# COMP4030-Cwk-20298647

20298647 Zhida Tian

## ANALYSIS AND PRE-PROCESSING

### 1.i

Table 1: centrality measure, dispersion measure and missing value of the dataset

| Features | Mode | Mean | Median | SD | IQR | Range | Missing |
|---|---|---|---|---|---|---|---|
| camcol | NA | NA | 4.0000 | NA | NA | NA | 50 |
| dec | -0.393345053 | 14.8267 | 0.4029 | 25.2083 | 36.1040 | 73.9249 | 49 |
| dia | 27.54042634 | 2000.0276 | 349.1009 | 22677.7304 | 464.1442 | 848144.2423 | 6646 |
| fiberid | 155 | NA | NA | NA | NA | NA | 32 |
| field | 301 | NA | NA | NA | NA | NA | 50 |
| flux | 189.7145297 | 183.5176 | 183.3246 | 50.2938 | 50.8474 | 309.5857 | 50 |
| g | 17.55623 | 17.3719 | 17.4951 | 0.9455 | 1.1951 | 7.1194 | 51 |
| i | 14.06613 | 16.5839 | 16.5552 | 1.1419 | 1.4048 | 16.2324 | 50 |
| m_unt | 1e-04 | 0.0002 | 0.0002 | 0.0001 | 0.0001 | 0.0004 | 46 |
| mjd | NA | 52944.4652 | NA | 1511.3414 | 2568.0000 | 5903.0000 | 50 |
| native | 1 | NA | NA | NA | NA | NA | 50 |
| objid | 1.24e+18 | NA | NA | NA | NA | NA | 0 |
| plate | 2558 | NA | NA | NA | NA | NA | 50 |
| r | 15.99986 | 16.8408 | 16.8586 | 1.0677 | 1.3392 | 12.3704 | 53 |
| ra | 189.6527639 | 175.5448 | 180.4418 | 47.7724 | 44.1627 | 252.6493 | 48 |
| redshift | 0 | 0.1437 | 0.0425 | 0.3887 | 0.0925 | 5.3580 | 50 |
| rerun | 301 | NA | NA | NA | NA | NA | 30 |
| run | 756 | NA | NA | NA | NA | NA | 50 |
| specobjid | 2.88e+18 | NA | NA | NA | NA | NA | 50 |
| u | 18.90212 | 18.6190 | 18.8528 | 0.8290 | 1.0817 | 6.6109 | 50 |
| z | 13.9471 | 16.4231 | 16.3899 | 1.2035 | 1.5222 | 11.2226 | 53 |

**ii**

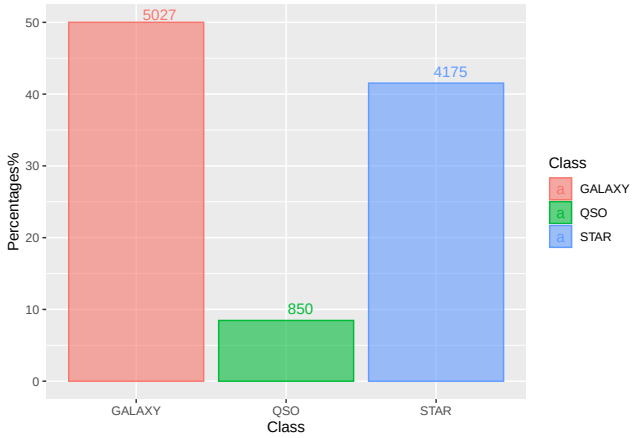


Figure 1: histogram of Class

The class "Galaxy" occurs the most with over 50% of the whole data set which could potentially cause overfitting. And *QSO* only takes up only 8.47% with 850 samples. This may also make it more difficult to predict the class "QSO", which occurs the least.

**iii.**

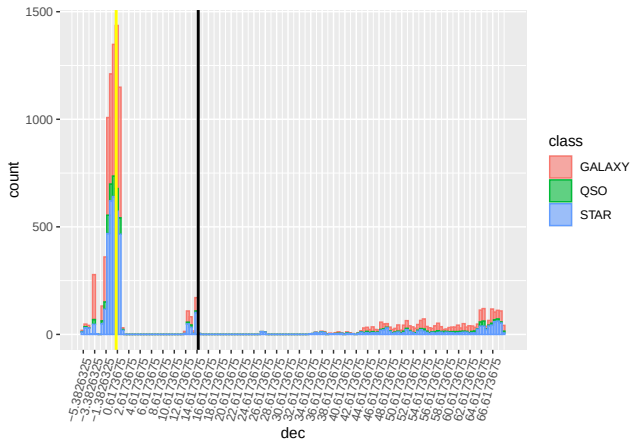


Figure 2: dispersion of dec

Dec is right skewed, as the majority of the data lies on the left side.

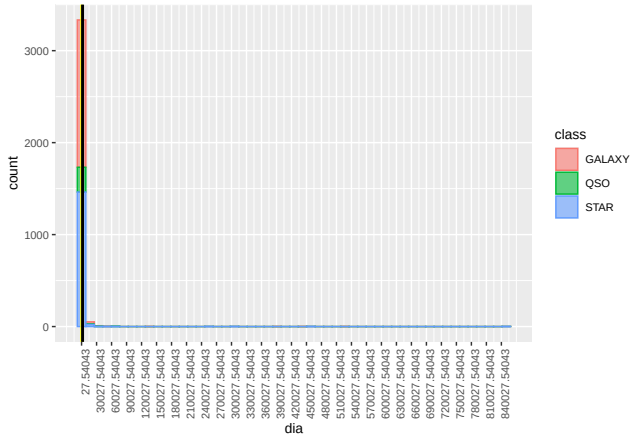


Figure 3: dispersion of dia

Dia: The shape of the distribution of the attribute of dia is right skewed.

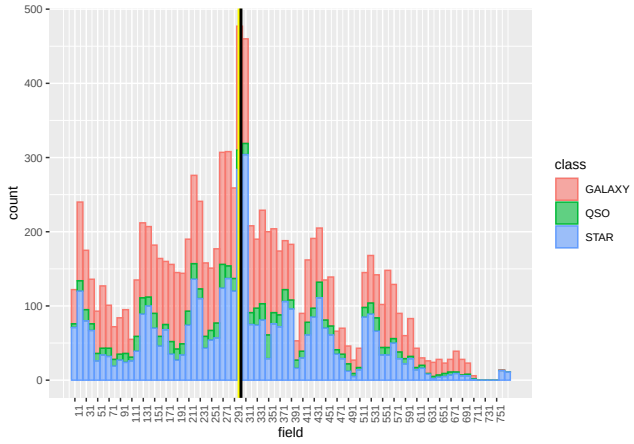


Figure 4: dispersion of field

Field: the shape is slightly right skew as the mean is slightly greater than the median. The mean is 302 while the median is 299.

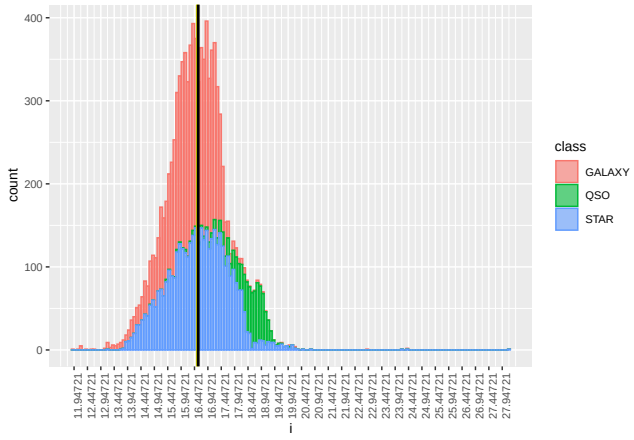


Figure 5: dispersion of i

I : The mean and median are the same integer of 16.58 and 16.56 they are not even an integer apart which implies there is no skewness within the infrared attribute.

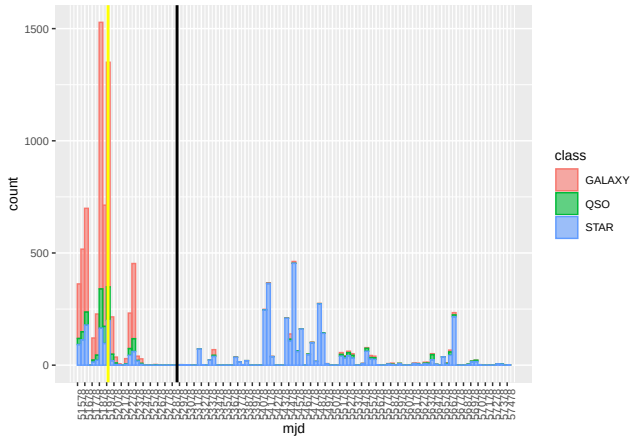


Figure 6: dispersion of mjd

Mjd: The mean is 52944.47 bigger than the median 51997 which implies the data is right skewed. This is visually apparent in the diagram as well.

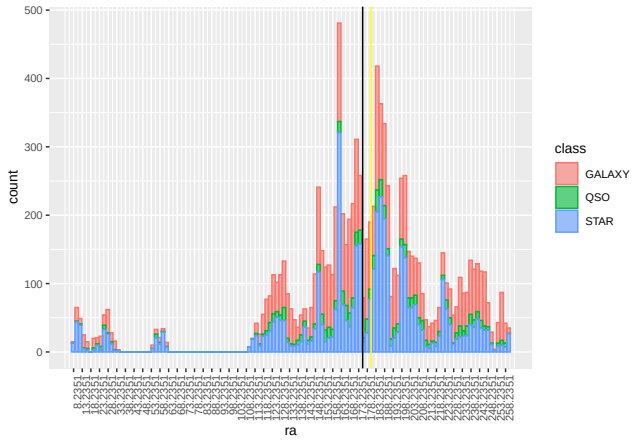


Figure 7: dispersion of ra

Ra: The shape of the distribution is left skew, as the mean is smaller than the median.

**2.i.**

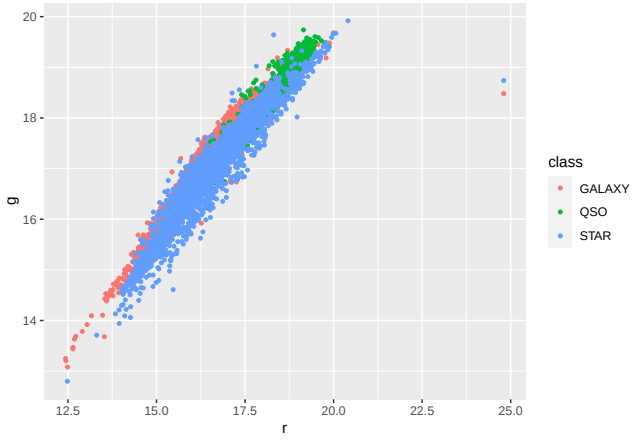


Figure 8: correlation between r and g

The correlation between r and g is 0.9581076, indicating a high, positive correlation between these variables. This is evident in the scatterplot.

**ii.**

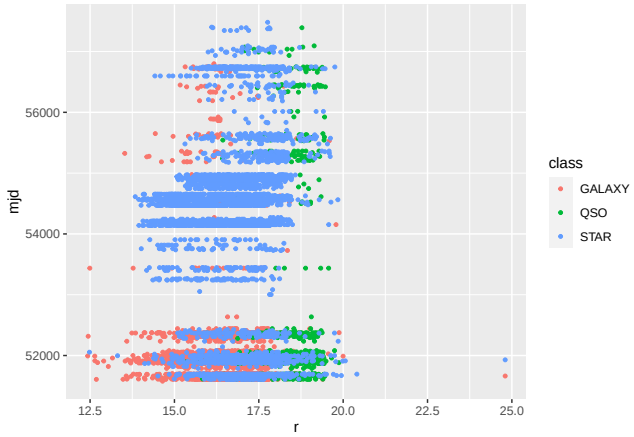


Figure 9: correlation between r and mjd

The correlation between mjd and r is -0.009189361, indicating a very low, negative correlation between these variables. This is also evidenced by this scatterplot which demonstrates that these two variables do not depend on each other.

**iii.**

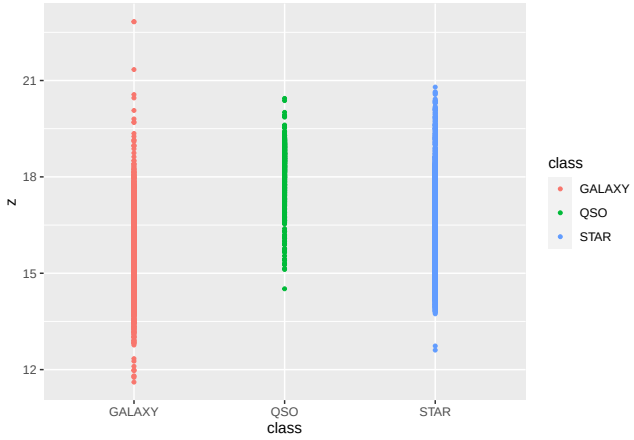


Figure 10: scatterplot between class and z

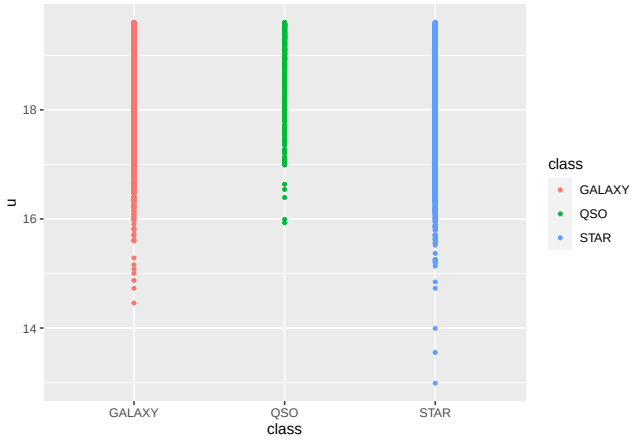


Figure 11: scatterplot between class and u

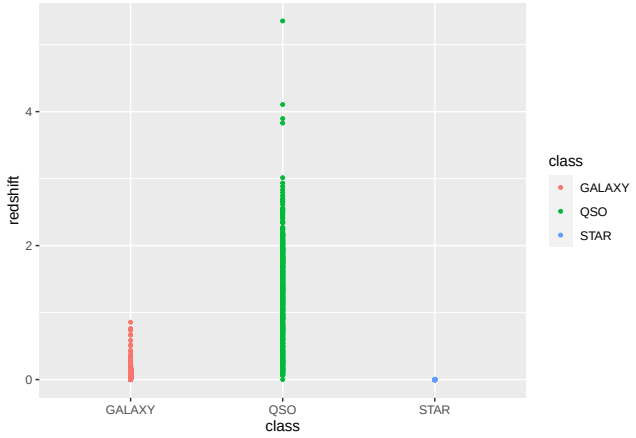


Figure 12: scatterplot between class and redshift

**iv.**

The boxplots are in the Appendix.

## 3.

I conclude that, the significant attributes are *redshift*, *g*, *r*, *i*, *z*, *mjd*, *plate*, *class* whereas the insignificant attributes are *rerun*, *objid*, *native*, *ra*, *dia*, *flux*, *m-unt*, *g*, *dec*, *camcol*, *fiberid*, *u*, *run*, *field*.

The boxplots obtained previously display that the attributes *redshift*, *g*, *r*, *i* and *z* help separate the class "QSO" from others. Additionally, from the scatterplot previously, I conclude that the attributes *r* and *mjd* have a low correlation. Moreover, the *mjd* boxplot of the three classes shows different distributions. Similarly, the *plate* histogram displays that the classes are more segregated for this attribute.

*objid* and *rerun* is the least significant attribute as it is consistent throughout the data and does not provide useful information or shows strong relationships with other attributes. *native* attribute

only has two values and from the histogram we can tell that with the value of this attribute changes the possibility of indicating different classes stay the same, thus providing no novelty. *ra*, *dia*, *flux*, *dec*, *camcol*, *fiberid*, *u*, *run*, *field* and *m-unt* have very similar distributions among all the classes. Therefore, they don't provide much information. Attribute *r* and *g* are highly correlated, which means they probably provide the same information given they have a strong relationship.

**4.**

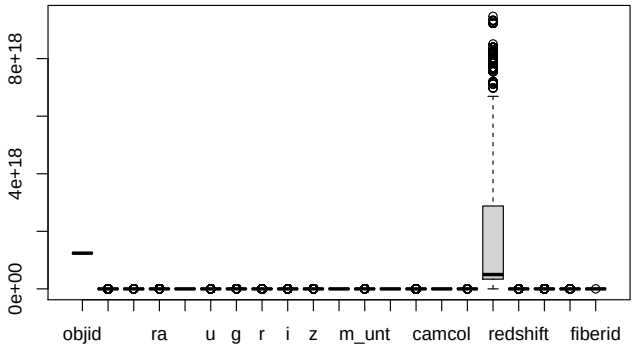


Figure 13: Boxplots of the data set replaced by zero value

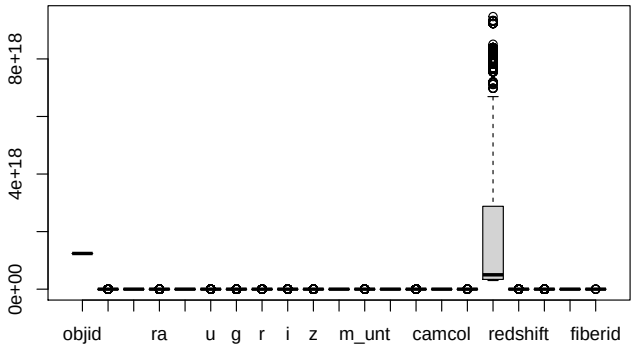


Figure 14: Boxplots of the data set replaced by mean value

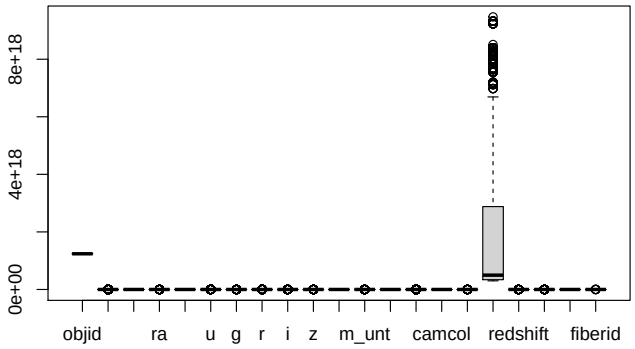


Figure 15: Boxplots of the data set replaced by median value

0 replacement prevents us from making false predictions of the values and has the advantage of not influencing the correlation either. It does not intefer with the relationships between the values conclusively it is the optimal way to deal with missing values.

mean replacement allows predictions to be unbiased, however there is disadvantage in how estimating values like the mean affect the conclusions we draw and decreases standard deviation.

Median has the similar disadvantages to the mean - of estimating values. Similar to the mean, it also provides lack of variability to our dataset. It has the advantage of eliminating noise however.

**5.**

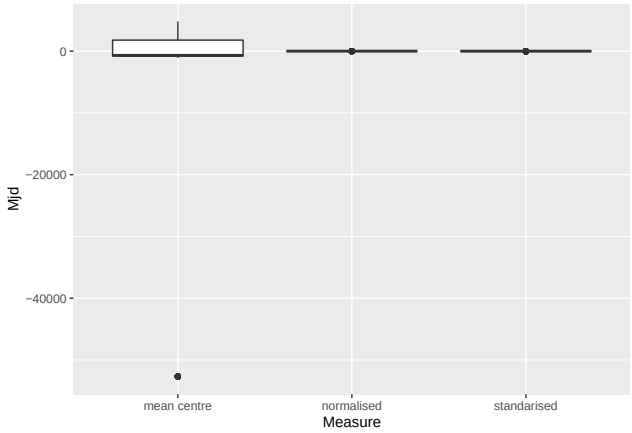


Figure 16: Boxplots of mjd attribute in data set replaced by zero

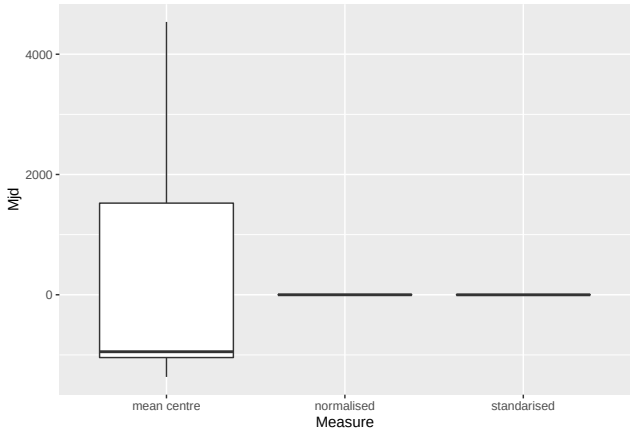


Figure 17: Boxplots of mjd attribute in data set replaced by mean value

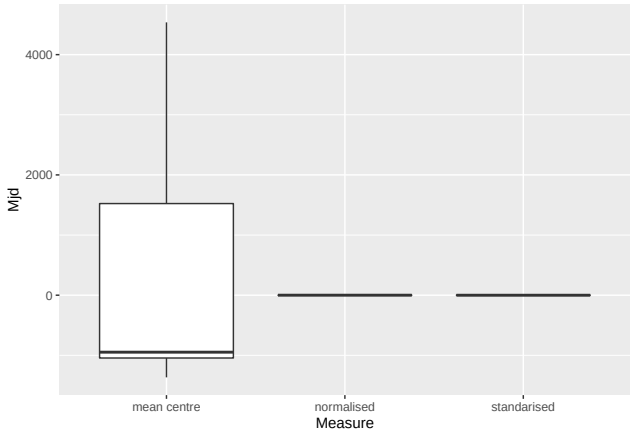


Figure 18: Boxplots of mjd attribute in data set replaced by median value

The Mean centring method calculates average value of each variable and subtracts this from the dataset. From the above boxplots I can conclude the mean centering method on the median and the mean are similar suggesting their averages are almost identical in their output. The number of outliers show a similar range. The zero boxplot has a slight decrease in range which indicates a wider spread of the data.

The Normalisation method scales the data .The boxplots proves an alike output to the mean centering where the zero has a smaller range than the others.

The standardisation method subtracts the mean and divide by the standard deviation. Each value reflects the distance away from the mean in standard deviations.In conclusion the boxplots, shows alikeness of the almost identical median and mean boxplots.

Out of the three methods, standardisation maintains outliars, this is useful as outliars provide information - therefore making it the most optimal method.

## 6. i.

Attribute *dia* has the highest missing value percentage(66.1162%), which is higher than 50%. Therefore, I decided to delete this attribute. The table contained full information about the missing value percentage within each attributes is in the Appendix.

In terms of the missing values per instances, I set the threshold as 50%. And there are 50 rows which have more than 50% of the missing value.

After removing those instances and attribute the data set will have 10002 samples and 21 attributes.

## ii.

Table 2: The correlation matrix of the data set been produced

| | dia | u | native | flux | redshift | mjd | fiberid |
|---|---|---|---|---|---|---|---|
| dia | 1.0000 | -0.0069 | 0.0015 | 0.0180 | -0.0185 | 0.0226 | 0.0072 |
| u | -0.0069 | 1.0000 | -0.0240 | 0.0364 | 0.1639 | -0.1698 | 0.0125 |

| | dia | u | native | flux | redshift | mjd | fiberid |
|---|---|---|---|---|---|---|---|
| native | 0.0015 | -0.0240 | 1.0000 | -0.0065 | -0.0130 | -0.0002 | -0.0222 |
| flux | 0.0180 | 0.0364 | -0.0065 | 1.0000 | 0.0481 | -0.0654 | 0.0956 |
| redshift | -0.0185 | 0.1639 | -0.0130 | 0.0481 | 1.0000 | -0.0583 | 0.0471 |
| mjd | 0.0226 | -0.1698 | -0.0002 | -0.0654 | -0.0583 | 1.0000 | 0.1882 |
| fiberid | 0.0072 | 0.0125 | -0.0222 | 0.0956 | 0.0471 | 0.1882 | 1.0000 |

I placed the missing value by mean values within different classes. Then generate the correlation matrix. From the table we can find that *objid* and *rerun* attributes have "NA" value to all other attributes. This is because there's no variation, there cannot be any co-variation, whence the covariances and the correlations are undefined. Therefore, it makes sense to delete them in the dataset. Attributes *dia* and *native* show low correlation to all of the rest attributes in the data set. Attributes *u*, *g*, *r*, *i* and *z* are highly correlated with each other. Similarly attributes *dec*, *run*, *ra*, *m-unt*, *flux* and *field* have high correlations. Attributes *specobjid*, *plate* and *mjd* are highly correlated.

### 7. i.

For PCA, I started from the raw dataset, replace the missing values with the mean according to each class to eliminate noise. I removed objid and rerun as these two attributes have the same value and would cause errors when standardizing due to their lack of variance.

**ii.**

Table 3: The PCA obtained a cumulative variance of at least 90%

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | PC8 |
|---|---|---|---|---|---|---|---|---|
| Standard deviation | 2.1486 | 1.7315 | 1.4697 | 1.3535 | 1.0006 | 0.9992 | 0.9803 | 0.9331 |
| Proportion of Variance | 0.2716 | 0.1764 | 0.1270 | 0.1078 | 0.0589 | 0.0587 | 0.0565 | 0.0512 |
| Cumulative Proportion | 0.2716 | 0.4479 | 0.5750 | 0.6828 | 0.7416 | 0.8004 | 0.8569 | 0.9081 |

In order to obtain at leaast 90% of cumulative variance we will need 8 PCs.

## CLUSTERING

### 1.

In this part I chose the data set from question 1.7i, the reduced PCA with 12 PCs. For the Internal metrics I used *average.between*, *average.within* and *dunn* criteria to evaluate the result. For external metrics: *precision*, *recall* and *F1*.

Table 4: Internal metrics for clustering result

| | hca | kmeans | pam |
|---|---|---|---|
| average.between | 35.19924 | 5.896196 | 6.016513 |
| average.within | 5.369524 | 4.692092 | 4.804679 |
| dunn | 0.4501908 | 0.001389611 | 0.006270207 |

Table 5: External metrics for HCA clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5001 | 0.9994 | 0.6667 |
| QSO | 0.0000 | 0.0000 | NaN |
| STAR | 1.0000 | 0.0002 | 0.0005 |

Table 6: External metrics for Kmeans clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.4907 | 0.3318 | 0.3959 |
| QSO | 0.1452 | 0.8835 | 0.2494 |
| STAR | 0.4811 | 0.1705 | 0.2518 |

Table 7: External metrics for PAM clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.8209 | 0.9664 | 0.8877 |
| STAR | 0.9479 | 0.7492 | 0.8369 |
| QSO | 0.8393 | 0.8235 | 0.8314 |

In evaluating the internal metrics, HCA produces the largest average distance with 35.2 which means HCA separates the cluster better. HCA also produces the most desirable dunn index result with 0.45. The smallest average within is the most desirable, as we want the clusters to be more compact, which kmeans produces with 4.7 - however all average.within results produced are within 1 integer of each other. In evaluating external metrics, QSO performed consistenly poor in all areas the HCA clustering method. PAM consistently showed to score well in all areas of external metrics for all classes showing it to be the most effective clustering method in terms of external metrics.

**2.**

For HCA, as presented in the table I changed the parameters for distance and "Meathod" for *hclust* function.

Table 8: Combinations of parameters for HCA clustering method

| HCA | Distance | Method |
|---|---|---|
| HCA1 | manhattan | complete |
| HCA2 | euclidean | complete |
| HCA3 | manhattan | single |
| HCA4 | euclidean | single |

Table 9: Internal metrics for clustering result

| | hca1 | hca2 | hca3 | hca4 |
|---|---|---|---|---|
| average.between | 35.19924 | 35.19924 | 35.19924 | 40.49842 |
| average.within | 5.369524 | 5.369524 | 5.369524 | 5.373386 |
| dunn | 0.4501908 | 0.4501908 | 0.4501908 | 0.3411657 |

Table 10: External metrices for HCA1 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5001 | 0.9994 | 0.6667 |
| QSO | 0.0000 | 0.0000 | NaN |
| STAR | 1.0000 | 0.0002 | 0.0005 |

As presented in the internal metrics, the *average.between* value is the same for HCA1 to HCA3. HCA4 generates the highest value. Therefore, it separates clusters the best. However,it has poor dunn value, which indicates it probably has higher diameter for the clusters - the clusters aren't compact.

In evaluating the external metrics the result are the same across all 4 hca methods. The precision and recall value are all 0 for QSO class, 1 for STAR and GALAXY class respectively. This indicates half of the GALAXY data are miss-calculated as QSO. Furthermore, there is almost no correct positive prediction to class QSO and STAR. Since all the result from internal and external metrics are almost the same, there is no clear conclusion of optimal performance.

For K-means method, I changed *nstart* and *Method* parameters.

Table 11: Combinations of parameters for kmeans clustering method

| Kmeans | nstart | Method |
|---|---|---|
| km1 | 5 | Hartigan-Wong |
| km2 | 10 | Hartigan-Wong |
| km3 | 5 | Lloyd |
| km4 | 10 | Lloyd |

Table 12: Internal metrics for Kmeans clustering result

| | km1 | km2 | km3 | km4 |
|---|---|---|---|---|
| average.between | 5.898279 | 5.898279 | 5.973041 | 5.898472 |
| average.within | 4.648573 | 4.648573 | 4.671534 | 4.648537 |
| dunn | 0.002835292 | 0.002835292 | 0.002569826 | 0.002835292 |

Table 13: External mectrices for km1 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5331 | 0.5049 | 0.5186 |
| QSO | 0.0058 | 0.0200 | 0.0090 |
| STAR | 0.3463 | 0.1954 | 0.2499 |

Table 14: External mectrices for km2 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5157 | 0.2417 | 0.3291 |
| QSO | 0.1067 | 0.5976 | 0.1811 |
| STAR | 0.5601 | 0.3938 | 0.4624 |

In the internal metrics km1, km3 and km2, km4 have almost the same value. The Higher *nstart* parameters produce better outputs with 0.0003 increment in dunn. Evidently all the dunn values are close to 0, showing worst clustering results.

There are no significant differences between the four k-means method. GALAXY class has a stable result in terms of all three criteria, while the results for class QSO and STAR varies with the change in "Method" parameter. In general km1 and km4 may have relatively higher F1 values. But It still proves to not be a successful indicator of predicting QSO class.

For PAM, I changed the parameters of distance metric and Pamonce as displayed in the table below.

Table 15: Combinations of parameters for pam clustering method

| PAM | Metric | Pamonce |
|---|---|---|
| pam1 | manhattan | 3 |
| pam2 | manhattan | 6 |
| pam3 | euclidean | 3 |
| pam4 | euclidean | 6 |

Table 16: Internal metrics for PAM clustering result

| | pam1 | pam2 | pam3 | pam4 |
|---|---|---|---|---|
| average.between | 5.630398 | 5.630838 | 6.016513 | 6.016513 |
| average.within | 4.993547 | 4.98387 | 4.804679 | 4.804679 |
| dunn | 0.001835621 | 0.002468855 | 0.006270207 | 0.006270207 |

Table 17: External mectrices for pam1 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| STAR | 0.5805 | 0.5732 | 0.5768 |
| GALAXY | 0.6012 | 0.4561 | 0.5187 |
| QSO | 0.1129 | 0.2812 | 0.1612 |

Table 18: External mectrices for pam2 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.6292 | 0.3700 | 0.4660 |
| QSO | 0.0926 | 0.4647 | 0.1545 |
| STAR | 0.6992 | 0.4743 | 0.5651 |

Table 19: External mectrices for pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.8209 | 0.9664 | 0.8877 |
| STAR | 0.9479 | 0.7492 | 0.8369 |
| QSO | 0.8393 | 0.8235 | 0.8314 |

For the Internal metrics for PAM, we can tell that pam1, pam2 and pam3, pam4 share the same value in average.between and average.within. Pam3 and pam4 show better seperation and compactness with 1 and 0.1 differences respectively. The dunn value is also higher in pam3 and pam4. But conclusively it is low as it's very close to 0.

In the external metrics, pam3 and pam4 have the exact same value. They both show relatively higher value in precision, recall and F1 compared with pam1 and pam2's value.

Conclusively, pam3 and pam4 produced better clustering result among all the experiments mentioned above.

**3.**

I choosed Pam clustering method with euclidean distance function and 6 for attribute *pamonce*.

**i.**

Table 20: Internal metrics for pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 6.0321 | 4.8206 | 0.008 |

Table 21: External metrics for pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.8211 | 0.9662 | 0.8878 |
| STAR | 0.9479 | 0.7497 | 0.8372 |
| QSO | 0.8383 | 0.8235 | 0.8309 |

**ii.**

Table 22: Internal metrics for pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 6.0165 | 4.8047 | 0.0063 |

Table 23: External metrics for pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.8209 | 0.9664 | 0.8877 |
| STAR | 0.9479 | 0.7492 | 0.8369 |
| QSO | 0.8393 | 0.8235 | 0.8314 |

**iii.**

Table 24: Internal metrics for pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 4908.854 | 1587.29 | 4e-04 |

Table 25: External metrics for pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.7277 | 0.9674 | 0.8306 |
| QSO | 0.0488 | 0.1929 | 0.0779 |

| Class | Precision | Recall | F1 |
|---|---|---|---|
| STAR | 0.5714 | 0.0010 | 0.0019 |

**iv.**

Table 26: Internal criterias for mean_centre_zero data set using pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 3.686891e+18 | 3.044154e+17 | 0.103 |

Table 27: External criterias for mean_centre_zero data set using pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.7356 | 0.9731 | 0.8379 |
| QSO | 0.0294 | 0.1000 | 0.0455 |
| STAR | 0.7767 | 0.0958 | 0.1706 |

Table 28: Internal metrics for mean_centre_mean data set using pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 3.67829e+18 | 3.021919e+17 | 0.103 |

Table 29: External metrics for mean_centre_mean data set using pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.7379 | 0.9731 | 0.8393 |
| QSO | 0.0292 | 0.1000 | 0.0452 |
| STAR | 0.7767 | 0.0958 | 0.1706 |

Table 30: Internal metrics for mean_centre_median data set using pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 3.679952e+18 | 3.015658e+17 | 0.103 |

Table 31: External metrics for mean_centre_median data set using pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.7379 | 0.9731 | 0.8393 |
| QSO | 0.0292 | 0.1000 | 0.0452 |
| STAR | 0.7767 | 0.0958 | 0.1706 |

**v.**

Among all the data set, the transformed PCA performed the best.

The three mean-centred datasets almost have the same internal metrics and external metrics.The recall for GALAXY class is quite high with the value of *0.97*. At the same time it has 0.73 in precision. However they're all not strong in predicting QSA and STAR. From the low recall we can tell that few of the data are predicted as STAR class, but most of them are correct. In terms of QSA, fewer data has been predicted as this class, and conclusively most of them are incorrect.

The dataset after deletion of instances and attributes performs similar as three mean-centred datasets.

PCA and reduced PCA with 12 PCs works very well if we only value the external metrics. They have very similar values with all the criteria above 0.8 apart from the recall for *STAR* class, which is still relatively high compares to the value other datasets produced. The internal metrics have lower dunn value which indicates that the clusters are not well seperated and compact.

# CLASSIFICATION

**1.**

Table 32: The confusion matrix for J48

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5026 | 1 | 0 |
| QSO | 0 | 849 | 1 |
| STAR | 1 | 0 | 4174 |

Table 33: The external metrics for J48

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 1.000 | 1.000 | 1.000 |
| QSO | 0.999 | 0.999 | 0.999 |
| STAR | 1.000 | 1.000 | 1.000 |

Table 34: The confusion matrix for OneR

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5025 | 1 | 1 |
| QSO | 0 | 849 | 1 |
| STAR | 1 | 0 | 4174 |

Table 35: The external metrics for OneR

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 1.000 | 1.000 | 1.000 |
| QSO | 0.999 | 0.999 | 0.999 |
| STAR | 1.000 | 1.000 | 1.000 |

I used reduced PCA with 12 PCs in the classification part. Because it works quite well with pam clustering algorithm. I used the 10 fold cross validation runs 10 times with different training and test data which makes use of all data - this prevents the problem of overfitting. The J48 (C4.5) and OneR are the algorithm produces the best results as. They have consistently perfect values for precision recall and F1, of 1 (with only one incorrect prediction with each class) which again suggests that they are the best approach.

**2.**

Table 36: Different parameters for KNN

| Groups | K | Search_Algorithm | Percentage_split |
|---|---|---|---|
| 1 | 5 | LinearNNSearch | 0.66 |
| 2 | 3 | LinearNNSearch | 0.66 |
| 3 | 5 | KDTree | 0.66 |
| 4 | 3 | KDTree | 0.66 |
| 5 | 5 | LinearNNSearch | 0.80 |
| 6 | 3 | LinearNNSearch | 0.80 |
| 7 | 5 | KDTree | 0.80 |
| 8 | 3 | KDTree | 0.80 |

Table 37: The confusion matrix for Group5

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 960 | 3 | 55 |
| QSO | 22 | 131 | 16 |
| STAR | 130 | 2 | 691 |

Table 38: The external metrics for Group5

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.863 | 0.943 | 0.901 |
| QSO | 0.963 | 0.775 | 0.859 |
| STAR | 0.907 | 0.840 | 0.872 |

Table 39: The confusion matrix for Group7

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 960 | 3 | 55 |
| QSO | 22 | 131 | 16 |
| STAR | 130 | 2 | 691 |

Table 40: The external metrics for Group7

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.863 | 0.943 | 0.901 |
| QSO | 0.963 | 0.775 | 0.859 |
| STAR | 0.907 | 0.840 | 0.872 |

The parameters altered are K to values of 5 and 3, Algorithm to LinearNNsearch and KDtree and the percentage split as 0.66 and 0.88, the varying combinations are evident in the table above. Conclusively the search algorithm only slightly affects the output of the results. Increasing the k values and percentage split improves the results. Therefore, Group 7 and 5 prove to be an optimal combination of parameters.

**3.**

Table 41: The confusion matrix for rpca12

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5026 | 1 | 0 |
| QSO | 0 | 849 | 1 |
| STAR | 1 | 0 | 4174 |

Table 42: The external metrics for rpca12

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 1.000 | 1.000 | 1.000 |
| QSO | 0.999 | 0.999 | 0.999 |

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| STAR | 1.000 | 1.000 | 1.000 |

**v.**

The reduced PCA with 12 PCA's has a good impact on the predictive ability of the algorithm. Both Galaxy and Star score a perfect 1 on precicision, recall and f measure. QSO consistently scores 0.99 on all these 3 areas, proving overall a near perfect result.

# APPENDIX

```
library(dplyr)
library(kableExtra)
library(knitr)
library(cluster.datasets)
library(cluster)
library(e1071)
library(fpc)
library(clv)
library(gridExtra)
library(ggplot2)
library(ggrepel)
```

```
res = read.csv("cw_data.csv", header = TRUE, stringsAsFactors = FALSE)
str(res)
head(res, 10)
```

## ANALYSIS AND PRE-PROCESSING

**1.**

```
nominal = c("objid", "rerun", "run", "native", "field", "specobjid", "plate", "fiberid")
interval = c("mjd")
ratio = c("dia", "ra", "dec", "u", "g", "r", "i", "z", "m_unt", "flux", "redshift")
ordinal = c("camcol")

res_nominal = select(res, nominal)
res_interval = select(res, interval)
res_ratio = select(res, ratio)
res_ordinal = select(res, ordinal)
```

```
Mode = function(x){
    ta = table(x)
    tam = max(ta)
    if (all(ta == tam))
          mod = NA
    else
        if(is.numeric(x))
            mod = as.numeric(names(ta)[ta == tam])
    else
        mod = names(ta)[ta == tam]
    return(mod)
}

mode = function(data){
  freq_table = table(data) # get the frequency table
  maximum = freq_table[which.max(freq_table)] # find which one is the most
  result = row.names(as.data.frame(maximum)) # coerce the data type of maxi to dataframe and t
  return(result)
}

attri = data.frame(Features = colnames(res))
mode = data.frame(Mode = sapply(cbind(res_nominal, res_ratio), mode))
missing = data.frame(Missing = sapply(res, function(x)
  {
      sum(is.na(x))
  }))
mean = data.frame(Mean = sapply(cbind(res_interval, res_ratio), mean, na.rm = TRUE))
median = data.frame(Median = sapply(cbind(res_ordinal, res_ratio), median, na.rm = TRUE))
sd = data.frame(SD = sapply(cbind(res_interval, res_ratio), sd, na.rm = TRUE))
iqr = data.frame(IQR = sapply(cbind(res_interval, res_ratio), IQR, na.rm = TRUE))
range = data.frame(Range = sapply(cbind(res_interval, res_ratio), function(x){
  max(x, na.rm = TRUE)-min(x, na.rm = TRUE)
}))

mode = bind_rows(mode, data.frame(Mode = rep(NA, length(interval)+length(ordinal)), row.names =
mean = bind_rows(mean, data.frame(Mean = rep(NA, length(nominal)+length(ordinal)), row.names =
median = bind_rows(median, data.frame(Median = rep(NA, length(nominal)+length(interval)), row.
sd = bind_rows(sd, data.frame(SD = rep(NA, length(nominal)+length(ordinal)), row.names = c(nom
iqr = bind_rows(iqr, data.frame(IQR = rep(NA, length(nominal)+length(ordinal)), row.names = c(
range = bind_rows(range, data.frame(Range = rep(NA, length(nominal)+length(ordinal)), row.name

mode$Features = rownames(mode)
mean$Features = rownames(mean)
median$Features = rownames(median)
sd$Features = rownames(sd)
iqr$Features = rownames(iqr)
range$Features = rownames(range)
```

```
missing$Features = rownames(missing)

DT = mode %>% merge(mean) %>% merge(median) %>% merge(sd) %>% merge(iqr) %>% merge(range) %>% r
kable(DT, caption = "centrality measure, dispersion measure and missing value of the dataset",
```

**i.**

Table 43: centrality measure, dispersion measure and missing value of the dataset

| Features | Mode | Mean | Median | SD | IQR | Range | Missing |
|---|---|---|---|---|---|---|---|
| camcol | NA | NA | 4.0000 | NA | NA | NA | 50 |
| dec | -0.393345053 | 14.8267 | 0.4029 | 25.2083 | 36.1040 | 73.9249 | 49 |
| dia | 27.54042634 | 2000.0276 | 349.1009 | 22677.7304 | 464.1442 | 848144.2423 | 6646 |
| fiberid | 155 | NA | NA | NA | NA | NA | 32 |
| field | 301 | NA | NA | NA | NA | NA | 50 |
| flux | 189.7145297 | 183.5176 | 183.3246 | 50.2938 | 50.8474 | 309.5857 | 50 |
| g | 17.55623 | 17.3719 | 17.4951 | 0.9455 | 1.1951 | 7.1194 | 51 |
| i | 14.06613 | 16.5839 | 16.5552 | 1.1419 | 1.4048 | 16.2324 | 50 |
| m_unt | 1e-04 | 0.0002 | 0.0002 | 0.0001 | 0.0001 | 0.0004 | 46 |
| mjd | NA | 52944.4652 | NA | 1511.3414 | 2568.0000 | 5903.0000 | 50 |
| native | 1 | NA | NA | NA | NA | NA | 50 |
| objid | 1.24e+18 | NA | NA | NA | NA | NA | 0 |
| plate | 2558 | NA | NA | NA | NA | NA | 50 |
| r | 15.99986 | 16.8408 | 16.8586 | 1.0677 | 1.3392 | 12.3704 | 53 |
| ra | 189.6527639 | 175.5448 | 180.4418 | 47.7724 | 44.1627 | 252.6493 | 48 |
| redshift | 0 | 0.1437 | 0.0425 | 0.3887 | 0.0925 | 5.3580 | 50 |
| rerun | 301 | NA | NA | NA | NA | NA | 30 |
| run | 756 | NA | NA | NA | NA | NA | 50 |
| specobjid | 2.88e+18 | NA | NA | NA | NA | NA | 50 |
| u | 18.90212 | 18.6190 | 18.8528 | 0.8290 | 1.0817 | 6.6109 | 50 |
| z | 13.9471 | 16.4231 | 16.3899 | 1.2035 | 1.5222 | 11.2226 | 53 |

```
class_data = res[,"class"]
class_percentages = aggregate(class_data, by = list(class_data),FUN = function(x){
  round(length(x) * 100/length(class_data),2)
})
class_count = aggregate(class_data, by = list(class_data), length)
colnames(class_percentages) = c("Class", "Percentages")
colnames(class_count) = c("Class", "Count")
class_agg = merge(class_percentages, class_count)
ggplot(class_agg, aes(x = Class, y = Percentages, fill = Class, color = Class))+
  geom_bar(stat = "identity", alpha = .6)+
```

```
  scale_y_continuous(name = "Percentages%")+
  geom_text_repel(aes(label = Count), force = 3)
```

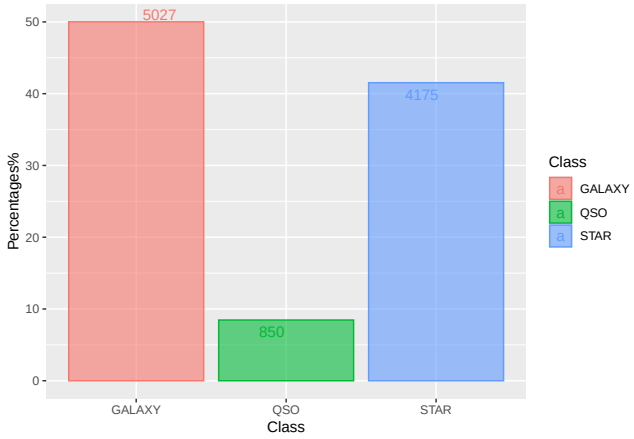


Figure 19: Bar chart of Class

**ii** There are three different classes in this data set. The graph shows that *GALAXY* has the most sample with 5027 in total, which takes up half of the data set. *STAR* has the second most sample with 4175 samples taking up 41.53%. Lastly *QSO* has the least sample taking up only 8.47% with 850 samples. Therefore, this data set is not balanced. It may be biased if accuracy is used to evaluate the training model.

```
res_frame = data.frame(res)
```

```
ggplot(res_frame)+
  geom_histogram(aes(x = dec, fill = class, color = class), binwidth = .5, alpha = .6)+
  scale_x_continuous(breaks = seq(min(res_frame$dec, na.rm = TRUE), max(res_frame$dec,na.rm = T
  theme(axis.text.x = element_text(angle = 75, hjust = 1)) +
  geom_vline(aes(xintercept = median(dec, na.rm=T)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(dec, na.rm=T)),size=1, color="black")
```

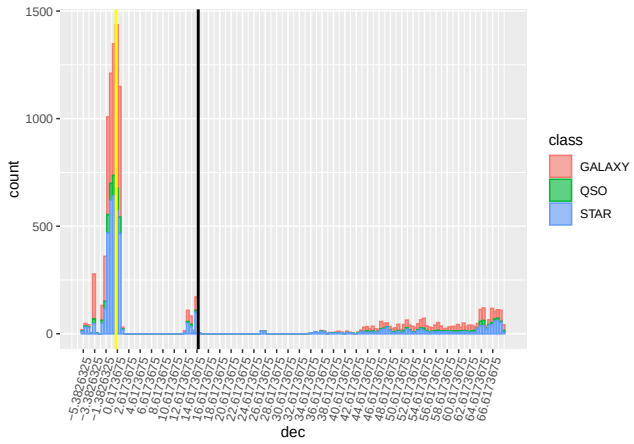


Figure 20: dispersion of dec

```
ggplot(res_frame)+
  geom_histogram(aes(x = dia, fill = class, color = class), binwidth = 17000, alpha = .6)+
  scale_x_continuous(breaks = seq(min(res_frame$dia, na.rm = TRUE), max(res_frame$dia,na.rm = T
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(dia, na.rm=TRUE)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(dia, na.rm=TRUE)),size=1, color="black")
```

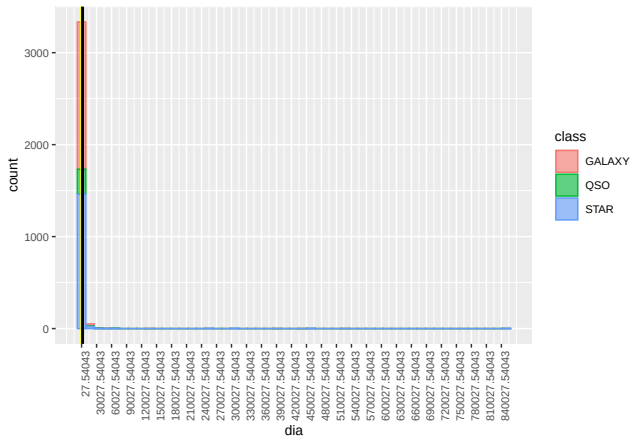


Figure 21: dispersion of dia

```
ggplot(res_frame)+
  geom_histogram(aes(x = camcol, fill = class, color = class), binwidth = .5, alpha = .6)+
  scale_x_continuous(breaks = seq(min(res_frame$camcol, na.rm = TRUE), max(res_frame$camcol,na
  geom_vline(aes(xintercept = median(camcol, na.rm=TRUE)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(camcol, na.rm=TRUE)),size=1, color="black")
```

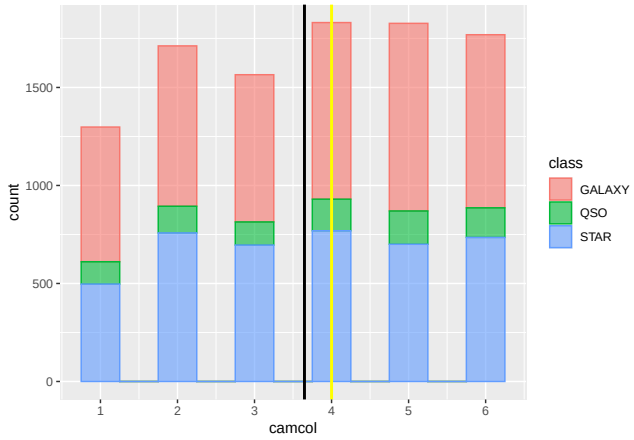


Figure 22: dispersion of camcol

```
ggplot(res_frame)+
  geom_histogram(aes(x = fiberid, color = class, fill = class), binwidth = 10, alpha = .6)+
  scale_x_continuous(breaks = seq(min(res_frame$fiberid, na.rm = TRUE), max(res_frame$fiberid,
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(fiberid, na.rm=TRUE)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(fiberid, na.rm=TRUE)),size=1, color="black")
```

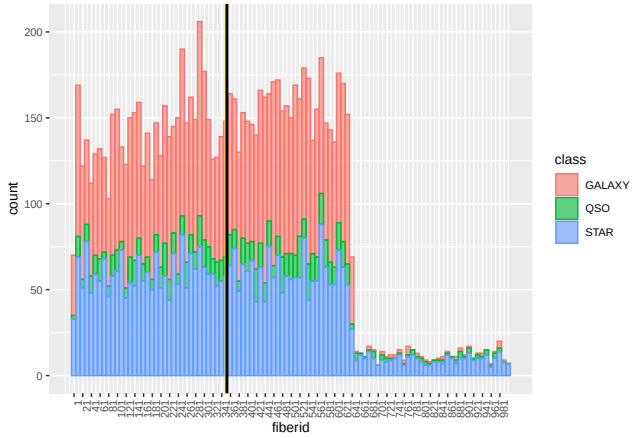


Figure 23: dispersion of fiberid

```
ggplot(res_frame)+
  geom_histogram(aes(x = field, color = class, fill = class), binwidth = 10, alpha = .6)+
  scale_x_continuous(breaks = seq(min(res_frame$field, na.rm = TRUE), max(res_frame$field, na.
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(field, na.rm=TRUE)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(field, na.rm=TRUE)),size=1, color="black")
```

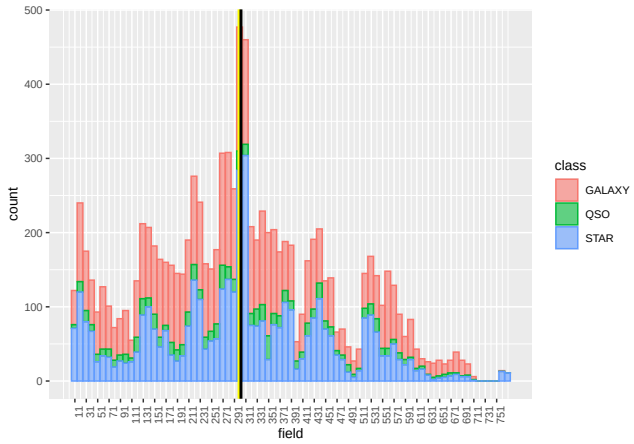


Figure 24: dispersion of field

```
ggplot(res_frame)+
  geom_histogram(aes(x = flux, color = class, fill = class), binwidth = 5, alpha = .6)+
  scale_x_continuous(breaks = seq(min(res_frame$flux, na.rm = TRUE), max(res_frame$flux, na.rm
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(flux, na.rm=TRUE)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(flux, na.rm=TRUE)),size=1, color="black")
```

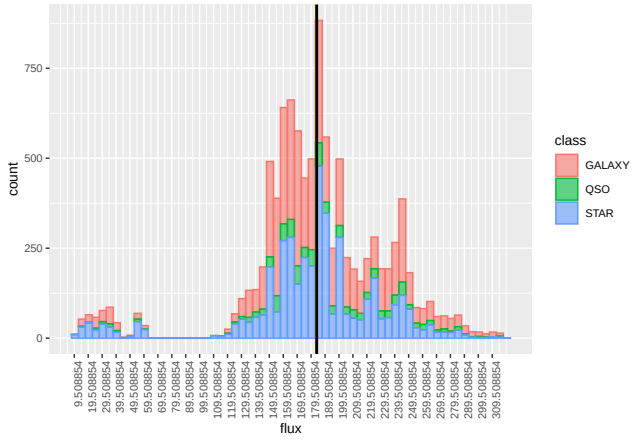


Figure 25: dispersion of flux

```
ggplot(res_frame)+
  geom_histogram(aes(x = g, color = class, fill = class), binwidth = .1, alpha = .6)+
  scale_x_continuous(breaks = seq(min(res_frame$g, na.rm = TRUE), max(res_frame$g,na.rm = TRUE
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(g, na.rm=TRUE)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(g, na.rm=TRUE)),size=1, color="black")
```

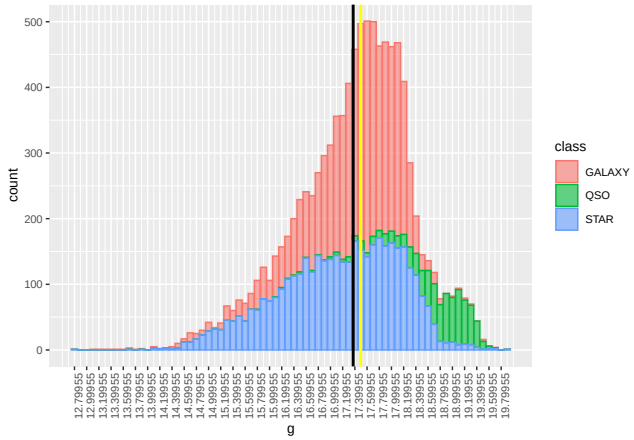


Figure 26: dispersion of g

```
ggplot(res_frame)+
  geom_histogram(aes(x = i, color = class, fill = class), binwidth = .1, alpha = .6)+
  scale_x_continuous(breaks = seq(min(res_frame$i, na.rm = TRUE), max(res_frame$i,na.rm = TRUE
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(i, na.rm=TRUE)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(i, na.rm=TRUE)),size=1, color="black")
```

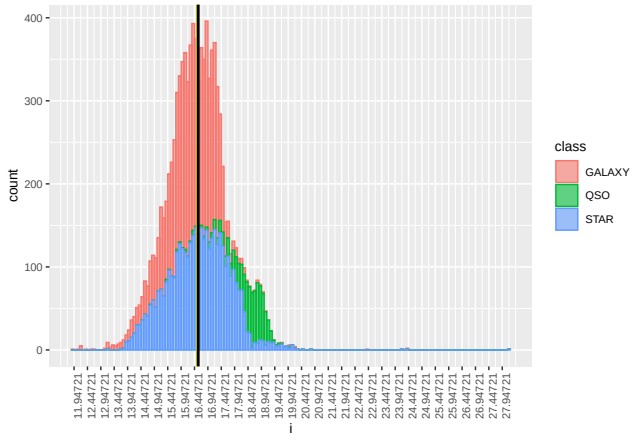


Figure 27: dispersion of i

```
ggplot(res_frame)+
  geom_histogram(aes(x = m_unt, color = class, fill = class), binwidth = .000005, alpha = .6)+
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(m_unt, na.rm=TRUE)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(m_unt, na.rm=TRUE)),size=1, color="black")
```

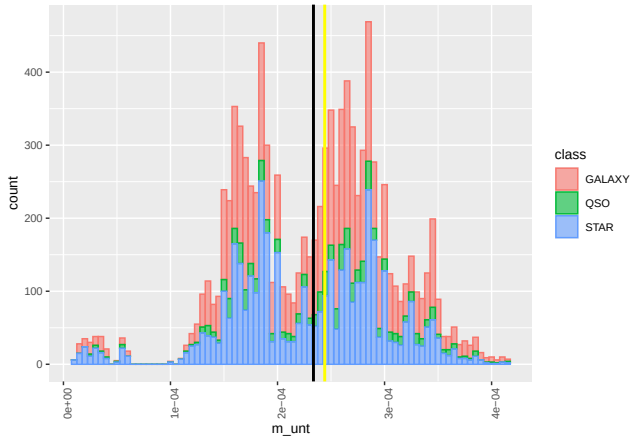


Figure 28: dispersion of m-unt

```
ggplot(res_frame)+
  geom_histogram(aes(x = mjd, color = class, fill = class), binwidth = 50, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$mjd, na.rm = TRUE), max(res_frame$mjd,na.rm =
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(mjd, na.rm=TRUE)),size=1, color="yellow")+
  geom_vline(aes(xintercept = mean(mjd, na.rm=TRUE)),size=1, color="black")
```

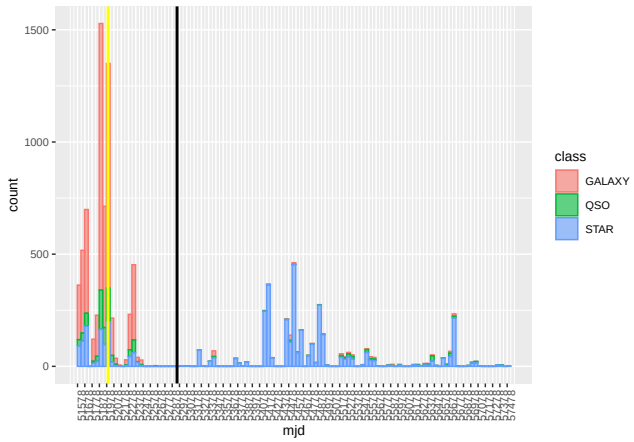


Figure 29: dispersion of mjd

```
ggplot(res_frame)+
  geom_histogram(aes(x = native, color = class, fill = class), binwidth = .01, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$native, na.rm = TRUE), max(res_frame$native,
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(native, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(native, na.rm=TRUE)),size=.5, color="black")
```

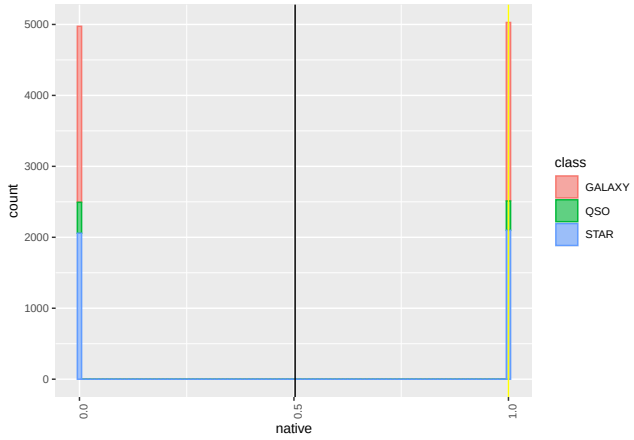


Figure 30: dispersion of native

```
ggplot(res_frame)+
  geom_histogram(aes(x = objid, color = class, fill = class), binwidth = 1, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$objid, na.rm = TRUE), max(res_frame$objid,na
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(objid, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(objid, na.rm=TRUE)),size=.5, color="black")
```


Figure 31: dispersion of objid

```
ggplot(res_frame)+
  geom_histogram(aes(x = plate, color = class, fill = class), binwidth = 100, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$plate, na.rm = TRUE), max(res_frame$plate,na
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(plate, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(plate, na.rm=TRUE)),size=.5, color="black")
```

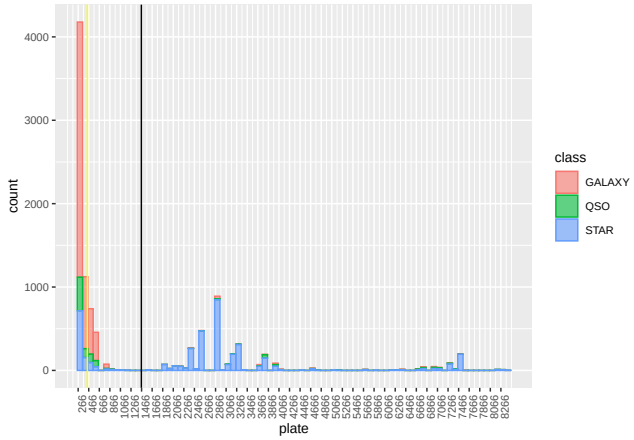


Figure 32: dispersion of plate

```
ggplot(res_frame)+
  geom_histogram(aes(x = r, color = class, fill = class), binwidth = .1, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$r, na.rm = TRUE), max(res_frame$r,na.rm = TRU
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(r, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(r, na.rm=TRUE)),size=.5, color="black")
```

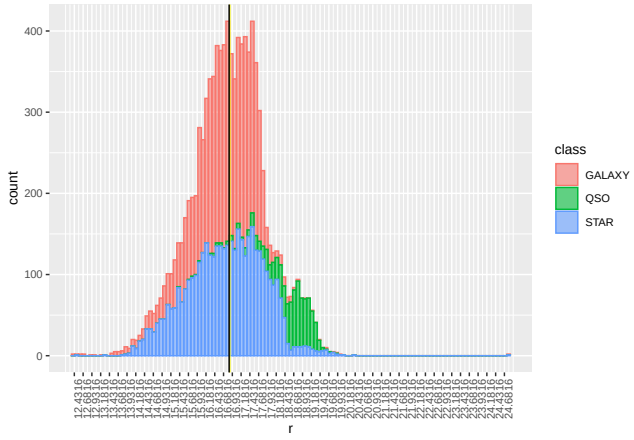


Figure 33: dispersion of r

```
ggplot(res_frame)+
  geom_histogram(aes(x = ra, color = class, fill = class), binwidth = 2, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$ra, na.rm = TRUE), max(res_frame$ra,na.rm =
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(ra, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(ra, na.rm=TRUE)),size=.5, color="black")
```

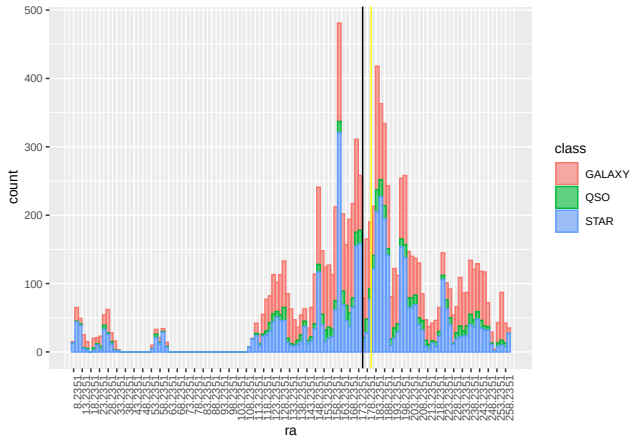


Figure 34: dispersion of ra

```
ggplot(res_frame)+
  geom_histogram(aes(x = redshift, color = class, fill = class), binwidth = .1, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$redshift, na.rm = TRUE), max(res_frame$redshi
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(redshift, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(redshift, na.rm=TRUE)),size=.5, color="black")
```

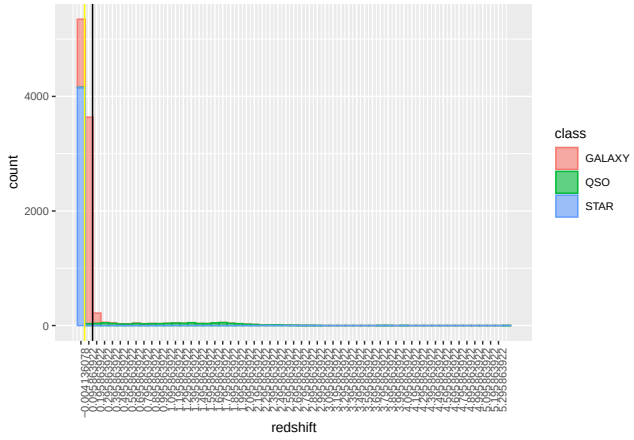


Figure 35: dispersion of redshift

```
ggplot(res_frame)+
  geom_histogram(aes(x = rerun, color = class, fill = class), binwidth = 1, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$rerun, na.rm = TRUE), max(res_frame$rerun,na
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(rerun, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(rerun, na.rm=TRUE)),size=.5, color="black")
```

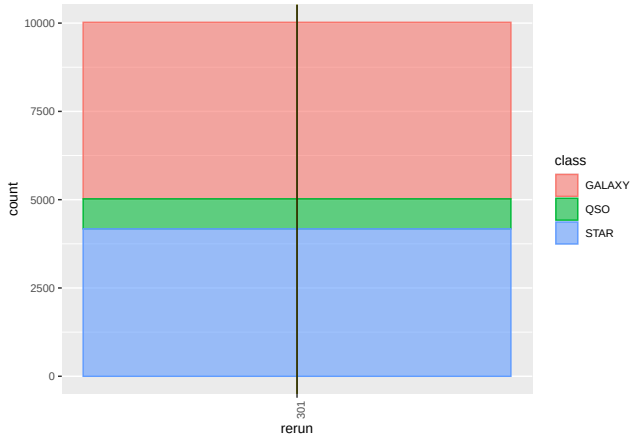


Figure 36: dispersion of rerun

```
ggplot(res_frame)+
  geom_histogram(aes(x = run, color = class, fill = class), binwidth = 10, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$run, na.rm = TRUE), max(res_frame$run,na.rm =
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(run, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(run, na.rm=TRUE)),size=.5, color="black")
```

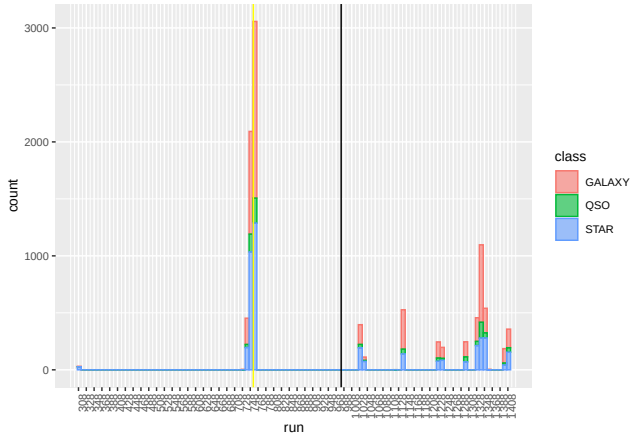


Figure 37: dispersion of run

```
ggplot(res_frame)+
  geom_histogram(aes(x = specobjid, color = class, fill = class), binwidth = 50000, alpha = .6)
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(specobjid, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(specobjid, na.rm=TRUE)),size=.5, color="black")
```

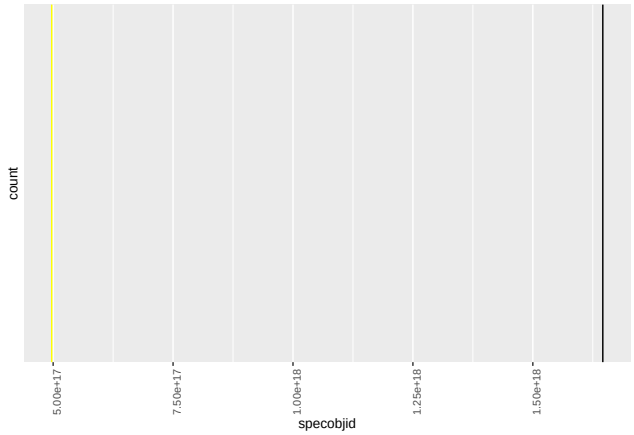


Figure 38: dispersion of specobjid

```
ggplot(res_frame)+
  geom_histogram(aes(x = u, color = class, fill = class), binwidth = .1, alpha = .6)+
    scale_x_continuous(breaks = seq(min(res_frame$u, na.rm = TRUE), max(res_frame$u,na.rm = TRU
  theme(text = element_text(size = 10),axis.text.x = element_text(angle = 90, hjust = 1, )) +
  geom_vline(aes(xintercept = median(u, na.rm=TRUE)),size=.5, color="yellow")+
  geom_vline(aes(xintercept = mean(u, na.rm=TRUE)),size=.5, color="black")
```

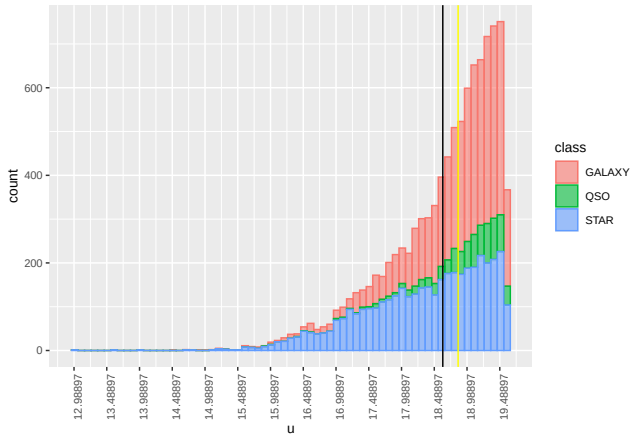


Figure 39: dispersion of u

**iii.**

**2.**

```
correlation = cor(res_frame$r, res_frame$g, method ="pearson", use = "pairwise.complete.obs")
```

```
ggplot(res_frame, aes(x = r, y = g))+
  geom_point(aes(color = class), size = 1)
```

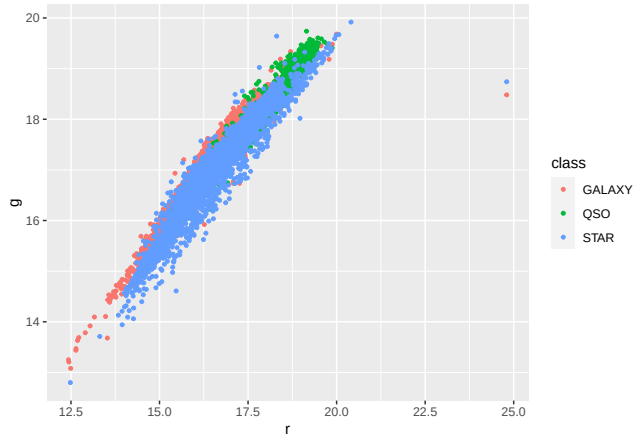


Figure 40: correlation between r and g

**i.**

```
correlation = cor(res_frame$r, res_frame$mjd, method ="pearson", use = "pairwise.complete.obs")

ggplot(res_frame, aes(x = r, y = mjd))+
  geom_point(aes(color = class), size = 1)
```

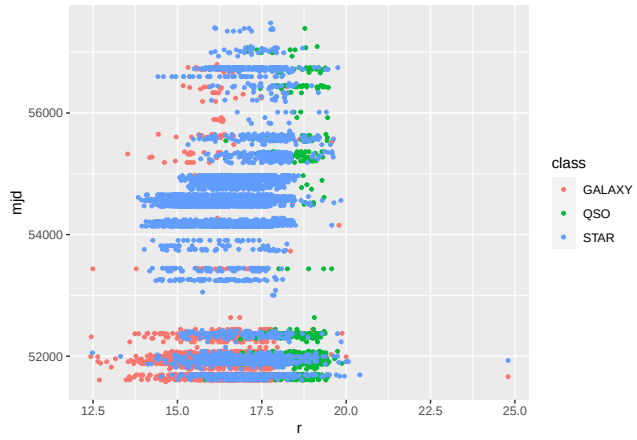


Figure 41: correlation between r and mjd

**ii.**

```
ggplot(res_frame, aes(x = class, y = z))+
  geom_point(aes(color = class), size = 1)
```

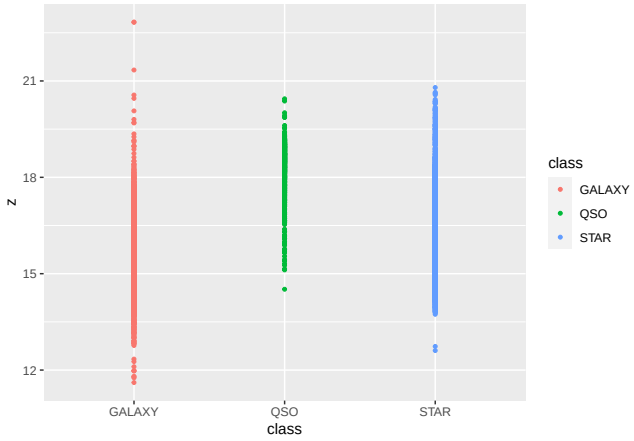


Figure 42: scatterplot between class and z

```
ggplot(res_frame, aes(x = class, y = u))+
  geom_point(aes(color = class), size = 1)
```

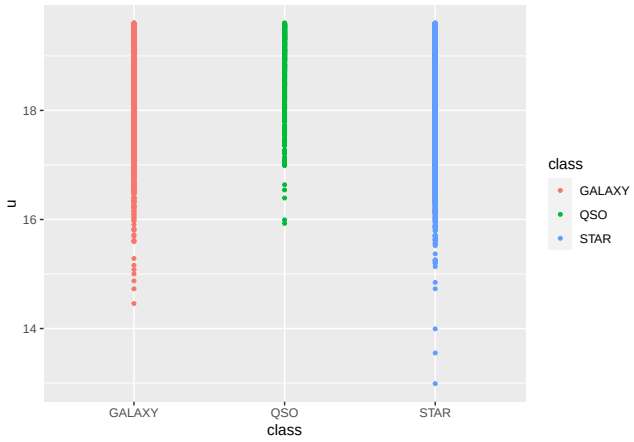


Figure 43: scatterplot between class and u

```
ggplot(res_frame, aes(x = class, y = redshift))+
  geom_point(aes(color = class), size = 1)
```

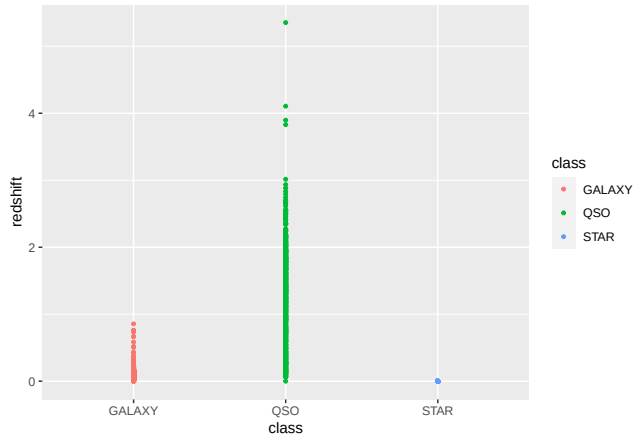


Figure 44: scatterplot between class and redshift

**iii.**

```
ggplot(res_frame, aes(x = class, y = redshift, color = class)) +
    geom_boxplot()
```

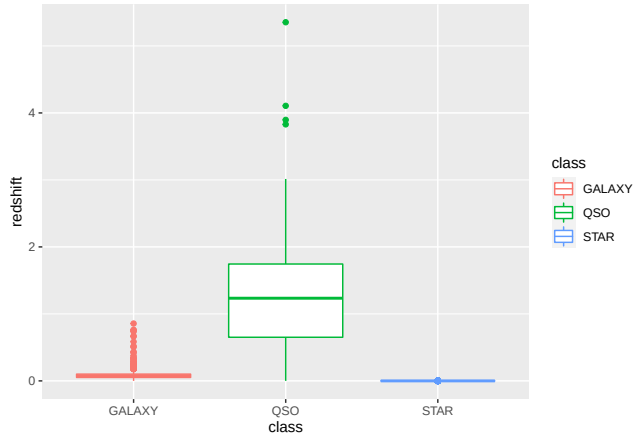


Figure 45: Boxplots of redshift from different classes

```
ggplot(res_frame, aes(x = class, y = mjd, color = class)) +
    geom_boxplot()
```

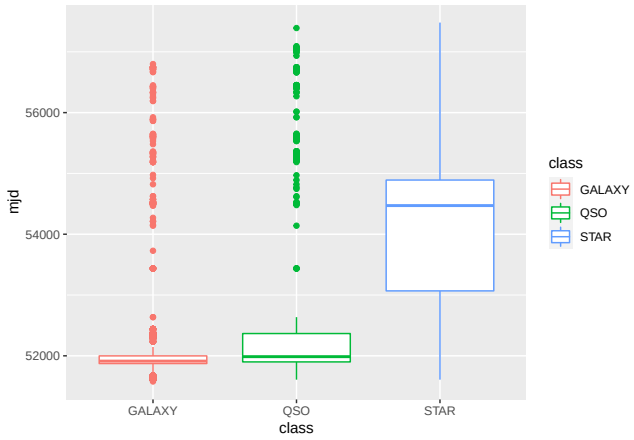


Figure 46: Boxplots of mjd from different classes

```
ggplot(res_frame, aes(x = class, y = dia, color = class)) +
    geom_boxplot()
```

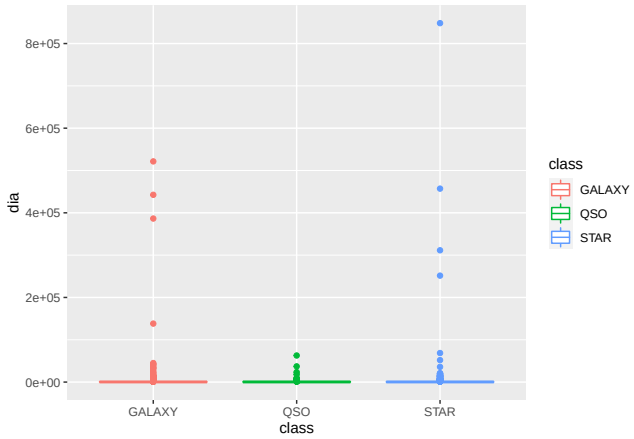


Figure 47: Boxplots of dia from different classes

```
ggplot(res_frame, aes(x = class, y = ra, color = class)) +
    geom_boxplot()
```

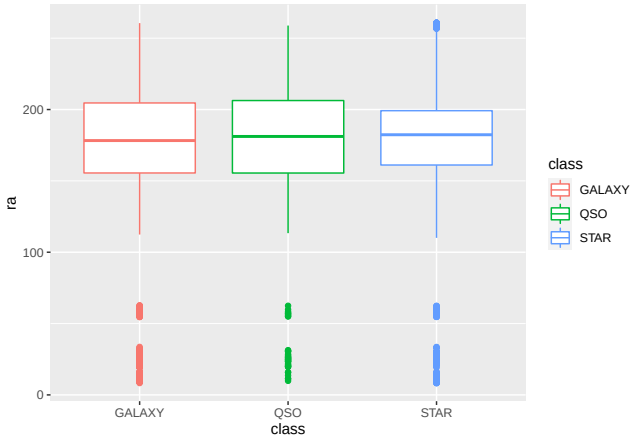


Figure 48: Boxplots of ra from different classes

```
ggplot(res_frame, aes(x = class, y = dec, color = class)) +
    geom_boxplot()
```

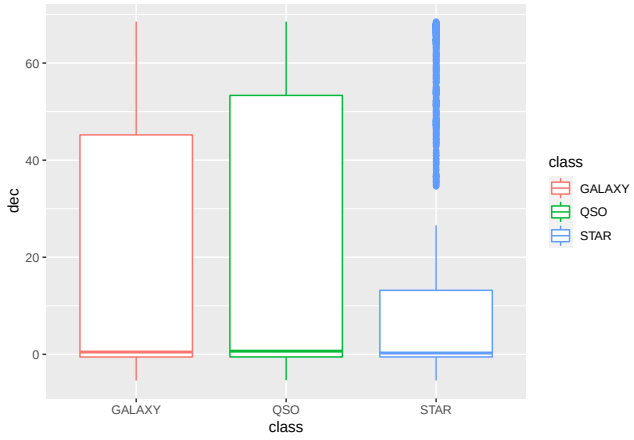


Figure 49: Boxplots of dec from different classes

```
ggplot(res_frame, aes(x = class, y = u, color = class)) +
    geom_boxplot()
```

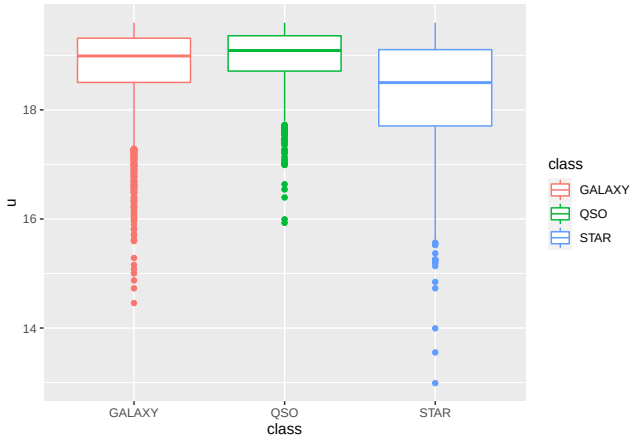


Figure 50: Boxplots of u from different classes

```
ggplot(res_frame, aes(x = class, y = g, color = class)) +
    geom_boxplot()
```

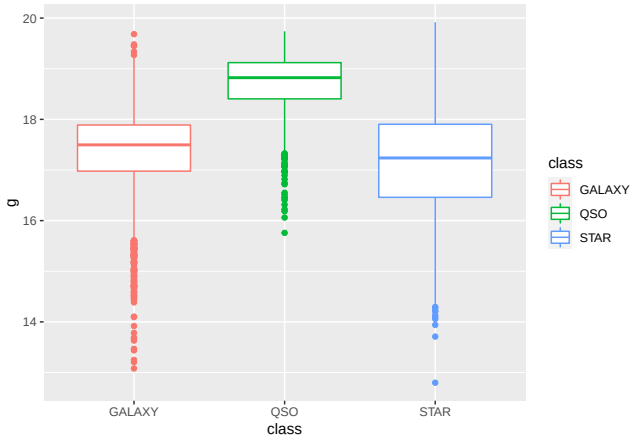


Figure 51: Boxplots of g from different classes

```
ggplot(res_frame, aes(x = class, y = r, color = class)) +
    geom_boxplot()
```

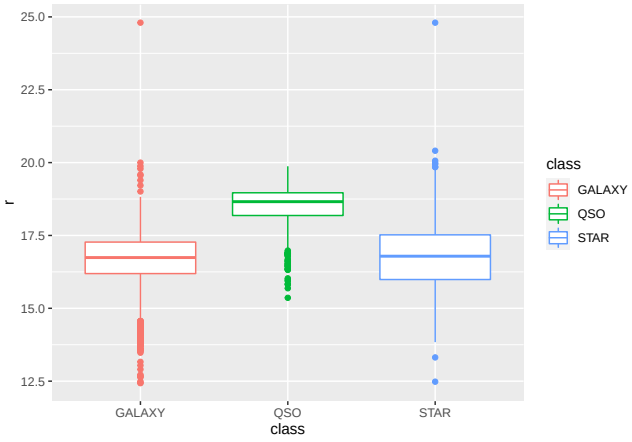


Figure 52: Boxplots of r from different classes

```
ggplot(res_frame, aes(x = class, y = i, color = class)) +
    geom_boxplot()
```

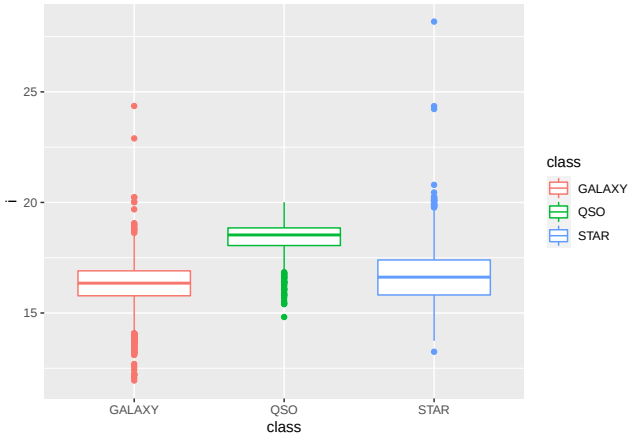


Figure 53: Boxplots of i from different classes

```
ggplot(res_frame, aes(x = class, y = z, color = class)) +
    geom_boxplot()
```

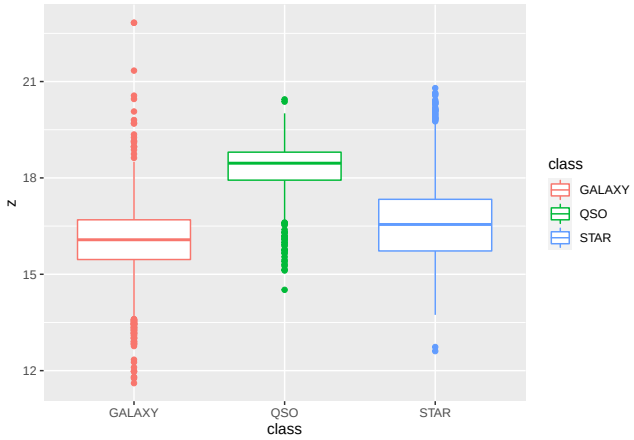


Figure 54: Boxplots of z from different classes

```
ggplot(res_frame, aes(x = class, y = m_unt, color = class)) +
    geom_boxplot()
```

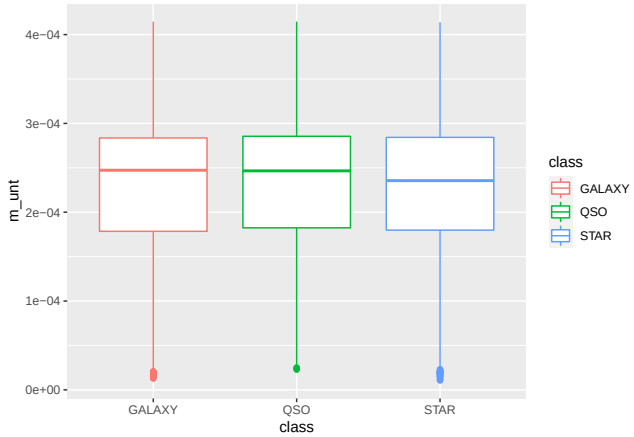


Figure 55: Boxplots of m-unt from different classes

```
ggplot(res_frame, aes(x = class, y = flux, color = class)) +
    geom_boxplot()
```

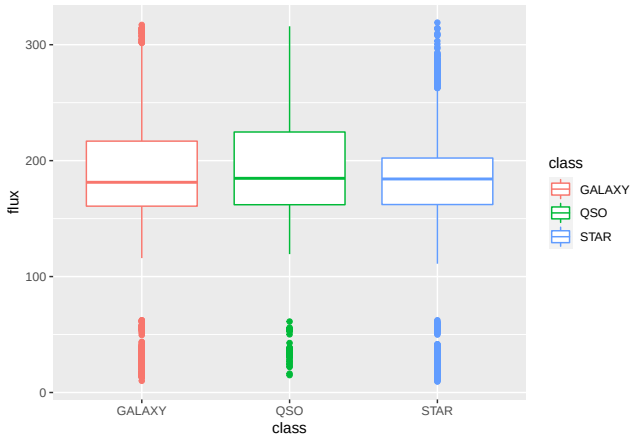


Figure 56: Boxplots of flux from different classes

**iv.**

**4.**

```
galaxy = select(filter(res_frame, class == "GALAXY"), -"class")
star = select(filter(res_frame, class == "STAR"), -"class")
qso = select(filter(res_frame, class == "QSO"), -"class")

galaxy_mean = sapply(galaxy, mean, na.rm = TRUE)
galaxy_median = sapply(galaxy, median, na.rm = TRUE)
star_mean = sapply(star, mean, na.rm = TRUE)
star_median = sapply(star, median, na.rm = TRUE)
qso_mean = sapply(qso, mean, na.rm = TRUE)
qso_median = sapply(qso, median, na.rm = TRUE)

zero_na = res_frame
mean_na = res_frame
median_na = res_frame

for (i in colnames(zero_na))
  {
  zero_na[is.na(zero_na[,i]),i]=0
  }

for (i in colnames(zero_na))
  {
    mean_na[is.na(mean_na[,i]) & mean_na$class == "GALAXY",i] = galaxy_mean[i]
    median_na[is.na(median_na[,i]) & median_na$class == "GALAXY",i] = galaxy_median[i]
  }
  for (i in colnames(zero_na))
  {
```

```
    mean_na[is.na(mean_na[,i]) & mean_na$class == "QSO",i] = qso_mean[i]
    median_na[is.na(median_na[,i]) & median_na$class == "QSO",i] = qso_median[i]
  }
  for (i in colnames(zero_na))
  {
    mean_na[is.na(mean_na[,i]) & mean_na$class == "STAR",i] = star_mean[i]
    median_na[is.na(median_na[,i]) & median_na$class == "STAR",i] = star_median[i]
  }
```

```
boxplot(select(res_frame, -class))
```

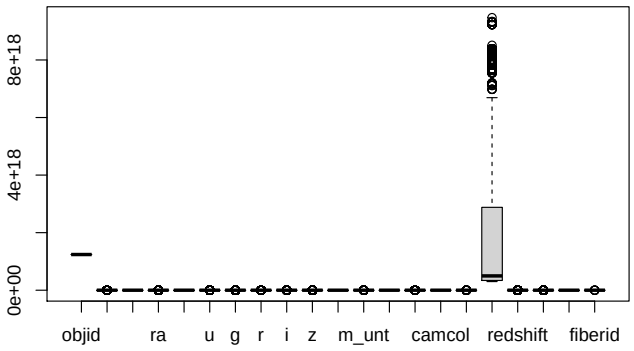


Figure 57: Boxplots of original dataset

```
boxplot(select(zero_na, -class))
```

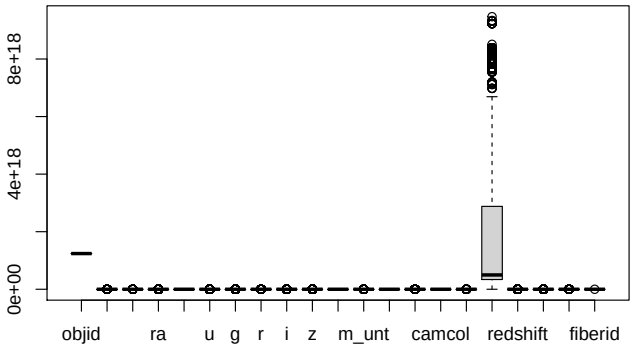


Figure 58: Boxplots of the data set replaced by zero value

```
boxplot(select(mean_na, -class))
```

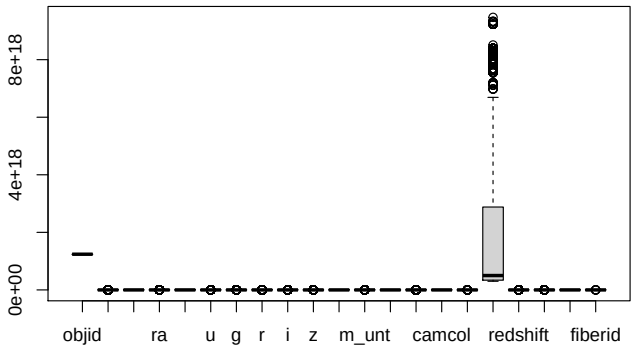


Figure 59: Boxplots of the data set replaced by mean value

```
boxplot(select(median_na, -class))
```

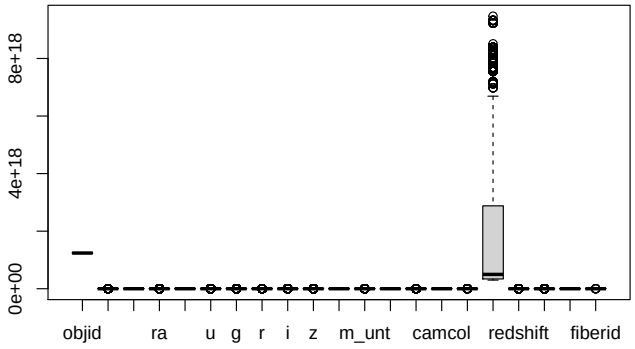


Figure 60: Boxplots of the data set replaced by median value

**5.**

```
normalize = function(x) {
return ((x - min(x)) / (max(x) - min(x)))
}

mean_centre_zero = data.frame(scale(select(zero_na, -"class"), scale = FALSE))
mean_centre_mean = data.frame(scale(select(mean_na, -"class"), scale = FALSE))
mean_centre_median = data.frame(scale(select(median_na, -"class"), scale = FALSE))

normalised_zero = data.frame(sapply(select(zero_na, -"class"), normalize))
normalised_mean = data.frame(sapply(select(mean_na, -"class"), normalize))
normalised_median = data.frame(sapply(select(median_na, -"class"), normalize))

standarised_zero = data.frame(scale(select(zero_na, -"class")))
```

```
standarised_mean = data.frame(scale(select(mean_na, -"class")))
standarised_median = data.frame(scale(select(median_na, -"class")))
```

```
mjd_mc_z = data.frame(Mjd = mean_centre_zero$mjd, Measure = "mean centre")
mjd_n_z = data.frame(Mjd = normalised_zero$mjd, Measure = "normalised")
mjd_s_z = data.frame(Mjd = standarised_zero$mjd, Measure = "standarised")
mjd_zero = bind_rows(mjd_mc_z,mjd_n_z,mjd_s_z)

mjd_mc_mean = data.frame(Mjd = mean_centre_mean$mjd, Measure = "mean centre")
mjd_n_mean = data.frame(Mjd = normalised_mean$mjd, Measure = "normalised")
mjd_s_mean = data.frame(Mjd = standarised_mean$mjd, Measure = "standarised")
mjd_mean = bind_rows(mjd_mc_mean,mjd_n_mean,mjd_s_mean)

mjd_mc_median = data.frame(Mjd = mean_centre_median$mjd, Measure = "mean centre")
mjd_n_median = data.frame(Mjd = normalised_median$mjd, Measure = "normalised")
mjd_s_median = data.frame(Mjd = standarised_median$mjd, Measure = "standarised")
mjd_median = bind_rows(mjd_mc_median,mjd_n_median,mjd_s_median)
```

```
ggplot(mjd_zero, color = Measure)+
  geom_boxplot(aes(y = Mjd, x = Measure ))
```

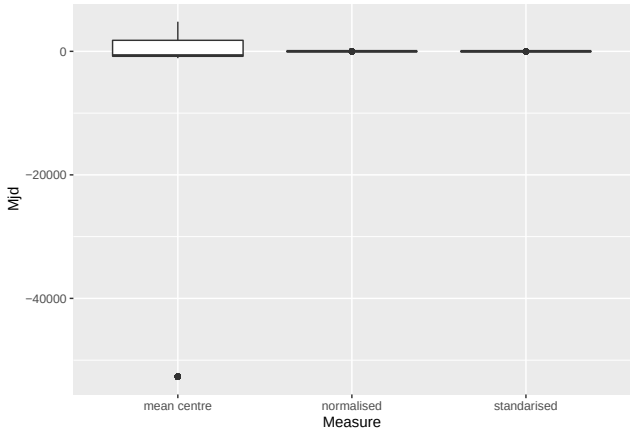


Figure 61: Boxplots of mjd attribute in data set replaced by zero

```
ggplot(mjd_mean, color = Measure)+
  geom_boxplot(aes(y = Mjd, x = Measure ))
```

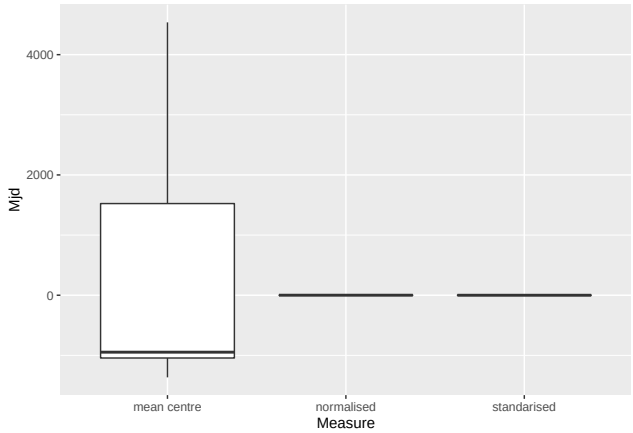


Figure 62: Boxplots of mjd attribute in data set replaced by mean value

```
ggplot(mjd_median, color = Measure)+
  geom_boxplot(aes(y = Mjd, x = Measure ))
```

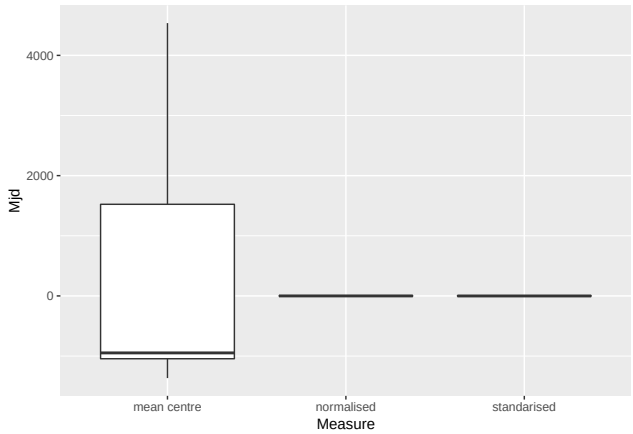


Figure 63: Boxplots of mjd attribute in data set replaced by median value

**6.**

```
res_noclass = select(res_frame, -class)
missing_percentages = data.frame(Features = colnames(res_noclass) , Percentages = sapply(res_n
  sum(is.na(x)) * 100/length(x)
}))
rownames(missing_percentages) = c()
index = apply(res_noclass, 1, function(x){
   sum(is.na(x))/length(x)<0.5
})
```

```
deleting_rows = res_frame[index,]

DF1 = select(deleting_rows, -dia)

kable(missing_percentages, caption = "The percentages of different attribute's missing value",
```

**i.**

Table 44: The percentages of different attribute's missing value

| Features | Percentages |
|---|---|
| objid | 0.0000 |
| dia | 66.1162 |
| rerun | 0.2984 |
| ra | 0.4775 |
| dec | 0.4875 |
| u | 0.4974 |
| g | 0.5074 |
| r | 0.5273 |
| i | 0.4974 |
| z | 0.5273 |
| run | 0.4974 |
| m_unt | 0.4576 |
| native | 0.4974 |
| flux | 0.4974 |
| camcol | 0.4974 |
| field | 0.4974 |
| specobjid | 0.4974 |
| redshift | 0.4974 |
| plate | 0.4974 |
| mjd | 0.4974 |
| fiberid | 0.3183 |

```
replace_missing = function(x){
  galaxy = select(filter(x, class == "GALAXY"), -"class")
  star = select(filter(x, class == "STAR"), -"class")
  qso = select(filter(x, class == "QSO"), -"class")

  galaxy_mean = sapply(galaxy, mean, na.rm = TRUE)
  star_mean = sapply(star, mean, na.rm = TRUE)
  qso_mean = sapply(qso, mean, na.rm = TRUE)

  replace_mean = x
```

```
  for (i in colnames(replace_mean))
    {
      replace_mean[is.na(replace_mean[,i]) & replace_mean$class == "GALAXY",i] = galaxy_mean[i]
    }
    for (i in colnames(replace_mean))
    {
      replace_mean[is.na(replace_mean[,i]) & replace_mean$class == "QSO",i] = qso_mean[i]
    }
    for (i in colnames(replace_mean))
    {
      replace_mean[is.na(replace_mean[,i]) & replace_mean$class == "STAR",i] = star_mean[i]
    }
  return(replace_mean)
}

replace_mean = replace_missing(res_frame)
#sapply(select(replace_mean, -class), function(x){sum(is.na(x))})

#data.frame(cor(select(replace_mean, -class), method = "pearson", use = "pairwise.complete.obs
```

```
uncorrelated_DS = select(replace_mean, -class)
uncorrelated_DS = select(uncorrelated_DS, c(-objid, -rerun, -g, -r, -i, -z, -ra, -specobjid, -p
correlation_mat = cor(uncorrelated_DS, method = "pearson", use = "pairwise.complete.obs")
kable(correlation_mat, caption = "The correlation matrix of the data set been produced", align
```

**ii.**

Table 45: The correlation matrix of the data set been produced

| | dia | u | native | flux | redshift | mjd | fiberid |
|---|---|---|---|---|---|---|---|
| dia | 1.0000 | -0.0069 | 0.0015 | 0.0180 | -0.0185 | 0.0226 | 0.0072 |
| u | -0.0069 | 1.0000 | -0.0240 | 0.0364 | 0.1639 | -0.1698 | 0.0125 |
| native | 0.0015 | -0.0240 | 1.0000 | -0.0065 | -0.0130 | -0.0002 | -0.0222 |
| flux | 0.0180 | 0.0364 | -0.0065 | 1.0000 | 0.0481 | -0.0654 | 0.0956 |
| redshift | -0.0185 | 0.1639 | -0.0130 | 0.0481 | 1.0000 | -0.0583 | 0.0471 |
| mjd | 0.0226 | -0.1698 | -0.0002 | -0.0654 | -0.0583 | 1.0000 | 0.1882 |
| fiberid | 0.0072 | 0.0125 | -0.0222 | 0.0956 | 0.0471 | 0.1882 | 1.0000 |

**7.**

```
autometic_pca = function(dataset, threshold, outPutData)
{
  #normalize the dataset
  t.stand = scale(dataset)
  pca_t = prcomp(t.stand)
  sum = summary(pca_t)
  select =sapply(sum$importance["Cumulative Proportion",], FUN = function(x){
    if (x<= threshold)
    {
      return(TRUE)
    }
    else{
      return(FALSE)
    }
  })
  select[select == FALSE][1] = TRUE
  if(outPutData){
      return(pca_t$x[,select])
  }else{
      return(sum$importance[,select])
  }
}
```

```
original = replace_missing(res_frame)
original_input = select(original, c(-class, -objid, -rerun, - objid, -specobjid, -fiberid))
tpca = autometic_pca(original_input, 1, TRUE)
rpca_12 = tpca[,1:12]
```

```
summary(tpca)
```

**i.**

```
##       PC1                PC2                PC3                PC4
##  Min.   :-9.13431   Min.   :-4.29849   Min.   :-2.1107   Min.   :-2.672630
##  1st Qu.:-1.29791   1st Qu.:-0.95973   1st Qu.:-1.2730   1st Qu.:-0.951459
##  Median :-0.07257   Median :-0.01074   Median :-0.1747   Median :-0.000748
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.000000
##  3rd Qu.: 1.25946   3rd Qu.: 0.78403   3rd Qu.: 0.8474   3rd Qu.: 0.912448
##  Max.   : 9.01051   Max.   : 6.41698   Max.   : 6.2323   Max.   : 3.882971
##       PC5                PC6                PC7                PC8
##  Min.   :-48.0495   Min.   :-41.58374   Min.   :-1.77969   Min.   :-1.4441
##  1st Qu.: -0.6344   1st Qu.: -0.76708   1st Qu.:-0.90052   1st Qu.:-0.5374
##  Median : -0.1816   Median : -0.00831   Median : 0.09891   Median :-0.2458
```

```
##  Mean   :  0.0000   Mean   :  0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:  0.6579   3rd Qu.:  0.76182   3rd Qu.: 0.74306   3rd Qu.: 0.2269
##  Max.   :  1.6433   Max.   :  2.05017   Max.   : 2.54579   Max.   :12.3272
##       PC9                PC10                PC11                PC12
##  Min.   :-2.2895   Min.   :-6.85479   Min.   :-1.5477   Min.   :-1.95580
##  1st Qu.:-0.4124   1st Qu.:-0.40026   1st Qu.:-0.4492   1st Qu.:-0.23203
##  Median : 0.0318   Median : 0.01659   Median :-0.1185   Median :-0.03011
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.5093   3rd Qu.: 0.42171   3rd Qu.: 0.5748   3rd Qu.: 0.20806
##  Max.   : 1.6551   Max.   : 3.77320   Max.   : 1.3624   Max.   : 1.68889
##       PC13                PC14                PC15
##  Min.   :-0.887827   Min.   :-5.891684   Min.   :-7.390944
##  1st Qu.:-0.087317   1st Qu.:-0.055332   1st Qu.:-0.040631
##  Median :-0.005398   Median : 0.008393   Median :-0.004313
##  Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.000000
##  3rd Qu.: 0.123448   3rd Qu.: 0.063255   3rd Qu.: 0.037681
##  Max.   : 0.677718   Max.   : 1.605095   Max.   : 1.229923
##       PC16                PC17
##  Min.   :-0.468730   Min.   :-0.907554
##  1st Qu.:-0.014487   1st Qu.:-0.018050
##  Median : 0.001073   Median :-0.000874
##  Mean   : 0.000000   Mean   : 0.000000
##  3rd Qu.: 0.015560   3rd Qu.: 0.018341
##  Max.   : 0.503294   Max.   : 3.963533
```

```
summary(rpca_12)
```

```
##       PC1                 PC2                 PC3                PC4
##  Min.   :-9.13431   Min.   :-4.29849   Min.   :-2.1107   Min.   :-2.672630
##  1st Qu.:-1.29791   1st Qu.:-0.95973   1st Qu.:-1.2730   1st Qu.:-0.951459
##  Median :-0.07257   Median :-0.01074   Median :-0.1747   Median :-0.000748
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.000000
##  3rd Qu.: 1.25946   3rd Qu.: 0.78403   3rd Qu.: 0.8474   3rd Qu.: 0.912448
##  Max.   : 9.01051   Max.   : 6.41698   Max.   : 6.2323   Max.   : 3.882971
##       PC5                 PC6                 PC7                PC8
##  Min.   :-48.0495   Min.   :-41.58374   Min.   :-1.77969   Min.   :-1.4441
##  1st Qu.: -0.6344   1st Qu.: -0.76708   1st Qu.:-0.90052   1st Qu.:-0.5374
##  Median : -0.1816   Median : -0.00831   Median : 0.09891   Median :-0.2458
##  Mean   :  0.0000   Mean   :  0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:  0.6579   3rd Qu.:  0.76182   3rd Qu.: 0.74306   3rd Qu.: 0.2269
##  Max.   :  1.6433   Max.   :  2.05017   Max.   : 2.54579   Max.   :12.3272
##       PC9                PC10                PC11                PC12
##  Min.   :-2.2895   Min.   :-6.85479   Min.   :-1.5477   Min.   :-1.95580
##  1st Qu.:-0.4124   1st Qu.:-0.40026   1st Qu.:-0.4492   1st Qu.:-0.23203
##  Median : 0.0318   Median : 0.01659   Median :-0.1185   Median :-0.03011
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.5093   3rd Qu.: 0.42171   3rd Qu.: 0.5748   3rd Qu.: 0.20806
##  Max.   : 1.6551   Max.   : 3.77320   Max.   : 1.3624   Max.   : 1.68889
```

```
rpca_90 = autometic_pca(original_input, 0.9, TRUE)
rpca_90_sum = autometic_pca(original_input, 0.9, FALSE)
kable(rpca_90_sum, caption = "The PCA obtained a cumulative variance of at least 90%", align =
```

**ii.**

Table 46: The PCA obtained a cumulative variance of at least 90%

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | PC8 |
|---|---|---|---|---|---|---|---|---|
| Standard deviation | 2.1486 | 1.7315 | 1.4697 | 1.3535 | 1.0006 | 0.9992 | 0.9803 | 0.9331 |
| Proportion of Variance | 0.2716 | 0.1764 | 0.1270 | 0.1078 | 0.0589 | 0.0587 | 0.0565 | 0.0512 |
| Cumulative Proportion | 0.2716 | 0.4479 | 0.5750 | 0.6828 | 0.7416 | 0.8004 | 0.8569 | 0.9081 |

## CLUSTERING

**1.**

```
cresult = data.frame(class = res_frame$class, hca = 0, kmeans = 0, pam = 0)
k = 3
maximize_diag = function(matrix) {
    if (nrow(matrix) != ncol(matrix)) stop("Not diagonal")
    if(is.null(rownames(matrix))) rownames(matrix) <- 1:nrow(matrix)

    row.max = apply(matrix,1,which.max)

    return(matrix[names(sort(row.max)),])
}
```

```
hc = hclust(dist(rpca_12))
cresult$hca = cutree(hc,k)

km = kmeans(rpca_12, 3)
cresult$kmeans = km$cluster

pam = pam(rpca_12, 3)
cresult$pam = pam$cluster
```

```
internal_matric = function(x, dataset)
{
  distance = dist(dataset)
  if(ncol(x)>2)
    {
```

```
    all_cresult = x[,c(2:ncol(x))]
    summ=sapply(all_cresult, FUN = function(x){cluster.stats(distance,clustering = x, silhouett
    total = as.data.frame(summ[c("average.between","average.within", "dunn"),])
  }else{
    all_cresult = x[,2]
    summ=cluster.stats(distance,clustering = all_cresult, silhouette = TRUE)
    total = as.data.frame(summ[c("average.between","average.within", "dunn")])
  }
  return(total)
}
```

```
inter_matr = internal_matric(cresult, rpca_12)

kable(inter_matr, caption = "Internal metrics for clustering result", align = "c", digits = 4,
```

Table 47: Internal metrics for clustering result

| | hca | kmeans | pam |
|---|---|---|---|
| average.between | 35.19924 | 5.973809 | 6.016513 |
| average.within | 5.369524 | 4.671595 | 4.804679 |
| dunn | 0.4501908 | 0.002569826 | 0.006270207 |

```
hca_table = table(cresult$class,cresult$hca)
km_table = table(cresult$class,cresult$kmeans)
pam_table = table(cresult$class,cresult$pam)

hca_external  = maximize_diag(hca_table)
km_external  = maximize_diag(km_table)
pam_external  = maximize_diag(pam_table)

kable(hca_external, caption = "External metrics for HCA clustering result", align = "c", digits
```

Table 48: External metrics for HCA clustering result

| | 1 | 2 | 3 |
|---|---|---|---|
| GALAXY | 5024 | 3 | 0 |
| QSO | 850 | 0 | 0 |
| STAR | 4171 | 3 | 1 |

```
kable(km_external, caption = "External metrics for kmeans clustering result", align = "c", digi
```

Table 49: External metrics for kmeans clustering result

| | 1 | 2 | 3 |
|---|---|---|---|
| STAR | 1706 | 1250 | 1219 |
| QSO | 11 | 759 | 80 |
| GALAXY | 1144 | 183 | 3700 |

```
kable(pam_external, caption = "External metrics for PAM clustering result", align = "c", digits
```

Table 50: External metrics for PAM clustering result

| | 1 | 2 | 3 |
|---|---|---|---|
| GALAXY | 4858 | 134 | 35 |
| STAR | 948 | 3128 | 99 |
| QSO | 112 | 38 | 700 |

```
precision = function(x)
{
  diag= diag(x)
  colsums=apply(x, 2, sum)
  precision = diag / colsums
  return(precision)
}
recall = function(x){
  diag= diag(x)
  rowsums = apply(x, 1, sum)
  recall = diag / rowsums
  return(recall)
}
f1 = function(precision, recall)
{
  f1 = 2 * precision * recall / (precision + recall)
  return(f1)
}
```

```
hca_precision = precision(hca_external)
hca_recall = recall(hca_external)
hca_f1 = f1(hca_precision, hca_recall)

HCA_evaluate = data.frame(Class = names(hca_recall), Precision = hca_precision, Recall = hca_re

km_precision = precision(km_external)
km_recall = recall(km_external)
km_f1 = f1(km_precision, km_recall)
```

```
km_evaluate = data.frame(Class = names(km_recall), Precision = km_precision, Recall = km_recall

pam_precision = precision(pam_external)
pam_recall = recall(pam_external)
pam_f1 = f1(pam_precision, pam_recall)

pam_evaluate = data.frame(Class = names(pam_recall), Precision = pam_precision, Recall = pam_re
```

```
kable(HCA_evaluate, caption = "External metrics for HCA clustering method", align = "c", digits
```

Table 51: External metrics for HCA clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5001 | 0.9994 | 0.6667 |
| QSO | 0.0000 | 0.0000 | NaN |
| STAR | 1.0000 | 0.0002 | 0.0005 |

```
kable(km_evaluate, caption = "External metrics for Kmeans clustering method", align = "c", digi
```

Table 52: External metrics for Kmeans clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| STAR | 0.5963 | 0.4086 | 0.4849 |
| QSO | 0.3463 | 0.8929 | 0.4990 |
| GALAXY | 0.7401 | 0.7360 | 0.7381 |

```
kable(pam_evaluate, caption = "External metrics for PAM clustering method", align = "c", digits
```

Table 53: External metrics for PAM clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.8209 | 0.9664 | 0.8877 |
| STAR | 0.9479 | 0.7492 | 0.8369 |
| QSO | 0.8393 | 0.8235 | 0.8314 |

**2.**

```
# HCA
HCA = data.frame(HCA = c("HCA1","HCA2","HCA3","HCA4"), Distance = c("manhattan", "euclidean","m
```

```
cresult_HCA = data.frame(class = res_frame$class, hca1 = 0, hca2 = 0, hca3 = 0, hca4 = 0)
hca1 = hclust(dist(rpca_12, "manhattan"), "complete")
hca2 = hclust(dist(rpca_12, "euclidean"), "complete")
hca3 = hclust(dist(rpca_12, "manhattan"), "single")
hca4 = hclust(dist(rpca_12, "euclidean"), "single")

cresult_HCA$hca1 = cutree(hca1,k)
cresult_HCA$hca2 = cutree(hca2,k)
cresult_HCA$hca3 = cutree(hca3,k)
cresult_HCA$hca4 = cutree(hca4,k)

HCA_inter_matr = internal_matric(cresult_HCA, rpca_12)

hca1_table = table(cresult_HCA$class,cresult_HCA$hca1)
hca2_table = table(cresult_HCA$class,cresult_HCA$hca2)
hca3_table = table(cresult_HCA$class,cresult_HCA$hca3)
hca4_table = table(cresult_HCA$class,cresult_HCA$hca4)

hca1_external  = maximize_diag(hca1_table)
hca2_external  = maximize_diag(hca2_table)
hca3_external  = maximize_diag(hca3_table)
hca4_external  = maximize_diag(hca4_table)

hca1_precision = precision(hca1_external)
hca2_precision = precision(hca2_external)
hca3_precision = precision(hca3_external)
hca4_precision = precision(hca4_external)

hca1_recall = recall(hca1_external)
hca2_recall = recall(hca2_external)
hca3_recall = recall(hca3_external)
hca4_recall = recall(hca4_external)

hca1_f1 = f1(hca1_precision, hca1_recall)
hca2_f1 = f1(hca2_precision, hca2_recall)
hca3_f1 = f1(hca3_precision, hca3_recall)
hca4_f1 = f1(hca4_precision, hca4_recall)

HCA1_evaluate = data.frame(Class = names(hca1_recall), Precision = hca1_precision, Recall = hca
HCA2_evaluate = data.frame(Class = names(hca2_recall), Precision = hca2_precision, Recall = hca
HCA3_evaluate = data.frame(Class = names(hca3_recall), Precision = hca3_precision, Recall = hca
HCA4_evaluate = data.frame(Class = names(hca4_recall), Precision = hca4_precision, Recall = hca

kable(HCA, caption = "Combinations of parameters for HCA clustering method", align = "c", digi
```

Table 54: Combinations of parameters for HCA clustering method

| HCA | Distance | Method |
|---|---|---|
| HCA1 | manhattan | complete |
| HCA2 | euclidean | complete |
| HCA3 | manhattan | single |
| HCA4 | euclidean | single |

```
kable(HCA_inter_matr, caption = "Internal metrics for clustering result", align = "c", digits =
```

Table 55: Internal metrics for clustering result

| | hca1 | hca2 | hca3 | hca4 |
|---|---|---|---|---|
| average.between | 35.19924 | 35.19924 | 35.19924 | 40.49842 |
| average.within | 5.369524 | 5.369524 | 5.369524 | 5.373386 |
| dunn | 0.4501908 | 0.4501908 | 0.4501908 | 0.3411657 |

```
kable(HCA1_evaluate, caption = "External metrices for HCA1 clustering method", align = "c", dig
```

Table 56: External metrices for HCA1 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5001 | 0.9994 | 0.6667 |
| QSO | 0.0000 | 0.0000 | NaN |
| STAR | 1.0000 | 0.0002 | 0.0005 |

```
kable(HCA2_evaluate, caption = "External metrices for HCA2 clustering method", align = "c", dig
```

Table 57: External metrices for HCA2 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5001 | 0.9994 | 0.6667 |
| QSO | 0.0000 | 0.0000 | NaN |
| STAR | 1.0000 | 0.0002 | 0.0005 |

```
kable(HCA3_evaluate, caption = "External metrices for HCA3 clustering method", align = "c", dig
```

Table 58: External metrices for HCA3 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5001 | 0.9994 | 0.6667 |
| QSO | 0.0000 | 0.0000 | NaN |
| STAR | 1.0000 | 0.0002 | 0.0005 |

```
kable(HCA4_evaluate, caption = "External metrices for HCA4 clustering method", align = "c", dig
```

Table 59: External metrices for HCA4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5 | 0.9994 | 0.6666 |
| QSO | 0.0 | 0.0000 | NaN |
| STAR | 1.0 | 0.0002 | 0.0005 |

```
# Kmean
Kmean = data.frame(Kmeans = c("km1","km2","km3","km4"), nstart = c(5, 10, 5, 10), Method = c("E
cresult_Kmeans = data.frame(class = res_frame$class, km1 = 0, km2 = 0, km3 = 0, km4 = 0)

km1 = kmeans(rpca_12, 3, nstart = 5)
km2 = kmeans(rpca_12, 3, nstart = 10)
km3 = kmeans(rpca_12, 3, iter.max = 1000, nstart = 5, "Lloyd")
km4 = kmeans(rpca_12, 3, iter.max = 1000, nstart = 10, "Lloyd")

cresult_Kmeans$km1 = km1$cluster
cresult_Kmeans$km2 = km2$cluster
cresult_Kmeans$km3 = km3$cluster
cresult_Kmeans$km4 = km4$cluster

Kmeans_inter_matr = internal_matric(cresult_Kmeans, rpca_12)

km1_table = table(cresult_Kmeans$class,cresult_Kmeans$km1)
km2_table = table(cresult_Kmeans$class,cresult_Kmeans$km2)
km3_table = table(cresult_Kmeans$class,cresult_Kmeans$km3)
km4_table = table(cresult_Kmeans$class,cresult_Kmeans$km4)

km1_external  = maximize_diag(km1_table)
km2_external  = maximize_diag(km2_table)
km3_external  = maximize_diag(km3_table)
km4_external  = maximize_diag(km4_table)

km1_precision = precision(km1_external)
km2_precision = precision(km2_external)
km3_precision = precision(km3_external)
```

```
km4_precision = precision(km4_external)

km1_recall = recall(km1_external)
km2_recall = recall(km2_external)
km3_recall = recall(km3_external)
km4_recall = recall(km4_external)

km1_f1 = f1(km1_precision, km1_recall)
km2_f1 = f1(km2_precision, km2_recall)
km3_f1 = f1(km3_precision, km3_recall)
km4_f1 = f1(km4_precision, km4_recall)

km1_evaluate = data.frame(Class = names(km1_recall), Precision = km1_precision, Recall = km1_re
km2_evaluate = data.frame(Class = names(km2_recall), Precision = km2_precision, Recall = km2_re
km3_evaluate = data.frame(Class = names(km3_recall), Precision = km3_precision, Recall = km3_re
km4_evaluate = data.frame(Class = names(km4_recall), Precision = km4_precision, Recall = km4_re

kable(Kmean, caption = "Combinations of parameters for kmeans clustering method", align = "c",
```

Table 60: Combinations of parameters for kmeans clustering method

| Kmeans | nstart | Method |
|---|---|---|
| km1 | 5 | Hartigan-Wong |
| km2 | 10 | Hartigan-Wong |
| km3 | 5 | Lloyd |
| km4 | 10 | Lloyd |

```
kable(Kmeans_inter_matr, caption = "Internal metrics for Kmeans clustering result", align = "c"
```

Table 61: Internal metrics for Kmeans clustering result

| | km1 | km2 | km3 | km4 |
|---|---|---|---|---|
| average.between | 5.973809 | 5.898279 | 5.974079 | 5.898137 |
| average.within | 4.671595 | 4.648573 | 4.671553 | 4.648607 |
| dunn | 0.002569826 | 0.002835292 | 0.002569826 | 0.002835292 |

```
kable(km1_evaluate, caption = "External mectrices for km1 clustering method", align = "c", digi
```

Table 62: External mectrices for km1 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.7401 | 0.7360 | 0.7381 |
| STAR | 0.5963 | 0.4086 | 0.4849 |
| QSO | 0.3463 | 0.8929 | 0.4990 |

```
kable(km2_evaluate, caption = "External mectrices for km2 clustering method", align = "c", digi
```

Table 63: External mectrices for km2 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5157 | 0.2417 | 0.3291 |
| QSO | 0.0058 | 0.0200 | 0.0090 |
| STAR | 0.3602 | 0.4108 | 0.3838 |

```
kable(km3_evaluate, caption = "External mectrices for km3 clustering method", align = "c", digi
```

Table 64: External mectrices for km3 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| QSO | 0.3459 | 0.8929 | 0.4987 |
| STAR | 0.5968 | 0.4084 | 0.4849 |
| GALAXY | 0.7403 | 0.7364 | 0.7383 |

```
kable(km4_evaluate, caption = "External mectrices for km4 clustering method", align = "c", digi
```

Table 65: External mectrices for km4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.5161 | 0.2417 | 0.3292 |
| QSO | 0.1069 | 0.5988 | 0.1814 |
| STAR | 0.5603 | 0.3940 | 0.4627 |

```
PAM = data.frame(PAM = c("pam1","pam2","pam3","pam4"), Metric = c("manhattan", "manhattan", "eu

cresult_PAM = data.frame(class = res_frame$class, pam1 = 0, pam2 = 0, pam3 = 0, pam4 = 0)

pam1 = pam(rpca_12, 3, metric = "manhattan", pamonce = 3)
pam2 = pam(rpca_12, 3, metric = "manhattan", pamonce = 6)
pam3 = pam(rpca_12, 3, metric = "euclidean", pamonce = 3)
```

```
pam4 = pam(rpca_12, 3, metric = "euclidean", pamonce = 6)

cresult_PAM$pam1 = pam1$cluster
cresult_PAM$pam2 = pam2$cluster
cresult_PAM$pam3 = pam3$cluster
cresult_PAM$pam4 = pam4$cluster

PAM_inter_matr = internal_matric(cresult_PAM, rpca_12)

pam1_table = table(cresult_PAM$class,cresult_PAM$pam1)
pam2_table = table(cresult_PAM$class,cresult_PAM$pam2)
pam3_table = table(cresult_PAM$class,cresult_PAM$pam3)
pam4_table = table(cresult_PAM$class,cresult_PAM$pam4)

pam1_external  = maximize_diag(pam1_table)
pam2_external  = maximize_diag(pam2_table)
pam3_external  = maximize_diag(pam3_table)
pam4_external  = maximize_diag(pam4_table)

pam1_precision = precision(pam1_external)
pam2_precision = precision(pam2_external)
pam3_precision = precision(pam3_external)
pam4_precision = precision(pam4_external)

pam1_recall = recall(pam1_external)
pam2_recall = recall(pam2_external)
pam3_recall = recall(pam3_external)
pam4_recall = recall(pam4_external)

pam1_f1 = f1(pam1_precision, pam1_recall)
pam2_f1 = f1(pam2_precision, pam2_recall)
pam3_f1 = f1(pam3_precision, pam3_recall)
pam4_f1 = f1(pam4_precision, pam4_recall)

pam1_evaluate = data.frame(Class = names(pam1_recall), Precision = pam1_precision, Recall = pa
pam2_evaluate = data.frame(Class = names(pam2_recall), Precision = pam2_precision, Recall = pa
pam3_evaluate = data.frame(Class = names(pam3_recall), Precision = pam3_precision, Recall = pa
pam4_evaluate = data.frame(Class = names(pam4_recall), Precision = pam4_precision, Recall = pa

kable(PAM, caption = "Combinations of parameters for pam clustering method", align = "c", digi
```

Table 66: Combinations of parameters for pam clustering method

| PAM | Metric | Pamonce |
|---|---|---|
| pam1 | manhattan | 3 |
| pam2 | manhattan | 6 |

| PAM | Metric | Pamonce |
|---|---|---|
| pam3 | euclidean | 3 |
| pam4 | euclidean | 6 |

```
kable(PAM_inter_matr, caption = "Internal metrics for PAM clustering result", align = "c", digi
```

Table 67: Internal metrics for PAM clustering result

| | pam1 | pam2 | pam3 | pam4 |
|---|---|---|---|---|
| average.between | 5.630398 | 5.630838 | 6.016513 | 6.016513 |
| average.within | 4.993547 | 4.98387 | 4.804679 | 4.804679 |
| dunn | 0.001835621 | 0.002468855 | 0.006270207 | 0.006270207 |

```
kable(pam1_evaluate, caption = "External mectrices for pam1 clustering method", align = "c", di
```

Table 68: External mectrices for pam1 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| STAR | 0.5805 | 0.5732 | 0.5768 |
| GALAXY | 0.6012 | 0.4561 | 0.5187 |
| QSO | 0.1129 | 0.2812 | 0.1612 |

```
kable(pam2_evaluate, caption = "External mectrices for pam2 clustering method", align = "c", di
```

Table 69: External mectrices for pam2 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.6292 | 0.3700 | 0.4660 |
| QSO | 0.0926 | 0.4647 | 0.1545 |
| STAR | 0.6992 | 0.4743 | 0.5651 |

```
kable(pam3_evaluate, caption = "External mectrices for pam3 clustering method", align = "c", di
```

Table 70: External mectrices for pam3 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.8209 | 0.9664 | 0.8877 |
| STAR | 0.9479 | 0.7492 | 0.8369 |
| QSO | 0.8393 | 0.8235 | 0.8314 |

```
kable(pam4_evaluate, caption = "External mectrices for pam4 clustering method", align = "c", di
```

Table 71: External mectrices for pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.8209 | 0.9664 | 0.8877 |
| STAR | 0.9479 | 0.7492 | 0.8369 |
| QSO | 0.8393 | 0.8235 | 0.8314 |

**3.**

```
tpca_result = data.frame(class = res_frame$class, pam = 0)
tpca_pam = pam(tpca, 3, metric = "euclidean", pamonce = 6)
tpca_result$pam = tpca_pam$cluster
tpca_inter_matr = internal_matric(tpca_result, tpca)

tpca_table = table(tpca_result$class,tpca_result$pam)
tpca_external  = maximize_diag(tpca_table)
tpca_precision = precision(tpca_external)
tpca_recall = recall(tpca_external)
tpca_f1 = f1(tpca_precision, tpca_recall)
tpca_evaluate = data.frame(Class = names(tpca_recall), Precision = tpca_precision, Recall = tpc

kable(tpca_inter_matr, caption = "Internal metrics for pam4 clustering method", align = "c", di
```

**i.**

Table 72: Internal metrics for pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 6.0321 | 4.8206 | 0.008 |

```
kable(tpca_evaluate, caption = "External metrics for pam4 clustering method", align = "c", digi
```

Table 73: External metrics for pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.8211 | 0.9662 | 0.8878 |
| STAR | 0.9479 | 0.7497 | 0.8372 |
| QSO | 0.8383 | 0.8235 | 0.8309 |

```
rpca12_result = data.frame(class = res_frame$class, pam = 0)
rpca12_pam = pam(rpca_12, 3, metric = "euclidean", pamonce = 6)
rpca12_result$pam = rpca12_pam$cluster
rpca12_inter_matr = internal_matric(rpca12_result, rpca_12)

rpca12_table = table(rpca12_result$class,rpca12_result$pam)
rpca12_external  = maximize_diag(rpca12_table)
rpca12_precision = precision(rpca12_external)
rpca12_recall = recall(rpca12_external)
rpca12_f1 = f1(rpca12_precision, rpca12_recall)
rpca12_evaluate = data.frame(Class = names(rpca12_recall), Precision = rpca12_precision, Recall

kable(rpca12_inter_matr, caption = "Internal metrics for pam4 clustering method", align = "c",
```

**ii.**

Table 74: Internal metrics for pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 6.0165 | 4.8047 | 0.0063 |

```
kable(rpca12_evaluate, caption = "External metrics for pam4 clustering method", align = "c", di
```

Table 75: External metrics for pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.8209 | 0.9664 | 0.8877 |
| STAR | 0.9479 | 0.7492 | 0.8369 |
| QSO | 0.8393 | 0.8235 | 0.8314 |

```
reduced_result = data.frame(class = res_frame$class, pam = 0)
reduced_pam = pam(uncorrelated_DS, 3, metric = "euclidean", pamonce = 6)
reduced_result$pam = reduced_pam$cluster
reduced_inter_matr = internal_matric(reduced_result, uncorrelated_DS)

reduced_table = table(reduced_result$class,reduced_result$pam)
reduced_external  = maximize_diag(reduced_table)
reduced_precision = precision(reduced_external)
reduced_recall = recall(reduced_external)
reduced_f1 = f1(reduced_precision, reduced_recall)
```

```
reduced_evaluate = data.frame(Class = names(reduced_recall), Precision = reduced_precision, Re

kable(reduced_inter_matr, caption = "Internal metrics for pam4 clustering method", align = "c"
```

**iii.**

Table 76: Internal metrics for pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 4908.854 | 1587.29 | 4e-04 |

```
kable(reduced_evaluate, caption = "External metrics for pam4 clustering method", align = "c", 
```

Table 77: External metrics for pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.7277 | 0.9674 | 0.8306 |
| QSO | 0.0488 | 0.1929 | 0.0779 |
| STAR | 0.5714 | 0.0010 | 0.0019 |

```
# zero_na
mc_zero_result = data.frame(class = res_frame$class, pam = 0)
mc_zero_pam = pam(mean_centre_zero, 3, metric = "euclidean", pamonce = 6)
mc_zero_result$pam = mc_zero_pam$cluster
mc_zero_inter_matr = internal_matric(mc_zero_result, mean_centre_zero)

mc_zero_table = table(mc_zero_result$class,mc_zero_result$pam)
mc_zero_external  = maximize_diag(mc_zero_table)
mc_zero_precision = precision(mc_zero_external)
mc_zero_recall = recall(mc_zero_external)
mc_zero_f1 = f1(mc_zero_precision, mc_zero_recall)
mc_zero_evaluate = data.frame(Class = names(mc_zero_recall), Precision = mc_zero_precision, Re

kable(mc_zero_inter_matr, caption = "Internal criterias for mean_centre_zero data set using pa
```

**iv.**

Table 78: Internal criterias for mean_centre_zero data set using pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 3.686891e+18 | 3.044154e+17 | 0.103 |

```
kable(mc_zero_evaluate, caption = "External criterias for mean_centre_zero data set using pam4
```

Table 79: External criterias for mean_centre_zero data set using pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.7356 | 0.9731 | 0.8379 |
| QSO | 0.0294 | 0.1000 | 0.0455 |
| STAR | 0.7767 | 0.0958 | 0.1706 |

```
# mean_centre_mean
mc_mean_result = data.frame(class = res_frame$class, pam = 0)
mc_mean_pam = pam(mean_centre_mean, 3, metric = "euclidean", pamonce = 6)
mc_mean_result$pam = mc_mean_pam$cluster
mc_mean_inter_matr = internal_matric(mc_mean_result, mean_centre_mean)

mc_mean_table = table(mc_mean_result$class,mc_mean_result$pam)
mc_mean_external  = maximize_diag(mc_mean_table)
mc_mean_precision = precision(mc_mean_external)
mc_mean_recall = recall(mc_mean_external)
mc_mean_f1 = f1(mc_mean_precision, mc_mean_recall)
mc_mean_evaluate = data.frame(Class = names(mc_mean_recall), Precision = mc_mean_precision, Re

kable(mc_mean_inter_matr, caption = "Internal metrics for mean_centre_mean data set using pam4
```

Table 80: Internal metrics for mean_centre_mean data set using pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 3.67829e+18 | 3.021919e+17 | 0.103 |

```
kable(mc_mean_evaluate, caption = "External metrics for mean_centre_mean data set using pam4 cl
```

Table 81: External metrics for mean_centre_mean data set using pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.7379 | 0.9731 | 0.8393 |
| QSO | 0.0292 | 0.1000 | 0.0452 |
| STAR | 0.7767 | 0.0958 | 0.1706 |

```
# mean_centre_mean
mc_median_result = data.frame(class = res_frame$class, pam = 0)
mc_median_pam = pam(mean_centre_median, 3, metric = "euclidean", pamonce = 6)
mc_median_result$pam = mc_median_pam$cluster
mc_median_inter_matr = internal_matric(mc_median_result, mean_centre_median)

mc_median_table = table(mc_median_result$class,mc_median_result$pam)
mc_median_external  = maximize_diag(mc_median_table)
mc_median_precision = precision(mc_median_external)
mc_median_recall = recall(mc_median_external)
mc_median_f1 = f1(mc_median_precision, mc_median_recall)
mc_median_evaluate = data.frame(Class = names(mc_median_recall), Precision = mc_median_precisi

kable(mc_median_inter_matr, caption = "Internal metrics for mean_centre_median data set using
```

Table 82: Internal metrics for mean_centre_median data set using pam4 clustering method

| average.between | average.within | dunn |
|---|---|---|
| 3.679952e+18 | 3.015658e+17 | 0.103 |

```
kable(mc_median_evaluate, caption = "External metrics for mean_centre_median data set using pa
```

Table 83: External metrics for mean_centre_median data set using pam4 clustering method

| Class | Precision | Recall | F1 |
|---|---|---|---|
| GALAXY | 0.7379 | 0.9731 | 0.8393 |
| QSO | 0.0292 | 0.1000 | 0.0452 |
| STAR | 0.7767 | 0.0958 | 0.1706 |

## CLASSIFICATION

```
rpca12_with_class = data.frame(rpca_12)
rpca12_with_class$CLASS = res_frame$class
write.csv(rpca12_with_class, file = "rpca_12.csv")
```

```
tpca_with_class = data.frame(tpca)
tpca_with_class$CLASS = res_frame$class
write.csv(tpca_with_class, file = "tpca.csv")
```

```
uncorrelated_DS$CLASS = res_frame$class
write.csv(uncorrelated_DS, file = "reduced.csv")
```

```
normalised_zero$class = res_frame$class
normalised_mean$class = res_frame$class
normalised_median$class = res_frame$class
write.csv(normalised_zero, file = "normalised_zero.csv")
write.csv(normalised_mean, file = "normalised_mean.csv")
write.csv(normalised_median, file = "normalised_median.csv")
```

**1.**

```
#IBK
confusion_matrix_1 = matrix(c(5019,53,1,1,766,3,7,31,4171), nrow = 3, dimnames = list(c("GALAX
accuracy_table_1 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.989,0.995,0.9
kable(confusion_matrix_1, caption = "The confusion matrix for IBK with k = 5 ", align = "c", di
```

Table 84: The confusion matrix for IBK with k = 5

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5019 | 1 | 7 |
| QSO | 53 | 766 | 31 |
| STAR | 1 | 3 | 4171 |

```
kable(accuracy_table_1, caption = "The external matrix for IBK with k = 5", align = "c", digits
```

Table 85: The external matrix for IBK with k = 5

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.989 | 0.998 | 0.994 |
| QSO | 0.995 | 0.901 | 0.946 |

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| STAR | 0.991 | 0.999 | 0.995 |

```
#
confusion_matrix_2 = matrix(c(5026,0,1,1,849,0,0,1,4174), nrow = 3, dimnames = list(c("GALAXY"
accuracy_table_2 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(1.000,0.999,1.0
kable(confusion_matrix_2, caption = "The confusion matrix for J48", align = "c", digits = 4, "s
```

Table 86: The confusion matrix for J48

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5026 | 1 | 0 |
| QSO | 0 | 849 | 1 |
| STAR | 1 | 0 | 4174 |

```
kable(accuracy_table_2, caption = "The external metrics for J48", align = "c", digits = 4, "sim
```

Table 87: The external metrics for J48

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 1.000 | 1.000 | 1.000 |
| QSO | 0.999 | 0.999 | 0.999 |
| STAR | 1.000 | 1.000 | 1.000 |

```
#NavieBayes
confusion_matrix_3 = matrix(c(4949,54,15,0,767,2,78,29,4158), nrow = 3, dimnames = list(c("GALA
accuracy_table_3 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.986,0.997,0.9
kable(confusion_matrix_3, caption = "The confusion matrix for NaiveBayes", align = "c", digits
```

Table 88: The confusion matrix for NaiveBayes

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 4949 | 0 | 78 |
| QSO | 54 | 767 | 29 |
| STAR | 15 | 2 | 4158 |

```
kable(accuracy_table_3, caption = "The external metrics for NaiveBayes", align = "c", digits =
```

Table 89: The external metrics for NaiveBayes

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.986 | 0.984 | 0.985 |
| QSO | 0.997 | 0.902 | 0.947 |
| STAR | 0.975 | 0.996 | 0.985 |

```
#OneR
confusion_matrix_4 = matrix(c(5025,0,1,1,849,0,1,1,4174), nrow = 3, dimnames = list(c("GALAXY"
accuracy_table_4 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(1.000,0.999,1.0
kable(confusion_matrix_4, caption = "The confusion matrix for OneR", align = "c", digits = 4, "
```

Table 90: The confusion matrix for OneR

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5025 | 1 | 1 |
| QSO | 0 | 849 | 1 |
| STAR | 1 | 0 | 4174 |

```
kable(accuracy_table_4, caption = "The external metrics for OneR", align = "c", digits = 4, "si
```

Table 91: The external metrics for OneR

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 1.000 | 1.000 | 1.000 |
| QSO | 0.999 | 0.999 | 0.999 |
| STAR | 1.000 | 1.000 | 1.000 |

```
#ZeroR
confusion_matrix_5 = matrix(c(5027,850,4175,0,0,0,0,0,0), nrow = 3, dimnames = list(c("GALAXY"
accuracy_table_5 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.500,"?","?")
kable(confusion_matrix_5, caption = "The confusion matrix for ZeroR", align = "c", digits = 4,
```

Table 92: The confusion matrix for ZeroR

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5027 | 0 | 0 |
| QSO | 850 | 0 | 0 |
| STAR | 4175 | 0 | 0 |

```
kable(accuracy_table_5, caption = "The external metrics for ZeroR", align = "c", digits = 4, "
```

Table 93: The external metrics for ZeroR

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.5 | 1 | 0.667 |
| QSO | ? | 0 | ? |
| STAR | ? | 0 | ? |

**2.**

```
k = c(5,3,5,3,5,3,5,3)
Search_Algorithm = c("LinearNNSearch", "LinearNNSearch", "KDTree", "KDTree", "LinearNNSearch",
Percentage_split = c(0.66, 0.66, 0.66, 0.66, 0.80, 0.80, 0.80, 0.80)

Parameters_table = data.frame(Groups = seq(1,8,1), K = k, Search_Algorithm = Search_Algorithm,

kable(Parameters_table, caption = "Different parameters for KNN", align = "c", digits = 4, "sir
```

Table 94: Different parameters for KNN

| Groups | K | Search_Algorithm | Percentage_split |
|---|---|---|---|
| 1 | 5 | LinearNNSearch | 0.66 |
| 2 | 3 | LinearNNSearch | 0.66 |
| 3 | 5 | KDTree | 0.66 |
| 4 | 3 | KDTree | 0.66 |
| 5 | 5 | LinearNNSearch | 0.80 |
| 6 | 3 | LinearNNSearch | 0.80 |
| 7 | 5 | KDTree | 0.80 |
| 8 | 3 | KDTree | 0.80 |

```
#Group1
confusion_matrix_G1 = matrix(c(1635,54,225,3,217,4,85,21,1174), nrow = 3, dimnames = list(c("GA
accuracy_table_G1 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.854,0.969,0

kable(confusion_matrix_G1, caption = "The confusion matrix for Group1", align = "c", digits = 4
```

Table 95: The confusion matrix for Group1

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 1635 | 3 | 85 |
| QSO | 54 | 217 | 21 |

| | GALAXY | QSO | STAR |
|---|---|---|---|
| STAR | 225 | 4 | 1174 |

```
kable(accuracy_table_G1, caption = "The external metrics for Group1", align = "c", digits = 4,
```

Table 96: The external metrics for Group1

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.854 | 0.949 | 0.899 |
| QSO | 0.969 | 0.743 | 0.841 |
| STAR | 0.917 | 0.837 | 0.875 |

```
#Group2
confusion_matrix_G2 = matrix(c(1604,49,216,7,223,8,112,20,1179), nrow = 3, dimnames = list(c("
accuracy_table_G2 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.858,0.937,0

kable(confusion_matrix_G2, caption = "The confusion matrix for Group2", align = "c", digits = 4
```

Table 97: The confusion matrix for Group2

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 1604 | 7 | 112 |
| QSO | 49 | 223 | 20 |
| STAR | 216 | 8 | 1179 |

```
kable(accuracy_table_G2, caption = "The external metrics for Group2", align = "c", digits = 4,
```

Table 98: The external metrics for Group2

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.858 | 0.931 | 0.893 |
| QSO | 0.937 | 0.764 | 0.842 |
| STAR | 0.899 | 0.840 | 0.869 |

```
#Group3
confusion_matrix_G3 = matrix(c(1635,54,225,3,217,4,85,21,1174), nrow = 3, dimnames = list(c("GA
accuracy_table_G3 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.854,0.969,0
kable(confusion_matrix_G3, caption = "The confusion matrix for Group3", align = "c", digits = 4
```

Table 99: The confusion matrix for Group3

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 1635 | 3 | 85 |
| QSO | 54 | 217 | 21 |
| STAR | 225 | 4 | 1174 |

```
kable(accuracy_table_G3, caption = "The external metrics for Group3", align = "c", digits = 4,
```

Table 100: The external metrics for Group3

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.854 | 0.949 | 0.899 |
| QSO | 0.969 | 0.743 | 0.841 |
| STAR | 0.917 | 0.837 | 0.875 |

```
#Group4
confusion_matrix_G4 = matrix(c(1604,49,216,7,223,8,112,20,1179), nrow = 3, dimnames = list(c("
accuracy_table_G4 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.858,0.937,0
kable(confusion_matrix_G4, caption = "The confusion matrix for Group4", align = "c", digits = 4
```

Table 101: The confusion matrix for Group4

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 1604 | 7 | 112 |
| QSO | 49 | 223 | 20 |
| STAR | 216 | 8 | 1179 |

```
kable(accuracy_table_G4, caption = "The external metrics for Group4", align = "c", digits = 4,
```

Table 102: The external metrics for Group4

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.858 | 0.931 | 0.893 |
| QSO | 0.937 | 0.764 | 0.842 |
| STAR | 0.899 | 0.840 | 0.869 |

```
#Group5
confusion_matrix_G5 = matrix(c(960,22,130,3,131,2,55,16,691), nrow = 3, dimnames = list(c("GAL
accuracy_table_G5 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.863,0.963,0
kable(confusion_matrix_G5, caption = "The confusion matrix for Group5", align = "c", digits = 4
```

Table 103: The confusion matrix for Group5

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 960 | 3 | 55 |
| QSO | 22 | 131 | 16 |
| STAR | 130 | 2 | 691 |

```
kable(accuracy_table_G5, caption = "The external metrics for Group5", align = "c", digits = 4,
```

Table 104: The external metrics for Group5

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.863 | 0.943 | 0.901 |
| QSO | 0.963 | 0.775 | 0.859 |
| STAR | 0.907 | 0.840 | 0.872 |

```
#Group6
confusion_matrix_G6 = matrix(c(949,25,130,5,131,2,64,13,691), nrow = 3, dimnames = list(c("GAL
accuracy_table_G6 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.860,0.949,0
kable(confusion_matrix_G6, caption = "The confusion matrix for Group6", align = "c", digits = 4
```

Table 105: The confusion matrix for Group6

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 949 | 5 | 64 |
| QSO | 25 | 131 | 13 |
| STAR | 130 | 2 | 691 |

```
kable(accuracy_table_G6, caption = "The external metrics for Group6", align = "c", digits = 4,
```

Table 106: The external metrics for Group6

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.860 | 0.932 | 0.894 |
| QSO | 0.949 | 0.775 | 0.853 |
| STAR | 0.900 | 0.840 | 0.869 |

```
#Group7
confusion_matrix_G7 = matrix(c(960,22,130,3,131,2,55,16,691), nrow = 3, dimnames = list(c("GAL
accuracy_table_G7 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.863,0.963,0
kable(confusion_matrix_G7, caption = "The confusion matrix for Group7", align = "c", digits = 4
```

Table 107: The confusion matrix for Group7

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 960 | 3 | 55 |
| QSO | 22 | 131 | 16 |
| STAR | 130 | 2 | 691 |

```
kable(accuracy_table_G7, caption = "The external metrics for Group7", align = "c", digits = 4,
```

Table 108: The external metrics for Group7

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.863 | 0.943 | 0.901 |
| QSO | 0.963 | 0.775 | 0.859 |
| STAR | 0.907 | 0.840 | 0.872 |

```
#Group8
confusion_matrix_G8 = matrix(c(949,25,130,5,131,2,64,13,691), nrow = 3, dimnames = list(c("GAL
accuracy_table_G8 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.860,0.949,0
kable(confusion_matrix_G8, caption = "The confusion matrix for Group8", align = "c", digits = 4
```

Table 109: The confusion matrix for Group8

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 949 | 5 | 64 |
| QSO | 25 | 131 | 13 |
| STAR | 130 | 2 | 691 |

```
kable(accuracy_table_G8, caption = "The external metrics for Group8", align = "c", digits = 4,
```

Table 110: The external metrics for Group8

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.860 | 0.932 | 0.894 |
| QSO | 0.949 | 0.775 | 0.853 |
| STAR | 0.900 | 0.840 | 0.869 |

**3.**

```
#tpca
confusion_matrix_tpca = matrix(c(4790,60,195,35,767,12,202,23,3968), nrow = 3, dimnames = list
accuracy_table_tpca = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.949,0.942
kable(confusion_matrix_tpca, caption = "The confusion matrix for tpca", align = "c", digits = 4
```

**i.**

Table 111: The confusion matrix for tpca

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 4790 | 35 | 202 |
| QSO | 60 | 767 | 23 |
| STAR | 195 | 12 | 3968 |

```
kable(accuracy_table_tpca, caption = "The external metrics for tpca", align = "c", digits = 4,
```

Table 112: The external metrics for tpca

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.949 | 0.953 | 0.951 |
| QSO | 0.942 | 0.902 | 0.922 |
| STAR | 0.946 | 0.950 | 0.948 |

```
#rpca_12
confusion_matrix_rpca12= matrix(c(5026,0,1,1,849,0,0,1,4174), nrow = 3, dimnames = list(c("GAL
accuracy_table_rpca12 = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(1.000,0.9
kable(confusion_matrix_rpca12, caption = "The confusion matrix for rpca12", align = "c", digit
```

**ii.**

Table 113: The confusion matrix for rpca12

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5026 | 1 | 0 |
| QSO | 0 | 849 | 1 |
| STAR | 1 | 0 | 4174 |

```
kable(accuracy_table_rpca12, caption = "The external metrics for rpca12", align = "c", digits =
```

Table 114: The external metrics for rpca12

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 1.000 | 1.000 | 1.000 |
| QSO | 0.999 | 0.999 | 0.999 |
| STAR | 1.000 | 1.000 | 1.000 |

```
#reduced
confusion_matrix_reduced = matrix(c(5006,17,5,10,832,0,11,1,4170), nrow = 3, dimnames = list(c
accuracy_table_reduced = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.996,0.9
kable(confusion_matrix_reduced, caption = "The confusion matrix for reduced", align = "c", digi
```

**iii.**

Table 115: The confusion matrix for reduced

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5006 | 10 | 11 |
| QSO | 17 | 832 | 1 |
| STAR | 5 | 0 | 4170 |

```
kable(accuracy_table_reduced, caption = "The external metrics for reduced", align = "c", digits
```

Table 116: The external metrics for reduced

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.996 | 0.996 | 0.996 |
| QSO | 0.988 | 0.979 | 0.983 |
| STAR | 0.997 | 0.999 | 0.998 |

```
#nomalised data set whose missing value has been replaced by zero
confusion_matrix_n_zero = matrix(c(4954,62,4,42,787,0,31,1,4171), nrow = 3, dimnames = list(c('
accuracy_table_n_zero = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.987,0.94
kable(confusion_matrix_n_zero, caption = "The confusion matrix for normalized_zero", align = "c
```

**iv.**

Table 117: The confusion matrix for normalized_zero

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 4954 | 42 | 31 |
| QSO | 62 | 787 | 1 |
| STAR | 4 | 0 | 4171 |

```
kable(accuracy_table_n_zero, caption = "The external metrics for normalized_zero", align = "c"
```

Table 118: The external metrics for normalized_zero

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.987 | 0.985 | 0.986 |
| QSO | 0.949 | 0.926 | 0.937 |
| STAR | 0.992 | 0.999 | 0.996 |

```
#nomalised data set whose missing value has been replaced by mean
confusion_matrix_n_mean = matrix(c(5004,18,5,12,831,0,11,1,4171), nrow = 3, dimnames = list(c('
accuracy_table_n_mean = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.995,0.98
kable(confusion_matrix_n_mean, caption = "The confusion matrix for normalized_mean", align = "c
```

Table 119: The confusion matrix for normalized_mean

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 5004 | 12 | 11 |
| QSO | 18 | 831 | 1 |
| STAR | 5 | 0 | 4171 |

```
kable(accuracy_table_n_mean, caption = "The external metrics for normalized_mean", align = "c"
```

Table 120: The external metrics for normalized_mean

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.995 | 0.995 | 0.995 |
| QSO | 0.986 | 0.978 | 0.982 |
| STAR | 0.997 | 0.999 | 0.998 |

```
#nomalised data set whose missing value has been replaced by median
confusion_matrix_n_median = matrix(c(4981,38,4,17,811,0,29,1,4171), nrow = 3, dimnames = list(c
accuracy_table_n_median = data.frame(Class = c("GALAXY", "QSO", "STAR"), Precision = c(0.992,0
kable(confusion_matrix_n_median, caption = "The confusion matrix for normalized_median", align
```

Table 121: The confusion matrix for normalized_median

| | GALAXY | QSO | STAR |
|---|---|---|---|
| GALAXY | 4981 | 17 | 29 |
| QSO | 38 | 811 | 1 |
| STAR | 4 | 0 | 4171 |

```
kable(accuracy_table_n_median, caption = "The external metrics for normalized_median", align =
```

Table 122: The external metrics for normalized_median

| Class | Precision | Recall | F_Measure |
|---|---|---|---|
| GALAXY | 0.992 | 0.991 | 0.991 |
| QSO | 0.979 | 0.954 | 0.967 |
| STAR | 0.993 | 0.999 | 0.996 |