.23, 54.21, 52.67, 52.94, 49.82, 51.61, 49.84, 50.75, 48.86, 47.45, 45.83, 45.08, 43.28, 41.92,
41.74, 44.46, 46.55, 47.13, 49.23, 47.65, 46.22, 46.45, 45.74, 45.23, 47.71, 46.44, 49.22, 48.11, 47.
74, 46.82, 45.67, 46.0, 45.48, 44.63, 44.41, 43.47, 44.28, 43.34, 42.9, 43.94, 44.16, 43.47, 42.77, 4
2.26, 43.46, 43.35, 45.38, 45.2, 46.73, 45.91, 45.17, 46.39, 45.95, 46.33, 46.89, 46.63, 47.11, 46.6
6, 46.17, 45.65, 45.96, 45.59, 47.54, 48.42, 48.67, 50.56, 51.33, 50.64, 50.47, 52.2, 51.81, 51.13, 5
0.55, 49.99, 49.54, 49.39, 50.35, 50.05, 49.44, 49.22, 48.89, 49.15, 49.85, 49.76, 49.31, 48.77, 49.3
8, 49.17, 48.64, 49.37, 48.99, 49.73, 49.85, 50.92, 51.11, 52.33, 51.78, 52.14, 52.62, 52.28, 52.06,
51.54, 52.25, 52.5, 52.05, 51.57, 51.86, 52.65, 52.61, 52.17, 51.69, 51.58, 51.59, 51.24, 50.8, 50.9,
51.57, 51.31, 51.29, 51.31, 51.43, 52.61, 52.24, 51.84, 51.87, 51.5, 51.96, 51.91, 52.18, 52.02, 52.3
8, 52.26, 51.87, 51.47, 51.08, 50.86, 50.66, 50.36, 50.07, 50.27, 50.93, 50.6, 50.9, 50.96, 51.43, 5
1.83, 51.51, 52.12, 51.82, 52.91, 52.85, 52.94, 53.42, 53.1, 53.12, 53.67, 53.31, 53.82, 54.48, 54.1
4, 54.36, 54.58, 54.29, 53.95, 54.68, 55.23, 54.89, 54.55, 55.05, 55.76, 55.4, 55.63, 55.46, 55.24, 5
5.14, 55.55, 55.39, 55.74, 56.09, 56.37, 57.35, 57.46, 57.33, 58.11, 58.21, 58.13, 58.21, 58.69, 59.6
2, 59.9, 59.64, 59.84, 59.51, 59.76, 59.68, 59.6, 59.51, 59.51, 59.42, 59.74, 60.09, 59.8, 59.58, 59.
35, 59.47, 59.57, 59.61, 59.67, 59.71, 59.53, 59.71, 59.4, 59.72, 60.31, 60.02, 59.71, 59.84, 59.62,
59.71, 59.86, 59.81, 59.59, 59.34]
239
```

```
[41.0, 30.5, 39.5, 31.33, 26.5, 27.8, 24.0, 20.57, 21.25, 27.0, 30.0, 26.67, 25.1, 26.8, 26.64, 24.5,
23.0, 22.64, 22.87, 21.94, 21.88, 21.11, 21.0, 20.0, 20.52, 21.23, 20.74, 24.08, 23.6, 25.65, 25.07,
27.78, 30.5, 31.76, 31.93, 32.42, 35.71, 35.84, 34.91, 34.0, 35.74, 36.75, 37.51, 39.29, 40.76, 40.2
6, 40.35, 41.07, 40.69, 39.86, 39.23, 41.67, 41.76, 41.74, 41.94, 41.65, 41.18, 40.41, 40.04, 39.4, 3
8.81, 39.78, 40.4, 41.07, 41.16, 41.45, 41.25, 42.25, 41.68, 41.23, 42.31, 41.97, 41.31, 43.2, 43.64,
42.99, 42.35, 42.61, 44.17, 44.24, 43.68, 43.79, 44.1, 44.26, 43.67, 43.88, 44.13, 43.59, 43.23, 42.9
7, 42.66, 42.69, 42.21, 41.84, 41.55, 41.93, 42.25, 41.87, 42.23, 41.76, 41.72, 41.31, 40.93, 41.41,
42.02, 42.97, 43.32, 43.0, 42.76, 42.47]
110
```

In [96]:
```python
kw_average_list=[]
for i in range(len(kw_run_counter)):
    kw_average_list.append(round((kw_run_counter[i]/kw_out_counter[i]),2))
print(kw_average_list)
print(len(kw_average_list))
```

```
[0.0, 0.0, 4.33, 30.25, 24.2, 24.33, 25.0, 24.5, 26.44, 25.3, 28.7, 26.55, 30.0, 29.33, 30.5, 38.83,
38.54, 36.07, 36.2, 37.38, 35.94, 35.17, 34.58, 35.75, 34.48, 33.05, 34.61, 33.38, 32.88, 31.81, 37.3
8, 36.22, 37.57, 37.41, 36.4, 35.23, 35.84, 35.33, 34.76, 35.29, 36.2, 35.42, 35.73, 35.21, 36.13, 3
7.15, 37.83, 38.36, 39.51, 38.84, 40.43, 40.56, 42.35, 43.51, 42.92, 44.14, 43.78, 44.82, 45.0, 46.2
6, 46.46, 46.31, 46.47, 47.29, 47.04, 46.23, 46.0, 45.32, 44.77, 44.77, 45.55, 46.7, 47.38, 47.42, 4
8.17, 48.79, 48.76, 48.16, 48.03, 48.18, 47.68, 47.86, 47.35, 47.84, 47.21, 47.38, 47.0, 46.44, 47.3
4, 47.02, 46.95, 46.71, 46.26, 46.67, 46.28, 46.1, 45.74, 46.82, 46.56, 46.44, 46.58, 46.83, 46.57, 4
6.3, 45.91, 46.47, 46.89, 46.99, 46.58, 46.14, 46.32, 46.24, 46.9, 46.63, 46.89, 46.75, 46.51, 46.16,
47.19, 47.17, 46.87, 46.69, 46.29, 46.55, 46.15, 46.23, 46.38, 46.15, 46.0, 45.71, 45.65, 45.36, 45.
9, 45.85, 46.5, 47.37, 48.19, 48.13, 48.06, 47.9, 48.05, 47.91, 47.71, 47.48]
144
```

In [97]:
```python
jr_average_list=[]
for i in range(len(jr_run_counter)):
    jr_average_list.append(round((jr_run_counter[i]/jr_out_counter[i]),2))
print(jr_average_list)
print(len(jr_average_list))
```

```
[36.0, 37.5, 66.0, 54.33, 54.75, 74.5, 81.5, 71.2, 64.0, 59.57, 53.62, 55.22, 53.5, 53.0, 49.17, 45.6
2, 43.21, 40.33, 39.12, 37.0, 35.06, 36.11, 36.15, 35.52, 38.77, 37.83, 38.12, 36.6, 36.85, 35.85, 3
4.71, 33.59, 33.93, 36.48, 35.41, 35.61, 37.06, 37.03, 40.09, 40.51, 41.61, 40.62, 41.37, 40.38, 40.
0, 39.15, 39.31, 38.42, 40.3, 40.04, 41.43, 40.68, 40.96, 43.17, 42.37, 41.52, 41.94, 41.35, 41.91, 4
2.09, 42.02, 43.5, 44.65, 44.34, 43.63, 43.81, 44.17, 44.97, 45.23, 45.92, 46.53, 46.28, 45.71, 46.1
9, 46.31, 45.7, 47.0, 47.77, 48.47, 48.82, 48.65, 48.52, 47.89, 49.69, 49.88, 49.42, 49.38, 49.44, 4
9.87, 49.42, 50.0, 51.14, 51.71, 51.41, 50.78, 50.92, 51.15, 50.66, 50.3, 50.9, 51.16, 50.91, 50.9, 5
0.58, 50.62, 50.36, 49.83, 49.32, 50.54, 51.61, 51.88, 52.09, 51.62, 51.96, 51.52, 52.04, 51.88, 51.
4, 50.9, 50.47, 50.36, 50.29, 50.15, 50.48, 50.48, 51.01, 50.73, 51.66, 51.99, 52.04, 51.64, 51.57, 5
1.33, 51.76, 51.37, 51.07, 51.05]
137
```

In [ ]:

In [98]:
```python
mm=[]
for i in range(239):
    mm.append(i+1)
print(mm)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 1
23, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 14
3, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 16
3, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 18
3, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 20
3, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 22
3, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239]
```

In [99]:
```python
nn=[]
for i in range(217):
    nn.append(i+1)
print(nn)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 1
23, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 14
3, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 16
3, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 18
3, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 20
3, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217]
```

In [100]:
```python
oo=[]
for i in range(110):
    oo.append(i+1)
print(oo)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110]
```

In [101]:
```python
pp=[]
for i in range(144):
    pp.append(i+1)
print(pp)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 1
23, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 14
3, 144]
```

In [102]:
```python
qq=[]
for i in range(137):
    qq.append(i+1)
print(qq)
```

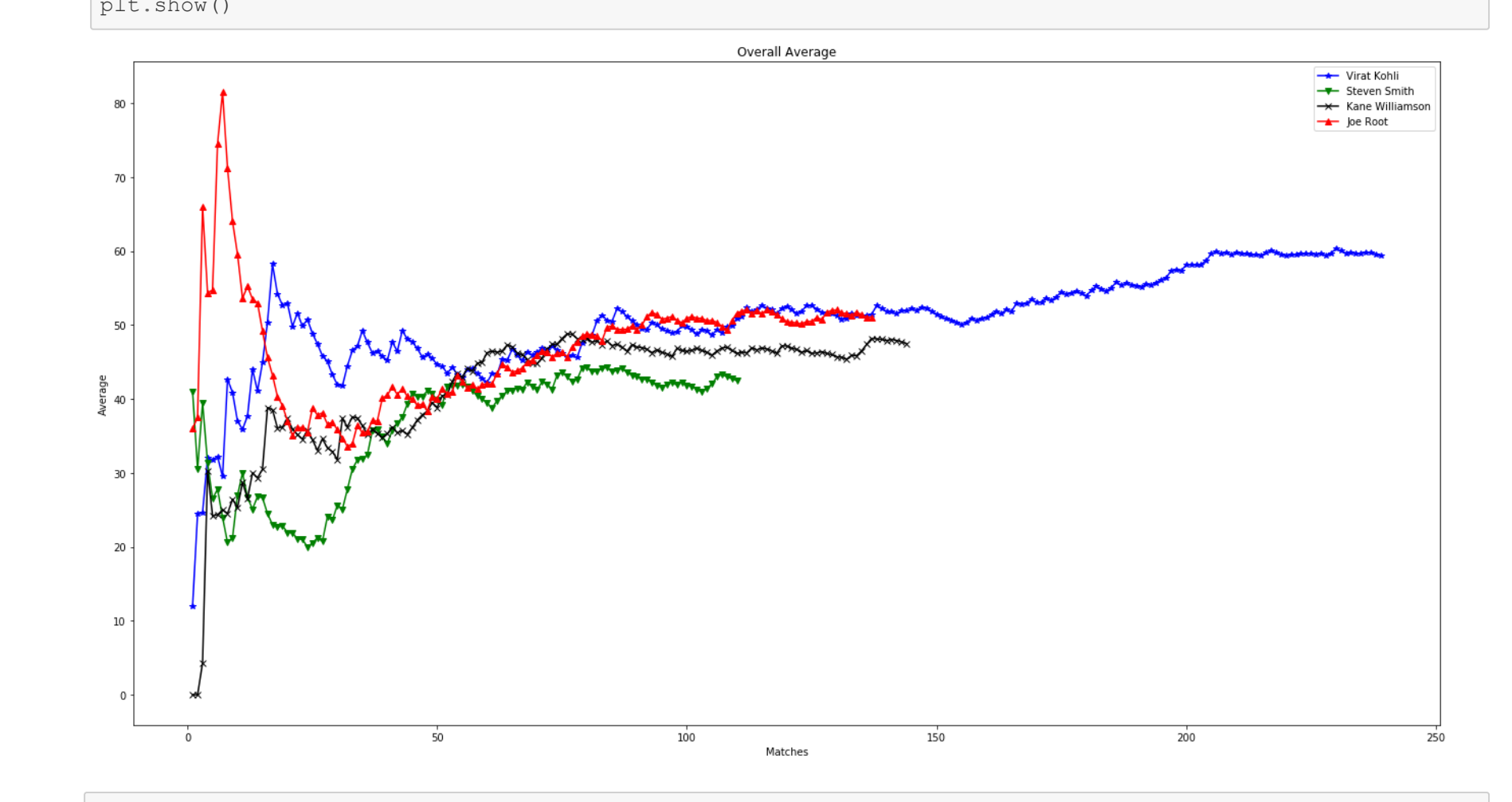
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 1
23, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137]
```

In [ ]:

In [103]:
```python
#plt.hist(average, density=False, bins=30)
#plt.axis([xmin, xmax, ymin, ymax])
#data Visualization starts
```

In [104]:
```python
#import plotly.graph_objects as go
#ani=rgs_average_list
#fig = go.Figure([go.Bar(y=ani, x=mm,width=0.7)])
#fig.show()
```

In [105]:
```python
plt.figure(figsize=(23,12))
plt.plot(mm,vk_average_list, color='blue', marker='*')
#plt.plot(nn,rgs_average_list, color='blue', marker='o')
plt.plot(oo,ss_average_list, color='green', marker='o')
plt.plot(pp,kw_average_list, color='black', marker='x')
plt.plot(qq,jr_average_list, color='Red', marker='*')
plt.ylabel("Average")
plt.xlabel("Matches")
plt.title("Overall Average")
plt.legend(('Virat Kohli', 'Steven Smith','Kane Williamson','Joe Root'))
#plt.savefig('Greatest Odi1.png', dpi=300, bbox_inches='tight')
plt.show()
```



In [ ]:

In [106]:
```python
#Virat Kohli innings wise
vk_ing12=vk[vk['Runs']<=183]
#vk_ing12
```

In [107]:
```python
k=list(vk_ing12['Innings'])
print(k)
print(len(k))
```

```
[1.0, 2.0, 1.0, 1.0, 2.0, 1.0, 2.0, 2.0, 2.0, 2.0, [illegible]
239
```

In [108]:
```python
vk_ing2_dis=[]
vk_ing2_runs=[]
vk_ing1_dis=[]
vk_ing1_runs=[]
for i in range(len(k)):
    if k[i]==2:
        vk_ing2_dis.append(vk_without_nan_dis[i])
        vk_ing2_runs.append(vk_without_nan_runs[i])
    else:
        vk_ing1_dis.append(vk_without_nan_dis[i])
        vk_ing1_runs.append(vk_without_nan_runs[i])
print(vk_ing1_runs)
print(vk_ing1_dis)
```

```
[12, 25, 54, 2, 27, 54, 9, 2, 31, 0, 18, 68, 10, 28, 8, 105, 22, 100, 8, 1, 59, 9, 94, 55, 9, 7, 16, [illegible]
[4, 5, 3, 2, 3, 4, 1, 1, 1, 5, 1, 3, 1, 1, 3, 1, [illegible]
```

In [109]:
```python
count=0
vk_ing2_out_counter=[]
for i in range(len(vk_ing2_dis)):
    if vk_ing2_dis[i]==2:
        count-=1
    count+=1
    vk_ing2_out_counter.append(count)
print(len(vk_ing2_out_counter))
print(vk_ing2_out_counter)

#1st Innings out counter
count=0
vk_ing1_out_counter=[]
for i in range(len(vk_ing1_dis)):
    if vk_ing1_dis[i]==2:
        count-=1
    count+=1
    vk_ing1_out_counter.append(count)
print(len(vk_ing1_out_counter))
print(vk_ing1_out_counter)
```

```
134
[1, 2, 3, 3, 4, 5, 6, 7, 7, 7, 8, 9, 10, 11, 12, 13, 14, 15, 15, 16, 17, 18, 19, 19, 20, 21, 22, 23,
24, 25, 26, 27, 28, 28, 29, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 39, 40, 41, 41, 42, 43, 43, 4
4, 45, 45, 45, 46, 47, 48, 48, 49, 49, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 59, 60, 61, 62, 6
3, 63, 63, 64, 64, 65, 66, 67, 68, 69, 70, 71, 72, 72, 73, 73, 74, 75, 76, 76, 76, 77, 78, 78, 78, 7
9, 80, 80, 81, 82, 83, 84, 85, 85, 85, 86, 87, 88, 89, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9
9, 100, 100, 101, 102, 103]
105
[1, 2, 3, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 2
6, 27, 28, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 5
0, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 70, 71, 71, 7
2, 73, 74, 75, 76, 77, 78, 78, 79, 80, 81, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 9
5, 96, 97]
```

In [110]:
```python
vk_ing2_run_counter=[]
for i in range(len(vk_ing2_runs)):
    if i==0:
        vk_ing2_run_counter.append(vk_ing2_runs[0])
    else:
        vk_ing2_run_counter.append(vk_ing2_runs[i]+vk_ing2_run_counter[i-1])
print(vk_ing2_run_counter)
print(len(vk_ing2_run_counter))


#1st Innings run counter
vk_ing1_run_counter=[]
for i in range(len(vk_ing1_runs)):
    if i==0:
        vk_ing1_run_counter.append(vk_ing1_runs[0])
    else:
        vk_ing1_run_counter.append(vk_ing1_runs[i]+vk_ing1_run_counter[i-1])
print(vk_ing1_run_counter)
print(len(vk_ing1_run_counter))
```

```
[37, 68, 84, 163, 193, 203, 310, 401, 472, 574, 631, 713, 724, 742, 742, 779, 897, 961, 1024, 1024, 1
026, 1080, 1108, 1195, 1197, 1231, 1243, 1267, 1302, 1304, 1385, 1385, 1407, 1519, 1554, 1640, 1643,
1760, 1780, 1811, 1888, 1906, 1921, 1933, 1999, 2020, 2153, 2336, 2374, 2502, 2508, 2523, 2600, 2626,
2648, 2670, 2728, 2730, 2732, 2847, 2915, 2976, 3076, 3191, 3277, 3296, 3327, 3327, 3450, 3528, 3534,
3616, 3752, 3792, 3793, 3806, 3808, 3857, 3910, 4049, 4082, 4115, 4159, 4197, 4198, 4199, 4210, 4287,
4294, 4400, 4408, 4493, 4502, 4656, 4701, 4823, 4878, 4954, 5050, 5055, 5058, 5169, 5251, 5255, 5258,
5368, 5396, 5417, 5456, 5485, 5597, 5643, 5772, 5847, 5892, 6032, 6139, 6172, 6175, 6279, 6325, 6370,
6430, 6474, 6597, 6617, 6635, 6701, 6735, 6736, 6850, 6935, 7024, 7039]
134
[12, 37, 91, 93, 120, 174, 183, 185, 216, 216, 234, 302, 312, 340, 348, 453, 475, 575, 583, 584, 643,
652, 746, 801, 810, 817, 833, 940, 977, 977, 1000, 1080, 1188, 1254, 1360, 1361, 1384, 1384, 1391, 14
28, 1428, 1459, 1502, 1513, 1615, 1646, 1660, 1728, 1728, 1827, 1829, 1877, 1882, 1882, 1944, 2071, 2
093, 2159, 2168, 2172, 2175, 2183, 2290, 2336, 2339, 2362, 2387, 2399, 2537, 2628, 2687, 2804, 2869,
2877, 2958, 2958, 2990, 3077, 3088, 3219, 3219, 3311, 3432, 3545, 3705, 3780, 3816, 3887, 4044, 4060,
4103, 4219, 4226, 4308, 4385, 4452, 4524, 4550, 4670, 4674, 4674, 4690, 4768, 4819, 4828]
105
```

In [111]:
```python
vk_ing2_average_list=[]
for i in range(len(vk_ing2_run_counter)):
    vk_ing2_average_list.append(round((vk_ing2_run_counter[i]/vk_ing2_out_counter[i]),2))
print(vk_ing2_average_list)
print(len(vk_ing2_average_list))
```

```
[37.0, 34.0, 28.0, 54.33, 48.25, 40.6, 51.67, 57.29, 67.43, 82.0, 78.88, 79.22, 72.4, 67.45, 61.83, 5
9.92, 64.07, 64.07, 68.27, 64.0, 60.35, 60.0, 58.32, 62.89, 59.85, 58.62, 56.5, 55.09, 54.25, 52.16,
53.27, 51.3, 50.25, 54.25, 53.59, 56.55, 54.77, 56.77, 55.62, 54.88, 55.53, 54.46, 53.36, 52.24, 52.6
1, 51.79, 55.21, 58.4, 57.9, 61.02, 59.71, 58.67, 60.47, 59.68, 58.84, 59.33, 60.62, 59.35, 58.13, 5
9.31, 60.73, 60.73, 62.78, 65.12, 65.54, 64.63, 63.98, 62.77, 63.89, 64.15, 63.11, 63.44, 64.69, 64.2
7, 64.29, 63.43, 62.43, 62.21, 62.06, 64.27, 64.79, 64.3, 64.98, 64.57, 63.61, 62.67, 61.91, 62.13, 6
1.34, 61.97, 61.22, 62.4, 61.67, 63.78, 63.53, 64.31, 64.18, 65.18, 66.45, 65.65, 64.85, 66.27, 67.3
2, 66.52, 65.72, 67.1, 66.62, 66.06, 65.73, 65.3, 65.85, 66.39, 67.91, 67.99, 67.72, 68.55, 68.98, 6
9.35, 68.61, 69.0, 68.75, 68.49, 68.4, 68.15, 68.72, 68.22, 67.7, 67.69, 68.03, 67.36, 68.5, 68.66, 6
8.86, 68.34]
134
```

In [112]:
```python
vk_ing1_average_list=[]
for i in range(len(vk_ing1_run_counter)):
    vk_ing1_average_list.append(round((vk_ing1_run_counter[i]/vk_ing1_out_counter[i]),2))
print(vk_ing1_average_list)
print(len(vk_ing1_average_list))
```

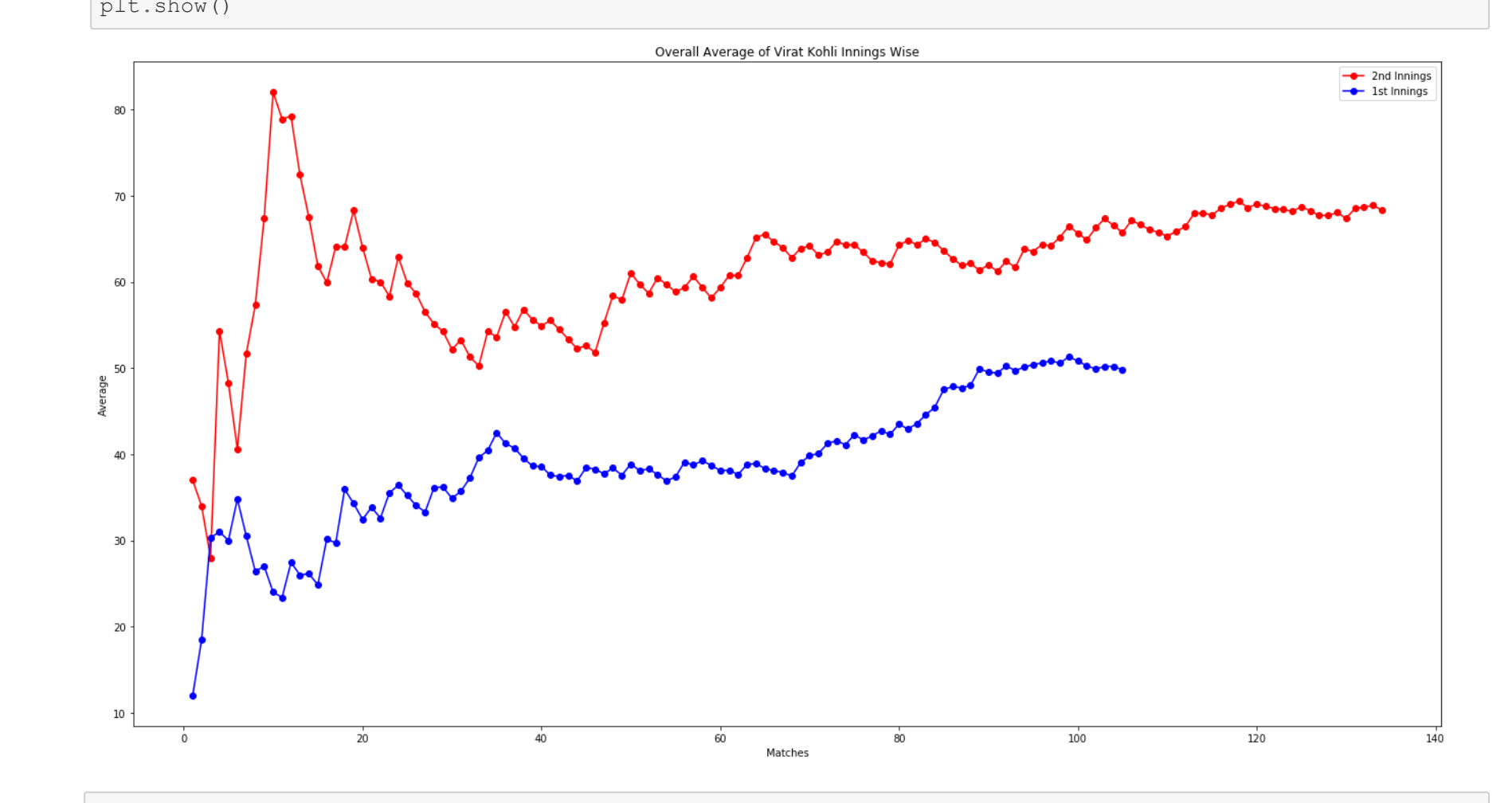
```
[12.0, 18.5, 30.33, 31.0, 30.0, 34.8, 30.5, 26.43, 27.0, 24.0, 23.4, 27.45, 26.0, 26.15, 24.86, 30.2,
29.69, 35.94, 34.29, 32.44, 33.84, 32.6, 35.52, 36.41, 35.22, 34.04, 33.32, 36.15, 36.19, 34.89, 35.7
1, 37.24, 39.6, 40.45, 42.5, 41.24, 40.71, 39.54, 38.64, 38.59, 37.58, 37.41, 37.55, 36.9, 38.45, 38.
28, 37.73, 38.4, 37.57, 38.87, 38.1, 38.31, 37.64, 36.9, 37.38, 39.08, 38.76, 39.25, 38.71, 38.11, 3
8.16, 37.64, 38.81, 38.93, 38.34, 38.1, 37.89, 37.48, 39.03, 39.82, 40.1, 41.24, 41.58, 41.1, 42.26,
41.66, 42.11, 42.74, 42.3, 43.5, 42.92, 43.57, 44.57, 45.45, 47.5, 47.85, 47.7, 47.99, 49.93, 49.51,
49.43, 50.23, 49.72, 50.09, 50.4, 50.59, 50.83, 50.56, 51.32, 50.8, 50.26, 49.89, 50.19, 50.2, 49.77]
105
```

In [113]:
```python
rr=[]
for i in range(134):
    rr.append(i+1)
print(rr)


tt=[]
for i in range(105):
    tt.append(i+1)
print(tt)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 1
23, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105]
```

In [114]:
```python
plt.figure(figsize=(23,12))
plt.plot(rr,vk_ing2_average_list, color='red', marker='o')
plt.plot(tt,vk_ing1_average_list, color='blue', marker='o')
plt.ylabel("Average")
plt.xlabel("Matches")
plt.title("Overall Average of Virat Kohli Innings Wise")
plt.legend(('2nd Innings', '1st Innings'))
plt.show()
```



In [ ]:

In [115]:
```python
#Rohit Sharma innings wise
rgs_ing12=rgs[rgs['Runs']<=264]
rgs_ing12
```

Out[115]:

	Runs	Balls Faced	4s	6s	Strike Rate	Position	Dismissal	Fielder	Bowler	Bowler Type	Espn Commentary	Cricbuzz Commentary	ICC Commentary	Inn
1	8.0	9.0	0.0	0.0	88.88	7.0	1.0	NaN	JH Kallis	Pacer	NaN	NaN	NaN	
2	1.0	4.0	0.0	0.0	25.00	7.0	1.0	B Lee	JR Hopes	NaN	NaN	NaN	NaN	
3	52.0	61.0	3.0	1.0	85.24	5.0	1.0	Yasir Hameed	Shoaib Malik	Spin	NaN	NaN	NaN	
4	29.0	43.0	5.0	0.0	67.44	4.0	1.0	AC Gilchrist	B Lee	Pacer	NaN	NaN	NaN	
5	0.0	2.0	0.0	0.0	0.00	5.0	1.0	KC Sangakkara	M Muralitharan	Spin	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
219	159.0	138.0	17.0	5.0	115.21	1.0	1.0	SD Hope	SS Cottrell	Pacer	NaN	NaN	NaN	
220	63.0	63.0	8.0	1.0	100.00	1.0	1.0	SD Hope	JO Holder	Pacer	NaN	NaN	NaN	
221	10.0	15.0	2.0	0.0	66.66	1.0	1.0	DA Warner	MA Starc	Pacer	NaN	NaN	NaN	
222	42.0	44.0	6.0	0.0	95.45	1.0	4.0	NaN	A Zampa	Spin	NaN	NaN	NaN	
223	119.0	128.0	8.0	6.0	92.96	1.0	1.0	MA Starc	A Zampa	Spin	NaN	NaN	NaN	

217 rows × 22 columns

In [116]:
```python
krgs=list(rgs_ing12['Innings'])
#print(krgs)
print(len(krgs))
```

```
217
```

In [117]:
```python
rgs_ing2_dis=[]
rgs_ing2_runs=[]
rgs_ing1_dis=[]
rgs_ing1_runs=[]
for i in range(len(krgs)):
    if krgs[i]==2:
        rgs_ing2_dis.append(rgs_without_nan_dis[i])
        rgs_ing2_runs.append(rgs_without_nan_runs[i])
    else:
        rgs_ing1_dis.append(rgs_without_nan_dis[i])
        rgs_ing1_runs.append(rgs_without_nan_runs[i])
print(rgs_ing1_runs)
print(rgs_ing1_dis)
#print(len(rgs_ing1_dis))
```

```
[8, 29, 0, 70, 2, 9, 11, 58, 19, 32, 18, 11, 3, 4, 43, 4, 0, 114, 13, 32, 69, 41, 11, 9, 57, 0, 27, 2
1, 4, 5, 0, 4, 4, 4, 65, 9, 60, 46, 48, 1, 11, 209, 12, 79, 13, 56, 52, 264, 138, 15, 0, 137, 0, 29,
3, 21, 171, 124, 6, 70, 91, 78, 104, 28, 7, 20, 147, 2, 208, 0, 5, 115, 2, 23, 4, 162, 87, 7, 2, 0, 9
5, 57, 140, 1, 18, 104, 18, 36, 159, 10, 42]
[3, 1, 1, 2, 1, 3, 5, 1, 1, 1, 1, 2, 1, 2, 2, 1, 1, 1, 5, 3, 5, 1, 4, 4, 3, 2, 3, 3, 5, 3, 3, 3, 1,
1, 1, 3, 1, 1, 2, 1, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 5, 3, 1, 1, 3, 1, 2, 5, 1, 1, 5, 1, 1, 1, 1, 3, 1,
1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 4]
```

In [118]:
```python
count=0
rgs_ing2_out_counter=[]
for i in range(len(rgs_ing2_dis)):
    if rgs_ing2_dis[i]==2:
        count-=1
    count+=1
    rgs_ing2_out_counter.append(count)
print(len(rgs_ing2_out_counter))
#print(rgs_ing2_out_counter)

#1st Innings out counter
count=0
rgs_ing1_out_counter=[]
for i in range(len(rgs_ing1_dis)):
    if rgs_ing1_dis[i]==2:
        count-=1
    count+=1
    rgs_ing1_out_counter.append(count)
print(len(rgs_ing1_out_counter))
#print(rgs_ing1_out_counter)
```

```
126
91
```

In [119]:
```python
rgs_ing2_run_counter=[]
for i in range(len(rgs_ing2_runs)):
    if i==0:
        rgs_ing2_run_counter.append(rgs_ing2_runs[0])
    else:
        rgs_ing2_run_counter.append(rgs_ing2_runs[i]+rgs_ing2_run_counter[i-1])
#print(rgs_ing2_run_counter)
print(len(rgs_ing2_run_counter))


#1st Innings run counter
rgs_ing1_run_counter=[]
for i in range(len(rgs_ing1_runs)):
    if i==0:
        rgs_ing1_run_counter.append(rgs_ing1_runs[0])
    else:
        rgs_ing1_run_counter.append(rgs_ing1_runs[i]+rgs_ing1_run_counter[i-1])
#print(rgs_ing1_run_counter)
print(len(rgs_ing1_run_counter))
```

```
126
91
```

In [120]:
```python
rgs_ing2_average_list=[]
for i in range(len(rgs_ing2_run_counter)):
    rgs_ing2_average_list.append(round((rgs_ing2_run_counter[i]/rgs_ing2_out_counter[i]),2))
#print(rgs_ing2_average_list)
print(len(rgs_ing2_average_list))
```

```
126
```

In [121]:
```python
rgs_ing1_average_list=[]
for i in range(len(rgs_ing1_run_counter)):
    rgs_ing1_average_list.append(round((rgs_ing1_run_counter[i]/rgs_ing1_out_counter[i]),2))
#print(rgs_ing1_average_list)
print(len(rgs_ing1_average_list))
```

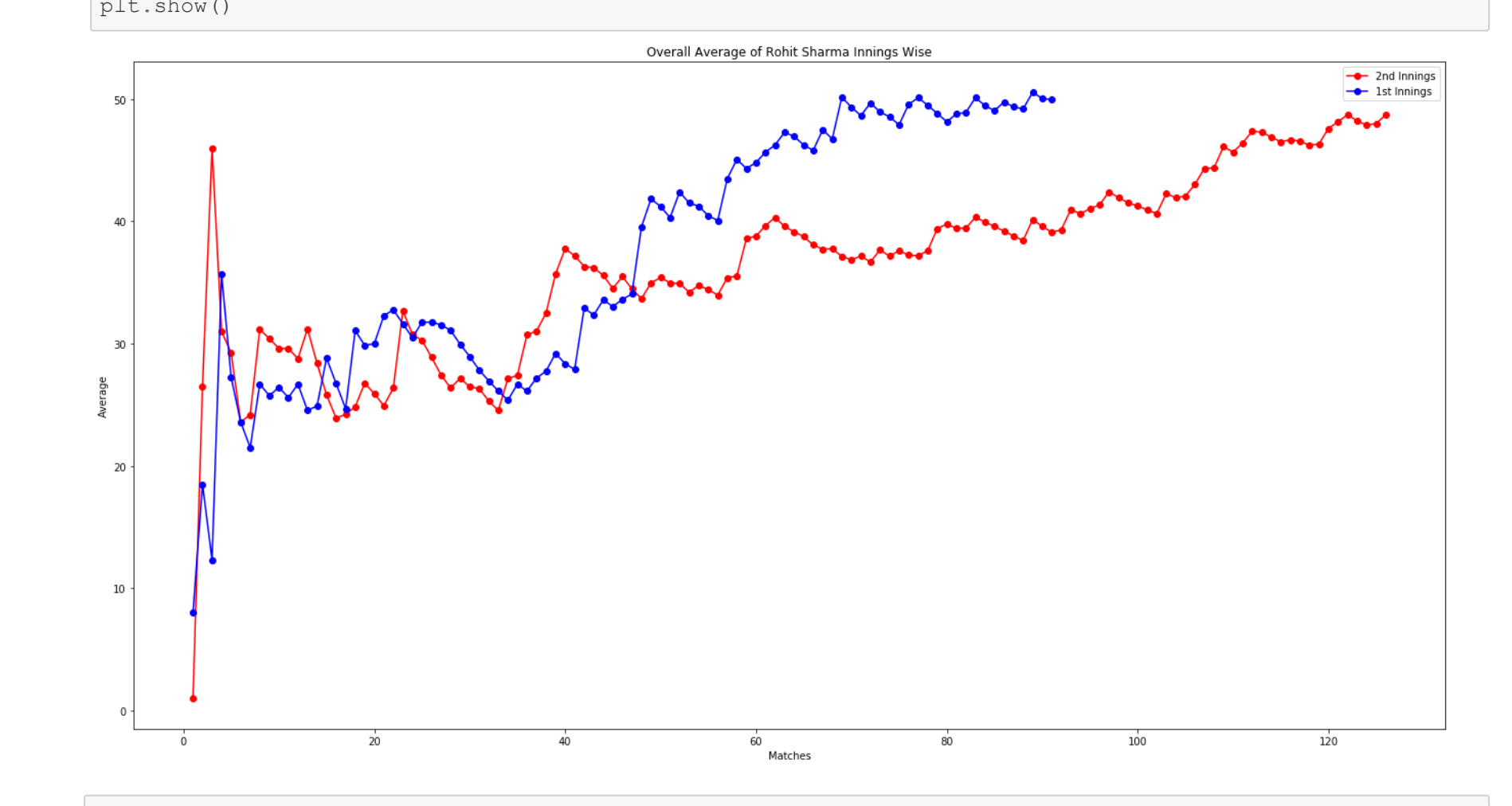
```
91
```

In [122]:
```python
uu=[]
for i in range(126):
    uu.append(i+1)
print(uu)


vv=[]
for i in range(91):
    vv.append(i+1)
print(vv)
```

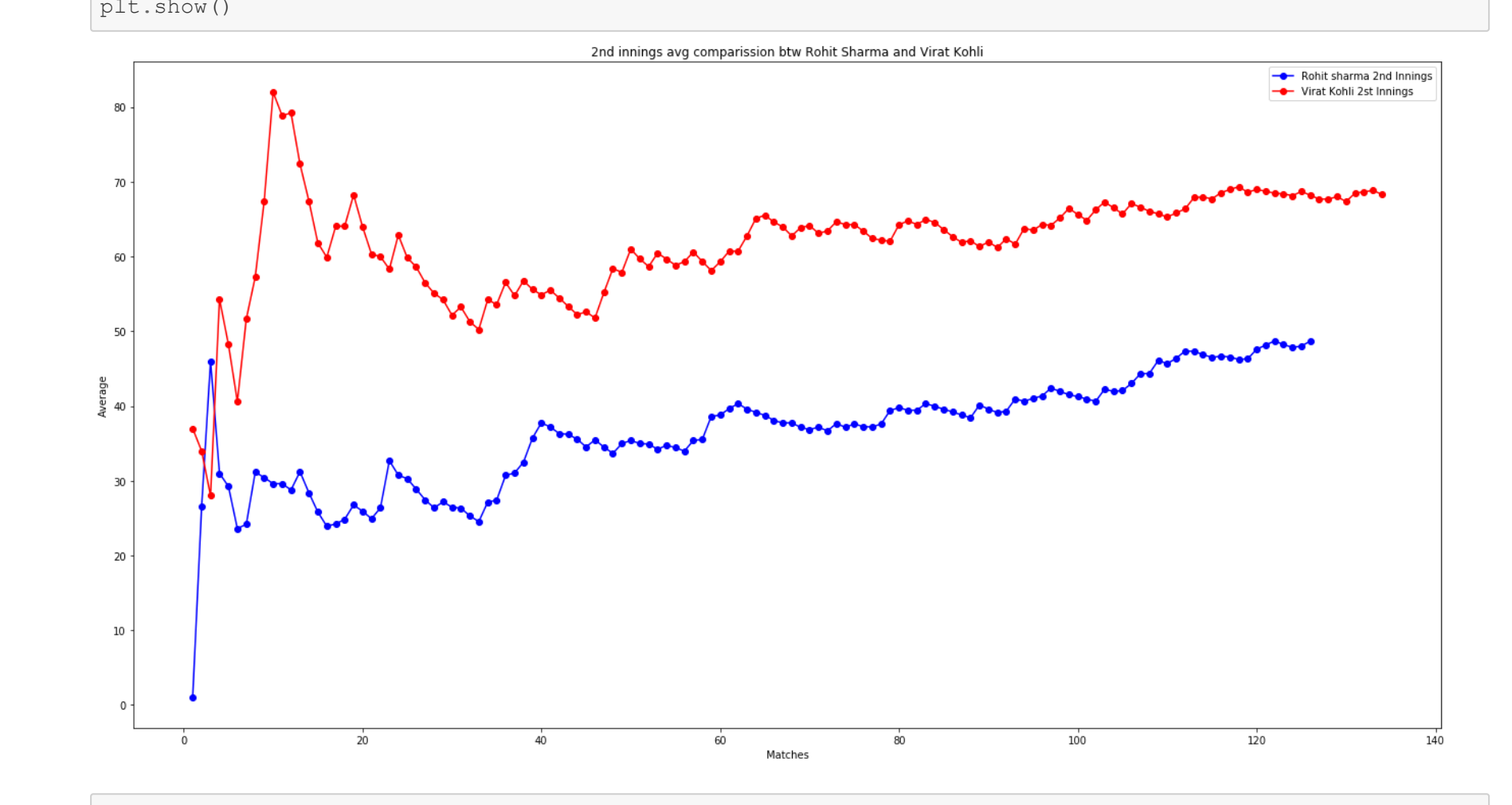
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 1
23, 124, 125, 126]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91]
```

In [123]:
```python
plt.figure(figsize=(23,12))
plt.plot(uu,rgs_ing2_average_list, color='red', marker='o')
plt.plot(vv,rgs_ing1_average_list, color='blue', marker='o')
plt.ylabel("Average")
plt.xlabel("Matches")
plt.title("Overall Average of Rohit Sharma Innings Wise")
plt.legend(('2nd Innings', '1st Innings'))
plt.show()
```

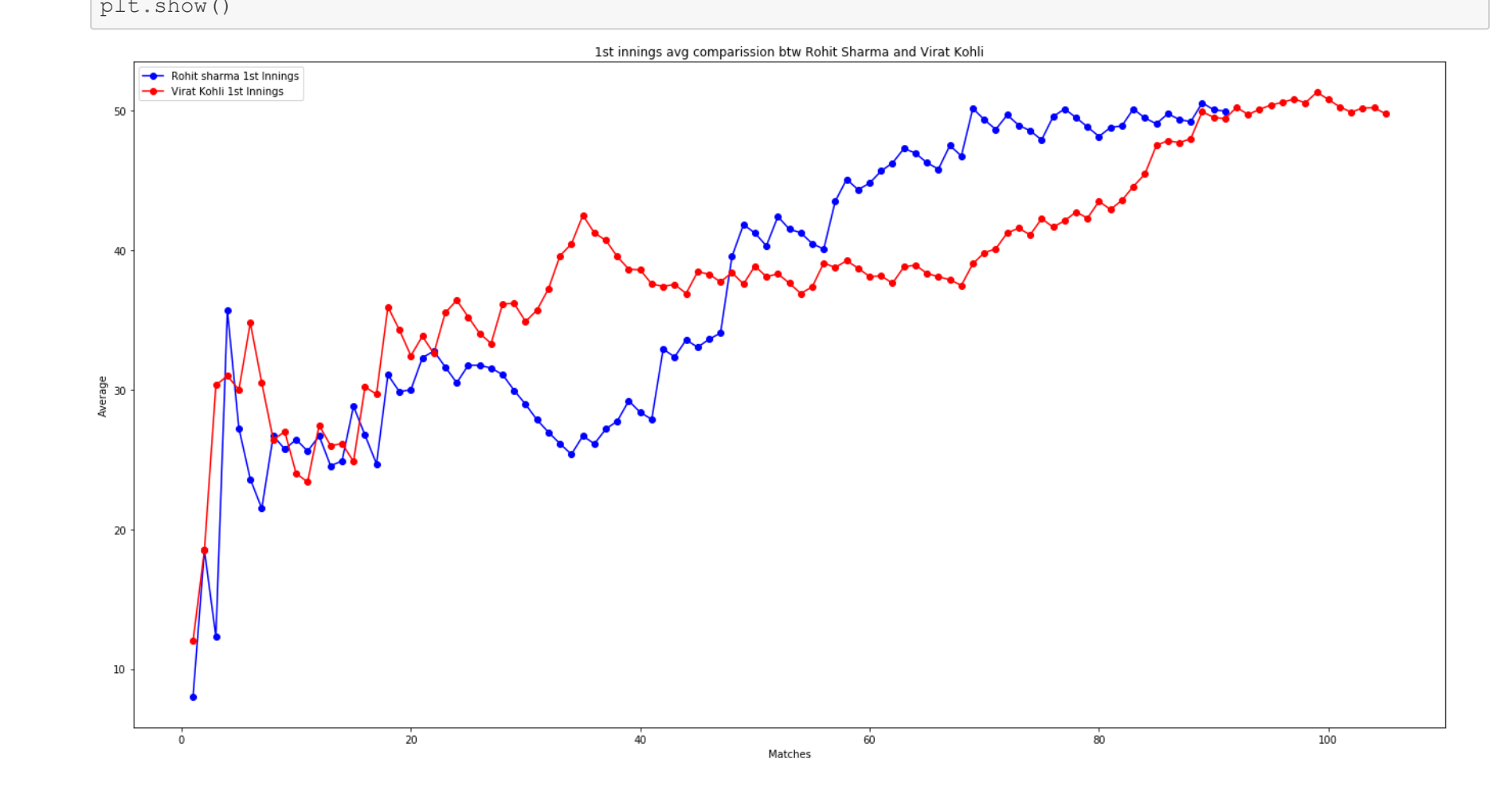


In [ ]:

In [124]:
```python
plt.figure(figsize=(23,12))
plt.plot(uu,rgs_ing2_average_list, color='blue', marker='o')
plt.plot(rr,vk_ing2_average_list, color='red', marker='o')
plt.ylabel("Average")
plt.xlabel("Matches")
plt.title("2nd innings avg comparission btw Rohit Sharma and Virat Kohli")
plt.legend(('Rohit sharma 2nd Innings', 'Virat Kohli 2nd Innings'))
plt.show()
```



In [125]:
```python
plt.figure(figsize=(23,12))
plt.plot(vv,rgs_ing1_average_list, color='blue', marker='o')
plt.plot(tt,vk_ing1_average_list, color='Red', marker='o')
plt.ylabel("Average")
plt.xlabel("Matches")
plt.title("1st innings avg comparission btw Rohit Sharma and Virat Kohli")
plt.legend(('Rohit sharma 1st Innings', 'Virat Kohli 1st Innings'))
plt.show()
```



In [ ]:

In [ ]:

In [126]:
```python
#vk[(vk['As Captain']=='Yes')&(vk['Match Result']=='won')]
```

In [127]:
```python
#vk[(vk['Innings']==2) & (vk['Match Result']=='won')]
```

In [ ]:

In [ ]:

Best batsman at #3 in Fab4

In [128]:
```python
vk['Position'] = vk['Position'].astype(float)
aa=vk[vk['Position']==3]
#aa
```

In [129]:
```python
vk_at3_runs=list(aa['Runs'])
#print(vk_at3_runs)
print(len(vk_at3_runs))
```

```
185
```

In [130]:
```python
vk_at3_dis=list(aa['Dismissal'])
#print(vk_at3_dis)
print(len(vk_at3_dis))
```

```
185
```

In [131]:
```python
count=0
vk_at3_out_counter=[]
for i in range(len(vk_at3_dis)):
    if vk_at3_dis[i]==2:
        count-=1
    count+=1
    vk_at3_out_counter.append(count)
print(len(vk_at3_out_counter))
#print(vk_at3_out_counter)
```

```
185
```

In [132]:
```python
vk_at3_run_counter=[]
for i in range(len(vk_at3_runs)):
    if i==0:
        vk_at3_run_counter.append(vk_at3_runs[0])
    else:
        vk_at3_run_counter.append(vk_at3_runs[i]+vk_at3_run_counter[i-1])
#print(vk_at3_run_counter)
print(len(vk_at3_run_counter))
```

```
185
```

In [133]:
```python
vk_at3_average_list=[]
for i in range(len(vk_at3_run_counter)):
    vk_at3_average_list.append(round((vk_at3_run_counter[i]/vk_at3_out_counter[i]),2))
#print(vk_at3_average_list)
print(len(vk_at3_average_list))
```

```
185
```

In [134]:
```python
ww=[]
for i in range(185):
    ww.append(i+1)
print(ww)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 1
23, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 14
3, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 16
3, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 18
3, 184, 185]
```

In [ ]:

In [135]:
```python
ss['Position'] = ss['Position'].astype(float)
bb=ss[ss['Position']==3]
#bb
```

In [136]:
```python
ss_at3_runs=list(bb['Runs'])
#print(ss_at3_runs)
print(len(ss_at3_runs))
```

```
62
```

In [137]:
```python
ss_at3_dis=list(bb['Dismissal'])
#print(ss_at3_dis)
print(len(ss_at3_dis))
```

```
62
```

In [138]:
```python
count=0
ss_at3_out_counter=[]
for i in range(len(ss_at3_dis)):
    if ss_at3_dis[i]==2:
        count-=1
    count+=1
    ss_at3_out_counter.append(count)
print(len(ss_at3_out_counter))
#print(ss_at3_out_counter)
```

```
62
```

In [139]:
```python
ss_at3_run_counter=[]
for i in range(len(ss_at3_runs)):
    if i==0:
        ss_at3_run_counter.append(ss_at3_runs[0])
    else:
        ss_at3_run_counter.append(ss_at3_runs[i]+ss_at3_run_counter[i-1])
#print(ss_at3_run_counter)
print(len(ss_at3_run_counter))
```

```
62
```

In [140]:
```python
ss_at3_average_list=[]
for i in range(len(ss_at3_run_counter)):
    ss_at3_average_list.append(round((ss_at3_run_counter[i]/ss_at3_out_counter[i]),2))
#print(ss_at3_average_list)
print(len(ss_at3_average_list))
```

```
62
```

In [141]:
```python
xx=[]
for i in range(62):
    xx.append(i+1)
print(xx)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62]
```

In [ ]:

In [142]:
```python
kw['Position'] = kw['Position'].astype(float)
cc=kw[kw['Position']==3]
#cc
```

In [143]:
```python
kw_at3_runs=list(cc['Runs'])
#print(kw_at3_runs)
print(len(kw_at3_runs))
```

```
117
```

In [144]:
```python
kw_at3_dis=list(cc['Dismissal'])
#print(kw_at3_dis)
print(len(kw_at3_dis))
```

```
117
```

In [145]:
```python
count=0
kw_at3_out_counter=[]
for i in range(len(kw_at3_dis)):
    if kw_at3_dis[i]==2:
        count-=1
    count+=1
    kw_at3_out_counter.append(count)
print(len(kw_at3_out_counter))
#print(kw_at3_out_counter)
```

```
117
```

In [146]:
```
kw_at3_run_counter=[]
for i in range(len(kw_at3_runs)):
    if i==0:
        kw_at3_run_counter.append(kw_at3_runs[0])
    else:
        kw_at3_run_counter.append(kw_at3_runs[i]+kw_at3_run_counter[i-1])
#print(kw_at3_run_counter)
print(len(kw_at3_run_counter))
```
```
117
```

In [147]:
```
kw_at3_average_list=[]
for i in range(len(kw_at3_run_counter)):
    kw_at3_average_list.append(round((kw_at3_run_counter[i]/kw_at3_out_counter[i]),2))
#print(kw_at3_average_list)
print(len(kw_at3_average_list))
```
```
117
```

In [148]:
```
yy=[]
for i in range(117):
    yy.append(i+1)
print(yy)
```
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117]
```

In [ ]:

In [149]:
```
jr['Position'] = jr['Position'].astype(float)
dd=jr[jr['Position']==3]
#dd
```

In [150]:
```
jr_at3_runs=list(dd['Runs'])
#print(jr_at3_runs)
print(len(jr_at3_runs))
```
```
78
```

In [151]:
```
jr_at3_dis=list(dd['Dismissal'])
#print(jr_at3_dis)
print(len(jr_at3_dis))
```
```
78
```

In [152]:
```
count=0
jr_at3_out_counter=[]
for i in range(len(jr_at3_dis)):
    if jr_at3_dis[i]==2:
        count-=1
    count+=1
    jr_at3_out_counter.append(count)
print(len(jr_at3_out_counter))
#print(jr_at3_out_counter)
```
```
78
```

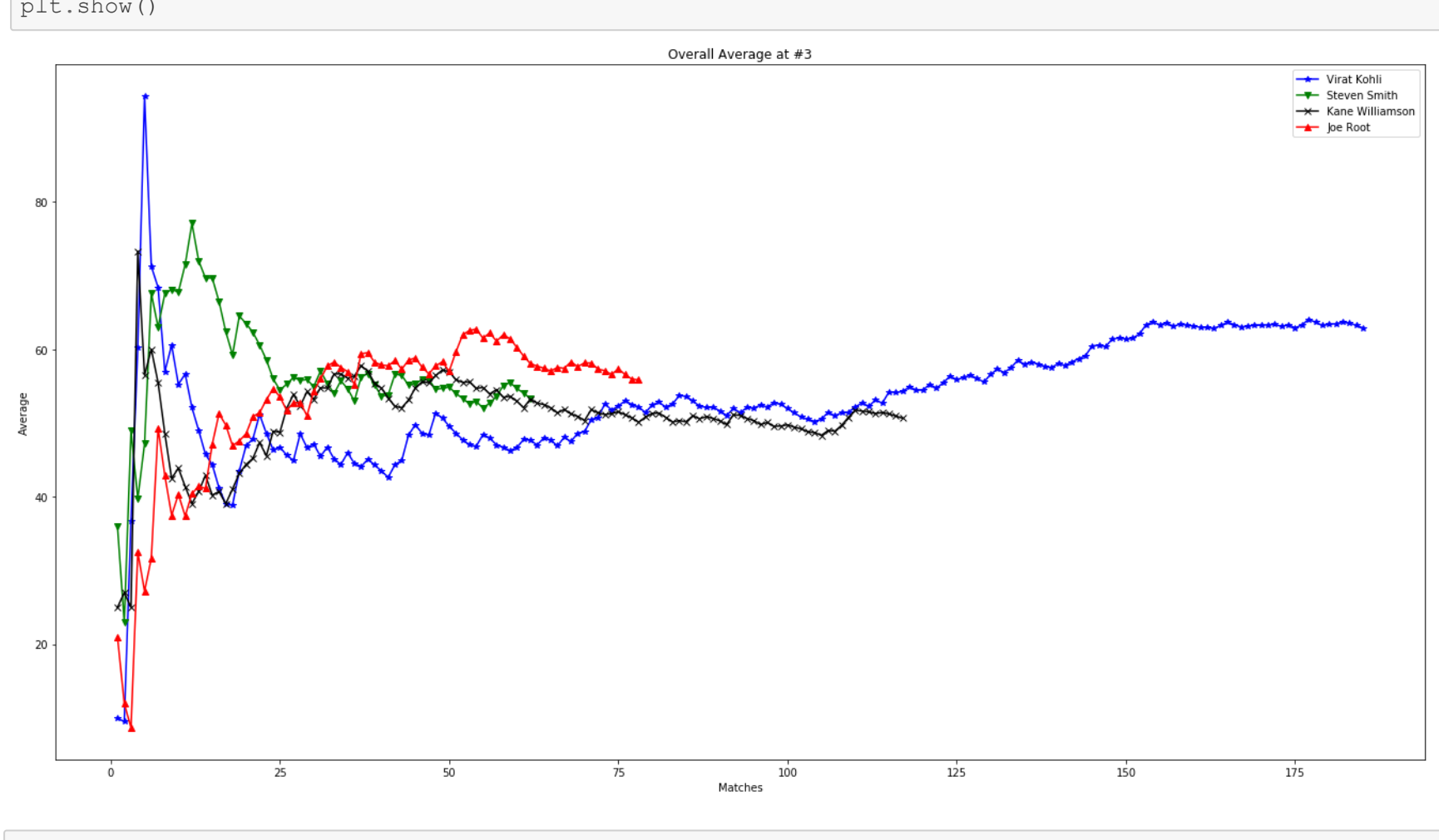
In [153]:
```
jr_at3_run_counter=[]
for i in range(len(jr_at3_runs)):
    if i==0:
        jr_at3_run_counter.append(jr_at3_runs[0])
    else:
        jr_at3_run_counter.append(jr_at3_runs[i]+jr_at3_run_counter[i-1])
#print(jr_at3_run_counter)
print(len(jr_at3_run_counter))
```
```
78
```

In [154]:
```
jr_at3_average_list=[]
for i in range(len(jr_at3_run_counter)):
    jr_at3_average_list.append(round((jr_at3_run_counter[i]/jr_at3_out_counter[i]),2))
#print(jr_at3_average_list)
print(len(jr_at3_average_list))
```
```
78
```

In [155]:
```
zz=[]
for i in range(78):
    zz.append(i+1)
print(zz)
```
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5
3, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 7
8]
```

In [ ]:

In [156]:
```
plt.figure(figsize=(23,12))
plt.plot(ww,vk_at3_average_list, color='Blue', marker='*')
plt.plot(xx,ss_at3_average_list, color='green', marker='v')
plt.plot(yy,kw_at3_average_list, color='black', marker='x')
plt.plot(zz,jr_at3_average_list, color='Red', marker='^')
plt.ylabel("Average")
plt.xlabel("Matches")
plt.title("Overall Average at #3")
plt.legend(('Virat Kohli', 'Steven Smith','Kane Williamson','Joe Root'))
#plt.savefig('Greatest Odi at #3.png', dpi=300, bbox_inches='tight')
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [160]:
```
#tot_dis=odidf['Bowler Type'].value_counts().sum()
#tot_dis
```

In [162]:
```
'''pacer_count=0
for i in odidf['Bowler Type']:
    if i=='Pacer':
        pacer_count+=1
print(pacer_count/tot_dis)'''
```

Out[162]:
```
"pacer_count=0\nfor i in odidf['Bowler Type']:\n    if i=='Pacer':\n        pacer_count+=1\nprint(pacer_count/tot_dis)"
```

In [164]:
```
#k=(pacer_count/tot_dis)
#fig = go.Figure(data=[go.Pie(labels=['Pacer','Spin'], textinfo='label+percent',values=[k,1-k], hole=.
2)])
#fig.update_layout(height=600,title_text="Virat Kohli's Dismissals between Pace and spin in ODI's",fon
t=dict(family='Courier New, monospace', size=18, color='#000000'))
#fig.show()
#plt.title("Virat Kohli's Dismissals between Pace and spin in ODI's",loc='center',fontweight='bold')
#plt.show()
```

In [168]:
```
#played_count=odidf['Dismissal'].value_counts().sum()
#played_count
```

In [169]:
```
'''catch_count=0
bowled_count=0
lbw_count=0
run_out=0
stump_count=0
hitwicket_count=0
not_available=0
notout_count=0
for i in odidf['Dismissal']:
    if i==1.0:
        catch_count+=1
    elif i==3.0:
        bowled_count+=1
    elif i==4.0:
        lbw_count+=1
    elif i==5.0:
        run_out+=1
    elif i==6.0:
        stump_count+=1
    elif i==7.0:
        hitwicket_count+=1
    elif i==2.0:
        notout_count+=1
    else:
        not_available+=1
print(catch_count,notout_count,bowled_count,lbw_count,run_out,stump_count,hitwicket_count,not_availabl
e)
odidf['Dismissal'].value_counts()'''
```

Out[169]:
```
"catch_count=0\nbowled_count=0\nlbw_count=0\nrun_out=0\nstump_count=0\nhitwicket_count=0\nnot_availab
le=0\nnotout_count=0\nfor i in odidf['Dismissal']:\n    if i==1.0:\n        catch_count+=1\n    elif
i==3.0:\n        bowled_count+=1\n    elif i==4.0:\n        lbw_count+=1\n    elif i==5.0:\n        r
un_out+=1\n    elif i==6.0:\n        stump_count+=1\n    elif i==7.0:\n        hitwicket_count+=1\n
  elif i==2.0:\n        notout_count+=1\n    else:\n        not_available+=1\nprint(catch_count,notou
t_count,bowled_count,lbw_count,run_out,stump_count,hitwicket_count,not_available)\nodidf['Dismissa
l'].value_counts()"
```

In [171]:
```
#dismissed_count=played_count-notout_count
#catch_per=catch_count/dismissed_count
#bowled_per=bowled_count/dismissed_count
#lbw_per=lbw_count/dismissed_count
#runout_per=run_out/dismissed_count
#stump_per=stump_count/dismissed_count
#hitwicket_per=hitwicket_count/dismissed_count
#print(catch_per+bowled_per+lbw_per+runout_per+stump_per+hitwicket_per)
#plt.pie([catch_per,bowled_per,lbw_per,runout_per,stump_per,hitwicket_per],radius=3,labels=['Caugh
t','Bowled','Lbw','Run Out','Stumped','Hit Wicket'],autopct='%1.1f%%')
#plt.title("Virat Kohli's Dismissal types in ODI's",loc='center',fontweight='bold')
#plt.show()
```

In [174]:
```
#dismissed_count
```

In [175]:
```
'''import plotly.graph_objects as go
dismissed_count=played_count-notout_count
catch_per=catch_count/dismissed_count
bowled_per=bowled_count/dismissed_count
lbw_per=lbw_count/dismissed_count
runout_per=run_out/dismissed_count
stump_per=stump_count/dismissed_count
hitwicket_per=hitwicket_count/dismissed_count
labels = ['Caught','Bowled','Lbw','Run Out','Stumped','Hit Wicket']
values = [catch_per,bowled_per,lbw_per,runout_per,stump_per,hitwicket_per]
fig = go.Figure(data=[go.Pie(labels=labels, textinfo='label+percent',values=values, hole=.2)])
fig.update_layout(height=600,
    title_text="Percentage of Dismissal types by Virat Kohli in ODI's",
                font=dict(family='Courier New, monospace', size=18, color='#000000'))
fig.show()'''
```

Out[175]:
```
'import plotly.graph_objects as go\ndismissed_count=played_count-notout_count\ncatch_per=catch_count/
dismissed_count\nbowled_per=bowled_count/dismissed_count\nlbw_per=lbw_count/dismissed_count\nrunout_p
er=run_out/dismissed_count\nstump_per=stump_count/dismissed_count\nhitwicket_per=hitwicket_count/dism
issed_count\nlabels = [\'Caught\',\'Bowled\',\'Lbw\',\'Run Out\',\'Stumped\',\'Hit Wicket\']\nvalues
= [catch_per,bowled_per,lbw_per,runout_per,stump_per,hitwicket_per]\nfig = go.Figure(data=[go.Pie(lab
els=labels, textinfo=\'label+percent\',values=values, hole=.2)])\nfig.update_layout(height=600,\n
title_text="Percentage of Dismissal types by Virat Kohli in ODI\'s",\n                font=dict(fam
ily=\'Courier New, monospace\', size=18, color=\'#000000\'))\nfig.show()'
```

In [176]:
```
'''checker50=0
checker100=0
for i in odidf['Runs']:
    if i>=100:
        checker100+=1
        #print(i)
    if i>=50:
        checker50+=1
print(checker100)
print(checker50)'''
```

Out[176]:
```
"checker50=0\nchecker100=0\nfor i in odidf['Runs']:\n    if i>=100:\n        checker100+=1\n        #
print(i)\n    if i>=50:\n        checker50+=1\nprint(checker100)\nprint(checker50)"
```

In [177]:
```
'''k=(checker100/checker50)
fig = go.Figure(data=[go.Pie(labels=['>=100','50> &<100'], textinfo='label+percent',values=[k,1-k], ho
le=.2)])
fig.update_layout(height=600,
    title_text="Virat Kohli's conversion rate from 50's to 100's in ODI's",
                font=dict(family='Courier New, monospace', size=18, color='#000000'))
fig.show()'''
```

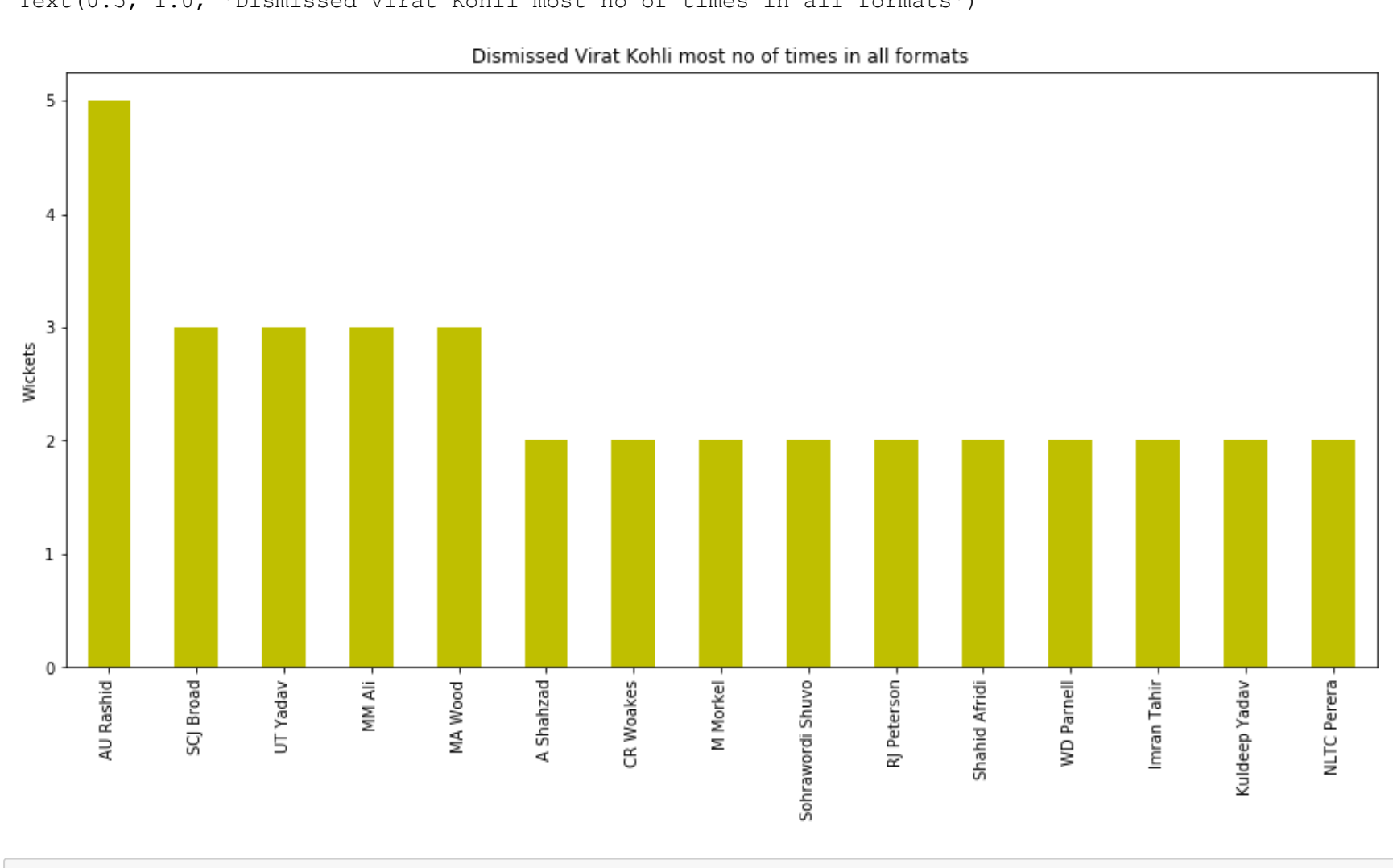
Out[177]:
```
'k=(checker100/checker50)\nfig = go.Figure(data=[go.Pie(labels=[\'>=100\',\'50> &<100\'], textinfo=
\'label+percent\',values=[k,1-k], hole=.2)])\nfig.update_layout(height=600,\n    title_text="Virat Ko
hli\'s conversion rate from 50\'s to 100\'s in ODI\'s",\n                font=dict(family=\'Courier
New, monospace\', size=18, color=\'#000000\'))\nfig.show()'
```

In [ ]:

In [180]:
```
#df['Bowler'].value_counts().head(15)
```

In [179]:
```
ss['Bowler'].value_counts().head(15).plot(kind='bar',color='y',figsize=(15,7))
plt.ylabel("Wickets")
plt.title("Dismissed Virat Kohli most no of times in all formats")
```

Out[179]:
```
Text(0.5, 1.0, 'Dismissed Virat Kohli most no of times in all formats')
```



In [ ]:

In [ ]:

In [ ]: