

1 ANALYSIS AND PRE-PROCESSING

i Explore the data

i

Table 1 mean,dispersion, and number of NA in the dataset

	Mean	Dispersion	Number of missing values
objid	1.240000e+18	0	0
dia	2000.028	22677.73	6646
rerun	301	0	30
ra	175.5448	47.7724	48
dec	14.82675	25.20827	49
u	18.61902	0.8289577	50
g	17.37185	0.9454507	51
r	16.84077	1.067718	53
i	16.58386	1.141931	50
z	16.42307	1.203479	53
run	980.9299	273.2851	50
m_unt	0.0002335742	0.00007074365	46
native	0.5026995	0.5000177	50
flux	183.5176	50.29383	50
camcol	3.64827	1.666068	50
field	302.3935	162.5536	50
specobjid	1645881000000000000	2013805000000000000	50
redshift	0.1436843	0.388743	50
plate	1461.867	1788.82	50
mjd	52944.47	1511.341	50
fiberid	352.7553	206.5167	32

ii. Analyse the class variable using appropriate statistics and visualisations.

From Figure 1, we can know that the data can be classified into 3 categories: GALAXY, QSO and STAR. Most of the samples belong to the category GALAXY, and STAR has the second highest number of samples, while QSO has the least number of samples.

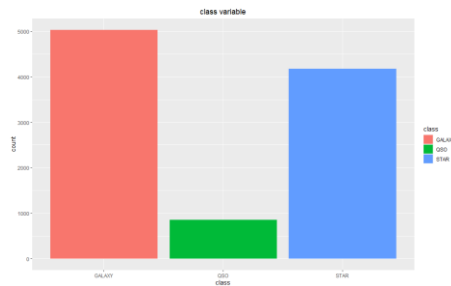


Figure 1 Histogram of class variable distribution

From Figure 2, I am seeing that 50% of the samples belong to the category GALAXY, 42% of the samples belong to the category STAR, and 8% of the samples belong to the category QSO.

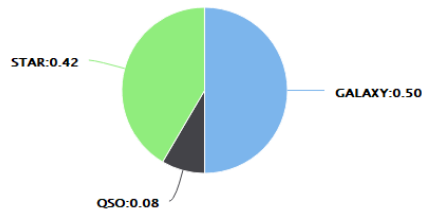


Figure 2 The Pie Chart of the class

iii.

Because histograms can only be used to represent continuous data instead of categorical data, I didn't use histograms to describe objid, dia, rerun, native, camcol.

As data are numerical, they are naturally ordered from lowest to highest in all the histograms. Firstly, I use the R software to automatically bin the data (most of the bin width are generated by the R software), and the height of each bar represents the count of values in the corresponding bin. The shape of the distribution is described as followed:

1) From the histogram, I am seeing that the distribution is unimodal. Now the distribution is not symmetric, rather it is slightly positively skewed.

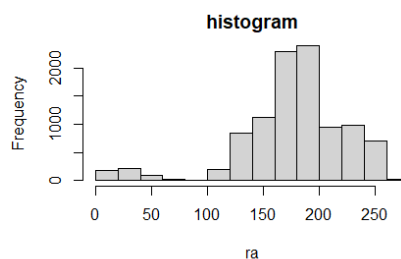


Figure 3 Histogram of ra

2) From the histogram, I am seeing that the distribution is a unimodal distribution. The distribution is negatively skewed.

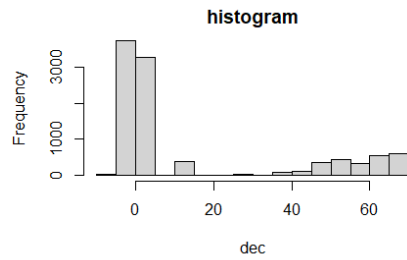


Figure 4 Histogram of dec

3) From the histogram, I am seeing that the distribution is a unimodal distribution. The distribution is positively skewed.

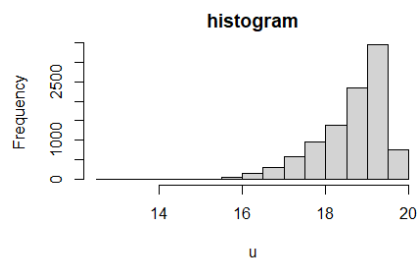


Figure 5 Histogram of u

4) From the histogram, I am seeing that the distribution is a unimodal distribution. The distribution is positively skewed.

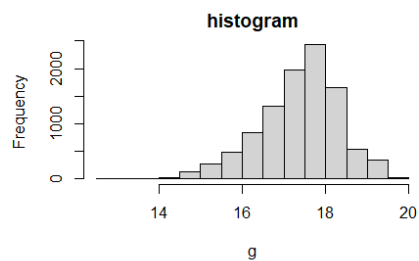


Figure 6 Histogram of g

5) From the histogram, I am seeing that the distribution is a unimodal distribution. The distribution is symmetric.

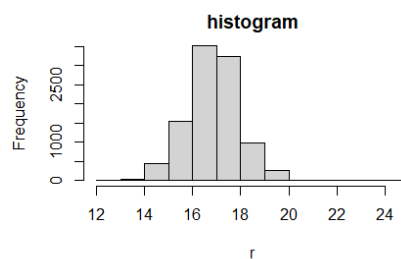


Figure 7 Histogram of r

6) From the histogram, I am seeing that the distribution is a unimodal distribution. The distribution is symmetric.

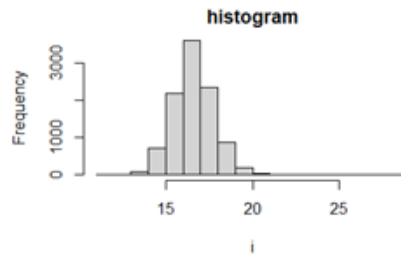


Figure 8 Histogram of i

7) From Figure 9, I am seeing that the distribution is a unimodal distribution. The distribution is symmetric.

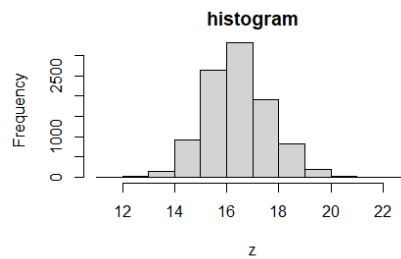


Figure 9 Histogram of z

8) From Figure 10, I am seeing that the distribution is a bimodal distribution. The distribution is negatively skewed.

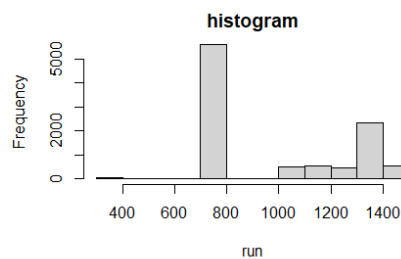


Figure 10 histogram of run

9) From Figure 11, I am seeing that the distribution is a bimodal distribution. The distribution is Symmetric.

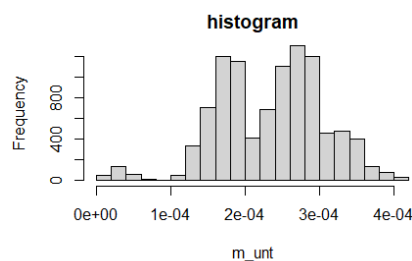


Figure 11 Histogram of m_unt

10) From Figure 12, I am seeing that the distribution is a unimodal distribution. The distribution is positively skewed.

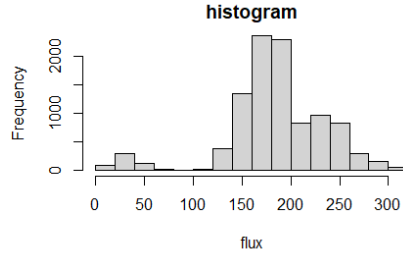


Figure 12 Histogram of flux

11) From Figure 13, I am seeing that the distribution is a unimodal distribution. The distribution is negatively skewed.

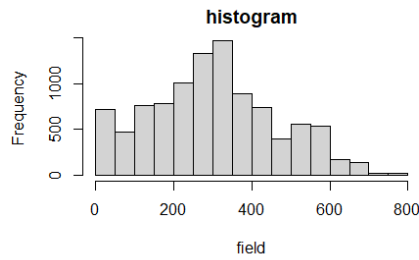


Figure 13 Histogram of field

12) From Figure 14, I am seeing that the distribution is a unimodal distribution. The distribution is negatively skewed.

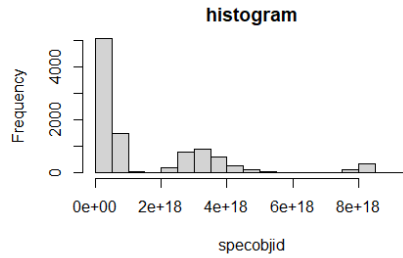


Figure 14 Histogram of specobjid

13) From Figure 15, I am seeing that the distribution is a unimodal distribution. The distribution is negatively skewed.

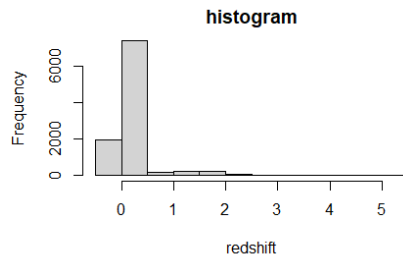


Figure 15 Histogram of redshift

14) From Figure 16, I am seeing that the distribution is a unimodal distribution. The distribution is negatively skewed.

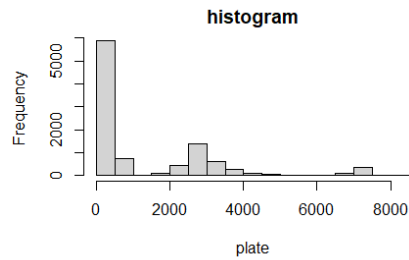


Figure 16 Histogram of plate

15) From Figure 17, I am seeing that the distribution is a unimodal distribution. The distribution is negatively skewed.

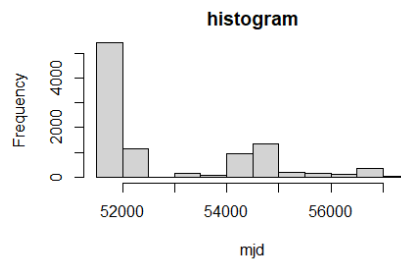


Figure 17 Histogram of mjd

16) From the histogram, I am seeing that the distribution is uniform.

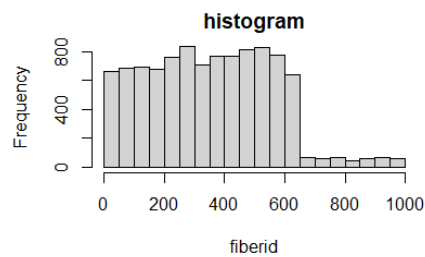


Figure 18 Histogram of fibered

2. Explore the relationships between data

i.

The correlation coefficient is 0.9581076, which is very close to 1. There is strong correlation between r and g. Also, the correlation is positive correlation. The scatterplot for the variables r and g is shown in Figure 19. From the scatterplot, I am seeing that there is a linear correlation between r and g.

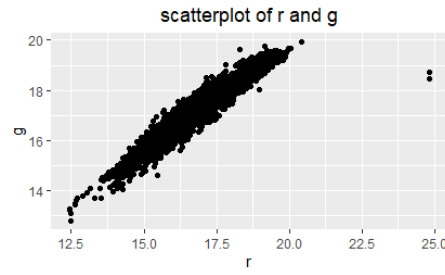


Figure 19 scatterplot of r and g

ii.

The correlation coefficient is -0.009189361, which is very close to 0. There is no correlation between r and mjd. The scatterplot for the variables mjd and r is shown in Figure 19.

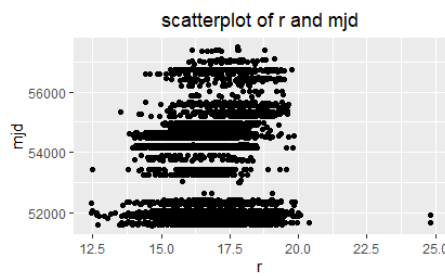


Figure 20 Scatterplot of r and mjd

iii.

From Figure 21, I am seeing that there is no correlation between u and redshift. From Figure 22, I am seeing that there is no correlation between z and redshift. From Figure 23, I am seeing that there is a moderate positive correlation between u and z.

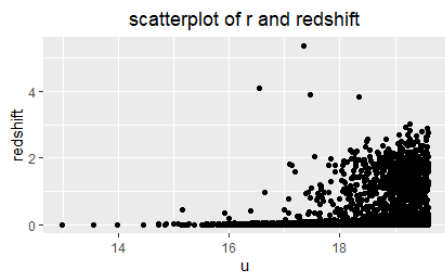


Figure 21 Scatterplot of r and redshift

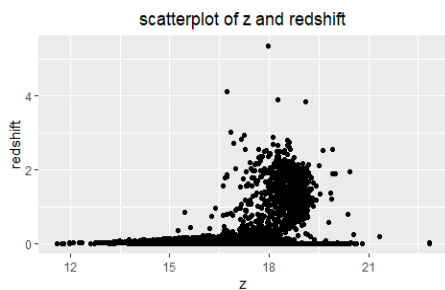


Figure 22 Scatterplot of z and redshift

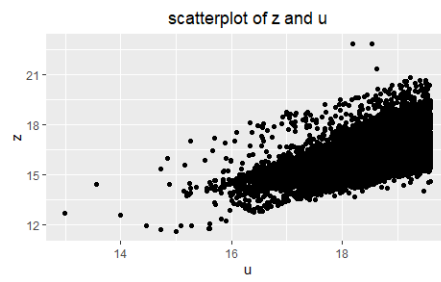
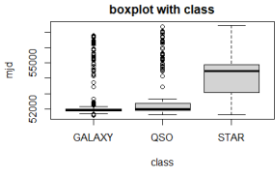



Figure 23 Scatterplot of z and u

iv.

Table 2

ra:		dec:	
u:		g:	
r:		i:	
Z:		m_unt:	
flux:		camcol:	
field:		specobjid:	
redshift:		plate:	

mjd:		fiberid:	
------	---	----------	--

3 General Conclusions ^[5]

The attribute rerun is insignificant, because the dispersion of this attribute is 0, which means that the values in this feature are all the same. The feature objid is insignificant, because the dispersion of this attribute is 0, which means that the values in this feature are all the same. The attribute native is insignificant, because the eigenvalues of the feature are only 0 and 1. The attribute dia is insignificant, because there are 6646 missing values in this attribute, which means that more than half of the values in this attribute are missing values.

From the boxplots, I am seeing that the attribute m_un, camcol, field, fibered may be insignificant, because the distribution intervals of three classes are almost the same in these attributes separately.

The attribute u is significant, because the distribution interval of the class STAR in u is different from the other two classes, so we can use u to make classifications between the class STAR and the other two classes. The attributes g, r, i, z are significant, because in all these 4 attributes, the distribution interval of the class QSO is different from the other two classes. We can use these four attributes as features to make classifications between the class QSO and the other two classes.

4 Dealing with missing values in R ^[6]

Define:

Replacing missing value with 0 means that using 0 to fill all the missing values that we want to fill within the database.

Replacing missing value with mean means that for a missing quantitative data, we replace it with the mean of the other data in the attribute that the missing value belongs to..

Replacing missing value with median means that for a missing quantitative data, we replace it with the median of the other data in the attribute that the missing value belongs to.

Compare:

Generally, if the distribution of the feature is a normal distribution, the effect of replacing missing values with mean is better. When the distribution not a normal distribution due to the existence of outlier values, the effect of replacing missing values with median or 0 is better.

Contrast:

Although these three methods are simple, they may introduce noise to the data or change the original distribution of features.

Their effects on the data:

After I replace the missing value with 0, the min of most of the attributes with missing value becomes 0, the average and median of the data in these attributes are reduced. In Figure 24, Figure 25, and Figure 26, I can see that the missing value distribution of attribute i is not random, but concentrated in the interval near the middle, so the original distribution has not changed by replacing missing value with median or mean.

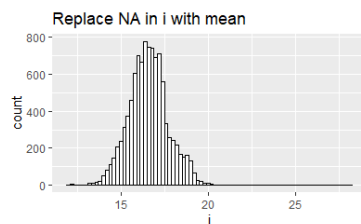


Figure 24 The distribution of data after replacing NA with mean

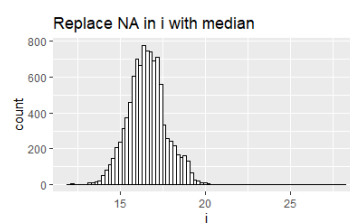


Figure 25 The distribution of data after replacing NA with median

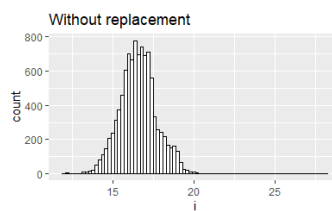


Figure 26 The distribution of the original data

5 Attribute transformation

define:

Mean centering is the act of subtracting a variable's mean from all observations on that variable in the dataset such that the variable's new mean is zero.

Normalisation is a technology that used to limit the data to a certain range after processing it with a certain algorithm. The purpose of Normalisation is to reduce data redundancy.

The standardization of data is to scale the data to a small specific interval.

Contrast:

- 1) Normalization, standardization are similar. They both pre-process the data so that the values fall into a uniform value range, so that in the modeling process, each feature quantity is treated the same.
- 2) These three methods can eliminate the bias caused by some numerical differences. The transformed data can speed up the training speed and promote the convergence of the algorithm.

Compare: Normalization is generally to limit the data to the required range, such as $[0, 1]$, thereby eliminating the influence of data dimensions on modeling. Standardization generally refers to normalizing the data so that the mean value 1 and the variance are 0. Mean centering is the act of subtracting a variable's mean from all observations on that variable in the dataset such that the variable's new mean is zero.

Effect on the data:

The mean centering set the mean of the dataset to 0.

The normalization limits the range of data to $[0,1]$

After the standardization, the mean of the dataset becomes 1 and the variance becomes 0.

.

6 Attribute/instance selection

- i. First, I delete the column dia, because the proportion of missing data in this column is greater than 80%. Then I deleted the columns objid and rerun, because the data in these two columns are duplicate data. Then I delete the rows with missing data. In the end, I only kept one row for the rows with duplicate data,

but no rows were deleted in this operation.

The available sample are reduced by almost half, which will affect the prediction results.

ii.

From Figure 27, I am seeing that the correlation between mjd and fiberid is very high.

Also, the correlation between u and mjd is very high. Because of this, I delete the attribute

mjd,

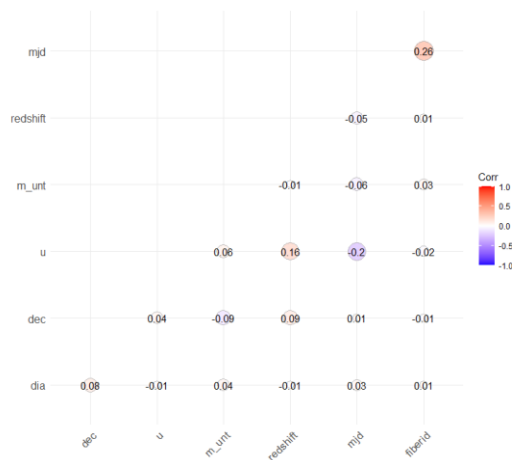


Figure 27, the correlation between different attributes.

7 Attribute transformation/reduction

i.

I selected attributes ra,dec,u,g,r,i,z,m_unit,flux,field,redshift,plate,fiberid to build a new database. I replaced the missing values in the new database with the mean value.

PCA is a mathematical method of dimensionality reduction. Using orthogonal transformation, linearly dependent variables may become linearly uncorrelated variables, that is, principal components.

And PCA as a dimension reduction tool is to reduce the complexity of a model and avoid overfitting.

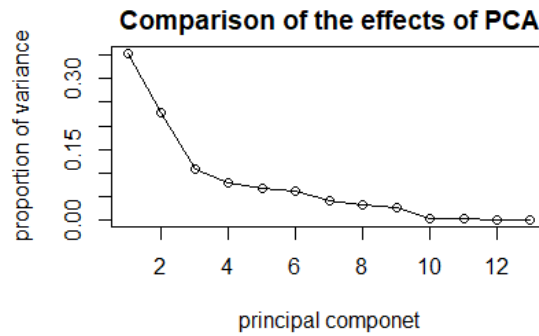


Figure 28 Comparison of the effects of PCA

ii.

According to Figure 28 ,7 PCs should be used to obtain a cumulative variance of at least 90%.

PC1	PC2	PC3	PC4	PC5
0.35491	0.58263	0.69017	0.76783	0.83486
PC6	PC7	PC8	PC9	PC10
0.89480	0.93590	0.96882	0.99569	0.99726
PC11	PC12	PC13		
0.99849	0.99954	1.00000		

Figure 29 cumulative variance of different number of PCs

2 CLUSTERING

2.1 I used the transformed dataset featuring 12 Principal Components.

I used the `cluster.stats()` function to calculate the dunn index and `avg.silwidth` to evaluate which algorithm produces better results. I divided the minimum of the pairwise distance by the maximal intra-cluster distance to get the dunn index. I compared the stability of the resulting clusters based on average silhouettes distance. The higher the dunn index and the average silhouettes distance, the better the cluster[1].

Table 3 `avg.silwidth` represent the cluster average silhouette widths.

	Dunn index	Avg.silwidth
K-means	0.0009711597	0.1879026
HCA		
PAM		

2.2 I optimised the kmeans algorithm.

I selected the parameter iter.max and algorithm. iter.max is the maximum number of iterations allowed, algorithm is the algorithm used. And I used grid search to find the best result, because grid-search is a popular method to find the optimal hyperparameters of a model.

Table 4

	Iter.max	algorithm	Dunn index	Avg.silwidth
1	5	Hartigan-Wong	0.0008373118	0.1661997
2	10	Hartigan-Wong	0.002265518	0.2285436
3	5	Lloyd	0.002059701	0.1944481
4	10	Lloyd	0.001946776	0.2285066
5	5	Macqueen	0.001546564	0.1585743
6	10	Macqueen	0.001240095	0.2280544

I found that setting inter.max to be 10 and set algorithm to be Hartigan-Wong can achieve the best clustering among the 6 experiments.

2.3

Table 5 The performance of K-means of different datasets

	Dunn index	Avg.silwidth
i. The transformed dataset featuring all Principal Components	0.0016008	0.3700917
ii.The reduced dataset featuring 12 Principal Components	0.002265518	0.2285436
iii.The dataset after deletion of instances and attributes	0.0004867355	0.356427
iv mean-centered median	0.0001984186	0.3558209
v mean-centered mean	0.0005206894	0.3499976
vi mean-centered 0	0.0003837383	0.3548755

The reduced dataset featuring 12 Principal Components and the transformed dataset featuring all Principal Components have a positive impact on the quality of the clustering. Because the dunn index of the results generated by the these two datasets are much higher than other databases.

3 CLASSIFICATION

3.1 Because many datasets have been generated in part1, and I made predictions first using J48 based on several datasets. The results are as followed:

Table 6 Accuracy of all data sets

Data Name	Accuracy
1The reduced dataset featuring 12 principal components	89.1378%
2The dataset in which the missing data are replaced by 0	98.6868%
3The dataset in which the missing data are replaced by mean	99.5623%
4The dataset in which the missing data are replaced by median	99.5623%

Accept the prediction based on the dataset featuring 12 principal components, the accuracy of the prediction made based on the other dataset is too high, which is cause by the overfitting problem. So I use the dataset featuring 12 Principal Components.

I used Kappa, F1 Score, Confusion matrix and accuracy to evaluate the result. TP means the number of True Positives, TN means the number of True Negatives, FP means the number of False Positives, FN means the number of False Negatives. The Confusion Matrix is defined as:

$$cm = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}. \quad (1)$$

Based on the Confusion Matrix, the precision and recall are defined as:

$$\text{Precision} = TP/(TP+FP) \quad (2)$$

$$\text{Recall} = TP/(TP+FN) \quad (3)$$

The F1 Score is defined as:

$$F1 = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \quad (4)$$

The accuracy is defined as:

$$\text{Accuracy} = (TP+TN)/(TP+FP+FN+TN) \quad (5)$$

The Kappa coefficient is a comprehensive statistic that measures the classification model, which can be calculated by the confusion matrix:

$$\text{Kappa} = \frac{p_o - p_e}{1 - p_e}. \quad (6)$$

p_o is the overall classification accuracy rate, namely Accuracy.

According to the confusion matrix, assuming that the number of real samples for each type is $\{a_1, a_2, \dots, a_n\}$, the number of prediction samples is $\{b_1, b_2, \dots, b_n\}$, and p_e is defined as:

$$p_e = \frac{a_1 \times b_1 + a_2 \times b_2 + \dots + a_n \times b_n}{n \times n} \quad (7)$$

Based the accuracy rate, Kappa, F1 Score and other indicators, the model is verified using 10-fold:

Table 7 Prediction based on the reduced dataset featuring 12 principal components

Classifier name	Accuracy	F1 Score	Kappa
ZeroR	49.98%	0	0
OneR	73.22%	0.700	0.501
NaïveBayes	84.59%	0.845	0.7243
IBk(5-NN)	87.27%	0.872	0.7725
J48	93.13%	0.891	0.8089

Table 8 Confusion matrix

<p>=== ZeroR Confusion Matrix ===</p> <p>a b c <-- classified as</p> <p>4997 0 0 a = GALAXY</p> <p>847 0 0 b = QSO</p> <p>4154 0 0 c = STAR</p>	<p>=== OneR Confusion Matrix ===</p> <p>a b c <-- classified as</p> <p>4302 8 687 a = GALAXY</p> <p>410 5 432 b = QSO</p> <p>1116 24 3014 c = STAR</p>
<p>=== NaïveBayes Confusion Matrix ===</p> <p>a b c <-- classified as</p> <p>4591 3 403 a = GALAXY</p> <p>100 646 101 b = QSO</p> <p>931 2 3221 c = STAR</p>	<p>=== IBk Confusion Matrix ===</p> <p>a b c <-- classified as</p> <p>4702 7 288 a = GALAXY</p> <p>146 630 71 b = QSO</p> <p>739 21 3394 c = STAR</p>
<p>=== J48 Confusion Matrix ===</p> <p>a b c <-- classified as</p> <p>4516 30 451 a = GALAXY</p> <p>55 749 43 b = QSO</p> <p>487 20 3647 c = STAR</p>	

As can be seen from Table 2 and Table 3,

- 1) ZeroR classifier classifies all the data into one category (confusion matrix), so Kappa and F1 are all 0;
- 2) OneR classifier has 5 correct QSO categories;
- 3) NaïveBayes is better than ZeroR and OneR;
- 4) The best classifier is J48, the accuracy is 0.9313, the F1 Score is 0.891, and the kappa is 0.80

3.2 At this stage, the data used in 3.1 is optimized by adjusting the train/test split percentage, number of neighbours and distance metrics. The three parameters are set as follows:

<p>1: {25%, 5, Chebyshe }</p> <p>Accuracy: 83.31, Precision: 0.844, ecall: 0.833</p> <p>=== Confusion Matrix ===</p> <table><tr><td>a</td><td>b</td><td>c</td><td><-- classified as</td></tr><tr><td>3498</td><td>10</td><td>213</td><td>a = GALAXY</td></tr><tr><td>177</td><td>404</td><td>64</td><td>b = QSO</td></tr><tr><td>701</td><td>27</td><td>2404</td><td>c = STAR</td></tr></table>	a	b	c	<-- classified as	3498	10	213	a = GALAXY	177	404	64	b = QSO	701	27	2404	c = STAR	<p>2: {25%, 5, Euclidean }</p> <p>Accuracy: 84.10, recision: 0.850, ecall: 0.841</p> <p>=== Confusion Matrix ===</p> <table><tr><td>a</td><td>b</td><td>c</td><td><-- classified as</td></tr><tr><td>3498</td><td>10</td><td>213</td><td>a = GALAXY</td></tr><tr><td>177</td><td>404</td><td>64</td><td>b = QSO</td></tr><tr><td>701</td><td>27</td><td>2404</td><td>c = STAR</td></tr></table>	a	b	c	<-- classified as	3498	10	213	a = GALAXY	177	404	64	b = QSO	701	27	2404	c = STAR
a	b	c	<-- classified as																														
3498	10	213	a = GALAXY																														
177	404	64	b = QSO																														
701	27	2404	c = STAR																														
a	b	c	<-- classified as																														
3498	10	213	a = GALAXY																														
177	404	64	b = QSO																														
701	27	2404	c = STAR																														
<p>3: {25%, 5, Manhattan }</p> <p>Accuracy: 82.2219, precision: 0.835, recall: 0.822</p> <p>=== Confusion Matrix ===</p> <table><tr><td>a</td><td>b</td><td>c</td><td><-- classified as</td></tr><tr><td>3499</td><td>7</td><td>215</td><td>a = GALAXY</td></tr><tr><td>255</td><td>319</td><td>71</td><td>b = QSO</td></tr><tr><td>756</td><td>29</td><td>2347</td><td>c = STAR</td></tr></table>	a	b	c	<-- classified as	3499	7	215	a = GALAXY	255	319	71	b = QSO	756	29	2347	c = STAR	<p>4: {25%, 10, Chebyshe }</p> <p>Accuracy: 81.235, precision: 0.834, recall: 0.812</p> <p>=== Confusion Matrix ===</p> <table><tr><td>a</td><td>b</td><td>c</td><td><-- classified as</td></tr><tr><td>3550</td><td>5</td><td>166</td><td>a = GALAXY</td></tr><tr><td>243</td><td>317</td><td>85</td><td>b = QSO</td></tr><tr><td>902</td><td>6</td><td>2224</td><td>c = STAR</td></tr></table>	a	b	c	<-- classified as	3550	5	166	a = GALAXY	243	317	85	b = QSO	902	6	2224	c = STAR
a	b	c	<-- classified as																														
3499	7	215	a = GALAXY																														
255	319	71	b = QSO																														
756	29	2347	c = STAR																														
a	b	c	<-- classified as																														
3550	5	166	a = GALAXY																														
243	317	85	b = QSO																														
902	6	2224	c = STAR																														
<p>5: {25%, 10, Euclidean }</p> <p>Accuracy: 81.155, precision: 0.838, recall: 0.812</p> <p>=== Confusion Matrix ===</p> <table><tr><td>a</td><td>b</td><td>c</td><td><-- classified as</td></tr><tr><td>3578</td><td>3</td><td>140</td><td>a = GALAXY</td></tr><tr><td>276</td><td>310</td><td>59</td><td>b = QSO</td></tr><tr><td>931</td><td>4</td><td>2197</td><td>c = STAR</td></tr></table>	a	b	c	<-- classified as	3578	3	140	a = GALAXY	276	310	59	b = QSO	931	4	2197	c = STAR	<p>6: {25%, 5, Manhattan }</p> <p>Accuracy: 80.3681, precision: 0.834, recall: 0.804</p> <p>=== Confusion Matrix ===</p> <table><tr><td>a</td><td>b</td><td>c</td><td><-- classified as</td></tr><tr><td>3586</td><td>1</td><td>134</td><td>a = GALAXY</td></tr><tr><td>320</td><td>255</td><td>70</td><td>b = QSO</td></tr><tr><td>944</td><td>3</td><td>2185</td><td>c = STAR</td></tr></table>	a	b	c	<-- classified as	3586	1	134	a = GALAXY	320	255	70	b = QSO	944	3	2185	c = STAR
a	b	c	<-- classified as																														
3578	3	140	a = GALAXY																														
276	310	59	b = QSO																														
931	4	2197	c = STAR																														
a	b	c	<-- classified as																														
3586	1	134	a = GALAXY																														
320	255	70	b = QSO																														
944	3	2185	c = STAR																														
<p>7: {25%, 15, Chebyshe }</p> <p>Accuracy: 80.2614, precision: 0.826, recall: 0.803</p> <p>=== Confusion Matrix ===</p> <table><tr><td>a</td><td>b</td><td>c</td><td><-- classified as</td></tr><tr><td>3530</td><td>2</td><td>189</td><td>a = GALAXY</td></tr><tr><td>271</td><td>281</td><td>93</td><td>b = QSO</td></tr><tr><td>924</td><td>1</td><td>2207</td><td>c = STAR</td></tr></table>	a	b	c	<-- classified as	3530	2	189	a = GALAXY	271	281	93	b = QSO	924	1	2207	c = STAR	<p>8: {25%, 15, Euclidean }</p> <p>Accuracy: 80.128, precision: 0.829, recall: 0.801</p> <p>=== Confusion Matrix ===</p> <table><tr><td>a</td><td>b</td><td>c</td><td><-- classified as</td></tr><tr><td>3560</td><td>2</td><td>159</td><td>a = GALAXY</td></tr><tr><td>303</td><td>262</td><td>80</td><td>b = QSO</td></tr><tr><td>946</td><td>0</td><td>2186</td><td>c = STAR</td></tr></table>	a	b	c	<-- classified as	3560	2	159	a = GALAXY	303	262	80	b = QSO	946	0	2186	c = STAR
a	b	c	<-- classified as																														
3530	2	189	a = GALAXY																														
271	281	93	b = QSO																														
924	1	2207	c = STAR																														
a	b	c	<-- classified as																														
3560	2	159	a = GALAXY																														
303	262	80	b = QSO																														
946	0	2186	c = STAR																														
<p>9: {25%, 15, Manhattan }</p> <p>Accuracy: 79.9547, precision: 0.831, recall: 0.800</p> <p>=== Confusion Matrix ===</p>	<p>10: {50%, 5, Chebyshe }</p> <p>Accuracy: 85.237, precision: 0.858, recall: 0.852</p> <p>=== Confusion Matrix ===</p>																																

<pre> a b c <-- classified as 3578 1 142 a = GALAXY 348 219 78 b = QSO 934 0 2198 c = STAR </pre>	<pre> a b c <-- classified as 2317 5 162 a = GALAXY 98 301 35 b = QSO 420 18 1643 c = STAR </pre>
<pre> 11: {50%, 5, Euclidean } Accuracy: 85.237, precision: 0.858, recall: 0.852 === Confusion Matrix === a b c <-- classified as 2314 4 166 a = GALAXY 99 296 39 b = QSO 415 15 1651 c = STAR </pre>	<pre> 12: {50%, 5, Manhattan } Accuracy: 84.33, precision: 0.851, recall: 0.843 === Confusion Matrix === a b c <-- classified as 2325 3 156 a = GALAXY 129 259 46 b = QSO 430 19 1632 c = STAR </pre>
<pre> 13: {50%, 10, Chebyshe } Accuracy: 83.79, precision: 0.854, recall: 0.838 === Confusion Matrix === a b c <-- classified as 2370 1 113 a = GALAXY 142 255 37 b = QSO 511 6 1564 c = STAR </pre>	<pre> 14: {50%, 10, Euclidean } Accuracy: 84.01, precision: 0.852, recall: 0.840 === Confusion Matrix === a b c <-- classified as 2349 3 132 a = GALAXY 116 269 49 b = QSO 494 5 1582 c = STAR </pre>
<pre> 15: {50%, 10, Manhattan } Accuracy: 82.8, precision: 0.848, recall: 0.828 === Confusion Matrix === a b c <-- classified as 2376 1 107 a = GALAXY 170 221 43 b = QSO 535 3 1543 c = STAR </pre>	<pre> 16: {50%, 15, Chebyshe } Accuracy: 83.43, precision: 0.849, recall: 0.834 === Confusion Matrix === a b c <-- classified as 2345 1 138 a = GALAXY 146 244 44 b = QSO 497 2 1582 c = STAR </pre>
<pre> 17: {50%, 15, Euclidean } Accuracy: 83.27, precision: 0.850, recall: 0.833 === Confusion Matrix === a b c <-- classified as 2367 2 115 a = GALAXY 160 226 48 b = QSO 509 2 1570 c = STAR </pre>	<pre> 18: {50%, 15, Manhattan } Accuracy: 82.51, precision: 0.847, recall: 0.825 === Confusion Matrix === a b c <-- classified as 2367 2 115 a = GALAXY 160 226 48 b = QSO 509 2 1570 c = STAR </pre>

It can be seen from the above results that when {train/test split percentage, number of neighbours, distance metrics} are {50%, 5, Euclidean}, Accuracy: 85.237, precision: 0.858 , Recall: 0.852 up to the highest

3.3

Table 9 Classification result of J48

Classifier name	Accuracy	F1 Score	Kappa
i, all Principal Component	93.8712 %	0.939	0.8924
ii, 12 Principal Components	93.2014 %	0.932	0.8808
iii. Deletion of instances and attributes	99.52%	0.995	0.9916
iv. all Principal Components of the normalised	93.8212 %	0.938	0.8916
v Principal Components of the normalised	93.1914 %	0.932	0.8806
vi Deletion of instances and attributes of the normalised	99.10%	0.991	0.9842

conclusion

- 1) It can be seen from Table 4 that there is an over-fitting phenomenon in the data that has only been deleted by attributes and instances, but the data processed by PCA has good model performance.
- 2) The accuracy of PCA using all Principle Components is higher than that using 12 Principle Components.
- 3) The performance of normalised data and unprocessed data is not improved much, but normalised can better solve the problem of overfitting.
- 4) The best data set is (iv)all Principal Components of the normalised, because it does not have the problem of overfitting.

REFERENCE:

- [1] Rousseeuw, J. Peter (1987). Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics 20: 53–65. doi:10.1016/0377-0427(87)90125-7.

Code

```
library(tidyverse)
```

```
library(MASS)
```

```
library(naniar)
```

```
library(ggcorrplot)
```

```
library(readxl)
```

```
library(corrplot)
```

```
library(Hmisc)
```

```
dat<-read.csv("cw_data.csv", header = TRUE)
```

```
head(dat)
```

```
dim(dat)
```

Explore the data

```
dat.numeric <- subset(dat,select = -c(class))
```

```
summary(dat.numeric)
```

```
mean = as.vector(sapply(dat.numeric, mean, na.rm = TRUE))
```

```
dispersion = as.vector(sapply(dat.numeric, sd, na.rm = TRUE))
```

```
dat_basic = cbind( mean)
```

```
dat_basic = cbind(dat_basic,dispersion)
```

```
num_na = c(0,6646,30,48,49,50,51,53,50,53,50,46,50,50,50,50,50,50,50,32)
```

```
mean
```

```
dispersion
```

#1, ii. Analyse the class variable using appropriate statistics and visualisations.

```
attach(dat)
```

```

dat%>%
  group_by(class)%>%
  summarise(count = n())%>%
  ggplot(.,aes(class,count,fill = class))+
  geom_bar(stat = "identity")+
  labs(title = "class variable")+
  theme(plot.title = element_text(hjust = 0.5))

```

#draw the pie chart

```

datpie<-dat%>%
  group_by(class)%>%
  summarise(count = n())%>%
  mutate(freq = count/sum(count))

```

```
library(highcharter)
```

```
hchart(datpie,"pie",hcaes(x = "class",y = "freq"))%>%
```

```
  hc_plotOptions(pie = list(dataLabels = list(enabled = TRUE,
```

```
                                format =
```

```
                                '{point.class}: {point.freq:.2f} %}'))
```

1.3

```
dim(dat)
```

```
par(mfrow = c(1, 1))
```

```
for (i in 1:21){
```

```
  hist(dat[[i]], main = "histogram", xlab = colnames(dat)[i])
```

```
}
```

```
summary(dat)
```

2,

```
cor.r.g <- cor(r, g,use = "pairwise.complete.obs",method = "pearson")
```

```
cor.r.g

ggplot(dat,aes(r,g))+
  geom_point()+
  labs(title = "scatterplot of r and g",
        xlab = "r", ylab = "g")+
  theme(plot.title = element_text(hjust = 0.5))

cor.r.g = cor(r, g, use = "pairwise.complete.obs", method = "pearson");cor.r.g
plot(r, g, xlab = "r", ylab = "g", main = "scatterplot of r and g")
```

```
cor.r.mjd <- cor(r, mjd, use = "pairwise.complete.obs", method = "pearson")
cor.r.mjd

ggplot(dat,aes(r,mjd))+
  geom_point()+
  labs(title = "scatterplot of r and mjd",
        xlab = "r", ylab = "mjd")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
cor.r.mjd = cor(r, mjd, use = "pairwise.complete.obs", method = "pearson");cor.r.mjd
plot(r, mjd, xlab = "r", ylab = "mjd", main = "scatterplot of r and mjd")
```

#1.2 iii --- Produce scatterplots between the class variable and u, z, and redshift. What do these three scatterplots tell you about the relationships between these variables and the class?

```
par(mfrow = c(1,1))
ggplot(dat,aes(u,redshift))+
  geom_point()+
  labs(title = "scatterplot of r and redshift",
        xlab = "u", ylab = "redshift")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
ggplot(dat,aes(z,redshift))+
  geom_point()+
  labs(title = "scatterplot of z and redshift",
        xlab = "z", ylab = "redshift")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
ggplot(dat,aes(u,z))+
  geom_point()+
  labs(title = "scatterplot of z and u",
        xlab = "z", ylab = "u")+
  theme(plot.title = element_text(hjust = 0.5))
```

#1.2 iv.--- Produce boxplots for all of the appropriate attributes in the dataset grouping each variable according to the class attribute

```
for (i in 1:21){
  boxplot(dat[[i]]~class, main = "boxplot with class", ylab = colnames(dat)[i])
}
```

##1.4--- Dealing with missing values in R

#Replace missing values in the dataset using three strategies: replacement with 0, mean and median. Define, compare and contrast these approaches, and explain their effects on the data. For mean and median replacement, take the class of the instances into consideration.

```
dat.0 <- dat
dat.0[is.na(dat.0)] <- 0
```



```

summary(dat.0)

dispersion = as.vector(sapply(dat.0, sd, na.rm = TRUE))

dispersion

dat.mean <- dat
for (i in 1:21){
  dat.mean[[i]][is.na(dat.mean[[i]])] = mean(dat.mean[[i]], na.rm = TRUE)
}
summary(dat.mean)

dat.median <- dat
for (i in 1:21){
  dat.median[[i]][is.na(dat.median[[i]])] = median(dat.median[[i]], na.rm = TRUE)
}
summary(dat.median)

attach(dat.mean)
ggplot(dat.mean, aes(x=i)) + geom_histogram(binwidth=.2,colour="black",
  fill="white")+
labs(title = "Replace NA in i with mean")

attach(dat.median)
ggplot(dat.median, aes(x=i)) + geom_histogram(binwidth=.2,colour="black",
  fill="white")+
labs(title = "Replace NA in i with median")

attach(dat)
ggplot(dat, aes(x=i)) + geom_histogram(binwidth=.2,colour="black",
  fill="white")+
labs(title = "Without replacement")

```

```

dat.0center = dat.0
for (i in 1:21){
  dat.0center[[i]] = scale(dat.0center[[i]], center = TRUE, scale = FALSE)
}

summary(dat.0center)

dat.meancenter = dat.mean
for (i in 1:21){
  dat.meancenter[[i]] = scale(dat.meancenter[[i]], center = TRUE, scale = FALSE)
}

summary(dat.meancenter)

dat.mediancenter = dat.median
for (i in 1:21){
  dat.mediancenter[[i]] = scale(dat.mediancenter[[i]], center = TRUE, scale = FALSE)
}

summary(dat.mediancenter)

dat.0stand <- dat.0
for (i in 1:21){
  dat.0stand[[i]] = scale(dat.0stand[[i]], center = TRUE, scale = TRUE)
}

summary(dat.0stand)

```

```

dat.meanstand <- dat.mean
for (i in 1:21){
  dat.meanstand[[i]] = scale(dat.meanstand[[i]], center = TRUE, scale = TRUE)
}
summary(dat.meanstand)

dat.medianstand <- dat.median
for (i in 1:21){
  dat.medianstand[[i]] = scale(dat.medianstand[[i]], center = TRUE, scale = TRUE)
}
summary(dat.medianstand)

dat.0normal <- dat.0
for (i in 1:21){
  dat.0normal[[i]] = (dat.0normal[[i]] - min(dat.0normal[[i]]
                                     ))/(max(dat.0normal[[i]]
                                     - min(dat.0normal[[i]])))
}
summary(dat.0normal)
```


```

dat.meannormal <- dat.mean
for (i in 1:21){
  dat.meannormal[[i]] = (dat.meannormal[[i]] - min(dat.meannormal[[i]]
                                     ))/(max(dat.meannormal[[i]]
                                     - min(dat.meannormal[[i]])))
}
summary(dat.meannormal)

ggplot(dat.mean, aes(x=i)) + geom_histogram(binwidth=.2,colour="black",

```


```

```

 fill="white")+
labs(title = "Replace NA in i with mean")
```

```{r}
dat.mediannormal <- dat.median
for (i in 1:21){
 dat.mediannormal[[i]] = (dat.mediannormal[[i]] - min(dat.mediannormal[[i]]
))/(max(dat.mediannormal[[i]]
 - min(dat.mediannormal[[i]]))
}
summary(dat.mediannormal)
```

```

6, Attribute/instance selection

#i. Starting again from the raw data, consider attribute and instance deletion strategies to deal with missing and duplicated values. Choose a number of missing values per instance or per attribute and delete instances or attributes accordingly. Explain your choices and its effects on the dataset.

```

```{r}
colSums(is.na(dat))
```

```

```

```{r}
dat.del<-dat%>%
 dplyr::select(-c(dia, objid, rerun))
dat.del<-dat.del[1:10005,]

```

```
```
```

```
```{r}
```

```
dat.del<-na.omit(dat.del)
```

```
count(dat.del)
```

```
count(dat.del)
```

```
```
```

#ii. Start from the raw data, use correlations between attributes to reduce the number of attributes. Try to reduce the dataset to contain only uncorrelated attributes and no missing values. Explain your choices and its effects on the dataset.

```
```{r}
```

```
dat.num<-dat%>%
```

```
 dplyr::select(dia, rerun, ra, dec, u, g, r, i, z, m_unt, flux, redshift, mjd)
```

```
cor_data = dat[, -c(13,22)]
```

```
correlation = cor(cor_data, method = "pearson", use = "complete.obs")
```

```
round(correlation, 2)
```

```
corrplot(correlation, method = 'circle')
```

```
p_corr = rcorr(as.matrix(cor_data), type = c("pearson", "spearman"))
```

```
round(p_corr$P, 4)
```

```
dat.cor_rm = subset(dat, select = -c(g, r, i, z, plate, specobjid, flux, field, ra, camcol, run))
```

```
cor_data = subset(dat.cor_rm, select = -c(native, class))
```

```
correlation = cor(cor_data, method = "pearson", use = "complete.obs")
```

```
round(correlation, 2)
```

```
corrplot(correlation, method = 'circle')
```

```
p_corr = rcorr(as.matrix(cor_data), type = c("pearson", "spearman"))
```

```
round(p_corr$P, 4)
```

```
```\n
```

```
## 1.7 Attribute transformation/ reduction
```

```
```\n{r}
```

```
datpca <- dat[which(rowMeans(!is.na(dat)) > 0.8), which(colMeans(!is.na(dat)) > 0.8)]
```

```
summary(datpca)
```

```
```\n
```

```
```\n{r}
```

```
#pcada<-as.data.frame(pca1$x)
```

```
#write.csv(datpca,"pcaclass.csv")
```

```
```\n
```

```
```\n{r}
```

```
library(tidyverse)
```

```
datpca<-datpca%>%
```

```
 dplyr::select(ra,dec,u,g,r,i,z,m_unt,flux,field,redshift,plate,fiberid)
```

```
for (i in 1:13){
```

```
 datpca[[i]][is.na(datpca[[i]])] = mean(datpca[[i]],na.rm=TRUE)
```

```
}
```

```
```\n
```

```
#dat1 <- factor(dat1)
```

```
#dat1 <- as.numeric(dat1)
```

```
```\n{r}
```

```

pca13 <- prcomp(datpca,center = TRUE,scale. = TRUE)
plot(summary(pca13)$importance[2,], ylab = "proportion of variance",
 xlab = "principal componet", type = "o",main="Comparison of the effects of PCA")
cum_var = summary(pca13)$importance[3,]
cum_var
summary(pca13)
```



```

```{r}
## 2
cum_var = summary(pca13)$importance[3,]
cum_var
```

```


```

Clustering

```
##2.1
```

```
```{r}
```

```
library(fpc)
```

```
```
```

```
```{r}
```

```
library(cluster)
```

```
```
```

```
```{r}
```

```

##HCA clustering

pca12<-pca13$x[,1:12]
'''

#HCA
'''{r}
hclustfunc <- function(x, method = "complete", dmeth = "euclidean") {
 hclust(dist(x, method = dmeth), method = method)
}
'''

'''{r}
fit <- hclustfunc(pca12)
summ(fit)
'''

'''{r}
a <- cluster.stats(dist(pca12),fit$cluster)
'''

'''{r}
#Split hierarchical clustering
da_diana<-diana(pcapart2)
plot(da_diana,which.plot = 2,main = "DIANA algorithm")
'''

2.2

```



```

```{r}
kmdt1 <- kmeans(pca12,3,iter.max = 5,algorithm = ("Hartigan-Wong"))
kmst1 <- cluster.stats(dist(pca12),kmdt1$cluster)

kmst1
```

#2.2

```{r}
kmdt2 <- kmeans(pca12,3,iter.max = 10,algorithm = ("Hartigan-Wong"))
kmst2 <- cluster.stats(dist(pca12),kmdt2$cluster)

kmst2
```

```{r}
kmdt3 <- kmeans(pca12,3,iter.max = 5,algorithm = ("Lloyd"))
kmst3 <- cluster.stats(dist(pca12),kmdt3$cluster)

kmst3
```

```

```

```{r}
kmdt4 <- kmeans(pca12,3,iter.max = 10,algorithm = ("Lloyd"))
kmst4 <- cluster.stats(dist(pca12),kmdt4$cluster)

```

```

kmst4
```
```{r}
kmdt5 <- kmeans(pca12,3,iter.max = 5,algorithm = ("MacQueen"))
kmst5 <- cluster.stats(dist(pca12),kmdt5$cluster)
kmst5
```

```{r}
kmdt6 <- kmeans(pca12,3,iter.max = 10,algorithm = ("MacQueen"))
kmst6 <- cluster.stats(dist(pca12),kmdt6$cluster)
kmst6
```

#2.3

```{r}
kmdt7 <- kmeans(datpca,3,iter.max = 10,algorithm = ("Hartigan-Wong"))
kmst7 <- cluster.stats(dist(datpca),kmdt7$cluster)
kmst7
```

```{r}
dat.mean<-dat.mean%>%
  dplyr::select(ra,dec,u,g,r,i,z,m_unt,flux,field,redshift,plate,fiberid)
kmdt8 <- kmeans(dat.mean,3,iter.max = 10,algorithm = ("Hartigan-Wong"))
kmst8 <- cluster.stats(dist(dat.mean),kmdt8$cluster)
kmst8

```

```

'''
'''{r}
dat.mediancenter<-dat.mediancenter%>%
  dplyr::select(ra,dec,u,g,r,i,z,m_unt,flux,field,redshift,plate,fiberid)
kmdt9 <- kmeans(dat.mediancenter,3,iter.max = 10,algorithm = ("Hartigan-Wong"))
kmst9 <- cluster.stats(dist(dat.mediancenter),kmdt9$cluster)
kmst9
'''
'''{r}
dat.meancenter<-dat.meancenter%>%
  dplyr::select(ra,dec,u,g,r,i,z,m_unt,flux,field,redshift,plate,fiberid)
kmdt10 <- kmeans(dat.meancenter,3,iter.max = 10,algorithm = ("Hartigan-Wong"))
kmst10 <- cluster.stats(dist(dat.meancenter),kmdt10$cluster)
kmst10
'''
'''{r}
dat.0center<-dat.0center%>%
  dplyr::select(ra,dec,u,g,r,i,z,m_unt,flux,field,redshift,plate,fiberid)
kmdt10 <- kmeans(dat.0center,3,iter.max = 10,algorithm = ("Hartigan-Wong"))
kmst10 <- cluster.stats(dist(dat.0center),kmdt10$cluster)
kmst10
'''

'''{r}
dat.del<-dat.del%>%
  dplyr::select(ra,dec,u,g,r,i,z,m_unt,flux,field,redshift,plate,fiberid)
kmdt11 <- kmeans(dat.del,3,iter.max = 10,algorithm = ("Hartigan-Wong"))
kmst11 <- cluster.stats(dist(dat.del),kmdt11$cluster)
kmst11

```

'''

'''