

# Exoplanetas

Constanza de Galvagni, Leonel Saire Choque, Agustin Passalacqua

17/03/2023

El trabajo se lleva a cabo a partir de un dataset de exoplanetas realizado por la NASA y descargado desde la pagina web Kaggle.

Un exoplaneta se define como un cuerpo celeste que no emite luz propia y orbita una estrella fuera de nuestro sistema solar.

Algunas de las unidades y medidas utilizadas para llevar a cabo el analisis son las siguientes:

*Astronomical Unit (AU): mide distancias entre planetas y 1 AU equivale a  $1,49 \times 10^8$  km.*

*Magnitud Estelar (Mag): mide el brillo de un cuerpo celeste y depende de la distancia. A menor mag mayor el brillo.*

*Excentricidad: determina que tan circular es la orbita de un planeta siendo 0 totalmente circular. Crece si la orbita es eliptica.*

*Años luz: representa la distancia que recorre la luz cuando viaja durante un año.*

Cargamos el dataset e importamos las librerias a utilizar

```
data<-read.csv("exoplanetas.csv")
library(dplyr)

## 
## Attaching package: 'dplyr'

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

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

library(ggplot2)
library(gridExtra)

## 
## Attaching package: 'gridExtra'

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

```

library(rgl)
library(viridis)

## Loading required package: viridisLite

library(RColorBrewer)
library(class)
library(caret)

## Loading required package: lattice

library(constants)

## Package 'errors' not found. Constants with uncertainty ('syms_with_errors') not available.

## Package 'units' not found. Constants with units ('syms_with_units') not available.

## Package 'quantities' not found. Constants with uncertainty+units ('syms_with_quantities') not available

```

## Filtrado de datos

Eliminamos NA's y pasamos character a factor donde corresponde

```

data <- data %>%
  dplyr::select(name, distance, stellar_magnitude, planet_type, discovery_year, mass_multiplier, mass_wt)
  na.omit()

data[, (4)]<-as.factor(data[, 4])
data[, (7)]<-as.factor(data[, 7])
data[, (9)]<-as.factor(data[, 9])
data[, (13)]<-as.factor(data[, 13])

```

Unificamos las columnas de mass multiplier y mass wrt en una sola: "masa"

```

masa<-c()
for (i in 1:nrow(data)){
  if (data[i, 7]=="Earth"){
    masa<-c(masa, (data[i, 6]*(5.97 * 10^24)))
  }
  else{
    masa<-c(masa, (data[i, 6]*(1.9 * 10^27)))
  }
}

data<-cbind(data, masa)

```

Unificamos las columnas de radius multiplier y radius wrt en una sola: "radio"

```

radio<-c()
for (i in 1:nrow(data)){
  if (data[i, 9]=="Earth"){
    radio<-c(radio, (data[i, 8]*6371))
  }
  else{
    radio<-c(radio, (data[i, 8]*69911))
  }
}

data<-cbind(data, radio)

```

Nos deshacemos de las columnas de multiplier y wrt que ya no nos sirven

```

data <- data %>%
  dplyr::select(name, distance, stellar_magnitude, planet_type, discovery_year, orbital_radius, orbital_

```

Calculamos la gravedad y la añadimos como columna

```

gravedades <- c()
G <- codata[191, 4] # $m^3/kg*s^2$ 
for (i in 1:nrow(data)) {
  gravedades <- c(gravedades, (G*data$masa[i]) / ((data$radio [i]*1000)**2))
}

data <- cbind(data, gravedades)

```

Nos deshacemos de valores de excentricidad carentes de sentido

```

data <- data[data$eccentricity>=0,]

```

Creamos una secuencia de colores para asignar de ahora en adelante un color por tipo de planeta

```

colores_por_tipo<-c('lightpink2', 'lightgreen', 'cadetblue2', 'mediumorchid1')

```

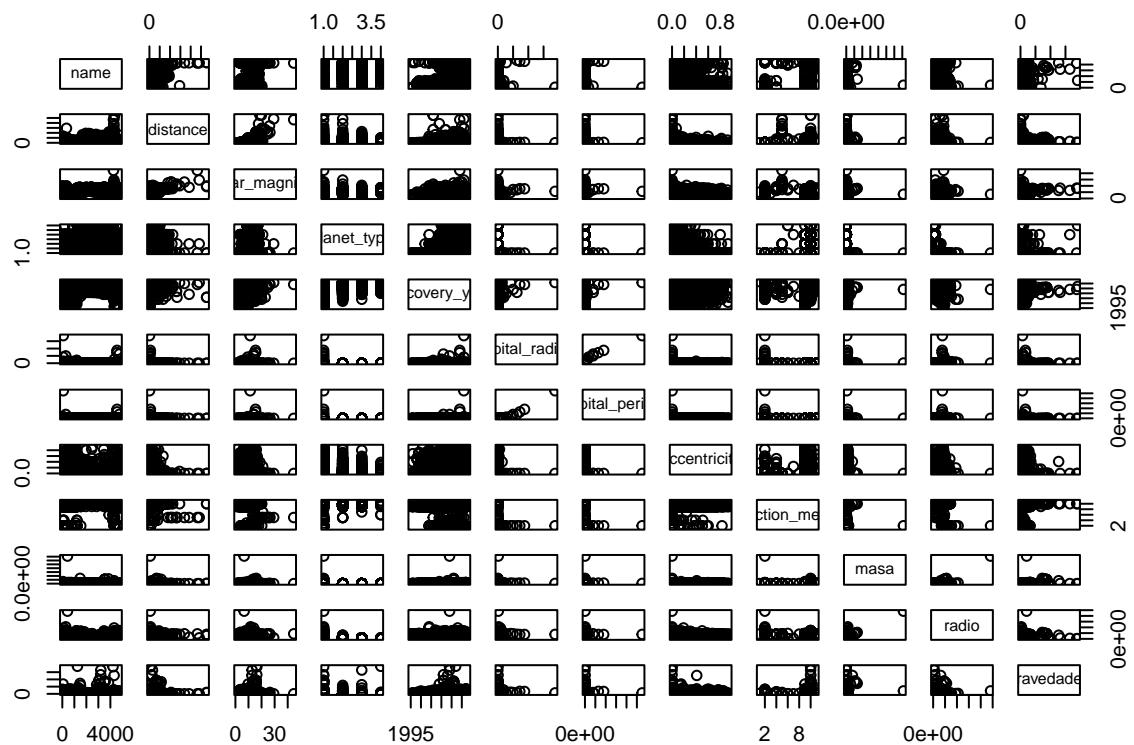
## Analisis descriptivo

Vemos las relaciones entre las variables en un solo grafico

```

plot(data)

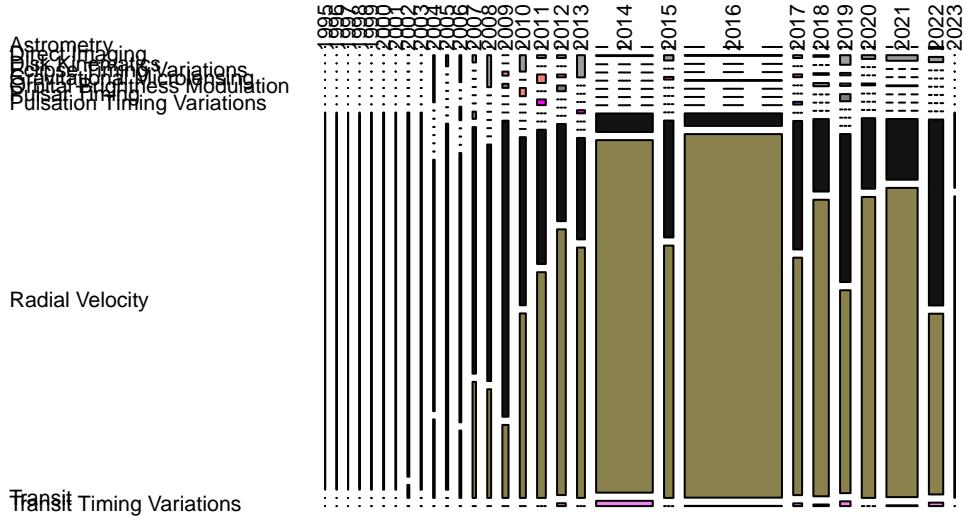
```



Relacionamos en el siguiente grafico el año de descubrimiento y el metodo. Radial velocity es el metodo mas utilizado hasta 2006 y luego crece en popularidad el metodo Transit. Hay un pico de descubrimiento de exoplanetas en 2016. A la derecha vemos el gráfico acotado a las barras de mayor área.

```
mosaicplot(table(data$discovery_year,data$detection_method),col=sample(x=colors(),size=length(unique(da
```

## METODO Y AÑO DE DESCUBRIMIENTO



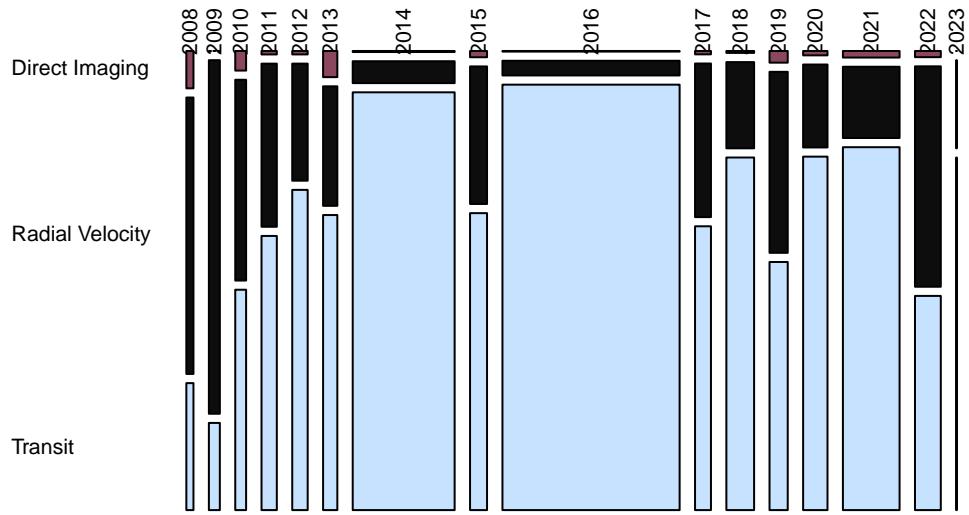
```
aniosAcotados <- data$discovery_year >= 2008
dataAcotadoMA <- data[aniosAcotados,]

metodosMasDatos <- dataAcotadoMA$detection_method == "Transit" | dataAcotadoMA$detection_method == "Radial Velocity"
dataAcotadoMA <- dataAcotadoMA[metodosMasDatos,]

dataAcotadoMA$detection_method <- droplevels(dataAcotadoMA$detection_method)

mosaicplot(table(dataAcotadoMA$discovery_year,dataAcotadoMA$detection_method),col=sample(x=colors(),size=3))
```

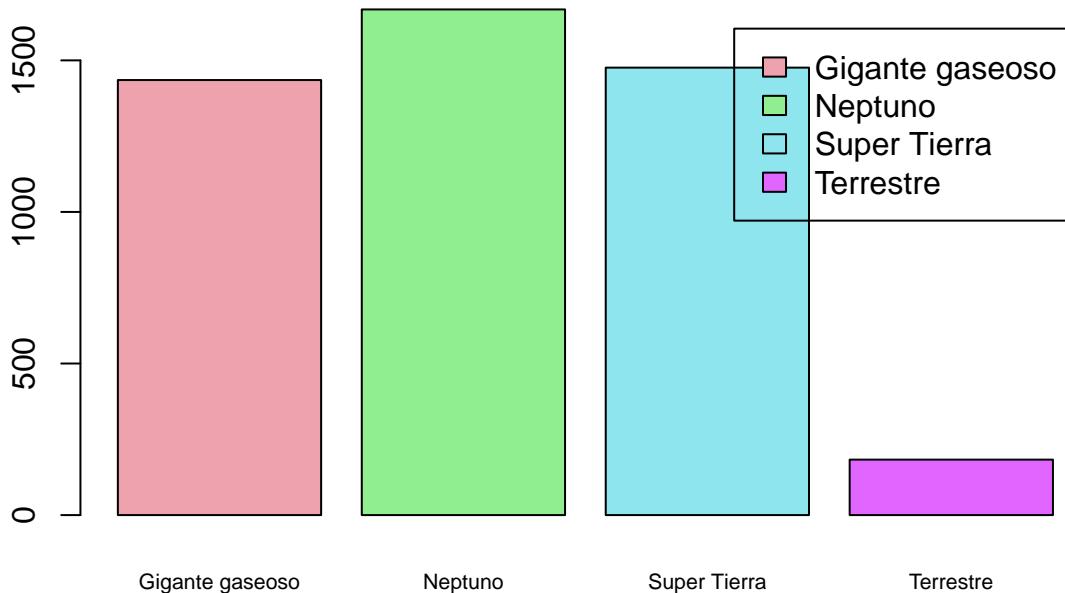
## METODO Y AÑO DE DESCUBRIMIENTO



Vemos en el siguiente grafico las distribucion de nuestra muestra por tipo de planeta. Tenemos una cantidad similar de gigantes gaseosos, planetas tipo neptunos y super tierras. La cantidad de exoplanetas de tipo terrestre es muy inferior en relacion a los demas tipos.

```
barplot(table(data$planet_type), legend=c('Gigante gaseoso', 'Neptuno', 'Super Tierra', 'Terrestre'), col=
```

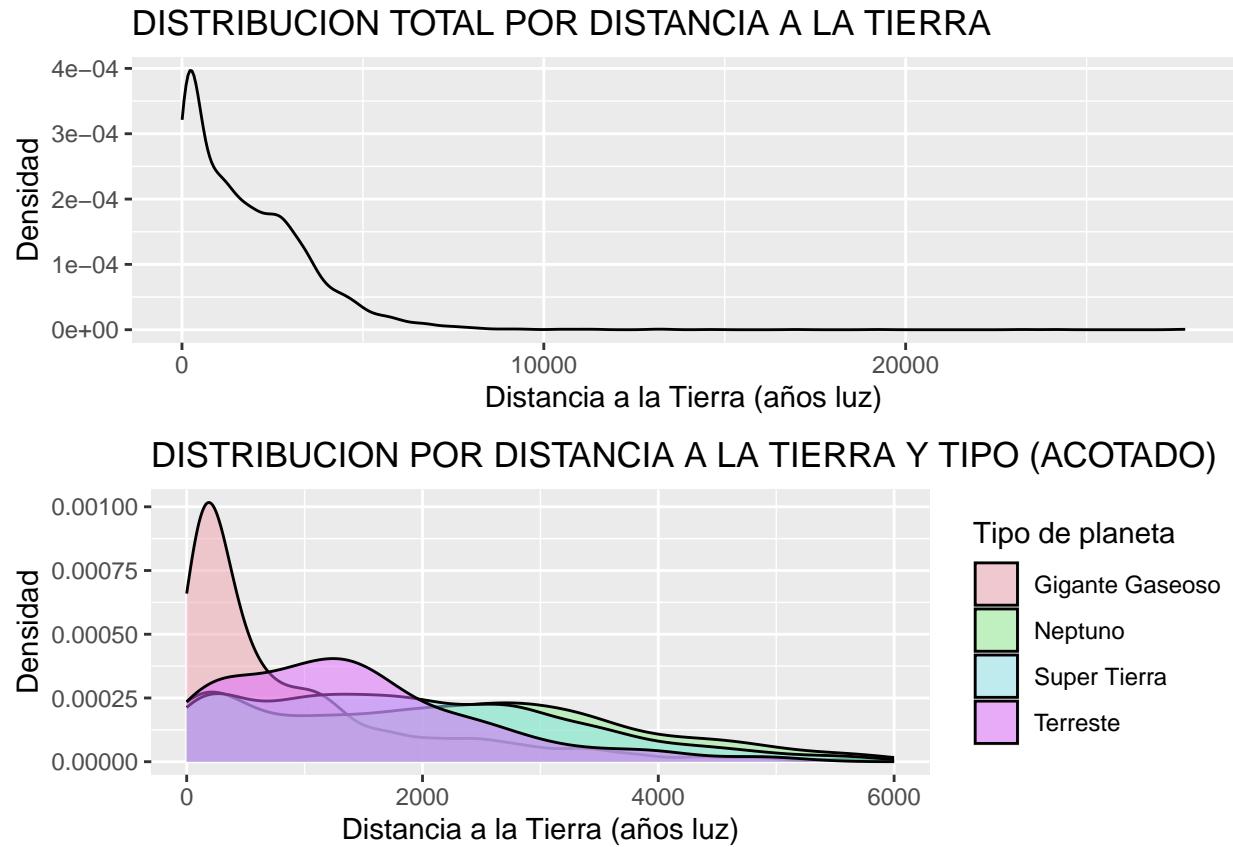
## Cantidad de planetas por tipo



Graficamos la distribucion por tipo de planeta y distancia a la Tierra. Observamos una corta tendencia creciente que alcanza rapidamente un pico y disminuye. Acotamos los datos hasta 6000 años luz para observar claramente la distribucion por tipo. Se observa una fuerte acumulacion de gigantes gaseosos hasta los 1000 años luz de distancia. La acumulacion es un poco mas leve para los exoplanetas de tipo terrestre hasta los 2000 años luz. Super Tierras y Neptunos muestran tambien una distribucion decreciente con la distancia, aunque mas uniforme que los anteriores.

```
grafdist1<-ggplot(data, aes(x=distance)) +  
  geom_density(alpha=0.5) +  
  ylab("Densidad") +  
  xlab("Distancia a la Tierra (años luz)")+  
  ggtitle("DISTRIBUCION TOTAL POR DISTANCIA A LA TIERRA")  
  
grafdist2<-ggplot(data, aes(x=distance, fill=planet_type)) +  
  geom_density(alpha=0.5) +  
  scale_fill_viridis(discrete=TRUE) +  
  ylab("Densidad") +  
  xlab("Distancia a la Tierra (años luz)")+  
  ggtitle("DISTRIBUCION POR DISTANCIA A LA TIERRA Y TIPO (ACOTADO)")+  
  scale_fill_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "  
  xlim(c(0, 6000))  
  
## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.
```

```
grid.arrange(grafdist1, grafdist2, ncol=1, nrow =2)
```



Graficamos la distribucion de nuestra muestra por tipo de planeta y brillo. Acotamos dentro de los valores mas representativos. Observamos que la mayoria tiene picos muy marcados a excepcion de Gigantes gaseosos donde se observa un maximo absoluto alrededor de 8 mag y un maximo relativo alrededor de 12 mag. En general, los astros mas brillantes son gigantes gaseosos, seguidos de terrestres, super tierras y neptunos.

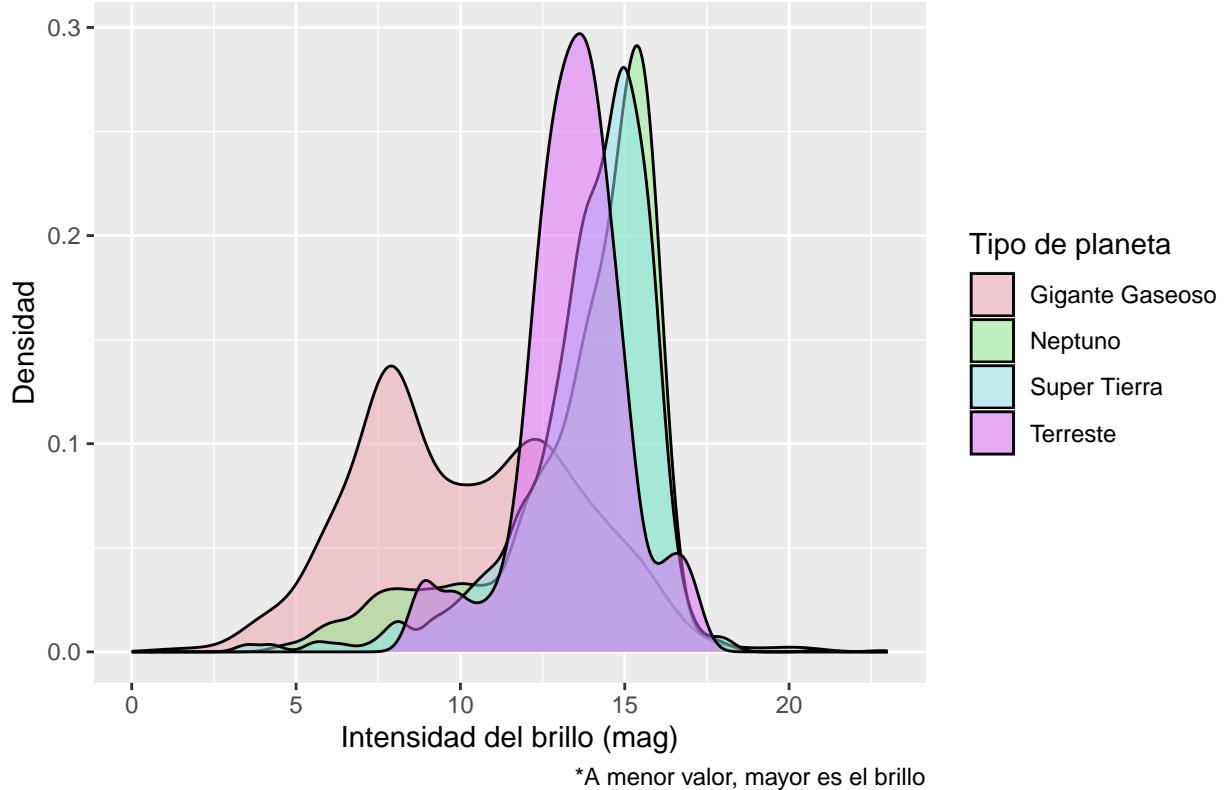
```
ggplot(data, aes(x=stellar_magnitude, fill=planet_type)) +
  geom_density(alpha=0.5) +
  scale_fill_viridis(discrete=TRUE) +
  ylab("Densidad") +
  xlab("Intensidad del brillo (mag)")+
  ggtitle("DISTRIBUCION POR BRILLO Y TIPO DE PLANETA (ACOTADO)")+
  scale_fill_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', 'Terreste'))+
  xlim(c(0, 23))+
```

**Labs**

```
  labs(caption = "*A menor valor, mayor es el brillo")
```

## Scale for fill is already present.  
 ## Adding another scale for fill, which will replace the existing scale.

## DISTRIBUCION POR BRILLO Y TIPO DE PLANETA (ACOTADO)



Observamos tambien la relacion que existe entre el brillo y la distancia. Naturalmente, el brillo decrece a medida que crece la distancia a la Tierra. La tendencia es mas notoria para gigantes gaseosos, neptunos y super Tierras. En el segundo grafico aproximamos la tendencia con una recta para cada tipo.

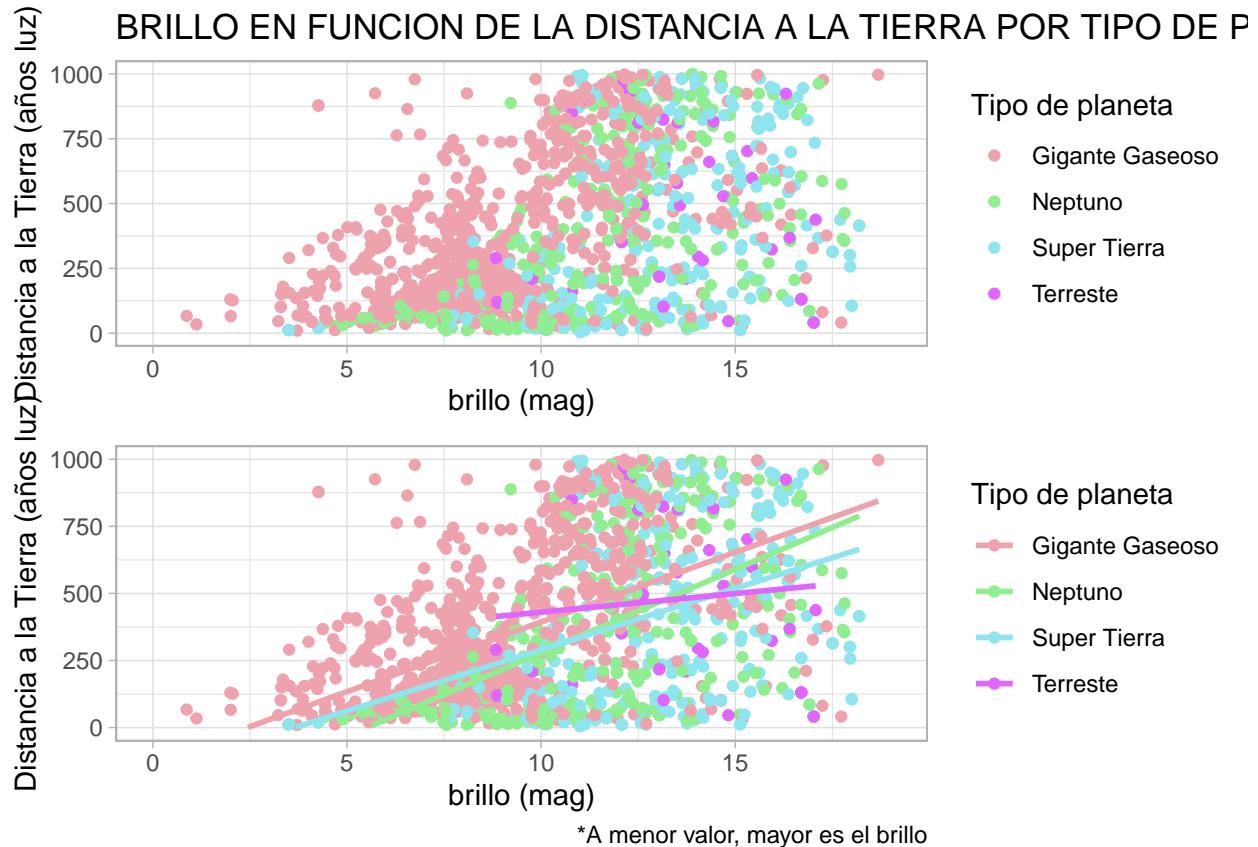
```
brillo_por_distancia_ajus<-ggplot(data, aes(x=stellar_magnitude, y=distance, color=planet_type)) +
  geom_jitter(alpha=1) +
  geom_smooth(method="lm", se=FALSE) +
  xlab("brillo (mag)") +
  ylab("Distancia a la Tierra (años luz)")+
  scale_color_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno',      'Super Tierra',
  theme_light()+
  ylim(c(0, 1000))+ 
  xlim(c(0, 19))+ 
  labs(caption = "*A menor valor, mayor es el brillo")
```

```
brillo_por_distancia<-ggplot(data, aes(x=stellar_magnitude, y=distance, color=planet_type))+ 
  geom_jitter(alpha=1) +
  xlab("brillo (mag)") +
  ylab("Distancia a la Tierra (años luz)")+
  ggtitle("BRILLO EN FUNCION DE LA DISTANCIA A LA TIERRA POR TIPO DE PLANETA")+
  scale_color_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra',
  theme_light()+
  ylim(c(0, 1000))+ 
  xlim(c(0, 19))
```

```
grid.arrange(brillo_por_distancia, brillo_por_distancia_ajus, nrow=2)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Observamos la masa por tipo de planeta. Se puede observar que la masa es muy variable segun el tipo de planeta e incluso dentro de cada tipo. Se ppuede ver en el primer grafico valores de Gigantes Gaseosos que impiden apreciar la relacion entre los tipos. Incluso eliminando los outliers y eliminando el 10% mas grande y el 10% mas chico (segundo grafico) existe una diferencia abismal entre gigantes gaseosos y el resto de los tipos de planetas. Solo cuando eliminamos Gigantes gaseosos y outliers vemos que la masa de los planetas tipo Neptuno es superior a las demas y que si bien los planetas Terrestres son mas masivos que los de tipo Super Tierra, algunos planetas de ambos tipos tienen masas similares.

```
graf_masa<-ggplot(data, aes(y=masa, x=planet_type)) +
  geom_boxplot(alpha=1) +
  ylab("masa(kg)") +
  xlab("Tipo de planeta")+
  ggtitle("MASA POR TIPO")

graf_masa1<-ggplot(data, aes(y=masa, x=planet_type)) +
  geom_boxplot(outlier.shape = NA) +
  scale_y_continuous(limits = quantile(data$masa, c(0.1, 0.9)))+
  geom_boxplot(alpha=1) +
  ylab("masa(kg)") +
  xlab("Tipo de planeta")+
  ggtitle("MASA POR TIPO (SIN OUTLIERS)")
```

```

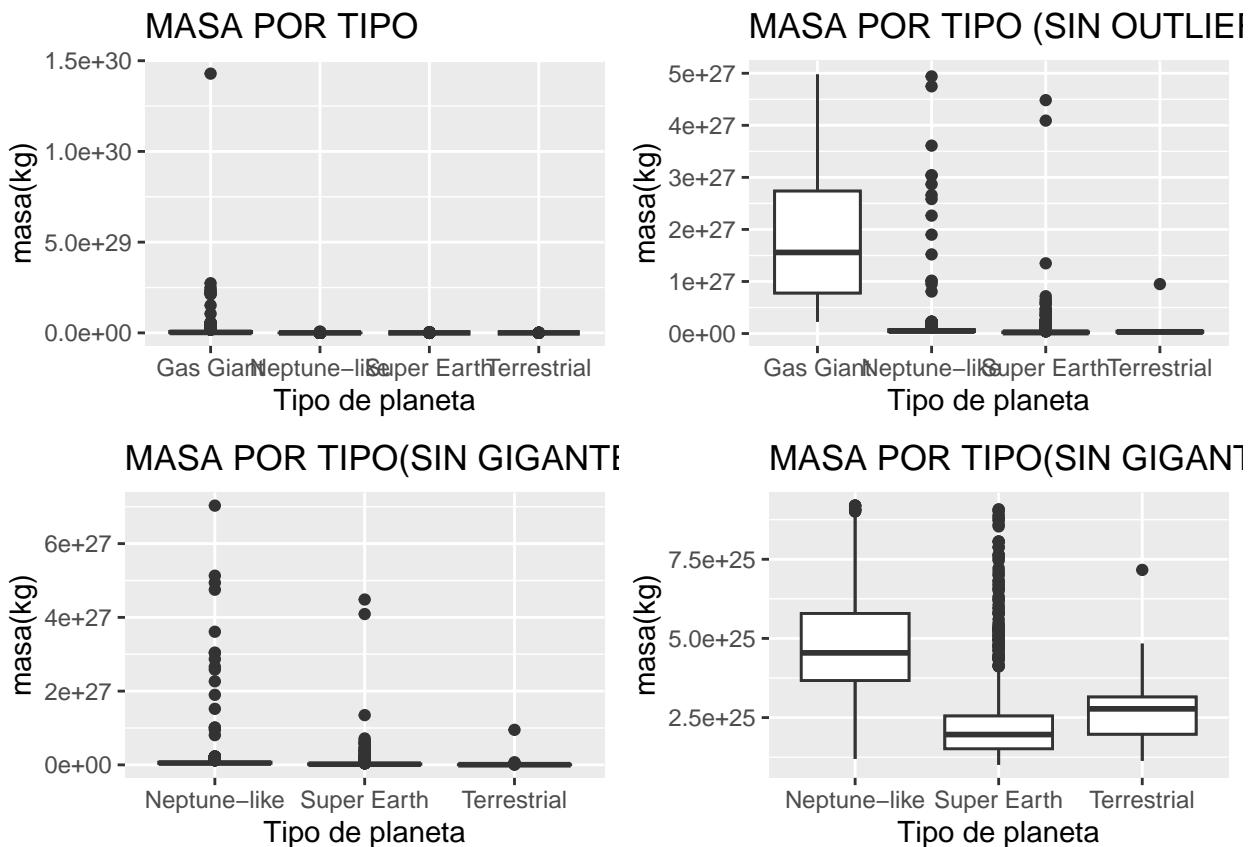
no_gas_giant <- c('Neptune-like', 'Super Earth', 'Terrestrial')
sin_gigantes_gaseosos <- data.frame(data[data$planet_type %in% no_gas_giant,])

graf_masa2<-ggplot(sin_gigantes_gaseosos, aes(y=masa, x=planet_type)) +
  geom_boxplot(alpha=1) +
  ylab("masa(kg)") +
  xlab("Tipo de planeta")+
  ggtitle("MASA POR TIPO(SIN GIGANTES GASEOSOS)")

graf_masa3<-ggplot(sin_gigantes_gaseosos, aes(y=masa, x=planet_type)) +
  geom_boxplot(outlier.shape = NA) +
  scale_y_continuous(limits = quantile(sin_gigantes_gaseosos$masa, c(0.1, 0.9)))+
  geom_boxplot(alpha=1) +
  ylab("masa(kg)") +
  xlab("Tipo de planeta")+
  ggtitle("MASA POR TIPO(SIN GIGANTES GASEOSOS)")

grid.arrange(graf_masa, graf_masa1, graf_masa2, graf_masa3, nrow=2, ncol=2)

```



Con respecto al radio, nuevamente observamos que el radio de los Gigantes gaseosos supera ampliamente al resto. Sacando los outliers, el 10% menor de cada tipo, el 10% mayor de cada tipo y los Gigantes gaseosos del grafico, se observa que la cantidad mas representativa de terrestres presenta un radio similar a la cantidad mas representativa de neptunos, mientras que la cantidad mas representativa de super Tierras esta por debajo

del resto.

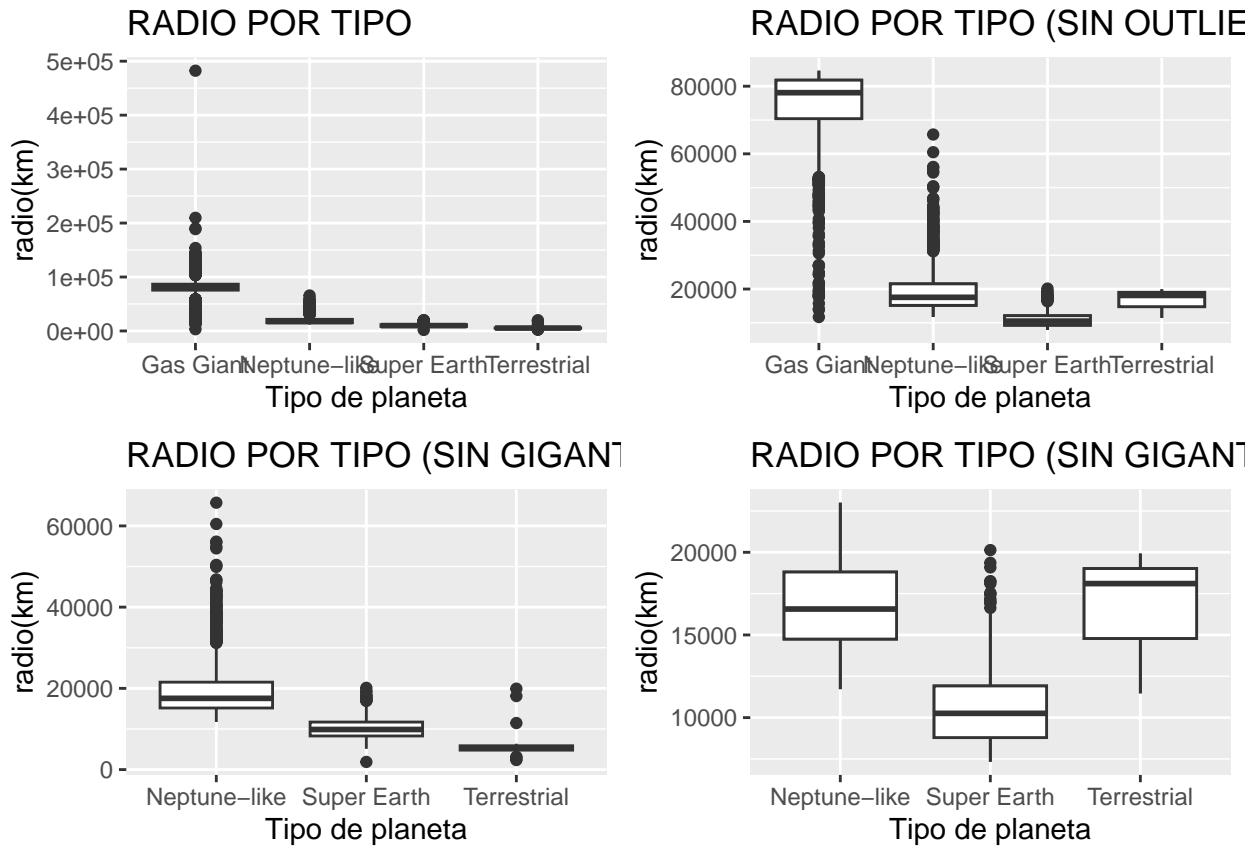
```
graf_radio<-ggplot(data, aes(y=radio, x=planet_type)) +
  geom_boxplot(alpha=1) +
  ylab("radio(km)") +
  xlab("Tipo de planeta")+
  ggtitle("RADIO POR TIPO")

graf_radio1<-ggplot(data, aes(y=radio, x=planet_type)) +
  geom_boxplot(alpha=1) +
  geom_boxplot(outlier.shape = NA) +
  scale_y_continuous(limits = quantile(data$radio, c(0.1, 0.9)))+
  ylab("radio(km)") +
  xlab("Tipo de planeta")+
  ggtitle("RADIO POR TIPO (SIN OUTLIERS)")

graf_radio2 <- ggplot(sin_gigantes_gaseosos, aes(y=radio, x=planet_type)) +
  geom_boxplot(alpha=1) +
  ylab("radio(km)") +
  xlab("Tipo de planeta")+
  ggtitle("RADIO POR TIPO (SIN GIGANTES GASEOSOS)")

graf_radio3<-ggplot(sin_gigantes_gaseosos, aes(y=radio, x=planet_type)) +
  geom_boxplot(outlier.shape = NA) +
  scale_y_continuous(limits = quantile(sin_gigantes_gaseosos$radio, c(0.1, 0.9)))+
  geom_boxplot(alpha=1) +
  ylab("radio(km)") +
  xlab("Tipo de planeta")+
  ggtitle("RADIO POR TIPO (SIN GIGANTES GASEOSOS)")

grid.arrange(graf_radio, graf_radio1, graf_radio2, graf_radio3, nrow=2, ncol=2)
```



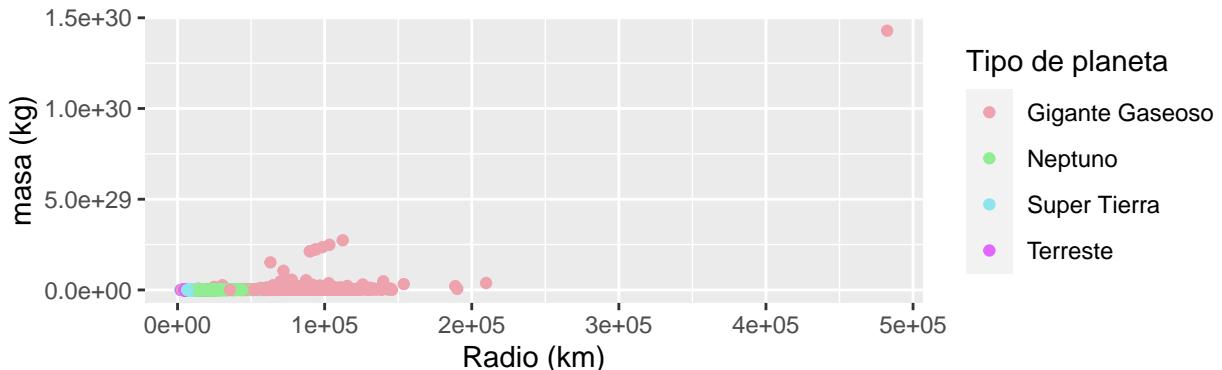
En los siguientes dos graficos se puede ver que cada tipo de planeta se ubica en un rango de radios particulares, mientras que la mayoria se acumula en un rango de masa acotado indiferentemente de su tipo. Existe un pequeno intervalo de radio para el cual los gigantes gaseosos presentan una gran variabilidad de masa.

```
graf_radioymasa<-ggplot(data, aes(x=radio, y=masa, color=planet_type)) +
  geom_jitter(alpha=1) +
  ylab("masa (kg)") +
  xlab("Radio (km)")+
  ggtitle("MASA VS. RADIO SEGUN TIPO DE PLANETA")+
  scale_color_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', 'Terrestre', 'Neptuno-Like'))

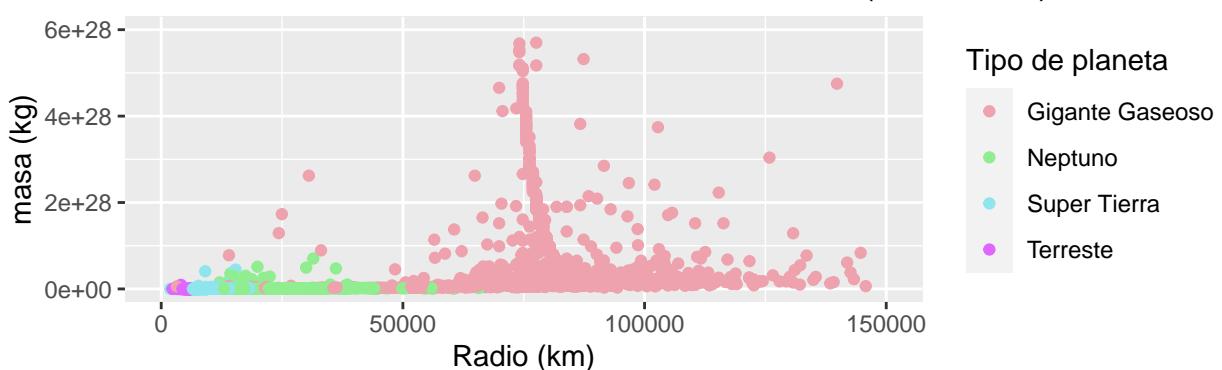
graf_radioymasa_ac<-ggplot(data, aes(x=radio, y=masa, color=planet_type)) +
  geom_jitter(alpha=1) +
  ylab("masa (kg)") +
  xlab("Radio (km)")+
  ggtitle("MASA VS. RADIO SEGUN TIPO DE PLANETA (ACOTADO)")+
  scale_color_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', 'Terrestre', 'Neptuno-Like'))
  xlim(c(0, 1.5 * 10^05))+ 
  ylim(c(0, 6 * 10^28))

grid.arrange(graf_radioymasa,graf_radioymasa_ac,nrow =2)
```

## MASA VS. RADIO SEGUN TIPO DE PLANETA



## MASA VS. RADIO SEGUN TIPO DE PLANETA (ACOTADO)



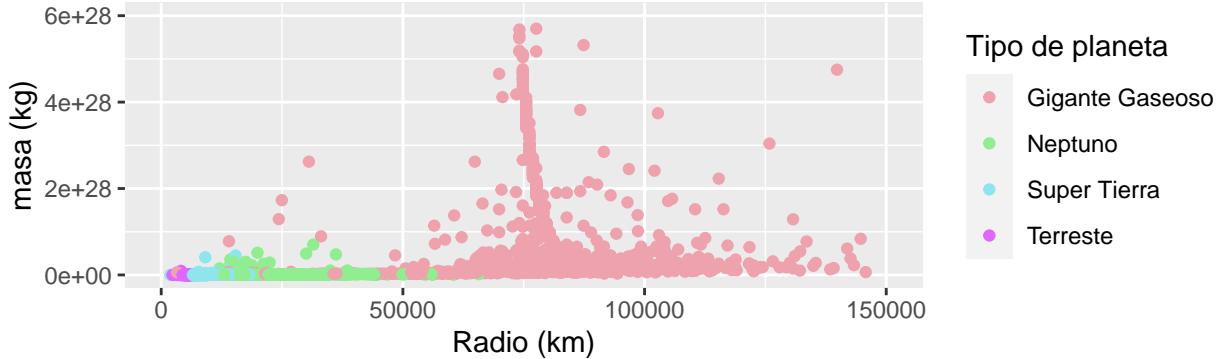
Intentamos explicar este comportamiento en gigantes gaseosos por el método de descubrimiento:

```
dataTresMetodos <- data[data$detection_method == "Transit" | data$detection_method == "Radial Velocity"]
masaTresMetodos <- dataTresMetodos$masa
radioTresMetodos <- dataTresMetodos$radio
dataTresMetodos$detection_method <- droplevels(dataTresMetodos$detection_method)
tres_metodos <- dataTresMetodos$detection_method

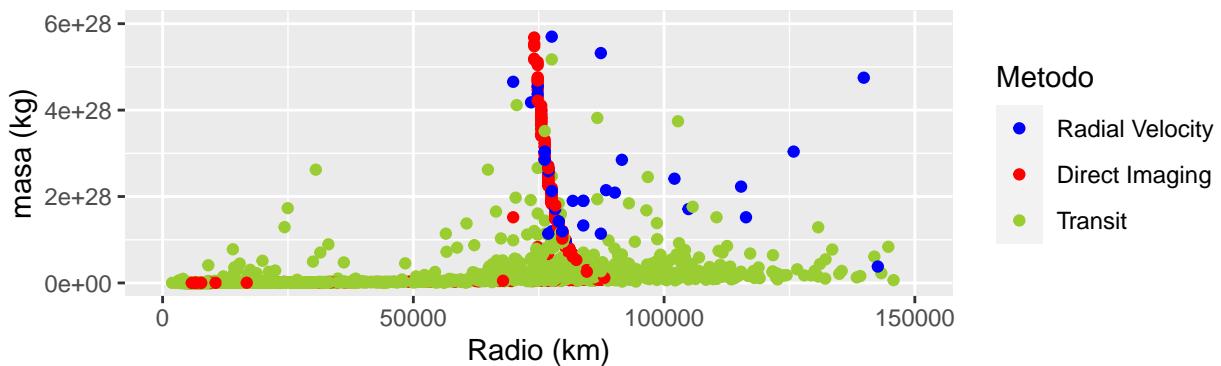
graf_radioymasa_ac_Metodoyanio <- ggplot(dataTresMetodos, aes(x=radioTresMetodos, y=masaTresMetodos, color=tres_metodos))
  geom_jitter(alpha=1) +
  ylab("masa (kg)") +
  xlab("Radio (km)")+
  ggtitle("MASA VS. RADIO SEGUN METODO DE DESCUBRIMIENTO")+
  scale_color_manual(name = "Metodo", labels = unique(tres_metodos), values = c("blue","red","yellowgreen"))
  xlim(c(0, 1.5 * 10^05))+
  ylim(c(0, 6 * 10^28))

grid.arrange(graf_radioymasa_ac, graf_radioymasa_ac_Metodoyanio, nrow=2)
```

## MASA VS. RADIO SEGUN TIPO DE PLANETA (ACOTADO)



## MASA VS. RADIO SEGUN METODO DE DESCUBRIMIENTO



Observamos que la mayoria de planetas en la columna del grafico fueron detectados con el metodo de imagen directa, lo que podria indicar que el rango acotado de radios en el que se encuentra puede tener que ver con la precision del metodo o alguna particularidad del mismo.

Procedemos a realizar un analisis de las variables relacionadas a la orbita y el movimiento del exoplaneta. Observamos un patron en el grafico que describe el periodo orbital en funcion del radio orbital. Podriamos decir que a mayor radio orbital, el periodo tambien es mayor. Con respecto a la excentricidad en funcion de las otras dos variables, no se puede observar un patron claro que permita establecer relaciones. Vemos que la mayoria de los datos se acumula por debajo del 0,5 de excentricidad, lo que indica que la mayoria de los exoplanetas en nuestra muestra presenta orbitas cercanas a circulares y a su vez la excentricidad seria independiente del periodo orbital y del radio de la orbita.

```

periodo_vs_radio<-ggplot(data[data$orbital_radius<=1,], aes(x=orbital_radius, y=orbital_period)) +
  geom_jitter(alpha=1) +
  ylab("Periodo Orbital (años)") +
  xlab("Radio Orbital (AU)")+
  ggtitle("PERIODO EN FUNCION DEL RADIO ORBITAL")

ex_vs_radio<-ggplot(data[data$orbital_radius<=1,], aes(x=orbital_radius, y=eccentricity)) +
  geom_jitter(alpha=1) +
  ylab("Excentricidad") +
  xlab("Radio Orbital (AU)")+
  ggtitle("EXCENTRICIDAD EN FUNCION DEL RADIO ORBITAL")

ex_vs_periodo<-ggplot(data[data$orbital_radius<=1,], aes(x=orbital_period, y=eccentricity)) +
  geom_jitter(alpha=1) +

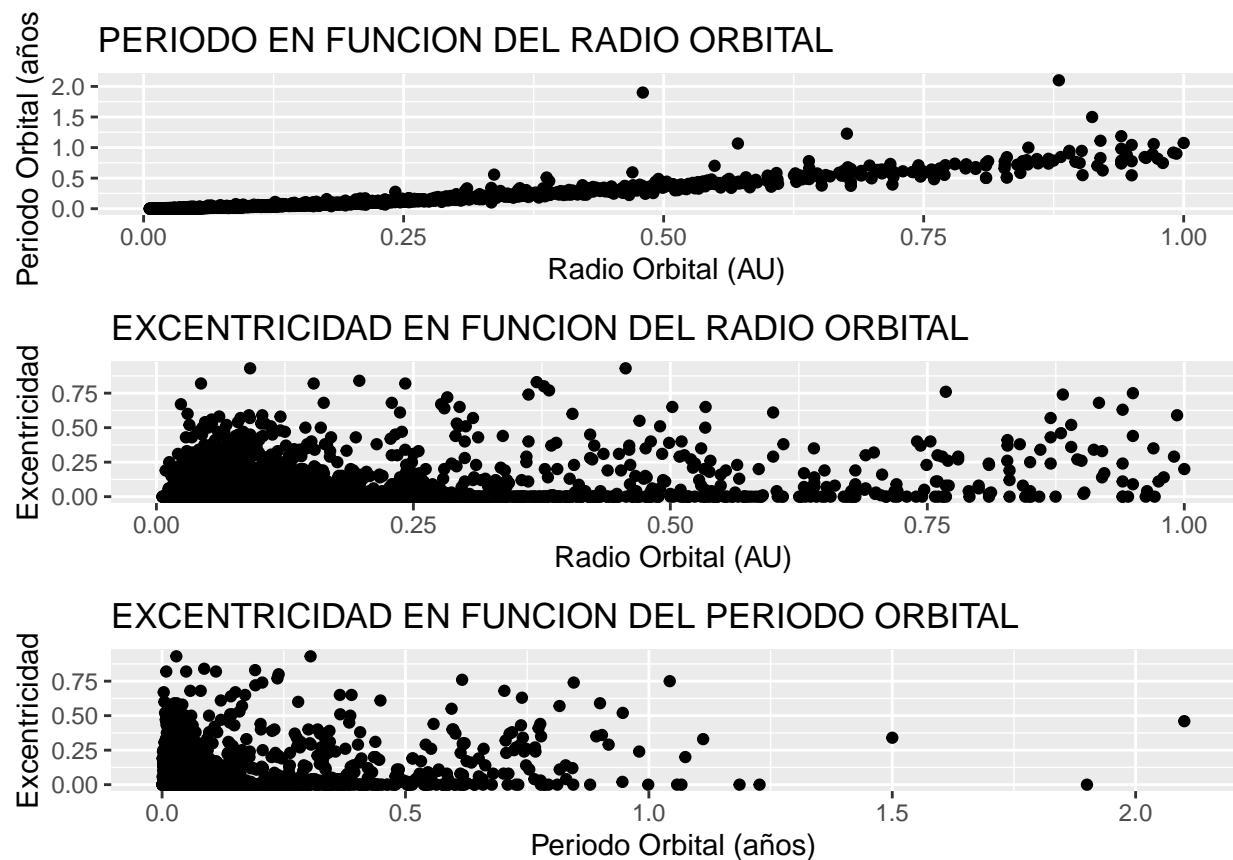
```

```

ylab("Excentricidad") +
xlab("Periodo Orbital (años)")+
ggtitle("EXCENTRICIDAD EN FUNCION DEL PERIODO ORBITAL")

grid.arrange(periodo_vs_radio, ex_vs_radio, ex_vs_periodo, nrow=3)

```



## Clusterizacion

Para llevar a cabo la clusterizacion establecemos los siguientes presupuestos sobre nuestras variables:

1. Masa y radio son variables relacionadas.
2. Excentricidad, periodo y radio orbital son variables relacionadas, al ser descriptivas de la orbita y el movimiento del exoplaneta.
3. Excentricidad, periodo y radio orbital no se relacionan con masa y radio.

Normalizamos valores continuos en un nuevo dataset, omitiendo los datos extremos.

```

dataNorm <- data[(data$orbital_period<1&data$orbital_radius<1),c(2,3,6,7,8,10,11)]

for(j in 1:ncol(dataNorm)){
  for(i in 1:nrow(dataNorm)){

```

```

        dataNorm[i,j] <- (dataNorm[i,j]-min(dataNorm[,j]))/(max(dataNorm[,j])-min(dataNorm[,j]))
    }
}

```

Comenzamos con la clusterizacion por radio y masa. Creamos una matriz con los datos a tener en cuenta y graficamos distancia entre clusters por cantidad de clusters para definir que cantidad de centroides seria optima.

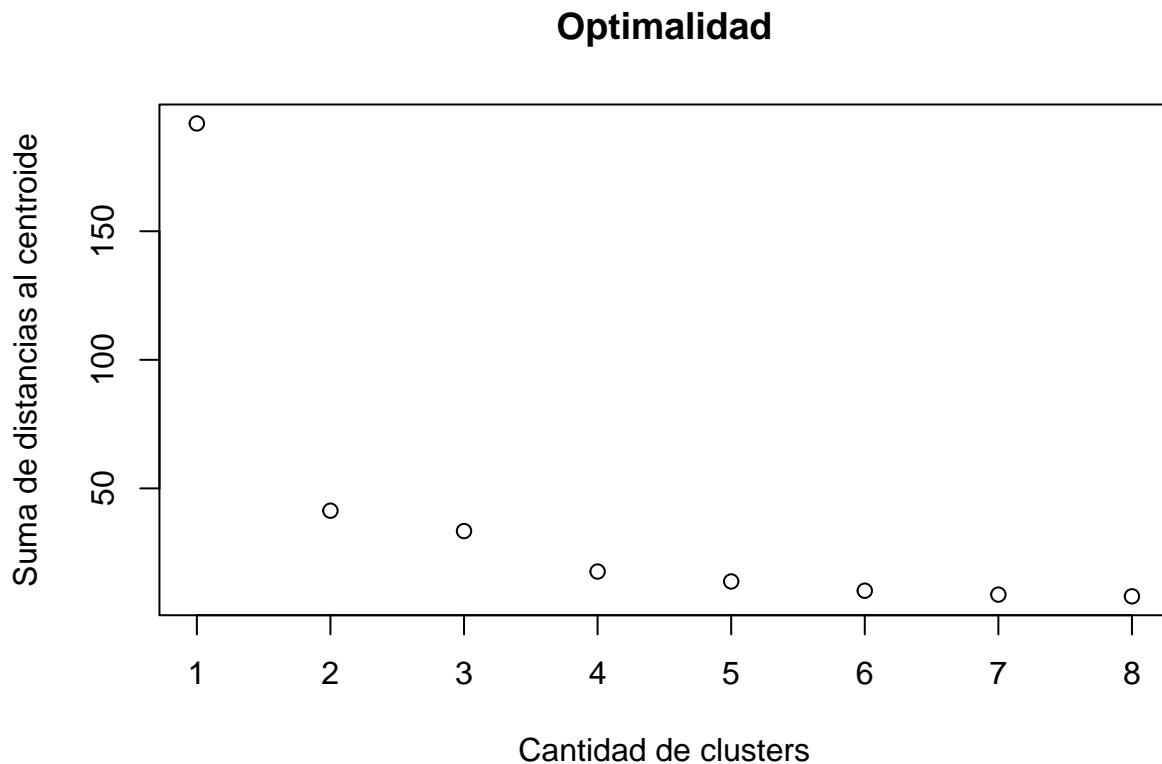
```

set.seed(1512)

matriz_radiomasa<-matrix(c(dataNorm$radio, dataNorm$masa), nrow=4159, ncol=2)
distancias <- c ()
for (k in 1:8) {
  micluster <- kmeans (matriz_radiomasa, k)
  distancias [k] <- sum (micluster$withinss)

}
plot (distancias, xlab="Cantidad de clusters", ylab= "Suma de distancias al centroide", main= "Optimalidad")

```



Vemos que el numero optimo de centroides es 4. Procedemos a realizar la segmentacion por radio y masa con 4 clusters y graficamos el resultado.

```

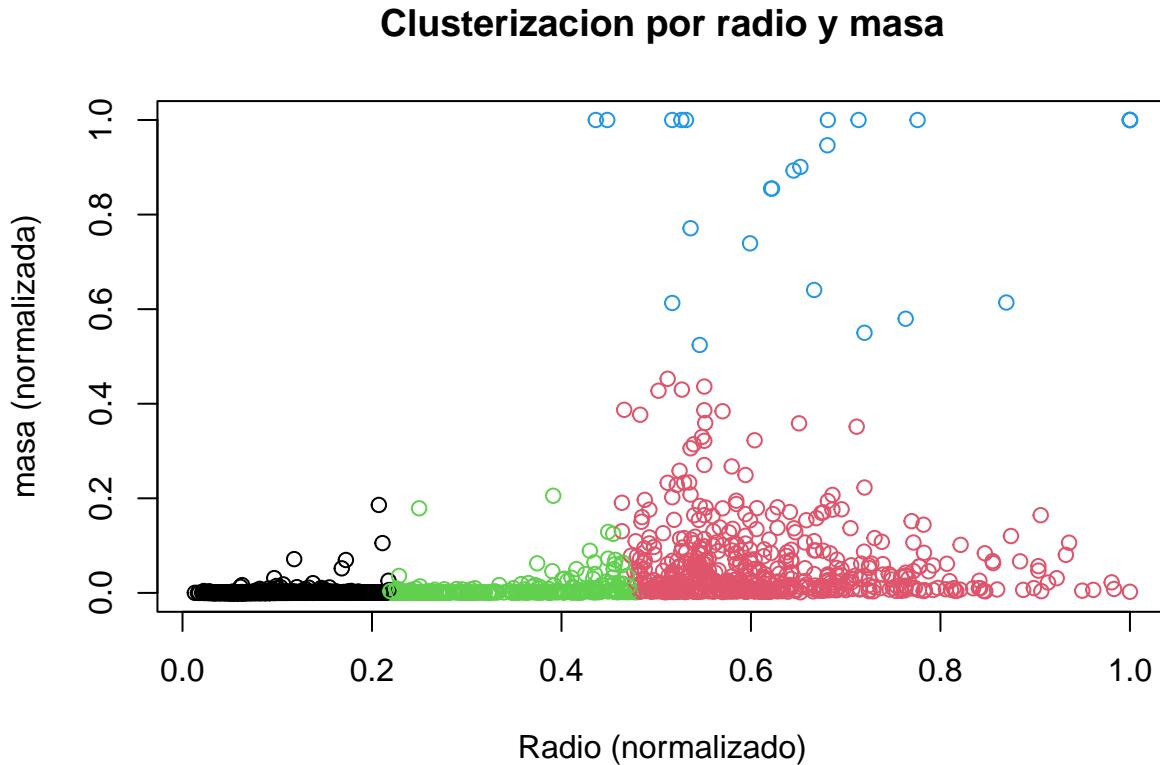
kmedias_radiomasa <- kmeans(matriz_radiomasa, iter.max=100, centers=4)
pertenencia <- kmedias_radiomasa$cluster

```

```

matriz_radiomasa<-cbind(matriz_radiomasa, pertenencia)
plot(matriz_radiomasa[, 1], matriz_radiomasa[, 2], col=matriz_radiomasa[, 3], xlab="Radio (normalizado")

```



Vemos que el radio juega un rol muy importante en la clusterizacion por masa y radio, a tal punto que en los primeros dos clusters no se ve division por masa mientras que se ve a cada cluster en un intervalo claro de radio. En la parte derecha del grafico, donde la variabilididad de masa es mas notoria, vemos que la masa juega un papel importante en la clusterizacion y separa claramente un cluster rojo de uno negro en un mismo intervalo de radios.

Relacionamos la clusterizacion por radio y masa con el tipo de planeta para ver cuantos planetas de cada tipo caen en cada cluster. En la tabla y el heatmap a continuacion observamos la relacion.

```

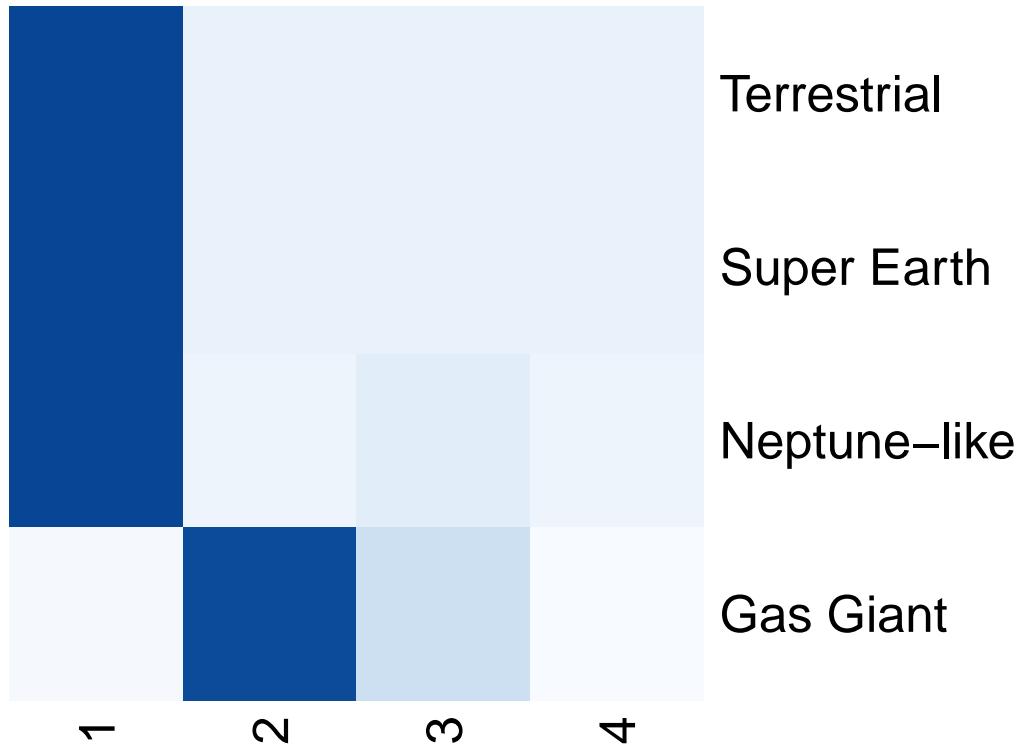
tipos<-data[(data$orbital_period<1&data$orbital_radius<1),]$planet_type
clusters<-matriz_radiomasa[, 3]

table(data.frame(tipos,clusters))

```

	clusters			
## tipos	1	2	3	4
## Gas Giant	28	639	173	22
## Neptune-like	1530	0	111	0
## Super Earth	1471	0	0	2
## Terrestrial	182	0	0	1

```
heatmap(table(data.frame(tipos,clusters)), Colv = NA, Rowv = NA, col=colorRampPalette(brewer.pal(8, "Blues")))
```



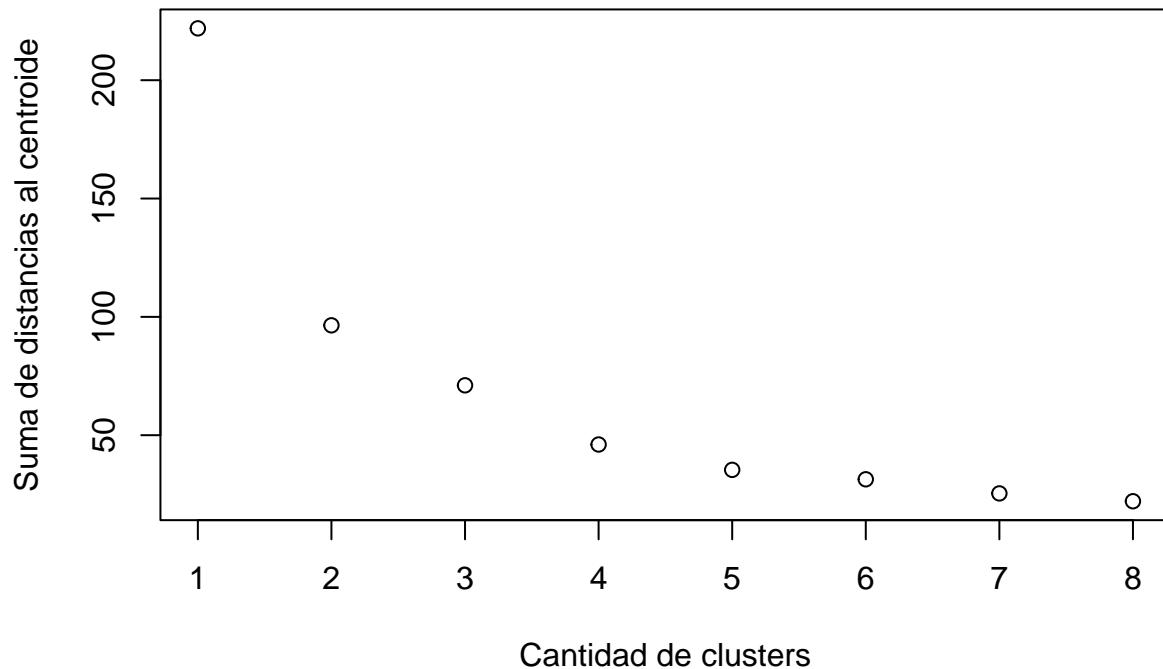
Se ve que en su mayoria, casi todos los tipos caen en el cluster 1, existiendo mayor variabilidad en los gigantes gaseosos, que se ubican mayoritariamente en el cluster 2 y conforman casi la totalidad del cluster 3 y 4.

Procedemos a realizar la clusterizacion segun variables relacionadas a la orbita y al movimiento del exoplaneta creando una matriz con los datos relevantes y graficando la cantidad optima de clusters.

```
matriz_orbita<-matrix(c(dataNorm$orbital_radius, dataNorm$orbital_period, dataNorm$eccentricity), nrow=8, byrow=TRUE)

distancias2 <- c ()
for (k in 1:8) {
  micluster2 <- kmeans (matriz_orbita, k)
  distancias2 [k] <- sum (micluster2$withinss)
}
plot (distancias2, xlab="Cantidad de clusters", ylab= "Suma de distancias al centroide", main= "Optimal number of clusters")
```

## Optimalidad

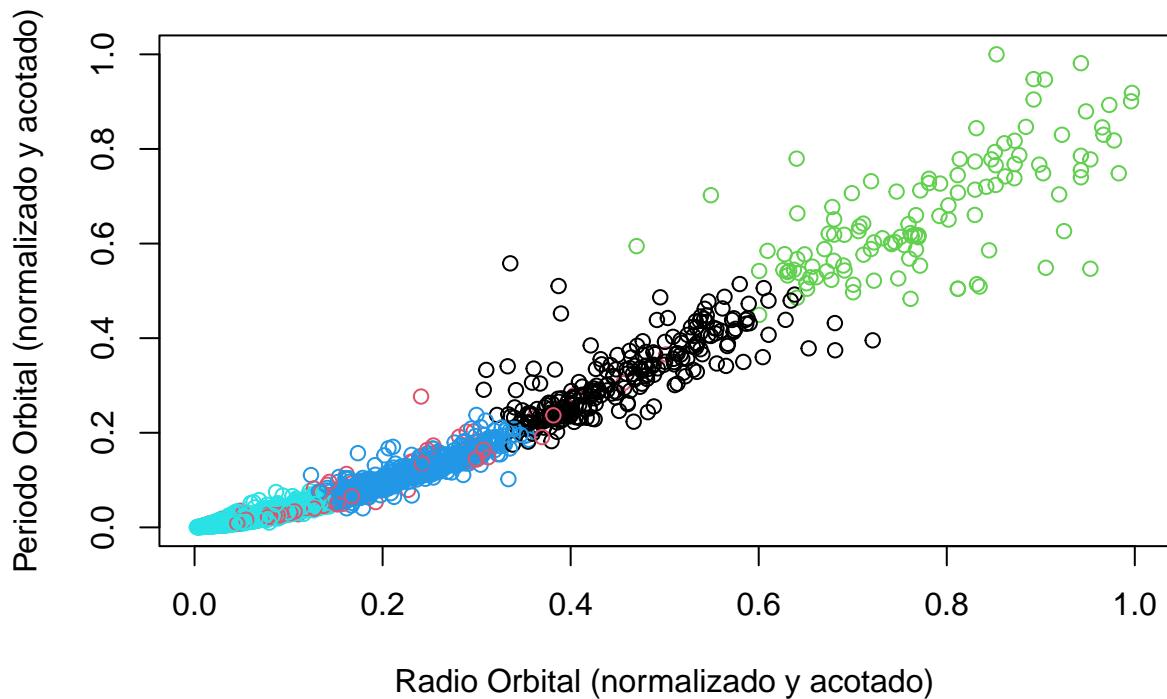


El numero optimo de centroides esta vez es 5. Graficamos los resultados.

```
kmedias_orbita <- kmeans(matriz_orbita, iter.max=100, centers=5)
pertenencia2 <- kmedias_orbita$cluster

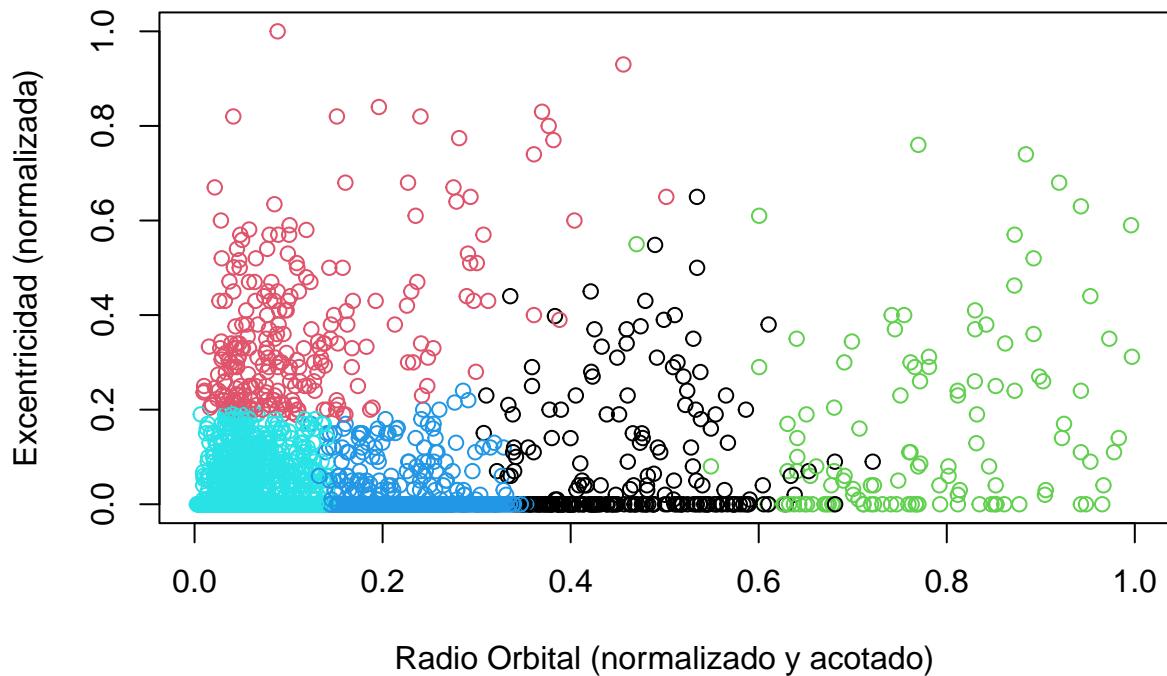
matriz_orbita<-cbind(matriz_orbita, pertenencia2)
plot(matriz_orbita[, 1], matriz_orbita[, 2], col=matriz_orbita[, 4], xlab="Radio Orbital (normalizado y
```

## Clusterizacion por radio orbital, periodo y excentricidad



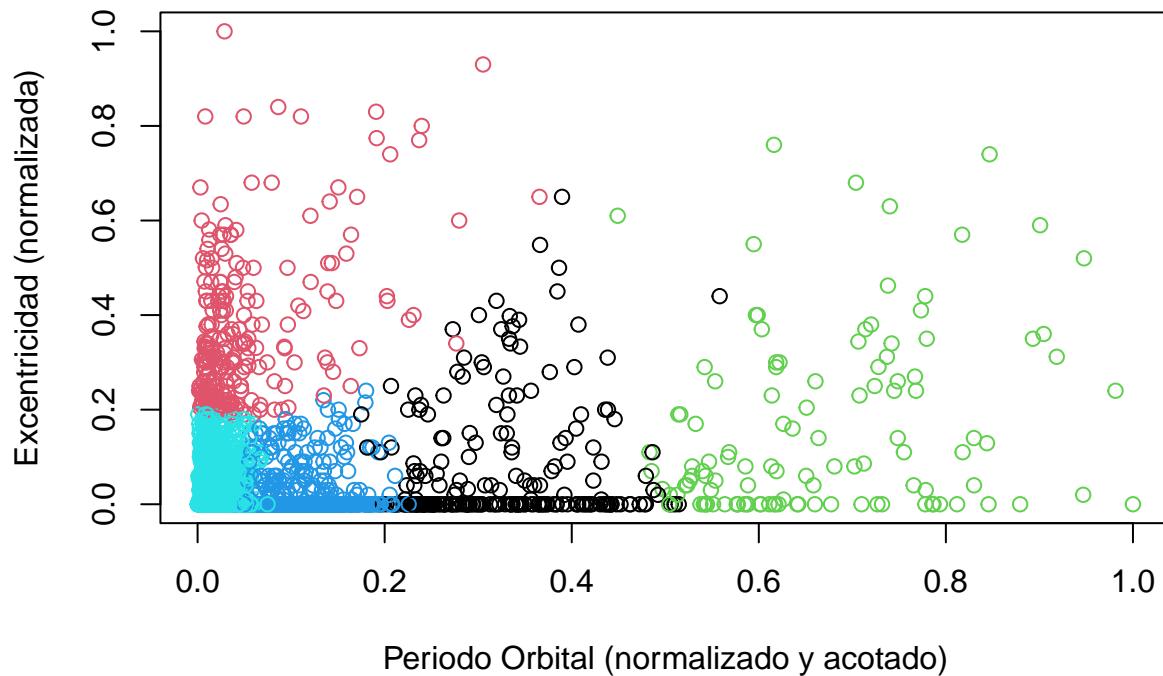
```
plot(matriz_orbita[, 1], matriz_orbita[, 3], col=matriz_orbita[, 4], xlab="Radio Orbital (normalizado y acotado)", ylab="Periodo Orbital (normalizado y acotado)", main="Clusterizacion por radio orbital, periodo y excentricidad")
```

## Clusterizacion por radio orbital, periodo y excentricidad



```
plot(matriz_orbita[, 2], matriz_orbita[, 3], col=matriz_orbita[, 4], xlab="Periodo Orbital (normalizado")
```

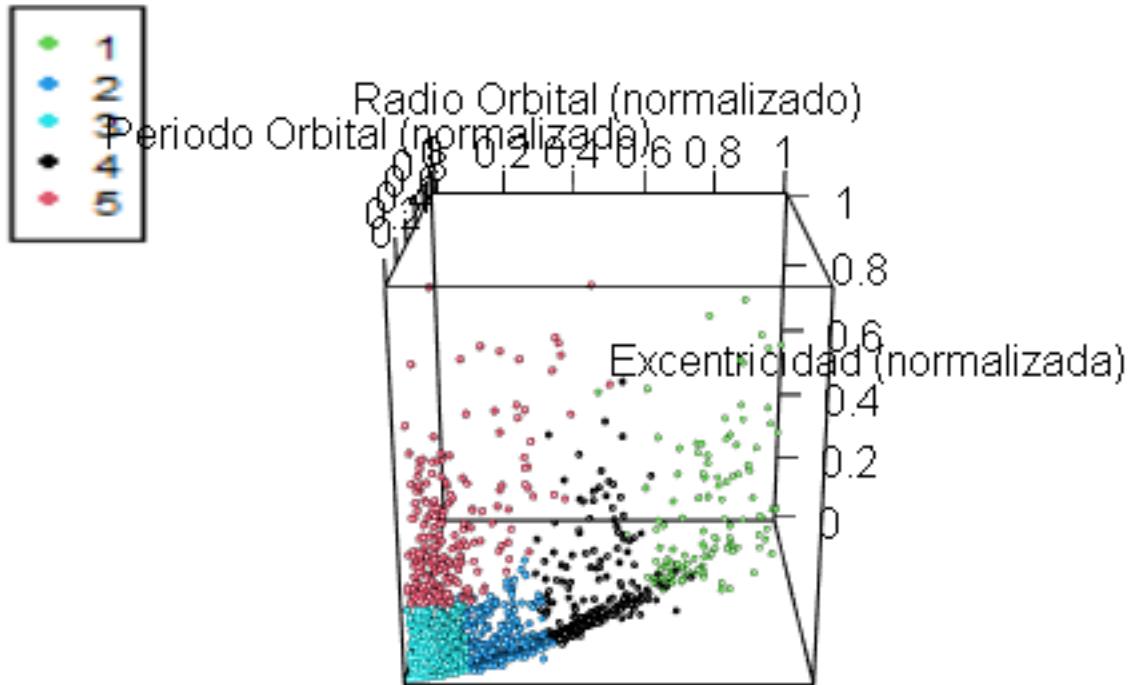
## Clusterizacion por radio orbital, periodo y excentricidad



Se puede ver en el primer grafico que la distincion entre clusters es mas clara a mayor radio Orbital. Entre el valor 0 y 0,5 hay una dispersion de circulos rojos que se camufla entre los clusters turquesa, azul y negro. La distincion es mas clara em el segundo y tercer grafico, cuando vemos que la excentricidad juega un rol importante en los valores de radio orbital hasta 0,5 y periodo orbital hasta 0,4. La relacion se puede apreciar claramente en el siguiente grafico 3d.

```
plot3d(
  x=matriz_orbita[, 1], matriz_orbita[, 2], z=matriz_orbita[, 3],
  col = matriz_orbita[, 4],
  type="s",
  radius=0.01,
  xlab="Radio Orbital (normalizado)", ylab="Periodo Orbital (normalizado)", zlab="Excentricidad (normalizada)",
  legend3d("topleft", legend = paste(sort(unique(matriz_orbita[, 4]))), col= paste(unique(matriz_orbita[, 4])))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead
```

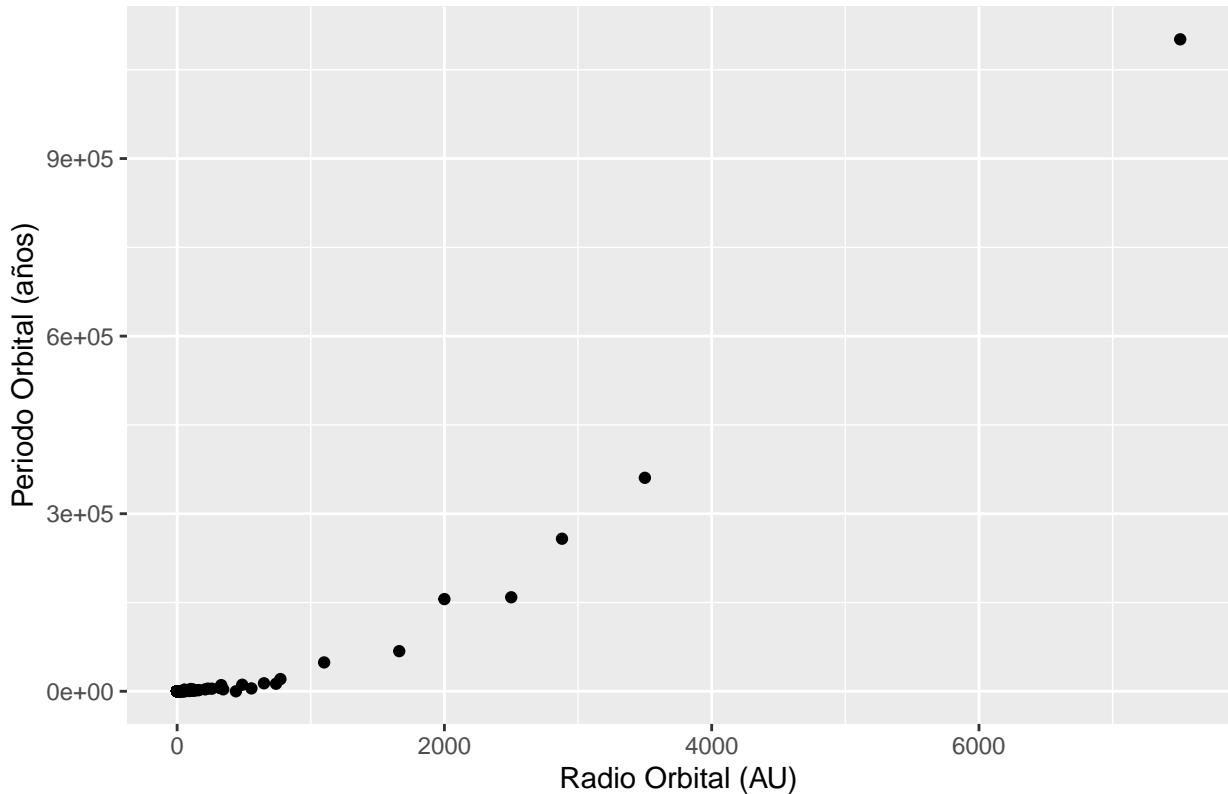


## Modelado

Notamos que podria existir un patron en el grafico de periodo en funcion del radio orbital y decidimos modelarlo.

```
ggplot(data, aes(x=orbital_radius, y=orbital_period)) +
  geom_jitter(alpha=1) +
  ylab("Periodo Orbital (años)") +
  xlab("Radio Orbital (AU)")+
  ggtitle("PERIODO EN FUNCION DEL RADIO ORBITAL")
```

## PERIODO EN FUNCION DEL RADIO ORBITAL



Notamos que la trayectoria de los puntos se asemeja a una funcion de grado superior a uno. Decidimos en primer lugar ajustar con una funcion de grado 2 y calcular el error con validacion cruzada.

```

periodos <- data$orbital_period
n<-length(periodos)
predichos_periodo.oos<-rep(NA,n)

for (i in 1:n) {
  ajus.cv<-lm(orbital_period~poly(orbital_radius,2),data=data[-i,])
  predichos_periodo.oos[i]<-predict(ajus.cv,newdata=data[i,])
}

# MAE
mean(abs(periodos-predichos_periodo.oos))

## [1] 249.8234

# PMAE
pmaeM2.cv<-mean(abs(periodos-predichos_periodo.oos))/mean(periodos)
pmaeM2.cv

## [1] 0.5201123

```

Vemos que el pmae ronda el 0,52. Intentamos nuevamente con un polinomio de grado 3 y calculamos el error.

```

n<-length(periodos)
predichos_periodo.oos<-rep(NA,n)

for (i in 1:n) {
  ajus.cv<-lm(orbital_period~poly(orbital_radius,3),data=data[-i,])
  predichos_periodo.oos[i]<-predict(ajus.cv,newdata=data[i,])
}

# MAE
mean(abs(periodos-predichos_periodo.oos))

```

## [1] 164.8207

```

# PMAE
pmaeM2.cv<-mean(abs(periodos-predichos_periodo.oos))/mean(periodos)
pmaeM2.cv

```

## [1] 0.3431435

El pmae se reduce a 0,34. La descripción del periodo en función del radio resulta más acertada con un polinomio de grado 3 con los siguientes coeficientes.

```

ajus.RadioVsPeriodo <- lm(orbital_period~poly(orbital_radius,3), data = data)
ajus.RadioVsPeriodo$coefficients

```

```

##                   (Intercept) poly(orbital_radius, 3)1 poly(orbital_radius, 3)2
##                   480.3258          1153517.6185         356633.6830
## poly(orbital_radius, 3)3
##                   -73868.7176

```

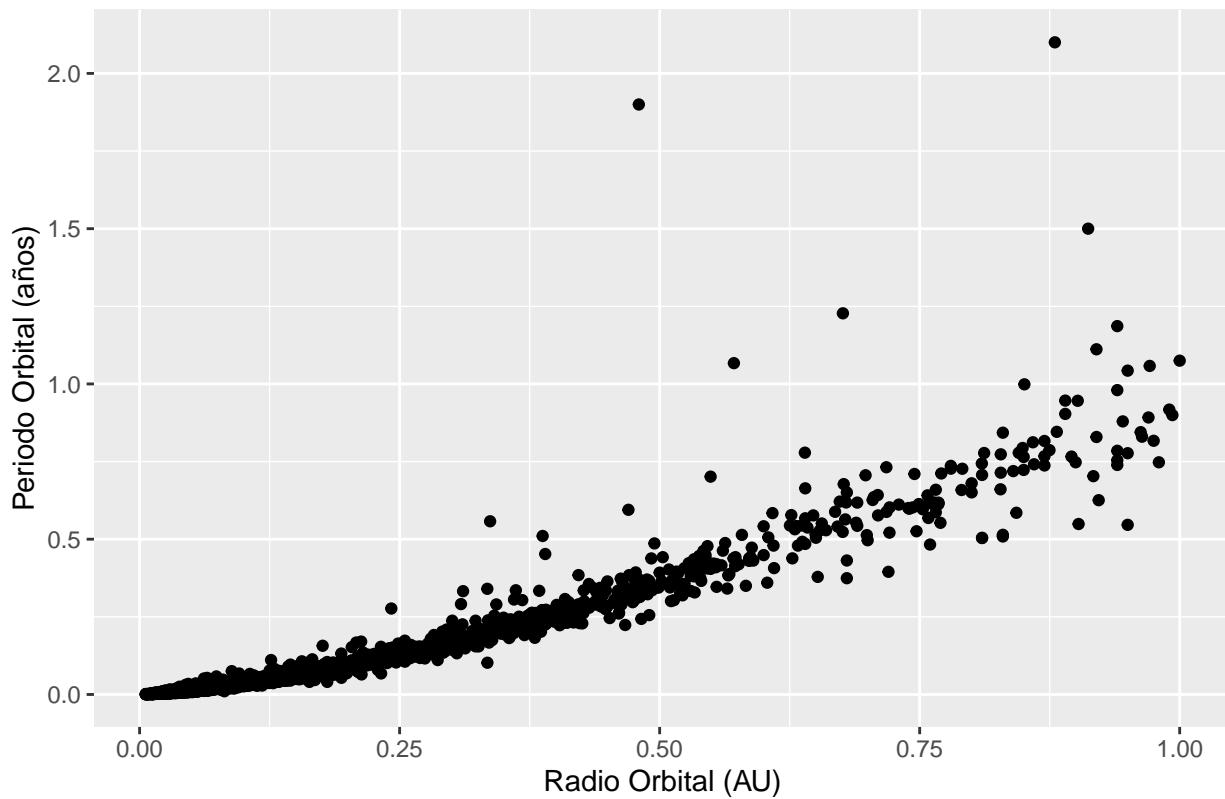
Notamos que alrededor del 87,5% de nuestros datos se encuentra en un radio orbital menor a 1 AU. Observamos el gráfico y notamos que los puntos siguen una trayectoria similar a una función lineal.

```

ggplot(data[data$orbital_radius<=1], aes(x=orbital_radius, y=orbital_period)) +
  geom_jitter(alpha=1) +
  ylab("Período Orbital (años)") +
  xlab("Radio Orbital (AU)")+
  ggtitle("PERÍODO EN FUNCION DEL RADIO ORBITAL")

```

## PERIODO EN FUNCION DEL RADIO ORBITAL



Ajustamos los datos de interes a una funcion lineal de Periodo en funcion del Radio y calculamos el error.

```

data_modelado <- data[data$orbital_radius<=1,]
periodos <- data_modelado$orbital_period

n<-length(periodos)
predichos_periodo.oos<-rep(NA,n)

for (i in 1:n) {
  ajus.cv<-lm(orbital_period~orbital_radius,data=data_modelado[-i,])
  predichos_periodo.oos[i]<-predict(ajus.cv,newdata=data_modelado[i,])
}

# MAE
mean(abs(periodos-predichos_periodo.oos))

## [1] 0.02312142

# PMAE
pmaeM2.cv<-mean(abs(periodos-predichos_periodo.oos))/mean(periodos)
pmaeM2.cv

## [1] 0.3060836

```

Observamos que el pmae ronda el 0,30. Creemos que podemos ajustar el modelo a una funcion cuadratica para obtener mejores resultados.

```

n<-length(periodos)
predichos_periodo.oos<-rep(NA,n)

for (i in 1:n) {
  ajus.cv<-lm(orbital_period~poly(orbital_radius,2),data=data_modelado[-i,])
  predichos_periodo.oos[i]<-predict(ajus.cv,newdata=data_modelado[i,])
}

# MAE
mean(abs(periodos-predichos_periodo.oos))

```

## [1] 0.01131883

```

# PMAE
pmaeM2.cv<-mean(abs(periodos-predichos_periodo.oos))/mean(periodos)
pmaeM2.cv

```

## [1] 0.1498397

Resulta ser el modelo con menor error que encontramos hasta ahora, con un pmae de 0,14. La funcion que ajusta esta relacion tiene los siguientes coeficientes.

```

ajus.final <- lm(orbital_period~poly(orbital_radius,2),data=data_modelado)
ajus.final$coefficients

```

	(Intercept)	poly(orbital_radius, 2)1	poly(orbital_radius, 2)2
##	0.07553958	8.80095074	1.52885518

Continuamos con un modelado del brillo en funcion de la distancia. Ajustamos en el grafico de brillo por distancia a funciones lineales por tipo de planeta y obtenemos una imagen que podria indicarnos una tendencia lineal creciente de la trayectoria que siguen los puntos (a menor distancia, mayor brillo)

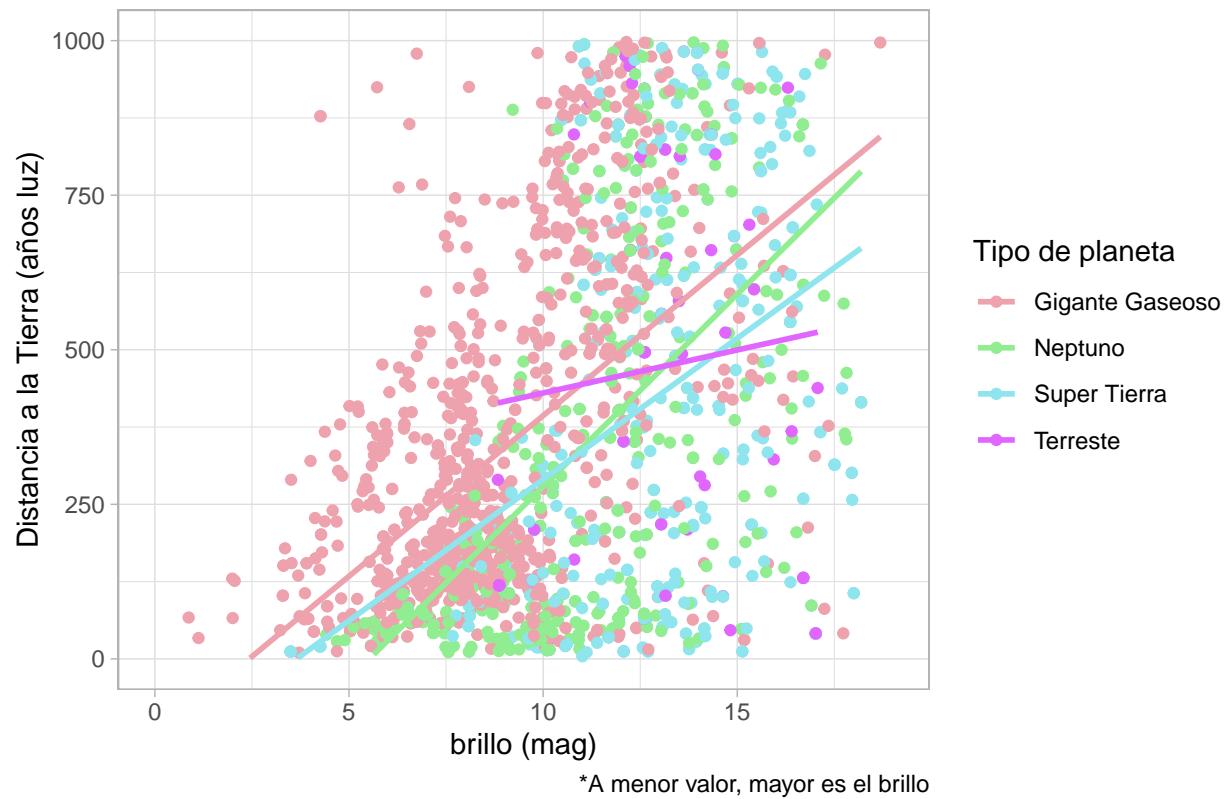
```

ggplot(data, aes(x=stellar_magnitude, y=distance, color=planet_type)) +
  geom_jitter(alpha=1) +
  geom_smooth(method="lm", se=FALSE) +
  xlab("brillo (mag)") +
  ylab("Distancia a la Tierra (años luz)") +
  ggtitle("BRILLO EN FUNCION DE LA DISTANCIA A LA TIERRA POR TIPO DE PLANETA") +
  scale_color_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra',
  theme_light()+
  ylim(c(0, 1000))+
  xlim(c(0, 19))+
  labs(caption = "*A menor valor, mayor es el brillo")

## `geom_smooth()` using formula = 'y ~ x'

```

## BRILLO EN FUNCION DE LA DISTANCIA A LA TIERRA POR TIPO DE P



Aproximamos y calculamos el error por validacion cruzada.

```

distancia <- data$distance

h<-length(distancia)
predichos_dist.oos<-rep(NA,h)

for (i in 1:h) {
  ajus.cv<-lm(distance~stellar_magnitude,data=data[-i,])
  predichos_dist.oos[i]<-predict(ajus.cv,newdata=data[i,])
}

#mae
mean(abs(distancia-predichos_dist.oos))

## [1] 945.7883

#pmae
pmaeBrilloDistancia.cv<-mean(abs(distancia-predichos_dist.oos))/mean(distancia)
pmaeBrilloDistancia.cv

## [1] 0.5261916

```

Obtenemos un pmae de 0,52 con el ajuste que tiene los siguientes coeficientes.

```

ajus.brillo<- lm(distance~stellar_magnitude, data = data)
ajus.brillo$coefficients

```

```

##           (Intercept) stellar_magnitude
## -2730.603             356.659

```

Observamos tambien que la tendencia lineal es muy marcada en los Gigantes Gaseosos y a su vez es uno de los tipos de planeta de los que tenemos mas datos. Graficamos un ajuste lineal y un ajuste cuadratico para Gigantes Gaseosos.

```

data_gaseosos <- data[data$planet_type=="Gas Giant",]

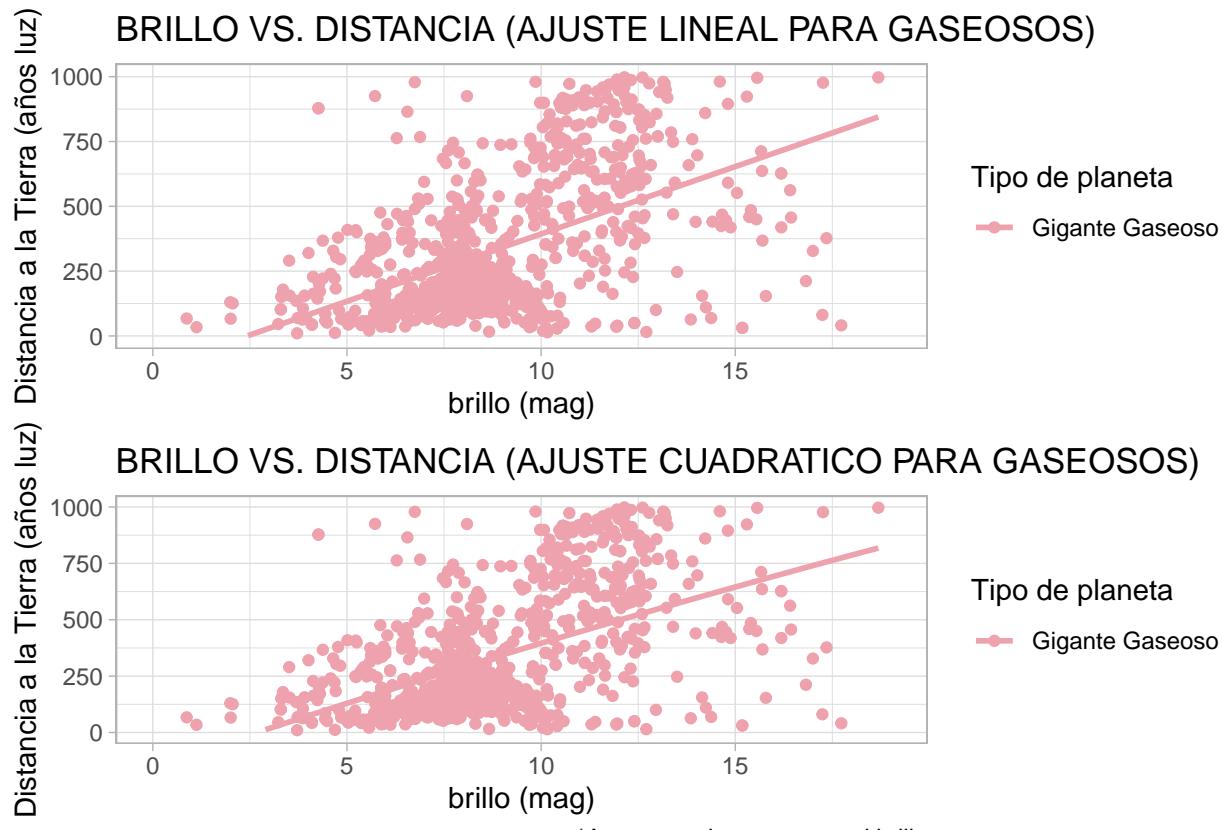
gaseosos_lineal<- ggplot(data_gaseosos, aes(x=stellar_magnitude, y=distance, color=planet_type)) +
  geom_jitter(alpha=1) +
  geom_smooth(method="lm", se=FALSE) +
  xlab("brillo (mag)") +
  ylab("Distancia a la Tierra (años luz)")+
  ggtitle("BRILLO VS. DISTANCIA (AJUSTE LINEAL PARA GASEOSOS)")+
  scale_color_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', 'Terraformable', 'Terrestre', 'Habitable', 'Anomalia'), guide=FALSE)+theme_light()+
  ylim(c(0, 1000))+
  xlim(c(0, 19))

gaseosos_cuadratico<-ggplot(data_gaseosos, aes(x=stellar_magnitude, y=distance, color=planet_type)) +
  geom_jitter(alpha=1) +
  geom_smooth(method="lm", formula = y ~ poly(x, 2, raw = TRUE), se=FALSE) +
  xlab("brillo (mag)") +
  ylab("Distancia a la Tierra (años luz)")+
  ggtitle("BRILLO VS. DISTANCIA (AJUSTE CUADRATICO PARA GASEOSOS)")+
  scale_color_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', 'Terraformable', 'Terrestre', 'Habitable', 'Anomalia'), guide=FALSE)+theme_light()+
  ylim(c(0, 1000))+
  xlim(c(0, 19))+
  labs(caption = "*A menor valor, mayor es el brillo")

grid.arrange(gaseosos_lineal, gaseosos_cuadratico, nrow=2)

## `geom_smooth()` using formula = 'y ~ x'

```



El ajuste cuadrático es observacionalmente similar al lineal. Procedemos a realizar un ajuste lineal y calcular el error por validación cruzada.

```

distancia<-data_gaseosos$distance

h<-length(distancia)
predichos_dist.oos<-rep(NA,h)

for (i in 1:h) {
  ajus.cv<-lm(distance~stellar_magnitude,data=data_gaseosos[-i,])
  predichos_dist.oos[i]<-predict(ajus.cv,newdata=data_gaseosos[i,])
}

#mae
mean(abs(distancia-predichos_dist.oos))

## [1] 758.4186

#pmae
pmae_gaseosos.cv<-mean(abs(distancia-predichos_dist.oos))/mean(distancia)
pmae_gaseosos.cv

## [1] 0.6962304

```

El pmae resulta ser 0,69. Incluso mayor que en la aproximacion que no considera el tipo de planeta. Esto puede deberse a la enorme variabilidad que existe en el brillo de los gigantes gaseosos. Realizamos el ajuste cuadratico.

```

h<-length(distancia)
predichos_dist.oos<-rep(NA,h)

for (i in 1:h) {
  ajus.cv<-lm(distance~poly(stellar_magnitude,2),data=data_gaseosos[-i,])
  predichos_dist.oos[i]<-predict(ajus.cv,newdata=data_gaseosos[i,])
}

#mae
mean(abs(distancia-predichos_dist.oos))

## [1] 636.4902

#pmae
pmae_gaseosos.cv<-mean(abs(distancia-predichos_dist.oos))/mean(distancia)
pmae_gaseosos.cv

## [1] 0.5842998

```

El pmae ronda 0,58 y concluimos que el ajuste cuadratico aproxima mejor a los valores de nuestro dataset y tiene los siguientes coeficientes.

```

ajus.brillo_gaseoso<- lm(distance~poly(stellar_magnitude,2),data=data_gaseosos[-i,])
ajus.brillo_gaseoso$coefficients

##           (Intercept) poly(stellar_magnitude, 2)1
##                 1089.832                  46000.861
## poly(stellar_magnitude, 2)2
##                 21325.756

```

## Clasificacion

Primero realizamos histogramas de todas las variables numéricas, con el objetivo de analizar qué variables son las que más diferencias imponen en cuanto a los tipos de planetas.

```

m <- ggplot(data, aes(x = masa, fill = planet_type, colour=planet_type)) +
  geom_histogram(aes(y=..density..),position = "identity", alpha = 0.7,bins=50)+ 
  scale_color_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  scale_fill_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  xlim(c(0,quantile(data$masa,c(0.7))))+
  xlab("Masa(kg)")+
  ylab("")+
  guides(fill = guide_legend(title = "Tipo de planeta"),
        colour = guide_legend(title = "Tipo de planeta"))+
  theme(legend.position="none")+
  scale_y_continuous(

```

```

    labels = scales::number_format(accuracy = 0.01))

r <- ggplot(data, aes(x = radio, fill = planet_type, colour=planet_type)) +
  geom_histogram(aes(y=..density..),position = "identity", alpha = 0.7,bins=50) +
  xlim(c(0,quantile(data$radio,c(0.9))))+
  xlab("Radio(km)")+
  ylab("")+
  guides(fill = guide_legend(title = "Tipo de planeta"),
        colour = guide_legend(title = "Tipo de planeta"))+
  scale_color_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  scale_fill_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  theme(legend.position="none")

d <- ggplot(data, aes(x = distance, fill = planet_type, colour=planet_type)) +
  geom_histogram(aes(y=..density..),position = "identity", alpha = 0.7,bins=50) +
  xlim(c(0,quantile(data$distance,c(0.95))))+
  xlab("Distancia a Tierra (años luz)")+
  ylab("")+
  guides(fill = guide_legend(title = "Tipo de planeta"),
        colour = guide_legend(title = "Tipo de planeta"))+
  scale_color_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  scale_fill_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  theme(legend.position="none")

b <- ggplot(data, aes(x = stellar_magnitude, fill = planet_type, colour=planet_type)) +
  geom_histogram(aes(y=..density..),position = "identity", alpha = 0.7,bins=50) +
  xlim(c(0,quantile(data$stellar_magnitude,c(0.9))))+
  xlab("Brillo(mag)")+
  ylab("")+
  guides(fill = guide_legend(title = "Tipo de planeta"),
        colour = guide_legend(title = "Tipo de planeta"))+
  scale_color_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  scale_fill_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  theme(legend.position="none")

o_r <- ggplot(data, aes(x = orbital_radius, fill = planet_type, colour=planet_type)) +
  geom_histogram(aes(y=..density..),position = "identity", alpha = 0.7,bins=50) +
  xlim(c(0,quantile(data$orbital_radius,c(0.9))))+
  xlab("Radio orbital(km)")+
  ylab("")+
  guides(fill = guide_legend(title = "Tipo de planeta"),
        colour = guide_legend(title = "Tipo de planeta"))+
  scale_color_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  scale_fill_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
  theme(legend.position="none")

o_p <- ggplot(data) +
  geom_histogram(aes(x = orbital_period, fill = planet_type, colour=planet_type,y=..density..),position =
  xlim(c(0,quantile(data$orbital_period,c(0.9))))+
  xlab("Periodo Orbital")+
  ylab("")+
  guides(fill = guide_legend(title = "Tipo de planeta"),

```

```

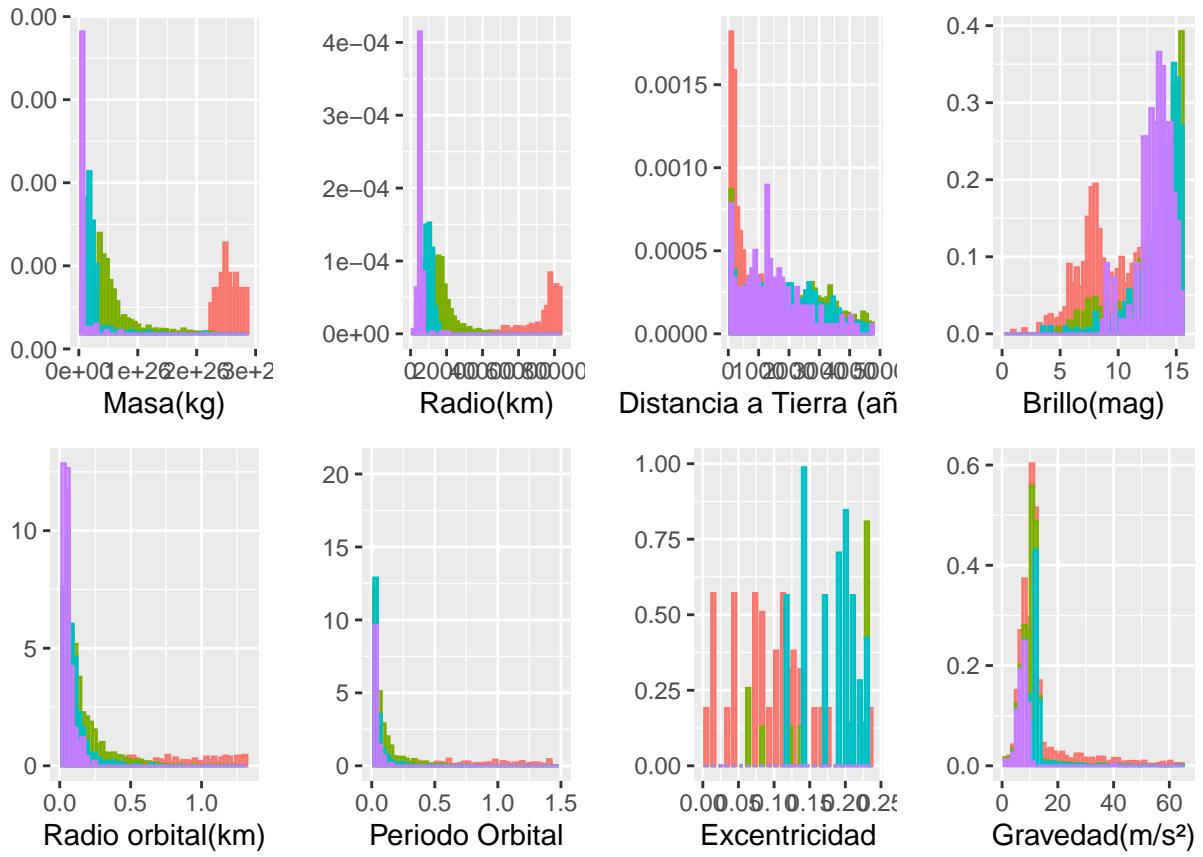
    colour = guide_legend(title = "Tipo de planeta"))+
scale_color_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
scale_fill_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
theme(legend.position="none")

e <- ggplot(data) +
geom_histogram(aes(x = eccentricity, fill = planet_type,colour=planet_type,y=..density..), alpha = 0.7,
xlim(c(0,quantile(data$eccentricity,c(0.9))))+
xlab("Excentricidad")+
ylim(c(0,1))+
ylab("")+
guides(fill = guide_legend(title = "Tipo de planeta"),
      colour = guide_legend(title = "Tipo de planeta"))+
scale_color_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
scale_fill_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
theme(legend.position="none")

g <- ggplot(data) +
geom_histogram(aes(x = gravedades, fill = planet_type,colour=planet_type,y=..density..), alpha = 0.7,
xlim(c(0,quantile(data$gravedades,c(0.9))))+
xlab("Gravedad(m/s2)")+
ylab("")+
guides(fill = guide_legend(title = "Tipo de planeta"),
      colour = guide_legend(title = "Tipo de planeta"))+
scale_color_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
scale_fill_discrete(labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', "Terreste"))+
theme(legend.position="none")

grid.arrange(m,r,d,b,o_r,o_p,e,g,nrow=2,ncol=4)

```



Pareciera ser que las variables en donde se observa más separación entre tipos de planetas son masa y radio, mientras que en las otras se observa mayor superposición.

Teniendo en cuenta esto, se propuso que quizá la masa y el radio son las variables predominantes a la hora de clasificar las mediciones, y el resto no contribuye o bien suma error a las predicciones. Para evaluar esta hipótesis primero se halló el mejor candidato a K para cada caso, y haciendo uso de este se comparó la exactitud predictiva mediante KNN en cada caso.

```

Nrep = 20 # #reps por cada k
Ntest = 30 #num de valores a predecir por cada test
max_val = 50
valores = 1:max_val
set.seed(256)

resultados_crossval_train_todas = matrix(NA, length(valores), Nrep)
resultados_crossval_test_todas = matrix(NA, length(valores), Nrep)

for(n in 1:Nrep){

  indices = sample(seq(1, nrow(data)), Ntest)
  test = data[indices, c(-1,-4,-5,-9)]
  train = data[-indices, c(-1,-4,-5,-9)]
  test_labels = data$planet_type[indices]
  train_labels = data$planet_type[-indices]

  for(k in 1:max_val){
    clasificador_data_k_train <- knn(train = train, test = test, cl = train_labels, k = k)
  }
}

```

```

clasificador_data_k_test <- knn(train = train, test = test, cl = train_labels, k = k)

exactitud_train = confusionMatrix(clasificador_data_k_train, train_labels)$overall[[1]]
exactitud_test = confusionMatrix(clasificador_data_k_test, test_labels)$overall[[1]]

resultados_crossval_train_todas[k, n] = exactitud_train
resultados_crossval_test_todas[k, n] = exactitud_test
}
}

exact_train_todas = rowMeans(resultados_crossval_train_todas)
exact_test_todas = rowMeans(resultados_crossval_test_todas)

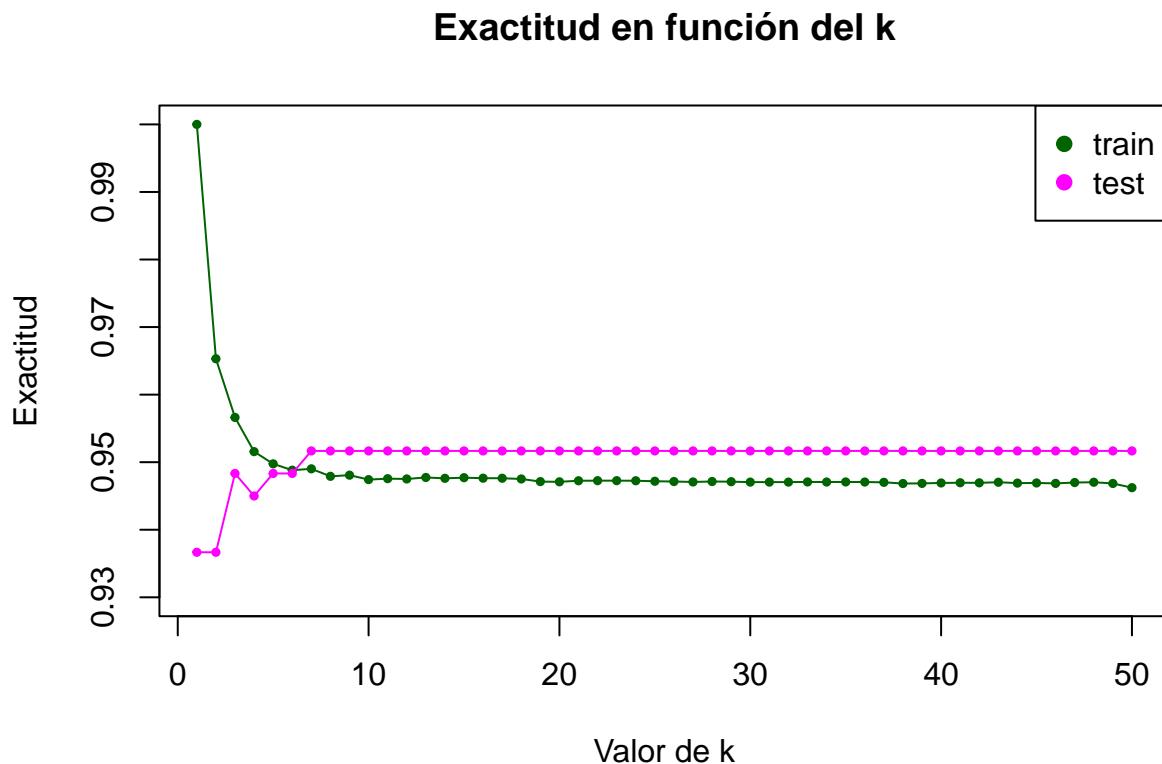
```

Observemos gráficamente el comportamiento del método con el mejor k hallado.

```

plot(valores, exact_train_todas, col = 'darkgreen', pch = 19, cex = 0.5, xlab = 'Valor de k', ylab = 'Exactitud')
lines(valores, exact_train_todas, col = 'darkgreen')
points(valores, exact_test_todas, col = 'magenta', pch = 19, cex = 0.5)
lines(valores, exact_test_todas, col = 'magenta')
legend('topright', legend = c('train', 'test'), col = c('darkgreen', 'magenta'), pch = 19)

```



```

mejor_k_todas = which.max(exact_test_todas)
mejor_k_todas

```

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

```

clasif_data_todas <- knn(train = data[, c(-1,-4,-5,-9)], test = data[, c(-1,-4,-5,-9)], cl = data[,4], k=5)
summary(clasif_data_todas)

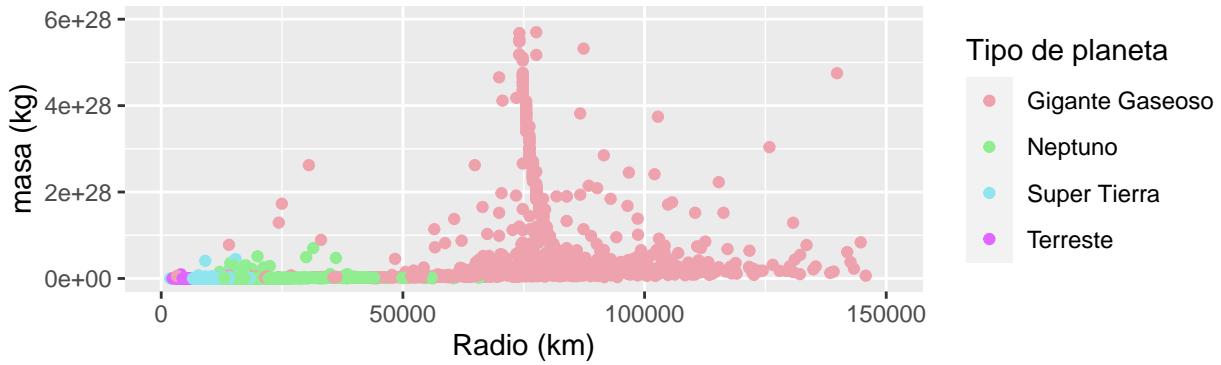
##      Gas Giant Neptune-like Super Earth Terrestrial
##        1471          1757         1358          176

plot_clasif_todas <- ggplot(data, aes(x=radio, y=masa, color=clasif_data_todas)) +
  geom_jitter(alpha=1) +
  ylab("masa (kg)") +
  xlab("Radio (km)")+
  ggtitle("MASA VS. RADIO POR TIPO (CLASIFICACION)")+
  xlim(c(0, 1.5 * 10^05))+ 
  ylim(c(0, 6 * 10^28))+ 
  scale_color_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso','Neptuno', 'Super Tierra', 'Terreste'))
  theme(legend.position="none")

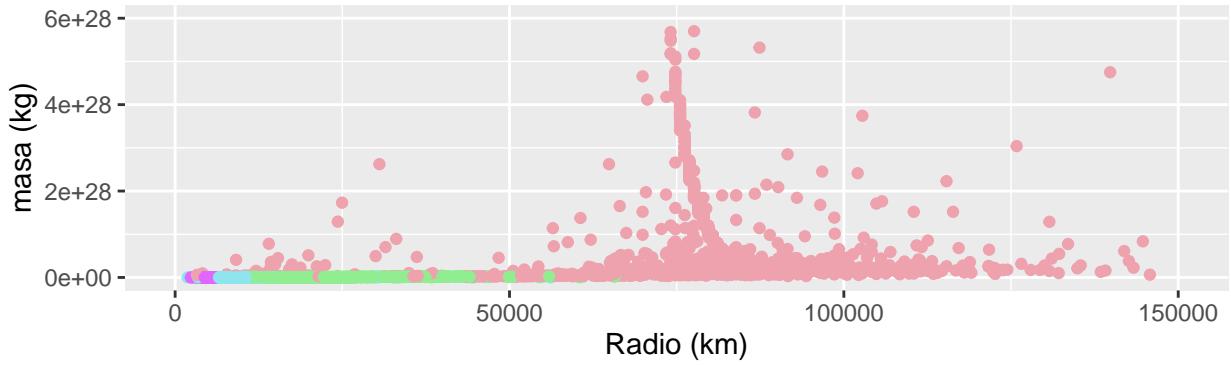
grid.arrange(graf_radioymasa_ac, plot_clasif_todas)

```

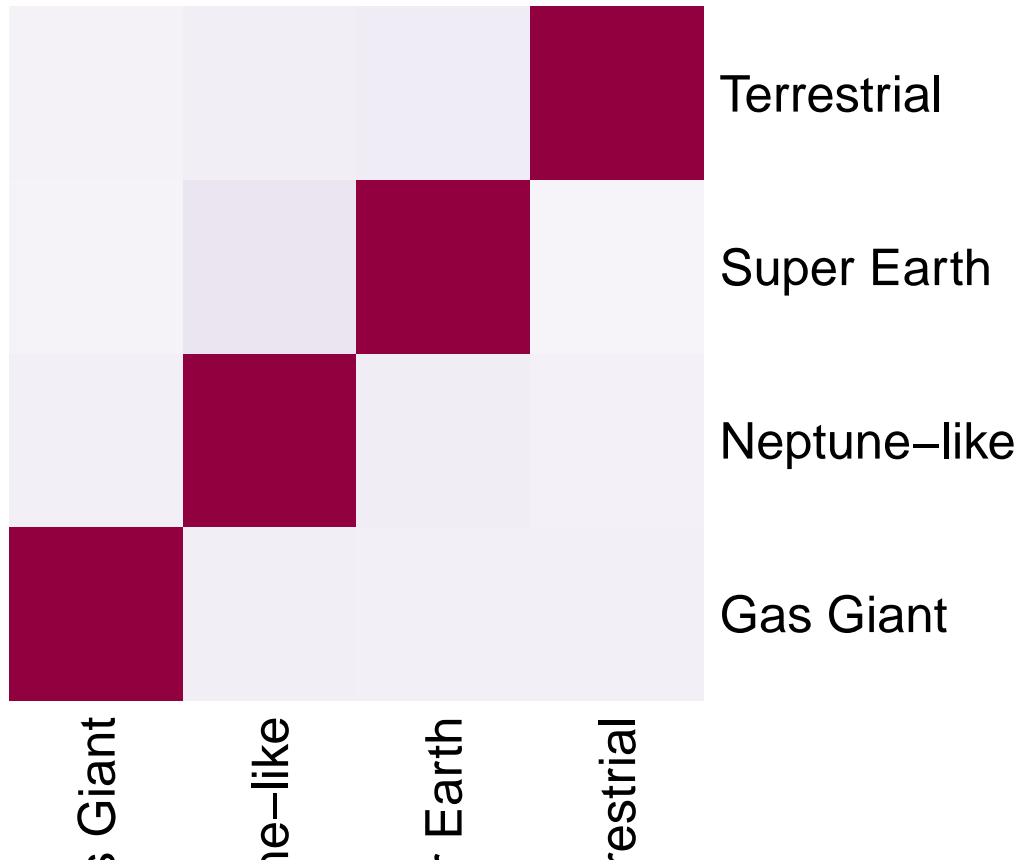
MASA VS. RADIO SEGUN TIPO DE PLANETA (ACOTADO)



MASA VS. RADIO POR TIPO (CLASIFICACION)



```
heatmap(table(data.frame(data$planet_type,clasif_data_todas)),Colv = NA, Rowv = NA,col=colorRampPalette
```



```
table(data.frame(clasif_data_todas,data$planet_type))
```

```
##          data.planet_type
## clasif_data_todas Gas Giant Neptune-like Super Earth Terrestrial
##      Gas Giant      1433        17       20        1
##      Neptune-like      2      1607      144        4
##      Super Earth      0        44     1307        7
##      Terrestrial      0        0        5      171
```

```
confusionMatrix(clasif_data_todas, data[,4])$overall[[1]]
```

```
## [1] 0.948761
```

Tanto en el heatmap como en la tabla se observa que parece haber una buena correlación entre las predicciones y los tipos reales, así como los gráficos entre la predicción y la realidad de los puntos son muy similares. Veamos de cuánto es la exactitud.

En el caso en que se consideraron todas las variables numéricas se consiguió una exactitud de 0.948761.

```
resultados_crossval_train_masayradio = matrix(NA, length(valores), Nrep)
resultados_crossval_test_masayradio = matrix(NA, length(valores), Nrep)

for(n in 1:Nrep){
```

```

indices = sample(seq(1, nrow(data)), Ntest)
test = data[indices, c(10,11)]
train = data[-indices, c(10,11)]
test_labels = data$planet_type[indices]
train_labels = data$planet_type[-indices]

for(k in 1:max_val){
  clasificador_data_k_train <- knn(train = train, test = train, cl = train_labels, k = k)
  clasificador_data_k_test <- knn(train = train, test = test, cl = train_labels, k = k)

  exactitud_train_masayradio = confusionMatrix(clasificador_data_k_train, train_labels)$overall[[1]]
  exactitud_test_masayradio = confusionMatrix(clasificador_data_k_test, test_labels)$overall[[1]]

  resultados_crossval_train_masayradio[k, n] = exactitud_train_masayradio
  resultados_crossval_test_masayradio[k, n] = exactitud_test_masayradio
}
exact_train_masayradio = rowMeans(resultados_crossval_train_masayradio)
exact_test_masayradio = rowMeans(resultados_crossval_test_masayradio)

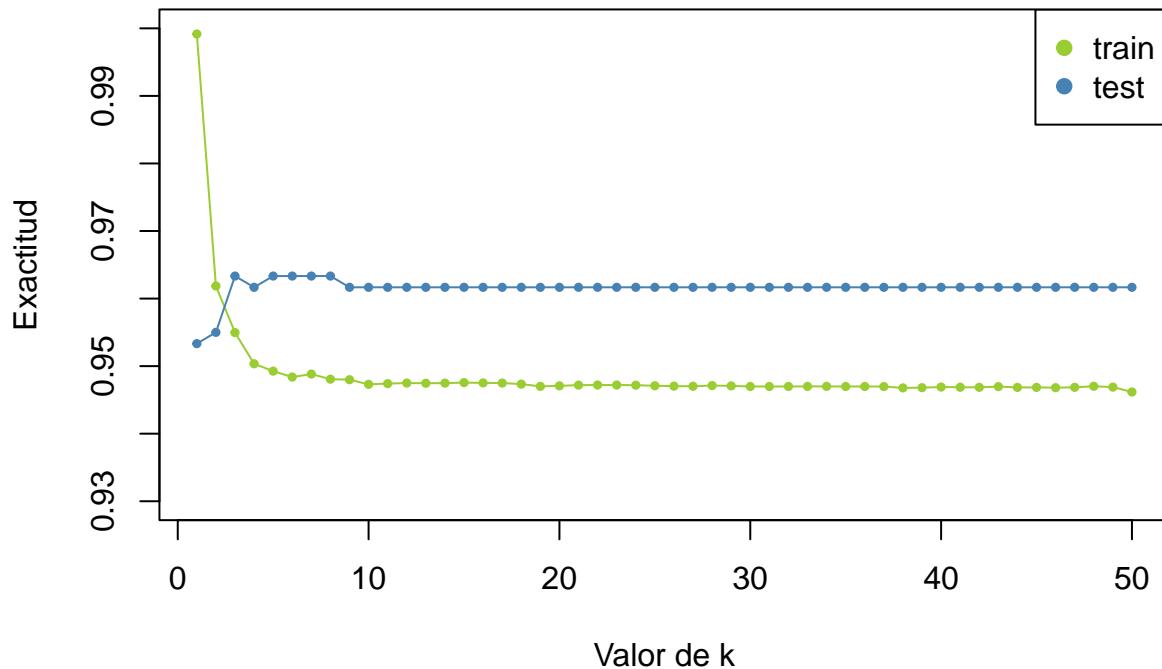
mejor_k_masayradio = which.max(exact_test_masayradio)
mejor_k_masayradio

## [1] 3

plot(valores, exact_train_masayradio, col = 'yellowgreen', pch = 19, cex = 0.5, xlab = 'Valor de k', yla
lines(valores, exact_train_masayradio, col = 'yellowgreen')
points(valores, exact_test_masayradio, col = 'steelblue', pch = 19, cex = 0.5)
lines(valores, exact_test_masayradio, col = 'steelblue')
legend('topright', legend = c('train', 'test'), col = c('yellowgreen', 'steelblue'), pch = 19)

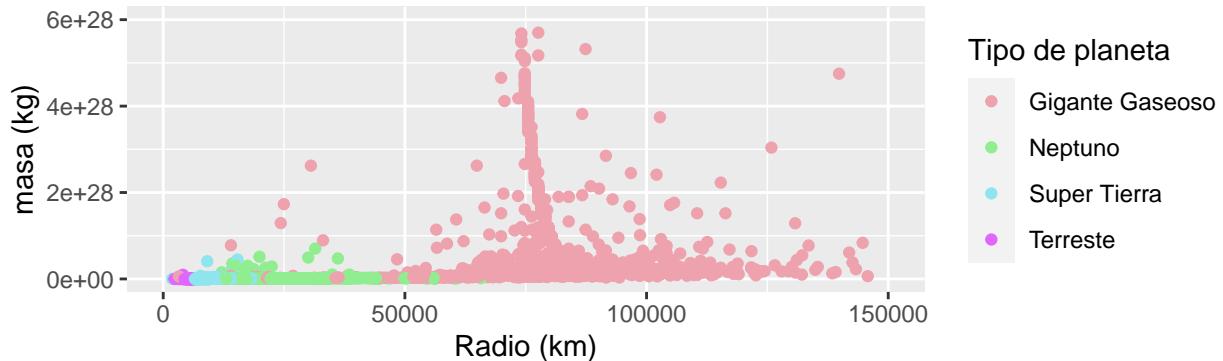
```

## Exactitud en función del k

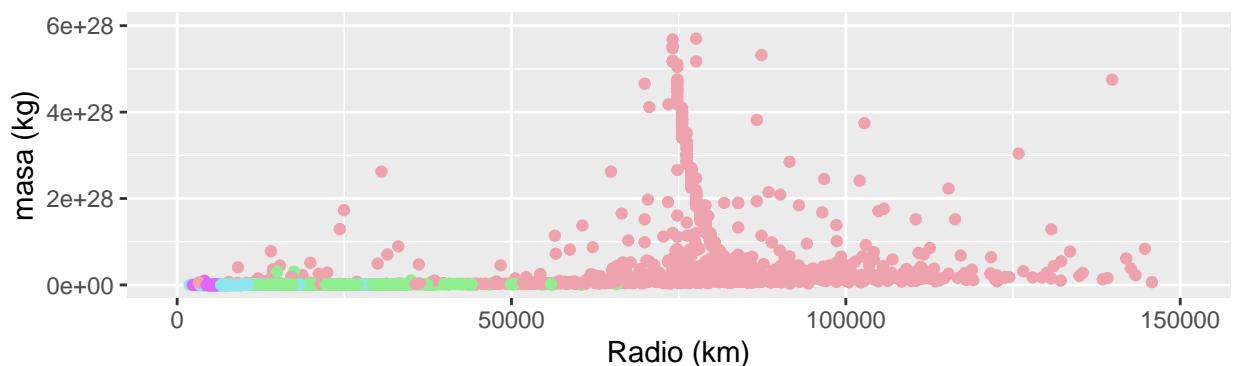


```
clasif_data_masayradio <- knn(train = data[, c(10,11)], test = data[, c(10,11)], cl = data[,4], k = mejor_k)  
summary(clasif_data_masayradio)  
  
##      Gas Giant Neptune-like   Super Earth   Terrestrial  
##        1469           1732           1385            176  
  
plot_clasif_masayradio <- ggplot(data, aes(x=radio, y=masa, color=clasif_data_masayradio)) +  
  geom_jitter(alpha=1) +  
  ylab("masa (kg)") +  
  xlab("Radio (km)") +  
  ggtitle("MASA VS. RADIO POR TIPO (CLASIFICACION CON MASA Y RADIO)") +  
  xlim(c(0, 1.5 * 10^05)) +  
  ylim(c(0, 6 * 10^28)) +  
  scale_color_manual(name = "Tipo de planeta", labels = c('Gigante Gaseoso', 'Neptuno', 'Super Tierra', 'Terrestre'))  
  theme(legend.position="none")  
  
grid.arrange(graf_radioymasa_ac, plot_clasif_masayradio)
```

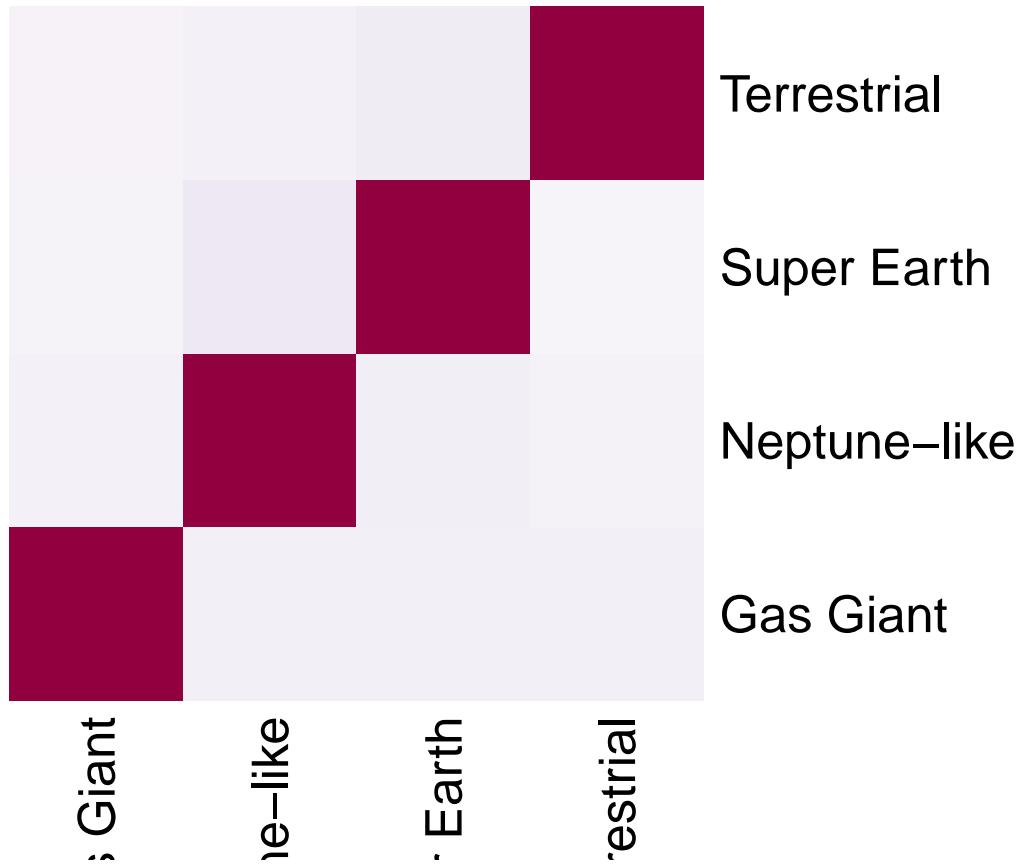
### MASA VS. RADIO SEGUN TIPO DE PLANETA (ACOTADO)



### MASA VS. RADIO POR TIPO (CLASIFICACION CON MASA Y RADIO)



```
heatmap(table(data.frame(data$planet_type,clasif_data_masayradio)),Colv = NA, Rowv = NA,col=colorRampPa
```



```
table(data.frame(data$planet_type,clasif_data_masayradio))
```

```
##                                     clasif_data_masayradio
## data.planet_type Gas Giant Neptune-like Super Earth Terrestrial
##      Gas Giant          1435            0            0            0
##      Neptune-like         14           1611           43            0
##      Super Earth          20            118          1333            5
##      Terrestrial           0             3              9          171
```

```
confusionMatrix(clasif_data_masayradio, data[,4])$overall[[1]]
```

```
## [1] 0.9554809
```

Realizando los mismos gráficos se observan resultados muy acordes a la realidad y muy similares a los obtenidos al considerar todas las variables. Al calcular la exactitud del método se obtuvo que la misma es de 0.9554809, el cual es un valor muy similar al obtenido al considerar todas las variables, por lo que se concluye que la masa y el radio son los elementos más influyentes a la hora de diferenciar los tipos de planetas. Esto se corresponde con la forma en que los exoplanetas son clasificados al ser descubiertos: la categoría de pertenencia está dada por la masa.