

# Vamsi Krishna Kannaji

## Data Processing

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
df=pd.read_excel("C:\\Users\\dell\\Desktop\\cricket Batsmen career statistics.xlsx")
```

In [3]:

```
df.head()
```

Out[3]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
0	SR Tendulkar (INDIA)	1989-2013	563	657	55	27786	248*	46.15	78	136	29
1	RT Ponting (AUS/ICC)	1995-2012	514	606	64	24978	257	46.08	64	137	36
2	KC Sangakkara (Asia/ICC/SL)	2000-2015	535	590	58	23653	287	44.46	49	132	26
3	V Kohli (INDIA)	2008-2022	436	490	71	22158	254*	52.88	63	119	30
4	DPMD Jayawardene (Asia/SL)	1998-2015	585	639	53	21712	374	37.05	42	118	43

In [4]:

```
df.shape
```

Out[4]:

```
(150, 11)
```

In [5]:

```
df.columns#all column names
```

Out[5]:

```
Index(['Player', 'Span', 'Mat', 'Inns', 'NO', 'Runs', 'HS', 'Ave', 100, 50, 0], dtype='object')
```

In [6]:

df.head()

Out[6]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
0	SR Tendulkar (INDIA)	1989-2013	563	657	55	27786	248*	46.15	78	136	29
1	RT Ponting (AUS/ICC)	1995-2012	514	606	64	24978	257	46.08	64	137	36
2	KC Sangakkara (Asia/ICC/SL)	2000-2015	535	590	58	23653	287	44.46	49	132	26
3	V Kohli (INDIA)	2008-2022	436	490	71	22158	254*	52.88	63	119	30
4	DPMD Jayawardene (Asia/SL)	1998-2015	585	639	53	21712	374	37.05	42	118	43

In [7]:

df[['Player', 'Inns']][0:20]

Out[7]:

	Player	Inns
0	SR Tendulkar (INDIA)	657
1	RT Ponting (AUS/ICC)	606
2	KC Sangakkara (Asia/ICC/SL)	590
3	V Kohli (INDIA)	490
4	DPMD Jayawardene (Asia/SL)	639
5	JH Kallis (Afr/ICC/SA)	530
6	R Dravid (Asia/ICC/INDIA)	498
7	Inzamam-ul-Haq (Asia/ICC/PAK)	491
8	BC Lara (ICC/WI)	456
9	ST Jayasuriya (Asia/SL)	583
10	AB de Villiers (Afr/SA)	434
11	S Chanderpaul (WI)	476
12	HM Amla (SA/World)	387
13	CH Gayle (ICC/WI)	485
14	SR Waugh (AUS)	473
15	LRPL Taylor (NZ)	448
16	RG Sharma (INDIA)	415
17	TM Dilshan (SL)	476
18	Younis Khan (PAK)	442
19	JE Root (ENG)	357

In [8]:

```
df.iloc[:]
```

Out[8]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
0	SR Tendulkar (INDIA)	1989-2013	563	657	55	27786	248*	46.15	78	136	29
1	RT Ponting (AUS/ICC)	1995-2012	514	606	64	24978	257	46.08	64	137	36
2	KC Sangakkara (Asia/ICC/SL)	2000-2015	535	590	58	23653	287	44.46	49	132	26
3	V Kohli (INDIA)	2008-2022	436	490	71	22158	254*	52.88	63	119	30
4	DPMD Jayawardene (Asia/SL)	1998-2015	585	639	53	21712	374	37.05	42	118	43
...	...	...	...	...	...	...	...	...	...	...	...
145	KL Rahul (INDIA)	2015-2022	140	165	16	5670	199	38.05	12	39	13
146	IJL Trott (ENG)	2007-2015	107	135	12	5654	226	45.96	10	38	12
147	MDKJ Perera (SL)	2013-2021	179	195	8	5633	153*	30.12	8	34	16
148	KJ O'Brien (IRE)	2006-2021	247	241	28	5632	142	26.44	4	23	18
149	DL Vettori (ICC/NZ)	1997-2015	381	332	75	5628	140	21.89	3	22	43

150 rows × 11 columns

In [9]:

df.iloc[0:31,0:6]

Out[9]:

	Player	Span	Mat	Inns	NO	Runs
0	SR Tendulkar (INDIA)	1989-2013	563	657	55	27786
1	RT Ponting (AUS/ICC)	1995-2012	514	606	64	24978
2	KC Sangakkara (Asia/ICC/SL)	2000-2015	535	590	58	23653
3	V Kohli (INDIA)	2008-2022	436	490	71	22158
4	DPMD Jayawardene (Asia/SL)	1998-2015	585	639	53	21712
5	JH Kallis (Afr/ICC/SA)	1996-2014	461	530	76	20852
6	R Dravid (Asia/ICC/INDIA)	1996-2012	431	498	58	18476
7	Inzamam-ul-Haq (Asia/ICC/PAK)	1991-2007	452	491	71	18409
8	BC Lara (ICC/WI)	1990-2007	378	456	36	18355
9	ST Jayasuriya (Asia/SL)	1989-2011	532	583	24	18139
10	AB de Villiers (Afr/SA)	2004-2018	385	434	61	17755
11	S Chanderpaul (WI)	1994-2015	390	476	77	17215
12	HM Amla (SA/World)	2004-2019	315	387	30	16464
13	CH Gayle (ICC/WI)	1999-2021	421	485	27	16236
14	SR Waugh (AUS)	1986-2003	438	473	91	16157
15	LRPL Taylor (NZ)	2006-2022	396	448	70	15744
16	RG Sharma (INDIA)	2007-2022	399	415	53	15689
17	TM Dilshan (SL)	1999-2016	452	476	58	15489
18	Younis Khan (PAK)	2000-2017	373	442	40	15409
19	JE Root (ENG)	2012-2022	278	357	38	15398
20	MS Dhoni (Asia/INDIA)	2004-2019	484	469	132	15207
21	SC Ganguly (Asia/INDIA)	1992-2008	366	420	32	15192
22	DA Warner (AUS)	2009-2022	303	370	22	15015
23	MJ Clarke (AUS)	2003-2015	359	407	63	14880
24	Mohammad Yousuf (Asia/PAK)	1998-2010	352	392	48	14762
25	ME Waugh (AUS)	1988-2002	337	395	33	14633
26	AC Gilchrist (AUS/ICC)	1996-2008	372	399	31	14626
27	GC Smith (ICC/SA)	2002-2014	311	374	24	14531
28	V Sehwag (Asia/ICC/INDIA)	1999-2013	323	375	11	13850
29	KS Williamson (NZ)	2010-2022	293	339	32	13813
30	Tamim Iqbal (BAN/ICC/World)	2007-2022	360	415	14	13623

In [10]:

df.tail()

Out[10]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
145	KL Rahul (INDIA)	2015-2022	140	165	16	5670	199	38.05	12	39	13
146	IJL Trott (ENG)	2007-2015	107	135	12	5654	226	45.96	10	38	12
147	MDKJ Perera (SL)	2013-2021	179	195	8	5633	153*	30.12	8	34	16
148	KJ O'Brien (IRE)	2006-2021	247	241	28	5632	142	26.44	4	23	18
149	DL Vettori (ICC/NZ)	1997-2015	381	332	75	5628	140	21.89	3	22	43

In [11]:

df.isnull().count()

Out[11]:

```

Player    150
Span      150
Mat        150
Inns      150
NO         150
Runs       150
HS         150
Ave        150
100        150
50         150
0          150
dtype: int64

```

In [12]:

df.describe()

Out[12]:

	Mat	Inns	NO	Runs	Ave	100	50
count	150.000000	150.000000	150.000000	150.000000	150.000000	150.000000	150.000000
mean	279.473333	313.200000	35.040000	10414.940000	37.404333	18.206667	58.200000
std	102.273203	106.045134	22.33091	4359.213619	7.874759	12.822116	24.868783
min	42.000000	65.000000	2.000000	5628.000000	21.890000	2.000000	10.000000
25%	208.750000	238.000000	19.000000	6919.750000	32.927500	9.000000	41.000000
50%	264.500000	294.000000	29.500000	9524.500000	36.830000	15.000000	53.500000
75%	355.000000	380.000000	49.000000	12422.500000	41.837500	25.000000	71.000000
max	585.000000	657.000000	132.000000	27786.000000	97.710000	78.000000	137.000000

In [13]:

```
df['Span'].describe()
```

Out[13]:

```
count          150
unique          111
top      2011-2022
freq              6
Name: Span, dtype: object
```

In [14]:

```
df['Span'].unique()
```

Out[14]:

```
array(['1989-2013', '1995-2012', '2000-2015', '2008-2022', '1998-2015',
      '1996-2014', '1996-2012', '1991-2007', '1990-2007', '1989-2011',
      '2004-2018', '1994-2015', '2004-2019', '1999-2021', '1986-2003',
      '2006-2022', '2007-2022', '1999-2016', '2000-2017', '2012-2022',
      '1992-2008', '2009-2022', '2003-2015', '1998-2010', '1988-2002',
      '1996-2008', '2002-2014', '1999-2013', '2010-2022', '1978-1994',
      '1993-2009', '2005-2022', '1996-2010', '1994-2008', '1985-2000',
      '2002-2016', '1984-2003', '1974-1991', '2006-2018', '1989-2003',
      '1975-1996', '1993-2004', '2003-2021', '2004-2015', '2004-2014',
      '2004-2013', '1996-2018', '1982-2000', '2011-2021', '2015-2022',
      '1984-1996', '1983-1996', '2000-2018', '1992-2003', '1982-1999',
      '1995-2007', '1975-1995', '2000-2013', '2004-2021', '2001-2017',
      '2011-2022', '2001-2019', '2003-2012', '1987-2003', '1998-2012',
      '1992-2006', '2005-2019', '2000-2006', '1986-2001', '1992-2000',
      '2003-2016', '1992-2010', '1989-1999', '1978-1992', '2005-2018',
      '1984-1994', '1994-2004', '1982-1992', '1991-2006', '1982-1995',
      '1971-1987', '1984-1997', '1993-2005', '1986-2004', '1971-1984',
      '1993-2001', '1997-2007', '1990-2000', '1966-1985', '1998-2009'])
```

In [15]:

```
df.head()
```

Out[15]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
0	SR Tendulkar (INDIA)	1989-2013	563	657	55	27786	248*	46.15	78	136	29
1	RT Ponting (AUS/ICC)	1995-2012	514	606	64	24978	257	46.08	64	137	36
2	KC Sangakkara (Asia/ICC/SL)	2000-2015	535	590	58	23653	287	44.46	49	132	26
3	V Kohli (INDIA)	2008-2022	436	490	71	22158	254*	52.88	63	119	30
4	DPMD Jayawardene (Asia/SL)	1998-2015	585	639	53	21712	374	37.05	42	118	43

In [16]:

```
df.isnull()
```

Out[16]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
145	False	False	False	False	False	False	False	False	False	False	False
146	False	False	False	False	False	False	False	False	False	False	False
147	False	False	False	False	False	False	False	False	False	False	False
148	False	False	False	False	False	False	False	False	False	False	False
149	False	False	False	False	False	False	False	False	False	False	False

150 rows × 11 columns

In [17]:

```
df.isnull().count()#count of isnull values
```

Out[17]:

```
Player    150
Span      150
Mat        150
Inns      150
NO         150
Runs      150
HS         150
Ave        150
100        150
50         150
0          150
dtype: int64
```

In [18]:

```
df.Span.head()
```

Out[18]:

```
0    1989-2013
1    1995-2012
2    2000-2015
3    2008-2022
4    1998-2015
Name: Span, dtype: object
```

In [19]:

```
df['Span']
```

Out[19]:

```

0      1989-2013
1      1995-2012
2      2000-2015
3      2008-2022
4      1998-2015

```

```

...
145     2015-2022
146     2007-2015
147     2013-2021
148     2006-2021
149     1997-2015

```

Name: Span, Length: 150, dtype: object

In [20]:

```
df[100]
```

Out[20]:

```

0      78
1      64
2      49
3      63
4      42

```

```

..
145     12
146     10
147      8
148      4
149      3

```

Name: 100, Length: 150, dtype: int64

**Removing Unwanted Symbols in DataFrame**

In [21]:

```
df['HS']=df['HS'].replace('\*', '', regex=True)
```

In [22]:

```
df['Player'] = df['Player'].str.strip('()')
```

In [23]:

```
df['Player'] = df['Player'].str.replace('(', '')
```

C:\Users\dell\AppData\Local\Temp\ipykernel\_1412\211109836.py:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.

```
df['Player'] = df['Player'].str.replace('(', '')
```



In [24]:

```
df['Player'] = df['Player'].str.replace('/', '')
```

In [25]:

```
df['Player'] = df['Player'].str.replace(' ', '')
```

In [26]:

```
df['Player'][0:40]
```

Out[26]:

```
0          SRTendulkar INDIA
1          RTPonting AUSICC
2      KCSangakkara AsiaICCSL
3          VKohli INDIA
4      DPMDJayawardene AsiaSL
5          JHKallis AfrICCSA
6          RDravid AsiaICCINDIA
7      Inzamam-ul-Haq AsiaICCPAK
8          BCLara ICCWI
9      STJayasuriya AsiaSL
10         ABdeVilliers AfrSA
11         SChanderpaul WI
12         HMAmla SAWorld
13         CHGayle ICCWI
14         SRWaugh AUS
15         LRPLTaylor NZ
16         RGSharma INDIA
17         TMDilshan SL
18         YounisKhan PAK
19         JERoot ENG
20         MSDhoni AsiaINDIA
21         SCGanguly AsiaINDIA
22         DAWarner AUS
23         MJClarke AUS
24      MohammadYousuf AsiaPAK
25         MEWaugh AUS
26         ACGilchrist AUSICC
27         GCSmith ICCSA
28         VSehwag AsiaICCINDIA
29         KSWilliamson NZ
30      TamimIqbal BANICCWorld
31         DLHaynes WI
32         MLHayden AUSICC
33      MushfiqurRahim BAN
34         HHGibbs SA
35         SPFleming ICCNZ
36      ShakibAlHasan BAN
37         MJGuptill NZ
38         ARBorder AUS
39      MAzharuddin INDIA
Name: Player, dtype: object
```

In [27]:

```
type(df['Player'])
```

Out[27]:

pandas.core.series.Series

In [28]:

```
df.head()
```

Out[28]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
0	SRTendulkar INDIA	1989-2013	563	657	55	27786	248	46.15	78	136	29
1	RTPonting AUSICC	1995-2012	514	606	64	24978	257	46.08	64	137	36
2	KCSangakkara AsialCCSL	2000-2015	535	590	58	23653	287	44.46	49	132	26
3	VKohli INDIA	2008-2022	436	490	71	22158	254	52.88	63	119	30
4	DPMDJayawardene AsiaSL	1998-2015	585	639	53	21712	374	37.05	42	118	43

In [29]:

```
df.Runs.idxmax()#Displays max runs scored player ID
```

Out[29]:

0

In [30]:

```
df.iloc[0:1,:]#Displays max runs scored player details
```

Out[30]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
0	SRTendulkar INDIA	1989-2013	563	657	55	27786	248	46.15	78	136	29

In [31]:

```
df.Runs.idxmin()#Displays min runs scored player ID
```

Out[31]:

149

In [32]:

```
df.iloc[149:150,:]# Displays min runs scored player details
```

Out[32]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
149	DLVettori ICCNZ	1997-2015	381	332	75	5628	140	21.89	3	22	43

In [33]:

```
df.Mat.idxmax()#Displays max matches played player index
```

Out[33]:

4

In [34]:

```
df.iloc[4:5,:]#Displays max matches palyed player details
```

Out[34]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
4	DPMDJayawardene AsiaSL	1998-2015	585	639	53	21712	374	37.05	42	118	43

In [35]:

```
df.Mat.idxmin()# Displays minimum matches played player index
```

Out[35]:

141

In [36]:

```
df.iloc[141:142,:3] #Displays minimum matches played player details
```

Out[36]:

	Player	Span	Mat
141	DGBradman AUS	1928-1948	42

In [37]:

```
df.iloc[141:142,:]#Displays minimum matches played player details
```

Out[37]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0
141	DGBradman AUS	1928-1948	42	65	6	5765	299	97.71	25	10	6

In [38]:

```
df[['Inns','Runs']].head() #top 5 rows of Innings and Runs
```

Out[38]:

	Inns	Runs
0	657	27786
1	606	24978
2	590	23653
3	490	22158
4	639	21712

In [39]:

```
result= df.Runs//df.Inns#calculating runs scored per Innings played
result
```

Out[39]:

```
0      42
1      41
2      40
3      45
4      33
..
145    34
146    41
147    28
148    23
149    16
Length: 150, dtype: int64
```

In [40]:

```
df['AverageRunsScoredPerInnings']=result
df #inserted new column
```

Out[40]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0	Avera
0	SRTendulkar INDIA	1989-2013	563	657	55	27786	248	46.15	78	136	29	
1	RTPonting AUSICC	1995-2012	514	606	64	24978	257	46.08	64	137	36	
2	KCSangakkara AsiaICCSL	2000-2015	535	590	58	23653	287	44.46	49	132	26	
3	VKohli INDIA	2008-2022	436	490	71	22158	254	52.88	63	119	30	
4	DPMDJayawardene AsiaSL	1998-2015	585	639	53	21712	374	37.05	42	118	43	
...	...	...	...	...	...	...	...	...	...	...	...	
145	KL Rahul INDIA	2015-2022	140	165	16	5670	199	38.05	12	39	13	
146	IJLTrott ENG	2007-2015	107	135	12	5654	226	45.96	10	38	12	
147	MDKJPerera SL	2013-2021	179	195	8	5633	153	30.12	8	34	16	
148	KJO'Brien IRE	2006-2021	247	241	28	5632	142	26.44	4	23	18	
149	DLVettori ICCNZ	1997-2015	381	332	75	5628	140	21.89	3	22	43	

150 rows × 12 columns



In [41]:

```
df['AverageRunsScoredPerMatch']=result
df #inserted new column
```

Out[41]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0	Avera
0	SRTendulkar INDIA	1989-2013	563	657	55	27786	248	46.15	78	136	29	
1	RTPonting AUSICC	1995-2012	514	606	64	24978	257	46.08	64	137	36	
2	KCSangakkara AsiaICCSL	2000-2015	535	590	58	23653	287	44.46	49	132	26	
3	VKohli INDIA	2008-2022	436	490	71	22158	254	52.88	63	119	30	
4	DPMDJayawardene AsiaSL	1998-2015	585	639	53	21712	374	37.05	42	118	43	
...	...	...	...	...	...	...	...	...	...	...	...	
145	KLRahul INDIA	2015-2022	140	165	16	5670	199	38.05	12	39	13	
146	IJLTrott ENG	2007-2015	107	135	12	5654	226	45.96	10	38	12	
147	MDKJPerera SL	2013-2021	179	195	8	5633	153	30.12	8	34	16	
148	KJO'Brien IRE	2006-2021	247	241	28	5632	142	26.44	4	23	18	
149	DLVettori ICCNZ	1997-2015	381	332	75	5628	140	21.89	3	22	43	

150 rows × 13 columns

In [42]:

```
df= df.drop('AverageRunsScoredPerMatch',1)#removing unwanted column
```

C:\Users\dell\AppData\Local\Temp\ipykernel\_1412\2386449952.py:1: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only.

```
df= df.drop('AverageRunsScoredPerMatch',1)#removing unwanted column
```

In [43]:

```
result1= df.Runs//df.Mat#calculatng run scored per match
result1
```

Out[43]:

```
0      49
1      48
2      44
3      50
4      37
..
145    40
146    52
147    31
148    22
149    14
Length: 150, dtype: int64
```

In [44]:

```
df['AverageRunsScoredPerMatch']=result1#inserting new column
```

In [45]:

```
df[['Player', 'AverageRunsScoredPerMatch']].head()#top 5 index players and their runs in p
```

Out[45]:

	Player	AverageRunsScoredPerMatch
0	SRTendulkar INDIA	49
1	RTPonting AUSICC	48
2	KCSangakkara AsialCCSL	44
3	VKohli INDIA	50
4	DPMDJayawardene AsiaSL	37

In [46]:

```
df[['Player', 'AverageRunsScoredPerMatch', 'AverageRunsScoredPerInnings']][0:31]#no.of runs
```

Out[46]:

	Player	AverageRunsScoredPerMatch	AverageRunsScoredPerInnings
0	SRTendulkar INDIA	49	42
1	RTPonting AUSICC	48	41
2	KCSangakkara AsiaCCSL	44	40
3	VKohli INDIA	50	45
4	DPMDJayawardene AsiaSL	37	33
5	JHKallis AfrICCSA	45	39
6	RDravid AsiaCCINDIA	42	37
7	Inzamam-ul-Haq AsiaCCPAK	40	37
8	BCLara ICCWI	48	40
9	STJayasuriya AsiaSL	34	31
10	ABdeVilliers AfrSA	46	40
11	SChanderpaul WI	44	36
12	HMAmla SAWorld	52	42
13	CHGayle ICCWI	38	33
14	SRWaugh AUS	36	34
15	LRPLTaylor NZ	39	35
16	RGSharma INDIA	39	37
17	TMDilshan SL	34	32
18	YounisKhan PAK	41	34
19	JERoot ENG	55	43
20	MSDhoni AsiaINDIA	31	32
21	SCGanguly AsiaINDIA	41	36
22	DAWarner AUS	49	40
23	MJClarke AUS	41	36
24	MohammadYousuf AsiaPAK	41	37
25	MEWaugh AUS	43	37
26	ACGilchrist AUSICC	39	36
27	GCSmith ICCSA	46	38
28	VSehwag AsiaCCINDIA	42	36
29	KSWilliamson NZ	47	40
30	TamimIqbal BANICCWorld	37	32

In [47]:

```
df.nlargest(10,[100])#top 10 maximum 100's scored players in cricket history
```

Out[47]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0	Averag
0	SRTendulkar INDIA	1989-2013	563	657	55	27786	248	46.15	78	136	29	
1	RTPonting AUSICC	1995-2012	514	606	64	24978	257	46.08	64	137	36	
3	VKohli INDIA	2008-2022	436	490	71	22158	254	52.88	63	119	30	
2	KCSangakkara AsiaICCSL	2000-2015	535	590	58	23653	287	44.46	49	132	26	
12	HMAmla SAWorld	2004-2019	315	387	30	16464	311	46.11	47	79	18	
4	DPMDJayawardene AsiaSL	1998-2015	585	639	53	21712	374	37.05	42	118	43	
5	JHKallis AfrICCSA	1996-2014	461	530	76	20852	224	45.92	42	129	27	
8	BCLara ICCWI	1990-2007	378	456	36	18355	226	43.70	41	100	30	
10	ABdeVilliers AfrSA	2004-2018	385	434	61	17755	217	47.60	41	102	19	
16	RGSharma INDIA	2007-2022	399	415	53	15689	264	43.33	41	82	26	



In [48]:

```
df.nlargest(10,[50])#top 10 maximum 50's scored players in cricket history
```

Out[48]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0	Averag
1	RTPonting AUSICC	1995-2012	514	606	64	24978	257	46.08	64	137	36	
0	SRTendulkar INDIA	1989-2013	563	657	55	27786	248	46.15	78	136	29	
2	KCSangakkara AsiaICCSL	2000-2015	535	590	58	23653	287	44.46	49	132	26	
5	JHKallis AfrICCSA	1996-2014	461	530	76	20852	224	45.92	42	129	27	
3	VKohli INDIA	2008-2022	436	490	71	22158	254	52.88	63	119	30	
4	DPMDJayawardene AsiaSL	1998-2015	585	639	53	21712	374	37.05	42	118	43	
6	RDravid AsiaCCINDIA	1996-2012	431	498	58	18476	270	41.99	31	115	20	
7	Inzamam-ul-Haq AsiaCCPAK	1991-2007	452	491	71	18409	329	43.83	29	115	27	
11	SChanderpaul WI	1994-2015	390	476	77	17215	203	43.14	31	104	20	
10	ABdeVilliers AfrSA	2004-2018	385	434	61	17755	217	47.60	41	102	19	

In [49]:

```
df['NO'].max()
```

Out[49]:

132

In [50]:

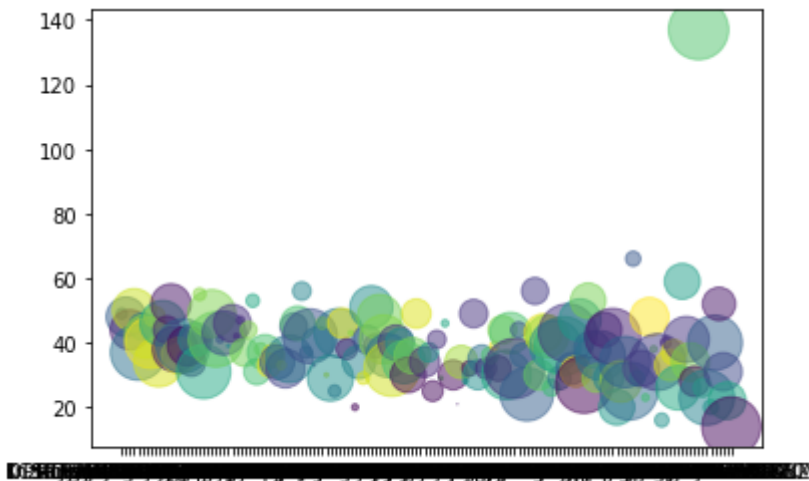
```
df.nlargest(5,['NO'])
```

Out[50]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	100	50	0	AverageRur
20	MSDhoni AsiaINDIA	2004-2019	484	469	132	15207	224	45.12	14	93	18	
132	SMPollock AfrICCSA	1996-2008	376	323	99	6087	130	27.17	2	25	29	
14	SRWaugh AUS	1986-2003	438	473	91	16157	200	42.29	33	77	31	
11	SChanderpaul WI	1994-2015	390	476	77	17215	203	43.14	31	104	20	
47	ADMathews SL	2008-2022	356	384	77	11879	200	38.69	10	71	18	

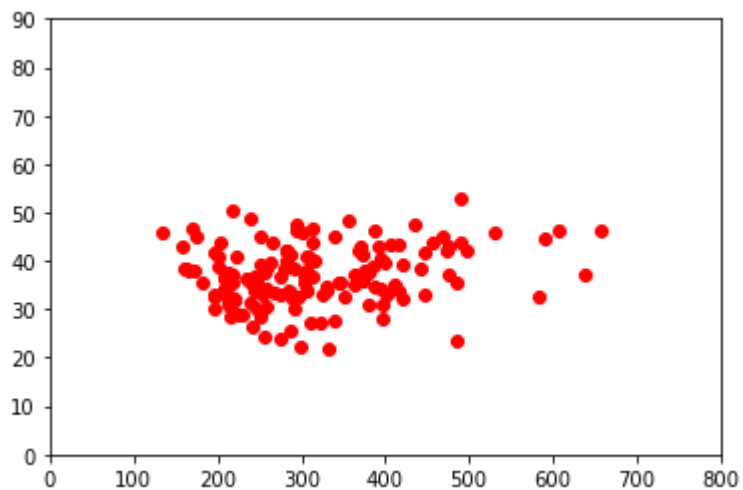
In [51]:

```
N=150
x=df['Player'][0:150]
y=df['AverageRunsScoredPerMatch'][0:150]
colors = np.random.rand(N)
area = (30 * np.random.rand(N))**2 # 0 to 15 point radii
plt.scatter(x,y, s=area, c=colors, alpha=0.5)
plt.show()
```



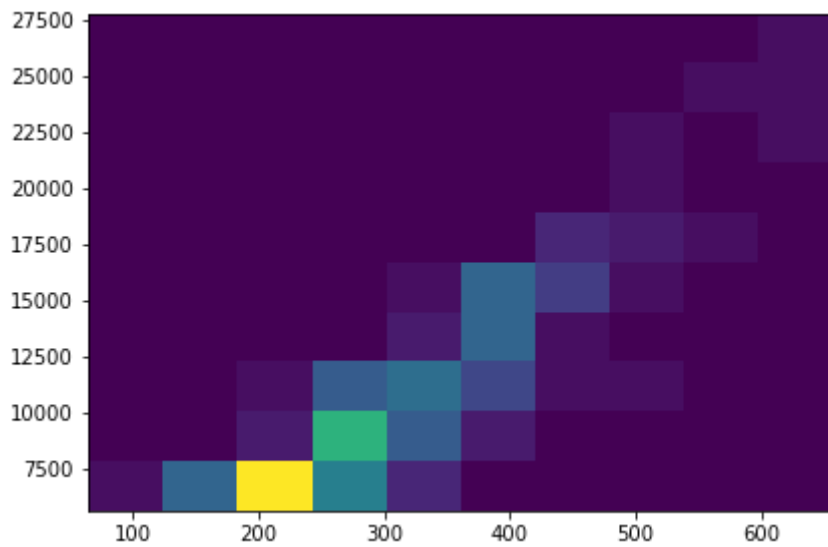
In [52]:

```
plt.plot(df['Inns'], df['Ave'], 'ro')
plt.axis([0,800,0,90])
plt.show()
```



In [53]:

```
fig, ax = plt.subplots(tight_layout=True)
hist = ax.hist2d(df['Inns'], df['Runs'])
```



In [54]:

```

N_points = 100000
n_bins = 20
rng = np.random.default_rng(150)
# Generate two normal distributions
dist1 = rng.standard_normal(N_points)
dist2 = 0.4 * rng.standard_normal(N_points) + 5

fig, axs = plt.subplots(1,2, sharey=True, tight_layout=True)

# We can set the number of bins with the *bins* keyword argument.
axs[0].hist(df['Mat'], bins=n_bins)
axs[1].hist(df['NO'], bins=n_bins)

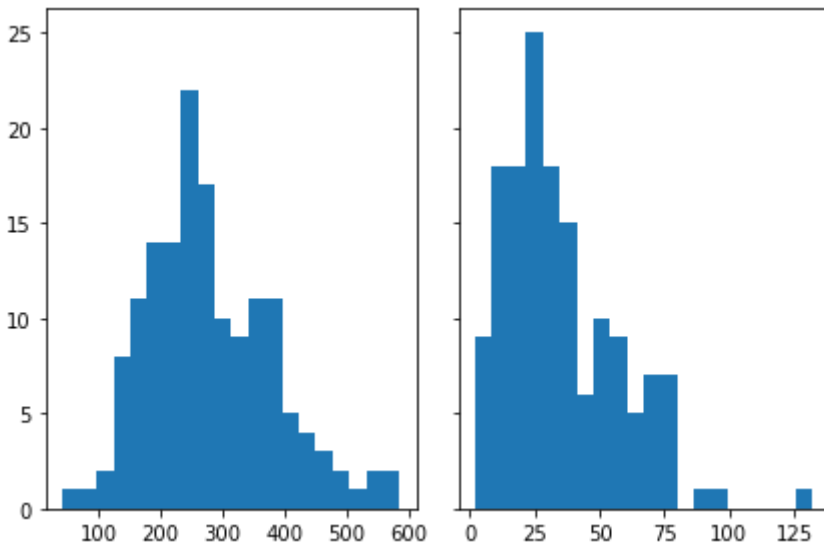
```

Out[54]:

```

(array([ 9., 18., 18., 25., 18., 15.,  6., 10.,  9.,  5.,  7.,  7.,  0.,
        1.,  1.,  0.,  0.,  0.,  0.,  1.]),
 array([ 2. ,  8.5, 15. , 21.5, 28. , 34.5, 41. , 47.5, 54. ,
        60.5, 67. , 73.5, 80. , 86.5, 93. , 99.5, 106. , 112.5,
        119. , 125.5, 132. ]),
 <BarContainer object of 20 artists>)

```

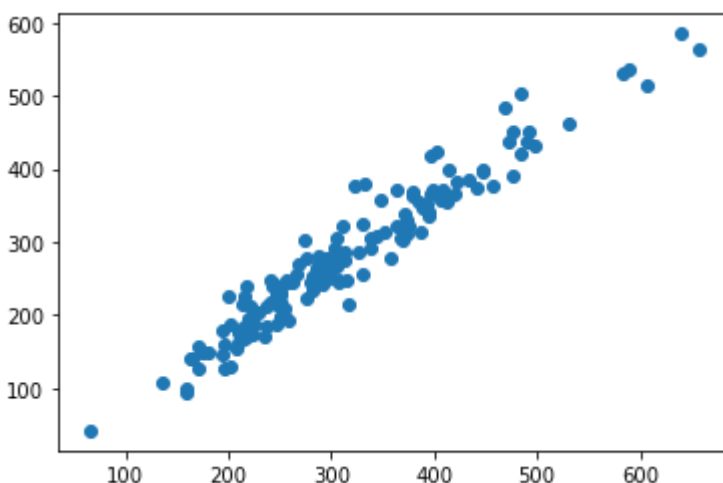


In [55]:

```

plt.scatter(x=df['Inns'], y=df['Mat'])
plt.show()

```

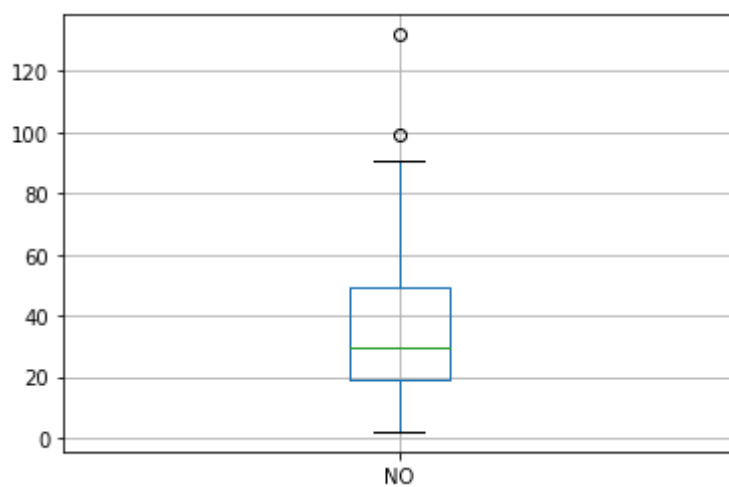


In [56]:

```
df[['NO']].boxplot()
```

Out[56]:

&lt;AxesSubplot:&gt;

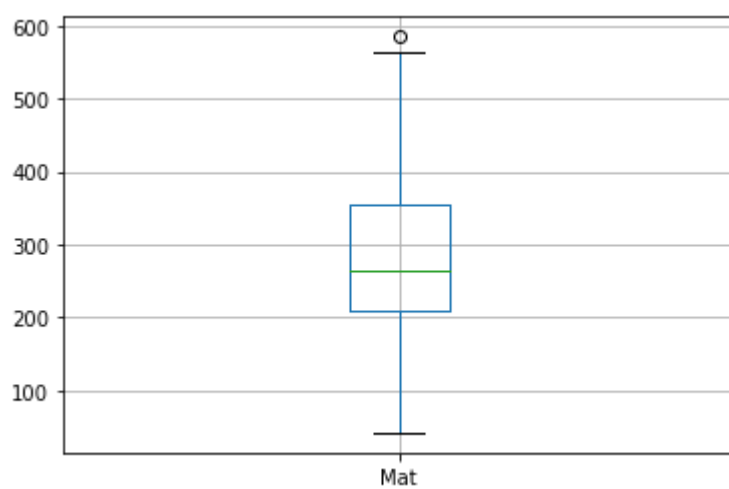


In [57]:

```
df[['Mat']].boxplot()
```

Out[57]:

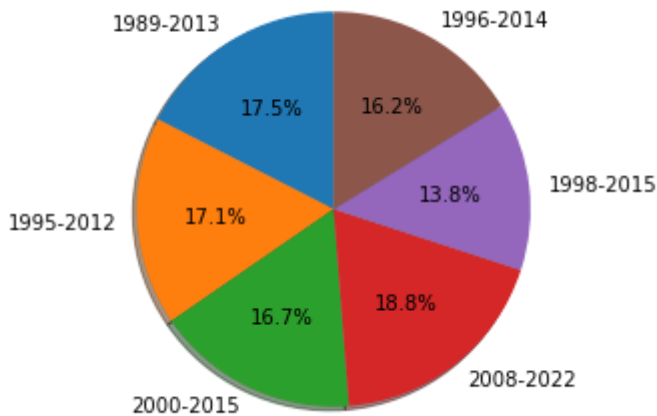
&lt;AxesSubplot:&gt;



In [58]:

```
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = df['Span'][0:6]
fig1, ax1 = plt.subplots()
ax1.pie(df['AverageRunsScoredPerInnings'][0:6], labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.

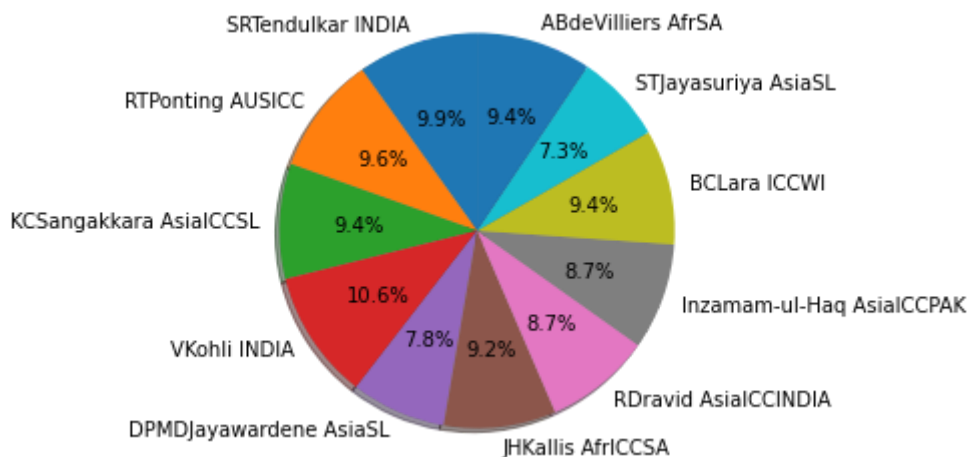
plt.show()
```



In [59]:

```
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = df['Player'][0:11]
fig1, ax1 = plt.subplots()
ax1.pie(df['AverageRunsScoredPerInnings'][0:11], labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.

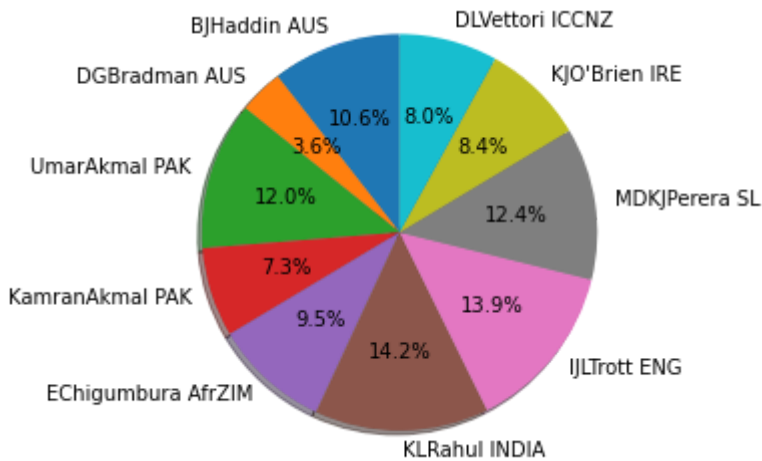
plt.show()
```



In [60]:

```
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = df['Player'][140:150]
fig1, ax1 = plt.subplots()
ax1.pie(df[50][140:150], labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```

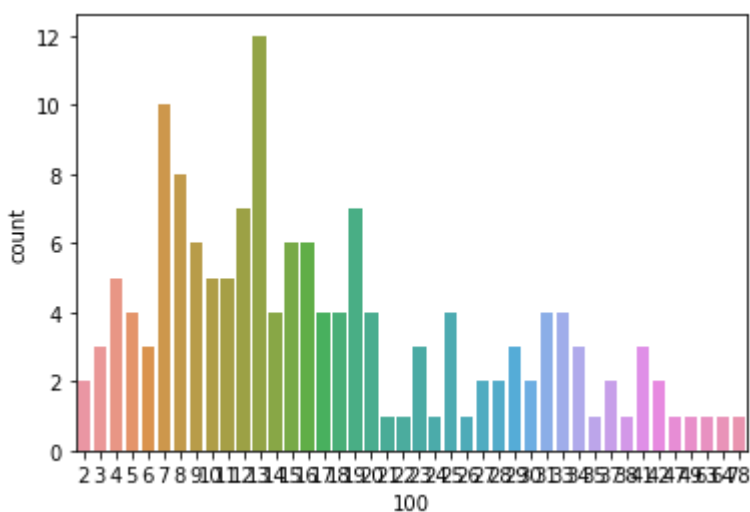


In [61]:

```
sns.countplot(df[100])
plt.show()
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

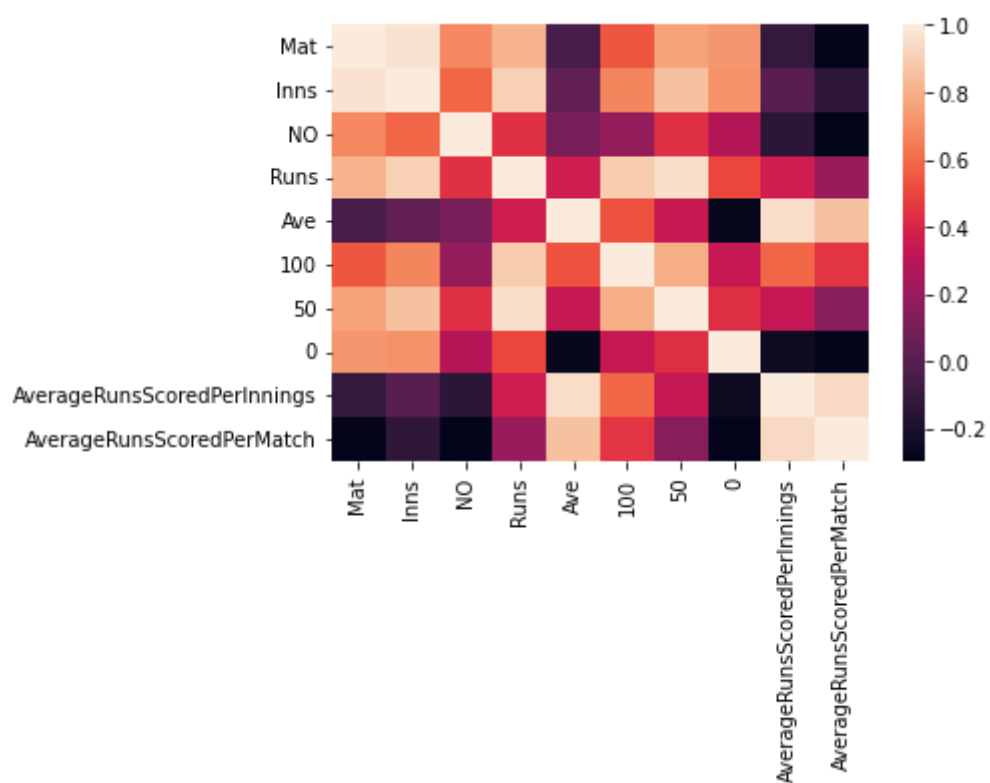


In [62]:

```
sns.heatmap(df.corr())
```

Out[62]:

&lt;AxesSubplot:&gt;



In [ ]: