COMP4030
Coursework 2

20306231, Soomin Myung

CONTENTS

LIST OF FIGURES

Chapter 1: ANALYSIS AND PRE-PROCESSING

1. Exploring the data

i.

Figure 1. centrality, dispersion, and number of missing values of input features

objid	
Min	1.24e+18
1st quartile	1.24e+18
Median	1.24e+18
3rd quartile	1.24e+18
Max	1.24e+18
Mean	1.24e+18
SD	0
NA's	0

dia	
Min	28
1st quartile	229
Median	349
3rd quartile	693
Max	848172
Mean	2000
SD	22678
NA's	6646

rerun	
Min	301
1st quartile	301
Median	301
3rd quartile	301
Max	301
Mean	301
SD	0
NA's	30

ra	
Min	8.24
1st quartile	157.38
Median	180.44
3rd quartile	201.54
Max	260.88
Mean	175.54
SD	48
NA's	48

dec	
Min	-5.38
1st quartile	-0.54
Median	0.40
3rd quartile	35.57
Max	68.54
Mean	14.83
SD	25
NA's	49

u	
Min	12.99
1st quartile	18.18
Median	18.85
3rd quartile	19.26
Max	19.60
Mean	18.62
SD	0.827
NA's	59

g	
Min	12.8
1st quartile	16.8
Median	17.5
3rd quartile	18.0
Max	19.9
Mean	17.4
SD	0.95
NA's	51

r	
Min	12.4
1st quartile	16.2
Median	16.9
3rd quartile	17.5
Max	24.8
Mean	16.8
SD	1.1
NA's	53

i	
Min	11.9
1st quartile	15.8
Median	16.6
3rd quartile	17.3
Max	28.2
Mean	16.6
SD	1.1
NA's	50

z	
Min	11.6
1st quartile	15.6
Median	16.4
3rd quartile	17.1
Max	22.8
Mean	16.4
SD	1.2
NA's	53

run	
Min	308
1st quartile	752
Median	756
3rd quartile	1331
Max	1412
Mean	981
SD	273
NA's	50

m_unt	
Min	0.00001
1st quartile	0.00018
Median	0.00024
3rd quartile	0.00028
Max	0.00041
Mean	0.00023
SD	0.00007
NA's	46

native	
Min	0
1st quartile	0
Median	1
3rd quartile	1
Max	1
Mean	0.5
SD	0.5
NA's	50

flux	
Min	9.5
1st quartile	161.7
Median	183.3
3rd quartile	212.5
Max	319.1
Mean	183.5
SD	50
NA's	50

camcol	
Min	1
1st quartile	2
Median	4
3rd quartile	5
Max	6
Mean	3.65
SD	1.7
NA's	50

field	
Min	11
1st quartile	185
Median	299
3rd quartile	414
Max	768
Mean	302
SD	163
NA's	50

specobjid	
Min	3.00e+17
1st quartile	3.39e+17
Median	4.97e+17
3rd quartile	2.88e+18
Max	9.47e+18
Mean	1.65e+18
SD	2e+18
NA's	50

redshift	
Min	0
1st quartile	0
Median	0.04
3rd quartile	0.09
Max	5.35
Mean	0.14
SD	0.39
NA's	50

plate	
Min	266
1st quartile	301
Median	441
3rd quartile	2559
Max	8410
Mean	1462
SD	1789
NA's	50

mjd	
Min	51578
1st quartile	51900
Median	51997
3rd quartile	54468
Max	57481
Mean	52944
SD	1511
NA's	50

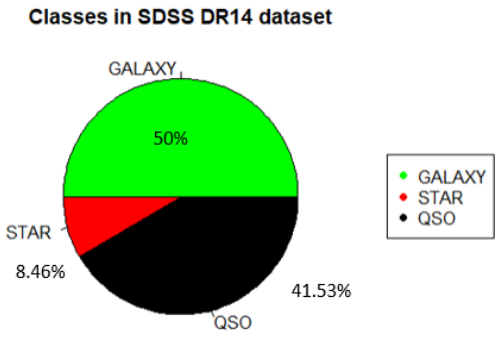
fiberid	
Min	1
1st quartile	186
Median	351
3rd quartile	510
Max	1000
Mean	353
SD	207
NA's	32

ii.

Figure 2. statistics of class variables

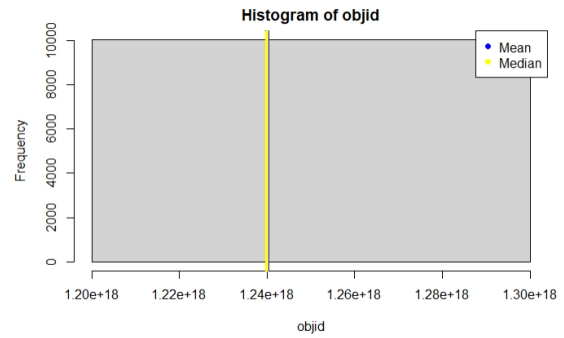
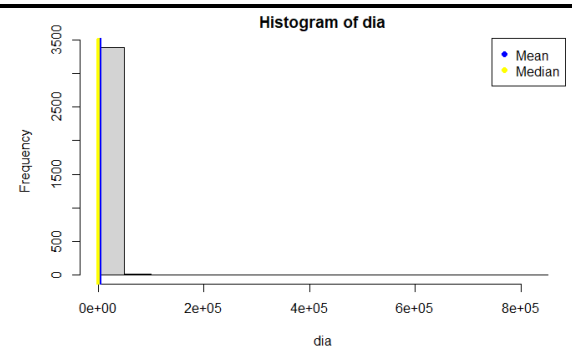
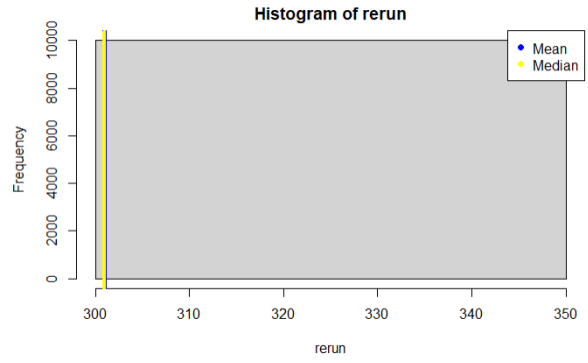
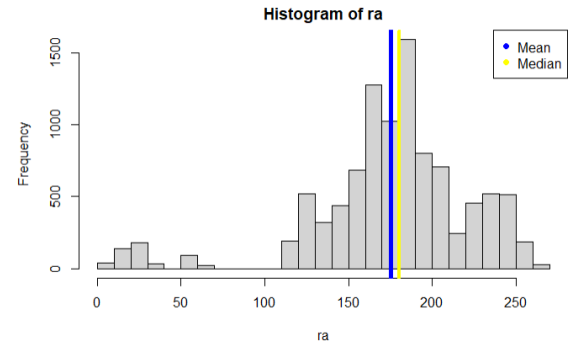
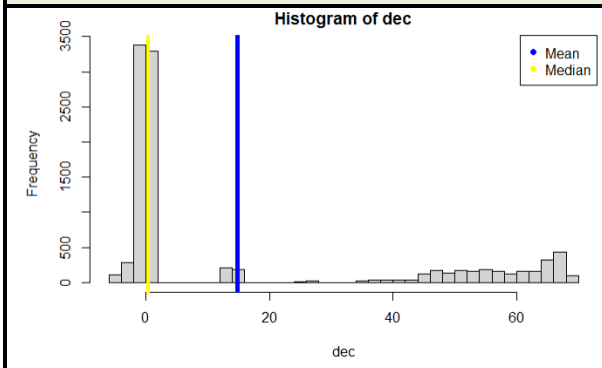
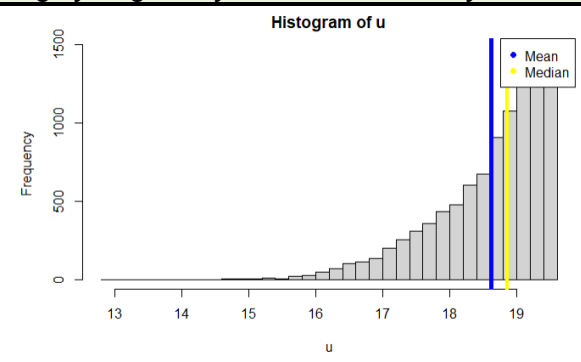
class	number of instances
GALAXY	5027
QSO	850
STAR	4175

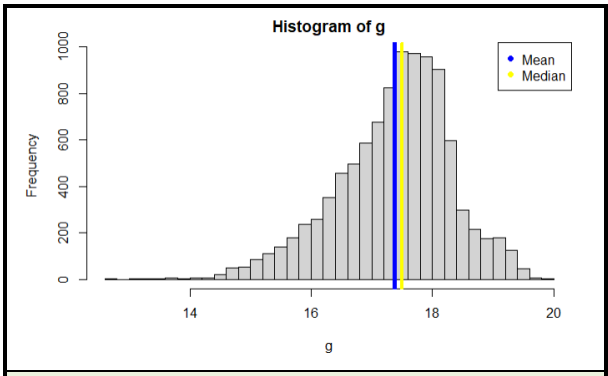
Figure 3. visualisations of class variables



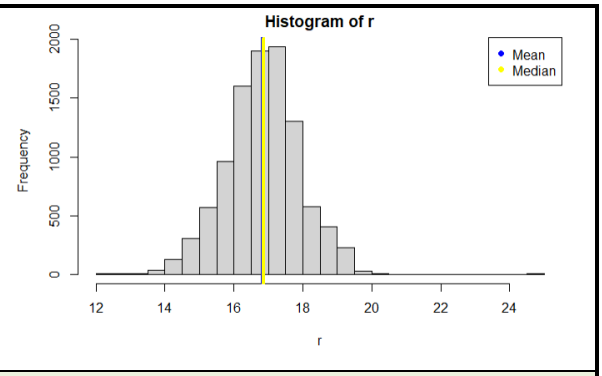
iii. Each histogram has been made by using hist() function. Bins are auto-generated according to each distribution with a maximum number of 30. Abline() was used to put mean and median on each histogram. In addition, skewness and kurtosis were calculated to make it easier to figure out the characteristics of each distribution.

Figure 4. Histograms of each input attribute

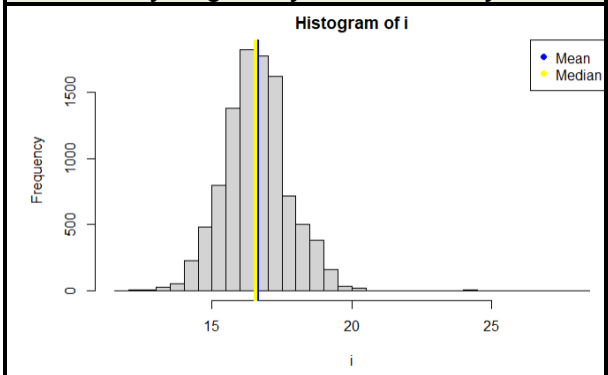
<table>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>objid - skewness: 0, kurtosis: 0
uniform distribution</td><td>dia - skewness: 25.97, kurtosis: 775.61
highly positively skewed</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>rerun - skewness: 0, kurtosis: 0
uniform distribution</td><td>ra - skewness: -1.23, kurtosis: 2.66
highly negatively skewed and heavy tailed</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>dec - skewness: 1.19, kurtosis: -0.40
highly positively skewed and light tailed</td><td>u - skewness: -1.22, kurtosis: 1.43
highly negatively skewed and heavy tailed</td></tr>
</table>



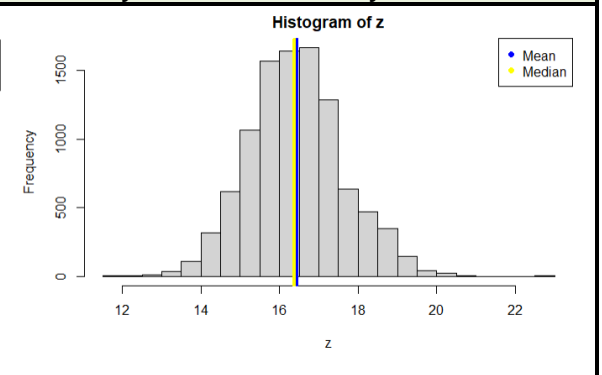
g - skewness: -0.54, kurtosis: 0.44
moderately negatively skewed,heavy tailed



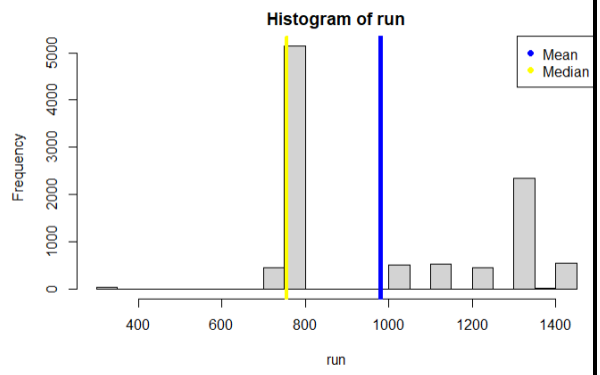
r - skewness: -0.2, kurtosis: 0.75
almost symmetric and heavy tailed



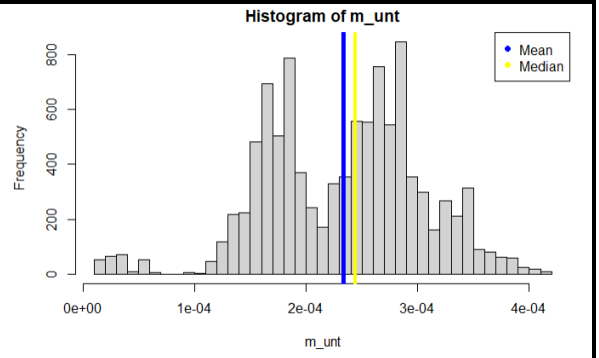
i - skewness: 0.29, kurtosis: 1.83
almost symmetric and heavy tailed



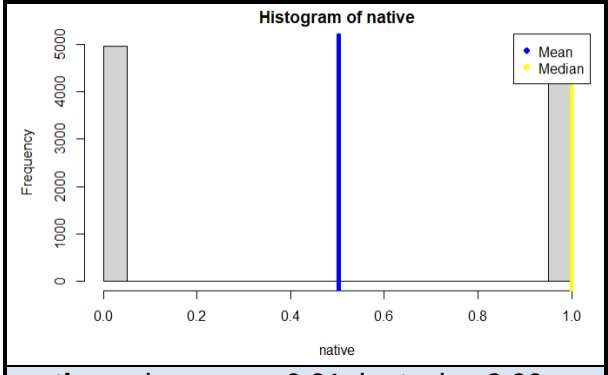
z - skewness: 0.21, kurtosis: 0.37
almost symmetric and heavy tailed



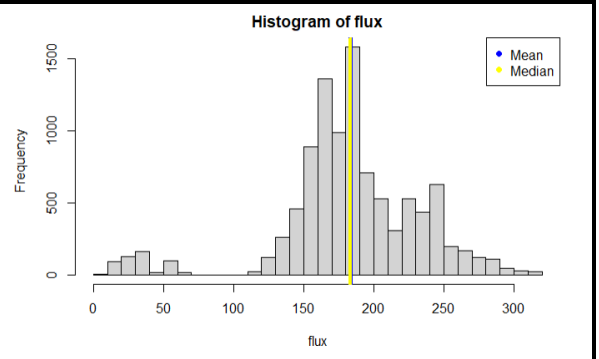
run - skewness: 0.41, kurtosis: -1.56
almost symmetric and light tailed



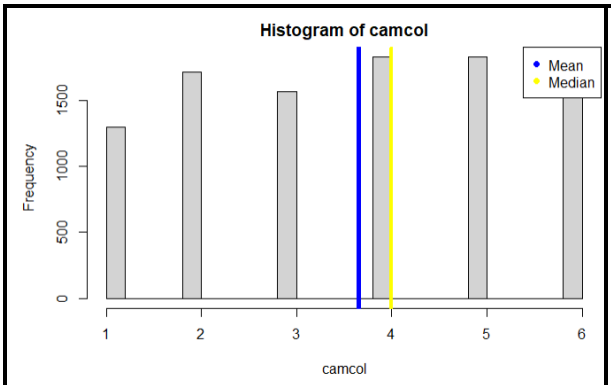
m_unt - skewness: -0.31, kurtosis: 0.13
almost symmetric and heavy tailed



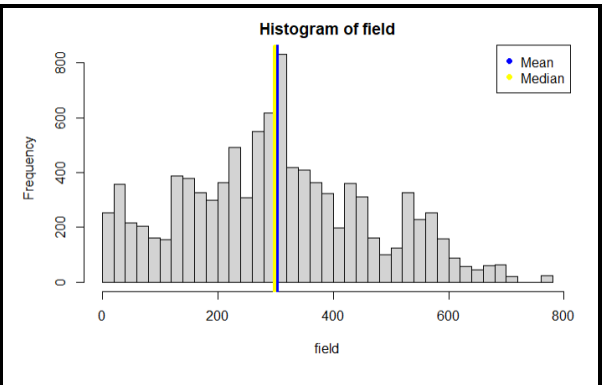
native - skewness: -0.01, kurtosis: -2.00
almost symmetric and light tailed



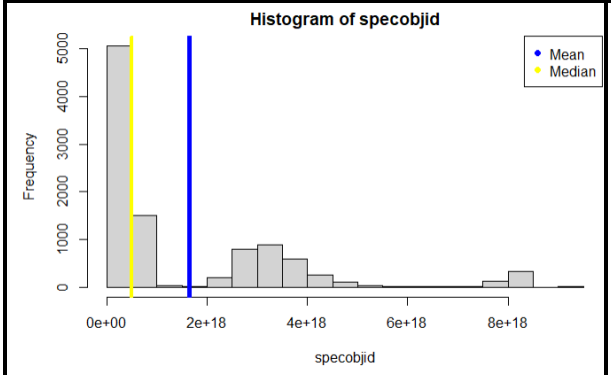
flux - skewness: -0.85, kurtosis: 2.30
moderately negatively skewed, heavy tailed



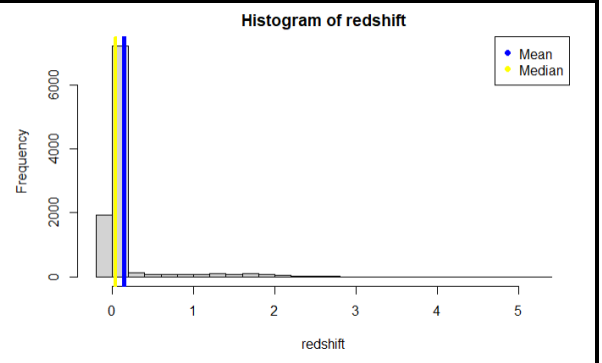
camcol - skewness: -0.10, kurtosis: -1.22
almost symmetric and light tailed



field - skewness: 0.25, kurtosis: -0.48
almost symmetric and light tailed



specobjid - skewness: 1.79, kurtosis: 2.96
highly positvely skewed and heavy tailed



redshift - skewness: 4.26, kurtosis: 20.54
highly positively skewed and heavy tailed

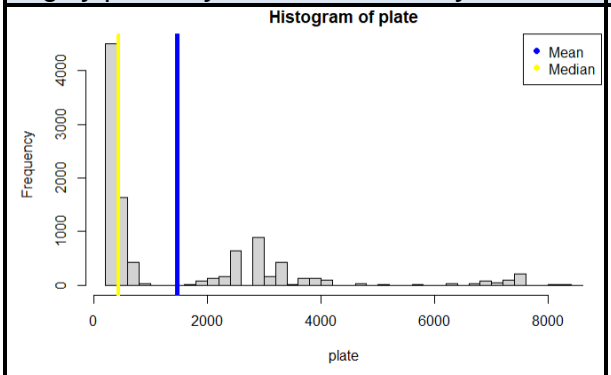
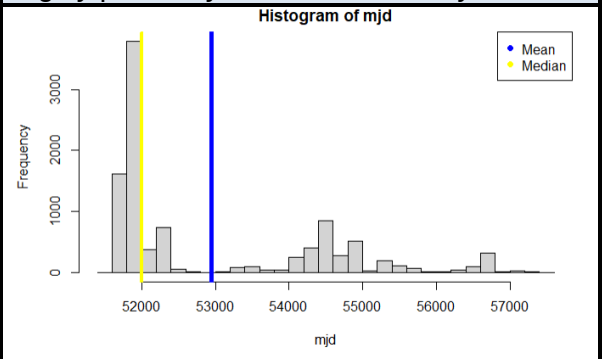
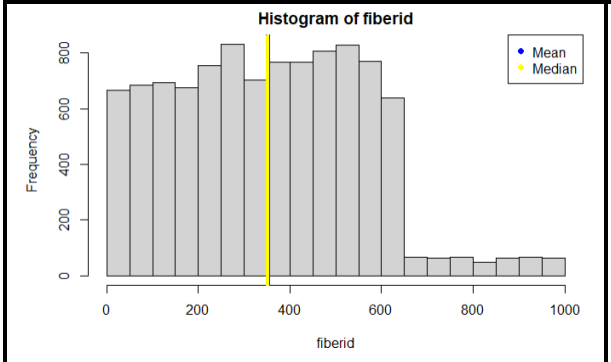


plate - skewness: 1.79, kurtosis: 2.96
highly positively skewed and heavy tailed



mjd - skewness: 1.04, kurtosis: -0.23
highly positively skewed and heavy tailed



fiberid - skewness: 0.31, kurtosis: -0.31
almost symmetric and light tailed

2. Exploring the relationships between the attributes, and between the class and attributes

i., ii.

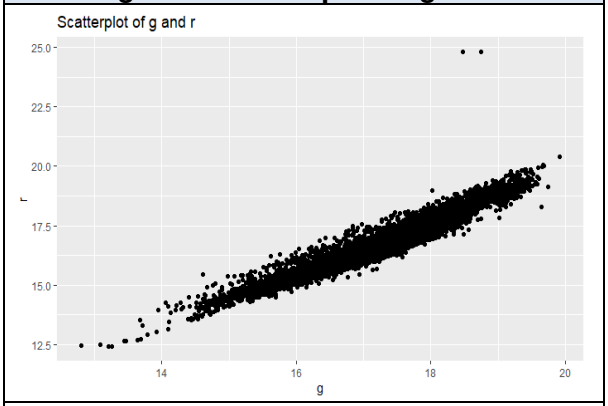
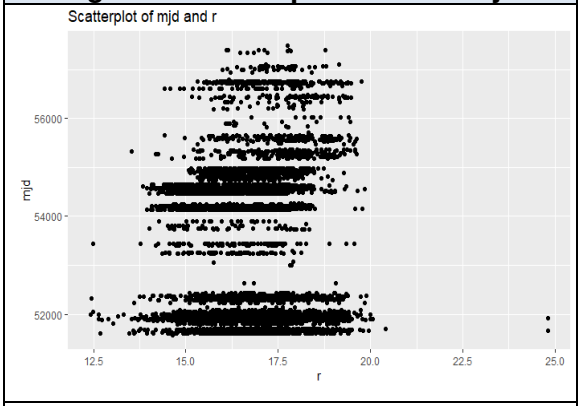
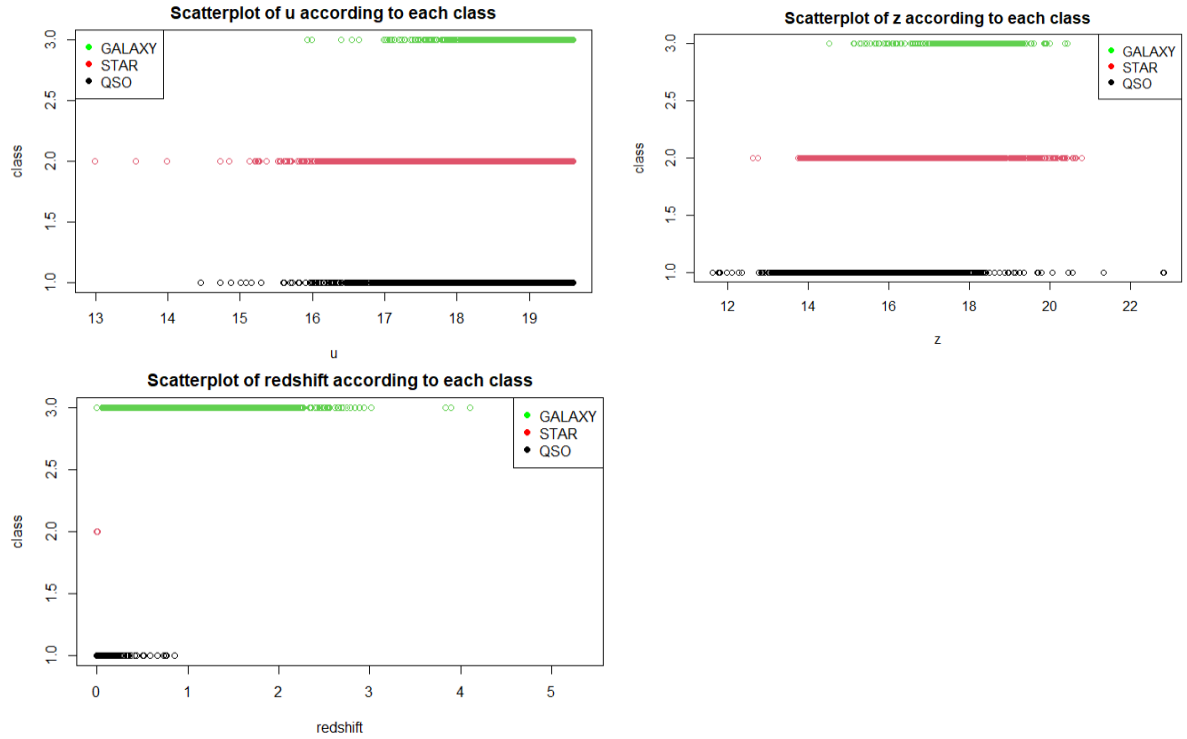
Figure 5. Scatterplot of g and r

correlation between r and g = 0.96
A very strong positive correlation between g and r. Therefore, there is a strong tendency that if g increases, then r also increases.

Figure 6. Scatterplot of r and mjd

correlation between r and mjd = -0.0092
Almost no correlation between r and mjd. Therefore, each value varies regardless of changes of the other.

iii.

Figure 7. Scatterplots between class and u, z, and redshift

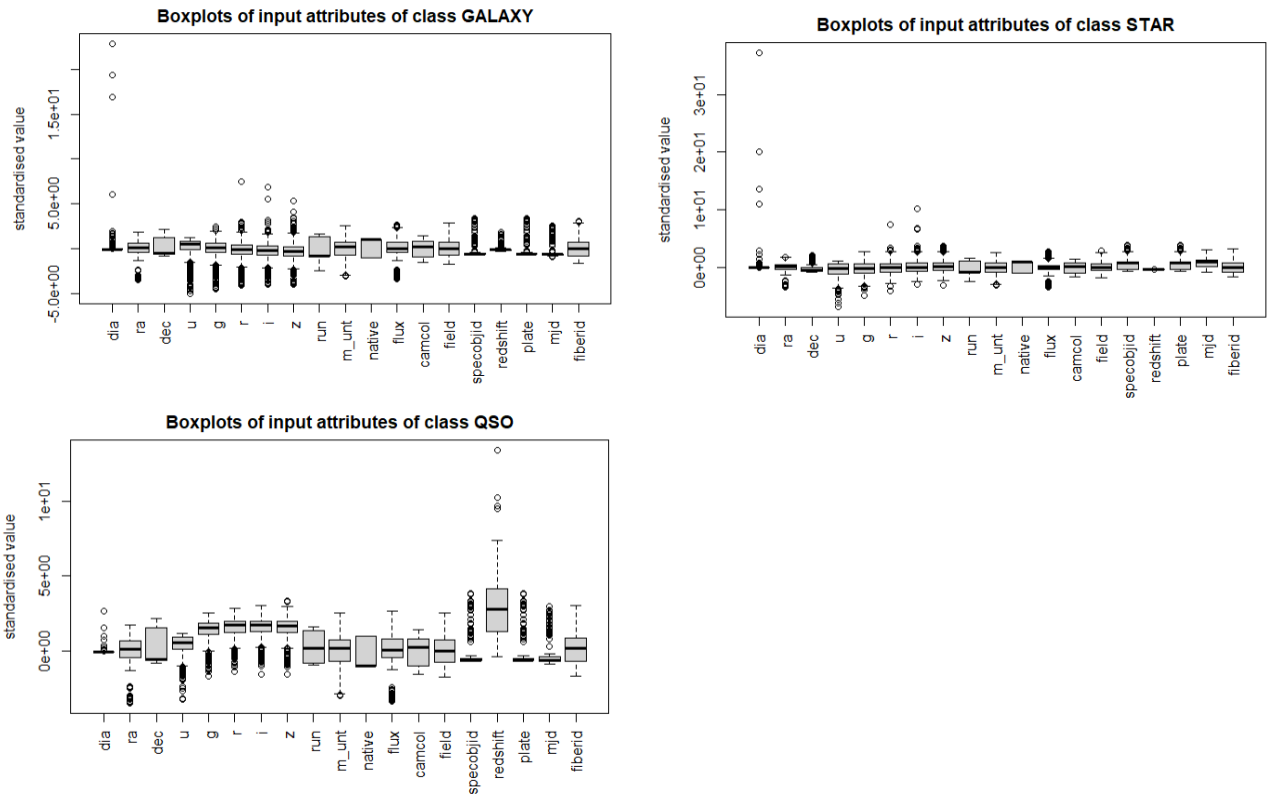


These results mean that there is a possibility that we can figure out the class of certain

instance by using the range of redshift. Unlike u and z, there seems to be a significant difference between redshift of each class in terms of the distribution range.

iv.

Figure 8. Boxplots of input attributes according to each class



3. General Conclusions

As shown in the statistics above, objid, rerun is constant across all of the instances so they do not seem to be significant information. Meanwhile, It may be possible to choose one between r and g and omit the other because there is a very strong positive correlation between them.

On the scatterplots in 2.iii., it seems that redshift attribute may be significant information to figure out the class.

4. Dealing with missing values in R

1) Replacement with 0

- Definition: replacing all missing values with 0.

- Effects on data: For input attributes with mean higher than 0, this replacement method decreases each mean of the attributes and increases standard deviation. For dia, more than 60% of them are missing, so this method decreases its standard deviation.

Figure 9. Effects of replacement with 0 method on data

run			
Min	308	Max	1412
1st quartile	752	Mean	981
Median	756	SD	273
3rd quartile	1331	NA's	50

→

run'			
Min	0	Max	1412
1st quartile	752	Mean	976
Median	756	SD	281
3rd quartile	1331	NA's	0

dia			
Min	28	Max	848172
1st quartile	229	Mean	2000
Median	349	SD	22678
3rd quartile	693	NA's	6646

→

dia'			
Min	0	Max	848172
1st quartile	0	Mean	678
Median	0	SD	11386
3rd quartile	234	NA's	0

2) Mean replacement

- Definition: replacing all missing values with mean value of the input attribute according to each class.

- Effects on data: except objid and rerun, this replacement method decreases standard deviation of each attribute.

Figure 10. Effects of replacement with mean method on data

u			
Min	12.99	Max	19.60
1st quartile	18.18	Mean	18.62
Median	18.85	SD	0.829
3rd quartile	19.26	NA's	59

→

u'			
Min	12.99	Max	19.60
1st quartile	18.18	Mean	18.62
Median	18.85	SD	0.827
3rd quartile	19.26	NA's	0

3) Median replacement

- Definition: replace all missing values with median value of the input attribute according to each class.

- Effects on data: except objid and rerun, this replacement method decreases standard deviation of each attribute. Also, it moves the distribution to the left if median is lower than mean. Otherwise, it moves the distribution to the right.

Figure 11. Effects of replacement with median method on data

dec			
Min	-5.38	Max	68.54
1st quartile	-0.54	Mean	14.83
Median	0.40	SD	25
3rd quartile	35.57	NA's	49

→

dec'			
Min	-5.38	Max	68.54
1st quartile	-0.53	Mean	14.76
Median	0.41	SD	25
3rd quartile	34.62	NA's	0

4) Comparison

Replacement with 0 may change the distribution of certain instance too much because this method does not take into account the original distribution. Meanwhile, between replacement with mean and replacement with median, it depends on the original distribution. If there are many outliers in the data, median replacement would be better than mean replacement. As shown in 1.iii., there are many outliers in some instances, so median replacement will be used.

5. Attribute transformation

1) Mean centering

- Definition: Subtracting mean values from each instance of an attribute to make the distribution center new mean 0.

- Effects on data: It changes the means of all attributes to 0, and the distribution remains the same around the new mean.

Figure 12. Effects of mean centering on "replacement with 0" data

run'			
Min	0	Max	1412
1st quartile	752	Mean	976
Median	756	SD	281
3rd quartile	1331	NA's	0

→

run''			
Min	-976	Max	436
1st quartile	-224	Mean	0
Median	-220	SD	281
3rd quartile	355	NA's	0

Figure 13. Effects of mean centering on "mean replacement" data

run'			
Min	308	Max	1412
1st quartile	752	Mean	981
Median	756	SD	273
3rd quartile	1331	NA's	0

→

run''			
Min	-673	Max	431
1st quartile	-229	Mean	0
Median	-225	SD	273
3rd quartile	350	NA's	0

Figure 14. Effects of mean centering on "median replacement" data

run'			
Min	308	Max	1412
1st quartile	752	Mean	980
Median	756	SD	273
3rd quartile	1331	NA's	0

→

run''			
Min	-672	Max	432
1st quartile	-228	Mean	0
Median	-224	SD	273
3rd quartile	351	NA's	0

2) Normalisation

- Definition: Scaling instances to make their values between 0 and 1.

It can be done by this equation: $x` = (x - x_{min})/(x_{max} - x_{min})$

- Effects on data: As mentioned above, this method puts instance values between 0 and 1. And the standard deviation considerably decreases.

Figure 15. Effects of normalisation on "replacement with 0" data

run'			
Min	0	Max	1412
1st quartile	752	Mean	976
Median	756	SD	281
3rd quartile	1331	NA's	0

→

run''			
Min	0	Max	1
1st quartile	0.53	Mean	0.69
Median	0.54	SD	0.199
3rd quartile	0.94	NA's	0

Figure 16. Effects of normalisation on "mean replacement" data

run'			
Min	308	Max	1412
1st quartile	752	Mean	981
Median	756	SD	273
3rd quartile	1331	NA's	0

→

run''			
Min	0	Max	1
1st quartile	0.4	Mean	0.61
Median	0.41	SD	0.247
3rd quartile	0.93	NA's	0

Figure 17. Effects of normalisation on "median replacement" data

run'			
Min	308	Max	1412
1st quartile	752	Mean	980
Median	756	SD	273
3rd quartile	1331	NA's	0

→

run''			
Min	0	Max	1
1st quartile	0.4	Mean	0.61
Median	0.41	SD	0.247
3rd quartile	0.93	NA's	0

3) Standardisation

- Definition: Scaling data around mean of 0 and standard deviation of 1.

It can be done by this equation: $Z = (x - \mu) / \sigma$

- Effects on data: As mentioned above, this method makes all instances to have a mean of 0 and a standard deviation of 1 regardless of missing value replacement method.

Figure 18. Effects of Standardisation on "replacement with 0" data

run'			
Min	0	Max	1412
1st quartile	752	Mean	976
Median	756	SD	281
3rd quartile	1331	NA's	0

→

run''			
Min	-3.5	Max	1.6
1st quartile	-0.8	Mean	0
Median	-0.8	SD	1
3rd quartile	1.3	NA's	0

Figure 19. Effects of Standardisation on "mean replacement" data

run'			
Min	308	Max	1412
1st quartile	752	Mean	981
Median	756	SD	273
3rd quartile	1331	NA's	0

→

run''			
Min	-2.47	Max	1.58
1st quartile	-0.84	Mean	0
Median	-0.83	SD	1
3rd quartile	1.28	NA's	0

Figure 20. Effects of Standardisation on "median replacement" data

run'			
Min	308	Max	1412
1st quartile	752	Mean	980
Median	756	SD	273
3rd quartile	1331	NA's	0

→

run''			
Min	-2.46	Max	1.58
1st quartile	-0.83	Mean	0
Median	-0.82	SD	1
3rd quartile	1.29	NA's	0

4) Comparison between the three transformation methods

Mean centering changes the mean of each attribute to 0, but the standard deviation remains the same. However, unlike mean centering, normalisation and standardisation change standard deviations also. Between normalisation and standardisation, the difference is that normalisation decreases standard deviations, but standardisation changes standard deviations to 1 regardless of the original values. Standardisation seems to be more appropriate because it makes it possible to compare attributes with the same standard, and it is also suitable for PCA.

6. Attribute / instance selection

i.

First, any attributes with more than 3000 missing values (about 30% of the number of instances) will be deleted. Therefore, dia attribute will be deleted. It is possible to replace the missing values with mean, median or 0, but it will change the original distribution a lot (as shown in 4.1) so it does not seem to have a positive impact on this model.

Secondly, any instance with more than 10 missing values (50% of the number of attributes except dia) will be deleted in the same context. This will result in deleting 50 instances.

Lastly, any duplicated instances will be deleted. After the last two sequences, there are only 2 duplicated instances. Furthermore, this dataset still contains attributes such as objid, camcol, and fiberid, so these duplicated instances seem to be a mistake in data collection. In consequence, 10000 instances will be left in the data.

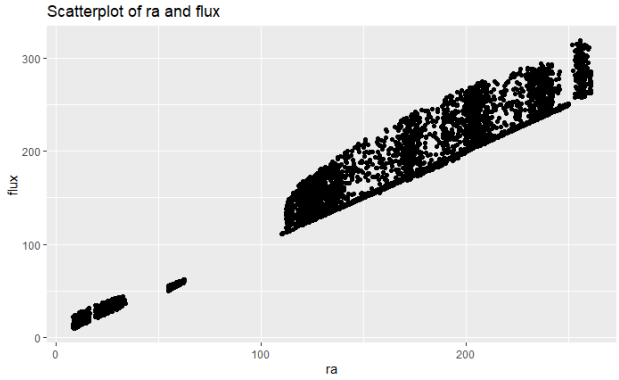
ii.

As shown in 2.i., there is a very strong positive correlation between r and g. Therefore, we can choose one of them and delete the other. In 1.iii., it has been revealed that r has a more desirable distribution than g (symmetric while g is negatively skewed). On the other hand, as can be seen in 1.i., g has fewer missing values (g: 51, r: 53). The difference in the number of missing values is relatively small, so g will be deleted from the data.

Meanwhile, as figured out in Figure 21. below, ra and flux is also highly correlated (very strong positive correlation, the correlation between ra and flux = 0.95). Therefore, one of them can be deleted. In 1.iiil, it seems that flux has more desirable distribution than r in terms of skewness. The number of missing values is 48 for ra, and 50 for flux. Therefore, ra will be deleted from the data with the same logic as above.

Consequently, there will be 18 attributes and 10000 instances in the data. There will be some missing values in some attributes (1 in rerun, 1 in u, 3 in r, 3 in z), so they will be replaced with the median of each instance according to each class.

Figure 21. Scatterplot of ra and flux



7. Attribute transformation / reduction

i. 1) Pre-processing for PCA

As mentioned above, the data with 18 attributes (dia, ra, and g are deleted) and 10000 instances (duplicated instances or any instance which contains more than 10 missing values are deleted) will be considered for use. However, objid and rerun will also be deleted. As mentioned in 3., the values of these attributes are constant throughout all of the instances. Therefore, they can be considered as insignificant attributes. In consequence, there will be 16 attributes and 10000 instances.

Any other missing values will be replaced with median according to each attribute and class. After that, the data will be standardised for PCA transformation.

2) PCA for data transformation and dimensionality reduction

Figure 22. The summary of all PCs

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
SD	1.921	1.778	1.465	1.395	1.1653	0.9996	0.9230	0.7256	0.6710
Proportion of Variance	0.231	0.198	0.134	0.122	0.0849	0.0624	0.0532	0.0329	0.0281
Cumulative Proportion	0.231	0.428	0.562	0.684	0.7687	0.8311	0.8844	0.9173	0.9454

	PC10	PC11	PC12	PC13	PC14	PC15	PC16
SD	0.6536	0.4981	0.35857	0.2041	0.12868	0.10494	0.000528
Proportion of Variance	0.0267	0.0155	0.00804	0.0026	0.00103	0.00069	0.000000
Cumulative Proportion	0.9721	0.9876	0.99567	0.9983	0.99931	1.00000	1.000000

By using PCA as a transformation technique, it becomes possible to arrange attributes in the order of proportion of variance (descending order). And then, by taking a certain number of PCs, we can reduce the dimension of the data. In Figure 22, only 12 of 16 PCs can be selected, but we still can explain about 99.5% of the variation in the data.

ii. To obtain a cumulative variance of at least 90%, 8 PCs should be used.

Chapter 2: CLUSTERING

1. The results of HCA, k-means, and PAM

1) Dataset used for clustering

The data with 16 attributes and 10000 instances mentioned in 7.i.1) in the previous chapter (but not PCA transformed).

2) Result and comparison

Figure 23. Internal Metrics and indexes for each clustering method

	Between	Within	Silhouette	Dunn	Davies-Bouldin (DB)
HCA	3.896003e+18	6.080752e+17	0.7217115	0.0802005	0.3460808
KMS	3.686294e+18	3.026229e+17	0.8939322	0.07859079	0.2647242
PAM	3.684864e+18	3.024502e+17	0.8939825	0.102981	0.26508

Figure 24. The aligned confusion matrices for HCA, k-means (KMS), and PAM on data

(G: GALAXY, S:STAR, Q:QSO)

HCA			
	1	**2**	**3**
G	4873	29	95
S	693	65	92
Q	2035	376	1742

KMS			
	1	**2**	**3**
G	4862	42	93
S	692	73	85
Q	1050	400	2703

PAM			
	1	**2**	**3**
G	4862	42	93
S	692	73	85
Q	1046	400	2707

Figure 25. Per-class and Average Recall, Precision Metrics of each clustering method

Recall				
	G	**S**	**Q**	**Avg**
HCA	0.98	0.08	0.42	0.49
KMS	0.97	0.09	0.65	0.57
PAM	0.97	0.09	0.65	0.57

Precision				
	G	**S**	**Q**	**Avg**
HCA	0.64	0.14	0.90	0.56
KMS	0.74	0.14	0.94	0.61
PAM	0.74	0.14	0.94	0.61

HCA and PAM were done by using Euclidean distance method. Silhouette and Dunn index were also calculated using Euclidean distance, while DB was calculated using Manhattan distance. All of the internal metrics and indexes are averages.

Between each method, the comparison results vary depending on the indicators to compare ("A > B" means A produced a better result than B).

- The distances between clusters: HCA > KMS > PAM // within clusters: PAM > KMS > HCA

- Silhouette: PAM > KMS > HCA // Dunn: PAM > HCA > KMS // DB: KMS > PAM > HCA

- Recall: KMS = PAM > HCA // Precision: KMS = PAM > HCA

In general, KMS and PAM showed better clustering result than HCA. The clusters in HCA were more compact, but the distances between clusters tend to be bigger in PAM and KMS than HCA (more distinctive clusters). Silhouette and DB index shows that PAM and KMS are better in general because these indexes show how similar a sample is to its cluster, and how different it is from other clusters. Meanwhile, Dunn index shows that KMS has the worst case between the three methods.

Between PAM and KMS, PAM is better than KMS in terms of internal metrics, but KMS is better than PAM in terms of external metrics. This means that while PAM is better off creating clusters in itself, KMS results reflect the actual classes better than PAM.

2. The results of clustering with different parameters

1) Chosen parameters

- HCA: distance (Euclidean / Manhattan), agglomeration method (complete / average)

- PAM: distance (Euclidean / Manhattan), standardisation (TRUE / FALSE)

- k-means : number of iterations (100 $\rightarrow$ 10000), nstart(n) (1 / 20)

2) Reasoning behind the choice of parameters

First, changing the distance method from Euclidean to Manhattan may change the result of clustering. Secondly, the agglomeration method for HCA was changed to see if it can make any improvements. Thirdly, as the target dataset is not standardised, so standardisation before clustering also may change the result of PAM. Lastly, increasing number of iterations for k-means may improve the result, and there is a probability that the result varies according to the initial number of random sets.

3) Results and Comparison

a) HCA (a-1: Manhattan, complete / a-2: Euclidean, average / a-3: Manhattan, average)

Figure 26. Internal Metrics and indexes for each chosen parameters of HCA

	Between	Within	Silhouette	Dunn	DB
a-1	3.896003e+18	6.080752e+17	0.7217115	0.0802005	0.3460808
a-2	3.685783e+18	3.026435e+17	0.8937609	0.1360202	0.2633123
a-3	3.685783e+18	3.026435e+17	0.8937609	0.1360202	0.2633123

Figure 27. The aligned confusion matrices for HCA

a-1			
	1	**2**	**3**
G	4873	29	95
S	693	65	92
Q	2035	376	1742

a-2			
	1	**2**	**3**
G	4862	39	96
S	692	71	87
Q	1046	397	2710

a-3			
	1	**2**	**3**
G	4862	39	96
S	692	71	87
Q	1046	397	2710

Figure 28. Per-class and Average Recall, Precision Metrics of HCA

Recall				
	G	**S**	**Q**	**Avg**
a-1	0.98	0.08	0.42	0.49
a-2	0.97	0.08	0.65	0.57
a-3	0.97	0.08	0.65	0.57

Precision				
	G	**S**	**Q**	**Avg**
a-1	0.64	0.14	0.90	0.56
a-2	0.74	0.14	0.94	0.60
a-3	0.74	0.14	0.94	0.60

As can be seen above, changing the distance method could not make any difference. However, it was possible to improve the clustering (except the distance between clusters) by changing the agglomeration method from complete to average. And the improved performance is similar to k-means and PAM.

b) k-means (b-1: iter = 100, n = 20 / b-2: iter = 10000, n = 1 / b-3: iter = 10000, n = 20)

Figure 29. Internal Metrics and indexes for each chosen parameters of k-means

	Between	**Within**	**Silhouette**	**Dunn**	**DB**
b-1	3.686294e+18	3.026229e+17	0.8939322	0.07859079	0.2647242
b-2	3.686294e+18	3.026229e+17	0.8939322	0.07859079	0.2647242
b-3	3.686294e+18	3.026229e+17	0.8939322	0.07859079	0.2647242

Figure 30. The aligned confusion matrices for k-means

b-1			
	1	**2**	**3**
G	4862	42	93
S	692	73	85
Q	1050	400	2703

b-2			
	1	**2**	**3**
G	4862	42	93
S	692	73	85
Q	1050	400	2703

b-3			
	1	**2**	**3**
G	4862	42	93
S	692	73	85
Q	1050	400	2703

Figure 31. Per-class and Average Recall, Precision Metrics of k-means

Recall				
	G	**S**	**Q**	**Avg**
b-1	0.97	0.09	0.65	0.57
b-2	0.97	0.09	0.65	0.57
b-3	0.97	0.09	0.65	0.57

Precision				
	G	**S**	**Q**	**Avg**
b-1	0.74	0.14	0.94	0.61
b-2	0.74	0.14	0.94	0.61
b-3	0.74	0.14	0.94	0.61

As shown above, changing the number of iterations and nstart could not make any difference.

c) PAM

c-1: No standardisation, Manhattan distance

c-2: Standardisation, Euclidean distance

c-3: Standardisation, Manhattan distance

Figure 32. Internal Metrics and indexes for each chosen parameters of PAM

	Between	Within	Silhouette	Dunn	DB
c-1	3.684864e+18	3.024502e+17	0.8939825	0.102981	0.26508
c-2	2.561772e+18	8.038227e+17	0.3753575	2.767906e-19	1.280494
c-3	2.477404e+18	8.799178e+17	0.33776	3.053595e-19	1.402114

Figure 33. The aligned confusion matrices for PAM

c-1			
	1	**2**	**3**
G	4862	42	93
S	692	73	85
Q	1046	400	2707

c-2			
	1	**2**	**3**
G	3390	1497	110
S	460	281	109
Q	773	620	2760

c-3			
	1	**2**	**3**
G	3194	1708	95
S	349	412	89
Q	748	638	2767

Figure 34. Per-class and Average Recall, Precision Metrics of PAM

Recall				
	G	**S**	**Q**	**Avg**
c-1	0.97	0.86	0.65	0.83
c-2	0.68	0.33	0.66	0.56
c-3	0.64	0.48	0.67	0.60

Precision				
	G	**S**	**Q**	**Avg**
c-1	0.74	0.14	0.94	0.61
c-2	0.73	0.12	0.93	0.59
c-3	0.74	0.15	0.94	0.61

As shown above, changing parameters of PAM resulted in worse clustering performance.

3. Comparing the clustering results from different datasets

k-means was chosen for the clustering method because it showed better clustering performance than the others in terms of external metrics.

i.

Figure 35. Clustering results of the transformed dataset featuring all PCs

Internal Metrics and indexes				
Between	**Within**	**Silhouette**	**Dunn**	**DB**
2.623297e+18	9.371842e+17	0.348798	3.431114e-19	1.243044

Aligned Confusion Matrix			
	1	**2**	**3**
G	3565	1325	107
S	679	13	158
Q	820	1485	1848

Recall			
G	**S**	**Q**	**Avg**
0.71	0.02	0.44	0.39
Precision			
0.70	0.005	0.87	0.53

ii.

Figure 36. Clustering results of the reduced dataset featuring 12 PCs

Internal Metrics and indexes				
Between	**Within**	**Silhouette**	**Dunn**	**DB**
2.617346e +18	9.393101e +18	0.3448522	3.431114e-19	1.244904

Aligned Confusion Matrix			
	1	2	3
G	3565	1325	107
S	679	13	158
Q	820	1485	1848

Recall			
G	S	Q	Avg
0.71	0.02	0.44	0.39
Precision			
0.70	0.005	0.87	0.53

iii.

Figure 37. Clustering results of the dataset after deletion

Internal Metrics and indexes				
Between	Within	Silhouette	Dunn	DB
3.686294e+18	3.026229e+17	0.8939322	0.07859079	0.2647242

Aligned Confusion Matrix			
	1	2	3
G	4862	42	93
S	692	73	85
Q	1050	400	2703

Recall			
G	S	Q	Avg
0.97	0.09	0.65	0.57
Precision			
0.74	0.14	0.94	0.61

iv.

Figure 38. Clustering results of the mean-centered, transformed data featuring all PCs

Internal Metrics and indexes				
Between	Within	Silhouette	Dunn	DB
2.623297e+18	9.371842e+17	0.348798	3.431114e-19	1.243044

Aligned Confusion Matrix			
	1	2	3
G	3565	1325	107
S	679	13	158
Q	820	1485	1848

Recall			
G	S	Q	Avg
0.71	0.02	0.44	0.39
Precision			
0.70	0.005	0.87	0.53

Figure 39. Clustering results of the mean-centered, reduced data featuring 12 PCs

Internal Metrics and indexes				
Between	Within	Silhouette	Dunn	DB
2.623297e+18	9.371842e+17	0.348798	3.431114e-19	1.243044

Aligned Confusion Matrix			
	1	2	3
G	3565	1325	107
S	679	13	158
Q	820	1485	1848

Recall			
G	S	Q	Avg
0.71	0.02	0.44	0.39
Precision			
0.70	0.005	0.87	0.53

Figure 40. Clustering results of the mean-centered data after deletion

Internal Metrics and indexes				
Between	Within	Silhouette	Dunn	DB
3.686294e+18	3.026229e+17	0.8939322	0.07859079	0.2647242

Aligned Confusion Matrix			
	1	**2**	**3**
G	4862	42	93
S	692	73	85
Q	1050	400	2703

Recall			
G	**S**	**Q**	**Avg**
0.97	0.09	0.65	0.57
Precision			
0.74	0.14	0.94	0.61

v. As shown above, dataset in iii. made better clustering result than the other cases in terms of all of the internal and external metrics. And changing centering method from median to mean did not make any significant difference. Between i, ii, and iii, it seems that PCA transformation or dimension reduction does not seem to be the best choice for this clustering. This may be because there is a certain attribute that is more important for clustering while showing relatively small variance than other attributes.

Chapter 3: CLASSIFICATION

1. The results of each classification algorithm

As mentioned in 3.v. of the previous section, the data with 16 attributes and 10000 instances after deletion brought better results. Therefore, that version of dataset was used again. To avoid overfitting, classification was done by using 10-fold cross-validation. The results will be compared by using recall and precision. For 5-NN, the Euclidean distance method was used.

Figure 41. The confusion matrices for each case

ZeroR			
	1	**2**	**3**
G	4997	0	0
S	850	0	0
Q	4153	0	0

OneR			
	1	**2**	**3**
G	4965	28	4
S	61	788	1
Q	8	0	4145

NaiveBayes			
	1	**2**	**3**
G	4849	127	21
S	56	793	1
Q	276	21	3856

5-NN			
	1	**2**	**3**
G	4755	11	231
S	187	577	86
Q	806	40	3307

J48			
	1	**2**	**3**
G	4937	35	25
S	66	783	1
Q	5	0	4148

Figure 42. The result of each classifier

Recall				
	G	**S**	**Q**	**Avg**
ZeroR	1	0	0	0.333
OneR	0.994	0.927	0.998	**0.973**
NaiveBayes	0.970	0.933	0.928	0.944
5-NN	0.952	0.679	0.796	0.809
J48	0.988	0.921	0.999	0.970

Precision				
	G	**S**	**Q**	**Avg**
ZeroR	0.5	0	0	0.167
OneR	0.986	0.966	0.999	**0.983**
NaiveBayes	0.936	0.843	0.994	0.924
5-NN	0.827	0.919	0.913	0.886
J48	0.986	0.957	0.994	0.979

Comparison: OneR > J48 > NaiveBeyes > 5-NN > ZeroR

The result of ZeroR was the worst. The fact that the result of OneR was better than the

others implies that there might be an attribute very closely related to the classes.

2. The classification results with different parameters

Figure 43. Combinations of parameters

Case 1	Case 2	Case 3
50:50, 3-NN, Euclidean	50:50, 3-NN, Manhattan	50:50, 5-NN, Euclidean
Case 4	**Case 5**	**Case 6**
50:50, 5-NN, Manhattan	80:20, 3-NN, Euclidean	80:20, 3-NN, Manhattan
Case 7	**Case 8**	
80:20, 5-NN, Euclidean	80:20, 5-NN, Manhattan	

Figure 44. The confusion matrices for each case

Case 1			
	1	**2**	**3**
G	2340	10	173
S	94	249	50
Q	412	22	1650

Case 2			
	1	**2**	**3**
G	2363	15	145
S	101	237	55
Q	402	33	1659

Case 3			
	1	**2**	**3**
G	2406	10	107
S	107	235	51
Q	443	24	1617

Case 4			
	1	**2**	**3**
G	2431	12	80
S	104	236	53
Q	441	20	1623

Case 5			
	1	**2**	**3**
G	938	3	57
S	40	108	18
Q	163	7	666

Case 6			
	1	**2**	**3**
G	935	6	57
S	37	111	18
Q	145	9	682

Case 7			
	1	**2**	**3**
G	948	4	46
S	44	105	17
Q	172	11	653

Case 8			
	1	**2**	**3**
G	951	4	43
S	41	106	19
Q	160	11	665

Figure 45. Accuracy, Precision, and recall of each case

Recall				
	G	**S**	**Q**	**Avg**
Case 1	0.927	0.634	0.792	0.784
Case 2	0.937	0.603	0.796	0.779
Case 3	0.954	0.598	0.776	0.776
Case 4	0.964	0.601	0.779	0.781
Case 5	0.940	0.651	0.797	0.796
Case 6	**0.937**	**0.669**	**0.816**	**0.807**
Case 7	0.950	0.633	0.781	0.788
Case 8	0.953	0.639	0.795	0.796

Precision				
	G	**S**	**Q**	**Avg**
Case 1	0.822	0.886	0.881	0.863
Case 2	0.824	0.862	0.892	0.859
Case 3	0.814	0.874	0.911	0.866
Case 4	0.817	0.881	0.924	0.874
Case 5	**0.822**	**0.915**	**0.899**	**0.879**
Case 6	0.837	0.881	0.901	0.873
Case 7	0.814	0.875	0.912	0.867
Case 8	0.826	0.876	0.915	0.872

Accuracy (%)			
Case 1	84.78	**Case 5**	85.6
Case 2	85.18	**Case 6**	**86.4**
Case 3	85.16	**Case 7**	85.3
Case 4	85.8	**Case 8**	86.1

As shown above, Case 6 (80:20, 3-NN, Manhattan) showed the best result in terms of recall and accuracy. Meanwhile, Case 5 (80:20, 3-NN, Euclidean) was better than the other cases in precision. This means that Case 6 resulted in relatively more false positives than Case 5.

3. The classification results of J48 from different datasets

(i. APCs **/ ii.** 12PCs **/ iii.** Del **/ iv.** N.APCs, N.12PCs, N.Del**)**

Figure 46. The confusion matrices for each dataset

APCs			
	1	**2**	**3**
G	4764	30	203
S	55	776	19
Q	137	13	4003

12PCs			
	1	**2**	**3**
G	4587	35	375
S	51	766	33
Q	451	17	3685

Del			
	1	**2**	**3**
G	4937	35	25
S	66	783	1
Q	5	0	4148

N.APCs			
	1	**2**	**3**
G	4764	29	204
S	56	776	18
Q	148	10	3995

N.12PCs			
	1	**2**	**3**
G	4566	34	397
S	51	767	32
Q	426	17	3710

N.Del			
	1	**2**	**3**
G	4931	36	30
S	66	783	1
Q	4	0	4149

Figure 47. Accuracy, Precision, and recall of each dataset

Recall				
	G	**S**	**Q**	**Avg**
APCs	0.953	0.913	0.964	0.943
12PCs	0.918	0.901	0.887	0.902
Del	**0.988**	**0.921**	**0.999**	**0.969**
N.APCs	0.953	0.913	0.962	0.943
N.12PCs	0.914	0.902	0.893	0.903
N.Del	0.987	0.921	0.999	0.969

Precision				
	G	**S**	**Q**	**Avg**
APCS	0.961	0.947	0.947	0.952
12PCS	0.901	0.936	0.900	0.912
Del	**0.986**	**0.957**	**0.994**	**0.979**
N.APCs	0.959	0.952	0.947	0.953
N.12PCs	0.905	0.938	0.896	0.913
N.Del	0.986	0.956	0.993	0.978

Accuracy (%)			
APCS	95.43	**N.APCs**	95.35
12PCS	90.38	**N.12PCs**	90.43
Del	**98.68**	**N.Del**	98.63

v. As shown above, J48 with 10-fold cross-validation on the dataset after deletion of instances and attributes produced the best result in terms of recall, precision, and accuracy. The second-best result was made by using the normalised dataset after deletion. There is very little difference between these two datasets, but it shows that normalisation makes classification results worse anyway.

Like 3.v. in the previous chapter, PCA transformation or data reduction does not seem to be the best choice for this classification. It implies that there might be a certain attribute that is more important for classification while showing relatively small variance than other variables.

Appendix: R code for analysis

cw2

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.4
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.0.5
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.0.5
```

```
library(cluster.datasets)
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 4.0.5
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.5
```

```
library(fpc)
```

```
## Warning: package 'fpc' was built under R version 4.0.5
```

```
library(clv)
```

```
## Warning: package 'clv' was built under R version 4.0.5
```

```
## Loading required package: class
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.0.5
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
Sys.setenv(LANGUAGE="en")
```

1 ANALYSIS AND PRE-PROCESSING [R ONLY, 40 MARKS]

1. Explore the data [6]

i. Provide a table for all the input features of the dataset including measures of centrality, dispersion, and how many missing values each attribute has.

```
cw2 = read.csv("cw_data.csv")
k = ncol(cw2) - 1
summary(cw2)
```

```
##      objid                dia                rerun           ra
##  Min.   :1.24e+18   Min.   :    27.5   Min.   :301   Min.   :  8.235
##  1st Qu.:1.24e+18   1st Qu.:   229.3   1st Qu.:301   1st Qu.:157.381
##  Median :1.24e+18   Median :   349.1   Median :301   Median :180.442
##  Mean   :1.24e+18   Mean   :  2000.0   Mean   :301   Mean   :175.545
##  3rd Qu.:1.24e+18   3rd Qu.:   693.4   3rd Qu.:301   3rd Qu.:201.544
##  Max.   :1.24e+18   Max.   :848171.8   Max.   :301   Max.   :260.884
##                     NA's   :6646       NA's   :30    NA's   :48
##       dec                u                g                r
##  Min.   :-5.3826   Min.   :12.99   Min.   :12.80   Min.   :12.43
```

```
##  1st Qu.:-0.5384   1st Qu.:18.18   1st Qu.:16.81   1st Qu.:16.17
##  Median : 0.4029   Median :18.85   Median :17.50   Median :16.86
##  Mean   :14.8268   Mean   :18.62   Mean   :17.37   Mean   :16.84
##  3rd Qu.:35.5656   3rd Qu.:19.26   3rd Qu.:18.01   3rd Qu.:17.51
##  Max.   :68.5423   Max.   :19.60   Max.   :19.92   Max.   :24.80
##  NA's   :49        NA's   :50      NA's   :51      NA's   :53
##        i               z              run            m_unt
##  Min.   :11.95   Min.   :11.61   Min.   : 308.0   Min.   :0.00001
##  1st Qu.:15.85   1st Qu.:15.62   1st Qu.: 752.0   1st Qu.:0.00018
##  Median :16.56   Median :16.39   Median : 756.0   Median :0.00024
##  Mean   :16.58   Mean   :16.42   Mean   : 980.9   Mean   :0.00023
##  3rd Qu.:17.26   3rd Qu.:17.14   3rd Qu.:1331.0   3rd Qu.:0.00028
##  Max.   :28.18   Max.   :22.83   Max.   :1412.0   Max.   :0.00041
##  NA's   :50      NA's   :53      NA's   :50       NA's   :46
##      native           flux            camcol          field
##  Min.   :0.0000   Min.   :  9.509   Min.   :1.000   Min.   : 11.0
##  1st Qu.:0.0000   1st Qu.:161.666   1st Qu.:2.000   1st Qu.:185.0
##  Median :1.0000   Median :183.325   Median :4.000   Median :299.0
##  Mean   :0.5027   Mean   :183.518   Mean   :3.648   Mean   :302.4
##  3rd Qu.:1.0000   3rd Qu.:212.513   3rd Qu.:5.000   3rd Qu.:414.0
##  Max.   :1.0000   Max.   :319.095   Max.   :6.000   Max.   :768.0
##  NA's   :50       NA's   :50        NA's   :50      NA's   :50
##    specobjid            redshift            plate           mjd
##  Min.   :3.000e+17   Min.   :-0.00414   Min.   : 266   Min.   :51578
##  1st Qu.:3.390e+17   1st Qu.: 0.00008   1st Qu.: 301   1st Qu.:51900
##  Median :4.970e+17   Median : 0.04254   Median : 441   Median :51997
##  Mean   :1.646e+18   Mean   : 0.14368   Mean   :1462   Mean   :52944
##  3rd Qu.:2.880e+18   3rd Qu.: 0.09256   3rd Qu.:2559   3rd Qu.:54468
##  Max.   :9.470e+18   Max.   : 5.35385   Max.   :8410   Max.   :57481
##  NA's   :50          NA's   :50         NA's   :50     NA's   :50
##     fiberid          class
##  Min.   :   1.0   Length:10052
##  1st Qu.: 186.0   Class :character
##  Median : 351.0   Mode  :character
##  Mean   : 352.8
##  3rd Qu.: 510.0
##  Max.   :1000.0
##  NA's   :32
```

```
options("scipen"=0, "digits"=2)
sapply(cw2[,1:21], FUN = sd, na.rm = TRUE)
```

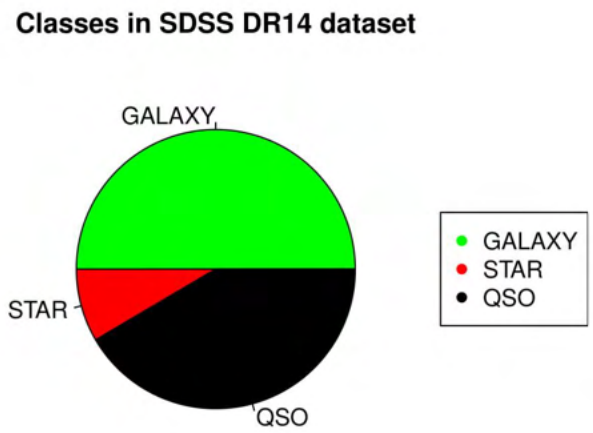
```
##     objid       dia     rerun        ra       dec         u         g         r
##   0.0e+00   2.3e+04   0.0e+00   4.8e+01   2.5e+01   8.3e-01   9.5e-01   1.1e+00
##         i         z       run    m_unt    native      flux    camcol     field
##   1.1e+00   1.2e+00   2.7e+02   7.1e-05   5.0e-01   5.0e+01   1.7e+00   1.6e+02
## specobjid  redshift     plate       mjd   fiberid
##   2.0e+18   3.9e-01   1.8e+03   1.5e+03   2.1e+02
```

ii. Analyse the class variable using appropriate statistics and visualisations

```
cw2 %>% count(class)
```

```
##    class    n
## 1 GALAXY 5027
## 2    QSO  850
## 3   STAR 4175
```

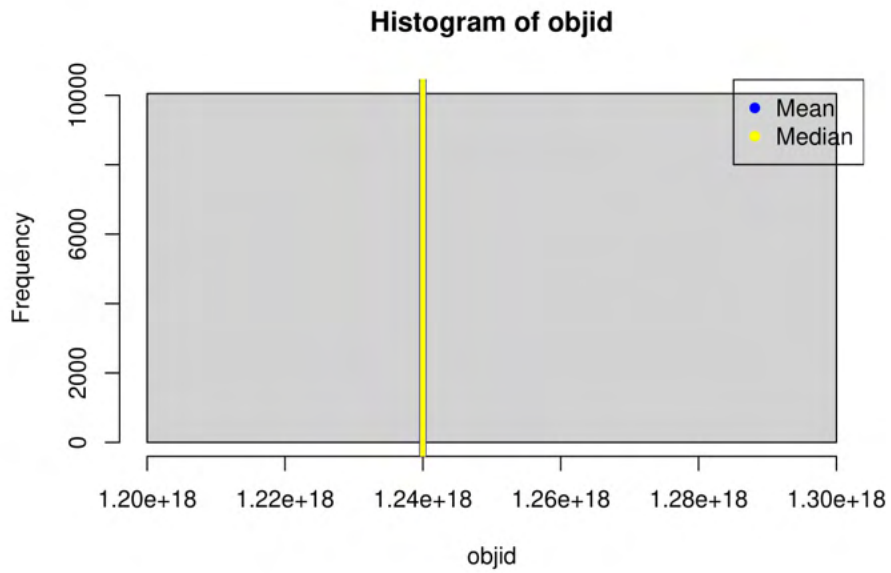
```
lbls = c("GALAXY", "STAR", "QSO")
pie(table(cw2$class), labels = lbls, col=c("green","red","black"), main = "Classes in SDSS DR14 dataset
legend("right", c("GALAXY", "STAR", "QSO"), col=c("green", "red", "black"), pch=16)
```



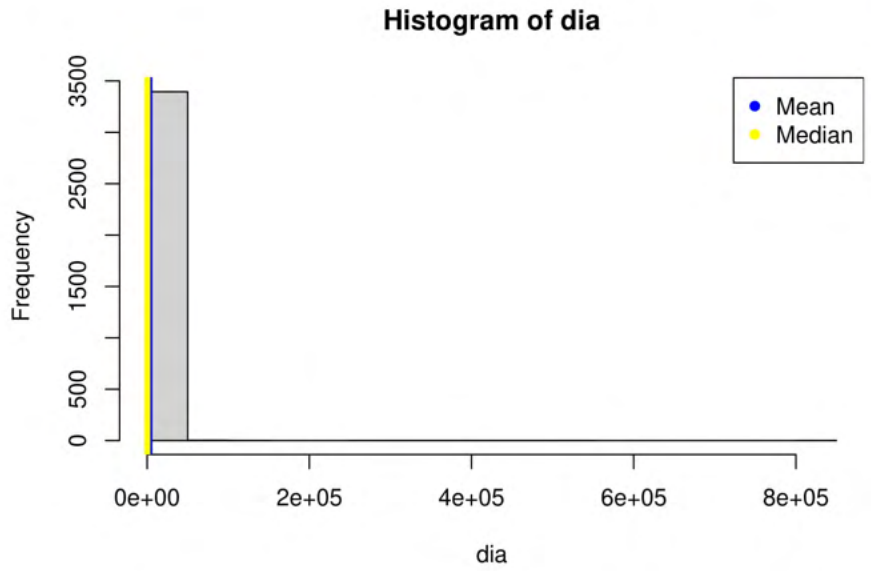
iii. Produce histograms for each input attribute and characterise all the distributions according to shape. Provide details on how you created the histograms. You may also use descriptive statistics to help you characterise the shape of the distribution.

```
plotHistogram = function(df, attb){
hist(df[,attb], breaks = 30,
main=paste("Histogram of", attb), xlab=attb)
abline(v=mean(df[,attb], na.rm = TRUE),col="blue", lwd=5)
abline(v=median(df[,attb], na.rm = TRUE),col="yellow", lwd=4)
legend("topright",c("Mean","Median"),col=c("blue","yellow"), pch=16)
print(attb)
print(paste("skewness = ", skewness(df[,attb], na.rm = TRUE)))
print(paste("kurtosis = ", kurtosis(df[,attb], na.rm = TRUE)))
```

```
print("------")
}

for (i in 1:21) {
  plotHistogram(cw2, colnames(cw2[i]))
}
```

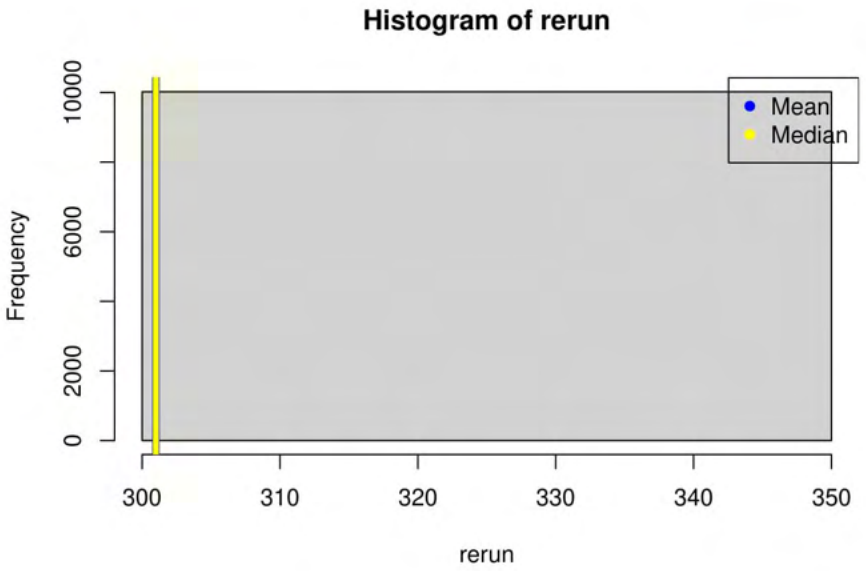



```
## [1] "objid"
## [1] "skewness =  NaN"
## [1] "kurtosis =  NaN"
## [1] "------"
```



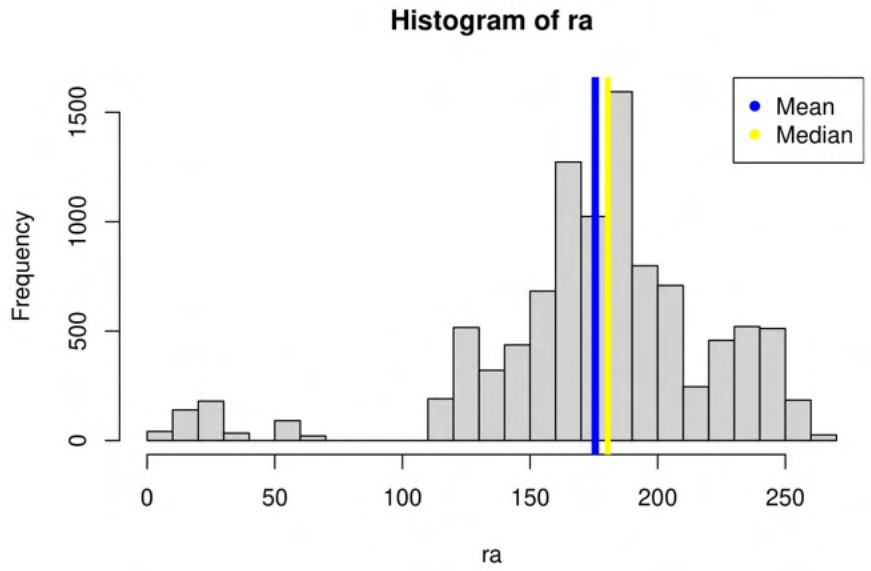

```
## [1] "dia"
## [1] "skewness =  25.9696061271663"
## [1] "kurtosis =  775.608957398551"
## [1] "------"
```

6



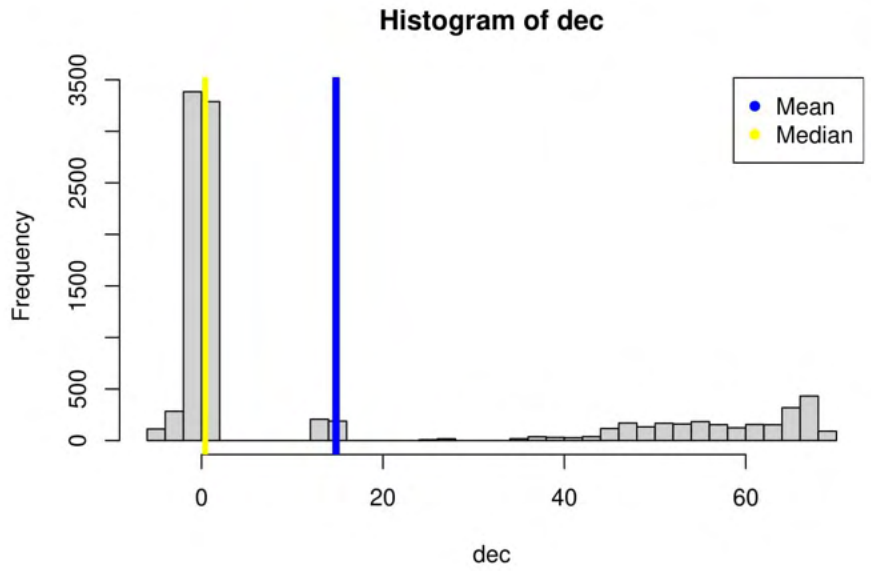

```
## [1] "rerun"
## [1] "skewness =  NaN"
## [1] "kurtosis =  NaN"
## [1] "------"
```

7



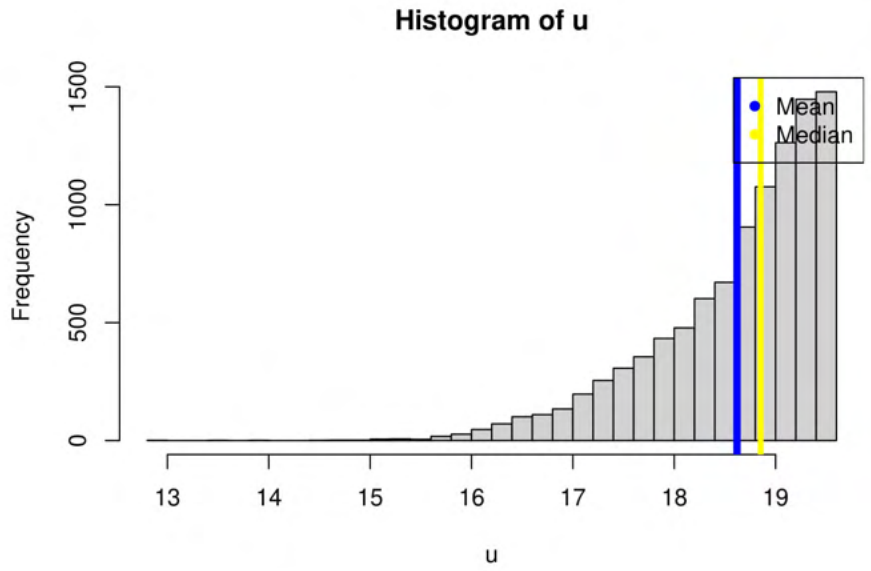

```
## [1] "ra"
## [1] "skewness =  -1.22814052934918"
## [1] "kurtosis =  2.66487275211312"
## [1] "------"
```

8



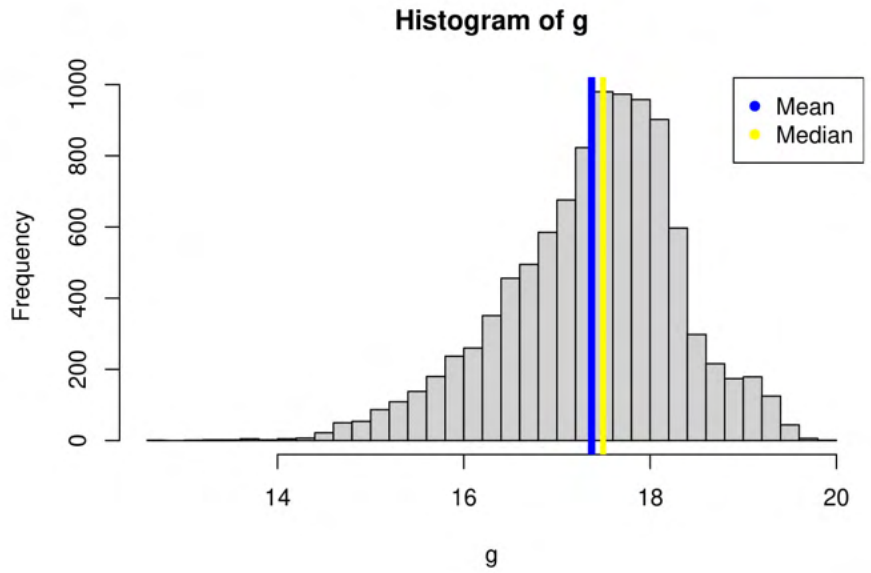

```
## [1] "dec"
## [1] "skewness =  1.19220617927254"
## [1] "kurtosis =  -0.404651241479174"
## [1] "------"
```

9



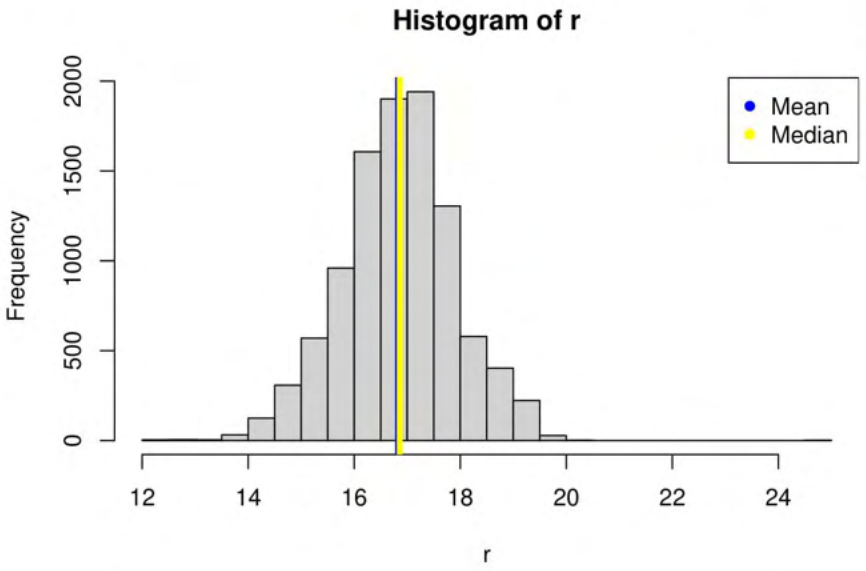

```
## [1] "u"
## [1] "skewness =  -1.219194571805"
## [1] "kurtosis =  1.4291918907626"
## [1] "------"
```

10

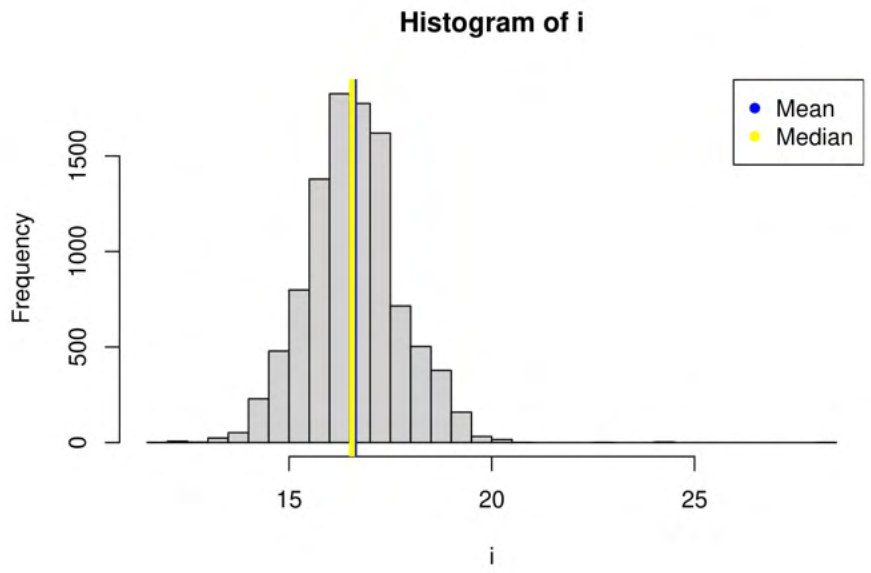



```
## [1] "g"
## [1] "skewness =  -0.536736018074237"
## [1] "kurtosis =  0.444333397540162"
## [1] "------"
```

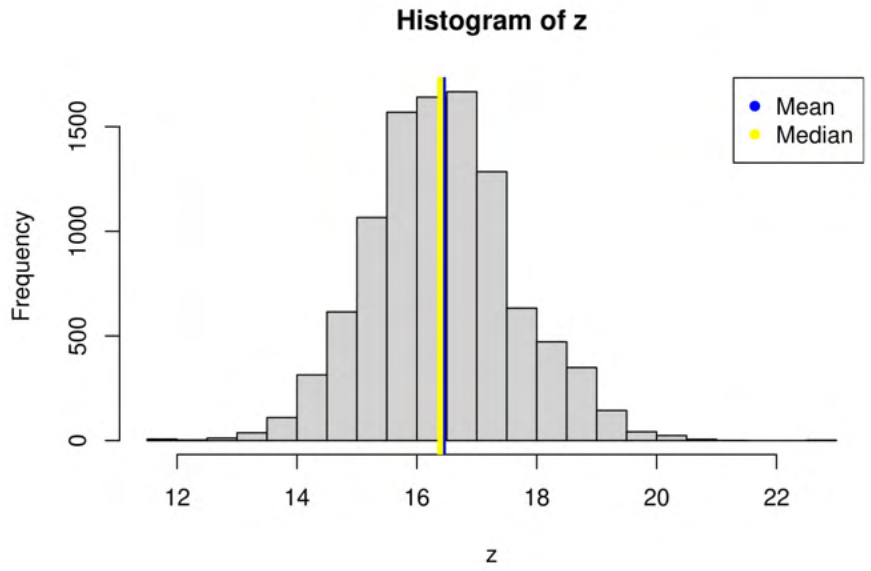
11




```
## [1] "r"
## [1] "skewness =  -0.0221657936000844"
## [1] "kurtosis =  0.752465607404395"
## [1] "------"
```

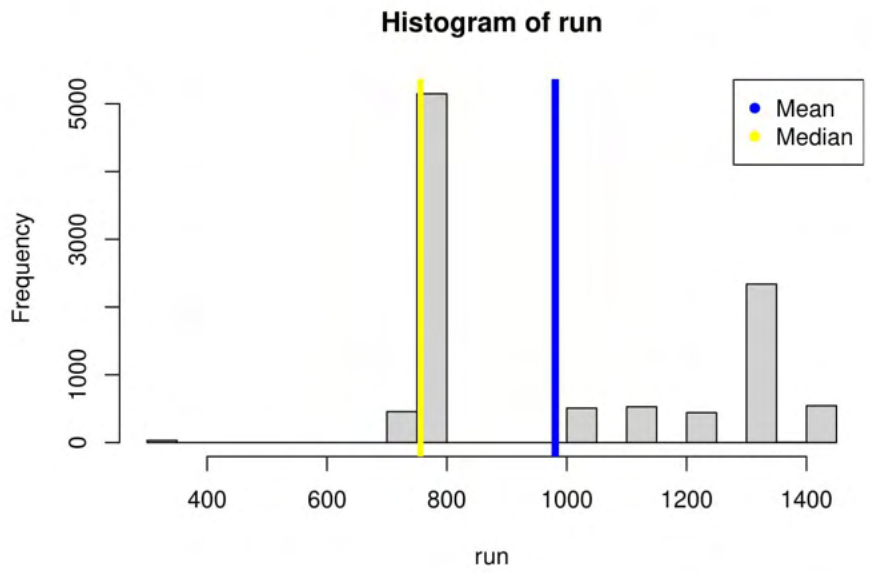



```
## [1] "i"
## [1] "skewness =  0.285511655066811"
## [1] "kurtosis =  1.8280125642078"
## [1] "------"
```

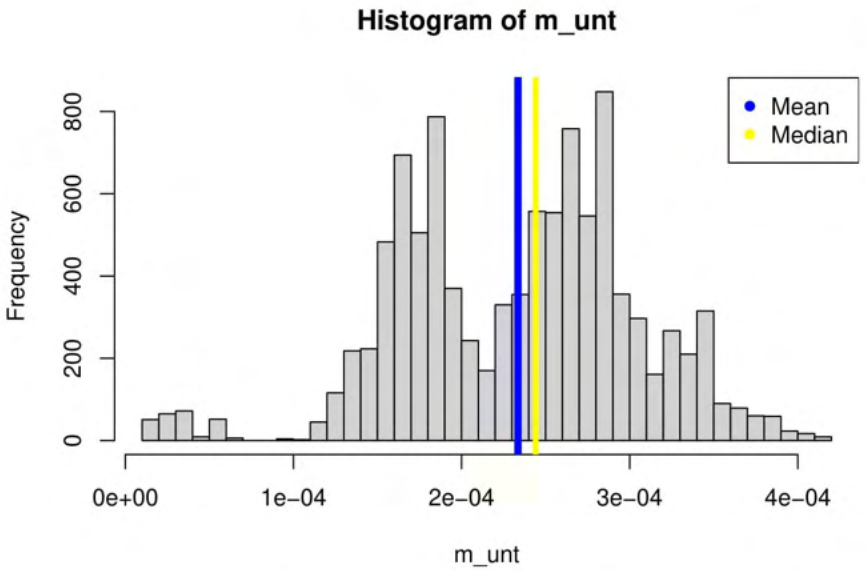



```
## [1] "z"
## [1] "skewness =  0.213670866803996"
## [1] "kurtosis =  0.367506350825824"
## [1] "------"
```

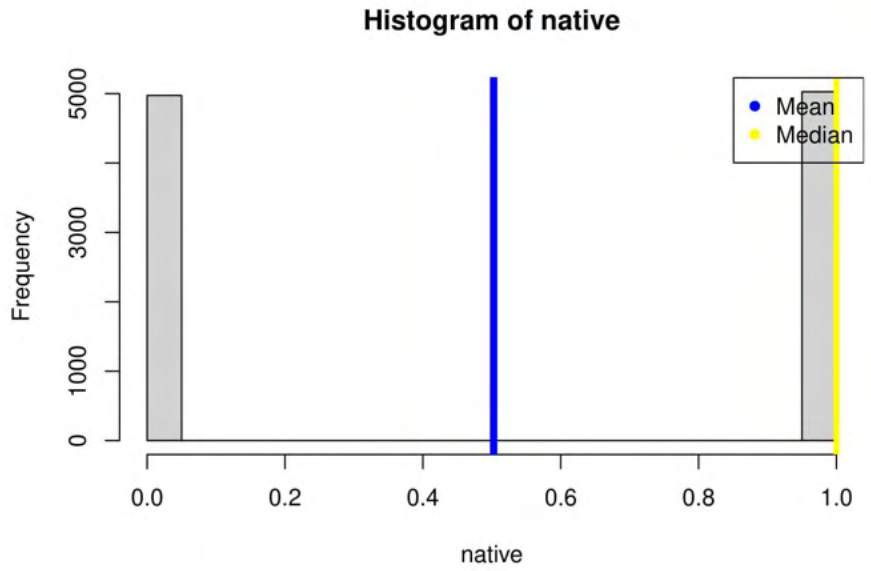
14




```
## [1] "run"
## [1] "skewness =  0.413194690467431"
## [1] "kurtosis =  -1.55830470653861"
## [1] "------"
```

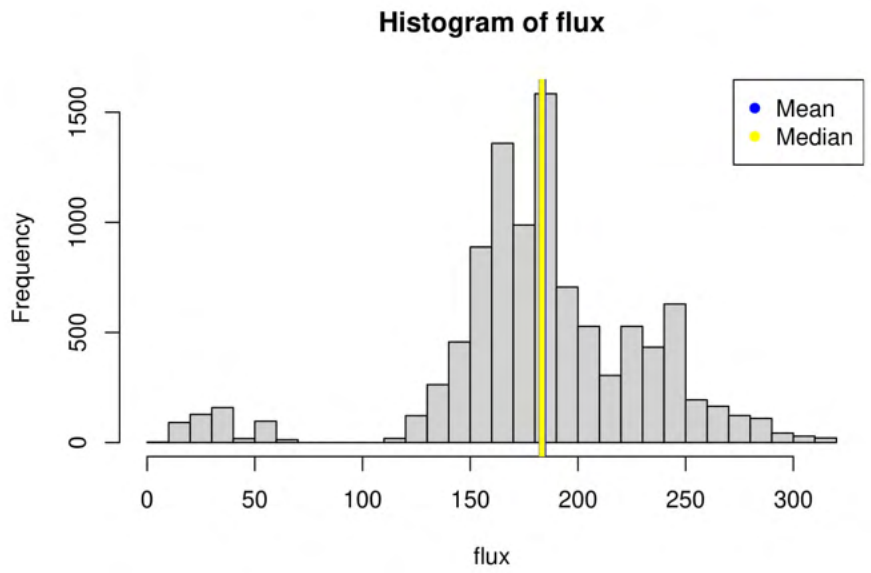



```
## [1] "m_unt"
## [1] "skewness =  -0.307896277587926"
## [1] "kurtosis =  0.127165200419335"
## [1] "------"
```

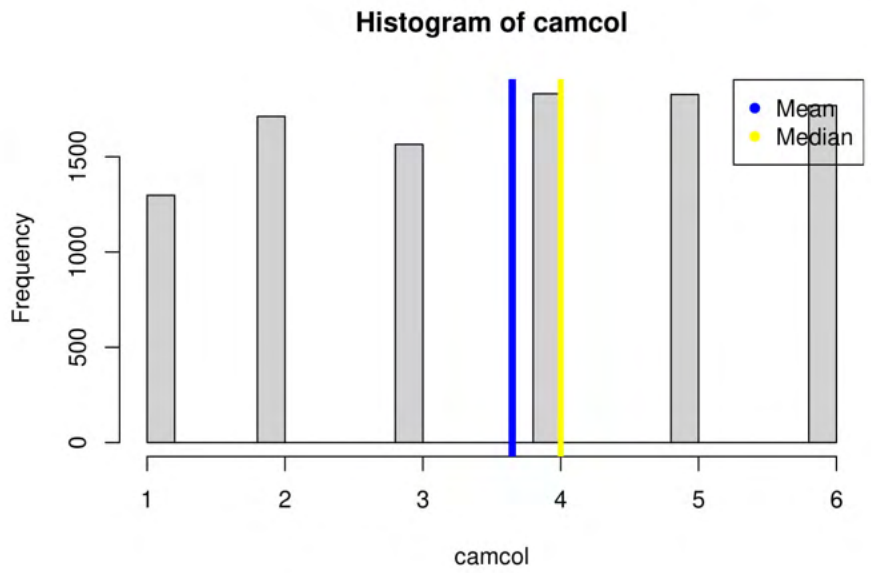



```
## [1] "native"
## [1] "skewness =  -0.0107963784695947"
## [1] "kurtosis =  -2.00008337656893"
## [1] "------"
```

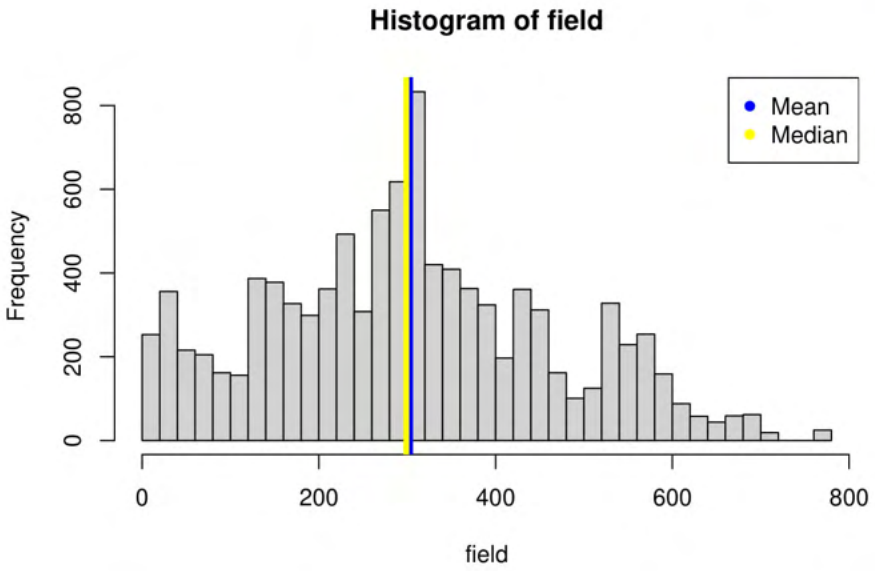
17




```
## [1] "flux"
## [1] "skewness =  -0.850227341226255"
## [1] "kurtosis =  2.30876902019816"
## [1] "------"
```

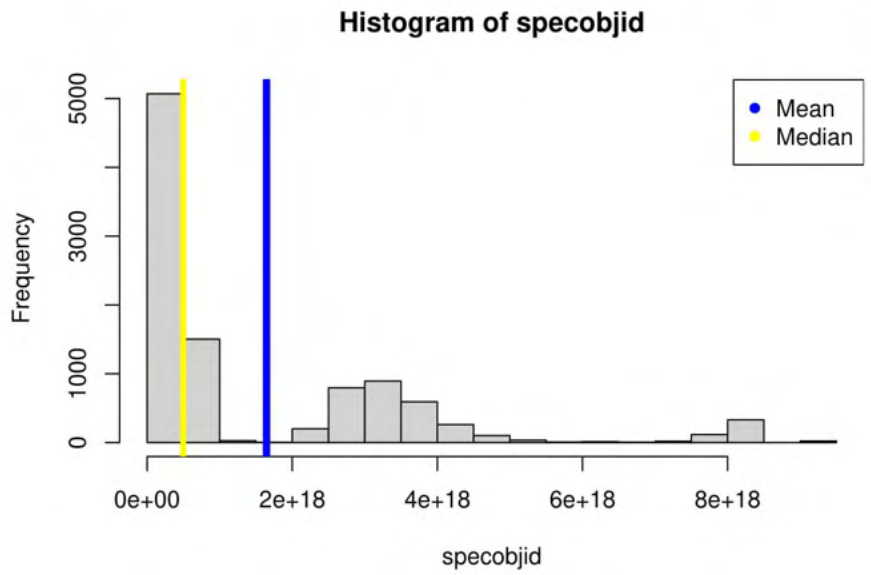



```
## [1] "camcol"
## [1] "skewness =  -0.0994489238759288"
## [1] "kurtosis =  -1.22235189267986"
## [1] "------"
```



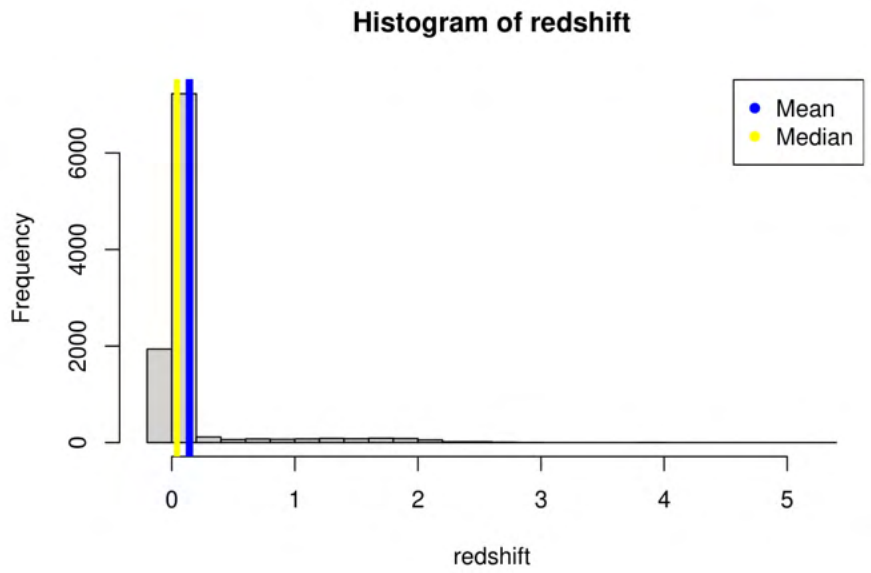

```
## [1] "field"
## [1] "skewness =  0.249622186350446"
## [1] "kurtosis =  -0.478085062001415"
## [1] "------"
```

20

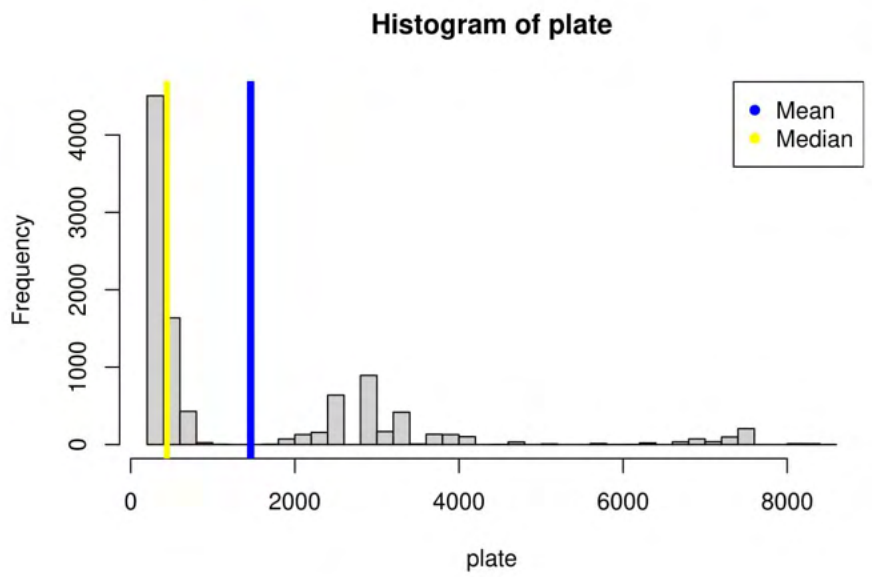



```
## [1] "specobjid"
## [1] "skewness =  1.79232365669917"
## [1] "kurtosis =  2.95629470123879"
## [1] "------"
```

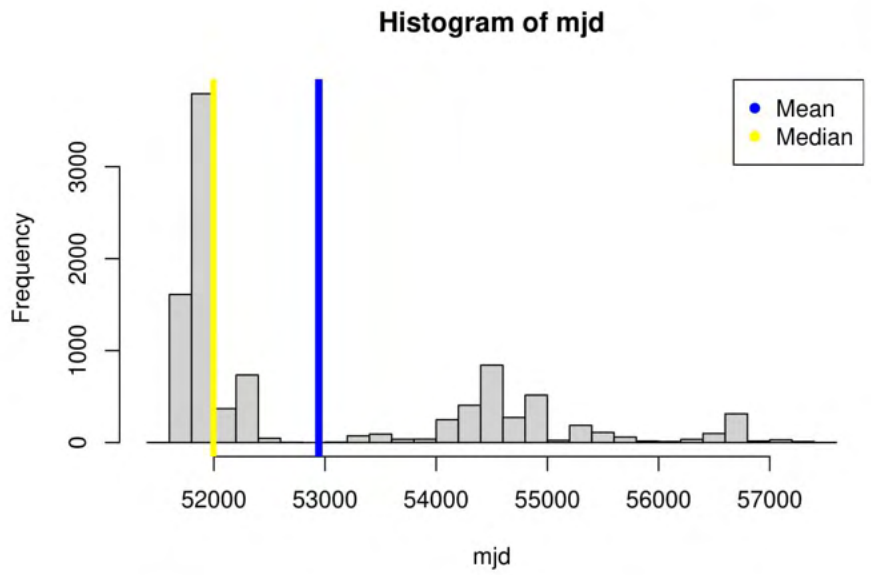
21



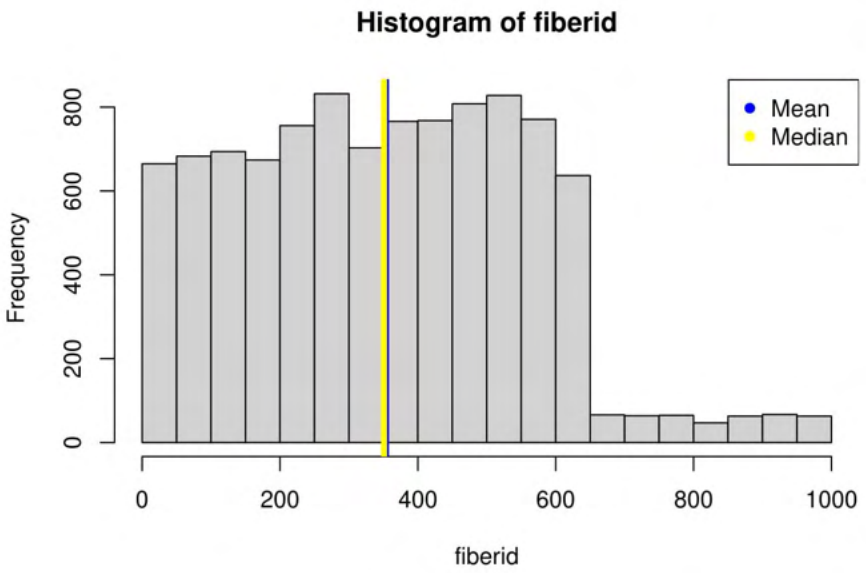

```
## [1] "redshift"
## [1] "skewness =  4.26492142286005"
## [1] "kurtosis =  20.5391088305466"
## [1] "------"
```




```
## [1] "plate"
## [1] "skewness =  1.79243226836025"
## [1] "kurtosis =  2.95694025596412"
## [1] "------"
```




```
## [1] "mjd"
## [1] "skewness =  1.03754105692284"
## [1] "kurtosis =  -0.22557420413968"
## [1] "------"
```




```
## [1] "fiberid"
## [1] "skewness =  0.31046908969123"
## [1] "kurtosis =  -0.309694238495374"
## [1] "------"
```

2. Explore the relationships between the attributes, and between the class and the attributes [8]

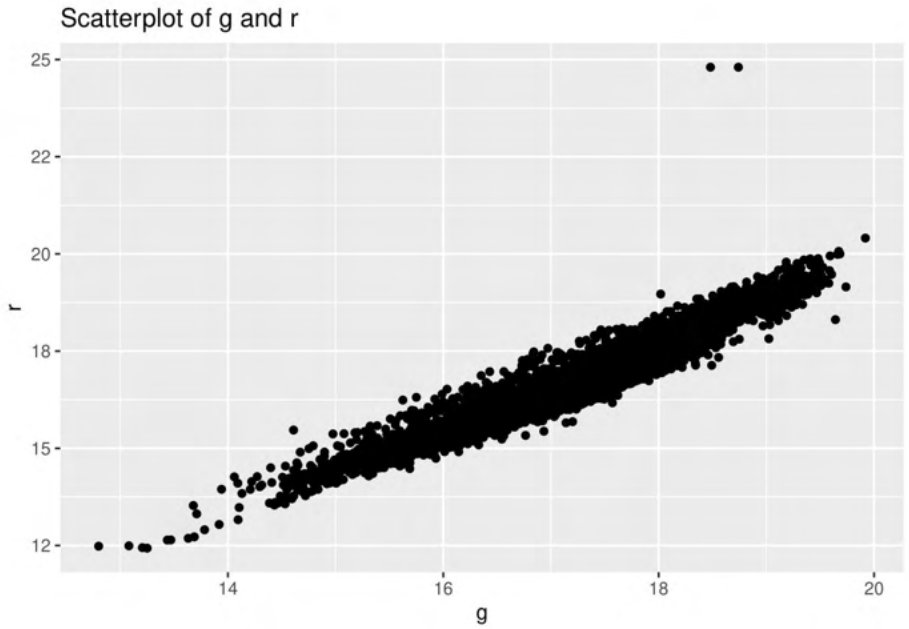
i. Calculate the correlation and produce a scatterplot for the variables: r and g. What does this correlation tell you about the relationships between these variables?

```
cor(cw2$r, cw2$g, use = "complete.obs")
```

```
## [1] 0.96
```

```
ggplot(cw2[,7:8], aes(x=g, y =r)) + geom_point() + ggtitle("Scatterplot of g and r")
```

```
## Warning: Removed 53 rows containing missing values (geom_point).
```



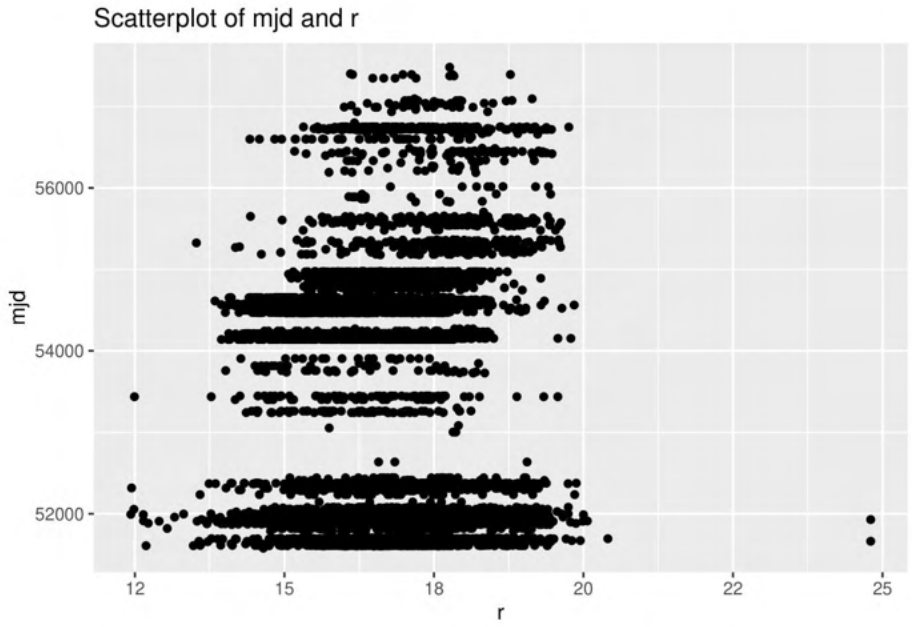
ii.Calculate the correlation and produce a scatterplot for the variables: mjd and r. What does this correlation tell you about the relationships between these variables?

```
cor(cw2$mjd, cw2$r, use = "complete.obs")
```

```
## [1] -0.0092
```

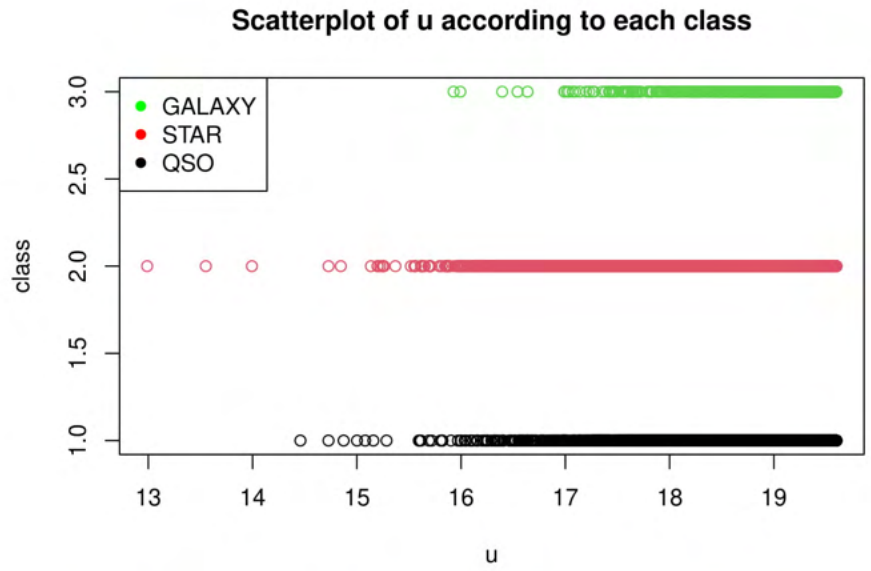
```
ggplot(cw2[,c(8,20)], aes(x=r, y =mjd)) + geom_point() + ggtitle("Scatterplot of mjd and r")
```

```
## Warning: Removed 53 rows containing missing values (geom_point).
```

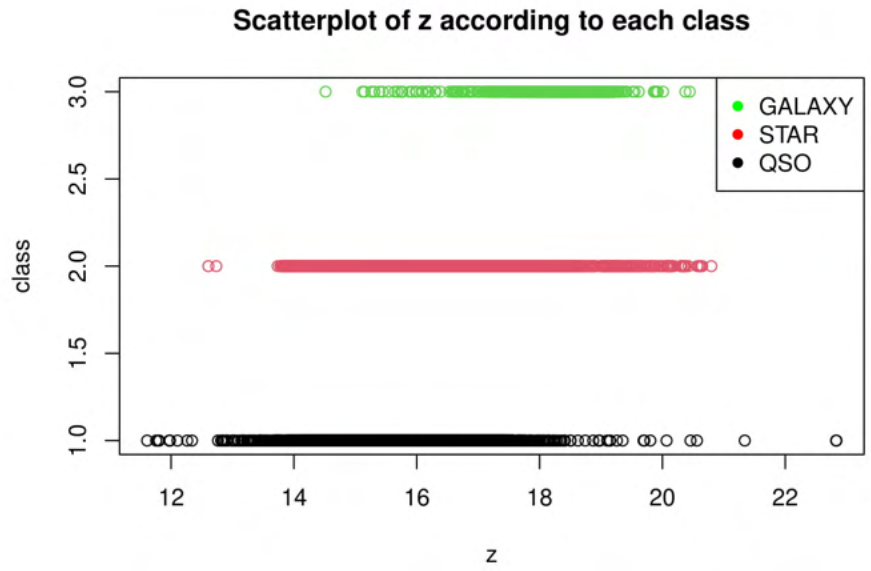


iii.Produce scatterplots between the class variable and u, z, and redshift. What do these three scatterplots tell you about the relationships between these variables and the class?

```
cw2$class2 = ifelse(cw2$class == "GALAXY", 1, ifelse(cw2$class == "STAR", 2, 3))

plot(cw2$u, cw2$class2, col = cw2$class2, xlab="u", ylab="class", main="Scatterplot of u according to e
legend("topleft", c("GALAXY", "STAR", "QSO"), col=c("green", "red", "black"), pch=16)
```

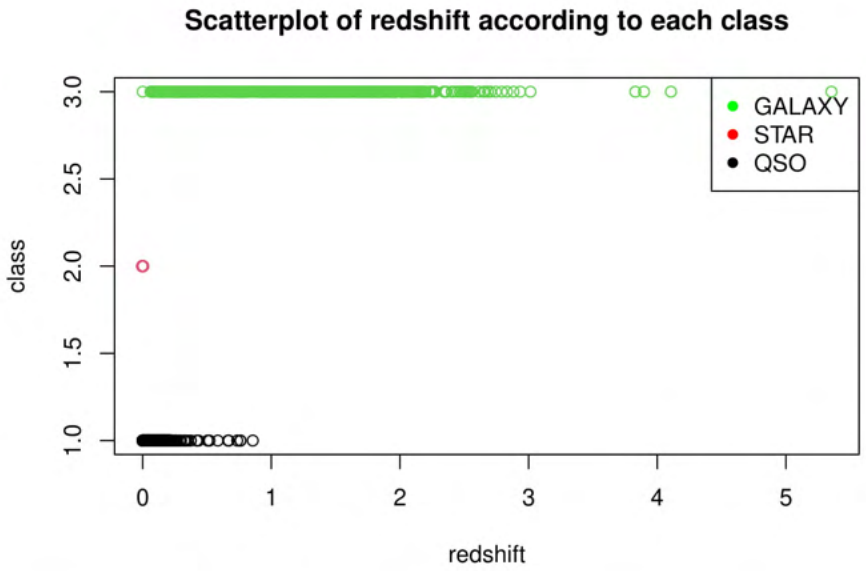



```
plot(cw2$z, cw2$class2, col = cw2$class2, xlab="z", ylab="class", main="Scatterplot of z according to e
legend("topright", c("GALAXY", "STAR", "QSO"), col=c("green", "red", "black"), pch=16)
```



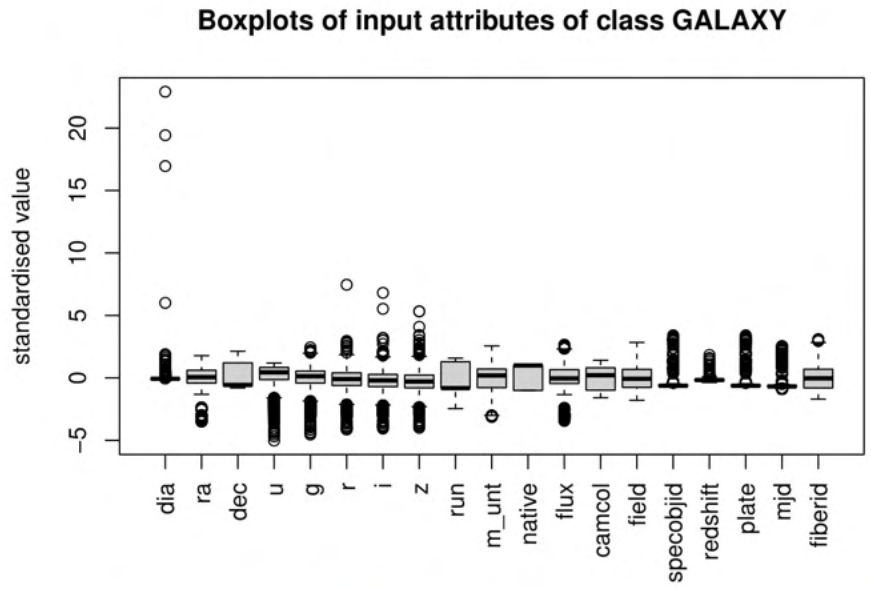

```
plot(cw2$redshift, cw2$class2, col = cw2$class2, xlab="redshift", ylab="class", main="Scatterplot of re
legend("topright", c("GALAXY", "STAR", "QSO"), col=c("green", "red", "black"), pch=16)
```

29

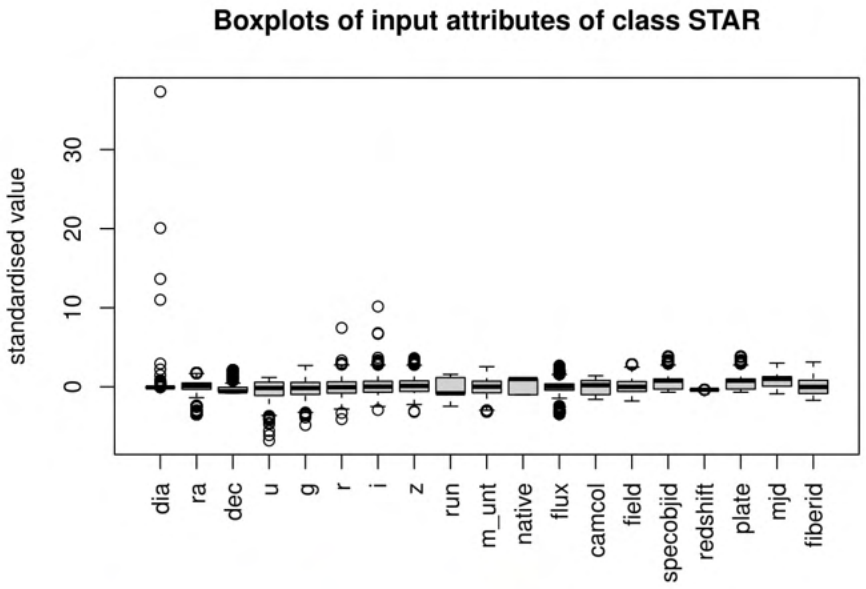


iv.Produce boxplots for all of the appropriate attributes in the dataset grouping each variable according to the class attribute.

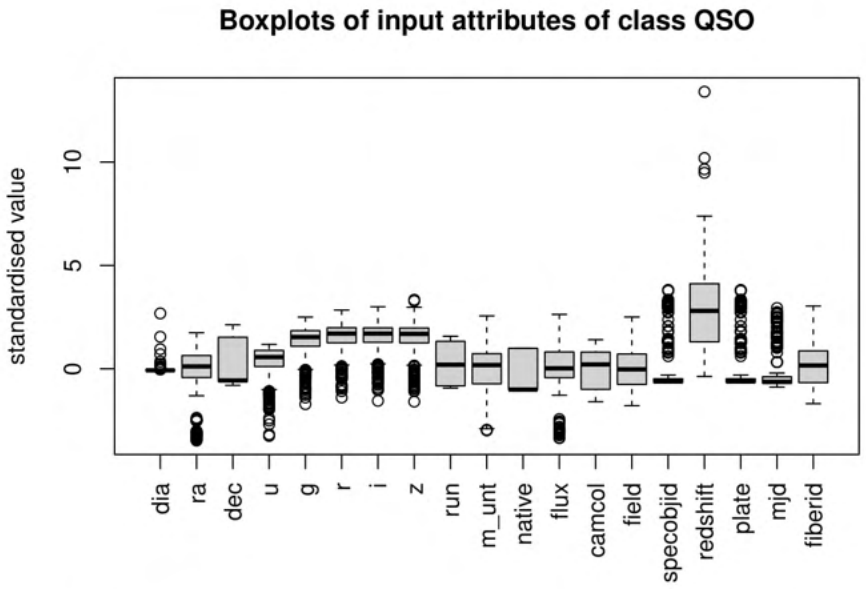
```
cw2 = read.csv("cw_data.csv")
cw2_scaled = data.frame(scale(cw2[,1:21]),cw2[,22])
colnames(cw2_scaled) = colnames(cw2)

cw2_scaled %>% filter(class =="GALAXY") %>% .[,c(2,4:21)] %>%
boxplot(las=3, main="Boxplots of input attributes of class GALAXY", ylab="standardised value")
```




```
cw2_scaled %>% filter(class =="STAR") %>% .[,c(2,4:21)] %>%
boxplot(las=3, main="Boxplots of input attributes of class STAR", ylab="standardised value")
```




```
cw2_scaled %>% filter(class =="QSO") %>% .[,c(2,4:21)] %>%
boxplot(las=3, main="Boxplots of input attributes of class QSO", ylab="standardised value")
```



4. Dealing with missing values in R [6] 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.

1) Replace with 0

```
cw2a = read.csv("cw_data.csv")
cw2a[is.na(cw2a)] = 0
summary(cw2a)
```

```
##      objid                dia              rerun          ra          dec
##  Min.   :1.24e+18   Min.   :     0   Min.   :  0   Min.   :  0   Min.   :-5
##  1st Qu.:1.24e+18   1st Qu.:     0   1st Qu.:301   1st Qu.:157   1st Qu.:-1
##  Median :1.24e+18   Median :     0   Median :301   Median :180   Median : 0
##  Mean   :1.24e+18   Mean   :   678   Mean   :300   Mean   :175   Mean   :15
##  3rd Qu.:1.24e+18   3rd Qu.:   234   3rd Qu.:301   3rd Qu.:201   3rd Qu.:35
##  Max.   :1.24e+18   Max.   :848172   Max.   :301   Max.   :261   Max.   :69
##        u                g               r               i               z
##  Min.   : 0.0   Min.   : 0.0   Min.   : 0.0   Min.   : 0.0   Min.   : 0.0
##  1st Qu.:18.2   1st Qu.:16.8   1st Qu.:16.2   1st Qu.:15.8   1st Qu.:15.6
##  Median :18.8   Median :17.5   Median :16.8   Median :16.5   Median :16.4
##  Mean   :18.5   Mean   :17.3   Mean   :16.8   Mean   :16.5   Mean   :16.3
##  3rd Qu.:19.3   3rd Qu.:18.0   3rd Qu.:17.5   3rd Qu.:17.3   3rd Qu.:17.1
##  Max.   :19.6   Max.   :19.9   Max.   :24.8   Max.   :28.2   Max.   :22.8
```

```
##       run             m_unt              native          flux          camcol
##  Min.   :   0   Min.   :0.00000   Min.   :0.0   Min.   :  0   Min.   :0.0
##  1st Qu.: 752   1st Qu.:0.00018   1st Qu.:0.0   1st Qu.:162   1st Qu.:2.0
##  Median : 756   Median :0.00024   Median :1.0   Median :183   Median :4.0
##  Mean   : 976   Mean   :0.00023   Mean   :0.5   Mean   :183   Mean   :3.6
##  3rd Qu.:1331   3rd Qu.:0.00028   3rd Qu.:1.0   3rd Qu.:212   3rd Qu.:5.0
##  Max.   :1412   Max.   :0.00041   Max.   :1.0   Max.   :319   Max.   :6.0
##      field        specobjid           redshift       plate           mjd
##  Min.   :  0   Min.   :0.00e+00   Min.   :0.0   Min.   :   0   Min.   :    0
##  1st Qu.:183   1st Qu.:3.38e+17   1st Qu.:0.0   1st Qu.: 300   1st Qu.:51900
##  Median :298   Median :4.97e+17   Median :0.0   Median : 441   Median :51997
##  Mean   :301   Mean   :1.64e+18   Mean   :0.1   Mean   :1455   Mean   :52681
##  3rd Qu.:413   3rd Qu.:2.88e+18   3rd Qu.:0.1   3rd Qu.:2559   3rd Qu.:54468
##  Max.   :768   Max.   :9.47e+18   Max.   :5.4   Max.   :8410   Max.   :57481
##     fiberid          class
##  Min.   :   0   Length:10052
##  1st Qu.: 184   Class :character
##  Median : 349   Mode  :character
##  Mean   : 352
##  3rd Qu.: 510
##  Max.   :1000
```

```
sapply(cw2a[1:21], FUN = sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##   0.0e+00   1.3e+04   1.6e+01   4.9e+01   2.5e+01   1.5e+00   1.6e+00   1.6e+00
##         i         z       run     m_unt    native      flux    camcol     field
##   1.6e+00   1.7e+00   2.8e+02   7.2e-05   5.0e-01   5.2e+01   1.7e+00   1.6e+02
## specobjid  redshift     plate       mjd   fiberid
##   2.0e+18   3.9e-01   1.8e+03   4.0e+03   2.1e+02
```

2) Replace with class mean

```
cw2b = read.csv("cw_data.csv")

sub1 = subset(cw2b, cw2$class == "GALAXY")
sub2 = subset(cw2b, cw2$class == "STAR")
sub3 = subset(cw2b, cw2$class == "QSO")
k = ncol(cw2b) - 1

for (i in 1:k) {
  cw2b[is.na(cw2b[,i]) & cw2b$class == "GALAXY", i] = mean(sub1[,i], na.rm = TRUE)
  cw2b[is.na(cw2b[,i]) & cw2b$class == "STAR", i] = mean(sub2[,i], na.rm = TRUE)
  cw2b[is.na(cw2b[,i]) & cw2b$class == "QSO", i] = mean(sub3[,i], na.rm = TRUE)
}

summary(cw2b)
```

```
##      objid              dia             rerun          ra          dec
##  Min.   :1.24e+18   Min.   :   28   Min.   :301   Min.   :  8   Min.   :-5
##  1st Qu.:1.24e+18   1st Qu.:  666   1st Qu.:301   1st Qu.:158   1st Qu.:-1
##  Median :1.24e+18   Median : 1912   Median :301   Median :180   Median : 0
```

```
##  Mean   :1.24e+18   Mean   :  1989   Mean   :301   Mean   :176   Mean   :15
##  3rd Qu.:1.24e+18   3rd Qu.:  2245   3rd Qu.:301   3rd Qu.:201   3rd Qu.:35
##  Max.   :1.24e+18   Max.   :848172   Max.   :301   Max.   :261   Max.   :69
##        u               g               r               i               z
##  Min.   :13.0   Min.   :12.8   Min.   :12.4   Min.   :11.9   Min.   :11.6
##  1st Qu.:18.2   1st Qu.:16.8   1st Qu.:16.2   1st Qu.:15.9   1st Qu.:15.6
##  Median :18.8   Median :17.5   Median :16.8   Median :16.6   Median :16.4
##  Mean   :18.6   Mean   :17.4   Mean   :16.8   Mean   :16.6   Mean   :16.4
##  3rd Qu.:19.3   3rd Qu.:18.0   3rd Qu.:17.5   3rd Qu.:17.3   3rd Qu.:17.1
##  Max.   :19.6   Max.   :19.9   Max.   :24.8   Max.   :28.2   Max.   :22.8
##       run           m_unt            native          flux          camcol
##  Min.   : 308   Min.   :0.00001   Min.   :0.0   Min.   : 10   Min.   :1.0
##  1st Qu.: 752   1st Qu.:0.00018   1st Qu.:0.0   1st Qu.:162   1st Qu.:2.0
##  Median : 756   Median :0.00024   Median :1.0   Median :183   Median :4.0
##  Mean   : 981   Mean   :0.00023   Mean   :0.5   Mean   :184   Mean   :3.6
##  3rd Qu.:1331   3rd Qu.:0.00028   3rd Qu.:1.0   3rd Qu.:212   3rd Qu.:5.0
##  Max.   :1412   Max.   :0.00041   Max.   :1.0   Max.   :319   Max.   :6.0
##      field        specobjid           redshift       plate           mjd
##  Min.   : 11   Min.   :3.00e+17   Min.   :0.0   Min.   : 266   Min.   :51578
##  1st Qu.:185   1st Qu.:3.39e+17   1st Qu.:0.0   1st Qu.: 301   1st Qu.:51900
##  Median :300   Median :4.99e+17   Median :0.0   Median : 443   Median :51997
##  Mean   :302   Mean   :1.65e+18   Mean   :0.1   Mean   :1461   Mean   :52944
##  3rd Qu.:413   3rd Qu.:2.88e+18   3rd Qu.:0.1   3rd Qu.:2559   3rd Qu.:54468
##  Max.   :768   Max.   :9.47e+18   Max.   :5.4   Max.   :8410   Max.   :57481
##     fiberid         class
##  Min.   :   1   Length:10052
##  1st Qu.: 186   Class :character
##  Median : 350   Mode  :character
##  Mean   : 353
##  3rd Qu.: 510
##  Max.   :1000
```

```
sapply(cw2b[1:21], FUN = sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##   0.0e+00   1.3e+04   0.0e+00   4.8e+01   2.5e+01   8.3e-01   9.4e-01   1.1e+00
##         i         z       run     m_unt    native      flux    camcol     field
##   1.1e+00   1.2e+00   2.7e+02   7.1e-05   5.0e-01   5.0e+01   1.7e+00   1.6e+02
## specobjid  redshift     plate       mjd   fiberid
##   2.0e+18   3.9e-01   1.8e+03   1.5e+03   2.1e+02
```

3) Replace with class median

```
cw2c = read.csv("cw_data.csv")

sub1 = subset(cw2c, cw2c$class == "GALAXY")
sub2 = subset(cw2c, cw2c$class == "STAR")
sub3 = subset(cw2c, cw2c$class == "QSO")
k = ncol(cw2) - 1

for (i in 1:k) {
  cw2c[is.na(cw2c[,i]) & cw2c$class == "GALAXY", i] = median(sub1[,i], na.rm = TRUE)
  cw2c[is.na(cw2c[,i]) & cw2c$class == "STAR", i] = median(sub2[,i], na.rm = TRUE)
```

```
  cw2c[is.na(cw2c[,i]) & cw2c$class == "QSO", i] = median(sub3[,i], na.rm = TRUE)
}
summary(cw2c)
```

```
##      objid                dia              rerun           ra           dec
##  Min.   :1.24e+18   Min.   :   28   Min.   :301   Min.   :  8   Min.   :-5
##  1st Qu.:1.24e+18   1st Qu.:  347   1st Qu.:301   1st Qu.:158   1st Qu.:-1
##  Median :1.24e+18   Median :  349   Median :301   Median :180   Median : 0
##  Mean   :1.24e+18   Mean   :  909   Mean   :301   Mean   :176   Mean   :15
##  3rd Qu.:1.24e+18   3rd Qu.:  349   3rd Qu.:301   3rd Qu.:201   3rd Qu.:35
##  Max.   :1.24e+18   Max.   :848172   Max.   :301   Max.   :261   Max.   :69
##        u               g               r               i               z
##  Min.   :13.0   Min.   :12.8   Min.   :12.4   Min.   :11.9   Min.   :11.6
##  1st Qu.:18.2   1st Qu.:16.8   1st Qu.:16.2   1st Qu.:15.9   1st Qu.:15.6
##  Median :18.9   Median :17.5   Median :16.8   Median :16.6   Median :16.4
##  Mean   :18.6   Mean   :17.4   Mean   :16.8   Mean   :16.6   Mean   :16.4
##  3rd Qu.:19.3   3rd Qu.:18.0   3rd Qu.:17.5   3rd Qu.:17.3   3rd Qu.:17.1
##  Max.   :19.6   Max.   :19.9   Max.   :24.8   Max.   :28.2   Max.   :22.8
##       run            m_unt              native          flux          camcol
##  Min.   : 308   Min.   :0.00001   Min.   :0.00   Min.   :  10   Min.   :1.0
##  1st Qu.: 752   1st Qu.:0.00018   1st Qu.:0.00   1st Qu.:162   1st Qu.:2.0
##  Median : 756   Median :0.00024   Median :1.00   Median :183   Median :4.0
##  Mean   : 980   Mean   :0.00023   Mean   :0.51   Mean   :184   Mean   :3.7
##  3rd Qu.:1331   3rd Qu.:0.00028   3rd Qu.:1.00   3rd Qu.:212   3rd Qu.:5.0
##  Max.   :1412   Max.   :0.00041   Max.   :1.00   Max.   :319   Max.   :6.0
##      field        specobjid           redshift       plate           mjd
##  Min.   : 11   Min.   :3.00e+17   Min.   :0.0   Min.   : 266   Min.   :51578
##  1st Qu.:185   1st Qu.:3.39e+17   1st Qu.:0.0   1st Qu.: 301   1st Qu.:51900
##  Median :299   Median :4.97e+17   Median :0.0   Median : 441   Median :51997
##  Mean   :302   Mean   :1.65e+18   Mean   :0.1   Mean   :1461   Mean   :52944
##  3rd Qu.:413   3rd Qu.:2.88e+18   3rd Qu.:0.1   3rd Qu.:2559   3rd Qu.:54468
##  Max.   :768   Max.   :9.47e+18   Max.   :5.4   Max.   :8410   Max.   :57481
##     fiberid         class
##  Min.   :   1   Length:10052
##  1st Qu.: 186   Class :character
##  Median : 350   Mode  :character
##  Mean   : 353
##  3rd Qu.: 510
##  Max.   :1000
```

```
sapply(cw2c[1:21], FUN = sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##   0.0e+00   1.3e+04   0.0e+00   4.8e+01   2.5e+01   8.3e-01   9.4e-01   1.1e+00
##         i         z       run     m_unt    native      flux    camcol     field
##   1.1e+00   1.2e+00   2.7e+02   7.1e-05   5.0e-01   5.0e+01   1.7e+00   1.6e+02
## specobjid  redshift     plate       mjd   fiberid
##   2.0e+18   3.9e-01   1.8e+03   1.5e+03   2.1e+02
```

5. Attribute transformation [6] Using the three datasets generated in 1.4, explore the use of three transformation techniques (mean centering, normalisation and standardisation) to scale the attributes. Define, compare and contrast these approaches, and explain their effects on the data.

1) mean-centering

```
cw2_mca = scale(cw2a[1:(ncol(cw2a) - 1)], center=TRUE, scale=FALSE)
summary(cw2_mca)
```

```
##      objid          dia              rerun           ra             dec
##  Min.   :0   Min.   :  -678   Min.   :-300   Min.   :-175   Min.   :-20
##  1st Qu.:0   1st Qu.:  -678   1st Qu.:   1   1st Qu.: -18   1st Qu.:-15
##  Median :0   Median :  -678   Median :   1   Median :   5   Median :-14
##  Mean   :0   Mean   :     0   Mean   :   0   Mean   :   0   Mean   :  0
##  3rd Qu.:0   3rd Qu.:  -443   3rd Qu.:   1   3rd Qu.:  27   3rd Qu.: 20
##  Max.   :0   Max.   :847494   Max.   :   1   Max.   :  86   Max.   : 54
##        u                g               r               i
##  Min.   :-18.5   Min.   :-17.3   Min.   :-16.8   Min.   :-16.5
##  1st Qu.: -0.4   1st Qu.: -0.5   1st Qu.: -0.6   1st Qu.: -0.7
##  Median :  0.3   Median :  0.2   Median :  0.1   Median :  0.0
##  Mean   :  0.0   Mean   :  0.0   Mean   :  0.0   Mean   :  0.0
##  3rd Qu.:  0.7   3rd Qu.:  0.7   3rd Qu.:  0.8   3rd Qu.:  0.8
##  Max.   :  1.1   Max.   :  2.6   Max.   :  8.1   Max.   : 11.7
##        z              run            m_unt                native
##  Min.   :-16.3   Min.   :-976   Min.   :-0.000233   Min.   :-0.5
##  1st Qu.: -0.7   1st Qu.:-224   1st Qu.:-0.000054   1st Qu.:-0.5
##  Median :  0.0   Median :-220   Median : 0.000011   Median : 0.5
##  Mean   :  0.0   Mean   :   0   Mean   : 0.000000   Mean   : 0.0
##  3rd Qu.:  0.8   3rd Qu.: 355   3rd Qu.: 0.000051   3rd Qu.: 0.5
##  Max.   :  6.5   Max.   : 436   Max.   : 0.000182   Max.   : 0.5
##       flux          camcol         field         specobjid
##  Min.   :-183   Min.   :-3.6   Min.   :-301   Min.   :-1.64e+18
##  1st Qu.: -21   1st Qu.:-1.6   1st Qu.:-118   1st Qu.:-1.30e+18
##  Median :   1   Median : 0.4   Median :  -2   Median :-1.14e+18
##  Mean   :   0   Mean   : 0.0   Mean   :   0   Mean   : 1.89e+02
##  3rd Qu.:  29   3rd Qu.: 1.4   3rd Qu.: 112   3rd Qu.: 1.24e+18
##  Max.   : 136   Max.   : 2.4   Max.   : 467   Max.   : 7.83e+18
##     redshift         plate           mjd            fiberid
##  Min.   :-0.1   Min.   :-1455   Min.   :-52681   Min.   :-352
##  1st Qu.:-0.1   1st Qu.:-1155   1st Qu.:  -781   1st Qu.:-168
##  Median :-0.1   Median :-1014   Median :  -684   Median :  -3
##  Mean   : 0.0   Mean   :    0   Mean   :     0   Mean   :   0
##  3rd Qu.:-0.1   3rd Qu.: 1104   3rd Qu.:  1787   3rd Qu.: 158
##  Max.   : 5.2   Max.   : 6955   Max.   :  4800   Max.   : 648
```

```
apply(cw2_mca[,1:ncol(cw2_mca)], 2, sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##   0.0e+00   1.3e+04   1.6e+01   4.9e+01   2.5e+01   1.5e+00   1.6e+00   1.6e+00
##         i         z       run     m_unt    native      flux    camcol     field
##   1.6e+00   1.7e+00   2.8e+02   7.2e-05   5.0e-01   5.2e+01   1.7e+00   1.6e+02
## specobjid  redshift     plate       mjd   fiberid
##   2.0e+18   3.9e-01   1.8e+03   4.0e+03   2.1e+02
```

```
cw2_mcb = scale(cw2b[1:(ncol(cw2b) - 1)], center=TRUE, scale=FALSE)
summary(cw2_mcb)
```

```
##      objid        dia              rerun         ra            dec
##  Min.   :0   Min.   : -1962   Min.   :0   Min.   :-167   Min.   :-20
##  1st Qu.:0   1st Qu.: -1323   1st Qu.:0   1st Qu.: -18   1st Qu.:-15
##  Median :0   Median :   -77   Median :0   Median :   5   Median :-14
##  Mean   :0   Mean   :     0   Mean   :0   Mean   :   0   Mean   :  0
##  3rd Qu.:0   3rd Qu.:   256   3rd Qu.:0   3rd Qu.:  26   3rd Qu.: 20
##  Max.   :0   Max.   :846182   Max.   :0   Max.   :  85   Max.   : 54
##        u                g               r               i               z
##  Min.   :-5.6   Min.   :-4.6   Min.   :-4.4   Min.   :-4.6   Min.   :-4.8
##  1st Qu.:-0.4   1st Qu.:-0.6   1st Qu.:-0.7   1st Qu.:-0.7   1st Qu.:-0.8
##  Median : 0.2   Median : 0.1   Median : 0.0   Median : 0.0   Median : 0.0
##  Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0
##  3rd Qu.: 0.6   3rd Qu.: 0.6   3rd Qu.: 0.7   3rd Qu.: 0.7   3rd Qu.: 0.7
##  Max.   : 1.0   Max.   : 2.5   Max.   : 8.0   Max.   :11.6   Max.   : 6.4
##       run           m_unt              native          flux
##  Min.   :-673   Min.   :-0.000223   Min.   :-0.5   Min.   :-174
##  1st Qu.:-229   1st Qu.:-0.000054   1st Qu.:-0.5   1st Qu.: -22
##  Median :-225   Median : 0.000010   Median : 0.5   Median :   0
##  Mean   :   0   Mean   : 0.000000   Mean   : 0.0   Mean   :   0
##  3rd Qu.: 350   3rd Qu.: 0.000050   3rd Qu.: 0.5   3rd Qu.:  28
##  Max.   : 431   Max.   : 0.000181   Max.   : 0.5   Max.   : 136
##      camcol          field        specobjid            redshift
##  Min.   :-2.65   Min.   :-291   Min.   :-1.35e+18   Min.   :-0.1
##  1st Qu.:-1.65   1st Qu.:-117   1st Qu.:-1.31e+18   1st Qu.:-0.1
##  Median : 0.35   Median :  -2   Median :-1.15e+18   Median :-0.1
##  Mean   : 0.00   Mean   :   0   Mean   : 5.40e+01   Mean   : 0.0
##  3rd Qu.: 1.35   3rd Qu.: 111   3rd Qu.: 1.23e+18   3rd Qu.:-0.1
##  Max.   : 2.35   Max.   : 466   Max.   : 7.82e+18   Max.   : 5.2
##      plate            mjd           fiberid
##  Min.   :-1195   Min.   :-1366   Min.   :-352
##  1st Qu.:-1160   1st Qu.:-1044   1st Qu.:-167
##  Median :-1018   Median : -947   Median :  -3
##  Mean   :    0   Mean   :    0   Mean   :   0
##  3rd Qu.: 1098   3rd Qu.: 1524   3rd Qu.: 157
##  Max.   : 6949   Max.   : 4537   Max.   : 647
```

```
apply(cw2_mcb[,1:ncol(cw2_mcb)], 2, sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##   0.0e+00   1.3e+04   0.0e+00   4.8e+01   2.5e+01   8.3e-01   9.4e-01   1.1e+00
##         i         z       run     m_unt    native      flux    camcol     field
##   1.1e+00   1.2e+00   2.7e+02   7.1e-05   5.0e-01   5.0e+01   1.7e+00   1.6e+02
## specobjid  redshift     plate       mjd   fiberid
##   2.0e+18   3.9e-01   1.8e+03   1.5e+03   2.1e+02
```

```
cw2_mcc = scale(cw2c[1:(ncol(cw2c) - 1)], center=TRUE, scale=FALSE)
summary(cw2_mcc)
```

```
##      objid        dia              rerun         ra            dec
##  Min.   :0   Min.   :  -881   Min.   :0   Min.   :-167   Min.   :-20
##  1st Qu.:0   1st Qu.:  -561   1st Qu.:0   1st Qu.: -18   1st Qu.:-15
##  Median :0   Median :  -560   Median :0   Median :   5   Median :-14
##  Mean   :0   Mean   :     0   Mean   :0   Mean   :   0   Mean   :  0
```

```
##  3rd Qu.:0   3rd Qu.:  -560   3rd Qu.:0   3rd Qu.:  26   3rd Qu.: 20
##  Max.   :0   Max.   :847263   Max.   :0   Max.   :  85   Max.   : 54
##        u              g              r              i              z
##  Min.   :-5.6   Min.   :-4.6   Min.   :-4.4   Min.   :-4.6   Min.   :-4.8
##  1st Qu.:-0.4   1st Qu.:-0.6   1st Qu.:-0.7   1st Qu.:-0.7   1st Qu.:-0.8
##  Median : 0.2   Median : 0.1   Median : 0.0   Median : 0.0   Median : 0.0
##  Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0
##  3rd Qu.: 0.6   3rd Qu.: 0.6   3rd Qu.: 0.7   3rd Qu.: 0.7   3rd Qu.: 0.7
##  Max.   : 1.0   Max.   : 2.5   Max.   : 8.0   Max.   :11.6   Max.   : 6.4
##       run            m_unt              native           flux
##  Min.   :-672   Min.   :-0.000223   Min.   :-0.51   Min.   :-174
##  1st Qu.:-228   1st Qu.:-0.000054   1st Qu.:-0.51   1st Qu.: -22
##  Median :-224   Median : 0.000011   Median : 0.49   Median :   0
##  Mean   :   0   Mean   : 0.000000   Mean   : 0.00   Mean   :   0
##  3rd Qu.: 351   3rd Qu.: 0.000050   3rd Qu.: 0.49   3rd Qu.:  28
##  Max.   : 432   Max.   : 0.000181   Max.   : 0.49   Max.   : 136
##      camcol          field         specobjid            redshift
##  Min.   :-2.65   Min.   :-291   Min.   :-1.35e+18   Min.   :-0.1
##  1st Qu.:-1.65   1st Qu.:-117   1st Qu.:-1.31e+18   1st Qu.:-0.1
##  Median : 0.35   Median :  -3   Median :-1.15e+18   Median :-0.1
##  Mean   : 0.00   Mean   :   0   Mean   : 1.10e+02   Mean   : 0.0
##  3rd Qu.: 1.35   3rd Qu.: 111   3rd Qu.: 1.23e+18   3rd Qu.:-0.1
##  Max.   : 2.35   Max.   : 466   Max.   : 7.82e+18   Max.   : 5.2
##      plate            mjd           fiberid
##  Min.   :-1195   Min.   :-1366   Min.   :-352
##  1st Qu.:-1160   1st Qu.:-1044   1st Qu.:-167
##  Median :-1020   Median : -947   Median :  -3
##  Mean   :    0   Mean   :    0   Mean   :   0
##  3rd Qu.: 1098   3rd Qu.: 1524   3rd Qu.: 157
##  Max.   : 6949   Max.   : 4537   Max.   : 647
```

```
apply(cw2_mcc[,1:ncol(cw2_mcc)], 2, sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##   0.0e+00   1.3e+04   0.0e+00   4.8e+01   2.5e+01   8.3e-01   9.4e-01   1.1e+00
##         i         z       run     m_unt    native      flux    camcol     field
##   1.1e+00   1.2e+00   2.7e+02   7.1e-05   5.0e-01   5.0e+01   1.7e+00   1.6e+02
## specobjid  redshift     plate       mjd   fiberid
##   2.0e+18   3.9e-01   1.8e+03   1.5e+03   2.1e+02
```

2) normalisation

```
k = ncol(cw2a) - 1
cw2_na = cw2a
for (i in 1:(ncol(cw2a) -1)) {
cw2_na[,i] = (cw2_na[,i] - min(cw2_na[,i])) / (max(cw2_na[,i]) - min(cw2_na[,i]))
}

summary(cw2_na)
```

```
##      objid            dia        rerun         ra            dec
##  Min.   : NA     Min.   :0   Min.   :0   Min.   :0.00   Min.   :0.00
```

```
##  1st Qu.: NA     1st Qu.:0   1st Qu.:1   1st Qu.:0.60   1st Qu.:0.07
##  Median : NA     Median :0   Median :1   Median :0.69   Median :0.08
##  Mean   :NaN     Mean   :0   Mean   :1   Mean   :0.67   Mean   :0.27
##  3rd Qu.: NA     3rd Qu.:0   3rd Qu.:1   3rd Qu.:0.77   3rd Qu.:0.54
##  Max.   : NA     Max.   :1   Max.   :1   Max.   :1.00   Max.   :1.00
##  NA's   :10052
##        u               g               r               i               z
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.93   1st Qu.:0.84   1st Qu.:0.65   1st Qu.:0.56   1st Qu.:0.68
##  Median :0.96   Median :0.88   Median :0.68   Median :0.59   Median :0.72
##  Mean   :0.95   Mean   :0.87   Mean   :0.68   Mean   :0.59   Mean   :0.72
##  3rd Qu.:0.98   3rd Qu.:0.90   3rd Qu.:0.71   3rd Qu.:0.61   3rd Qu.:0.75
##  Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
##
##       run           m_unt          native         flux          camcol
##  Min.   :0.00   Min.   :0.00   Min.   :0.0   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.53   1st Qu.:0.43   1st Qu.:0.0   1st Qu.:0.51   1st Qu.:0.33
##  Median :0.54   Median :0.59   Median :1.0   Median :0.57   Median :0.67
##  Mean   :0.69   Mean   :0.56   Mean   :0.5   Mean   :0.57   Mean   :0.61
##  3rd Qu.:0.94   3rd Qu.:0.68   3rd Qu.:1.0   3rd Qu.:0.66   3rd Qu.:0.83
##  Max.   :1.00   Max.   :1.00   Max.   :1.0   Max.   :1.00   Max.   :1.00
##
##      field        specobjid       redshift        plate           mjd
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.24   1st Qu.:0.04   1st Qu.:0.00   1st Qu.:0.04   1st Qu.:0.90
##  Median :0.39   Median :0.05   Median :0.01   Median :0.05   Median :0.90
##  Mean   :0.39   Mean   :0.17   Mean   :0.03   Mean   :0.17   Mean   :0.92
##  3rd Qu.:0.54   3rd Qu.:0.30   3rd Qu.:0.02   3rd Qu.:0.30   3rd Qu.:0.95
##  Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
##
##     fiberid         class
##  Min.   :0.00   Length:10052
##  1st Qu.:0.18   Class :character
##  Median :0.35   Mode  :character
##  Mean   :0.35
##  3rd Qu.:0.51
##  Max.   :1.00
##
```

```
apply(cw2_na[,1:(ncol(cw2_na) - 1)], 2, sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##        NA     0.016     0.055     0.188     0.340     0.079     0.078     0.065
##         i         z       run     m_unt    native      flux    camcol     field
##     0.058     0.074     0.199     0.174     0.500     0.162     0.280     0.213
## specobjid  redshift     plate       mjd   fiberid
##     0.212     0.072     0.213     0.070     0.207
```

```
k = ncol(cw2b) - 1
cw2_nb = cw2b
for (i in 1:(ncol(cw2b) -1)) {
cw2_nb[,i] = (cw2_nb[,i] - min(cw2_nb[,i])) / (max(cw2_nb[,i]) - min(cw2_nb[,i]))
}
```

```
summary(cw2_nb)
```

```
##      objid            dia        rerun            ra            dec
##  Min.   : NA     Min.   :0   Min.   : NA     Min.   :0.00   Min.   :0.00
##  1st Qu.: NA     1st Qu.:0   1st Qu.: NA     1st Qu.:0.59   1st Qu.:0.07
##  Median : NA     Median :0   Median : NA     Median :0.68   Median :0.08
##  Mean   :NaN     Mean   :0   Mean   :NaN     Mean   :0.66   Mean   :0.27
##  3rd Qu.: NA     3rd Qu.:0   3rd Qu.: NA     3rd Qu.:0.76   3rd Qu.:0.54
##  Max.   : NA     Max.   :1   Max.   : NA     Max.   :1.00   Max.   :1.00
##  NA's   :10052               NA's   :10052
##        u               g               r               i               z
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.79   1st Qu.:0.56   1st Qu.:0.30   1st Qu.:0.24   1st Qu.:0.36
##  Median :0.89   Median :0.66   Median :0.36   Median :0.28   Median :0.43
##  Mean   :0.85   Mean   :0.64   Mean   :0.36   Mean   :0.29   Mean   :0.43
##  3rd Qu.:0.95   3rd Qu.:0.73   3rd Qu.:0.41   3rd Qu.:0.33   3rd Qu.:0.49
##  Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
##
##       run             m_unt          native         flux          camcol
##  Min.   :0.00   Min.   :0.00   Min.   :0.0   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.40   1st Qu.:0.42   1st Qu.:0.0   1st Qu.:0.49   1st Qu.:0.20
##  Median :0.41   Median :0.58   Median :1.0   Median :0.56   Median :0.60
##  Mean   :0.61   Mean   :0.55   Mean   :0.5   Mean   :0.56   Mean   :0.53
##  3rd Qu.:0.93   3rd Qu.:0.68   3rd Qu.:1.0   3rd Qu.:0.65   3rd Qu.:0.80
##  Max.   :1.00   Max.   :1.00   Max.   :1.0   Max.   :1.00   Max.   :1.00
##
##      field        specobjid       redshift        plate           mjd
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.23   1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.05
##  Median :0.38   Median :0.02   Median :0.01   Median :0.02   Median :0.07
##  Mean   :0.38   Mean   :0.15   Mean   :0.03   Mean   :0.15   Mean   :0.23
##  3rd Qu.:0.53   3rd Qu.:0.28   3rd Qu.:0.02   3rd Qu.:0.28   3rd Qu.:0.49
##  Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
##
##     fiberid         class
##  Min.   :0.00   Length:10052
##  1st Qu.:0.19   Class :character
##  Median :0.35   Mode  :character
##  Mean   :0.35
##  3rd Qu.:0.51
##  Max.   :1.00
##
```

```
apply(cw2_nb[,1:(ncol(cw2_nb) - 1)], 2, sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##        NA     0.016        NA     0.189     0.340     0.125     0.132     0.086
##         i         z       run     m_unt    native      flux    camcol     field
##     0.070     0.107     0.247     0.175     0.499     0.162     0.332     0.214
## specobjid  redshift     plate       mjd   fiberid
##     0.219     0.072     0.219     0.256     0.206
```

```
k = ncol(cw2c) - 1
cw2_nc = cw2c
for (i in 1:(ncol(cw2c) -1)) {
cw2_nc[,i] = (cw2_nc[,i] - min(cw2_nc[,i])) / (max(cw2_nc[,i]) - min(cw2_nc[,i]))
}

summary(cw2_nc)
```

```
##      objid            dia        rerun            ra            dec
##  Min.   : NA     Min.   :0   Min.   : NA     Min.   :0.00   Min.   :0.00
##  1st Qu.: NA     1st Qu.:0   1st Qu.: NA     1st Qu.:0.59   1st Qu.:0.07
##  Median : NA     Median :0   Median : NA     Median :0.68   Median :0.08
##  Mean   :NaN     Mean   :0   Mean   :NaN     Mean   :0.66   Mean   :0.27
##  3rd Qu.: NA     3rd Qu.:0   3rd Qu.: NA     3rd Qu.:0.76   3rd Qu.:0.54
##  Max.   : NA     Max.   :1   Max.   : NA     Max.   :1.00   Max.   :1.00
##  NA's   :10052               NA's   :10052
##        u                g                r                i                z
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.79   1st Qu.:0.56   1st Qu.:0.30   1st Qu.:0.24   1st Qu.:0.36
##  Median :0.89   Median :0.66   Median :0.36   Median :0.28   Median :0.43
##  Mean   :0.85   Mean   :0.64   Mean   :0.36   Mean   :0.29   Mean   :0.43
##  3rd Qu.:0.95   3rd Qu.:0.73   3rd Qu.:0.41   3rd Qu.:0.33   3rd Qu.:0.49
##  Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
##
##       run            m_unt          native          flux          camcol
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.40   1st Qu.:0.42   1st Qu.:0.00   1st Qu.:0.49   1st Qu.:0.20
##  Median :0.41   Median :0.58   Median :1.00   Median :0.56   Median :0.60
##  Mean   :0.61   Mean   :0.55   Mean   :0.51   Mean   :0.56   Mean   :0.53
##  3rd Qu.:0.93   3rd Qu.:0.68   3rd Qu.:1.00   3rd Qu.:0.65   3rd Qu.:0.80
##  Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
##
##      field        specobjid       redshift        plate           mjd
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.23   1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.05
##  Median :0.38   Median :0.02   Median :0.01   Median :0.02   Median :0.07
##  Mean   :0.38   Mean   :0.15   Mean   :0.03   Mean   :0.15   Mean   :0.23
##  3rd Qu.:0.53   3rd Qu.:0.28   3rd Qu.:0.02   3rd Qu.:0.28   3rd Qu.:0.49
##  Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
##
##     fiberid         class
##  Min.   :0.00   Length:10052
##  1st Qu.:0.19   Class :character
##  Median :0.35   Mode  :character
##  Mean   :0.35
##  3rd Qu.:0.51
##  Max.   :1.00
##
```

```
apply(cw2_nc[,1:(ncol(cw2_nc) - 1)], 2, sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
```

```
##        NA     0.016        NA     0.189     0.340     0.125     0.132     0.086
##         i         z       run     m_unt    native      flux    camcol     field
##     0.070     0.107     0.247     0.175     0.500     0.162     0.332     0.214
## specobjid  redshift     plate       mjd   fiberid
##     0.219     0.072     0.219     0.256     0.206
```

3) standardisation

```
cw2_sa = scale(cw2a[1:(ncol(cw2a)-1)])
cw2_sb = scale(cw2b[1:(ncol(cw2b)-1)])
cw2_sc = scale(cw2c[1:(ncol(cw2c)-1)])

summary(cw2_sa)
```

```
##      objid            dia          rerun              ra            dec
##  Min.   : NA     Min.   : 0   Min.   :-18.3   Min.   :-3.6   Min.   :-0.80
##  1st Qu.: NA     1st Qu.: 0   1st Qu.:  0.1   1st Qu.:-0.4   1st Qu.:-0.61
##  Median : NA     Median : 0   Median :  0.1   Median : 0.1   Median :-0.57
##  Mean   :NaN     Mean   : 0   Mean   :  0.0   Mean   : 0.0   Mean   : 0.00
##  3rd Qu.: NA     3rd Qu.: 0   3rd Qu.:  0.1   3rd Qu.: 0.5   3rd Qu.: 0.79
##  Max.   : NA     Max.   :64   Max.   :  0.1   Max.   : 1.8   Max.   : 2.14
##  NA's   :10052
##        u                g                r                i                z
##  Min.   :-12.0   Min.   :-11.1   Min.   :-10.3   Min.   :-10.1   Min.   :-9.7
##  1st Qu.: -0.2   1st Qu.: -0.3   1st Qu.: -0.4   1st Qu.: -0.4   1st Qu.:-0.4
##  Median :  0.2   Median :  0.1   Median :  0.1   Median :  0.0   Median : 0.0
##  Mean   :  0.0   Mean   :  0.0   Mean   :  0.0   Mean   :  0.0   Mean   : 0.0
##  3rd Qu.:  0.5   3rd Qu.:  0.5   3rd Qu.:  0.5   3rd Qu.:  0.5   3rd Qu.: 0.5
##  Max.   :  0.7   Max.   :  1.7   Max.   :  5.0   Max.   :  7.2   Max.   : 3.8
##
##       run            m_unt          native        flux          camcol
##  Min.   :-3.5   Min.   :-3.2   Min.   :-1   Min.   :-3.5   Min.   :-2.16
##  1st Qu.:-0.8   1st Qu.:-0.8   1st Qu.:-1   1st Qu.:-0.4   1st Qu.:-0.97
##  Median :-0.8   Median : 0.2   Median : 1   Median : 0.0   Median : 0.22
##  Mean   : 0.0   Mean   : 0.0   Mean   : 0   Mean   : 0.0   Mean   : 0.00
##  3rd Qu.: 1.3   3rd Qu.: 0.7   3rd Qu.: 1   3rd Qu.: 0.6   3rd Qu.: 0.81
##  Max.   : 1.6   Max.   : 2.5   Max.   : 1   Max.   : 2.6   Max.   : 1.41
##
##      field          specobjid       redshift        plate           mjd
##  Min.   :-1.84   Min.   :-0.8   Min.   :-0.4   Min.   :-0.8   Min.   :-13.1
##  1st Qu.:-0.72   1st Qu.:-0.6   1st Qu.:-0.4   1st Qu.:-0.6   1st Qu.: -0.2
##  Median :-0.01   Median :-0.6   Median :-0.3   Median :-0.6   Median : -0.2
##  Mean   : 0.00   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   :  0.0
##  3rd Qu.: 0.69   3rd Qu.: 0.6   3rd Qu.:-0.1   3rd Qu.: 0.6   3rd Qu.:  0.4
##  Max.   : 2.86   Max.   : 3.9   Max.   :13.4   Max.   : 3.9   Max.   :  1.2
##
##     fiberid
##  Min.   :-1.70
##  1st Qu.:-0.81
##  Median :-0.01
##  Mean   : 0.00
##  3rd Qu.: 0.76
##  Max.   : 3.13
##
```

```
summary(cw2_sb)
```

```
##      objid            dia         rerun            ra          dec
##  Min.   : NA     Min.   : 0   Min.   : NA     Min.   :-3.5   Min.   :-0.80
##  1st Qu.: NA     1st Qu.: 0   1st Qu.: NA     1st Qu.:-0.4   1st Qu.:-0.61
##  Median : NA     Median : 0   Median : NA     Median : 0.1   Median :-0.57
##  Mean   :NaN     Mean   : 0   Mean   :NaN     Mean   : 0.0   Mean   : 0.00
##  3rd Qu.: NA     3rd Qu.: 0   3rd Qu.: NA     3rd Qu.: 0.5   3rd Qu.: 0.79
##  Max.   : NA     Max.   :64   Max.   : NA     Max.   : 1.8   Max.   : 2.14
##  NA's   :10052                NA's   :10052
##        u               g               r               i               z
##  Min.   :-6.8   Min.   :-4.8   Min.   :-4.1   Min.   :-4.1   Min.   :-4.0
##  1st Qu.:-0.5   1st Qu.:-0.6   1st Qu.:-0.6   1st Qu.:-0.6   1st Qu.:-0.7
##  Median : 0.3   Median : 0.1   Median : 0.0   Median : 0.0   Median : 0.0
##  Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0
##  3rd Qu.: 0.8   3rd Qu.: 0.7   3rd Qu.: 0.6   3rd Qu.: 0.6   3rd Qu.: 0.6
##  Max.   : 1.2   Max.   : 2.7   Max.   : 7.5   Max.   :10.2   Max.   : 5.3
##
##       run             m_unt           native           flux          camcol
##  Min.   :-2.47   Min.   :-3.16   Min.   :-1.01   Min.   :-3.5   Min.   :-1.59
##  1st Qu.:-0.84   1st Qu.:-0.77   1st Qu.:-1.01   1st Qu.:-0.4   1st Qu.:-0.99
##  Median :-0.83   Median : 0.14   Median : 1.00   Median : 0.0   Median : 0.21
##  Mean   : 0.00   Mean   : 0.00   Mean   : 0.00   Mean   : 0.0   Mean   : 0.00
##  3rd Qu.: 1.28   3rd Qu.: 0.71   3rd Qu.: 1.00   3rd Qu.: 0.6   3rd Qu.: 0.81
##  Max.   : 1.58   Max.   : 2.57   Max.   : 1.00   Max.   : 2.7   Max.   : 1.42
##
##      field         specobjid       redshift        plate           mjd
##  Min.   :-1.80   Min.   :-0.7   Min.   :-0.4   Min.   :-0.7   Min.   :-0.91
##  1st Qu.:-0.72   1st Qu.:-0.6   1st Qu.:-0.4   1st Qu.:-0.6   1st Qu.:-0.69
##  Median :-0.01   Median :-0.6   Median :-0.3   Median :-0.6   Median :-0.63
##  Mean   : 0.00   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.00
##  3rd Qu.: 0.68   3rd Qu.: 0.6   3rd Qu.:-0.1   3rd Qu.: 0.6   3rd Qu.: 1.01
##  Max.   : 2.87   Max.   : 3.9   Max.   :13.4   Max.   : 3.9   Max.   : 3.01
##
##     fiberid
##  Min.   :-1.71
##  1st Qu.:-0.81
##  Median :-0.02
##  Mean   : 0.00
##  3rd Qu.: 0.76
##  Max.   : 3.14
##
```

```
summary(cw2_sc)
```

```
##      objid            dia         rerun            ra          dec
##  Min.   : NA     Min.   : 0   Min.   : NA     Min.   :-3.5   Min.   :-0.80
##  1st Qu.: NA     1st Qu.: 0   1st Qu.: NA     1st Qu.:-0.4   1st Qu.:-0.61
##  Median : NA     Median : 0   Median : NA     Median : 0.1   Median :-0.57
##  Mean   :NaN     Mean   : 0   Mean   :NaN     Mean   : 0.0   Mean   : 0.00
##  3rd Qu.: NA     3rd Qu.: 0   3rd Qu.: NA     3rd Qu.: 0.5   3rd Qu.: 0.79
##  Max.   : NA     Max.   :64   Max.   : NA     Max.   : 1.8   Max.   : 2.14
```

```
##  NA's   :10052                NA's   :10052
##        u                g                r                i                z
##  Min.   :-6.8   Min.   :-4.8   Min.   :-4.1   Min.   :-4.1   Min.   :-4.0
##  1st Qu.:-0.5   1st Qu.:-0.6   1st Qu.:-0.6   1st Qu.:-0.6   1st Qu.:-0.7
##  Median : 0.3   Median : 0.1   Median : 0.0   Median : 0.0   Median : 0.0
##  Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0
##  3rd Qu.: 0.8   3rd Qu.: 0.7   3rd Qu.: 0.6   3rd Qu.: 0.6   3rd Qu.: 0.6
##  Max.   : 1.2   Max.   : 2.7   Max.   : 7.5   Max.   :10.2   Max.   : 5.3
##
##       run             m_unt            native           flux          camcol
##  Min.   :-2.46   Min.   :-3.16   Min.   :-1.01   Min.   :-3.5   Min.   :-1.59
##  1st Qu.:-0.83   1st Qu.:-0.77   1st Qu.:-1.01   1st Qu.:-0.4   1st Qu.:-0.99
##  Median :-0.82   Median : 0.15   Median : 0.99   Median : 0.0   Median : 0.21
##  Mean   : 0.00   Mean   : 0.00   Mean   : 0.00   Mean   : 0.0   Mean   : 0.00
##  3rd Qu.: 1.29   3rd Qu.: 0.71   3rd Qu.: 0.99   3rd Qu.: 0.6   3rd Qu.: 0.81
##  Max.   : 1.58   Max.   : 2.56   Max.   : 0.99   Max.   : 2.7   Max.   : 1.41
##
##      field          specobjid       redshift        plate           mjd
##  Min.   :-1.80   Min.   :-0.7   Min.   :-0.4   Min.   :-0.7   Min.   :-0.90
##  1st Qu.:-0.72   1st Qu.:-0.6   1st Qu.:-0.4   1st Qu.:-0.6   1st Qu.:-0.69
##  Median :-0.02   Median :-0.6   Median :-0.3   Median :-0.6   Median :-0.63
##  Mean   : 0.00   Mean   : 0.0   Mean   : 0.0   Mean   : 0.0   Mean   : 0.00
##  3rd Qu.: 0.68   3rd Qu.: 0.6   3rd Qu.:-0.1   3rd Qu.: 0.6   3rd Qu.: 1.01
##  Max.   : 2.87   Max.   : 3.9   Max.   :13.4   Max.   : 3.9   Max.   : 3.00
##
##     fiberid
##  Min.   :-1.71
##  1st Qu.:-0.81
##  Median :-0.02
##  Mean   : 0.00
##  3rd Qu.: 0.76
##  Max.   : 3.14
##
```

```
apply(cw2_sa[,1:ncol(cw2_sa)], 2, sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##        NA         1         1         1         1         1         1         1
##         i         z       run     m_unt    native      flux    camcol     field
##         1         1         1         1         1         1         1         1
## specobjid  redshift     plate       mjd   fiberid
##         1         1         1         1         1
```

```
apply(cw2_sb[,1:ncol(cw2_sb)], 2, sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##        NA         1        NA         1         1         1         1         1
##         i         z       run     m_unt    native      flux    camcol     field
##         1         1         1         1         1         1         1         1
## specobjid  redshift     plate       mjd   fiberid
##         1         1         1         1         1
```

```
apply(cw2_sc[,1:ncol(cw2_sc)], 2, sd)
```

```
##     objid       dia     rerun        ra       dec         u         g         r
##        NA         1        NA         1         1         1         1         1
##         i         z       run     m_unt    native      flux    camcol     field
##         1         1         1         1         1         1         1         1
## specobjid  redshift     plate       mjd   fiberid
##         1         1         1         1         1
```

6.

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.

```
cw2 = read.csv("cw_data.csv")
cw2_6 = cw2[,c(1,3:22)]
cw2_6$na_count = apply(cw2_6, 1, function(x) sum(is.na(x)))
cw2_6 = cw2_6[cw2_6$na_count < 5,]
cw2_6 = unique(cw2_6)
cw2_6 = cw2_6[, c(1,2,4,5,7:21)]
nrow(cw2_6)
```

```
## [1] 10000
```

```
ncol(cw2_6)
```

```
## [1] 19
```

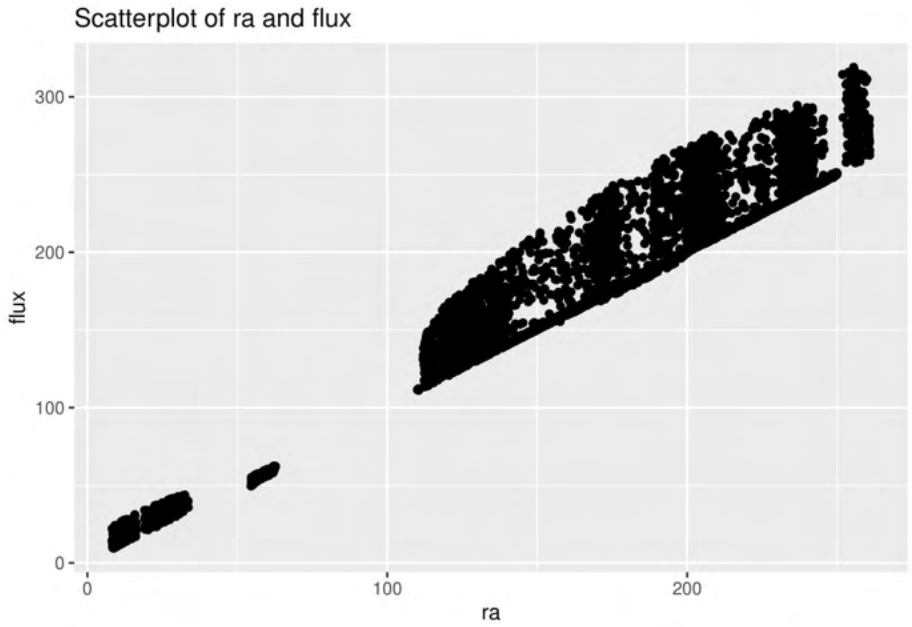
ii. . Start from the raw data, use correlations between attributes to reduce the number of attributes. Try to reduce the dataset to contain only uncorrelated attributes and no missing values. Explain your choices and its effects on the dataset.

```
cor(cw2$ra, cw2$flux, use = "complete.obs")
```

```
## [1] 0.95
```

```
ggplot(cw2[,c(4,14)], aes(x=ra, y =flux)) + geom_point() + ggtitle("Scatterplot of ra and flux")
```

```
## Warning: Removed 50 rows containing missing values (geom_point).
```



7. Starting from the raw data, perform appropriate pre-processing steps first, and then use Principal Component Analysis. Explain your process, along with the results obtained.

```
cw2 = read.csv("cw_data.csv")
cw2_6 = cw2[,c(1,3:22)]
cw2_6$na_count = apply(cw2_6, 1, function(x) sum(is.na(x)))
cw2_6 = cw2_6[cw2_6$na_count < 5,]
cw2_6 = unique(cw2_6)
cw2_6 = cw2_6[, c(1,2,4,5,7:21)]

sub1 = subset(cw2_6, cw2_6$class == "GALAXY")
sub2 = subset(cw2_6, cw2_6$class == "STAR")
sub3 = subset(cw2_6, cw2_6$class == "QSO")
k = ncol(cw2_6) - 1

for (i in 1:k) {
  cw2_6[is.na(cw2_6[,i]) & cw2_6$class == "GALAXY", i] = median(sub1[,i], na.rm = TRUE)
  cw2_6[is.na(cw2_6[,i]) & cw2_6$class == "STAR", i] = median(sub2[,i], na.rm = TRUE)
  cw2_6[is.na(cw2_6[,i]) & cw2_6$class == "QSO", i] = median(sub3[,i], na.rm = TRUE)
}

cw2_6s = data.frame(scale(cw2_6[,1:18]),cw2_6[,19])
cw2_6s = cw2_6s[,3:19]
colnames(cw2_6s)[17] ="class"
```

i. Compare the effects of Principal Component Analysis when looking at PCA as a transformation tech-

nique (i.e. considering all PCs) and as a dimensionality reduction technique in which the data will be reduced to 12 dimensions (i.e: PC1-PC12).

```
pca_t = prcomp(cw2_6s[,1:16], scale=TRUE)
summary(pca_t)
```

```
## Importance of components:
##                           PC1   PC2   PC3   PC4    PC5    PC6    PC7    PC8
## Standard deviation     1.921 1.778 1.465 1.395 1.1653 0.9996 0.9230 0.7256
## Proportion of Variance 0.231 0.198 0.134 0.122 0.0849 0.0624 0.0532 0.0329
## Cumulative Proportion  0.231 0.428 0.562 0.684 0.7687 0.8311 0.8844 0.9173
##                           PC9   PC10   PC11    PC12   PC13    PC14    PC15
## Standard deviation     0.6710 0.6536 0.4981 0.35857 0.2041 0.12868 0.10494
## Proportion of Variance 0.0281 0.0267 0.0155 0.00804 0.0026 0.00103 0.00069
## Cumulative Proportion  0.9454 0.9721 0.9876 0.99567 0.9983 0.99931 1.00000
##                           PC16
## Standard deviation     0.000528
## Proportion of Variance 0.000000
## Cumulative Proportion  1.000000
```

```
pca_12 = prcomp(cw2_6s[,1:16], rank=12, scale=TRUE)
summary(pca_12)
```

```
## Importance of first k=12 (out of 16) components:
##                           PC1   PC2   PC3   PC4    PC5    PC6    PC7    PC8
## Standard deviation     1.921 1.778 1.465 1.395 1.1653 0.9996 0.9230 0.7256
## Proportion of Variance 0.231 0.198 0.134 0.122 0.0849 0.0624 0.0532 0.0329
## Cumulative Proportion  0.231 0.428 0.562 0.684 0.7687 0.8311 0.8844 0.9173
##                           PC9   PC10   PC11    PC12
## Standard deviation     0.6710 0.6536 0.4981 0.35857
## Proportion of Variance 0.0281 0.0267 0.0155 0.00804
## Cumulative Proportion  0.9454 0.9721 0.9876 0.99567
```

ii. How many PCs should be used to obtain a cumulative variance of at least 90%?

```
pca_r = prcomp(cw2_6s[,1:16], rank=8, scale=TRUE)
summary(pca_r)
```

```
## Importance of first k=8 (out of 16) components:
##                           PC1   PC2   PC3   PC4    PC5    PC6    PC7    PC8
## Standard deviation     1.921 1.778 1.465 1.395 1.1653 0.9996 0.9230 0.7256
## Proportion of Variance 0.231 0.198 0.134 0.122 0.0849 0.0624 0.0532 0.0329
## Cumulative Proportion  0.231 0.428 0.562 0.684 0.7687 0.8311 0.8844 0.9173
```

2 CLUSTERING [R ONLY, 30 MARKS]

Using only R, explore the use of clustering techniques to find natural groupings in the data, without using the class variable – i.e. use only the appropriate input attributes to perform the clustering. Once the data is clustered, you may use the class variable to evaluate or interpret the results (how do the new clusters compare to the original classes?).

1. Choose an appropriate dataset and use HCA, k-means, and PAM as clustering algorithms to create groupings of three clusters and write the results. Which dataset have you used? Use a combination of internal and external metrics to evaluate which algorithm produces better results. Describe the metrics and how you calculated them [10].

```
cw2_6t = cw2_6[,3:19]
res = data.frame(cw2_6t$class)
colnames(res)[1] = "class"
cw2_6c = cw2_6t[,1:16]
```

1) HCA

```
v.hca = hclust(dist(cw2_6t[,1:16]))
res$hca = cutree(v.hca, 3)
table(cw2_6t$class, res$hca)
```

```
##
##              1    2    3
##   GALAXY 4873   95   29
##   QSO     693   92   65
##   STAR   2035 1742  376
```

```
intra.inter = cls.scatt.data(cw2_6c, res$hca, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.35
```

2) K-means

```
v.kms = kmeans(cw2_6t[,1:16], 3, iter.max= 100)
res$kms = v.kms$cluster
table(cw2_6t$class, v.kms$cluster)
```

```
##
##              1    2    3
##   GALAXY   93   42 4862
##   QSO      85   73  692
##   STAR   2703  400 1050
```

```
intra.inter = cls.scatt.data(cw2_6c, res$kms, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.26
```

3) PAM

```
v.pam = pam(cw2_6t[,1:16], 3)
res$pam = v.pam$clustering
table(cw2_6t$class, v.pam$clustering)
```

```
##
##              1    2    3
##   GALAXY 4862   93   42
##   QSO     692   85   73
##   STAR   1046 2707  400
```

```
intra.inter = cls.scatt.data(cw2_6c, res$pam, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.27
```

```
summ1 = sapply(res[,2:4], FUN= function(x){cluster.stats(dist(cw2_6c), clustering = x, silhouette = TRU
```

2.Using the dataset from the previous task, optimise each clustering method according to two parameters or more. Which parameters did you choose? Define them. Using the same metrics as in the previous exercise, which parameters produced the best results for each clustering algorithm? Provide the reasoning for the techniques you used to find the optimal parameters [10].

1) HCA

```
res2 = data.frame(cw2_6t$class)
colnames(res2)[1] = "class"

v.hca2 = hclust(dist(cw2_6t[,1:16], method="manhattan"))
res2$hca = cutree(v.hca2, 3)
table(cw2_6t$class, res2$hca)
```

```
##
##              1    2    3
##   GALAXY 4873   95   29
##   QSO     693   92   65
##   STAR   2035 1742  376
```

```
v.hca2 = hclust(dist(cw2_6t[,1:16]), method="average")
res2$hca2 = cutree(v.hca2, 3)
table(cw2_6t$class, res2$hca2)
```

```
##
##              1    2    3
##   GALAXY 4862   96   39
##   QSO     692   87   71
##   STAR   1046 2710  397
```

```
v.hca2 = hclust(dist(cw2_6t[,1:16], method="manhattan"), method="average")
res2$hca3 = cutree(v.hca2, 3)
table(cw2_6t$class, res2$hca3)
```

```
##
##              1    2    3
##   GALAXY 4862   96   39
##   QSO     692   87   71
##   STAR   1046 2710  397
```

```
summ2 = sapply(res2[,2:4], FUN= function(x){cluster.stats(dist(cw2_6c, method="manhattan"), clustering
```

```
intra.inter = cls.scatt.data(cw2_6c, res2$hca, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.35
```

```
intra.inter = cls.scatt.data(cw2_6c, res2$hca2, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.26
```

```
intra.inter = cls.scatt.data(cw2_6c, res2$hca3, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.26
```

2) k-means

```
res2 = data.frame(cw2_6t$class)
colnames(res2)[1] = "class"

v.kms2 = kmeans(cw2_6t[,1:16], 3, iter.max= 100, nstart = 20)
res2$kms = v.kms2$cluster
table(cw2_6t$class, v.kms2$cluster)
```

```
##
##              1    2    3
##   GALAXY   93   42 4862
##   QSO      85   73  692
##   STAR   2703  400 1050
```

```
v.kms2 = kmeans(cw2_6t[,1:16], 3, iter.max= 10000, nstart = 1)
res2$kms2 = v.kms2$cluster
table(cw2_6t$class, v.kms2$cluster)
```

```
##
##              1    2    3
##   GALAXY 4862   42   93
##   QSO     692   73   85
##   STAR   1050  400 2703
```

```
v.kms2 = kmeans(cw2_6t[,1:16], 3, iter.max= 10000, nstart = 20)
res2$kms3 = v.kms2$cluster
table(cw2_6t$class, v.kms2$cluster)
```

```
##
##              1    2    3
##   GALAXY   42 4862   93
##   QSO      73  692   85
##   STAR    400 1050 2703
```

```
summ3 = sapply(res2[,2:4], FUN= function(x){cluster.stats(dist(cw2_6c, method="manhattan"), clustering
```

```
intra.inter = cls.scatt.data(cw2_6c, res2$kms, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.26
```

```
intra.inter = cls.scatt.data(cw2_6c, res2$kms2, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.26
```

```
intra.inter = cls.scatt.data(cw2_6c, res2$kms3, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.26
```

3) PAM

```
res2 = data.frame(cw2_6t$class)
colnames(res2)[1] = "class"

v.pam3 = pam(cw2_6t[,1:16], 3, metric = "manhattan", stand = FALSE)
res2$pam = v.pam3$clustering
table(cw2_6t$class, v.pam3$clustering)
```

```
##
##              1    2    3
##   GALAXY 4862   93   42
##   QSO     692   85   73
##   STAR   1046 2707  400
```

```
v.pam3 = pam(cw2_6t[,1:16], 3, stand = TRUE)
res2$pam2 = v.pam3$clustering
table(cw2_6t$class, v.pam3$clustering)
```

```
##
##              1    2    3
##   GALAXY 3390  110 1497
##   QSO     460  109  281
##   STAR    773 2760  620
```

```
v.pam3 = pam(cw2_6t[,1:16], 3, metric = "manhattan", stand = TRUE)
res2$pam3 = v.pam3$clustering
table(cw2_6t$class, v.pam3$clustering)
```

```
##
##              1    2    3
##   GALAXY 1708 3194   95
##   QSO     412  349   89
##   STAR    638  748 2767
```

```
summ4 = sapply(res2[,2:4], FUN= function(x){cluster.stats(dist(cw2_6c, method="manhattan"), clustering
```

```
intra.inter = cls.scatt.data(cw2_6c, res2$pam, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.27
```

```
intra.inter = cls.scatt.data(cw2_6c, res2$pam2, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##     ave
## ave 1.3
```

```
intra.inter = cls.scatt.data(cw2_6c, res2$pam3, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##     ave
## ave 1.4
```

3.Choose one clustering algorithm of the above and a combination of internal and external metrics, then perform clustering on the following alternative datasets which you have produced in Part 1 [10]:

i. The transformed dataset featuring all Principal Components

```
res3 = data.frame(cw2_6t$class)
colnames(res3)[1] = "class"
```

```
t.kms = kmeans(pca_t$x, 3, iter.max= 100)
res3$t.kms = t.kms$cluster
table(cw2_6t$class, t.kms$cluster)
```

```
##
##              1    2    3
##   GALAXY 1325  107 3565
##   QSO      13  158  679
##   STAR   1485 1848  820
```

ii. The reduced dataset featuring 12 Principal Components.

```
r.kms2 = kmeans(pca_12$x, 3, iter.max= 100)
res3$r.kms2 = r.kms2$cluster
table(cw2_6t$class, t.kms$cluster)
```

```
##
##              1    2    3
##   GALAXY 1325  107 3565
##   QSO      13  158  679
##   STAR   1485 1848  820
```

iii. The dataset after deletion of instances and attributes.

```
d.kms = kmeans(cw2_6t[,1:16], 3, iter.max= 100)
res3$d.kms = d.kms$cluster
table(cw2_6t$class, d.kms$cluster)
```

```
##
##              1    2    3
##   GALAXY   93   42 4862
##   QSO      85   73  692
##   STAR   2703  400 1050
```

```
summ5 = sapply(res3[,2:4], FUN= function(x){cluster.stats(dist(cw2_6c, method="manhattan"), clustering
```

```
intra.inter = cls.scatt.data(cw2_6c, res3$t.kms, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##     ave
## ave 1.2
```

```
intra.inter = cls.scatt.data(cw2_6c, res3$r.kms2, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##     ave
## ave 1.2
```

```
intra.inter = cls.scatt.data(cw2_6c, res3$d.kms, dist="manhattan")
clv.Davies.Bouldin(intra.inter, intracls = "average", intercls = "average")
```

```
##      ave
## ave 0.26
```

iv. All three versions of the mean-centred data

```
cw2 = read.csv("cw_data.csv")
cw2_6 = cw2[,c(1,3:22)]
cw2_6$na_count = apply(cw2_6, 1, function(x) sum(is.na(x)))
cw2_6 = cw2_6[cw2_6$na_count < 5,]
cw2_6 = unique(cw2_6)
cw2_6 = cw2_6[, c(1,2,4,5,7:21)]

sub1 = subset(cw2_6, cw2_6$class == "GALAXY")
sub2 = subset(cw2_6, cw2_6$class == "STAR")
sub3 = subset(cw2_6, cw2_6$class == "QSO")
k = ncol(cw2_6) - 1

for (i in 1:k) {
  cw2_6[is.na(cw2_6[,i]) & cw2_6$class == "GALAXY", i] = mean(sub1[,i], na.rm = TRUE)
  cw2_6[is.na(cw2_6[,i]) & cw2_6$class == "STAR", i] = mean(sub2[,i], na.rm = TRUE)
  cw2_6[is.na(cw2_6[,i]) & cw2_6$class == "QSO", i] = mean(sub3[,i], na.rm = TRUE)
}

cw2_6s = data.frame(scale(cw2_6[,1:18]),cw2_6[,19])
cw2_6s = cw2_6s[,3:19]
colnames(cw2_6s)[17] ="class"

pca_t = prcomp(cw2_6s[,1:16], scale=TRUE)
pca_12 = prcomp(cw2_6s[,1:16], rank=12, scale=TRUE)

cw2_6t = cw2_6[,3:19]
res4 = data.frame(cw2_6t$class)
colnames(res4)[1] = "class"
cw2_6c = cw2_6t[,1:16]
```

```
t.kms = kmeans(pca_t$x, 3, iter.max= 100)
res4$t.kms = t.kms$cluster
table(cw2_6t$class, t.kms$cluster)
```

```
##
##              1    2    3
##   GALAXY 1422 3472  103
##   QSO     239  462  149
##   STAR    629  830 2694
```

```
r.kms2 = kmeans(pca_12$x, 3, iter.max= 100)
res4$r.kms2 = r.kms2$cluster
table(cw2_6t$class, t.kms$cluster)
```

```
##
##              1    2    3
##   GALAXY 1422 3472  103
##   QSO     239  462  149
##   STAR    629  830 2694
```

```
d.kms = kmeans(cw2_6t[,1:16], 3, iter.max= 100)
res4$d.kms = d.kms$cluster
table(cw2_6t$class, d.kms$cluster)
```

```
##
##              1    2    3
##   GALAXY   93   42 4862
##   QSO      85   73  692
##   STAR   2703  400 1050
```

```
summ6 = sapply(res4[,2:4], FUN= function(x){cluster.stats(dist(cw2_6c, method="manhattan"), clustering
```