CAPSTONE PROJECT 2 – IPL GAUGING PLAYERS PERFORMANCE

Q: Mitra decides to form homogeneous subgroups of players, which would better help him to express the nuances of T20 cricket. How would you go about implementing this?

Solution:-

We were given IPL 2019 data to find the gauging player performance.

In order to proceed with the analysis, we have to read the given data;

```
#IPL <- read.csv("clipboard", sep = "\t", header = T)
#View(IPL)
```

If you see the given data, we have null values in the second column. We have to omit that column to proceed further with the analysis

```
#IPL<- (IPL[, -c(2)])
#IPL1<- IPL
```

As we all know, the complete IPL is all about scoring runs to entertain the audience. We will consider the batting performance as a base index to take the players performance. We take the strike rate and batting average as parameters to identify the top and consistent performers.

```
#IPL1$BI <- IPL1$Avg*IPL$SR/100
```

As we have consider Average and Strike Rate to identify the players performance. We will drop these columns for now to proceed with the analysis.

```
#IPL1 <- (IPL1[, -c(5:6)])
```

Lets Scale the final data

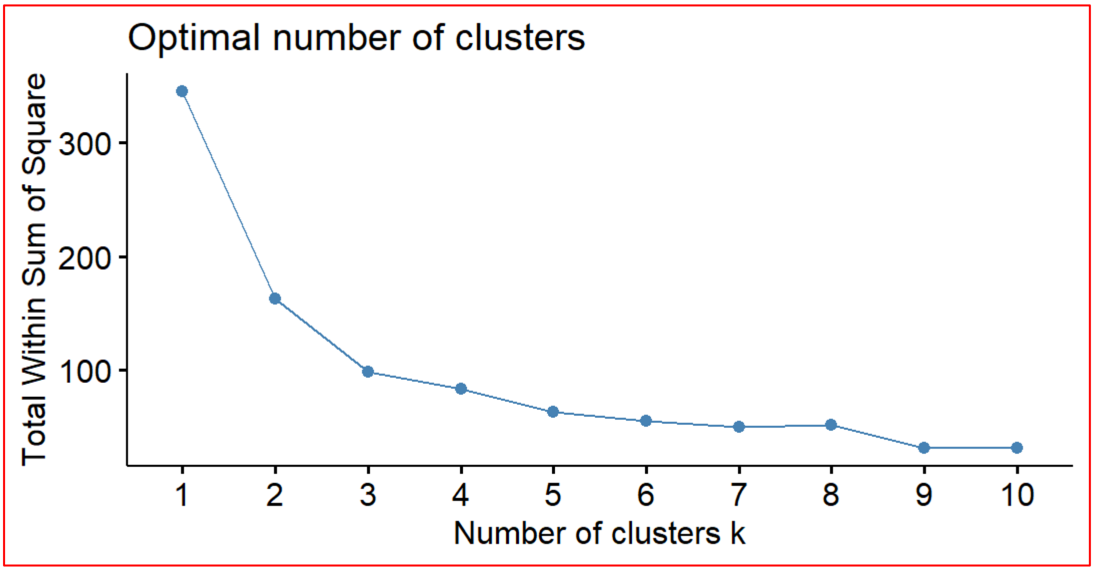
```
#IPLS <- scale(IPL1[, c(4:9)])
```

To identify the best performer from the revised that, we will do elbow test. In order to proceed further we need to install following packaged 1. Factoextra & 2. Ggplot2

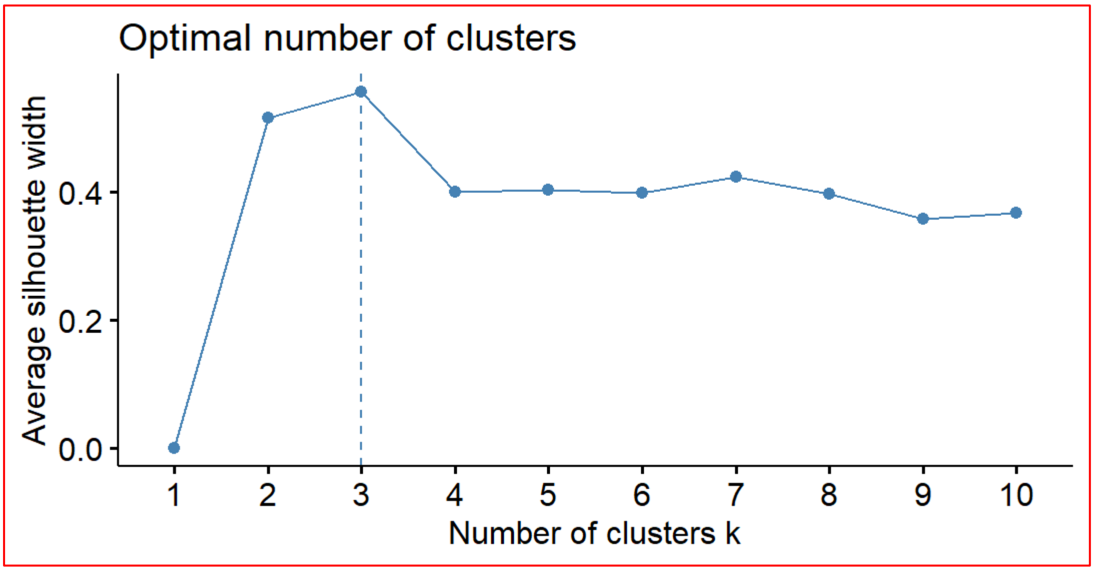
```
#install.packages("factoextra")
#library(factoextra)
```

#install.packages("ggplot2")
#library(ggplot2)

#fviz_nbclust(IPLS[, 1:5], kmeans, method = "wss")



#fviz_nbclust(IPLS[, 1:5], kmeans, method = "silhouette")



By observing the above graphs, we can come to an understanding that both the graphs have significant change at 3 cluster mark. Lets proceed with cluster test by taking the k as 3

#set.seed(1234)
#cluster_1<-kmeans(IPLS,3)
#cluster_1

K-means clustering with 3 clusters of sizes 6, 38, 26

Cluster means:

	Runs	Hundreds	Fifties
1	1.4157758	3.2425739	0.9693617
2	-0.7970102	-0.3039913	-0.6898410
3	0.8381436	-0.3039913	0.7845303

	Fours	Sixes	Salary
1	1.3713348	0.6253612	0.7560000
2	-0.7720593	-0.6145994	-0.2806928
3	0.8119325	0.7539466	0.2357817

Clustering vector:

```
 [1] 3 1 2 2 3 2 2 2 3 3 2 2 1 2 2
[16] 3 2 3 3 2 2 1 3 2 2 2 3 1 2 2
[31] 3 2 3 2 2 3 2 3 3 2 3 3 2 2 2
[46] 2 3 2 3 3 2 1 2 3 2 3 2 2 3 3
[61] 3 2 2 3 3 2 2 1 2 2
```

Within cluster sum of squares by cluster:

```
[1] 31.16488 43.71399 85.54479
 (between_SS / total_SS =  61.3 %)
```

Available components:

```
[1] "cluster"      "centers"
[3] "totss"        "withinss"
[5] "tot.withinss" "betweenss"
[7] "size"         "iter"
[9] "ifault"
```

```
# cluster_1$totss
[1] 414
# cluster_1$betweenss
[1] 253.5763
# cluster_1$tot.withinss
[1] 160.4237
# cluster_1$betweenss/cluster_1$totss*100
[1] 61.25032
# cluster_1$tot.withinss/cluster_1$totss*100
[1] 38.74968
```

Total = 414

Total of between = 226.1073 (55% heterogeneity)
Total of within = 187.8927 (45% heterogeneity)

The above data indicates high homogeneity within the clusters and low homogeneity between the clusters.

Now we will segregate the players as per below categories; 1. Valuable, 2. Extremely Valuable and 3. Under Performance

```
# IPL$cluster_1 <- cluster_1$cluster
# IPL$cluster_1<-replace(IPL$cluster_1, IPL$cluster_1 ==1, "Extremely Valuable")
# IPL$cluster_1<-replace(IPL$cluster_1, IPL$cluster_1 ==3, "Valuable")
# IPL$cluster_1<-replace(IPL$cluster_1, IPL$cluster_1 ==2, "Under performer")
#IPL
   X            Player
1   1      AB de Villiers
2   2      Ajinkya Rahane
3   3       Akshdeep Nath
4   4       Ambati Rayudu
5   5       Andre Russell
6   6          Axar Patel
7   7          Ben Stokes
8   8   Bhuvneshwar Kumar
9   9         Chris Gayle
10 10          Chris Lynn
11 11        Chris Morris
12 12        Colin Ingram
13 13        David Warner
14 14        David Miller
15 15        Deepak Hooda
16 16      Dinesh Karthik
17 17        Dwayne Bravo
18 18      Faf du Plessis
19 19       Hardik Pandya
20 20        Ishan Kishan
21 21        Jofra Archer
22 22      Jonny Bairstow
23 23         Jos Buttler
24 24     Kane Williamson
25 25        Kedar Jadhav
26 26          Keemo Paul
27 27      Kieron Pollard
28 28            KL Rahul
```

29	29	Krunal Pandya
30	30	Mandeep Singh
31	31	Manish Pandey
32	32	Marcus Stoinis
33	33	Mayank Agarwal
34	34	Moeen Ali
35	35	Mohammad Nabi
36	36	MS Dhoni
37	37	Nicholas Pooran
38	38	Nitish Rana
39	39	Parthiv Patel
40	40	Piyush Chawla
41	41	Prithvi Shaw
42	42	Quinton de Kock
43	43	Rahul Tripathi
44	44	Rashid Khan
45	45	Ravichandran Ashwin
46	46	Ravindra Jadeja
47	47	Rishabh Pant
48	48	Riyan Parag
49	49	Robin Uthappa
50	50	Rohit Sharma
51	51	Sam Curran
52	52	Sanju Samson
53	55	Sarfaraz Khan
54	56	Shane Watson
55	57	Sherfane Rutherford
56	60	Shikhar Dhawan
57	61	Shimron Hetmyer
58	63	Shreyas Gopal
59	64	Shreyas Iyer
60	65	Shubman Gill
61	67	Steve Smith
62	68	Stuart Binny
63	71	Sunil Narine
64	72	Suresh Raina
65	74	Suryakumar Yadav
66	76	Umesh Yadav
67	77	Vijay Shankar
68	80	Virat Kohli
69	84	Wriddhiman Saha
70	92	Yusuf Pathan
		Team

1	Royal Challengers Bangalore
2	Rajasthan Royals
3	Royal Challengers Bangalore
4	Chennai Super Kings
5	Kolkata Knight Riders
6	Royal Challengers Bangalore
7	Rajasthan Royals
8	Sunrisers Hyderabad
9	Kings XI Punjab
10	Kolkata Knight Riders
11	Delhi Capitals
12	Delhi Capitals
13	Sunrisers Hyderabad
14	Kings XI Punjab
15	Sunrisers Hyderabad
16	Kolkata Knight Riders
17	Chennai Super Kings
18	Chennai Super Kings
19	Mumbai Indians
20	Mumbai Indians
21	Rajasthan Royals
22	Sunrisers Hyderabad
23	Rajasthan Royals
24	Sunrisers Hyderabad
25	Chennai Super Kings
26	Rajasthan Royals
27	Mumbai Indians
28	Kings XI Punjab
29	Mumbai Indians
30	Kings XI Punjab
31	Sunrisers Hyderabad
32	Royal Challengers Bangalore
33	Kings XI Punjab
34	Royal Challengers Bangalore
35	Sunrisers Hyderabad
36	Chennai Super Kings
37	Kings XI Punjab
38	Kolkata Knight Riders
39	Royal Challengers Bangalore
40	Kolkata Knight Riders
41	Delhi Capitals
42	Mumbai Indians
43	Rajasthan Royals

44	Sunrisers Hyderabad
45	Kings XI Punjab
46	Chennai Super Kings
47	Delhi Capitals
48	Rajasthan Royals
49	Kolkata Knight Riders
50	Mumbai Indians
51	Kings XI Punjab
52	Rajasthan Royals
53	Kings XI Punjab
54	Chennai Super Kings
55	Delhi Capitals
56	Delhi Capitals
57	Royal Challengers Bangalore
58	Rajasthan Royals
59	Delhi Capitals
60	Kolkata Knight Riders
61	Rajasthan Royals
62	Rajasthan Royals
63	Kolkata Knight Riders
64	Chennai Super Kings
65	Mumbai Indians
66	Kolkata Knight Riders
67	Sunrisers Hyderabad
68	Royal Challengers Bangalore
69	Sunrisers Hyderabad
70	Sunrisers Hyderabad

	Runs	Avg	SR	Hundreds
1	442	44.20	154.00	0
2	393	32.75	137.89	1
3	61	12.20	107.01	0
4	282	23.50	93.06	0
5	510	56.66	204.81	0
6	110	18.33	125.00	0
7	123	20.50	124.24	0
8	12	4.00	63.15	0
9	490	40.83	153.60	0
10	405	31.15	139.65	0
11	32	5.33	86.48	0
12	184	18.40	119.48	0
13	692	69.20	143.86	1
14	213	26.62	129.87	0
15	64	10.66	101.58	0

16	253	31.62	146.24	0
17	80	16.00	121.21	0
18	396	36.00	123.36	0
19	402	44.66	191.42	0
20	101	16.83	101.00	0
21	67	33.50	167.50	0
22	445	55.62	157.24	1
23	311	38.87	151.70	0
24	156	22.28	120.00	0
25	162	18.00	95.85	0
26	18	3.60	75.00	0
27	279	34.87	156.74	0
28	593	53.90	135.38	1
29	183	16.63	122.00	0
30	165	41.25	137.50	0
31	344	43.00	130.79	0
32	211	52.75	135.25	0
33	332	25.53	141.88	0
34	220	27.50	165.41	0
35	115	19.16	151.31	0
36	416	83.20	134.62	0
37	168	28.00	157.00	0
38	344	34.40	146.38	0
39	373	26.64	139.17	0
40	42	14.00	113.51	0
41	353	22.06	133.71	0
42	529	35.26	132.91	0
43	141	23.50	119.49	0
44	34	6.80	147.82	0
45	42	8.40	150.00	0
46	106	35.33	120.45	0
47	488	37.53	162.66	0
48	160	32.00	126.98	0
49	282	31.33	115.10	0
50	405	28.92	128.57	0
51	95	23.75	172.72	0
52	342	34.20	148.69	1
53	180	45.00	125.87	0
54	398	23.41	127.56	0
55	73	14.60	135.18	0
56	521	34.73	135.67	0
57	90	18.00	123.28	0
58	63	15.75	136.95	0

59	463	30.86	119.94	0
60	296	32.88	124.36	0
61	319	39.87	116.00	0
62	70	23.33	175.00	0
63	143	17.87	166.27	0
64	383	23.93	121.97	0
65	424	32.61	130.86	0
66	25	12.50	100.00	0
67	244	20.33	126.42	0
68	464	33.14	141.46	1
69	86	17.20	162.26	0
70	40	13.33	88.88	0

	Fifties	Fours	Sixes	Salary
1	5	31	26	1.71875
2	1	45	9	0.62500
3	0	5	2	0.51430
4	1	20	7	0.34375
5	4	31	52	1.32813
6	0	10	3	0.71430
7	0	8	4	1.95313
8	0	1	0	1.32813
9	4	45	34	0.31250
10	4	41	22	1.50000
11	0	1	2	1.71875
12	0	20	5	0.91430
13	8	57	21	1.95313
14	1	19	7	0.46875
15	0	5	1	0.56250
16	2	22	14	1.15625
17	0	6	3	1.00000
18	3	36	15	0.25000
19	1	28	29	1.71875
20	0	8	4	0.96875
21	0	4	4	1.12500
22	2	48	18	0.31430
23	3	38	14	0.68750
24	1	12	5	0.46875
25	1	19	3	1.21875
26	0	1	1	0.07140
27	1	14	22	0.84375
28	6	49	25	1.71875
29	0	18	5	1.37500
30	0	10	4	0.21875

31	3	34	6	1.71875
32	0	14	10	0.96875
33	2	26	14	0.15625
34	2	16	17	0.26563
35	0	8	7	0.15625
36	3	22	23	2.34375
37	0	10	14	0.60000
38	3	27	21	0.53125
39	2	48	10	0.26563
40	0	4	2	0.65625
41	2	45	9	0.18750
42	4	45	25	0.43750
43	1	13	2	0.53125
44	0	2	2	1.40625
45	0	3	3	1.18750
46	0	7	4	1.09375
47	3	37	27	2.34375
48	1	17	5	0.02860
49	1	28	10	1.00000
50	2	52	10	2.34375
51	1	13	3	1.02860
52	0	28	13	1.25000
53	1	19	4	0.03570
54	3	42	20	0.62500
55	0	2	7	0.28570
56	5	64	11	0.81250
57	1	4	7	0.60000
58	0	8	1	0.03125
59	3	41	14	1.09375
60	3	21	10	0.28125
61	3	30	4	1.95313
62	0	5	4	0.07813
63	0	17	9	1.95313
64	3	45	9	1.71875
65	2	45	10	0.50000
66	0	3	1	0.65625
67	0	11	12	0.50000
68	2	46	13	2.65625
69	0	13	1	0.17140
70	0	1	1	0.29688

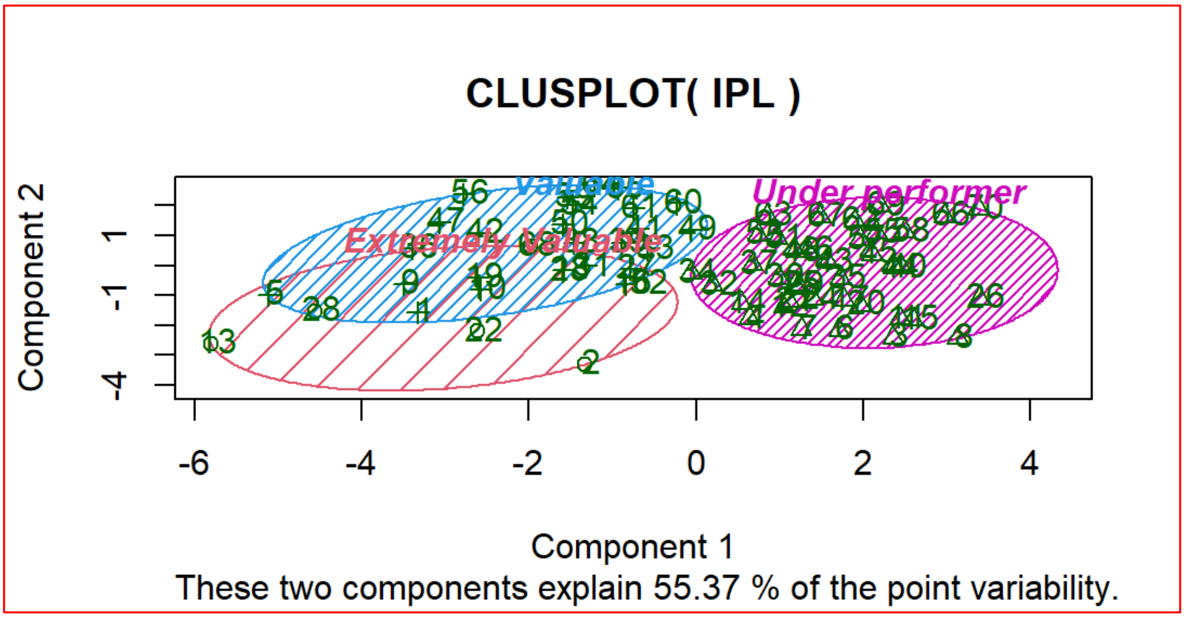
	cluster_1
1	Valuable
2	Extremely Valuable

3	Under performer
4	Under performer
5	Valuable
6	Under performer
7	Under performer
8	Under performer
9	Valuable
10	Valuable
11	Under performer
12	Under performer
13	Extremely Valuable
14	Under performer
15	Under performer
16	Valuable
17	Under performer
18	Valuable
19	Valuable
20	Under performer
21	Under performer
22	Extremely Valuable
23	Valuable
24	Under performer
25	Under performer
26	Under performer
27	Valuable
28	Extremely Valuable
29	Under performer
30	Under performer
31	Valuable
32	Under performer
33	Valuable
34	Under performer
35	Under performer
36	Valuable
37	Under performer
38	Valuable
39	Valuable
40	Under performer
41	Valuable
42	Valuable
43	Under performer
44	Under performer
45	Under performer

46   Under performer
47   Valuable
48   Under performer
49   Valuable
50   Valuable
51   Under performer
52 Extremely Valuable
53   Under performer
54   Valuable
55   Under performer
56   Valuable
57   Under performer
58   Under performer
59   Valuable
60   Valuable
61   Valuable
62   Under performer
63   Under performer
64   Valuable
65   Valuable
66   Under performer
67   Under performer
68 Extremely Valuable
69   Under performer
70   Under performer

We finally have the results as per given categories, let's put this in a ven diagram by using cluster packages

```
# install.packages("cluster")
# library("cluster")
# clusplot(IPL, IPL$cluster_1, color = TRUE, shade = TRUE, labels = 2, lines = 0)
```



With this above diagram we can draw the conclusion;

Cluster 1 (Valuable), the batsmen have scored - - high runs - a few 50s - maximum 6s - reasonable amount of 4s - have the second highest Batting Index.

#**Cluster 2** (Extremely valuable), the batsmen have scored - - the maximum runs - centuries - maximum 4s - reasonable amount of 6s - have a high Batting Index.

Cluster 3 (Under performer)

Thank You,

Sarveswara Sarma Nemani (Sarvesh)