Abstract:

This project uses the PLASTiCC Light Curves dataset to explore different properties of Transients and Supernovae. In particular, it aims to solve three different research problems. Firstly, it investigates how we can separate the type of astronomical object based on its light curve properties. Secondly, it explores the relationship between the redshift of an astronomical object and its astronomical flux. Finally, using the 3d position of the galaxies, it looks into pockets/areas where we have been able to discover a higher concentration of galaxies, and the reasons behind this.

We found that it is possible to categorize light curves by visual and numeric factors. Furthermore, there exists a weak relationship between the redshift of an astronomical object and its astronomical flux, although it wasn't as strong as we had expected, likely due to the luminosity of the objects in the dataset being inconsistent. It would be interesting to explore this relationship for more astronomical objects. Finally, we found that the true_z, true_distmod, hostgal_specz variables do not explain why we find more galaxies in certain pockets of space.

Research Problem 1: How can we separate the type of astronomical object based on its light curve properties?

Introduction:

The Transients and Supernovae data set include both a comprehensive overview of all light curve data and the individual light curves themselves. But it's hard to visualize the differences between the observations, So we asked the question, is there a way we can group observations to figure out more about the lightcurves? Could there potentially be more information about the light curves that aren't in the metadata files? And additionally could we develop methods that could help us know what kind of light curve we are looking at. If there were tangible differences between the different types of light curves, then what kind of functions would be useful to either visualize or quantify them?

Data:

In order to analyze light curves on a deeper level, it is essential to understand the fundamentals of light curves, and their respective datasets. The metadata.csv file contains information about the light curves more generally, such as redshift, distance, time when it is the brightest and others, crucially we are concerned with the type of light curve it is, and the unique object id. We can then take those variables and match it up with its respective object in light cure batch 1-11. This csv file contains the light curves flux for each mjd, or day it is observed. This will allow us to plot the data through its 6 passbands. This allows us to receive a comprehensive view of the metadata, and the small changes in the light curve that the metadata csv file can not show us.

Methods/Analysis:

The sections of the methods and analysis portion break down the process of visualizing light curves. First we filter all observations into specific datasets, grouping them by the True_target variable . Then randomly sample one observation from one of the data sets to visualize. And plot all 6 passbands of x=mjd, y= flux_u,g,r,i,z,y) with the geom_line() function.

Filtering The Observations

The first part of the question is to filter the observations into different tibbles for easier processing. In the project description paper, There is a list of the different types of transients and space objects in the data sets. So we began by sorting the observations in the metadata file by their respective kinds of objects by their respective True_Target variable. This allows us to ensure that we are talking only the objects that are in the same groups of Transients and Supernovae. For this example we will be taking the SINa group of Transients to base the example on.

```r
#now we will filter the observations into data sets
```{r}
SNIa <- plasticc_test_metadata %>% filter(true_target == 90)
SNIa_91bg <- plasticc_test_metadata %>% filter(true_target == 67)
SNIax <- plasticc_test_metadata %>% filter(true_target == 52)
SNII <- plasticc_test_metadata %>% filter(true_target == 42)
SNIbc <- plasticc_test_metadata %>% filter(true_target == 62)
SLSN_I <- plasticc_test_metadata %>% filter(true_target == 95)
TDE <- plasticc_test_metadata %>% filter(true_target == 15)
KN <- plasticc_test_metadata %>% filter(true_target == 64)
AGN <- plasticc_test_metadata %>% filter(true_target == 88)
RRL <- plasticc_test_metadata %>% filter(true_target == 92)
M_dwarf <- plasticc_test_metadata %>% filter(true_target == 65)
EB <- plasticc_test_metadata %>% filter(true_target == 16)
Mira <- plasticc_test_metadata %>% filter(true_target == 53)
uLens_Single <- plasticc_test_metadata %>% filter(true_target == 6)
uLens_Binary <- plasticc_test_metadata %>% filter(true_target == 991)
ILOT <- plasticc_test_metadata %>% filter(true_target == 992)
CaRT <- plasticc_test_metadata %>% filter(true_target == 993)
PISN <- plasticc_test_metadata %>% filter(true_target == 994)
uLens_String <- plasticc_test_metadata %>% filter(true_target == 995)

tibble(SNIa)
tibble(SNIa_91bg)
tibble(SNIax)
tibble(SNII)
tibble(SNIbc)
tibble(SLSN_I)
tibble(TDE)
tibble(KN)
tibble(AGN)
tibble(M_dwarf)
tibble(EB)
tibble(Mira)
tibble(uLens_Single)
```

Retrieving A Random Sample

Now we will need to retrieve a singular random sample from one of the tibbles we have created, in this example the SINa data set. To do this we'll utilize a quick sample function without replacement so as not to remove the data. This is an essential part of answering our question because if the sample is not random, there is no guarantee that we will be attaining randomly selected data, and the kind of observations we are getting are connected to the order they are sorted in the .csv file. Lastly the set seed is essential for this specific example so the random sample is replicable.

```r
#Step 2 Selecting one observation from one of our data sets
```{r}
library("dplyr")

set.seed(033)
Test_set <- SINa[sample(1:nrow(SINa), 1), ]
glimpse(Test_set)
observation_id <- Test_set %>% select(object_id)
```
```

**Visualizing the Random Sample**

Now we can visualize the random sample of the SINa observation given. One small thing makes the process quite difficult though. In order to split the observations into tibbles, we needed to use the metadata csv file, but that csv does not include the flux data we need to plot. So we create a variable called Observation_id, then put it in the data visualization code. We can do this because the object_ids are the same in the metadata csv, and the batches in the lightcurve data.

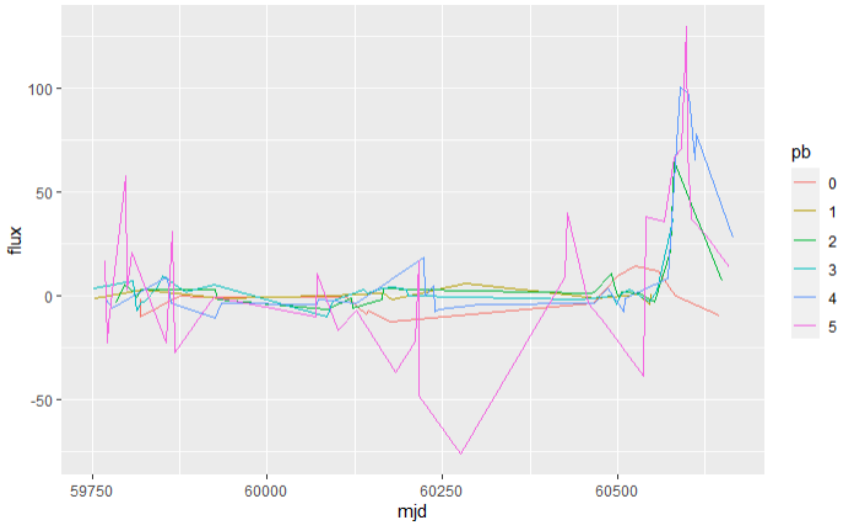
```r
#Kind of a hard step here, now we need to take our observation id from the metadata and find the matching light curve in test batches 1-11, just to make it easier heres a list of all the object ids and there respective batch ids, batch 1 = (13-342868), Batch 2 = (1000183-13952424), Batch 3 = (13952428-27000000) batch 4, (27000000-39000000) batch 5 = (39000000-53000000)

#Step 3 Plotting the test set with our visualization code
```{r}
library(readr)
plasticc_test_set_batch5 <- read_csv("~/PLASTICC/plasticc_test_set_batch5.csv")
```

```{r}
Test_LightCurve <- plasticc_test_set_batch5 %>% filter(object_id == as.numeric(observation_id))

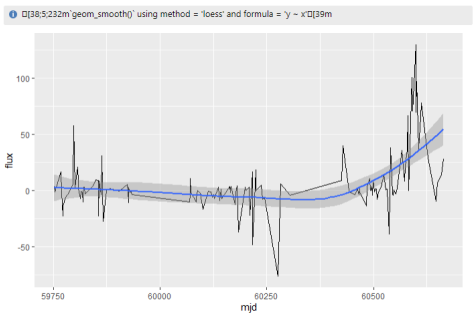
Test_LightCurve %>% group_by(pb=as.factor(passband)) %>% ggplot() + aes(x=mjd, y=flux, colour=pb) + geom_line()
```
```
```

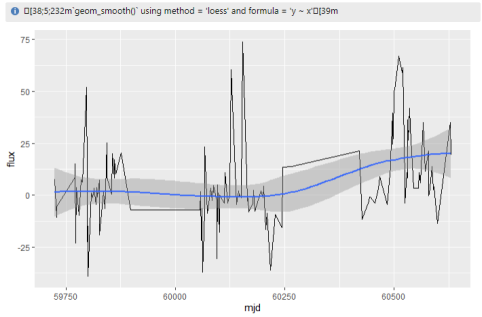
The visualization process is quite simple. We use geom_line to graph the flux vs mjd, which in this case represents time, and group the individual passbands in different colors. The visualization code gives us these results.

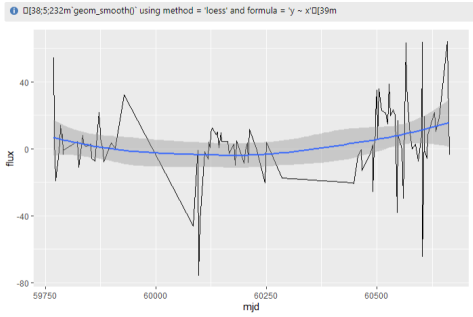


This light curve may be specific to the kind of Transient, or the specific sample we received, we need to understand the lightcurve more deeply. One way we can organize light curves is by their general trend, perhaps the curves of the SINa have a distinct shape? We achieve this by using the geom_smooth function for a non linear regression, we'll repeat this process several times until we have a good idea of what we can expect the SINa lightcurve to look like. By using a lossless linear approximation to get the trend of the curve, and then calculating the mean, median and range of flux to further quantify the properties of the light curve.

Results: Here are 3 examples of the SINa Visualisations with an approximation. Now we can identify a definite trend in the SINa light curve, we can identify a definite starting point at around 0 flux, an increasing slope and a significant dip after 60000 mjd, which is always the lowest point of the light curves flux, so the peak_mjd is around this point as well.







Now we have visual confirmation for the shape of a SINa lightcurve, now we will need to find some quantitative observations to give us more to go off of. Well use the range of the function, the mean, and median flux of the light curve.
Running this code gives us these values for the first observation(far left on the figure above)

```
#Amazing!! because it is a line graph, we can also summaraize the carachteristics of our object by utilzing
summarize and the lm function, and figure out more about this specifc light curve

#Step 4 Analizing The Light Curve
```{r}

Test_LightCurve %>% group_by(pb=as.factor(passband)) %>% ggplot() + aes(x=mjd, y=flux) + geom_line() +
geom_smooth()
lm(flux~poly(mjd,5,raw=TRUE),data=Test_LightCurve)
Test_LightCurve %>% summarise(mean = mean(flux))
Test_LightCurve %>% summarise(median = median(flux))

Test_LightCurve %>% select(flux) %>% min(na.rm = FALSE)
Test_LightCurve %>% select(flux) %>% max(na.rm = FALSE)
```
```

Minimum Flux value: -75.94791 Median Flux: 0.357056
Maximum Flux value: 129.9241 Mean Flux: 6.340362

Discussion: In conclusion, we began with knowing nothing about any of the light curves. Now we have a framework to observe the different kinds of behaviors that specific groups have. We can also take this one step further by compiling a data set of quantitative statistics, like the mean or standard deviation with a for loop. Additionally we could continue using this code to have a larger grasp of the visual form of objects. To answer our initial question, it is very possible to separate the kinds of objects by their qualities, and provide ourselves information to predict and understand what light curve we are looking at even if we don't have the type of object or other information.

Research Problem 2: Is there a relationship between the redshift of an astronomical object and its astronomical flux?

Introduction:

In order to better understand this question and how to go about solving it, it is important to define and understand redshift and flux. Essentially, redshift (and blueshift) are used by astronomers to work out how far an object is from Earth. Flux (or radiant flux), F, is the total amount of energy that crosses a unit area per unit time. The flux of an astronomical source depends on the luminosity of the object (L) and its distance from the Earth (r), according to the inverse square law: $F = \frac{L}{4\pi r^2}$. Therefore, if the luminosity of the astronomical objects in the data set is about the same for objects at different distances from earth (which it should be, since Luminosity is an intrinsic quantity that does not depend on distance), then we should expect to see a decrease in astronomical flux as the measured redshift (and therefore also distance) increases.

Data:

For the object's redshift, we are using the measured 'hostgal_specz' value from the metadata CSV files. The astronomical flux values are separated by their photometric passbands. We tested the relationship with flux from all the different passbands, and the results were relatively consistent throughout. For simplicity, we used just the ultraviolet flux ('tflux_u') for further analyzing the relationship. For this question, we filtered out all the data which did not have values for redshift or ultraviolet flux.

```{r}
metadata2 <- metadata %>% filter(hostgal_specz != 0, tflux_u != 0)
```

Methods/Analysis:

In order to analyze the relationship between the redshift of an astronomical object and its astronomical flux, we can use methods such as the 'cor()' function, as well as linear regression

using the 'lm()' function. In our case, the redshift is our predictor variable, and the flux is our response variable. Furthermore, we can use scatter plots with a line of best fit to easily visualize any relationships between the two.

Below is our first look at the relationship:

```{r, fig.height = 2}
ggplot(metadata2, aes(hostgal_specz, tflux_u)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```
```{r}
cor1 = cor(metadata2$hostgal_specz, metadata2$tflux_u)
```

Using the 'cor()' function, we get a value of approximately 0.023. This is very low, which suggests that there is no correlation between the redshift of an astronomical object and its astronomical flux. However, by examining our scatterplot, we can see that there does appear to be correlation for objects with larger astronomical flux values.

We can further explore this by filtering out all objects with flux values of less than 600:
```{r, fig.height=2}
metadata2 <- metadata2 %>% filter(tflux_u > 600)

ggplot(metadata2, aes(hostgal_specz, tflux_u)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

```{r}
cor2 = cor(metadata2$hostgal_specz, metadata2$tflux_u)
```

Now, using the 'cor()' function we get a value of approximately -0.24. This is much better than our previous value. It suggests that there exists a correlation (albeit a weak one) between the redshift of an astronomical object and its astronomical flux for objects with flux values over 600.

We can further explore this relationship with linear regression (using the 'lm()' function):
```{r}
lm(tflux_u ~ hostgal_specz, data = metadata2) %>% summary()
```

Results & Discussion:

As previously mentioned, the correlation value of approximately -0.24 suggests that a relationship between the redshift of an astronomical object and its astronomical flux for objects with flux values over 600 does exist, however it is not quite as strong as we had originally hoped. Similarly, our very low R-squared values (~0.05) tell us that the model is not great at making accurate predictions because there is a great deal of unexplained variance. On the other hand, the low p-value (~0.02) suggests that we can be reasonably sure that our predictor (redshift) does influence the dependent variable (flux). However, we know that this is not exactly the case since the flux equation, $F = \frac{L}{4\pi r^2}$, is not affected by redshift. Rather it is important to clarify that there exists a confounding variable, the object's distance to the Earth, which affects both the object's measured redshift value as well as its astronomical flux. That is, a change in an object's redshift value does not directly change its flux value, but rather an increase in its distance to the Earth would cause both an increase in its redshift value as well as a decrease in its flux value (and vice versa).

Furthermore, the reason for the weak relationship can likely be explained by the luminosity of objects in the data set varying more than expected. We had assumed that the luminosity of the astronomical objects in the data set would be about the same for objects at different distances from earth (since Luminosity is an intrinsic quantity that does not depend on distance), however if it was inconsistent then our results would vary because flux is dependent on both the object's luminosity as well as its distance to the earth.

Research Problem 3: Using the 3d position of the galaxies, are there pockets where we've been able to discover more galaxies than others? If so, why?

Introduction:

Redshift is a measure of how much the light from a distant galaxy has shifted to longer wavelengths due to the expansion of the universe. By measuring the redshift of a galaxy, astronomers can determine its distance from Earth. Right ascension and declination are the celestial coordinates used to locate an object in the sky. Right ascension is measured in hours, minutes, and seconds, and it corresponds to the object's position along the celestial equator. Declination is measured in degrees, and it corresponds to the object's position north or south of the celestial equator. By combining the redshift, right ascension, and declination of a galaxy, astronomers can determine its three-dimensional position in space. The redshift provides the distance to the galaxy, while the right ascension and declination provide its position on the celestial sphere. Using trigonometry, astronomers can then convert these coordinates into a three-dimensional position in space.

We want to see if there are pockets in the 3d positioning of the galaxies, where there are patterns and more galaxies found than others. Obviously, it makes sense for closer galaxies to be found easier, but is there something other than distance that could help us find galaxies? It could be something like brightness for example. Let's jump into the data.

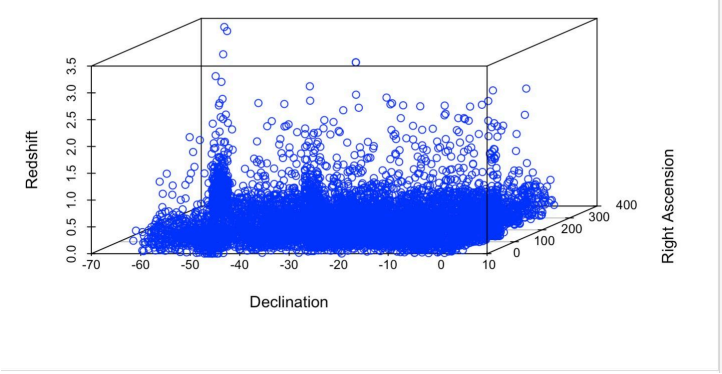
Data:

The data required for this question was obtained through the Metadata CSV file. The galaxy's exact 3d position in the universe can be obtained by using the “ra”, “decl”, and “hostgal_specz” variables all from the metadata CSV files. Ra is the right ascension, decl is the declination and hostgal_specz is the redshift. Redshift, right ascension, and declination provide the necessary information to determine a galaxy's three-dimensional position in space.

Methods/Analysis:

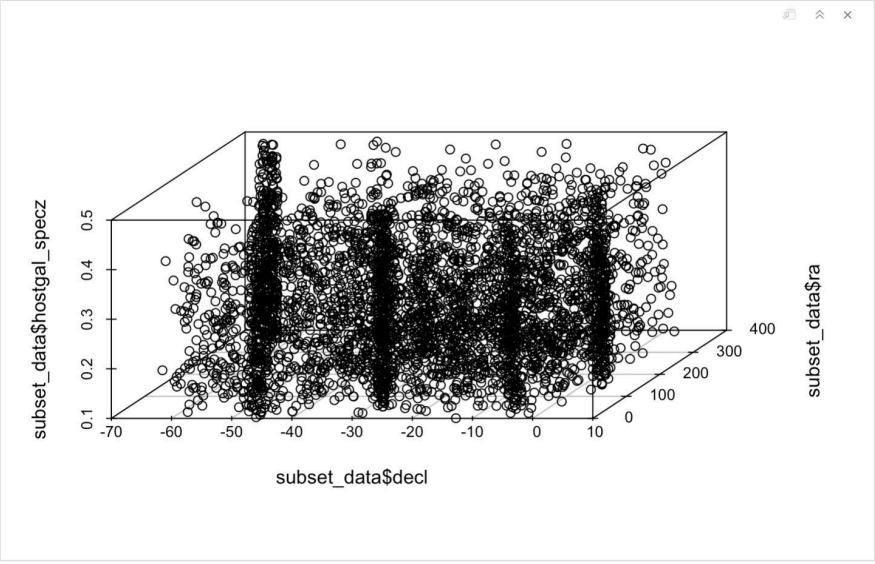
Below is the first look into the 3d positioning of the galaxies.

```{r}
scatterplot3d(metadata$decl, metadata$ra, metadata$hostgal_specz, color = "Blue",
              xlab = "Declination", ylab = "Right Ascension", zlab = "Redshift")
```



As you can see it’s very densely populated at the bottom. This is because these are the closest galaxies to us so it makes sense that we would know more of them. However, this doesn’t really help us too much and we’ll need to go a little further to see.

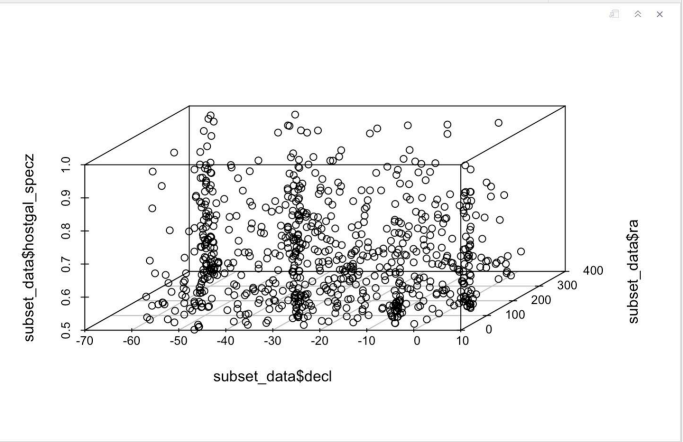
```{r}
subset_data <- metadata[metadata$hostgal_specz >= 0.1 & metadata$hostgal_specz <= 0.5, ]
scatterplot3d(subset_data$decl, subset_data$ra, subset_data$hostgal_specz)
```



Now this is where things get interesting. By filtering out galaxies that are within 0.1 and 0.5 redshift, we see a pattern. There are 4 distinct areas in the form of lines where there are much more densely populated galaxies. They are being found at a much higher rate than the others. What's the reason for us being able to find all these galaxies in these 4 specific pockets? Additionally, the pockets span across the hostgal_specz, which is our redshift. Redshift tells us how far away the galaxy is. Thus, these pockets exist regardless of how far away the galaxy is.

Going even further and shifting our filter to .5 to 1

```{r}
subset_data <- metadata[metadata$hostgal_specz >= 0.5 & metadata$hostgal_specz <= 1, ]
scatterplot3d(subset_data$decl, subset_data$ra, subset_data$hostgal_specz)
```



Although it isn't as clear as the last one, we can still see certain pockets where they are more dense than others.

Judging by those 2 examples, we can see that there should be a reason for the patterns.
There must be another variable in metadata that has a correlation.
We can find this out by using the cor() function in R, which calculates all the correlation values between each variable. Using this, we can see which variable is the cause for this. However, to use this function, we need to get the 3 positioning variables into 1.

This can be done by creating a new position variable and adding all the 3 variables used before. We need to get the 3 variables on the same scale, before we can add them together to get a single variable. Otherwise, we have ra, which will dominate the new variable as it is much larger than everything else. We can do this by using the scale() function in R, which standardizes each variable.

```{r}
metadata_standardized$pos_var <- metadata_standardized$ra + metadata_standardized$hostgal_specz +
metadata_standardized$decl

glimpse(metadata_standardized)

```

```
Rows: 7,848
Columns: 27
$ object_id          <dbl> 615, 713, 730, 745, 1124, 1227, 1598, 1632, 1920, 1926, 2072, 21…
$ ra                 <dbl[,1]> <matrix[31 x 1]>
$ decl               <dbl[,1]> <matrix[31 x 1]>
$ ddf_bool           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
$ hostgal_specz      <dbl[,1]> <matrix[31 x 1]>
$ hostgal_photoz     <dbl> 0.000, 1.627, 0.226, 0.281, 0.241, 0.000, 0.182, 0.701, 0.32…
$ hostgal_photoz_err <dbl> 0.000, 0.255, 0.016, 1.152, 0.018, 0.000, 0.030, 0.010, 0.336, 0…
$ distmod            <dbl> -9.000, 45.406, 40.256, 40.795, 40.417, -9.000, 39.728, 43.1…
$ mwebv              <dbl> 0.017, 0.007, 0.021, 0.007, 0.024, 0.020, 0.019, 0.021, 0.027, 0…
$ target             <dbl> 92, 88, 42, 90, 90, 65, 90, 42, 90, 65, 90, 42, 42, 90, 65, 16, …
$ true_target        <dbl> 92, 88, 42, 90, 90, 65, 90, 42, 90, 65, 90, 42, 42, 90, 65, 16, …
$ true_submodel      <dbl> 1, 1, 2, 1, 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2…
$ true_z             <dbl> 0.000, 1.817, 0.233, 0.301, 0.193, 0.000, 0.136, 0.688, 0.311, 0…
$ true_distmod       <dbl> 0.000, 45.703, 40.328, 40.969, 39.866, 0.000, 39.030, 43.097, 41…
$ true_lensdmu       <dbl> 0.000, 0.000, 0.004, -0.004, -0.002, 0.000, -0.002, -0.029, 0.00…
$ true_vpec          <dbl> 0.0, 0.0, 4.5, 257.7, -368.8, 0.0, -135.1, -626.6, -290.8, 0.0, …
$ true_rv            <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 3.1, 0.0, 0.0, 0.0, 0.0, 0.0,…
$ true_av            <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.051, 0.000, 0…
$ true_peakmjd       <dbl> 59570.00, 59570.00, 60444.38, 60130.45, 60452.64, 59570.00, 6062…
$ libid_cadence      <dbl> 69, 34, 9, 38, 1, 47, 20, 93, 107, 15, 96, 75, 13, 119, 17, 121,…
$ tflux_u            <dbl> 484.7, 108.7, 0.0, 0.0, 0.0, 2.3, 0.0, 0.0, 0.0, 16.6, 0.0, 0.0,…
$ tflux_g            <dbl> 3286.7, 117.7, 0.0, 0.0, 0.0, 11.6, 0.0, 0.0, 0.0, 130.6, 0.0, 0…
$ tflux_r            <dbl> 3214.1, 119.9, 0.0, 0.0, 0.0, 31.6, 0.0, 0.0, 0.0, 450.4, 0.0, 0…
$ tflux_i            <dbl> 3039.7, 149.6, 0.0, 0.0, 0.0, 240.0, 0.0, 0.0, 0.0, 2237.3, 0.0,…
$ tflux_z            <dbl> 2854.5, 147.9, 0.0, 0.0, 0.0, 632.4, 0.0, 0.0, 0.0, 4903.2, 0.0,…
$ tflux_y            <dbl> 2837.0, 150.5, 0.0, 0.0, 0.0, 1187.7, 0.0, 0.0, 0.0, 8229.6, 0.0…
$ pos_var            <dbl[,1]> <matrix[31 x 1]>
```

Now we have a single variable, pos_var, defining the position that we can use to find correlation. We can now use this to find the correlation values for our newly created position variable against every other variable in the dataset. And our results are really interesting.

```{r}
correlation_matrix2 <- cor(metadata_standardized)
correlation_matrix2["pos_var",]
```

```
      object_id              ra            decl         ddf_bool 
    0.005536356     0.479492152     0.475325620    -0.018221898 
  hostgal_specz hostgal_photoz hostgal_photoz_err         distmod 
    0.621328896     0.340962445     0.046629001     0.315008367 
         mwebv          target     true_target    true_submodel 
   -0.075118238     0.240735995     0.240735995    -0.026545656 
        true_z    true_distmod    true_lensdmu       true_vpec 
    0.621506227     0.328481220    -0.082937932    -0.002727539 
       true_rv         true_av    true_peakmjd   libid_cadence 
    0.104806777     0.027308456     0.080697451     0.016740052 
       tflux_u         tflux_g         tflux_r         tflux_i 
   -0.050534138    -0.074628834    -0.076662702    -0.087814229 
       tflux_z         tflux_y          pos_var 
   -0.101583906    -0.103582898     1.000000000 
```

Results & Discussion:

We see that we have a pretty even correlation amongst the 3 positioning variables, which shows that standardizing each worked pretty well. However, the only other variable with a good correlation is true_z. True_z measures redshift, thus it is essentially a different type of distance variable, and doesn't really answer our question on trying to find something other than distance as the reason we find certain galaxies. Going past true_z, we see a weaker correlation with true_distmod/distmod, however again these are just distance variables just like true_z. So unfortunately, there's nothing in this dataset that explains the pockets of space where we've been able to find galaxies regardless of distance from us. We were hoping for something such as the light properties showing correlation which would tell us that the brighter galaxies are found more often. However, the light property variables all have very low or non-zero correlation.

Conclusion: In conclusion our questions yielded results that gave us more information about the transient and supernovae data set. At the start of the project, we had a hard time visualizing the data set and understanding the variables. The process of creating questions we wanted to solve, researching the material then solving the questions familiarized us with the data set, and working with more complex material in general.

At the completion of the capstone project we discovered that it is possible to categorize light curves by visual and numeric factors. (Question 1) The correlation value of approximately -0.24 suggests that a relationship between the redshift of an astronomical object and its astronomical flux does exist, however it is not as strong as we had originally hoped. (Question 2) And the

true_z, true_distmod, hostgal_specz variables do not explain why we find more galaxies in certain pockets of space.(Question 3)

The results given may not have been what was expected initially, but the reasoning and tools used to find ways we could solve our problems proved that we understood the statistical process and the methods associated. Additionally, the research process from the beginning to the end has given us valuable experience in creating and solving our own statistical questions. These concepts and techniques we as a group learned through the duration of the capstone project can be used in future statistics projects.

Works Cited:

PLASTICC astronomical classification. Kaggle. (n.d.). Retrieved April 11, 2023, from https://www.kaggle.com/c/PLAsTiCC-2018