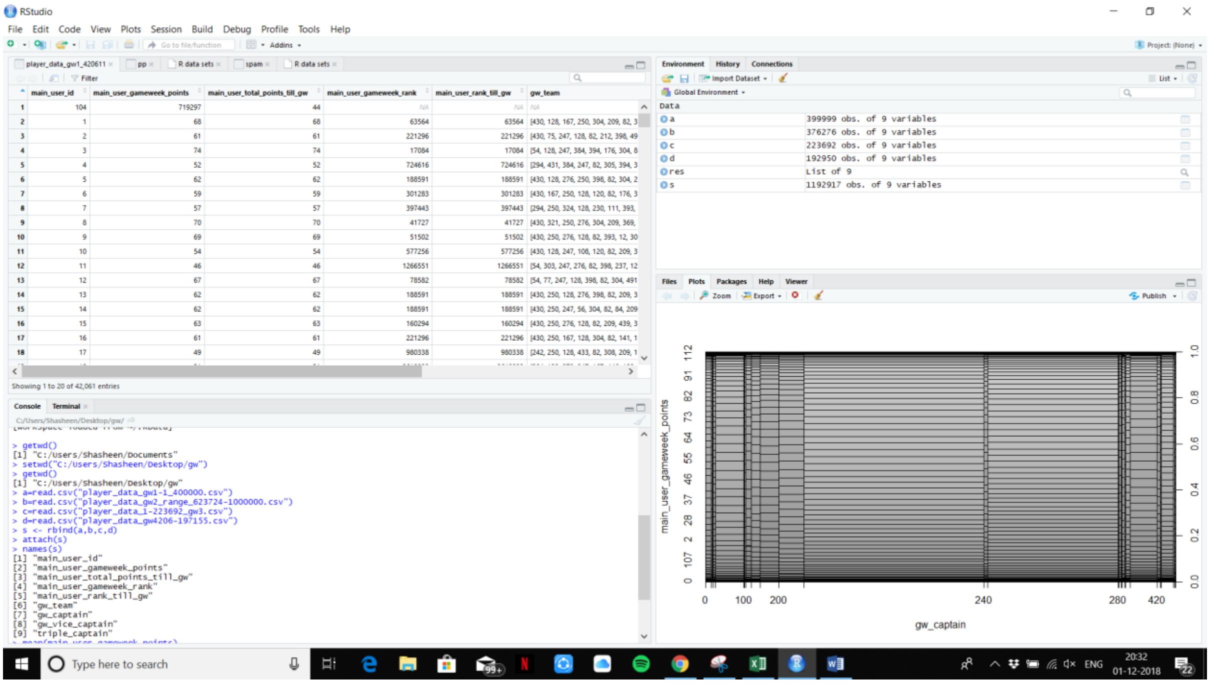
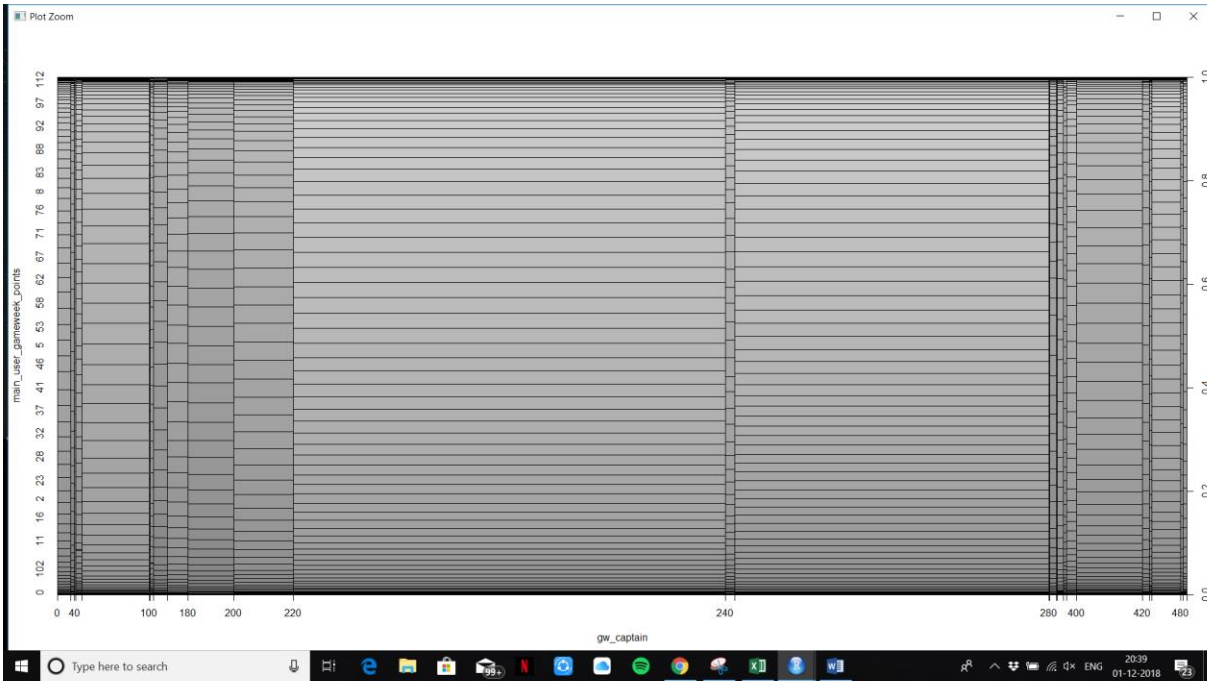
Fantasy Premier League Data Set () Final Project.

My fantasy premier league dataset contains csv files for 4 game weeks. So, this is a fantasy league so the number of users with their Unique user_id, game week points for that particular week, the number of users that got more points by their captaining a particular player. Etc. For this particular data set. I could only process 4 game weeks as the dataset was a big one. Basically how this works is that 4 csv files for each game week are merged into one to get consistency.

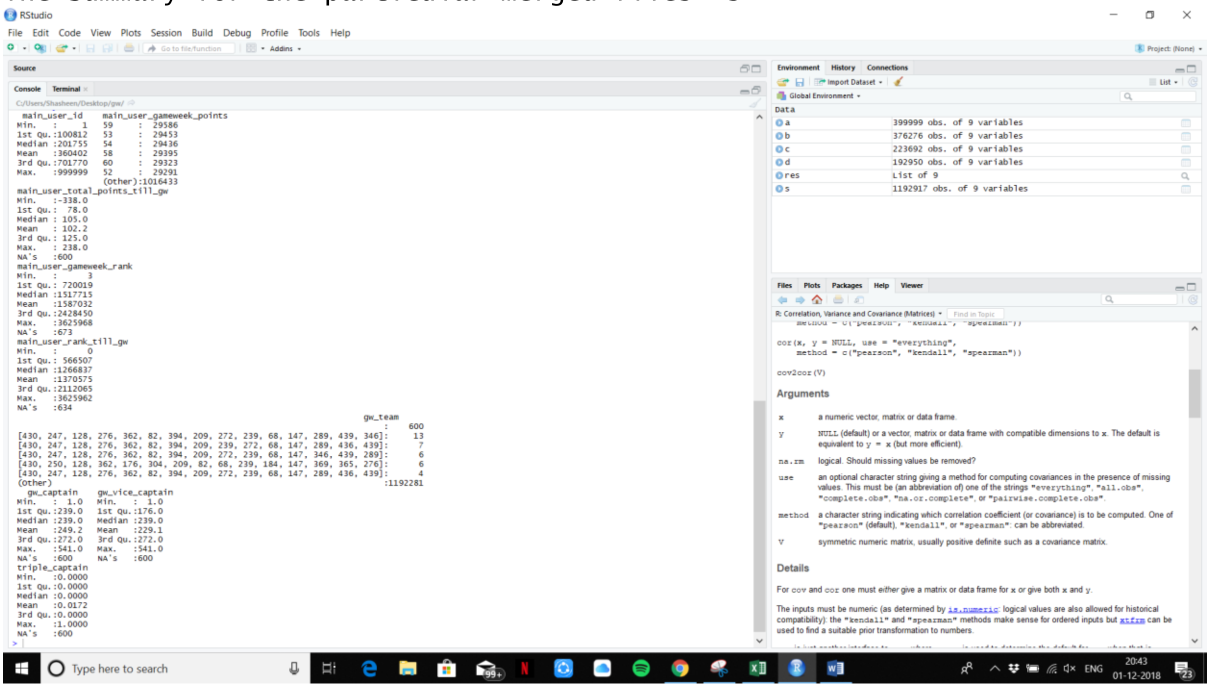


After binding 4 game week csv files we merge them into ne names as “S”.

Plotting main_user_gameweek_points~gw_captain
This shows the number of user points and compares them with the users who have significantly got more or less points because they captained a partic ular player and that player either performed very well for them or did not perform as good as they wanted him to.

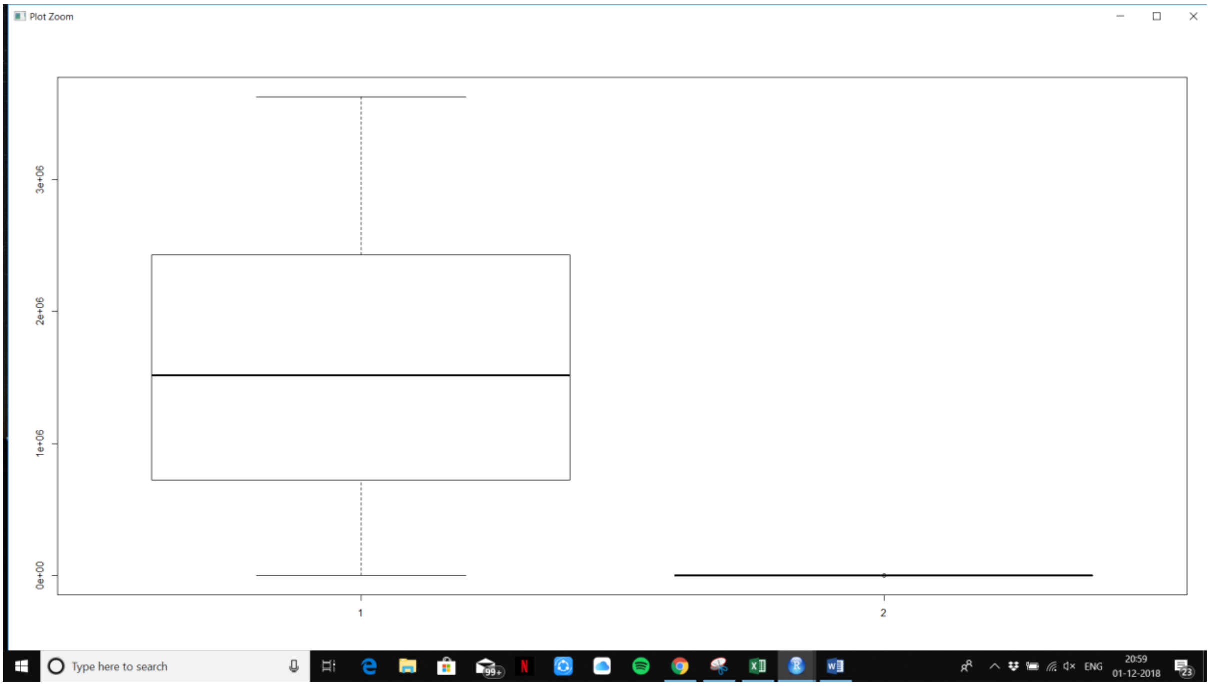


The Summary for the particular merged files “s”



From the summary we can analyse that over 29395 users have got a means gam e week score of “58” over the four game weeks.
The median was a score of 54 for the four game weeks.
We can also see that almost 250 users had a success rate of getting a high er score by captaining a player that was effective for their team. And alm ost 230 users were successful by vice captaining an effective player.
Note – The Captain and the Vice captain roles play a big part as if a play er who is captain or Vice captained scores a goal, his tally is doubled.
For eg A goals gives you 6 points and an assist gives you 3 points. So if a player has scored two goals and an assist, that takes his tally to 2x(2x6+1x3)= 30 when captained or vice captained. So the individual role o f a player plays a massive a part.

The following is a box plot for (main_user_gameweek_rank,gw_captain)



Applying Single linear model.
I was facing Trouble with inconsistent data with individual columns applying single linear model as there were some entries with no data and some with no entries ‘NA’. So I had to clean the data.

```
s[which(gw_captain=="-"),]
s[which(main_user_gameweek_points=="-"),]
```

for the specific columns.

```
> gw_captain=as.numeric(gw_captain)
> main_user_gameweek_points=as.numeric(main_user_gameweek_points)

> m=lm(main_user_gameweek_points~gw_captain)
> abline(coef(m)[1],coef(m)[2])
> summary(lm(main_user_gameweek_points~gw_captain))
```

```
Call:
lm(formula = main_user_gameweek_points ~ gw_captain)

Residuals:
    Min      1Q  Median      3Q     Max 
-58.452 -11.378   0.187  11.187  52.187 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.815e+01  4.436e-02  1310.9   <2e-16
gw_captain  2.769e-03  1.688e-04    16.4   <2e-16
            
(Intercept) ***
gw_captain  ***
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.34 on 1192315 degrees of freedom
  (600 observations deleted due to missingness)
```

```
Multiple R-squared:  0.0002256,       Adjusted R-squared:  0.0002248
F-statistic: 269.1 on 1 and 1192315 DF,  p-value: < 2.2e-16
```

The Adjusted R-squared value is very small. A standard error of just 15.34

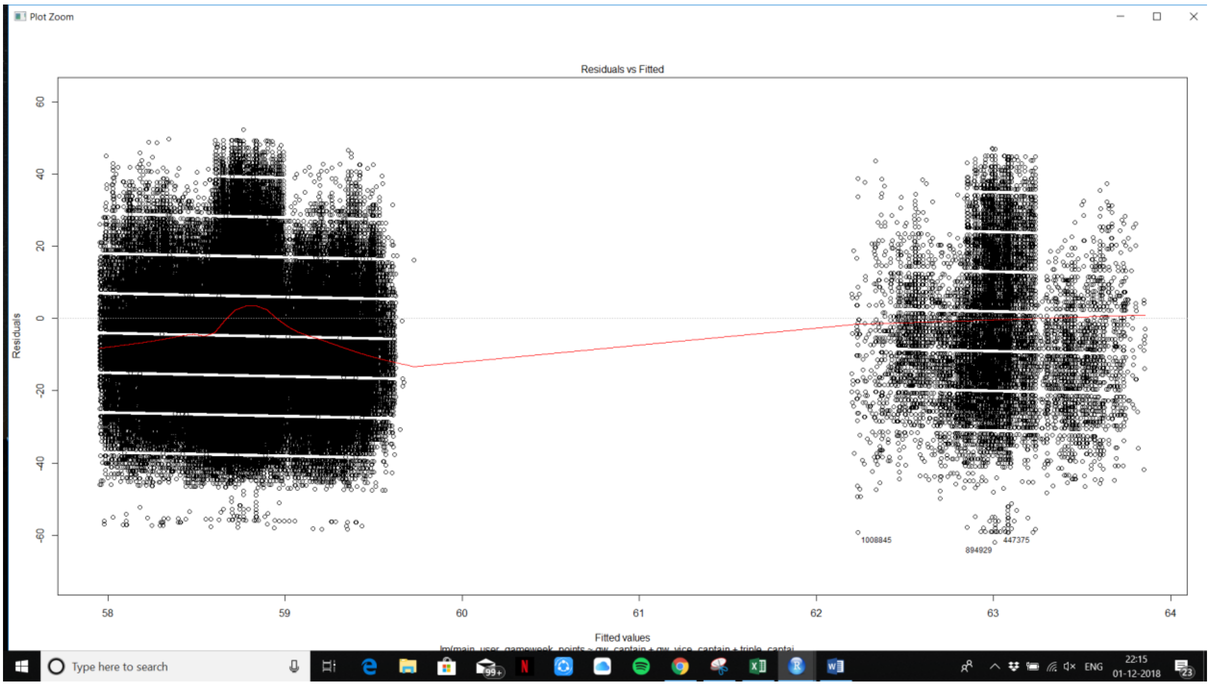
The lower the Rsquared value the better for the calculations.

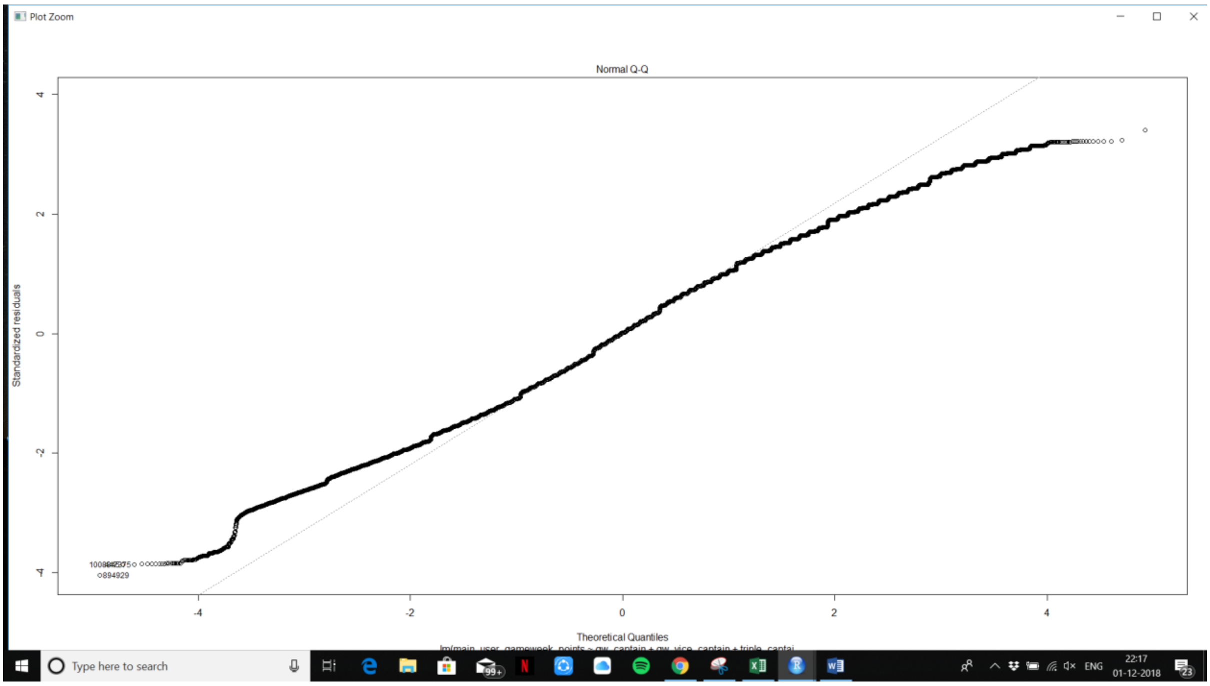
Multiple Linear Model.

```
> m=lm(main_user_gameweek_points~gw_captain+gw_vice_captain+triple_captain
)
> summary(m)

Call:
lm(formula = main_user_gameweek_points ~ gw_captain + gw_vice_captain +
    triple_captain)

Residuals:
    Min      1Q  Median      3Q     Max
-62.007 -11.358   0.222  11.290  52.235

Coefficients:
                 Estimate Std. Error t value
(Intercept)     5.795e+01  5.198e-02 1114.68
gw_captain      2.751e-03  1.687e-04   16.30
gw_vice_captain 5.907e-04  1.186e-04    4.98
triple_captain  4.244e+00  1.079e-01   39.34
                Pr(>|t|)
(Intercept)      < 2e-16 ***
gw_captain       < 2e-16 ***
gw_vice_captain 6.36e-07 ***
triple_captain   < 2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.33 on 1192313 degrees of freedom
  (600 observations deleted due to missingness)
Multiple R-squared:  0.001543, Adjusted R-squared:  0.00154
F-statistic: 614.1 on 3 and 1192313 DF,  p-value: < 2.2e-16
```

The Adjusted R-squared value is very small but comparatively a bit larger. A standard error of just 15.34 and is the same as the single linear model.

Plotting:





<u>Applying general linear model(glm) </u>Applying for 4 game weeks

```
> s$main_user_gameweek_points=factor(s$main_user_gameweek_poin
ts)
> b=glm(main_user_gameweek_points~gw_captain+gw_vice_captain+t
riple_captain,data=s,family = binomial)
> summary(b)

Call:
glm(formula = main_user_gameweek_points ~ gw_captain + gw_vice
_captain +
    triple_captain, family = binomial, data = s)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.3386   0.0034   0.0039   0.0046   0.0370

Coefficients:
                 Estimate Std. Error z value
(Intercept)      9.566847   0.794202  12.046
gw_captain       0.010635   0.002853   3.728
gw_vice_captain -0.001667   0.002234  -0.746
triple_captain  -1.502942   1.037750  -1.448
                Pr(>|z|)
(Intercept)      < 2e-16 ***
gw_captain      0.000193 ***
gw_vice_captain 0.455611
triple_captain  0.147542

The lower the value of P the better the result .In the case of
vice_captain does not give a significant contribution. 45% for
vice_captain. 45% of the value is completely random and high.
The best values are the ones that are close to zero and as you
can see in the result gw_captain has a much greater significan
ce.
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 345.87  on 1192316  degrees of freedom
Residual deviance: 331.40  on 1192313  degrees of freedom
  (600 observations deleted due to missingness)
AIC: 339.4
```

Number of Fisher Scoring iterations: 14

Applying Clustering Algorithms – KMEANS

Plotting `plot(s$main_user_gameweek_points,s$gw_captain)`

I need to plot labels and explain.**


```
> head(s)
  main_user_id main_user_gameweek_points
1            1                        58
2            2                        65
3            3                        76
4            4                        67
5            5                        58
6            6                        58
  main_user_total_points_till_gw
1                            126
2                            126
3                            142
4                            119
5                            120
6                            117
  main_user_gameweek_rank main_user_rank_till_gw
1                 1721874                 296242
2                 1058180                 309468
3                  268297                  46635
4                  865676                 572425
5                 1721874                 509674
6                 1721874                 643569
                                                                            gw_te
am
1  [430, 128, 431, 250, 304, 209, 82, 393, 68, 239, 184, 147, 416, 276, 16
7]
2   [430, 75, 247, 128, 82, 134, 398, 491, 239, 184, 68, 147, 439, 276, 36
2]
3   [54, 436, 247, 384, 362, 267, 176, 304, 82, 239, 355, 274, 128, 49, 41
6]
4 [339, 363, 384, 247, 82, 305, 394, 134, 403, 239, 313, 294, 431, 148, 41
8]
5  [430, 128, 276, 250, 398, 82, 304, 209, 68, 239, 184, 147, 416, 197, 34
6]
6  [430, 362, 250, 128, 120, 82, 176, 308, 68, 239, 184, 147, 276, 167, 43
9]
  gw_captain gw_vice_captain triple_captain
1        239              82              0
```

```
2          239              82               0
3          239             267               0
4          239              82               0
5          239             184               0
6          239             184               0
> sum(!sapply(s,is.finite))
[1] 3707
> sum(!sapply(s,is.nan))
[1] 10736253
```

Plot for 4 clusters

> plot(pp$main_user_gameweek_points, pp$gw_vice_captain)
> kmeans(data.frame(pp$main_user_gameweek_points, pp$gw_vice_captain), centers = 4, nstart = 20)
K-means clustering with 4 clusters of sizes 100160, 49870, 95280, 175290

Cluster means:

	pp.main_user_gameweek_points	pp.gw_vice_captain
1	505.77177	660.25619
2	450.61640	4106.41789
3	500.36734	1084.45823
4	509.60294	2620.40265

Clustering vector:
[1] 1 1 1 1 3 3 4 3 3 2 3 3 1 1 3 3 3 4 1 3 1 3 4 4 4 1 4 4 3 4 1 4 1 4 4 1 3 3 2 4 1 3 3 2 4 2
4 4 4 4 4 3 3 4 3 4 4
[58] 3 1 1 3 3 1 1 4 1 1 1 3 4 4 4 3 4 3 1 2 4 3 3 3 1 4 1 3 3 3 3 4 3 4 4 3 1 3 4 1 1 3 3 1 3 2
3 1 4 3 3 4 3 4 4 3 4
[115] 1 4 4 4 3 3 1 4 4 1 1 2 4 1 1 1 1 3 4 3 4 4 2 4 3 3 4 4 3 1 1 3 3 1 3 1 4 4 3 1 2 4 3 4 2 1
3 3 3 4 2 1 1 1 4 4 2
[172] 1 3 3 1 1 4 4 1 3 4 1 4 3 1 4 1 4 4 2 4 4 3 1 4 4 1 3 4 4 4 1 4 3 4 1 4 4 4 4 4 3 1 1 4 4 4
1 3 2 4 4 1 1 4 4 3 1
[229] 2 3 2 4 4 4 4 1 3 3 1 3 1 3 4 1 1 3 1 1 3 4 3 2 4 1 2 3 4 4 4 4 4 3 4 4 1 4 3 4 3 3 3 4 1 4
3 2 3 4 4 4 4 3 3 3 3
[286] 1 3 2 1 4 3 2 4 3 4 4 3 4 3 4 4 4 3 4 3 3 3 4 2 4 3 4 3 1 4 3 4 4 1 4 3 3 4 4 4 4 1 1 3 1 1
4 4 1 4 1 4 4 3 1 1 4
[343] 3 2 1 3 3 4 2 2 1 1 4 3 2 3 4 3 1 1 4 4 3 3 4 3 1 2 4 2 1 1 4 1 3 3 4 4 3 3 4 1 4 3 4 4 4 4
4 2 4 3 2 3 3 1 4 3 1
[400] 3 3 1 2 4 4 3 4 1 1 3 1 1 4 1 3 1 2 3 4 3 4 4 1 4 4 3 4 3 4 2 4 3 3 3 1 3 4 4 1 3 3 4 4 1 2
4 3 1 3 1 3 4 1 3 1 4
[457] 3 4 1 3 3 4 1 4 4 2 1 4 4 1 2 4 4 2 3 1 2 1 4 3 1 1 3 1 1 1 4 4 2 1 1 3 1 3 3 1 3 3 4 1 3 3
3 3 4 4 4 3 4 1 3 3 4
[514] 1 3 1 1 4 1 4 3 4 1 4 2 3 4 4 3 2 3 2 4 3 1 1 1 1 2 4 3 3 4 3 4 3 4 3 1 1 3 1 1 3 1 4 1 1 4
1 4 2 1 4 4 4 1 4 4 4
[571] 4 1 1 2 2 3 3 1 1 3 4 1 4 3 1 1 2 1 2 3 4 1 4 4 3 4 1 1 1 4 1 3 1 4 3 1 4 1 4 4 4 3 3 1 1 4
4 1 1 2 4 3 4 3 3 4 3
[628] 4 3 1 2 4 1 4 1 4 3 4 1 1 3 1 1 3 3 1 1 3 4 3 4 3 3 4 2 2 4 2 3 4 1 2 4 1 4 4 4 1 1 4 1 3 4
2 3 3 4 1 1 4 4 1 3 2
[685] 4 3 4 3 1 4 4 1 1 1 1 2 1 1 4 4 3 1 4 4 3 1 1 3 3 1 1 4 1 4 4 1 3 4 1 3 4 2 4 4 2 3 1 3 1 1
1 4 4 4 1 4 3 3 4 1 3
[742] 2 4 3 4 4 4 2 1 3 1 1 4 1 3 3 2 1 1 3 3 4 4 2 4 2 3 1 3 4 4 1 1 3 3 1 3 3 3 4 3 3 3 2 3 4 1
3 1 3 4 4 3 2 3 4 3 3
[799] 4 4 1 3 1 3 4 3 4 1 3 3 3 4 3 4 4 4 4 4 1 2 4 4 3 4 3 3 4 4 4 1 2 4 4 3 1 3 1 4 1 3 3 4 3 3
2 3 3 4 2 3 1 2 1 1 4
[856] 4 4 1 4 1 1 1 4 3 1 1 3 4 3 4 2 4 3 4 1 4 4 4 4 3 1 4 1 3 1 3 3 3 3 4 3 2 4 4 3 4 3 4 4 2 4
1 3 3 3 1 4 1 1 1 3 3

[913] 4 2 2 1 4 4 3 3 3 3 3 1 4 1 1 4 3 4 3 3 4 1 2 1 3 3 3 1 3 3 2 3 1 1 2 1 3 1 1 2 4 4 2 3 1 2
4 3 1 3 4 2 4 2 2 1 4
[970] 1 3 1 4 3 4 4 3 1 4 1 3 4 4 4 1 4 4 1 1 1 2 4 1 4 4 4 3 3 3 4
[reached getOption("max.print") -- omitted 41060 entries]

Within cluster sum of squares by cluster:
[1] 118435039 72223086 40902208 110892343
(between_SS / total_SS = 93.1 %)
93 % accuracy is very good for this dataset while computing for 4 clusters.

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" "size"
[8] "iter" "ifault"
> cluster = kmeans(data.frame(pp$main_user_gameweek_points, pp$gw_vice_captain), centers = 2, nstart = 20)
> plot(main_user_gameweek_points,gw_vice_captain, col=ifelse(cluster==1, "red", "blue"))
Error in ifelse(cluster == 1, "red", "blue") :
(list) object cannot be coerced to type 'double'
> cluster
K-means clustering with 2 clusters of sizes 176388, 244522

Cluster means:

	pp.main_user_gameweek_points	pp.gw_vice_captain
1	590.85832	1194.3376
2	490.37757	2898.9347

Clustering vector:
[1] 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 2 2 2 1 2 2 1 2 1 2 1 2 2 1 1 1 2 2 1 1 1 2 2 2
2 2 2 2 2 1 1 2 1 2 2
[58] 1 1 1 1 1 1 2 1 1 1 1 2 2 2 1 2 1 1 2 2 1 1 1 1 2 1 1 1 1 1 2 1 2 2 1 1 1 2 1 1 1 1 1 1 2
1 1 2 1 1 2 1 2 2 1 2
[115] 1 2 2 2 1 1 1 2 2 1 1 2 2 1 1 1 1 1 2 1 2 2 2 2 1 1 2 2 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 2 2 1
1 1 1 2 2 1 1 1 2 2 2
[172] 1 1 1 1 1 2 2 1 1 2 1 2 1 1 2 1 2 2 2 2 2 1 1 2 2 1 1 2 2 2 1 2 1 2 1 2 2 2 2 2 1 1 1 2 2 2
1 1 2 2 2 1 1 2 2 1 1
[229] 2 1 2 2 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 2 2 1 2 1 2 2 2 2 2 1 2 2 1 2 1 2 1 1 1 2 1 2
1 2 1 2 2 2 2 1 1 1 1
[286] 1 1 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 1 2 1 1 1 2 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 2 2 2 1 1 1 1 1
2 2 1 2 1 2 2 1 1 1 2
[343] 1 2 1 1 1 2 2 2 1 1 2 1 2 1 2 1 1 1 2 2 1 1 2 1 1 2 2 2 1 1 2 1 1 1 2 2 1 1 2 1 2 1 2 2 2 2
2 2 2 1 2 1 1 1 2 1 1
[400] 1 1 1 2 2 2 1 2 1 1 1 1 1 2 1 1 1 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 2 1 1 1 1 2 2 1 1 2 2 2 1 2
2 1 1 1 1 1 2 1 1 1 2
[457] 1 2 1 1 1 2 1 2 2 2 1 2 2 1 2 2 2 2 1 1 2 1 2 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1
1 1 2 2 2 1 2 1 1 1 2
[514] 1 1 1 1 2 1 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 1 1 1 1 2 2 1 1 2 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 2
1 2 2 1 2 2 2 1 2 2 2
[571] 2 1 1 2 2 1 1 1 1 1 2 1 2 1 1 1 2 1 2 1 2 1 2 2 1 2 1 1 1 2 1 1 1 2 1 1 2 1 2 2 2 1 1 1 1 2
2 1 1 2 2 1 2 1 1 2 1
[628] 2 2 1 2 2 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 1 2 2 2 1 1 2 1 1 2
2 1 1 2 1 1 2 2 1 1 2
[685] 2 1 2 1 1 2 2 1 1 1 1 2 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 2 1 2 2 1 1 2 1 1 2 2 2 2 2 1 1 1 1 1
1 2 2 2 1 2 1 1 2 1 1

```
 [742] 2 2 1 2 2 2 2 1 1 1 1 2 1 2 2 2 1 1 1 1 2 2 2 2 2 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1
1 1 1 2 2 1 2 1 2 1 1
 [799] 2 2 1 1 1 1 2 1 2 1 1 1 1 2 1 2 2 2 2 2 1 2 2 2 1 2 1 1 2 2 2 1 2 2 2 1 1 1 1 2 1 1 1 2 1 1
2 1 1 2 2 1 1 2 1 1 2
 [856] 2 2 1 2 1 1 1 2 1 1 1 1 2 1 2 2 2 1 2 1 2 2 2 2 2 1 2 1 1 1 1 1 1 1 2 1 2 2 2 1 2 1 2 2 2 2
1 2 1 1 1 2 1 1 1 1 1
 [913] 2 2 2 1 2 2 1 1 2 1 1 1 2 1 1 2 1 2 1 2 2 1 2 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 2 2 2 2 1 1 2
2 1 1 1 2 2 2 2 2 1 2
 [970] 1 1 1 2 1 2 2 1 1 2 1 2 2 2 2 1 2 2 1 1 1 2 2 1 2 2 2 1 2 2 2
 [ reached getOption("max.print") -- omitted 41060 entries ]

Within cluster sum of squares by cluster:
[1]  673630849 1241079804
 (between_SS / total_SS =  62.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
> cluster[0]
named list()
> cluster[1]
$`cluster`
   [1] 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 2 2 2 1 2 2 1 2 1 2 1 2 2 1 1 1 2 2 1 1 1 2 2 2
2 2 2 2 2 1 1 2 1 2 2
  [58] 1 1 1 1 1 1 1 2 1 1 1 1 2 2 2 1 2 1 1 2 2 1 1 1 1 2 1 1 1 1 1 2 1 2 2 1 1 1 2 1 1 1 1 1 1 2
1 1 2 1 1 2 1 2 2 1 2
 [115] 1 2 2 2 1 1 1 2 2 1 1 2 2 1 1 1 1 1 2 1 2 2 2 2 1 1 2 2 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 2 2 1
1 1 1 2 2 1 1 1 2 2 2
 [172] 1 1 1 1 1 2 2 1 1 2 1 2 1 1 2 1 2 2 2 2 2 1 1 2 2 1 1 2 2 2 1 2 1 2 1 2 2 2 2 2 1 1 1 2 2 2
1 1 2 2 2 1 1 2 2 1 1
 [229] 2 1 2 2 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 2 2 1 2 1 2 2 2 2 2 1 2 2 1 2 1 2 1 1 1 2 1 2
1 2 1 2 2 2 2 1 1 1 1
 [286] 1 1 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 1 2 1 1 1 2 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 2 2 2 1 1 1 1 1
2 2 1 2 1 2 2 1 1 1 2
 [343] 1 2 1 1 1 2 2 2 1 1 2 1 2 1 2 1 1 1 2 2 1 1 2 1 1 2 2 2 1 1 2 1 1 1 2 2 1 1 2 1 2 1 2 2 2 2
2 2 2 1 2 1 1 1 2 1 1
 [400] 1 1 1 2 2 2 1 2 1 1 1 1 1 2 1 1 1 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 2 1 1 1 1 2 2 1 1 2 2 2 1 2
2 1 1 1 1 1 2 1 1 1 2
 [457] 1 2 1 1 1 2 1 2 2 2 1 2 2 1 2 2 2 2 1 1 2 1 2 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1
1 1 2 2 2 1 2 1 1 1 2
 [514] 1 1 1 1 2 1 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 1 1 1 1 2 2 1 1 2 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 2
1 2 2 1 2 2 2 1 2 2 2
 [571] 2 1 1 2 2 1 1 1 1 1 2 1 2 1 1 1 2 1 2 1 2 1 2 2 1 2 1 1 1 2 1 1 1 2 1 1 2 1 2 2 2 1 1 1 1 2
2 1 1 2 2 1 2 1 1 2 1
 [628] 2 2 1 2 2 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 1 2 2 2 1 1 2 1 1 2
2 1 1 2 1 1 2 2 1 1 2
 [685] 2 1 2 1 1 2 2 1 1 1 1 2 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 2 1 2 2 1 1 2 1 1 2 2 2 2 2 1 1 1 1 1
1 2 2 2 1 2 1 1 2 1 1
 [742] 2 2 1 2 2 2 2 1 1 1 1 2 1 2 2 2 1 1 1 1 2 2 2 2 2 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1
1 1 1 2 2 1 2 1 2 1 1
 [799] 2 2 1 1 1 1 2 1 2 1 1 1 1 2 1 2 2 2 2 2 1 2 2 2 1 2 1 1 2 2 2 1 2 2 2 1 1 1 1 2 1 1 1 2 1 1
2 1 1 2 2 1 1 2 1 1 2
```

```
 [856] 2 2 1 2 1 1 1 2 1 1 1 1 2 1 2 2 2 1 2 1 2 2 2 2 2 1 2 1 1 1 1 1 1 1 2 1 2 2 2 1 2 1 2 2 2 2
1 2 1 1 1 2 1 1 1 1 1
 [913] 2 2 2 1 2 2 1 1 2 1 1 1 2 1 1 2 1 2 1 2 2 1 2 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 2 2 2 2 1 1 2
2 1 1 1 2 2 2 2 2 1 2
 [970] 1 1 1 2 1 2 2 1 1 2 1 2 2 2 2 1 2 2 1 1 1 2 2 1 2 2 2 1 2 2 2
 [ reached getOption("max.print") -- omitted 41060 entries ]

> cluster$cluster
   [1] 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 2 2 2 1 2 2 1 2 1 2 1 2 2 1 1 1 2 2 1 1 1 2 2 2
2 2 2 2 2 1 1 2 1 2 2
  [58] 1 1 1 1 1 1 1 2 1 1 1 1 2 2 2 1 2 1 1 2 2 1 1 1 1 2 1 1 1 1 1 2 1 2 2 1 1 1 2 1 1 1 1 1 1 2
1 1 2 1 1 2 1 2 2 1 2
 [115] 1 2 2 2 1 1 1 2 2 1 1 2 2 1 1 1 1 1 2 1 2 2 2 2 1 1 2 2 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 2 2 1
1 1 1 2 2 1 1 1 2 2 2
 [172] 1 1 1 1 1 2 2 1 1 2 1 2 1 1 2 1 2 2 2 2 2 1 1 2 2 1 1 2 2 2 1 2 1 2 1 2 2 2 2 2 1 1 1 2 2 2
1 1 2 2 2 1 1 2 2 1 1
 [229] 2 1 2 2 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 2 2 1 2 1 2 2 2 2 2 1 2 2 1 2 1 2 1 1 1 2 1 2
1 2 1 2 2 2 2 1 1 1 1
 [286] 1 1 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 1 2 1 1 1 2 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 2 2 2 1 1 1 1 1
2 2 1 2 1 2 2 1 1 1 2
 [343] 1 2 1 1 1 2 2 2 1 1 2 1 2 1 2 1 1 1 2 2 1 1 2 1 1 2 2 2 1 1 2 1 1 1 2 2 1 1 2 1 2 1 2 2 2 2
2 2 2 1 2 1 1 1 2 1 1
 [400] 1 1 1 2 2 2 1 2 1 1 1 1 1 2 1 1 1 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 2 1 1 1 1 2 2 1 1 2 2 2 1 2
2 1 1 1 1 1 2 1 1 1 2
 [457] 1 2 1 1 1 2 1 2 2 2 1 2 2 1 2 2 2 2 1 1 2 1 2 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1
1 1 2 2 2 1 2 1 1 1 2
 [514] 1 1 1 1 2 1 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 1 1 1 1 2 2 1 1 2 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 2
1 2 2 1 2 2 2 1 2 2 2
 [571] 2 1 1 2 2 1 1 1 1 1 2 1 2 1 1 1 2 1 2 1 2 1 2 2 1 2 1 1 1 2 1 1 1 2 1 1 2 1 2 2 2 1 1 1 1 2
2 1 1 2 2 1 2 1 1 2 1
 [628] 2 2 1 2 2 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 1 2 2 2 1 1 2 1 1 2
2 1 1 2 1 1 2 2 1 1 2
 [685] 2 1 2 1 1 2 2 1 1 1 1 2 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 2 1 2 2 1 1 2 1 1 2 2 2 2 2 1 1 1 1 1
1 2 2 2 1 2 1 1 2 1 1
 [742] 2 2 1 2 2 2 2 1 1 1 1 2 1 2 2 2 1 1 1 1 2 2 2 2 2 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1
1 1 1 2 2 1 2 1 2 1 1
 [799] 2 2 1 1 1 1 2 1 2 1 1 1 1 2 1 2 2 2 2 2 1 2 2 2 1 2 1 1 2 2 2 1 2 2 2 1 1 1 1 2 1 1 1 2 1 1
2 1 1 2 2 1 1 2 1 1 2
 [856] 2 2 1 2 1 1 1 2 1 1 1 1 2 1 2 2 2 1 2 1 2 2 2 2 2 1 2 1 1 1 1 1 1 1 2 1 2 2 2 1 2 1 2 2 2 2
1 2 1 1 1 2 1 1 1 1 1
 [913] 2 2 2 1 2 2 1 1 2 1 1 1 2 1 1 2 1 2 1 2 2 1 2 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 2 2 2 2 1 1 2
2 1 1 1 2 2 2 2 2 1 2
 [970] 1 1 1 2 1 2 2 1 1 2 1 2 2 2 2 1 2 2 1 1 1 2 2 1 2 2 2 1 2 2 2
 [ reached getOption("max.print") -- omitted 41060 entries ]
> plot(main_user_gameweek_points,gw_vice_captain, col=ifelse(cluster$cluster==1, "red",
"blue"))
```

As you can see with the graph. The data has no particular clustering center but is divided very clearly with the number of levels. This basically shows that the data is consistent through and fairly accurate. This is for 2 clusters.

Now for clusters.

```
> cluster = kmeans(data.frame(pp$main_user_gameweek_points, pp$gw_vice_captain),
centers = 4, nstart = 20)
> cluster$cluster
  [1] 4 4 4 4 1 1 2 1 1 3 1 1 4 4 1 1 1 2 4 1 4 1 2 2 2 4 2
 [28] 2 1 2 4 2 4 2 2 4 1 1 3 2 4 1 1 3 2 3 2 2 2 2 2 1 1 2
 [55] 1 2 2 1 4 4 1 1 4 4 2 4 4 4 1 2 2 2 1 2 1 4 3 2 1 1 1
 [82] 4 2 4 1 1 1 1 2 1 2 2 1 4 1 2 4 4 1 1 4 1 3 1 4 2 1 1
[109] 2 1 2 2 1 2 4 2 2 2 1 1 4 2 2 4 4 3 2 4 4 4 4 1 2 1 2
[136] 2 3 2 1 1 2 2 1 4 4 1 1 4 1 4 2 2 1 4 3 2 1 2 3 4 1 1
[163] 1 2 3 4 4 4 2 2 3 4 1 1 4 4 2 2 4 1 2 4 2 1 4 2 4 2 2
[190] 3 2 2 1 4 2 2 4 1 2 2 2 4 2 1 2 4 2 2 2 2 2 1 4 4 2 2
[217] 2 4 1 3 2 2 4 4 2 2 1 4 3 1 3 2 2 2 2 4 1 1 4 1 4 1 2
[244] 4 4 1 4 4 1 2 1 3 2 4 3 1 2 2 2 2 2 1 2 2 4 2 1 2 1 1
[271] 1 2 4 2 1 3 1 2 2 2 2 1 1 1 1 4 1 3 4 2 1 3 2 1 2 2 1
[298] 2 1 2 2 2 1 2 1 1 1 2 3 2 1 2 1 4 2 1 2 2 4 2 1 1 2 2
[325] 2 2 4 4 1 4 4 2 2 4 2 4 2 2 1 4 4 2 1 3 4 1 1 2 3 3 4
[352] 4 2 1 3 1 2 1 4 4 2 2 1 1 2 1 4 3 2 3 4 4 2 4 1 1 2 2
[379] 1 1 2 4 2 1 2 2 2 2 2 3 2 1 3 1 1 4 2 1 4 1 1 4 3 2 2
[406] 1 2 4 4 1 4 4 2 4 1 4 3 1 2 1 2 2 4 2 2 1 2 1 2 3 2 1
[433] 1 1 4 1 2 2 4 1 1 2 2 4 3 2 1 4 1 4 1 2 4 1 4 2 1 2 4
[460] 1 1 2 4 2 2 3 4 2 2 4 3 2 2 3 1 4 3 4 2 1 4 4 1 4 4 4
[487] 2 2 3 4 4 1 4 1 1 4 1 1 2 4 1 1 1 1 2 2 2 1 2 4 1 1 2
[514] 4 1 4 4 2 4 2 1 2 4 2 3 1 2 2 1 3 1 3 2 1 4 4 4 4 3 2
[541] 1 1 2 1 2 1 2 1 4 4 1 4 4 1 4 2 4 4 2 4 2 3 4 2 2 2 4
[568] 2 2 2 2 4 4 3 3 1 1 4 4 1 2 4 2 1 4 4 3 4 3 1 2 4 2 2
[595] 1 2 4 4 4 2 4 1 4 2 1 4 2 4 2 2 2 1 1 4 4 2 2 4 4 3 2
[622] 1 2 1 1 2 1 2 1 4 3 2 4 2 4 2 1 2 4 4 1 4 4 1 1 4 4 1
[649] 2 1 2 1 1 2 3 3 2 3 1 2 4 3 2 4 2 2 2 4 4 2 4 1 2 3 1
```

[676] 1 2 4 4 2 2 4 1 3 2 1 2 1 4 2 2 4 4 4 4 3 4 4 2 2 1 4
[703] 2 2 1 4 4 1 1 4 4 2 4 2 2 4 1 2 4 1 2 3 2 2 3 1 4 1 4
[730] 4 4 2 2 2 4 2 1 1 2 4 1 3 2 1 2 2 2 3 4 1 4 4 2 4 1 1
[757] 3 4 4 1 1 2 2 3 2 3 1 4 1 2 2 4 4 1 1 4 1 1 1 2 1 1 1
[784] 3 1 2 4 1 4 1 2 2 1 3 1 2 1 1 2 2 4 1 4 1 2 1 2 4 1 1
[811] 1 2 1 2 2 2 2 2 4 3 2 2 1 2 1 1 2 2 2 4 3 2 2 1 4 1 4
[838] 2 4 1 1 2 1 1 3 1 1 2 3 1 4 3 4 4 2 2 2 4 2 4 4 4 2 1
[865] 4 4 1 2 1 2 3 2 1 2 4 2 2 2 2 1 4 2 4 1 4 1 1 1 1 2 1
[892] 3 2 2 1 2 1 2 2 3 2 4 1 1 1 4 2 4 4 4 1 1 2 3 3 4 2 2
[919] 1 1 1 1 1 4 2 4 4 2 1 2 1 1 2 4 3 4 1 1 1 4 1 1 3 1 4
[946] 4 3 4 1 4 4 3 2 2 3 1 4 3 2 1 4 1 2 3 2 3 3 4 2 4 1 4
[973] 2 1 2 2 1 4 2 4 1 2 2 2 4 2 2 4 4 4 3 2 4 2 2 2 1 1 1
[1000] 2
[reached getOption("max.print") -- omitted 41060 entries]
> plot(main_user_gameweek_points,gw_vice_captain, col=ifelse(cluster$cluster==1, "red", ifelse (cluster$cluster==2,"blue", "green")))
> plot(main_user_gameweek_points,gw_vice_captain, col=ifelse(cluster$cluster==1, "red", ifelse (cluster$cluster==2,"blue", ifelse(cluster$cluster==3,"green","yellow"))))

KNN-
The dataset I chose is not suited for knn because there are no clear categories that can be used as labels. The dataset that is used is clearly too big to compute Knn for train and test data can gives inconsistent data.

Naïve Bayes-
The dataset I've chosen is not suited for naïve Bayes as Naïve bayes requires binary data which consists mainly of 3 or less coloumn.