Sloan Digital Sky Survey DR14

Personal Details

- By: Shubham Sharan
- Student Number : 101084842
- Date Finished : Dec 26th 2020

Content

The data consists of 10,000 observations of space taken by the SDSS. Every observation is described by 17 feature columns and 1 class column which identifies it to be either a star, galaxy or quasar.

Libraries

Here are the many libraries we will include to conduct our various analyses.

```
suppressPackageStartupMessages({ #Comment out if needed
  library(pheatmap)
  library(ggplot2)
  library(gridExtra)
  library(GGally)
  library(dplyr)
  library(factoextra)
  library(caret)
  library(cluster)
  library(corrplot)
  library(stats)
  library(pheatmap)
  library(dbscan)
  require(foreign)
  require(nnet)
  require(reshape2)
  library(clValid)
  library(randomForest)
}) #Comment out if needed
```

Data Exploration

This is being done to get a sense of the data in hand and to make sure the data is somewhat ready to give us an understanding of the topic in hand.

```
sloan_data <- read.csv("Skyserver_SQL2_27_2018 6_51_39 PM.csv") # LOADING THE DATA AFTER DOWNLOADING IT FROM KAGGLE
head(sloan_data) # First 5 data points
```

	objid <dbl>	ra <dbl>	dec <dbl>	u <dbl>	g <dbl>	r <dbl>	i <dbl>	z <dbl>	... <int>
1	1.23765e+18	183.5313	0.08969303	19.47406	17.04240	15.94699	15.50342	15.22531	752
2	1.23765e+18	183.5984	0.13528503	18.66280	17.21449	16.67637	16.48922	16.39150	752
3	1.23765e+18	183.6802	0.12618509	19.38298	18.19169	17.47428	17.08732	16.80125	752
4	1.23765e+18	183.8705	0.04991069	17.76536	16.60272	16.16116	15.98233	15.90438	752
5	1.23765e+18	183.8833	0.10255675	17.55025	16.26342	16.43869	16.55492	16.61326	752
6	1.23765e+18	183.8472	0.17369416	19.43133	18.46779	18.16451	18.01475	18.04155	752

6 rows | 1-10 of 19 columns

```
str(sloan_data) #Structure of the dataset and more importantly the data types
```

```
## 'data.frame':    10000 obs. of  18 variables:
##  $ objid    : num  1.24e+18 1.24e+18 1.24e+18 1.24e+18 1.24e+18 ...
##  $ ra       : num  184 184 184 184 184 ...
##  $ dec      : num  0.0897 0.1353 0.1262 0.0499 0.1026 ...
##  $ u        : num  19.5 18.7 19.4 17.8 17.6 ...
##  $ g        : num  17 17.2 18.2 16.6 16.3 ...
##  $ r        : num  15.9 16.7 17.5 16.2 16.4 ...
##  $ i        : num  15.5 16.5 17.1 16 16.6 ...
##  $ z        : num  15.2 16.4 16.8 15.9 16.6 ...
##  $ run      : int  752 752 752 752 752 752 752 752 752 752 ...
##  $ rerun    : int  301 301 301 301 301 301 301 301 301 301 ...
##  $ camcol   : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ field    : int  267 267 268 269 269 269 269 269 270 270 ...
##  $ specobjid: num  3.72e+18 3.64e+17 3.23e+17 3.72e+18 3.72e+18 ...
##  $ class    : chr  "STAR" "STAR" "GALAXY" "STAR" ...
##  $ redshift : num  -8.96e-06 -5.49e-05 1.23e-01 -1.11e-04 5.90e-04 ...
##  $ plate    : int  3306 323 287 3306 3306 324 287 3306 323 288 ...
##  $ mjd      : int  54922 51615 52023 54922 54922 51666 52023 54922 51615 52000 ...
##  $ fiberid  : int  491 541 513 510 512 594 559 515 595 400 ...
```

```
dim(sloan_data) # Get a sense of the dimensions we are working with
```

```
## [1] 10000    18
```

```
table(is.na(sloan_data)) # We know we have no missing values in any of the columns if FA
LSE
```

```
##
##  FALSE
## 180000
```

```
unique(sloan_data$class) # Predictor Variable and we will make this into a factor later
```

```
## [1] "STAR"   "GALAXY" "QSO"
```

We have our class being our 3 categories and note how we will be conducting on a non-binary classification for this dataset. We have 10000 observations with no missing values and 17 features (we will omit some during our preliminary analysis)

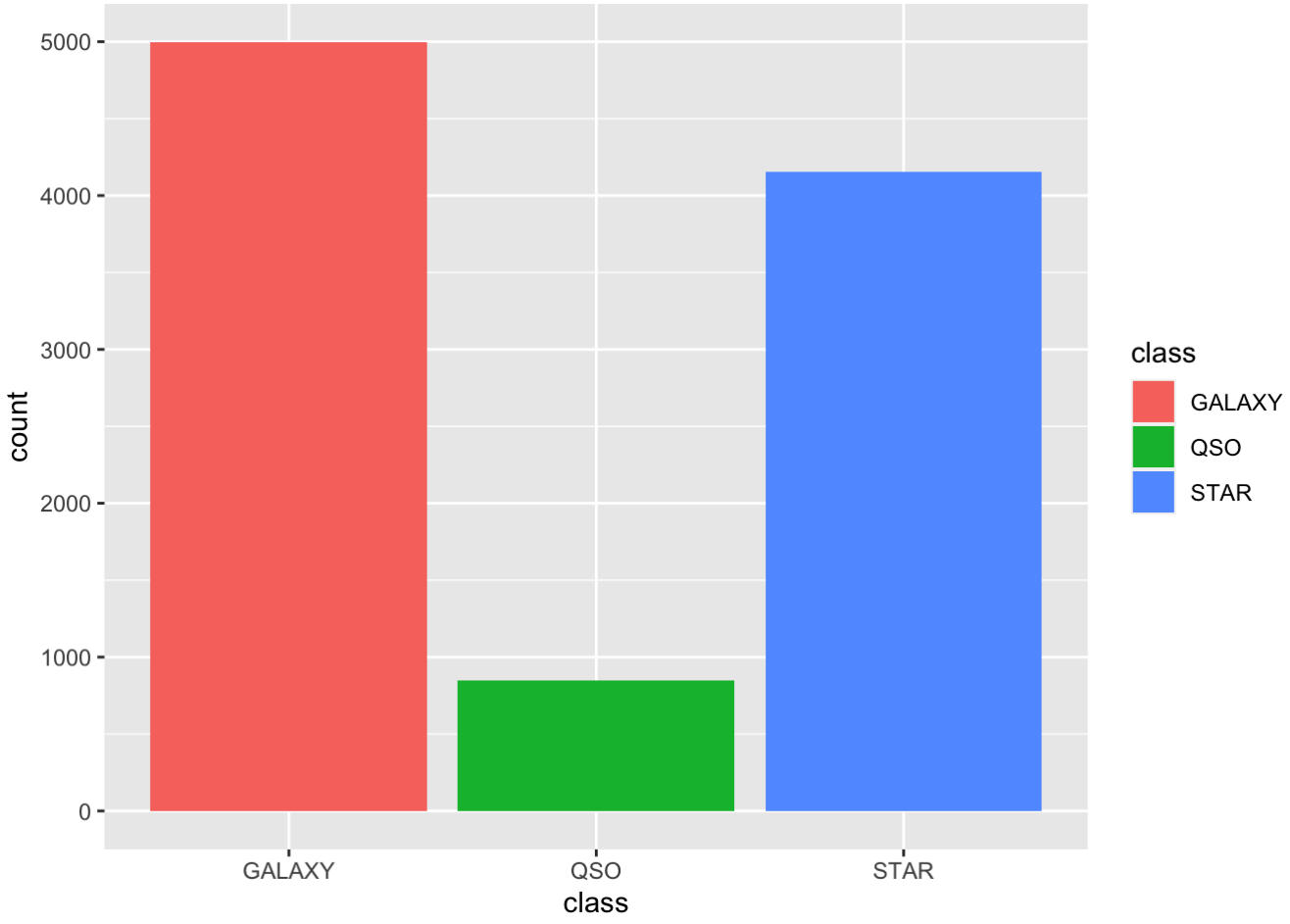
Preliminary Analysis

The focus of the preliminary analysis is to ensure that all the features utilized are a good representation to allow us to distinguish the astronomical bodies, with respect to their classes or any features that is able to identify any trends or features that are unique to Quasars or Galaxies or Stars

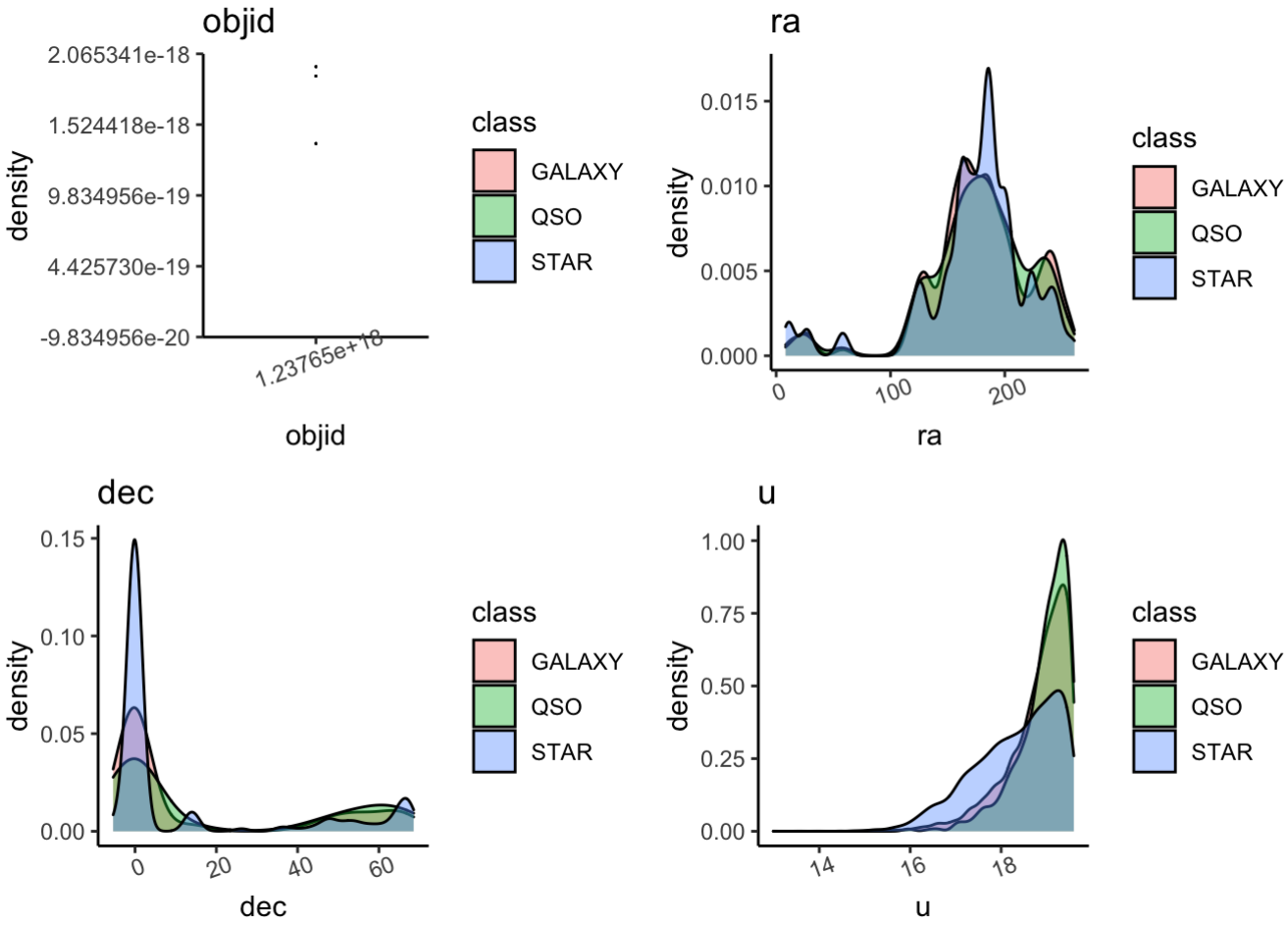
Data Visualization

Firstly we will start with getting an understanding how much of the data is provided for the 3 classes we will focus our clustering and classification on. Followed by using my same custom function for density plots with respect to our class as done in Assignment3, all the features are broken into 4x4 graphs for improved visibility.

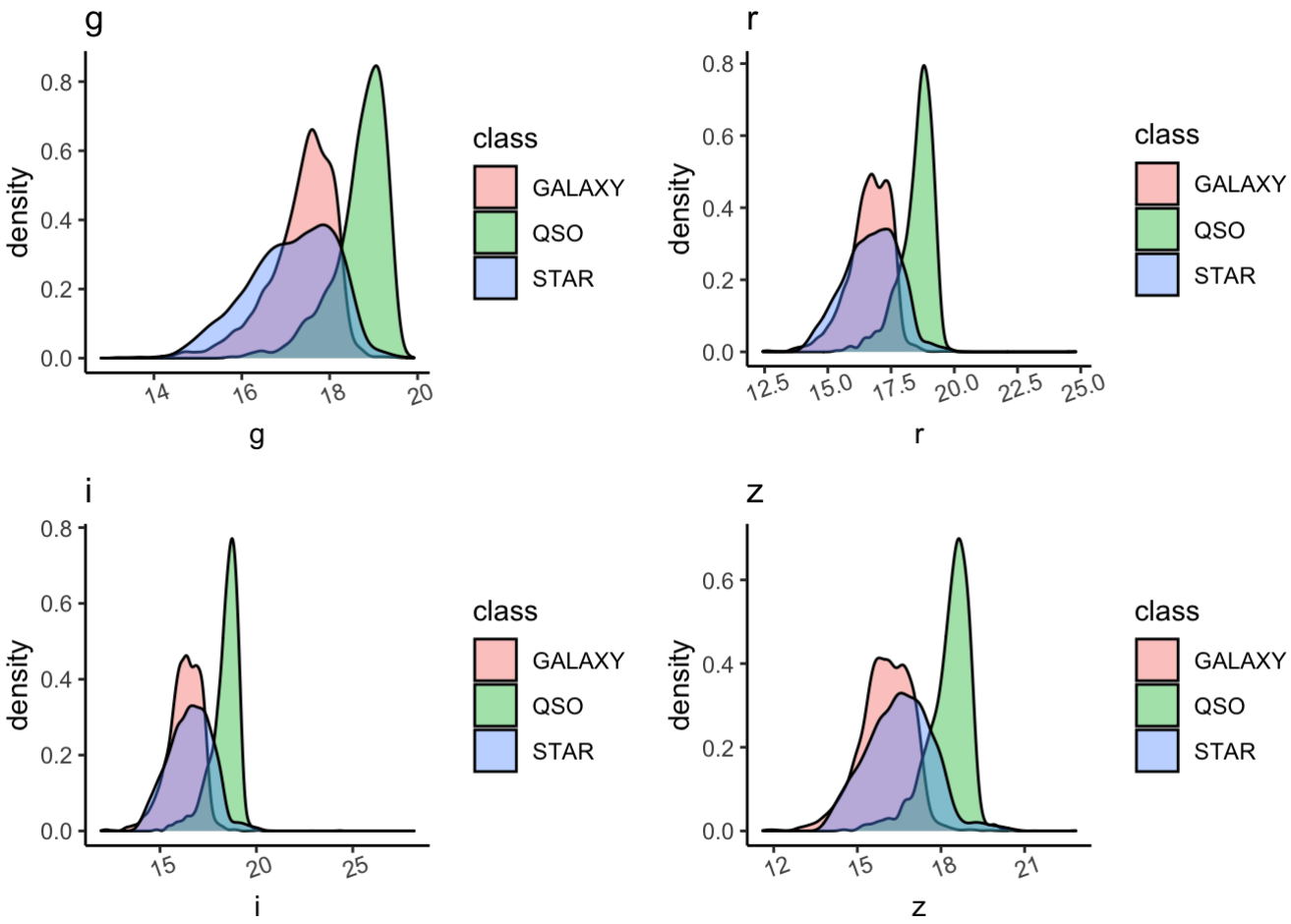
```
ggplot(sloan_data, aes(class, fill= class)) + geom_bar() # In the form of a bar chart
```



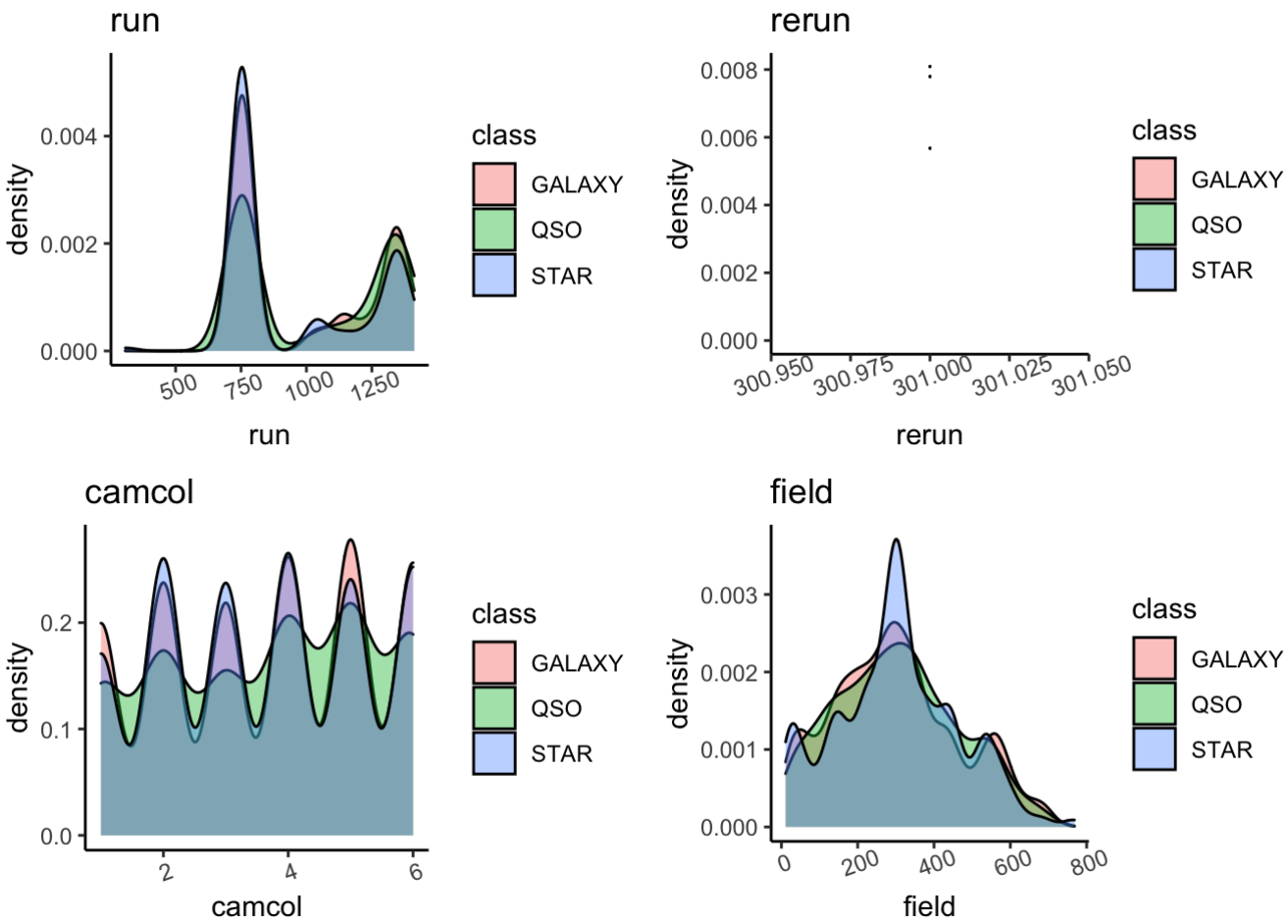
```
plot_data_column = function (data, column) {
     ggplot(data[2:18],aes(x=data[,column],fill=class))+geom_density(alpha=0.4)+ggtitle
(column)+theme_classic()+theme(axis.text.x = element_text(angle = 20))+xlab(label = colu
mn)
}

sloan_data <- sloan_data %>% relocate(class, .after = last_col())

myplots <- lapply(colnames(sloan_data), plot_data_column, data = sloan_data)
do.call("grid.arrange", c(myplots[1:4], ncol=2))
```



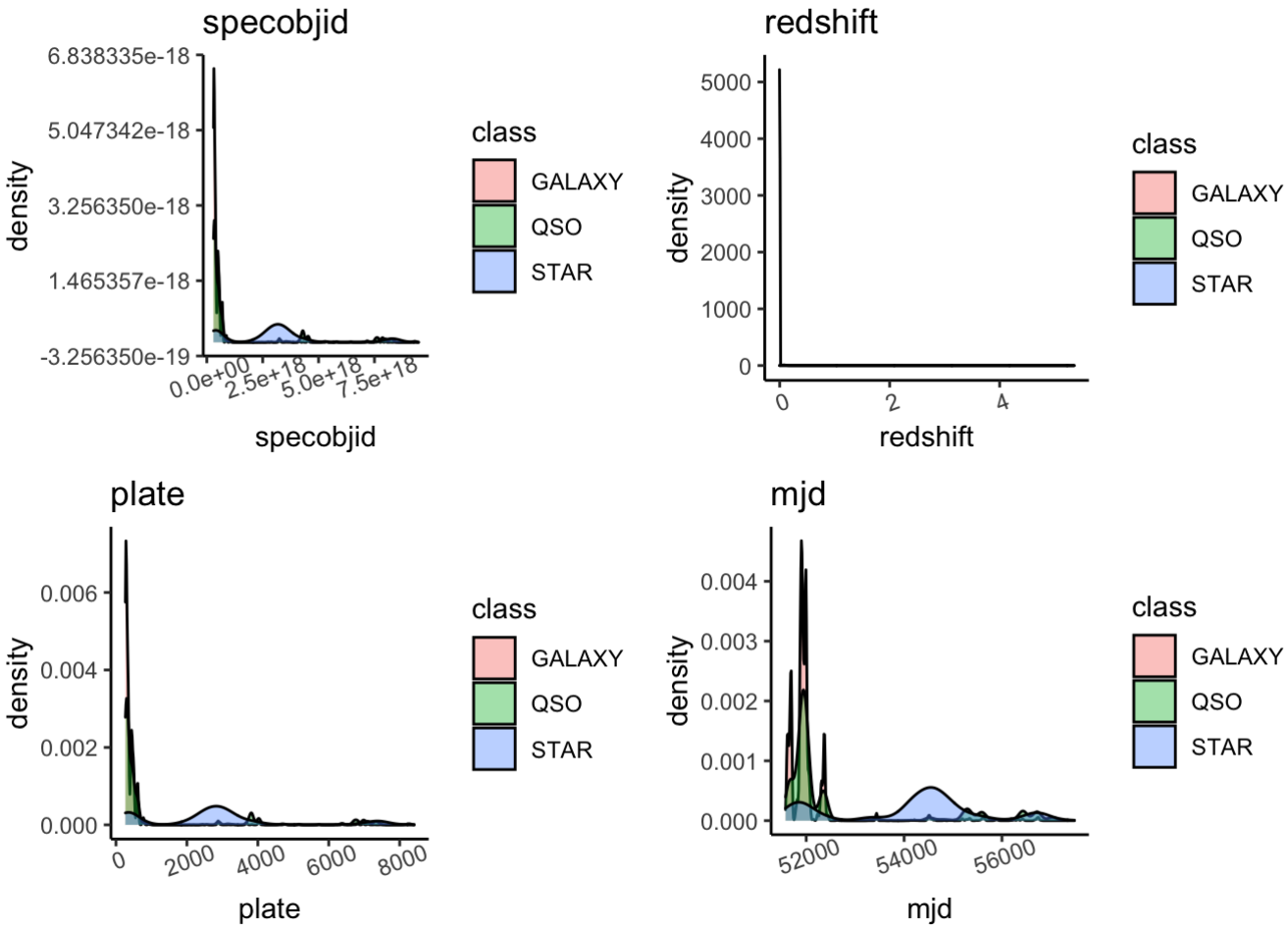

```
do.call("grid.arrange", c(myplots[5:8], ncol=2))
```



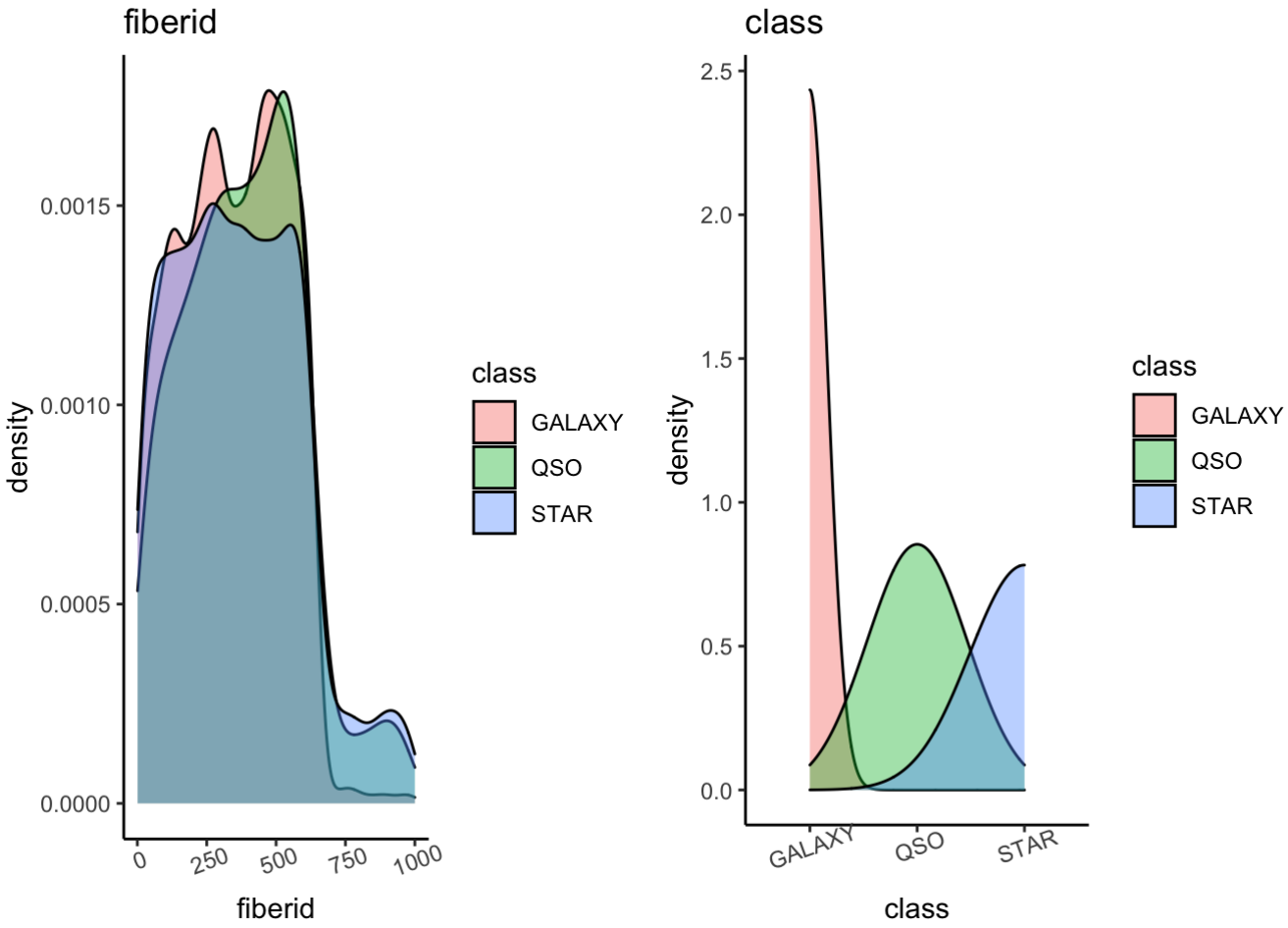

```
do.call("grid.arrange", c(myplots[9:12], ncol=2))
```




```
do.call("grid.arrange", c(myplots[13:16], ncol=2))
```




```
do.call("grid.arrange", c(myplots[17:18], ncol=2))
```

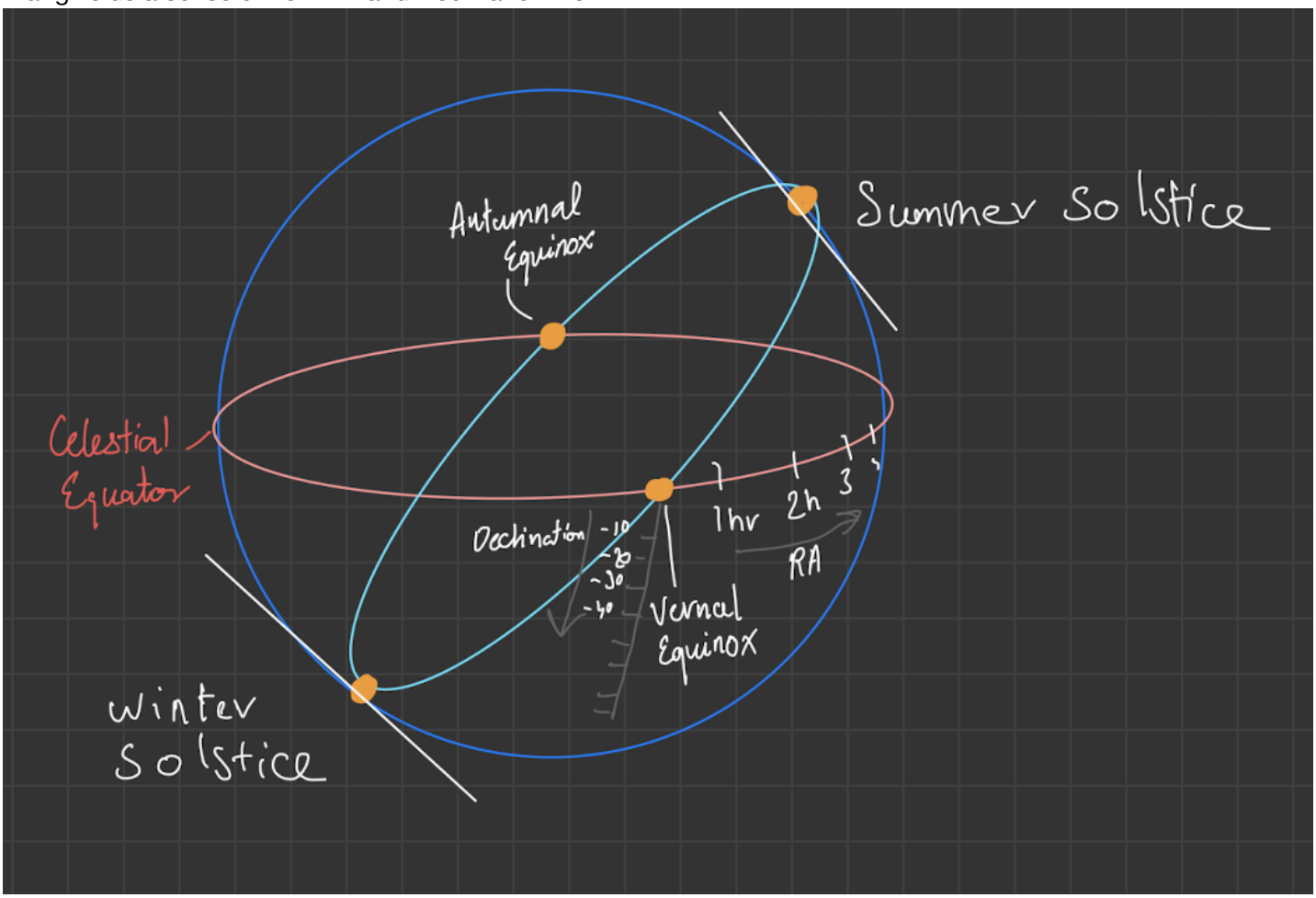


We see some features with 3 distinct peaks (e.g. photometry data) which make them ideal for our classification taks and some where the scales with respect to classes may vary to the point where they become ideal canidates (e.g. plate). Let's start of by describing the many features.

Location oriented descriptions below:

- objid = Object Identifier (Doesn't add any value in the classification) each catalog object has a unique combination of run-camcol-field-id-rerun; this combination is hashed into a single 64-bit integer called ObjID.
- ra = J2000 Right Ascension (r-band) is the angular distance measured eastward along the celestial equator from the Sun at the March equinox to the hour circle of the point above the earth in question.
- dec = J2000 Declination (r-band) these astronomical coordinates specify the direction of a point on the celestial sphere (traditionally called in English the skies or the sky) in the equatorial coordinate system.

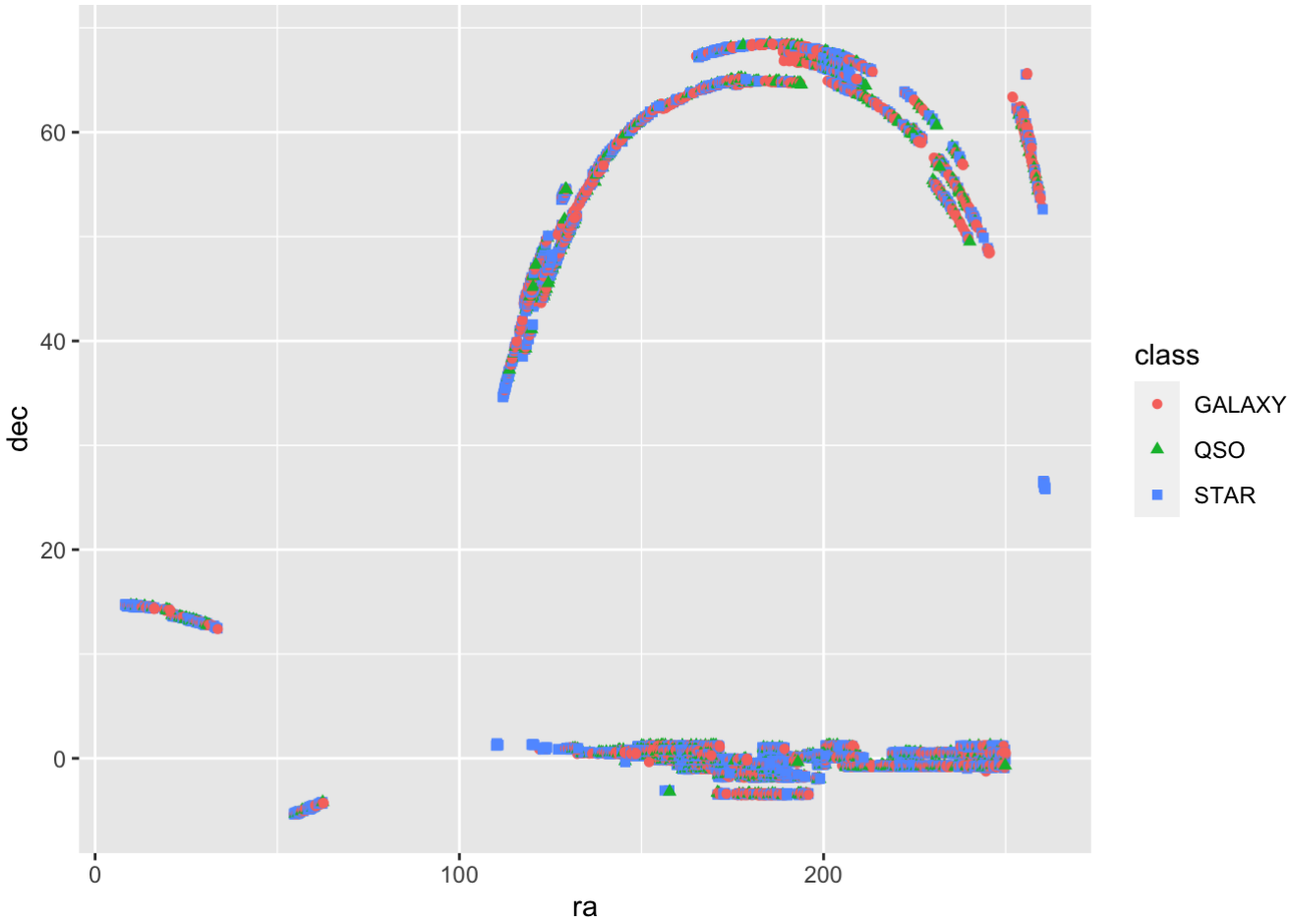
A bit about RA (Right Ascension) and Declination from my astronomy class PHYS1902 and some other features that give us a sense of how RA and Declination work.



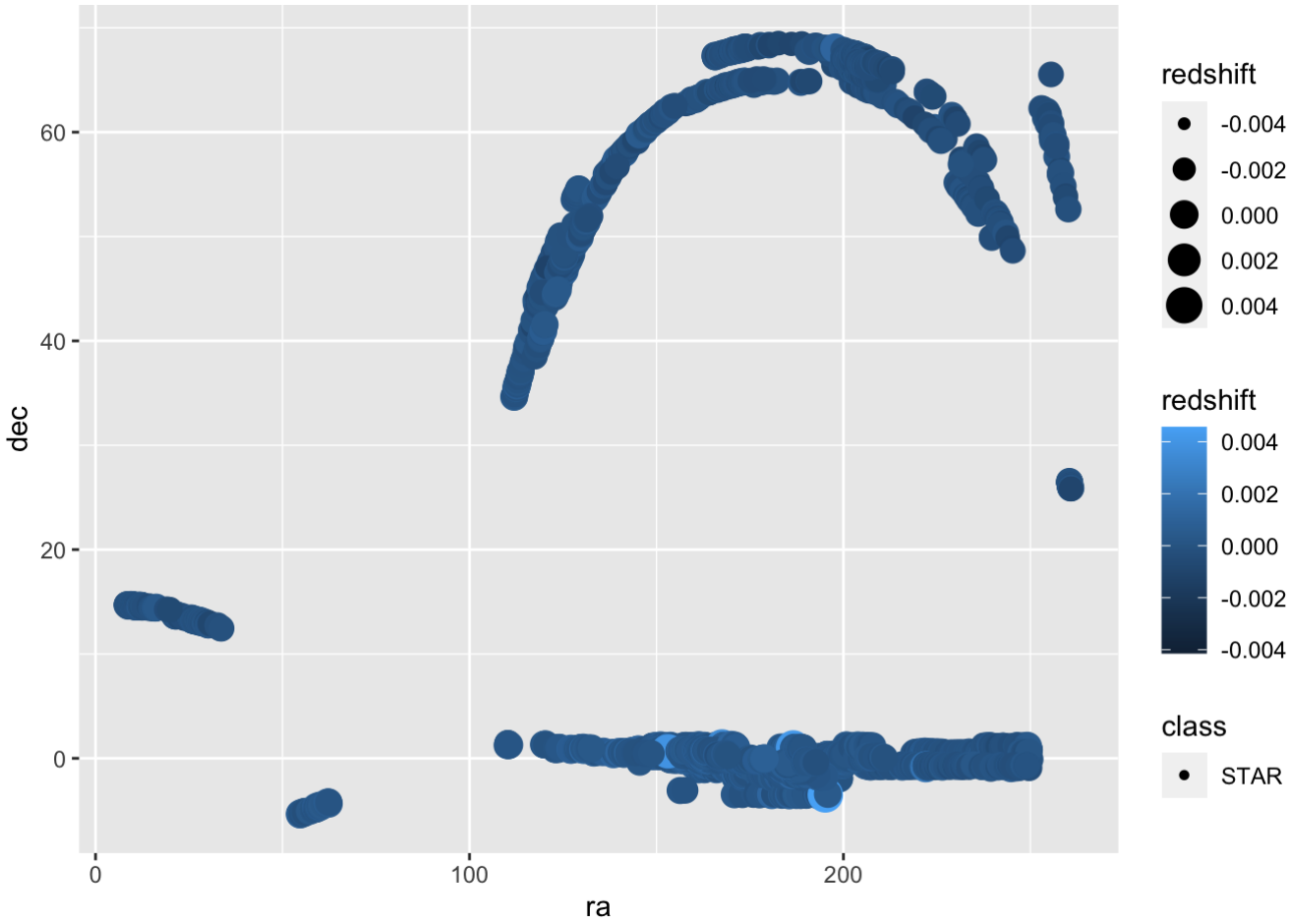
Features of the dataset some which pertain to the motion of the object and the some others to spectroscopic analysis:

- redshift = Final Redshift happens when light or other electromagnetic radiation from an object is increased in wavelength, or shifted to the red end of the spectrum. This is a great indicator as tell us if it moving towards or further from us and the brighter the object the easier it is for our telescopes to resolve. We will look further into this in the upcoming plots.
- plate = plate number where each spectroscopic exposure employs a large, thin, circular metal plate that positions optical fibers via holes drilled at the locations of the images in the telescope focal plane. These fibers then feed into the spectrographs. Each plate has a unique serial number, which is called plate in views.
- mjd = MJD of observation used to indicate the date that a given piece of SDSS data (image or spectrum) was taken. Days after November 17 1858.
- fiberid = fiber ID the SDSS spectrograph uses optical fibers to direct the light at the focal plane from individual objects to the slit head. Each object is assigned a corresponding fiberID. (We will be removing all unique forms of identifiers as these are not continous data points more so should be treated like a class feaatures)

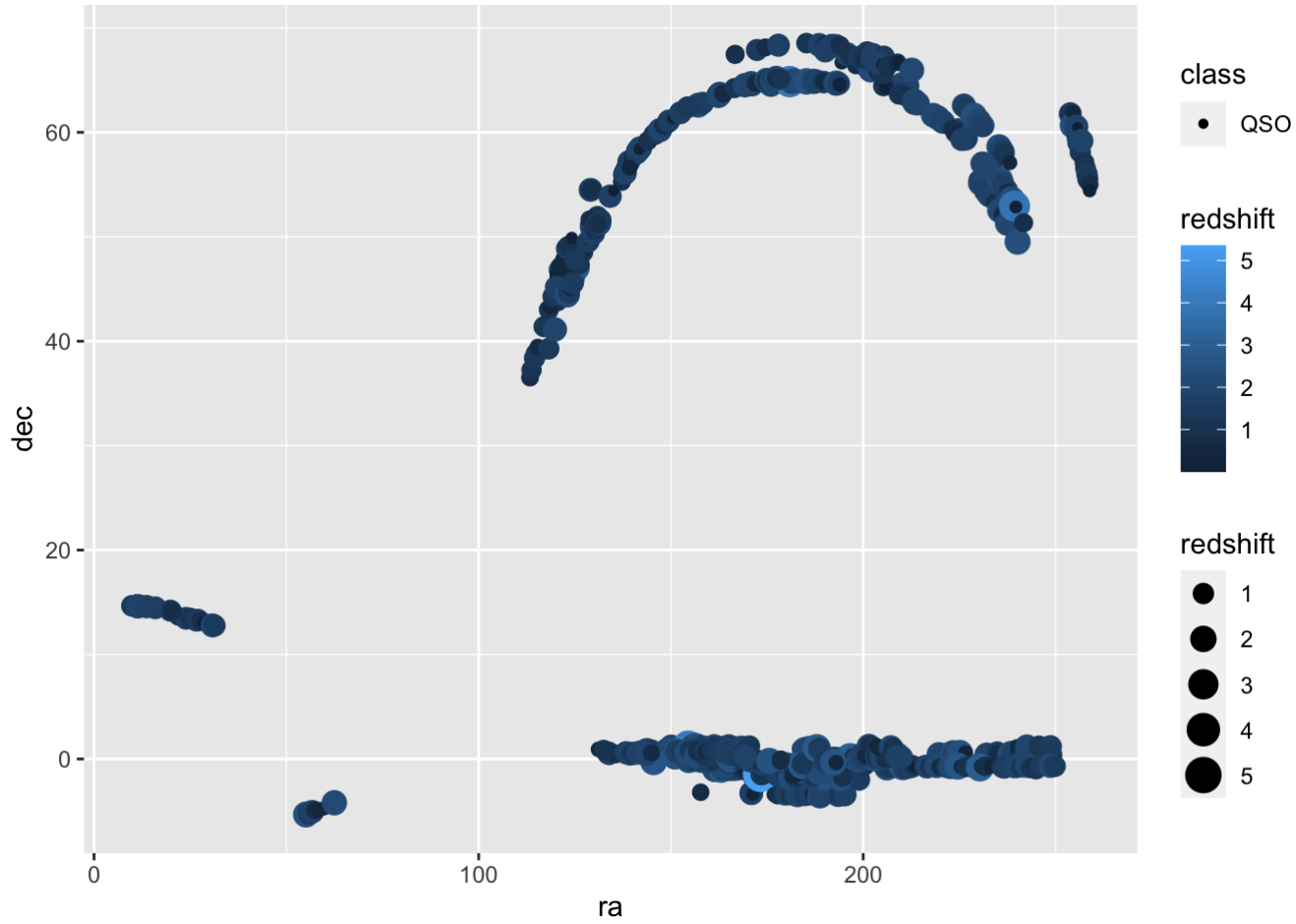
```
ggplot(sloan_data, aes(x=ra, y=dec, shape=class, color=class)) +
  geom_point()
```



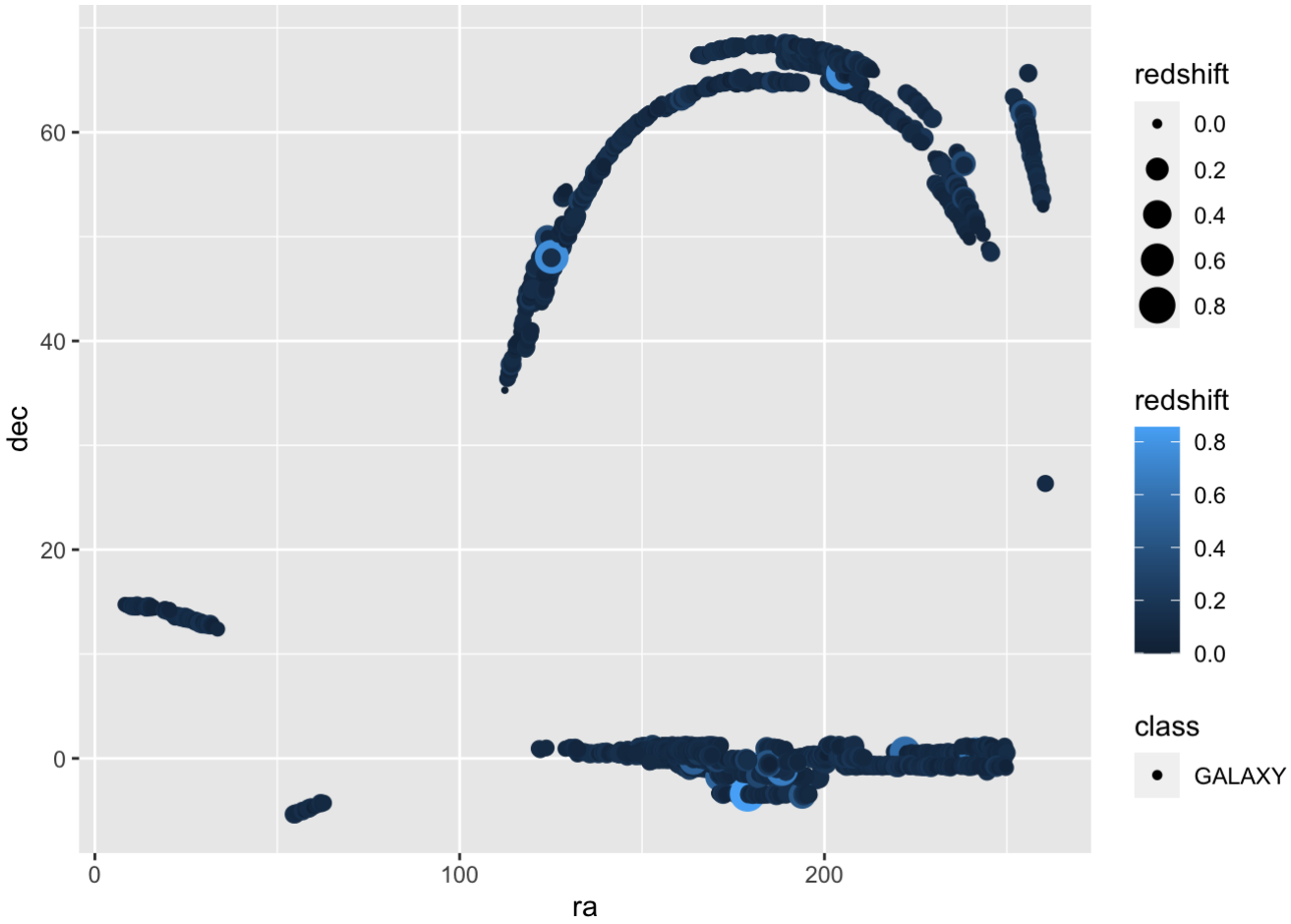

```
ggplot(sloan_data[sloan_data$class=="STAR",], aes(x=ra, y=dec, shape=class, color=redshi
ft, size = redshift)) +
  geom_point()
```



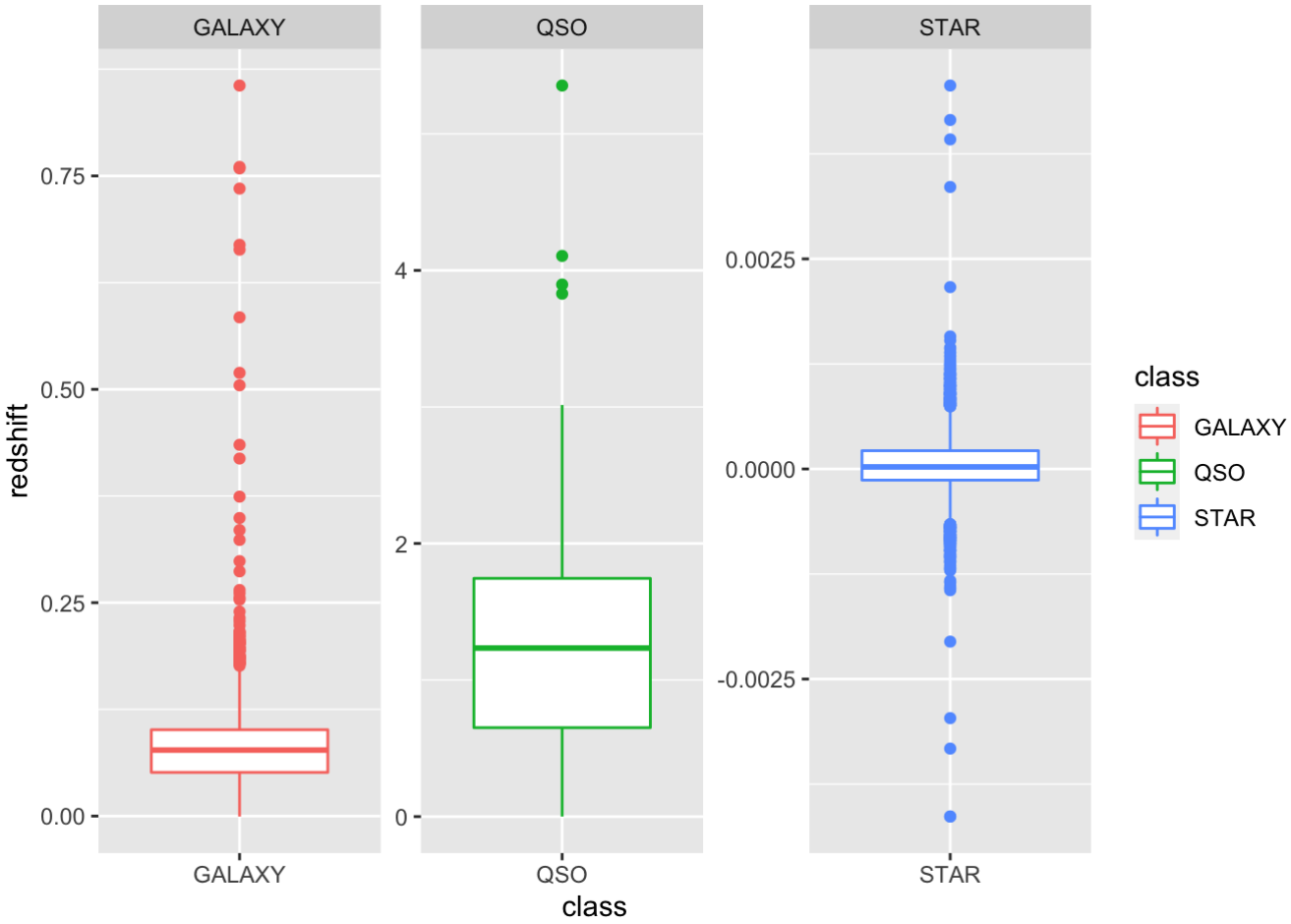

```
# NOTE HOW SOME OF THE STARS ARE ACTUALLY BLUE SHIFTED AND NOT RED SHIFTED
ggplot(sloan_data[sloan_data$class=="QSO",], aes(x=ra, y=dec, shape=class, color=redshif
t, size = redshift)) +
  geom_point()
```




```
ggplot(sloan_data[sloan_data$class=="GALAXY",], aes(x=ra, y=dec, shape=class, color=reds
hift, size = redshift)) +
  geom_point()
```




```
#NOTE HOW THE REDSHIFTS DIFFER BETWEEN THE CLASSES
ggplot(sloan_data, aes(x=class, y=redshift, shape=class, color=class)) +
  geom_boxplot() + facet_wrap(~class, scale="free")
```

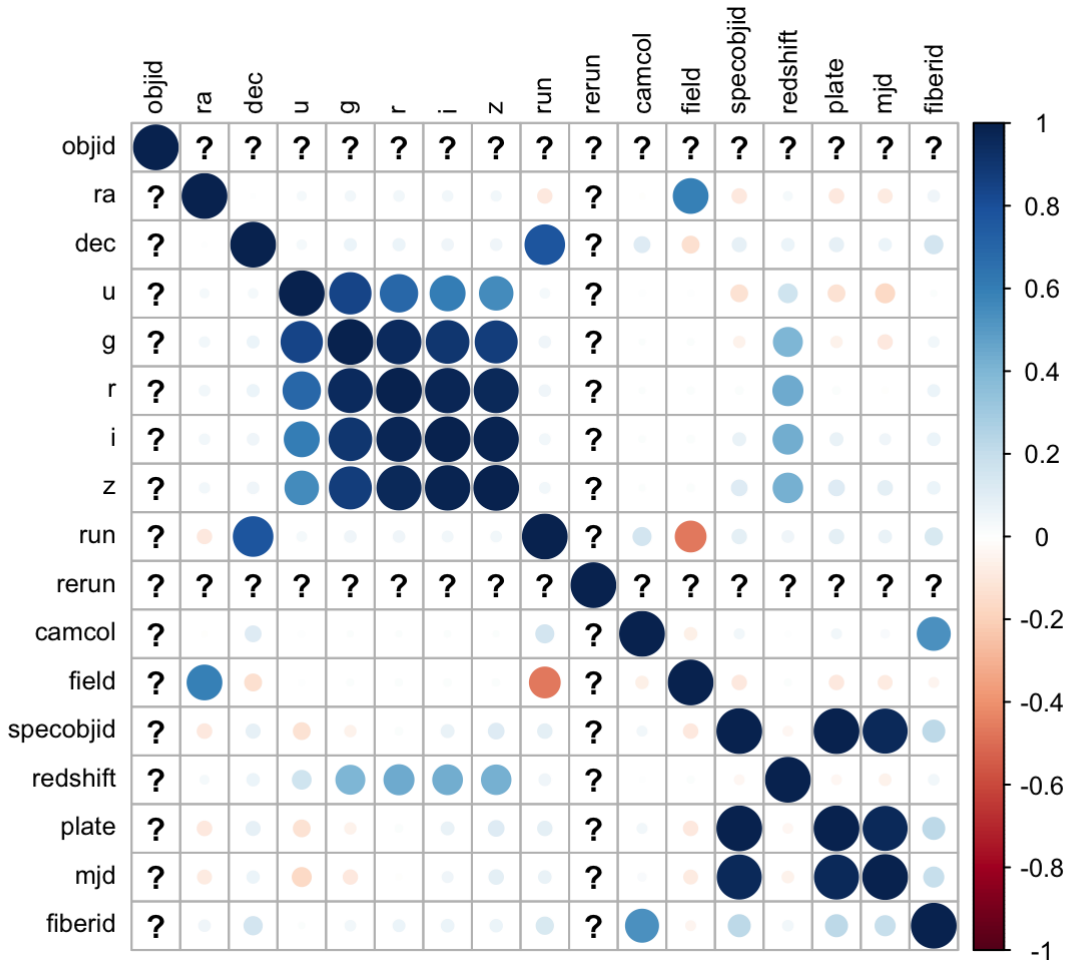


The scale of the many box plots drastically differ, from which we can conclude that this will be a good candidate when focusing on classification and identifying suitable clusters.

```
c <- cor(sloan_data[1:17],method="pearson")
```

```
## Warning in cor(sloan_data[1:17], method = "pearson"): the standard deviation is
## zero
```

```
corrplot(c, tl.cex=0.8,tl.col = "black")
```




```
unique(sloan_data$objid)
```

```
## [1] 1.23765e+18
```

```
unique(sloan_data$rerun)
```

```
## [1] 301
```

From this corrplot we see a lot of interactions between the five-band (u, g, r, i, z) CCD-based photometry and a few in the redshift and mjd. Even though some of the unique identifiers show correlation but is not relevant as if they were decided on the basis of the class then we shouldn't include it as it may tamper with our results (e.g. naming conventions for starts is different than quasars or galaxies). If it has nothing to do with the class then it has no value as it is more arbitrary, some unique identifiers are concatenation of multiple columns which still has no value. The question marks tell us that there is only one unique instance of that data point and has no value in the overall analysis. Let's look at some more features which really don't interact with each other that much.

Run, rerun, camcol and field are features which describe a field within an image taken by the SDSS. A field is basically a part of the entire image corresponding to 2048 by 1489 pixels. A field can be identified by:

- run = Run Number which identifies the specific scan
- rereun = Rerun Number, specifies how the image was processed.(Only has one value throughout the data).
- camcol = Camera column a number from 1 to 6, identifying the scanline within the run, and the field number

- field = Field number typically starts at 11 (after an initial ramp up time), and can be as large as 800 for particularly long runs.

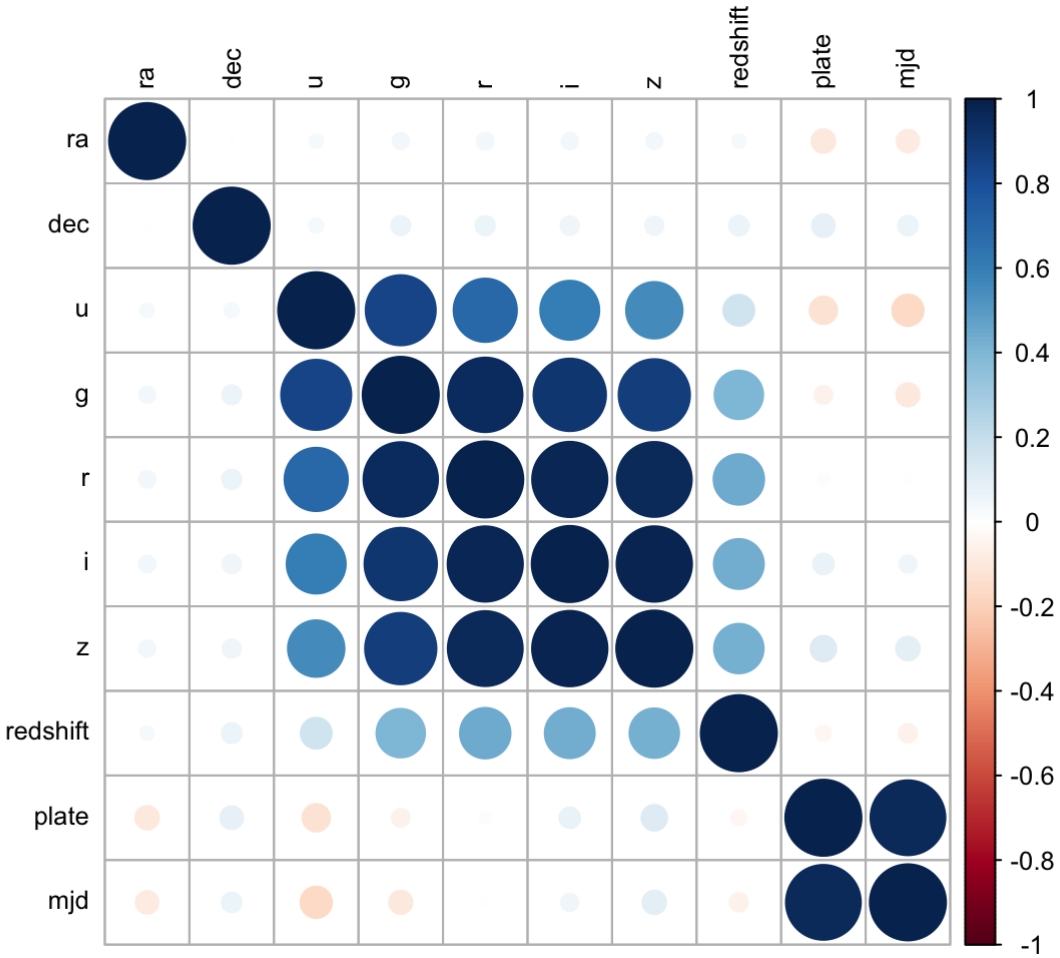
These features may not have much predictive power when it comes to identifying the class and was evident in the density plots and corrplots. It's also cause these focus more on the imaging method as compared to the attributes that contribute to a specific class.

specobjid = Object Identifier # This as well being a unique identifier adds any value to the overall analysis. class = object class (galaxy, star or quasar object)

```
# Remove all or any unique identifiers
uids <- c('objid','specobjid','fiberid')
sloan_data_clean <- sloan_data %>% select(-one_of(uids))

imgdescriptors <- c('run','rerun','camcol','field')
sloan_data_clean <- sloan_data_clean %>% select(-one_of(imgdescriptors ))
dim(sloan_data_clean)
```

```
## [1] 10000    11
```

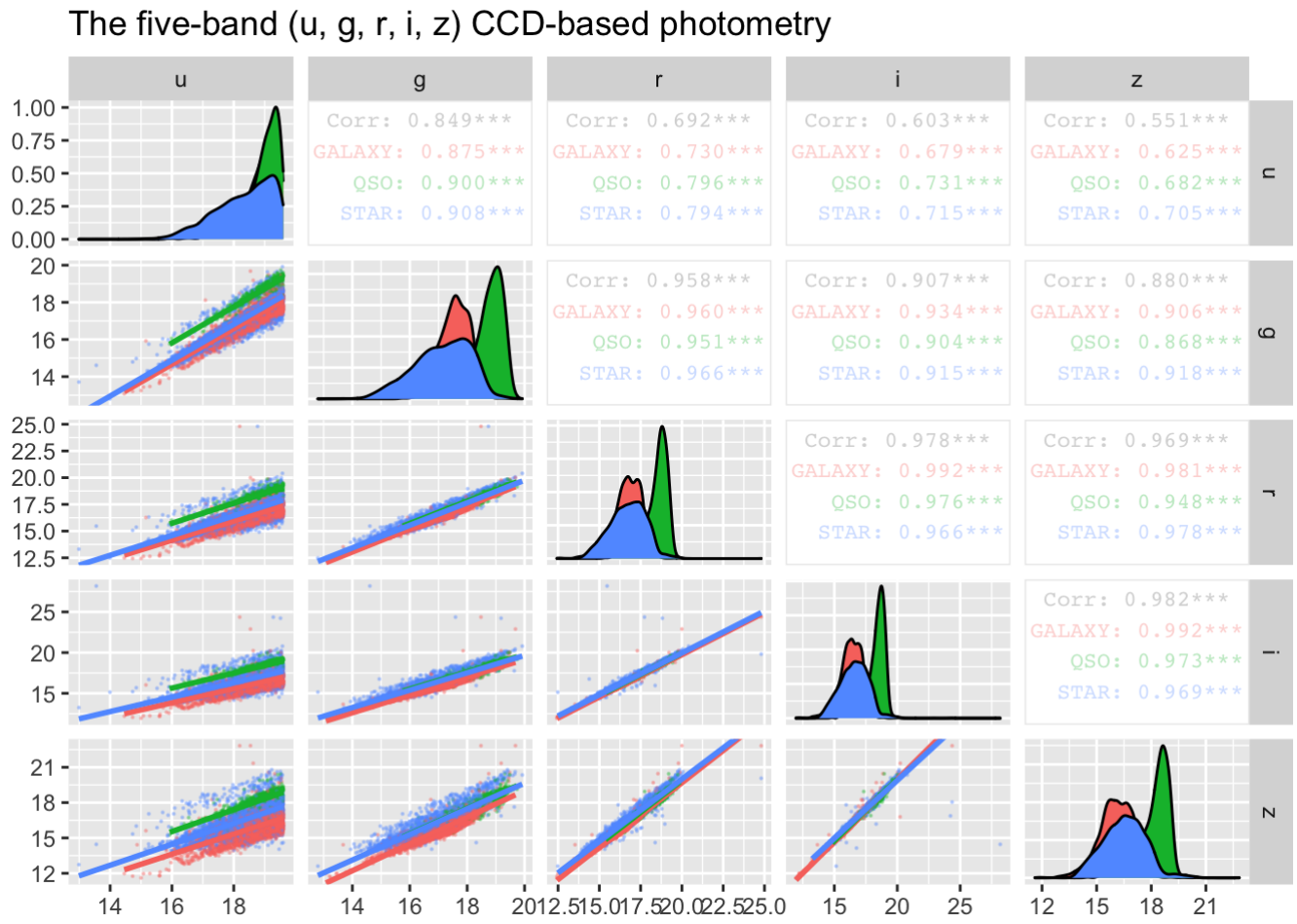
```
c <- cor(sloan_data_clean[1:10],method="pearson")
corrplot(c, tl.cex=0.8,tl.col = "black")
```



In this preliminary plotting we realize that many of the features have no properties which help us distinguish one from the other. The five-band (u, g, r, i, z) CCD-based photometry seems to be a good indicator as we see three individual peaks.

```
uids <- c('u', 'g', 'r', 'i', 'z','class')
fiveband <- sloan_data_clean %>% select(one_of(uids))

ggpairs(fiveband[1:5], title="The five-band (u, g, r, i, z) CCD-based photometry",messag
e=FALSE,progress=FALSE,  mapping=ggplot2::aes(colour = as.factor(sloan_data_clean$clas
s)), lower = list(continuous = wrap("smooth", alpha = 0.3, size=0.1)), upper = list( con
tinuous = wrap("cor", size=3, alpha=0.3)))
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'message' are being
## ignored. If these are meant to be aesthetics, submit them using the 'mapping'
## variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```

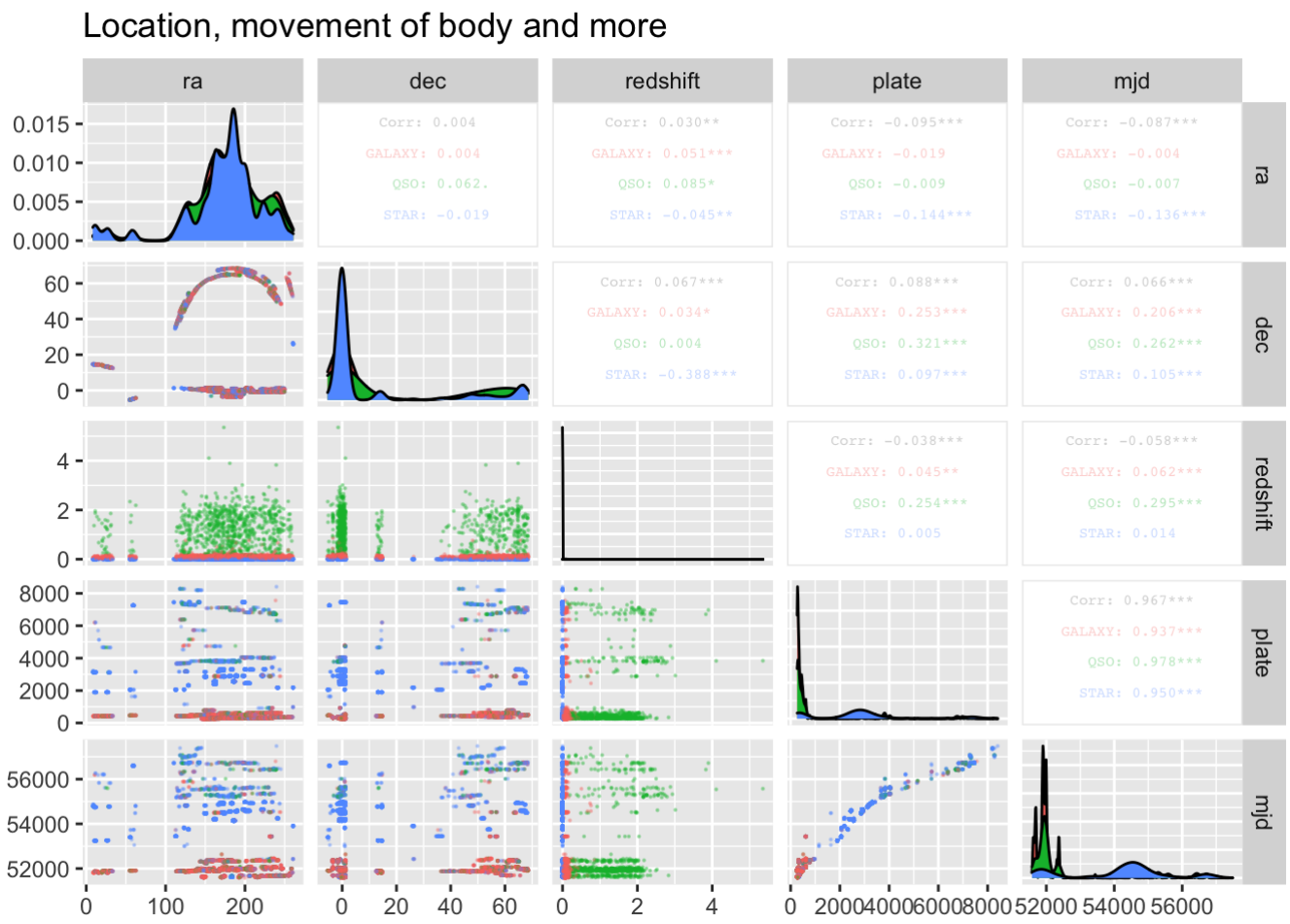


Here we see 3 clear distinct peaks and linear nature between the interactions terms which signify that the variables are correlated with one another and this was also evident in the corrplot.

Topic of redshifts was shown how the scale for the 3 different classes varies to signify that it is a good class indicator. Some of the other features show interactions with one another therefore, we decide to keep these and see how it impacts our model.

```
extrafeats <- c('ra', 'dec', 'redshift', 'plate','mjd','class')
loc <- sloan_data_clean %>% select(one_of(extrafeats))

#PAIRPLOT : To see feature interactions between Location, movement of spectroscopic feat
ures and more
ggpairs(loc[1:5], title="Location, movement of body and more ",message=FALSE,progress=FA
LSE,  mapping=ggplot2::aes(colour = as.factor(sloan_data_clean$class)), lower = list(con
tinuous = wrap("points", alpha = 0.3, size=0.1)), upper = list( continuous = wrap("cor",
size=2, alpha=0.3)))
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'message' are being
## ignored. If these are meant to be aesthetics, submit them using the 'mapping'
## variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```



This is informative by showing some of the interaction between the motion and position of the planet, along with some other spectroscopic features. Redshift we see a clear distinction that is not as visible in the density plots before and now. mjd being a date is a positive correlation and we see more blue which pertains to stars in left more side which could be a sign of how are telescopes have been improving the larger the mjd the more recent the data was collected, as stars are much smaller then a entire quasar or galaxy and is much harder to resolve.

Principal Component Analysis

PCA Analysis on entire dataset, just to get a sense of how much variance is being explained by x dimensions?

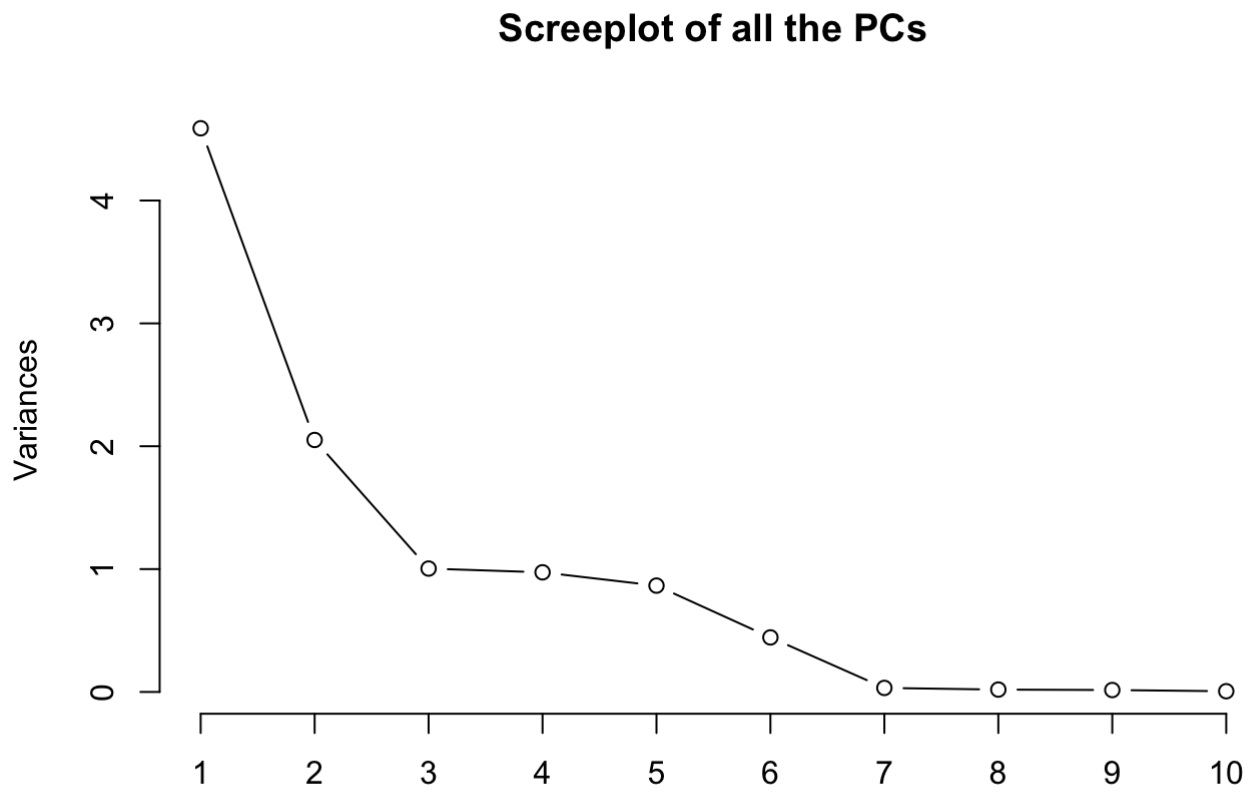
```
colnames(sloan_data_clean[1:10])
```

```
##  [1] "ra"       "dec"      "u"        "g"        "r"        "i"
##  [7] "z"        "redshift" "plate"    "mjd"
```

```
summary(space_pc0 <- prcomp(scale(sloan_data_clean[1:10]))) #SCALED
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.1420 1.4321 1.0021 0.98677 0.93035 0.66623 0.18111
## Proportion of Variance 0.4588 0.2051 0.1004 0.09737 0.08655 0.04439 0.00328
## Cumulative Proportion  0.4588 0.6639 0.7643 0.86170 0.94826 0.99265 0.99593
##                           PC8     PC9    PC10
## Standard deviation     0.13881 0.12513 0.07625
## Proportion of Variance 0.00193 0.00157 0.00058
## Cumulative Proportion  0.99785 0.99942 1.00000
```

```
screeplot(space_pc0, type = "line", main = "Screeplot of all the PCs") # Much of the var
iation is explained by the first 5 PC's
```



This PCA shows that all of the variance can be explained by the first 5 PC's after scaling. Since we don't want to reduce the dimensions just 5 we shall just reduce the dimensions of the highly correlated photometry data of 5 dimensions.

Here we will take photometry data of 5 dimensions to an appropriate number of principal components to see how much of the variation can be explained by fewer dimensions.

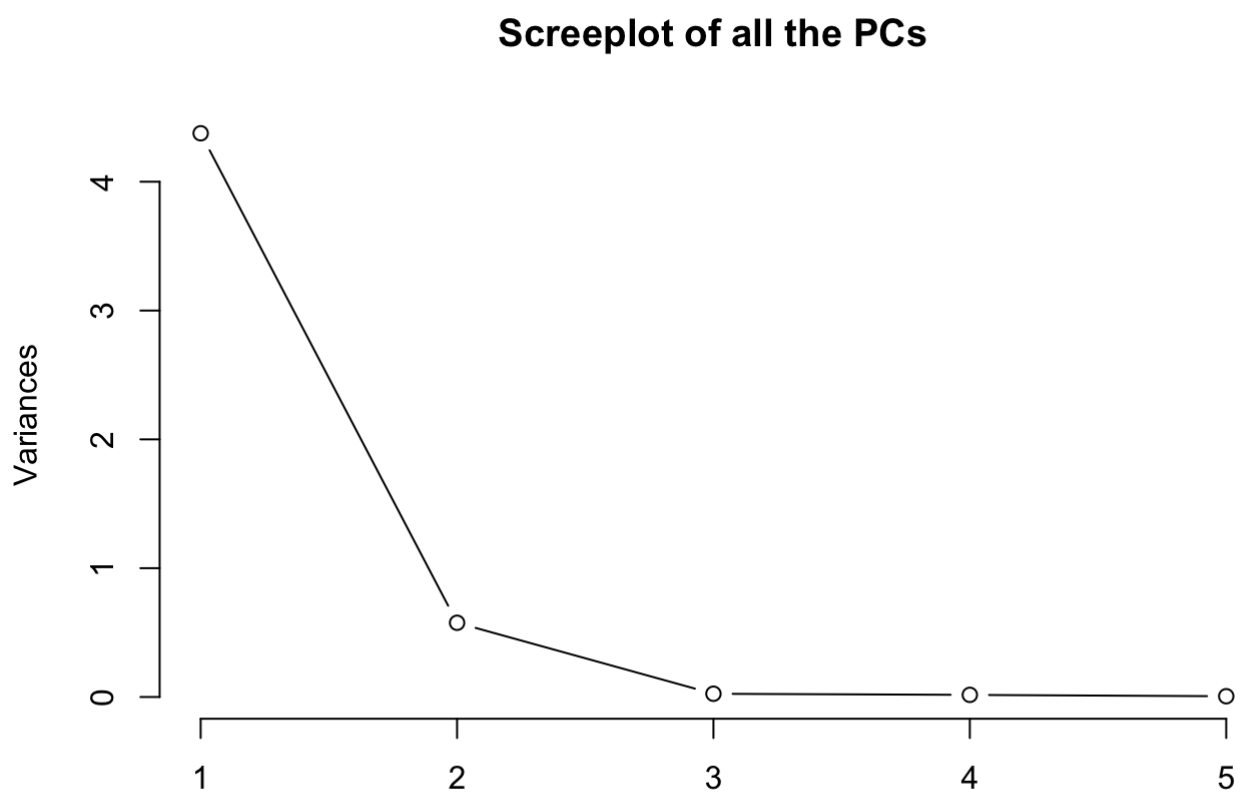
```
X_data <- scale(fiveband[1:5]) #Scaling the data is not needed here makes marginal impro
vement
y_data <- fiveband$class
(colnames(X_data)) # The column names tells me the names of all the features in data se
t.
```

```
## [1] "u" "g" "r" "i" "z"
```

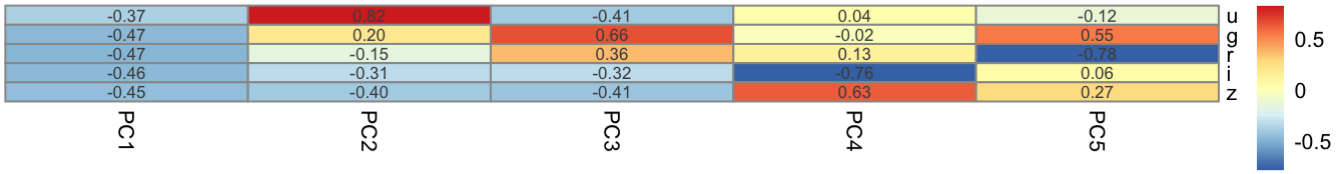
```
space_pc <- prcomp(X_data) # PCA Analysis
summary(space_pc)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5
## Standard deviation     2.0919 0.7590 0.15892 0.12885 0.07688
## Proportion of Variance 0.8752 0.1152 0.00505 0.00332 0.00118
## Cumulative Proportion  0.8752 0.9905 0.99550 0.99882 1.00000
```

```
screeplot(space_pc, type = "line", main = "Screeplot of all the PCs")
```



```
rotation <- as.matrix(space_pc$rotation) # This gives us a good sense of which features
 are being contribute to the PC
pheatmap(rotation,fontsize = 8,cellheight = 7.18,   cluster_cols = FALSE,display_numbers
= TRUE, cluster_rows = FALSE) # The closer it is to -1 or 1 the higher the contribution.
```



The five-band (u, g, r, i, z) CCD-based photometry's first 2 principal components can be explained by 99% plus and we can incorporate the other features as it is! We will make use of values from loc (7 features) + space_pc (2 Principal Components) features for our final model to focus on the classification.

```
temp <- cbind(scale(loc[1:5]),loc[6]) #SCALED OTHER Features
sloan_space <- cbind(space_pc$x[,1:2],temp)
cat('Data Dimensions Summary---------------------------------\n\n',"ORGINAL DATA MODEL
: ",dim(sloan_data),"\n","PCA MODEL            : ",dim(space_pc$x[,1:2]),"\n","Extra fea
tures MODEL : ",dim(loc),"\n","ORGINAL DATA MODEL   : ",dim(sloan_space),"\n","We have r
educed the number of features by more then half as of now!")
```

```
## Data Dimensions Summary---------------------------------
##
##  ORGINAL DATA MODEL   :  10000 18
##  PCA MODEL            :  10000 2
##  Extra features MODEL :  10000 6
##  ORGINAL DATA MODEL   :  10000 8
##  We have reduced the number of features by more then half as of now!
```

```
head(sloan_space) # This dataset is what we will be using which has all the features sca
led appropriately.
```

	PC1 <dbl>	PC2 <dbl>	ra <dbl>	dec <dbl>	redshift <dbl>	plate <dbl>	mjd <dbl>	class <chr>
1	1.0636082	1.60553875	0.1674500	-0.5848935	-0.3697126	1.0314378	1.3092452	STAR
2	0.1815453	0.07078950	0.1688531	-0.5830851	-0.3698308	-0.6361808	-0.8791534	STAR
3	-1.3758974	0.57441807	0.1705658	-0.5834461	-0.0530244	-0.6563062	-0.6091605	GALAXY
4	1.5039829	-0.57398649	0.1745488	-0.5864714	-0.3699741	1.0314378	1.3092452	STAR
5	1.1479036	-1.29082676	0.1748158	-0.5843833	-0.3681711	1.0314378	1.3092452	STAR
6	-2.6819022	-0.08587088	0.1740600	-0.5815617	-0.3688804	-0.6356217	-0.8454043	STAR

6 rows

In this preliminary analysis we focused on dimension reductions but not row data reductions as all there were no missing values, and there are no signs which indicate the data points are off and bad. However, we did do some column data reductions as some features were not nessecary and other features were transformed with the aid of PCA to preserve the variance.

Unsupervised Learning

K means Clusterings

When we scale the data for k means the unequal variances leads to us putting more weight on variables that have a smaller variance so to mitigate that we will be working with scaled data throughout.

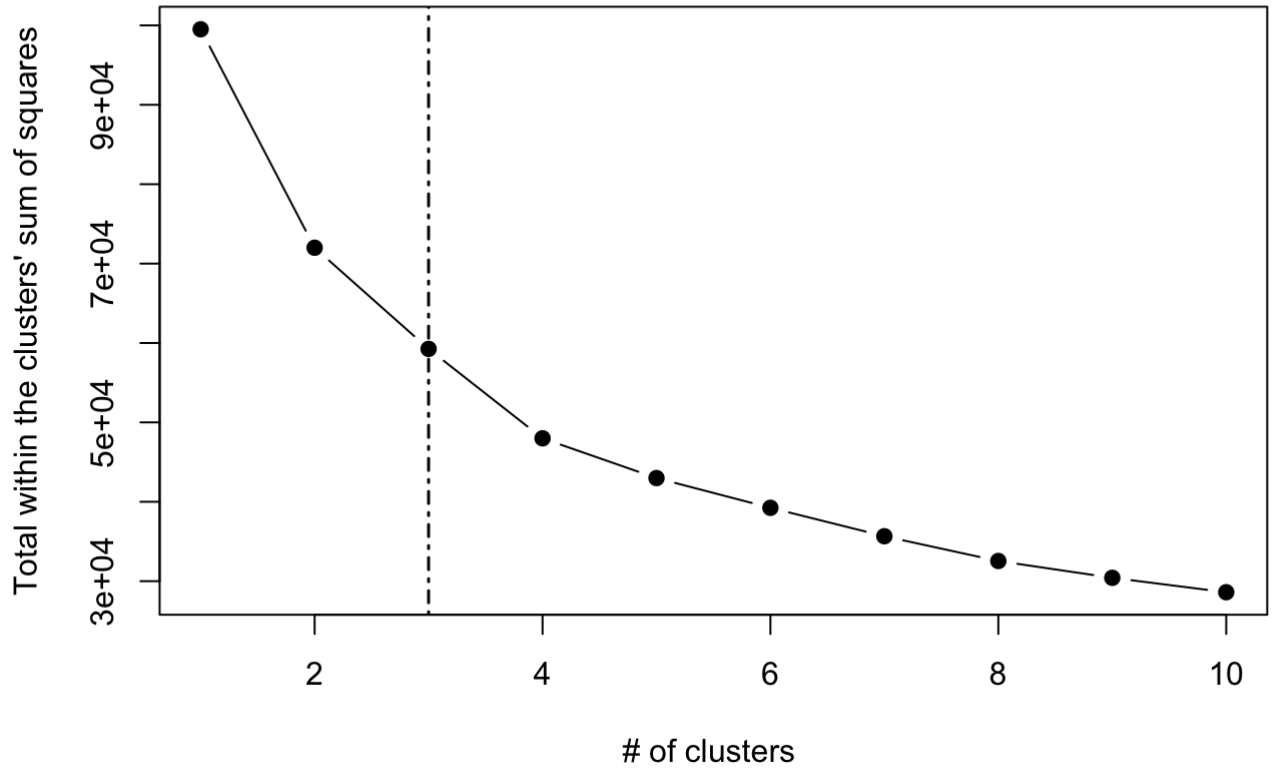
```
(colnames(sloan_space[1:7]))
```

```
## [1] "PC1"      "PC2"      "ra"       "dec"      "redshift" "plate"    "mjd"
```

```
sloan_x_data <- sloan_space[1:7]
sloan_y_data <- sloan_space$class

ss <- sapply(1:10, function(k){kmeans(sloan_x_data, k, nstart=100, iter.max = 10 )$tot.w
ithinss}) # The elbow method
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 500000)
```

```
plot(1:10, ss, type="b", pch = 19,xlab="# of clusters ", ylab="Total within the cluster
s' sum of squares") # The total within sum of squares is measuring how compact the clust
ers are and we are going to want to minimize that to a certain degree.
abline(v=3, lwd=1.5, lty=4)
```




```
k <- kmeans(sloan_x_data, 3)
(table(k$cluster,as.factor(sloan_y_data)))
```

```
##
##     GALAXY  QSO STAR
##   1    998   11 1674
##   2     85  756 1178
##   3   3915   83 1300
```

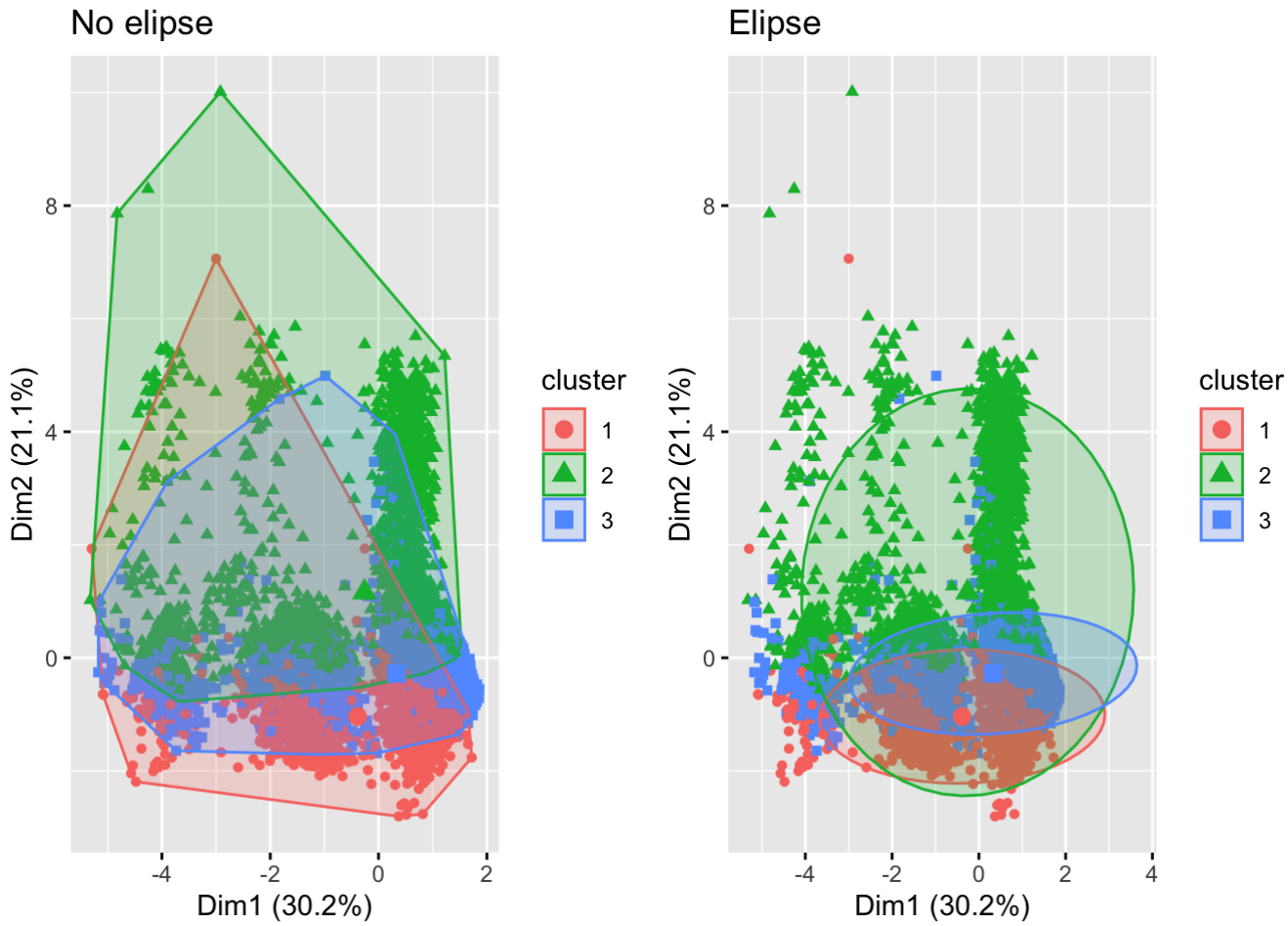
```
(colnames(sloan_space[1:4]))
```

```
## [1] "PC1" "PC2" "ra"  "dec"
```

```
sloan_x_data2 <- sloan_space[1:4]
k <- kmeans(sloan_x_data2, 3)
(table(k$cluster,as.factor(sloan_y_data)))
```

```
##
##     GALAXY  QSO STAR
##   1    884   11 1207
##   2    948  782 1310
##   3   3166   57 1635
```

```
p <- fviz_cluster(k, data = sloan_x_data,geom = "point") + ggtitle("No elipse")
pe <- fviz_cluster(k, data = sloan_x_data,geom = "point",ellipse.type = "norm") + ggtitl
e("Elipse")
grid.arrange(p, pe, ncol = 2)
```

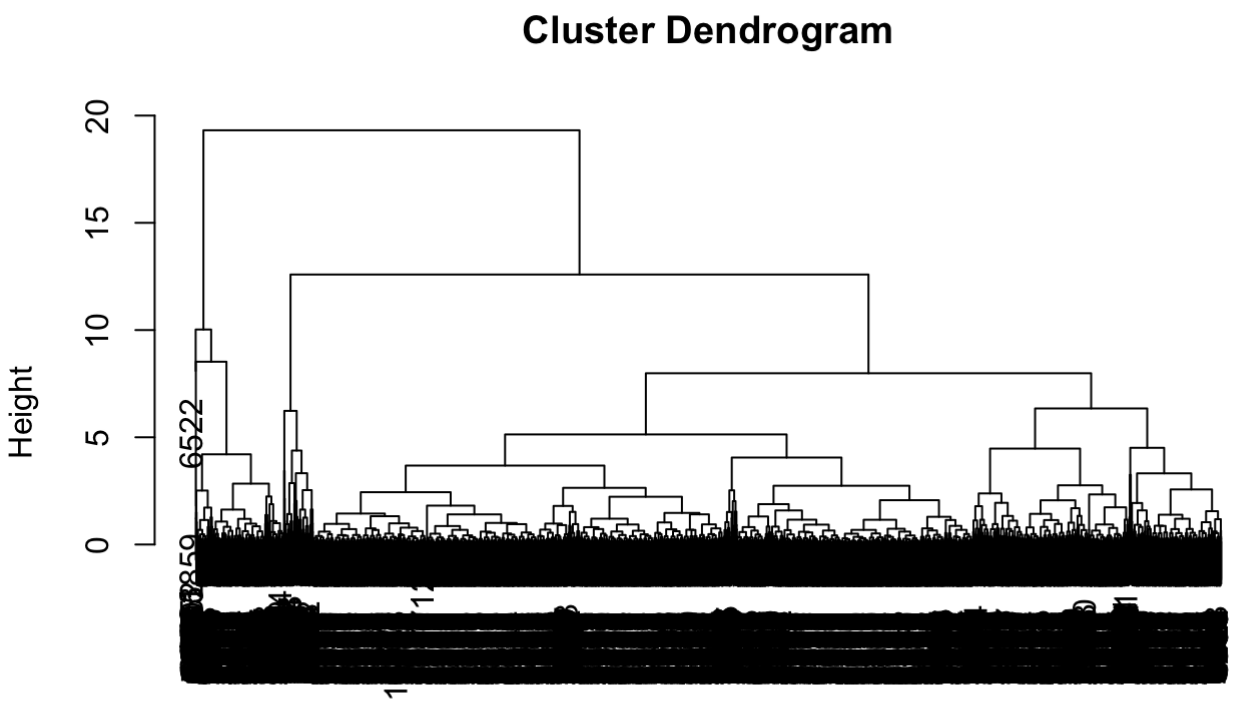


There is a significant amount of overlap in the three clusters which is not a good sign. In the confusion matrix we note that not each column has a max value in 3 separate rows which is not a good indicator.

Heirarchal Clustering

This is the bonus unsupervised technique we will focus on to see how it compares to the kmeans unsupervised technique in being able to identify clusters.

```
clusters <- hclust(dist(sloan_x_data[,1:2]))
plot(clusters) # hard to interpret
```



dist(sloan_x_data[, 1:2])
hclust (*, "complete")

```
treeforthree1 <- cutree(clusters,3)

clusters <- hclust(dist(sloan_x_data[,1:4]))
# plot(clusters) # hard to interpret
treeforthree2 <- cutree(clusters,3)

clusters <- hclust(dist(sloan_x_data))
# plot(clusters) # hard to interpret
treeforthree3 <- cutree(clusters,3)


table(sloan_y_data)
```

```
## sloan_y_data
## GALAXY    QSO   STAR
##   4998    850   4152
```

```
table(treeforthree1, as.factor(sloan_y_data))
```

```
##
## treeforthree1 GALAXY  QSO STAR
##             1   4866  238 3750
##             2     38  611  212
##             3     94    1  190
```

```
table(treeforthree2, as.factor(sloan_y_data))
```

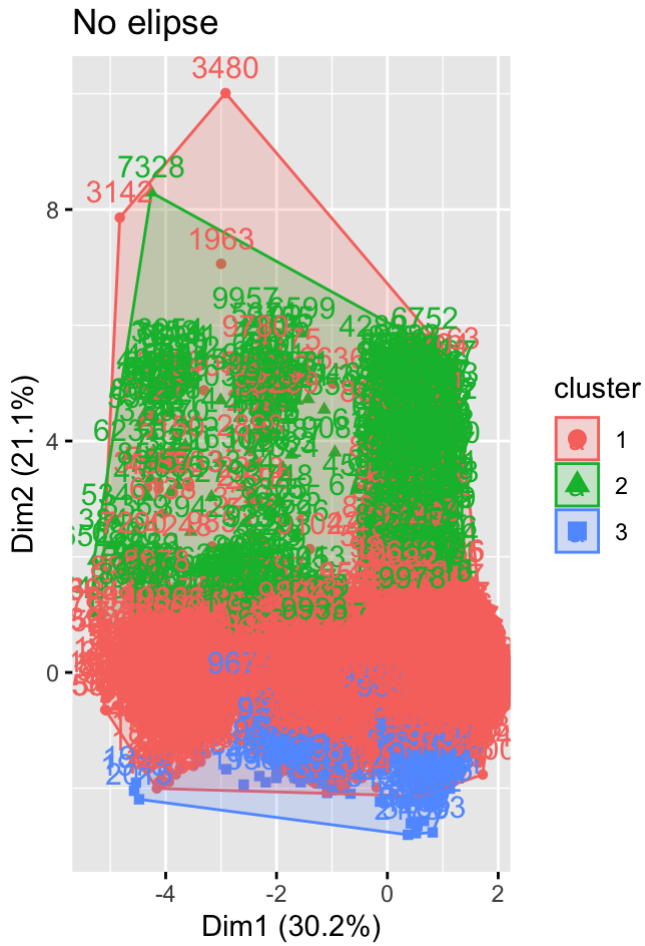
```
##
## treeforthree2 GALAXY  QSO STAR
##             1   4575  175 3082
##             2     96  673  583
##             3    327    2  487
```

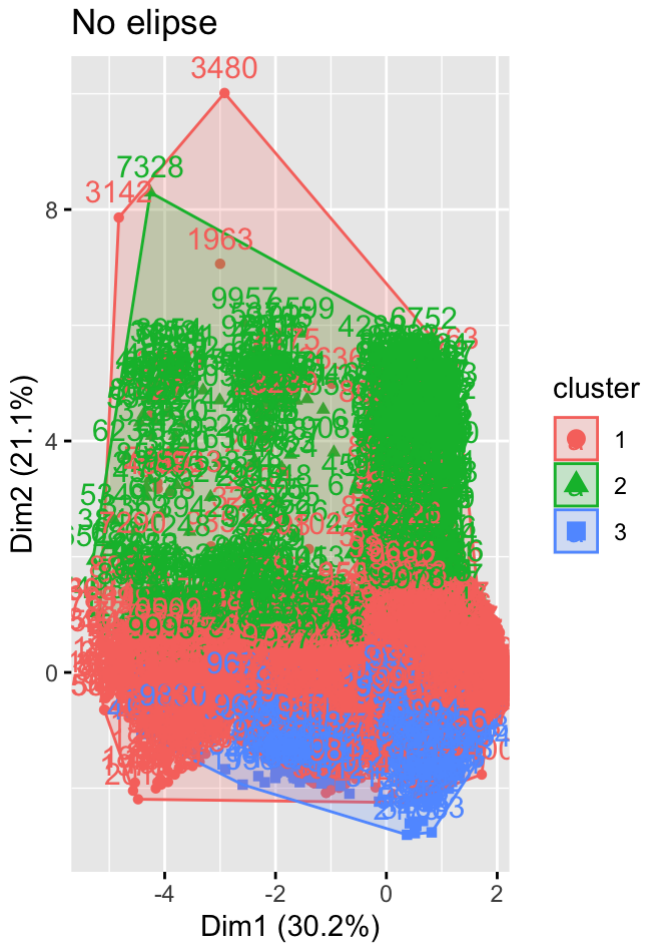
```
table(treeforthree3, as.factor(sloan_y_data))
```

```
##
## treeforthree3 GALAXY  QSO STAR
##             1   4869  850 3871
##             2    127    0  280
##             3      2    0    1
```

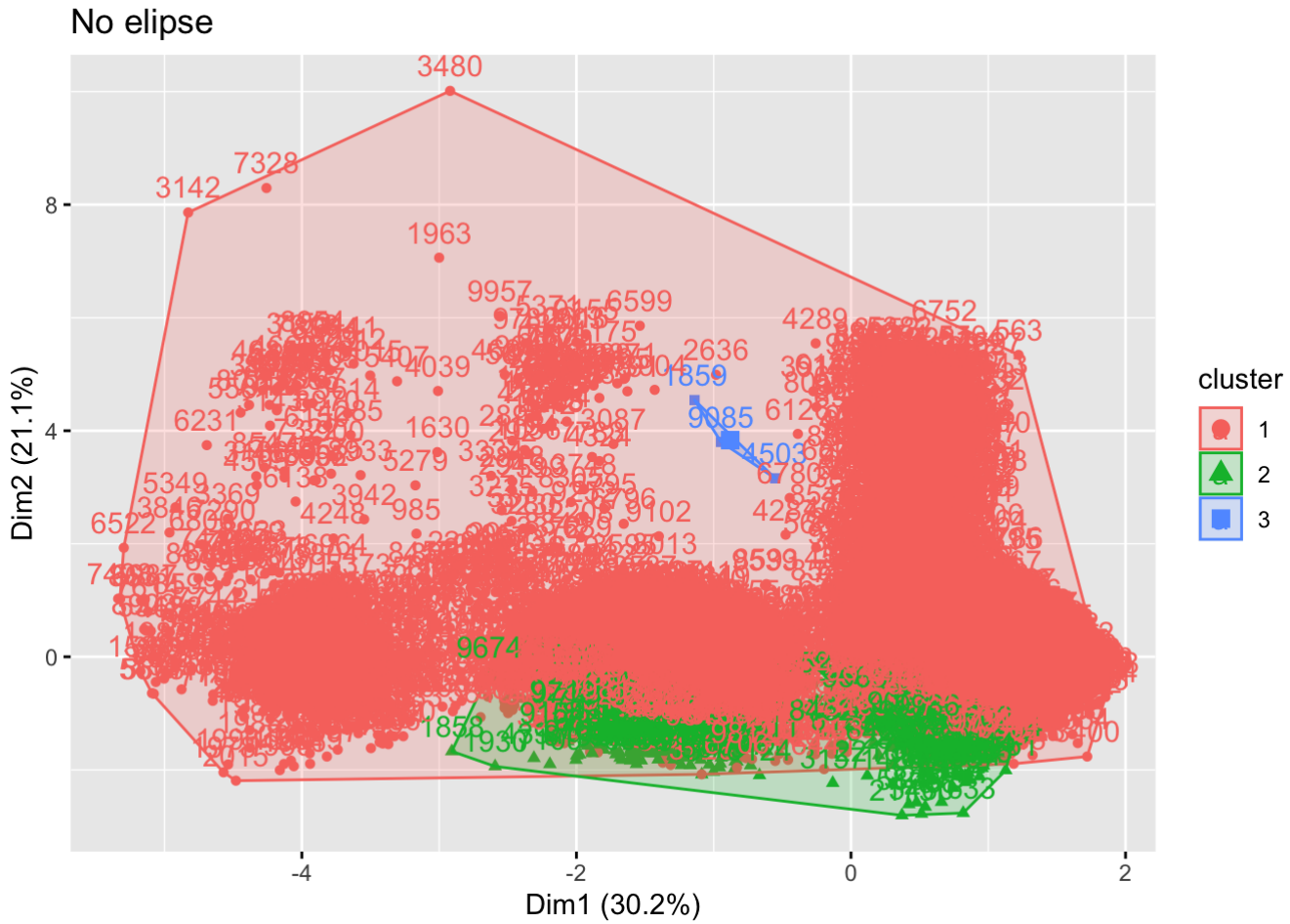
```
# We will not be using confusion matrices as it is not easy to identify which cluster is allocated to which but the more values we see one large value in each row column combinations and if we see a one of these large values in each row it's a good sign all we would be having is some misclassification which is expected. This is not a way to validate unsupervised techniques but is more so for supervised learning techniques.
#confusionMatrix((as.factor(treeforthree1)),as.factor(as.integer(as.factor(sloan_y_data))))
#confusionMatrix((as.factor(treeforthree2)),as.factor(as.integer(as.factor(sloan_y_data))))
#confusionMatrix((as.factor(treeforthree3)),as.factor(as.integer(as.factor(sloan_y_data))))
```

```
a <- fviz_cluster(list(data = sloan_x_data, cluster = treeforthree1,ellipse.type = "norm")) + ggtitle("No elipse")
b <- fviz_cluster(list(data = sloan_x_data, cluster = treeforthree2,ellipse.type = "norm")) + ggtitle("No elipse")
grid.arrange(a, b, ncol = 2)
```






```
fviz_cluster(list(data = sloan_x_data, cluster = treeforthree3,ellipse.type = "norm")) +
ggtitle("No elipse")
```



It's fair to conclude that heirarchal clustering didn't do a great job as we have a lot of overlap in the cluster maps but also if we look at the confusion matrices we note that it is starting to generalize the data into less than 2 clusters almost, which is not a good sign. K means was much better approach as compared to the heirarchal clustering techniques as there is less overlap but both are not good enough to accurately identify quasars from galaxies or stars. The next approach would be to train some models in a supervised manner. These parametric unsupervised learning techniques didn't bring too much insight to the table. If we are able to visualise the data in higher dimensions we would be able to see the segregation with more detail.

Supervised Learning

Train Test Split

We will be doing a 75/25 split on the data with our custom seed, to ensure replicability.

```
sloan_x_data <- sloan_space[1:7]
sloan_y_data <- sloan_space[8]

set.seed(4601) # ;)
# A good 75/25 train test split
train_index <- sample(1:nrow(sloan_x_data), 0.75 * nrow(sloan_x_data))
test_index <- setdiff(1:nrow(sloan_x_data), train_index)

X_train <- sloan_x_data[train_index,]
y_train <- sloan_y_data[train_index,]
class <- y_train
train <- cbind(X_train,class)
dim(X_train)
```

```
## [1] 7500    7
```

```
length(y_train)
```

```
## [1] 7500
```

```
#--------------------------------------------------------------------------
X_test <- sloan_x_data[test_index,]
y_test<- sloan_y_data[test_index,]
class <- y_test
test <- cbind(X_test,class)

dim(X_test)
```

```
## [1] 2500    7
```

```
length(y_test)
```

```
## [1] 2500
```

Multinomial Logistic Regression

Since this is a non-binary classification we will have to use multinomial logistic regression instead of simple logistic regression which is binary in nature.

```
(train)
```

	PC1	PC2	ra	dec	redshift	plate
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
4839	-0.6872564662	0.2560962885	0.050797777	-0.610731051	-0.1031589256	-0.65798336
9127	1.3369616756	-0.9763962020	0.701964403	1.991468979	-0.3700899200	-0.53834864

	PC1 <dbl>	PC2 <dbl>	ra <dbl>	dec <dbl>	redshift <dbl>	plate <dbl>
7197	0.5984478691	0.9911373240	-0.972892403	-0.551425592	-0.1517831604	1.86273138
3334	-3.6543757057	-1.6223197699	1.071680669	-0.618976667	2.4701219724	1.42835672
4927	0.4909534374	0.5957402174	0.114239916	-0.653527597	-0.1168870800	-0.63170844
1262	0.5915197000	-0.6039702547	-0.514708983	-0.540938742	-0.3684202998	-0.66636897
7349	-0.0247731382	-0.2015969076	1.359078786	1.486379807	-0.2815040917	-0.47014567
3529	-1.3909477247	-0.4947759525	1.433555605	-0.555410793	-0.3691671166	-0.62332283
6973	0.1871258761	0.3369515312	-0.512383043	1.851132947	-0.1836937533	-0.38628955
1624	-0.3918882066	-0.0504897959	-3.112503960	-0.067663774	-0.3694624604	0.24430841

1-10 of 7,500 rows | 1-8 of 9 columns Previous **1** 2 3 4 5 6 … 750 Next

```
multilog <- multinom(class ~ PC1+PC2+redshift, data = train )
```

```
## # weights:  15 (8 variable)
## initial  value 8239.592165
## iter  10 value 3671.029201
## iter  20 value 639.485030
## iter  30 value 445.206948
## iter  40 value 398.675009
## iter  50 value 398.464771
## iter  60 value 397.836295
## iter  70 value 397.773859
## iter  80 value 397.754958
## iter  90 value 397.739413
## final  value 397.738457
## converged
```

```
summary(multilog)
```

```
## Call:
## multinom(formula = class ~ PC1 + PC2 + redshift, data = train)
##
## Coefficients:
##      (Intercept)        PC1        PC2    redshift
## QSO    -3.388159 -0.4199493 -1.5371860    4.833551
## STAR -199.480799 -0.2190020 -0.6638012 -553.607497
##
## Std. Errors:
##      (Intercept)        PC1       PC2   redshift
## QSO      0.16976 0.09367841 0.2002817  0.5077468
## STAR    24.17400 0.10608725 0.2883214 65.6326557
##
## Residual Deviance: 795.4769
## AIC: 811.4769
```

```
train_pred <- predict(multilog, newdata = train, "class")
tbl <- table(train$class, train_pred)# Classification table

test_pred <- predict(multilog, newdata = test, "class")
tebl <- table(test$class, test_pred) # Classification table
cat("TRAIN ACCURACY : ", round((sum(diag(tbl))/sum(tbl))*100,2),"% \n","TEST ACCURACY  :
",round((sum(diag(tebl))/sum(tebl))*100,2),"% \n") # Accuracy is tested by summing the d
iagonal and dividing it by total obs.
```

```
## TRAIN ACCURACY :  98.67 %
##  TEST ACCURACY  :  98.56 %
```

Let's see if we can improve this

```
multilog2 <- multinom(class ~ ., data = train )
```

```
## # weights:  27 (16 variable)
## initial  value 8239.592165
## iter  10 value 2764.020749
## iter  20 value 1639.037509
## iter  30 value 657.481594
## iter  40 value 567.846560
## iter  50 value 499.361783
## iter  60 value 390.277201
## iter  70 value 388.197640
## iter  80 value 388.195544
## final  value 388.195521
## converged
```

```
summary(multilog2)
```

```
## Call:
## multinom(formula = class ~ ., data = train)
##
## Coefficients:
##      (Intercept)        PC1        PC2         ra         dec  redshift
## QSO    -3.718213 -0.4293338 -1.5504608 -0.1783097 -0.08789789    5.0002
## STAR -196.049271 -0.2033103 -0.6656137 -0.1123223 -0.30245252 -543.7780
##          plate        mjd
## QSO  3.0518455 -3.2119222
## STAR 0.8008921 -0.4464081
##
## Std. Errors:
##      (Intercept)       PC1       PC2        ra       dec   redshift     plate
## QSO    0.2480768 0.0951517 0.2103247 0.1169981 0.1301377  0.5223729 0.9438239
## STAR  26.5370297 0.1083173 0.3092462 0.2340455 0.2267404 72.1094350 1.0158069
##            mjd
## QSO  1.0299469
## STAR 0.9285649
##
## Residual Deviance: 776.391
## AIC: 808.391
```

```
train_pred <- predict(multilog2, newdata = train, "class")
tbl2 <- table(train$class, train_pred) # Classification table

test_pred <- predict(multilog2, newdata = test, "class")
tebl2 <- table(test$class, test_pred) # Classification table

cat("TRAIN ACCURACY  : ",round((sum(diag(tbl2))/sum(tbl2))*100,2),"% \n", "TEST ACCURACY
: ",round((sum(diag(tebl2))/sum(tebl2))*100,2),"% \n") # Accuracy
```

```
## TRAIN ACCURACY  :  98.75 %
##  TEST ACCURACY  :  98.68 %
```

```
#From previous model
#TRAIN ACCURACY :  98.67 %
#TEST ACCURACY  :  98.56 %
```

The train and test accuracy both rise by a small margin, so this means that the photometry data and redshift is enough to classify the data with good accuracy. Other features still contribute to the model. So to have a good understanding of what features are most relevant we will do some analysis in our next supervised learning algorithm called Randome Forest which has a unique plot which should help us see what features are most relevant in the classification

```
cat("SUMMARY----------------------------------------\n\nModel 1: \n")
```

```
## SUMMARY----------------------------------------
##
## Model 1:
```

```
(tbl)
```

```
##         train_pred
##          GALAXY  QSO STAR
##   GALAXY   3713   17   20
##   QSO        60  578    1
##   STAR        2    0 3109
```

```
(tebl)
```

```
##         test_pred
##          GALAXY  QSO STAR
##   GALAXY   1238    3    7
##   QSO        26  185    0
##   STAR        0    0 1041
```

```
cat("\nModel 2: \n")
```

```
##
## Model 2:
```

```
(tbl2)
```

```
##         train_pred
##          GALAXY  QSO STAR
##   GALAXY   3713   17   20
##   QSO        54  584    1
##   STAR        2    0 3109
```

```
(tebl2)
```

```
##         test_pred
##          GALAXY  QSO STAR
##   GALAXY   1239    3    6
##   QSO        24  187    0
##   STAR        0    0 1041
```

```
head(prob_table <- fitted(multilog2)) # The highest probability is the one that the clas
s indicator is identified.
```

```
##            GALAXY          QSO         STAR
## 4839 9.848256e-01 1.517436e-02 1.500126e-61
## 9127 8.604970e-03 9.252882e-05 9.913025e-01
## 7197 9.977897e-01 2.210350e-03 6.059954e-50
## 3334 1.333191e-05 9.999867e-01 0.000000e+00
## 4927 9.977725e-01 2.227503e-03 1.509262e-58
## 1262 1.030803e-02 1.061856e-04 9.895858e-01
```

Random Forest Classification

```
?randomForest
train$class <- factor(train$class) # Was not working without this fix.
#Should importance of predictors be assessed? YES
#Number of trees to grow. This should not be set to too small a number, to ensure that e
very input row gets predicted at least a few times.
(rf <- randomForest(formula = class ~ ., data=train, ntree=100, importance=TRUE, proximi
ty=TRUE))
```

```
##
## Call:
##  randomForest(formula = class ~ ., data = train, ntree = 100,      importance = TRUE,
proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 1.17%
## Confusion matrix:
##        GALAXY QSO STAR  class.error
## GALAXY   3709  22   19 0.0109333333
## QSO        43 595    1 0.0688575900
## STAR        3   0 3108 0.0009643202
```

The out of bag error estimate is really small which means much of the data was correctly classified. The next 2 confusion matrices focus more on data that the model was trained on and the data the model has never seen, making it the more interesting one to inspect and usually will be the one with poor accuracy as compared to the predictions made on the data that the model was trained on.

```
pred <- predict(rf,X_train)
(tb1 <- table(observed=train$class,predicted=pred))
```

```
##         predicted
## observed GALAXY  QSO STAR
##   GALAXY   3750    0    0
##   QSO         0  639    0
##   STAR        0    0 3111
```

```
pred <- predict(rf,X_test)
(tb2 <- table(observed=test$class,predicted=pred))
```

```
##          predicted
## observed GALAXY  QSO STAR
##   GALAXY   1240    3    5
##   QSO        16  195    0
##   STAR        1    0 1040
```

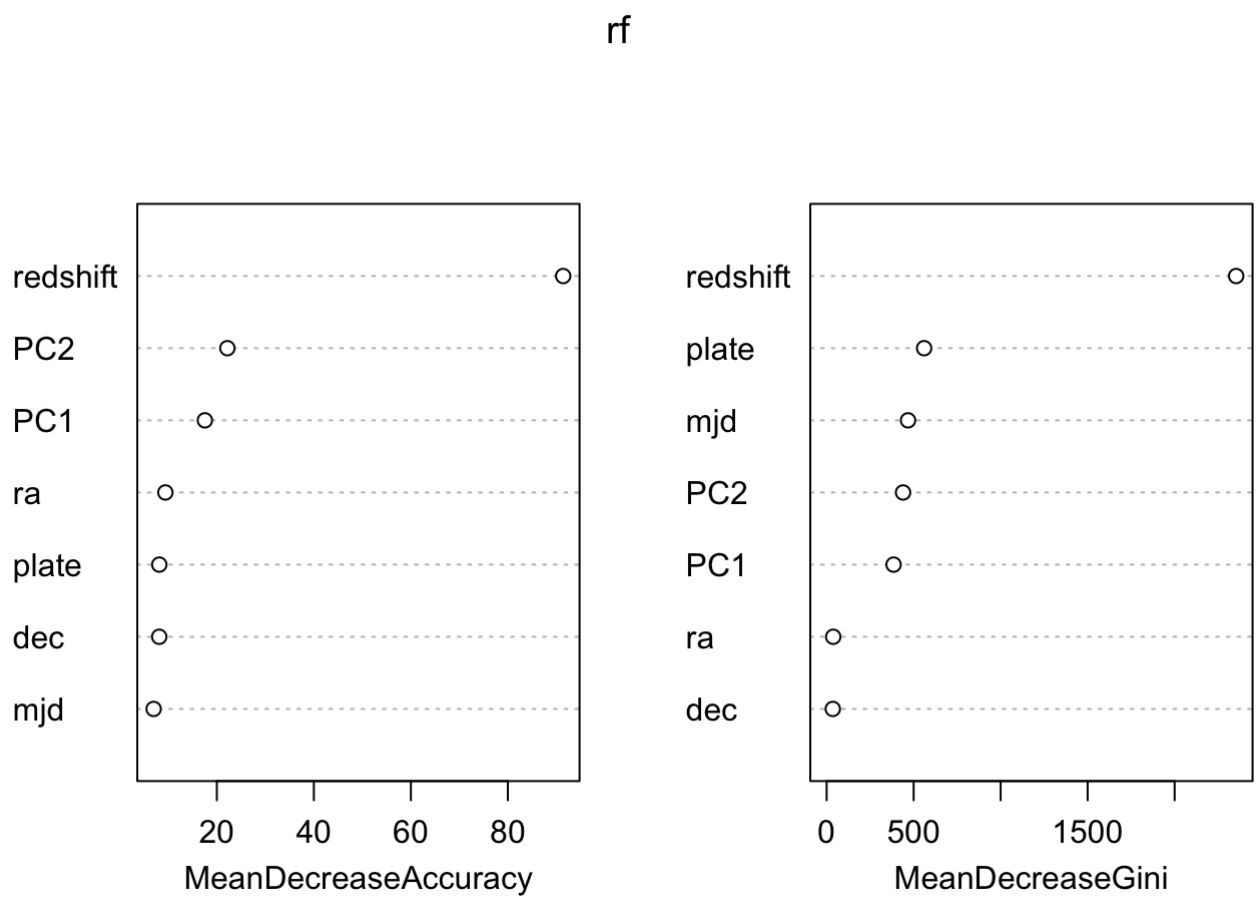
Here we see that the 3 most important features which intuitively from my understanding of the subject and as tested as the first model of the multinomial logistic regression models seem to be much more important in being able to classify the 3 classes appropriately.

```
cat("TRAIN ACCURACY  : ",round((sum(diag(tb1))/sum(tb1))*100,2),"% \n","TEST ACCURACY  :
",round((sum(diag(tb2))/sum(tb2))*100,2),"% \n") # Accuracy
```

```
## TRAIN ACCURACY  :  100 %
##  TEST ACCURACY  :  99 %
```

We see really good train accuracy but the more important one being the test is better than the multinomial regression models by a very small margin.

```
varImpPlot(rf) #Dotchart of variable importance as measured by a Random Forest
```



In this plot we see how redshift, PCA1 and PC2 are some of the more important features which help us distinguish between our classes as hypothesized in our multinomial logistic regression models due to minimum change in accuracy when all the other features were included. Cause the first graph of “Mean Decrease in

Accuracy" is the number of observations that are incorrectly classified by removing the feature from the model used in the random forest, which means the larger the value on the x axis the greater the impact it will have on the model. But in the other graph the higher Mean Decrease in Gini indicates higher the value on the x axis the higher the importance, so yes redshift is still important but plate and mjd seem more relevant than PC1 and PC2. The Mean Decrease in Gini measure is more so to do with how important a variable is in estimating the value of our class variable across all trees that make up our forest.

Conclusion

We saw that the unsupervised techniques were not able to able distinguish our space bodies with great accuracy and might need additional hyperparamter tuning for any improvements. We see that redshift and photometry data are good enough to help us classify the space objects. The spectroscopic data and location data didn't bring the same amount of value. mjd however had an interesting trend of finding fainter objects in the later end (closer to the present date) as technologies improved for us to better resolve fainter objects we can see in our observable universe. We also noted a drastic variation in the redshifts which could be due to the fact that the stars observed are much closer to us then the quasars and galaxies observed.