Daniel Crum
Dr. R. M. Musal
5/2/2021

A Logistic Regression on NASA's Exoplanet Archive

Goal:

Find a well fitted model able to provide the probability that a planet belongs to a Binary+ (more than one star) system given relevant and useful independent variables.

Executive Summary:

Based on my research I have found that Star Catalogues tend not to have good variables that may assist in the prediction of a binary system. This exoplanet catalogue downloaded as a .csv from NASA had good independent variables and a large dataset, better than one could hope for on average. However, although I was able to create a nice model, it has come with some drawbacks. I will further go into the drawbacks I found and the advantages later in the report.

In doing this research and modeling, I have found an effective way to interpret the probabilistic nature of a host star of an exoplanet (far away planet) being Binary+ in nature. Throughout this report I will refer to the Binary+ness of a star or the Binary+ as being an attribute where the host star belongs to a Star system with at least one other star. As this is an exoplanet catalogue, all the references to the star in relation to the planet are it's "Host Star". So, while a planet may belong to a Star System with 3 Stars, it will only have one "Host" Star due to the natural dominance of gravity over an object. So few and far between are planets having more than one "Host Star" that it is not provided as an attribute in the Catalogue from NASA (and likely never will be).

In the end, I was able to create a model using 7 independent variables that predict with 93.8% accuracy whether the host Star is Binary+. This can be useful for all endeavors whether you are a beginner astronomer, or intermediate or advanced, to quickly interpret the Binary+ness of a star given these features. I will include an R File that should, when paired with proper planets2.csv, will let the user click through each line of code so that all my efforts, in turn, are designed to work for the user as well.

Problem and Data Description:

The problem I am presented with is to create an accurate, meaningful, and useful model to predict the Binary+ness of any star, given certain features that are commonly available to all classes of astronomers.

This is a catalogue with information about the Host Star system of an exoplanet, and our objective is to predict whether the "Host Star" belongs to a Star System that is Binary+(2,3,4+) or if it is Solitary. The data is one of the largest exoplanet catalogues in existence and has an exorbitant number of columns that are unusable. Certain column headings like Planet Name and Method of Discovery were not included in the research due to my own discretion. However, this catalogue, once again, has one of the best collection of features that are relevant to a star being binary, even better than some star catalogues that I found (NASA does not have a star catalogue for public use).

Since this is an exoplanet catalogue, it is possible that the host star may be reiterated throughout the dataset. In fact, it is possible that the star system (if it is Binary+) may be referenced more than once since each row is an entry of another exoplanet. For example: if there is a Star system with 4 stars, and 20 planets, those four stars would be referenced a total of 20 times as the "Host Star", along with the host star's temperature, distance, etc. This could throw off the data and unduly increase the probability of a Binary+ system.

Other disadvantages that have come up are that other independent variables could be referenced more than once. Star temperature, for example, could reiterate more than once at the same value if there is, for example, a Star with 4 planets. Each entry of those 4 exoplanets would have a star temperature that is the same as the other 3.

Also, all these findings are predicated on the fact that they were entered into the database as detected exoplanets (using various detection methods), so the very processes or methodologies of detecting exoplanets themselves could throw off the data here and give skewed information relative to what we want to find. There are more drawbacks to the dataset that one could find, but I believe these are the primary ones.

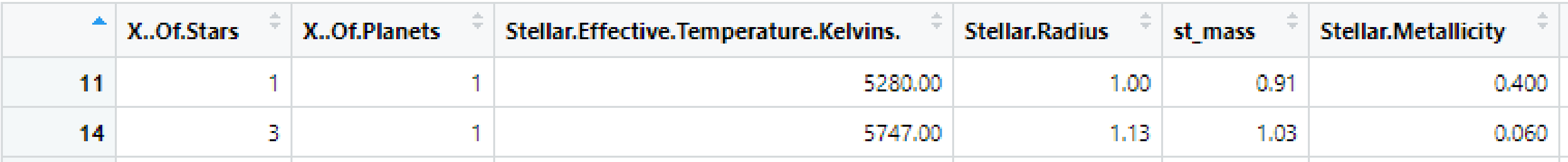
Describing and Visualizing the Data:

On the plus side, I was able to create a working dataset with 13,391 rows and 9 variables (one of them being the dependent variable we are to predict). I consider this a success. Originally my Dataset had approximately 29,387 Rows with 289 columns. I hand selected 22 Variables to include in Excel, and then I cleaned the data further with hopes to have over 1,000 rows to work with. There was a trade off when getting rid of NA values between the # of Columns I could have and the # of Rows. The more columns I wanted, the less rows I could have and vice versa. My Final dataset was agreed to be 13,391 rows and 8 independent variables. The 8 independent variables are indicated to be highly relevant to the Binary+ness of a host Star. My final model in fact only used 7 of those variables in predicting the Binary+ness of a star: # Of Planets, Stellar Temperature in Kelvins, Stellar Mass, Stellar Metallicity, Stellar Gravity, Stellar Density, and System Distance. I will remind you once again all the stellar properties are of the host star of the exoplanet in question, of which there can only be one.

In my mind the more rows that I have the more powerful my analysis will be and more meaningful my interpretations could be. I would have liked to have been able to use all 30,000 rows, but this would have left me with NA values and an unacceptable amount of skewness to the data (provided all the NA values). Also, the problem of creating a good model for predicting Binary+ness could have been approached in many ways. Since I ended up choosing a trade-off between rows and columns, this means that an analysis could have been done on the level of 1,500 rows instead of my 13,391 rows and could have provided meaningful results that way using different features. The step from 22 Independent variables (really 21 since one is the dependent variable) to 8, was one that is up to the interpretation of the modeler, and even preference. Also, for all duplicate entries into the data, one could have removed all the duplicate data, but this would have thrown off the data to a degree. The intrinsic problem comes once again from the lens of each entry being from the point of view of an Exoplanet and not the Star System itself. I believe there are certain advantages to this (like the relevant data, counts of the planets, correlation to the star) and disadvantages (multiple entries of host star temperature into the database; etc.).

My job in performing this analysis was to find a nice balance, where the interpretation was not meaningless, and I was not overfitting my model. It also important to acquire useful statistics based off of relevant independent variables.

It is also worth mentioning that initially the dataset includes a numerical value under # of Stars (my dependent variable). Figure 1 down below is a quick snapshot of the table containing that information in R.



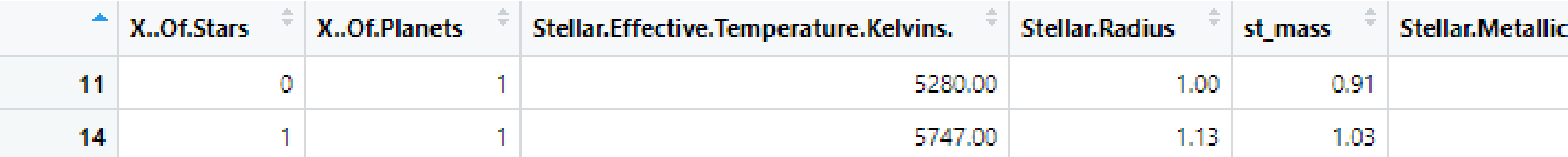
	X..Of.Stars	X..Of.Planets	Stellar.Effective.Temperature.Kelvins.	Stellar.Radius	st_mass	Stellar.Metallicity
11	1	1	5280.00	1.00	0.91	0.400
14	3	1	5747.00	1.13	1.03	0.060

Figure 1 ^

In Figure 1 the 14th entry has 3 stars associated with this planet. I then performed a loop, so that any entry with more than 2 stars is 1, and any solitary star is a 0.

The loop I am talking about is referenced in the **Code Appendix** **Section 2**.

This loop executed the functions that I wanted; leaving the same data to look like Figure 2 Below.



	X..Of.Stars	X..Of.Planets	Stellar.Effective.Temperature.Kelvins.	Stellar.Radius	st_mass	Stellar.Metallic
11	0	1	5280.00	1.00	0.91	
14	1	1	5747.00	1.13	1.03	

Figure 2 ^

I thought I might also provide the reader with some more visualizations to get a feel of the data. The following pages contain a number of GGPLOTS to give descriptive information.

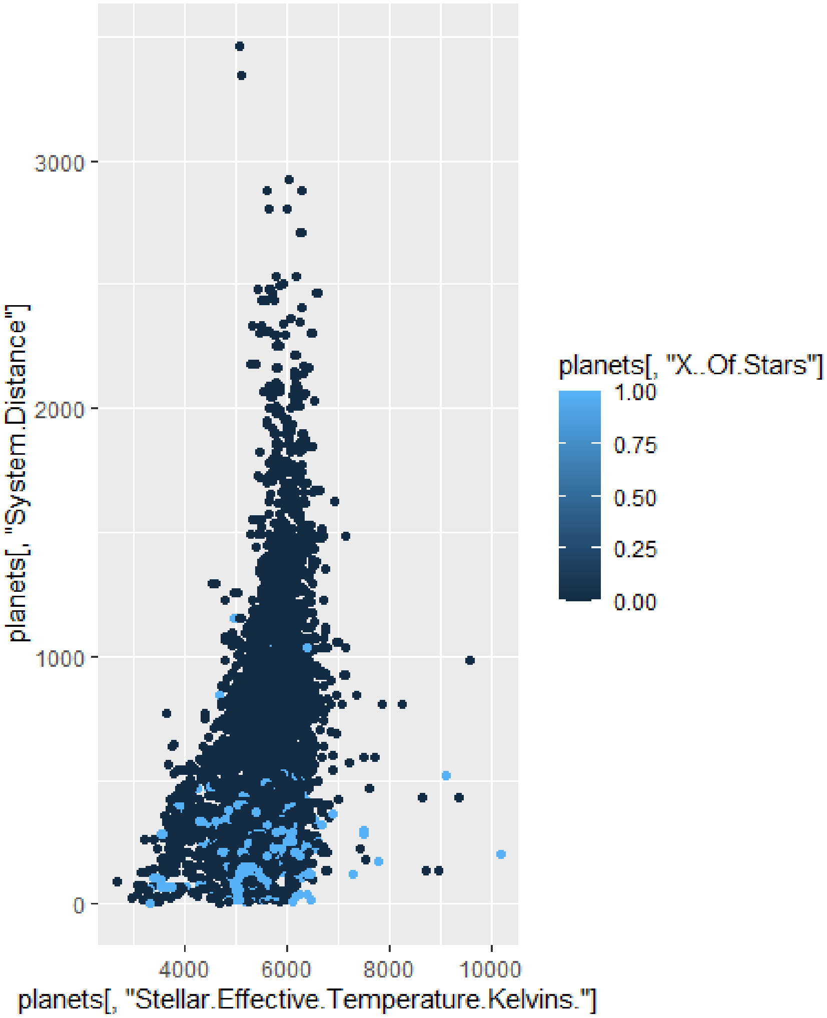


Figure 3

In the above graph I have shown you System Distance relative to Stellar Temperature, color coded by dark blue for singular and light blue for Binary+. One can see in these visualizations that there are a few outliers on either end of the spectrum of Temperature or Distance. Is there a correlation? There sure seems to be, but we must remember how this data is acquired and where it comes from, also what it is we are looking for. The techniques used to find planets may wear off after a certain distance. Since there are many billions of light years worth of distance to cover in our real search for exoplanets, the Y axis should be labeled accordingly. This would only be so if the techniques we used to discover exoplanets (or stars) were effective at those ranges, which clearly they are not. Furthermore, I have provided for the interested reader a similar plot of the Number of Planets in the system VS. the Stellar Mass of the host star in Figure 4 on the following page.

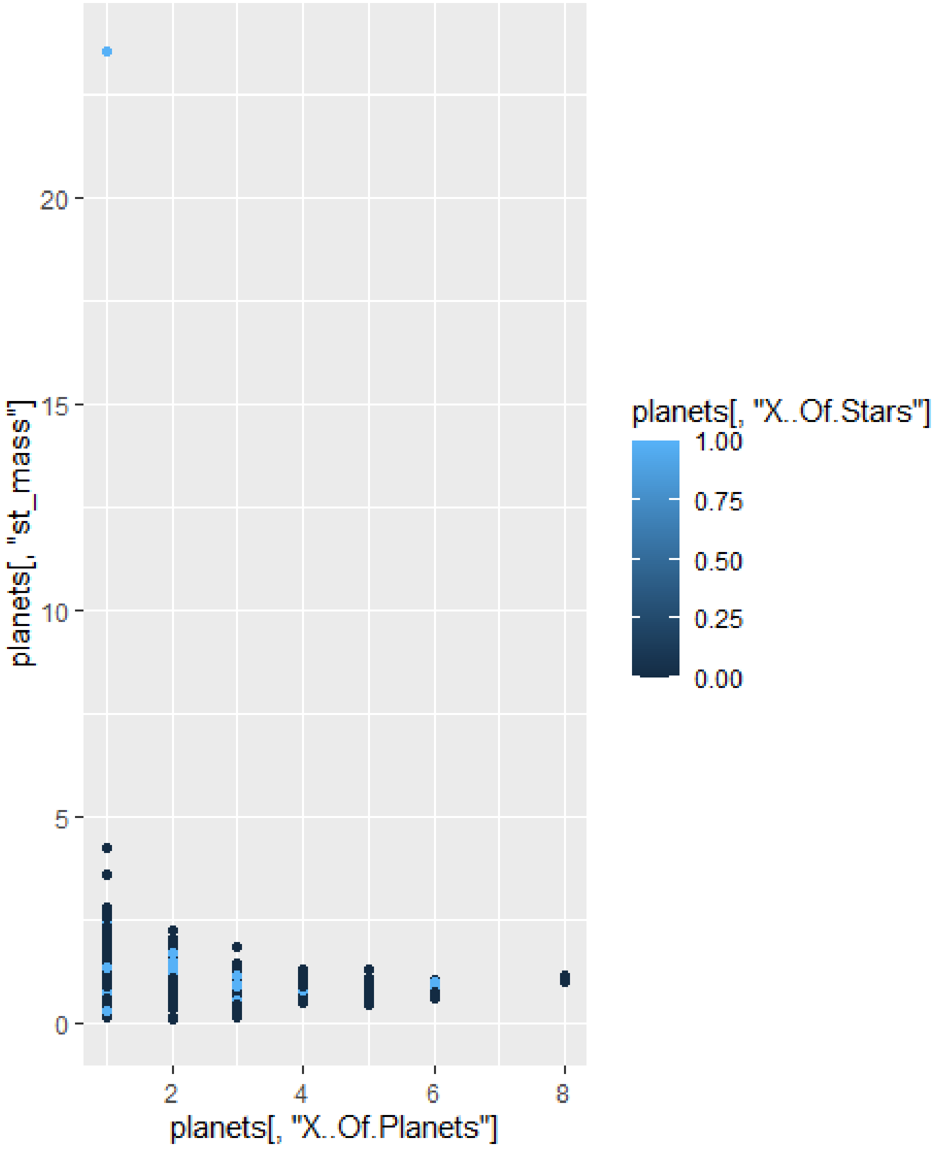


Figure 4

Besides having one clear outlier of one planet where the host star has ~23 time the mass of our own sun, the data is grouped into a set that has become very muddled due to the scaling of the graph. This graph would have to be huge to be able to view the data accurately. However, I have included this graph because it answers the question of, “How many planets are commonly found among our dataset and for what stellar masses do they belong?”. Similarly, I wish to answer another question the reader may have had, as I did, of: “Before converting the data to 0 or 1 (for solitary vs. Binary+ systems) what are the common numbers that star systems appear in? I have that graph represented in Figure 5 of the following page.

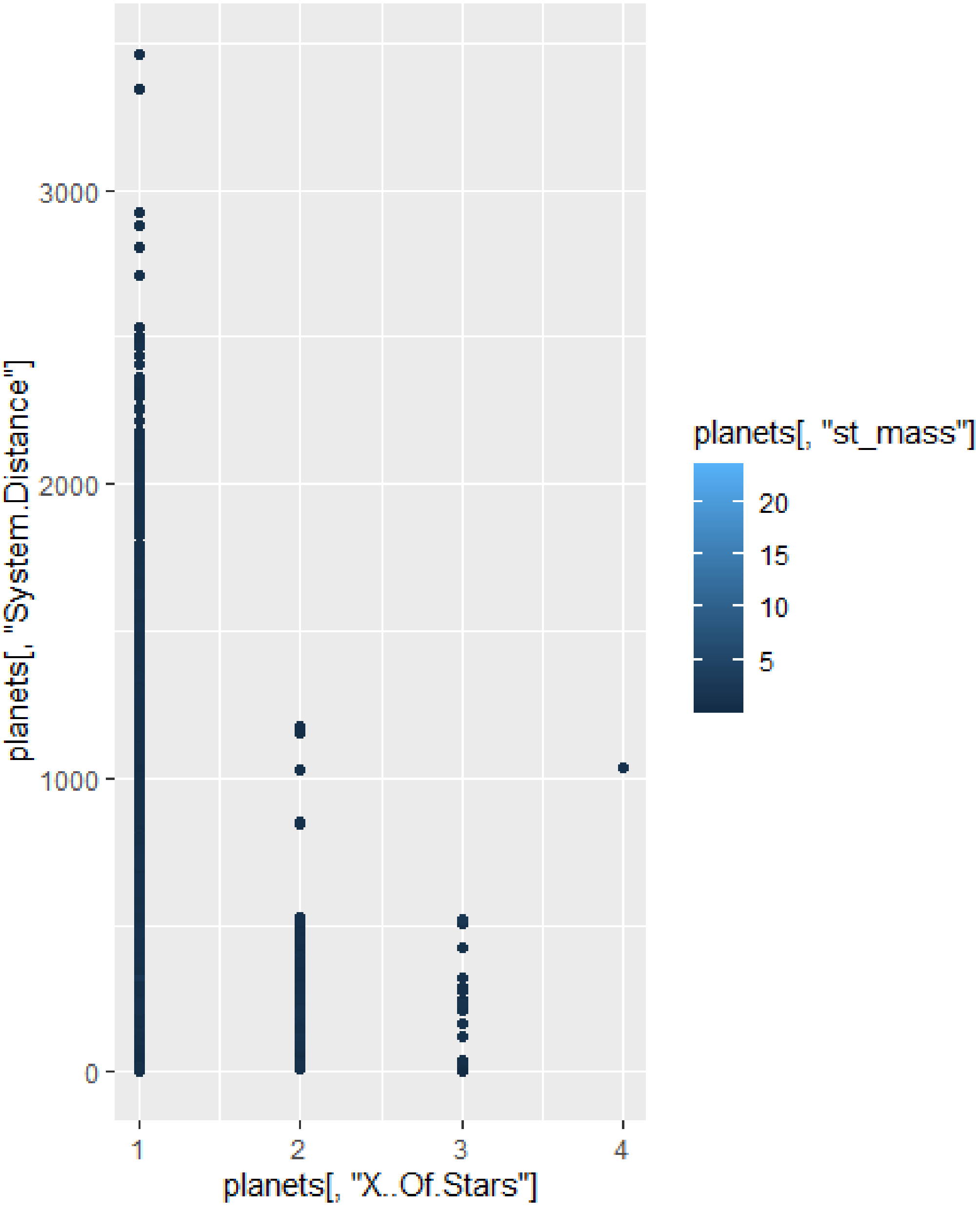


Figure 5

As you can see, there was a maximum of 4 stars found in any given star system. 4 stars was uncommon in this catalogue. Far more common are solitary star systems and Binary/Triple. Also, it is extremely rare for our data to have a star with a stellar mass more than 5x our own sun (as Figures 4 and 5 point out).

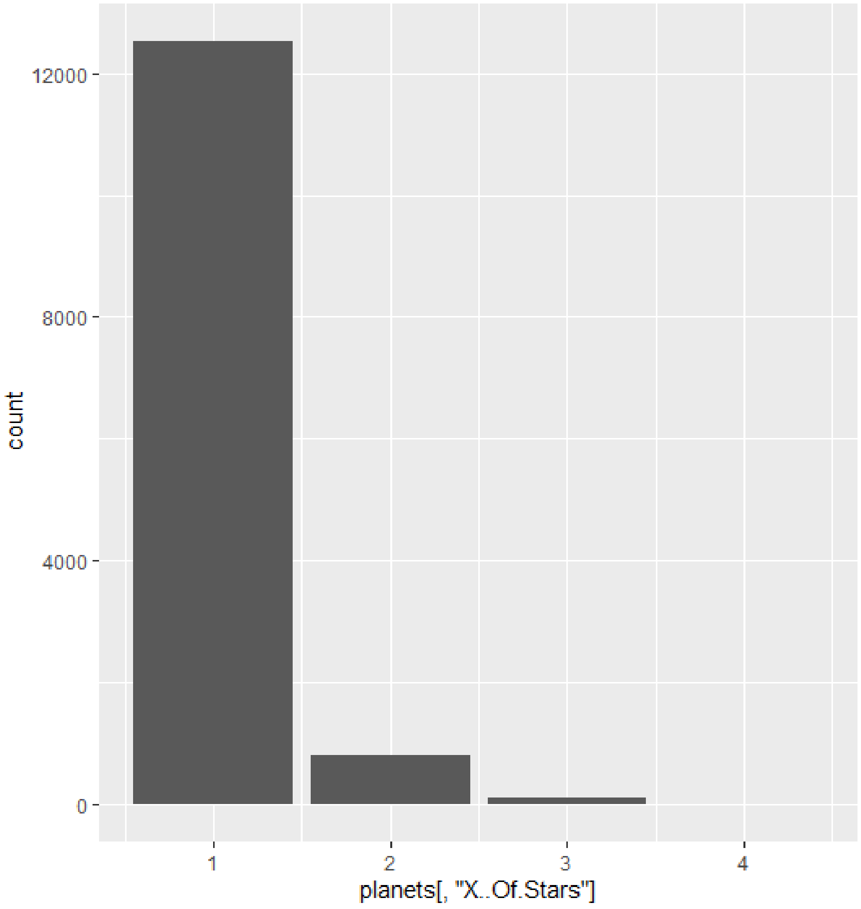


Figure 6

Figure 6 shows a histogram with a count of the number of stars in the system. Our count of solitary systems far outweighs that of Binary+ (for this catalogue). For hundreds of years astronomers have claimed that binary star systems were far more common than solitary star systems. Our closest companion Proxima Centauri is part of a binary star system. There is new information claiming to refute this belief that binaries are so common based on large numbers of red dwarfs that are undetectable, that we may in fact live in a universe with primary Solitary Systems. We simply do not know enough about the universe yet to be able to determine with what frequency binaries occur. Observations over the 20th century do in fact support the claim that up to 80% of star systems are binary (at minimum), however this data may be skewed as there are certain types of stars that are harder to detect. There is also modern scientific thought indicating that planet formation is easier around solitary stars, which provides one more way our data may be skewed. However, I still believe our model can be used practically to determine probabilistic functions for binary systems as our equation accounts for the number of planets, and other various factors.

Analysis:

Using the logistic regression model I have obtained from my research, I found the most effective model to use 7 variables plus the intercept value to predict the dependent variable. Here is the equation that I have concluded:

-1.5667125 + (-0.0061646 * Distance) + (0.2107421 * # of Planets) + (0.0004174 * Stellar Temperature in Kelvins) + (0.8431557 * Stellar Mass) + (0.0059899 * Stellar Density) + (-0.4663671 * Stellar Metallicity) + (-0.4190011 * Stellar Gravity)

This value will be applied to the inverse logit transformation to arrive at our predicted probability that the object belongs to a Binary+ system.

In addition, for hypothesis testing I created two models. One containing only the intercept and system distance, and one containing our more complicated model (7 additional variables). Our Ho: (Our basic model does as good as a job as the more complicated model) was rejected due to our P value (0) being lower than all reasonable Alpha values.

To provide a probability using this analysis, I present the example of an exoplanet whose host star has 1 planet, is 5000 degrees Kelvin, a Stellar mass the same as our Sun (1), Stellar Metallicity of .1, Stellar Gravity of 4.2, Stellar Density of 1, and System Distance of 500 (light years); to evaluate the predicted probability of being Binary+ to 3.53%. To interpret these coefficients further, controlling for all other independent variables to remain the same, if we increase the number of planets around the host star to 2 this bumps the predicted probability to 4.32%.

Our research also indicates an accuracy of 93.38%. This accuracy worries me a little bit as I may have an over-fitted model, or there is something I am missing in the data to suggest it is over-fitted. This accuracy can be increased to 93.82% by using a different cut off point using Information Value package.

Summary:

This project presented a few challenges. It was a large dataset with many columns. However, it was much less time consuming than I thought it would be to prep the data for insertion into R as a data frame. Creating a statistical model on raw data is an exciting thing, and we accomplished an accurate logistic regression model that lacks sensitivity.

At NASA and other places such as Tesla and SpaceX, where machine learning and statistical modeling are performed commonly, I can imagine there is much stress placed on the quality of the data one receives for a project such as this. Humans are not yet at the point where they have a singular reliable and completely filled database to catalogue all of our stars. When it comes to statistical analysis, one needs to have the proper data to start out with. Provided with a less biased data set filled with a very large sample size and filled features, I believe the predictive power of my algorithm would be much stronger. When cross-validating my algorithm with predictive data on the existence of binary+ stars, I ran into conflicting views. Some scientists think they are as common as 80% of all star systems , while others believe them far less common than that due to factors such as red dwarf detection and more. Either way, my data itself seems to be on the lower end of categorically Binary+ stars (as we have discussed).

Looking toward the future while considering the present: we are in an exciting time where many hours of hard work are going into creating this sort of database of knowledge. I imagine a day where we will be able to have an accurate, interactive map of our own galaxy and universal region with full descriptions of every feature imaginable. Perhaps this sort of topic will be included in general education sooner than we think!

Code Appendix:

```
# I took a lot of notes while I was coding. They are not necessarily intended to be vital part of
the report.

# I obtained my dataset from nasa archives for exoplanets

# I then noticed my excel file (csv) had an asymetrical list of all column names and descriptions

# I removed this list, and then put it into a word document

# Then I had to remove 283+ rows to get the data where it's just the headers with the data

# I will now go through the process of choosing which columns seem relevant to the project.

# Some Scientific knowledge is preferred here in choosing columns that could contribute to the
Probability of any chosen planet being part of a Binary Star System.

# I am not a physicist but I love this stuff, So I will choose the columns and do a little research
on the ones I'm not sure about to make the best selection I can

# There are 283 columns, so I'll take some time to go through them and delete ones that will
not provide statistical inference.

#After cleaning my CSV file I am left with 22 relevant column names, and 29387 rows

# After doing NA.omit for all these planets (sad) I am left with only 142 planets for my sample
size

# Notably, after handpicking my columns out of 289, being left with 22 columns some that are
not relevant to data, after dropping all NA values I am only left with 142

# Planets. So, I have done the task of making a trade off here. I want as much data as I can
possibly have, but I want ostensibly relevant, and pertinent data within

# the given context, with no NA values, but I don't want to have a small data set. So now I will
prioritize which columns have too many NA values and cause other good

# data to be lost. Unfortunately, one of my favorite columns Star Temperature and Metallicity
have so many NA values that I will have to get rid of them.

# Perhaps I will do a logistic regression on a smaller dataset, but for the project i will just find a
nice balance between column # and row #.

# After removing Stellar Luminosity, Planet Mass, Stellar Rotational Velocity, and Planet Density
I have more than twice as much rows at 340...However I would

# Like to still have more rows. Removing Planet to Stellar Radius Ratio leaves me with 526
Rows. I'm shooting for 1000 so let us look. Perfect. After removing
```

# Stellar Age, and Planetary Radius and a couple more column, I'm left with 13 thousand 391 rows to perform my Statistical Model.

# Planetary Orbital Period - 2758 NA Values

#Planet Radius - 8633 NA Values

#pLanet Mass 26863 NA Values

#Planet Density 27929 NA Values

#Planet Eq. Temperature 16377 NA Values

#Ratio of Planet to Stellar Radius 12885 NA Values

#Stellar Temp 1698 NA Values

#Stellar Radius NA 1580

#Stellar Mass NA 4430

#Stellar Metallicity 10687 NA Values

#Stellar Luminosity 21689 NA

#Stellar Gravity 5144 NA

#Stellar Age 23371 NA Values

#Stellar Density NA 12370

#Stellar Rotational Velocity NA 27555

#System Distance NA 712

#29387 Rows

#Section 1 – Importing Data

```
planets<-read.csv(header=TRUE,file="C:/Users/dc3pr/OneDrive/Desktop/Statistics Work/planets2.csv",sep=",")

detach(planets)

attach(planets)

names(planets)

dim(planets)
```

#Section 2 – Cleaning Data

```
planets <- subset (planets, select = -c(Stellar.Luminosity,Planet.Mass,Stellar.Rotational.Velocity,Planet.Density,Ratio.of.Planet.to.Stellar.Radius,Planetary.Orbital.Period,Planet.Radius,Planet.Equillibrium.Temperature,Stellar.Age))

names(planets)

dim(planets)

planets=na.omit(planets)

dim(planets)

#now I will drop columns that should be irrelevant statistically (planet name and so forth)

planets <- subset (planets, select = -c(ï..Index, Planet.Name,Host.Star.Name,Stellar.Metallicity.Ratio))

names(planets)

#I am left with 10 columns 1 being the dependent variable!

nrow(planets)

i=1

for(i in 1:nrow(planets)){

  if(planets[i,"X..Of.Stars"] >=2){

    planets[i,"X..Of.Stars"]=1

    i= i+1;

  }

  if(planets[i,"X..Of.Stars"] ==1){

    planets[i,"X..Of.Stars"]=0
    i= i+1;

  }

}

#there I have made binary+ is equal to 1 and if it is a single star system it has a value of 0

class(X..Of.Stars)
```

#Section 3 – Creating Logistic Regression Models

```
o1 = glm((X..Of.Stars)~X..Of.Planets,data=planets,family=binomial(link="logit"))

summary(o1)

#AIC = 6307.5


o2 = glm((X..Of.Stars)~Stellar.Effective.Temperature.Kelvins.,data=planets,family=binomial(link="logit"))

summary(o2)

#AIC 6425.6


o3 = glm((X..Of.Stars)~Stellar.Radius,data=planets,family=binomial(link="logit"))

summary(o3)

#AIC 6441.6


o4 = glm((X..Of.Stars)~st_mass,data=planets,family=binomial(link="logit"))

summary(o4)

#AIC = 6435.2


o5 = glm((X..Of.Stars)~Stellar.Metallicity,data=planets,family=binomial(link="logit"))

summary(o5)

#AIC =6437.4
```

```
o6 = glm((X..Of.Stars)~Stellar.Gravity,data=planets,family=binomial(link="logit"))

summary(o6)

#AIC = 6440.9



o7 = glm((X..Of.Stars)~Stellar.Density,data=planets,family=binomial(link="logit"))

summary(o7)

#AIC = 6435.7


o8 = glm((X..Of.Stars)~System.Distance,data=planets,family=binomial(link="logit"))

summary(o8)

#AIC = 4800.8
```

# Section 3 – Obtaining AIC of Each Model and Applying it to a Matrix

```
K=ncol(planets)-1

AICF=matrix(nrow=K,ncol=1)

sAICF=matrix(nrow=K,ncol=1)

droppedDecAic=matrix(nrow=K,ncol=1)

#dropped contains locations of columns we dropped from the model because they did not improve our model by decreasing AIC

# H0 states my intercept only model does as good of a job as my model that introduces more independent variables in fitting the dependent variable

# Ha states that my model does a better job

AICF[1]=o1$aic

AICF[2]=o2$aic

AICF[3]=o3$aic

AICF[4]=o4$aic

AICF[5]=o5$aic

AICF[6]=o6$aic

AICF[7]=o7$aic
```

```
AICF[8]=o8$aic

ord=order(AICF)

max(planets[,"X..Of.Stars"])

min(AICF)
```

# Section 4 – Running some Initial Hypothesis Tests and Comparisons

```
#hypothesis test 1 asks is my model better than just using an intercept value.

# I'm going to create my own model based off what I would think would provide a better fit

o9= glm((X..Of.Stars)~Stellar.Metallicity+Stellar.Effective.Temperature.Kelvins.,data=planets,family=binomial(link="logit"))

summary(o9)

# lets go ahead and compare this just to number of stars based off number of planets

summary(o1)


1-pchisq(6303.5-6415.7,13389-13388)


#unfortunately here I will have a value of 1, it appears that the model I selected off the top of my head  does not improve the model from using only number of planets and I fail to reject my H0.

#let's try one more model comparison I select

o10 = glm((X..Of.Stars)~X..Of.Planets+st_mass,data=planets,family=binomial(link="logit"))

summary(o10)

## let us compare using number of planets and stellar mass to only using number of planets

summary(o1)

1-pchisq(6303.5-6288.0,13389-13388)

#this calculation gives me an astronomically small p value, so it is safe to say for all normal alpha values I would be able to reject my null hypothesis that
```

```
#my model does as good as a job as the intercept value. My more complicated model, in this case is a success and provides a better fit.

# I will run a short calculation of probability for some given values for my model. It is later discovered, that this model is not entirely unlike the one that the

#step.model found, since in fact they both use # of planets and also stellar mass.

#can you provide the probability that an exoplanet belongs to a Binary+ star system given these values for your independent variables

# 4 planets in the system, and a stellar mass of 2.0 (twice the size of our sun)

summary(o10)

#calculate probability

val=-3.79011 + .30365*4 + .49793*2.0

probability2xSunMassAnd4Planets=1/(1+exp(-1*val))

probability2xSunMassAnd4Planets

#COOL! We obtained a probability of .1708451 that the star system is binary+ considering these values for independent variables.

# What is the effect of the probability if we increase the exoplanets by 1 controlling for all other variables remain the same

val2=-3.79011 + .30365*5 + .49793*2.0

probability2xSunMassAnd5Planets=1/(1+exp(-1*val2))

probability2xSunMassAnd5Planets

# This in fact increases our chances of having a binary+ star system by .04738683 controlling for all other variables

difference = probability2xSunMassAnd5Planets - probability2xSunMassAnd4Planets

difference
```

#Section 5 – Forward Model Building Approach

```
#here I will try to use the forward model building approach by building a model for each sequential variable in the order contained within my ord variable

o11 = glm((X..Of.Stars)~System.Distance,data=planets,family=binomial(link="logit"))

summary(o11)

o12 =
glm((X..Of.Stars)~System.Distance+X..Of.Planets,data=planets,family=binomial(link="logit"))
```

```
summary(o12)

o13 = glm((X..Of.Stars)~System.Distance+X..Of.Planets+Stellar.Effective.Temperature.Kelvins.,data=planets,family=binomial(link="logit"))

summary(o13)

o14 = glm((X..Of.Stars)~System.Distance+X..Of.Planets+Stellar.Effective.Temperature.Kelvins.+st_mass,data=planets,family=binomial(link="logit"))

summary(o14)

o15 = glm((X..Of.Stars)~System.Distance+X..Of.Planets+Stellar.Effective.Temperature.Kelvins.+st_mass+Stellar.Density,data=planets,family=binomial(link="logit"))

summary(o15)

o16 = glm((X..Of.Stars)~System.Distance+X..Of.Planets+Stellar.Effective.Temperature.Kelvins.+st_mass+Stellar.Density+Stellar.Metallicity,data=planets,family=binomial(link="logit"))

summary(o16)

o17 = glm((X..Of.Stars)~System.Distance+X..Of.Planets+Stellar.Effective.Temperature.Kelvins.+st_mass+Stellar.Density+Stellar.Metallicity+Stellar.Gravity,data=planets,family=binomial(link="logit"))

summary(o17)

o18 = glm((X..Of.Stars)~System.Distance+X..Of.Planets+Stellar.Effective.Temperature.Kelvins.+st_mass+Stellar.Density+Stellar.Metallicity+Stellar.Gravity+Stellar.Radius,data=planets,family=binomial(link="logit"))

summary(o18)

#After some testing, I have found that the addition of the Stellar Radius variable does in fact raise the AIC, so one would presume from a forward model building approach to implement the

#o17 model, which includes, System Distance, # of planets, STellar Temperature, Stellar Mass, Stellar Density, Stellar Metallicity, and Stellar Gravity in our model

summary(o17)
```

```
}
```

# Section 6 – Creating Model using stepAIC function from the MASS library

```
# now lets make a model with the stepAIC function from the mass library and see how this will effect our aic value

full.modelAprime <- glm(X..Of.Stars ~., data=planets,family=binomial(link="logit"), maxit=100)

library(MASS)

step.model <- stepAIC(full.modelAprime, direction = "both", trace = TRUE)

summary(step.model)
```

# Section 7 - Hypothesis Testing, P Values, And Coefficient Estimates, Probability from Inverse Logit Transformation

```
#this is infact the same AIC value as the model that we built using the variation on the forward model buliding approach.

#Since this is our best model, I will do a quick hypothesis test and some question and answer

#H0 says that just using the intercept with system distance does as good as a job as our model and Ha says that our more complicated model does a better job.

#Calculate P value and select a reasonable alpha. I will select .05

plainModel = o11

summary(plainModel)

summary(step.model)

pval = 1-pchisq(4796.8-4544.9,13389-13383)

pval

if(pval <= .05)

{

  print("We get to reject our H0 that using our basic model does as good of a job as our more complicated model!")

}else{

  print("We fail to reject our H0 that our basic model does as good of a job as our more complicated model!")

}
```

```
#in fact for any reasonable alpha value we have rejected our H0 since the p value was 0.



# now I will do a calculation when controlling for all other variables using my model

summary(step.model)

#calculate probability for 1 planet, 5000 degrees kelvin, Stellar Mass of 1, Stellar Metallicity of .1, Stellar Gravity of 4.2, Stellar Density of 1, System Distance of 500

val3=-1.5667125 + .2107421*1+.0004174*5000+.8431557*1+-.4663671*.1+-.4190011*4.2+.0059899*1+-.0061646*500

prob1Planet5000DegreesMass1MetalGravDensDist=1/(1+exp(-1*val3))

prob1Planet5000DegreesMass1MetalGravDensDist

#Therefore we can conclude that for independent variables of these values we may predict a %3.527849 probability that this exoplanet belongs to a binary+star system

#what would happen if we increase the planets around the star to 2 controlling for all other variables.

val4=-1.5667125 + .2107421*2+.0004174*5000+.8431557*1+-.4663671*.1+-.4190011*4.2+.0059899*1+-.0061646*500

prob2Planet5000DegreesMass1MetalGravDensDist=1/(1+exp(-1*val4))

prob2Planet5000DegreesMass1MetalGravDensDist

# by introducing an additional 1 value to number of planets, controlling for all other variables we have increased the probability of the exoplanet belonging to

# a binary + star system to 4.31971%


#I will do one more entry into our model using the averages of each of these values

stmassmean = mean(planets[,"st_mass"])

grvmean = mean(planets[,"Stellar.Gravity"])

densmean = mean(planets[,"Stellar.Density"])

metalmean = mean(planets[,"Stellar.Metallicity"])

distmean = mean(planets[,"System.Distance"])

planetmean = mean(planets[,"X..Of.Planets"])
```

```
tempmean = mean(planets[,"Stellar.Effective.Temperature.Kelvins."])

radiusmean = mean(planets[,"Stellar.Radius"])

# here it is

val5=-1.5667125 + .2107421*planetmean+.0004174*tempmean+.8431557*stmassmean+-.4663671*metalmean+-.4190011*grvmean+.0059899*densmean+-.0061646*distmean

probBinaryUsingAverages=1/(1+exp(-1*val5))

probBinaryUsingAverages
```

Section 8 – Accuracy and Sensitivity Calculations

```
#lets try to calculate accuracy and some other things

optimapred=(predict(o17,type="response")>0.5)*1



#accuracy

accuracy=(sum((optimapred==1 & X..Of.Stars==1))+sum((optimapred==0 & X..Of.Stars==0)))/length(X..Of.Stars)

accuracy

#.93383615861


#sensitivity

sensitivity= sum((optimapred==1 & X..Of.Stars==1))/sum(X..Of.Stars==1)

sensitivity

library(InformationValue)

optCutOff <- optimalCutoff(X..Of.Stars, predict(o17, type="response"))


predicted=(predict(o17,type="response")>optCutOff)*1
```

```
#accuracy using a different cutoff
accuracy2=(sum((predicted==1 & X..Of.Stars==1))+sum((predicted==0 & X..Of.Stars==0)))/length(X..Of.Stars)
accuracy2
#some ggplot
library(tidyverse)
names(planets)


ggplot(data = planets) +
  geom_point(mapping = aes(x = planets[,"Stellar.Effective.Temperature.Kelvins."], y = planets[,"System.Distance"],color=planets[,"X..Of.Stars"]))


ggplot(data = planets) +
  geom_point(mapping = aes(x = planets[,"X..Of.Planets"], y = planets[,"st_mass"], color=planets[,"X..Of.Stars"]))


#at this point, I have recalculated the dataframe without the loop (lines 1-52 so I can show the reader how is it that the number of stars in systems appears in
#this dataset without any manipulation)
ggplot(data = planets) +
  geom_point(mapping = aes(x = planets[,"X..Of.Stars"], y = planets[,"System.Distance"], color=planets[,"st_mass"]))


ggplot(data = planets) +
  geom_bar(mapping = aes(x = planets[,"X..Of.Stars"]))
ggplot(data = planets) +
  geom_bar(mapping = aes(x = planets[,"X..Of.Stars"], colour= planets[,"X..Of.Stars"]))
```