Exoplanets Catalogue & their methods of discovery

Harshad Kumar Elangovan - 19200349

11/08/2020

The provided dataset is an Exoplanet Catalogue which has the details of all discovered extra-solar planets by NASA. It has several variables which provides the data related the discoveries, such as - name of the discovered planet or stars, period, temperature, year of discovery, method used for discovery,etc.

We will use load the dataset and do the required analysis and produce the results based on that. First we would load all the required libraries.

```
#setting up seed
set.seed('19200349')

# Loading all the required libraries
library(readr)
library(tibble)
library(dplyr)
library(tidyr)
library(magrittr)
library(ggplot2)
library(shiny)
library(knitr)
library(ggiraph)
library(lubridate)
library(gganimate)
library(reshape2)
```

1. Import the dataset exo_data.csv as a tibble. Columns 1, 16, 17, 18, 25 should be characters. Columns 2, 14 should be factors. Column 15 should be integers. The remaining columns should be doubles.

After loading the dataset into tibble, we can see that the variables are loaded in double and character formats. So, we will update few variables as factors and use it for further analysis.

```
#  the dataset is loaded as a tibble
expodataset <- read_csv("exo_data.csv")

# Checking the format of the dataset
str(expodataset)
```

```
## tibble [3,659 x 25] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id       : chr [1:3659] "KOI-1843.03" "Kepler-974 b" "KOI-1843.02" "Kepler-9 b" ...
##  $ flag     : num [1:3659] 0 0 0 0 0 0 0 0 0 0 ...
##  $ mass     : num [1:3659] 0.0014 NA NA 0.25 0.17 0.022 0.0321 NA 0.6 5.21 ...
##  $ radius   : num [1:3659] 0.054 0.14 0.071 0.84 0.82 0.147 NA 0.192 1.24 NA ...
##  $ period   : num [1:3659] 0.177 4.194 6.356 19.224 39.031 ...
##  $ axis     : num [1:3659] 0.0048 0.039 0.052 0.143 0.229 0.0271 0.053 NA 0.0449 1.33 ...
##  $ ecc      : num [1:3659] NA NA NA 0.0626 0.0684 NA 0.06 NA NA 0.15 ...
##  $ per      : num [1:3659] NA NA NA NA NA ...
##  $ lon      : num [1:3659] NA NA NA NA NA NA NA NA NA NA ...
##  $ asc      : num [1:3659] NA NA NA NA NA NA NA NA NA NA ...
##  $ incl     : num [1:3659] 72 89.4 88.2 87.1 87.2 ...
##  $ temp     : num [1:3659] NA NA NA 707 558 ...
##  $ age      : logi [1:3659] NA NA NA NA NA NA ...
##  $ meth     : chr [1:3659] "transit" "transit" "transit" "transit" ...
##  $ year     : num [1:3659] 2012 NA NA 2010 2010 ...
##  $ recency  : chr [1:3659] "13/07/15" "17/11/28" NA "15/12/03" ...
##  $ r_asc    : chr [1:3659] "19 00 03.14" "19 00 03.14" "19 00 03.14" "19 02 17" ...
##  $ decl     : chr [1:3659] "+40 13 14.7" "+40 13 14.7" "+40 13 14.7" "+38 24 03" ...
##  $ dist     : num [1:3659] NA NA NA 650 650 ...
##  $ host_mass: num [1:3659] 0.52 0.52 0.52 1.07 1.07 1.07 0.69 0.83 1.07 0.82 ...
##  $ host_rad : num [1:3659] 0.5 0.5 0.5 1.02 1.02 1.02 NA 0.79 NA NA ...
##  $ host_met : num [1:3659] 0.07 0.07 0.07 0.12 0.12 0.12 NA -0.01 -0.02 -0.18 ...
##  $ host_temp: num [1:3659] 3687 3687 3687 5777 5777 ...
##  $ host_age : num [1:3659] NA NA NA NA NA NA NA NA NA NA ...
##  $ lists    : chr [1:3659] "Controversial" "Confirmed planets" "Controversial" "Confirmed
planets" ...
##  - attr(*, "problems")= tibble [2 x 5] (S3: tbl_df/tbl/data.frame)
##   ..$ row     : int [1:2] 1712 2970
##   ..$ col     : chr [1:2] "age" "age"
##   ..$ expected: chr [1:2] "1/0/T/F/TRUE/FALSE" "1/0/T/F/TRUE/FALSE"
##   ..$ actual  : chr [1:2] "0.0055" "3.0"
##   ..$ file    : chr [1:2] "'exo_data.csv'" "'exo_data.csv'"
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_character(),
##   ..   flag = col_double(),
##   ..   mass = col_double(),
##   ..   radius = col_double(),
##   ..   period = col_double(),
##   ..   axis = col_double(),
##   ..   ecc = col_double(),
##   ..   per = col_double(),
##   ..   lon = col_double(),
##   ..   asc = col_double(),
##   ..   incl = col_double(),
##   ..   temp = col_double(),
##   ..   age = col_logical(),
##   ..   meth = col_character(),
##   ..   year = col_double(),
##   ..   recency = col_character(),
##   ..   r_asc = col_character(),
##   ..   decl = col_character(),
##   ..   dist = col_double(),
##   ..   host_mass = col_double(),
##   ..   host_rad = col_double(),
##   ..   host_met = col_double(),
```

```
##   ..   host_temp = col_double(),
##   ..   host_age = col_double(),
##   ..   lists = col_character()
##   .. )
```

```
# printing the column names for required variables
colnames(expodataset[,c(1,2,14,15,16,17,18,25)])
```

```
## [1] "id"      "flag"    "meth"    "year"    "recency" "r_asc"   "decl"
## [8] "lists"
```

```
# converting the required columns as factor
expodataset$flag %<>% as.factor
expodataset$meth %<>% as.factor


# converting the required columns to integer
expodataset$year %<>% as.integer

# converting the required columns to character
expodataset$id %<>% as.character
expodataset$recency %<>% as.character
expodataset$r_asc %<>% as.character
expodataset$decl %<>% as.character
expodataset$lists %<>% as.character

# all other columns are already in col_double() format.
```

2. Exclude the exoplanets with an unknown method of discovery

We will exclude the rows of the exoplanets with an unknown method of discovery.

```
# We can drop rows containing missing values (unknown) of the method variable using is.na fun
ction

length(expodataset$meth)
```

```
## [1] 3659
```

```
expodataset <- expodataset[!is.na(expodataset$meth), ]
length(expodataset$meth)
```

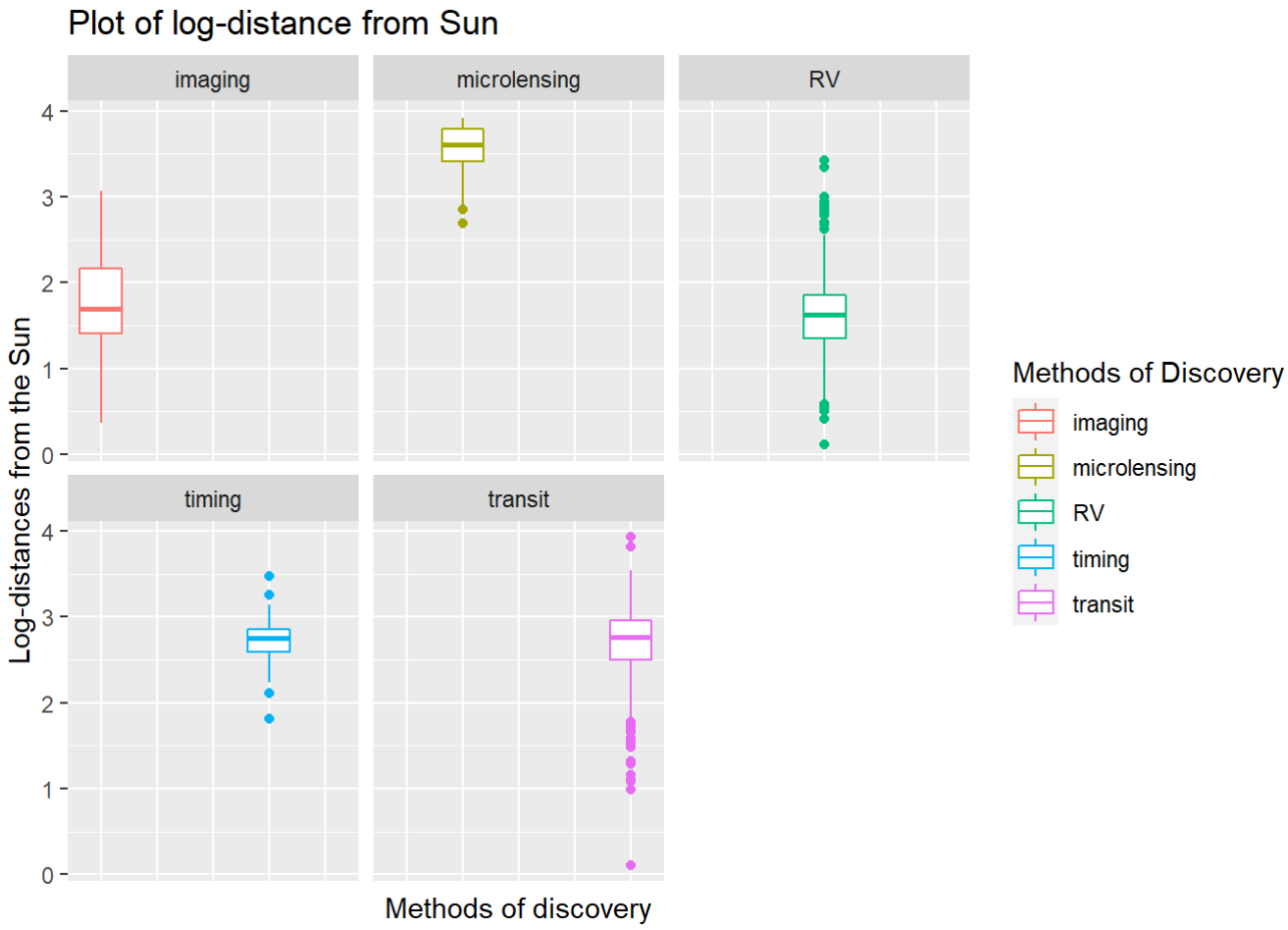
```
## [1] 3596
```

```
# From the length difference, we see that around 63 rows are removed from the dataset.
```

3. Create a graphic which illustrates the relationship between the log-distances from the Sun and the methods of discovery.

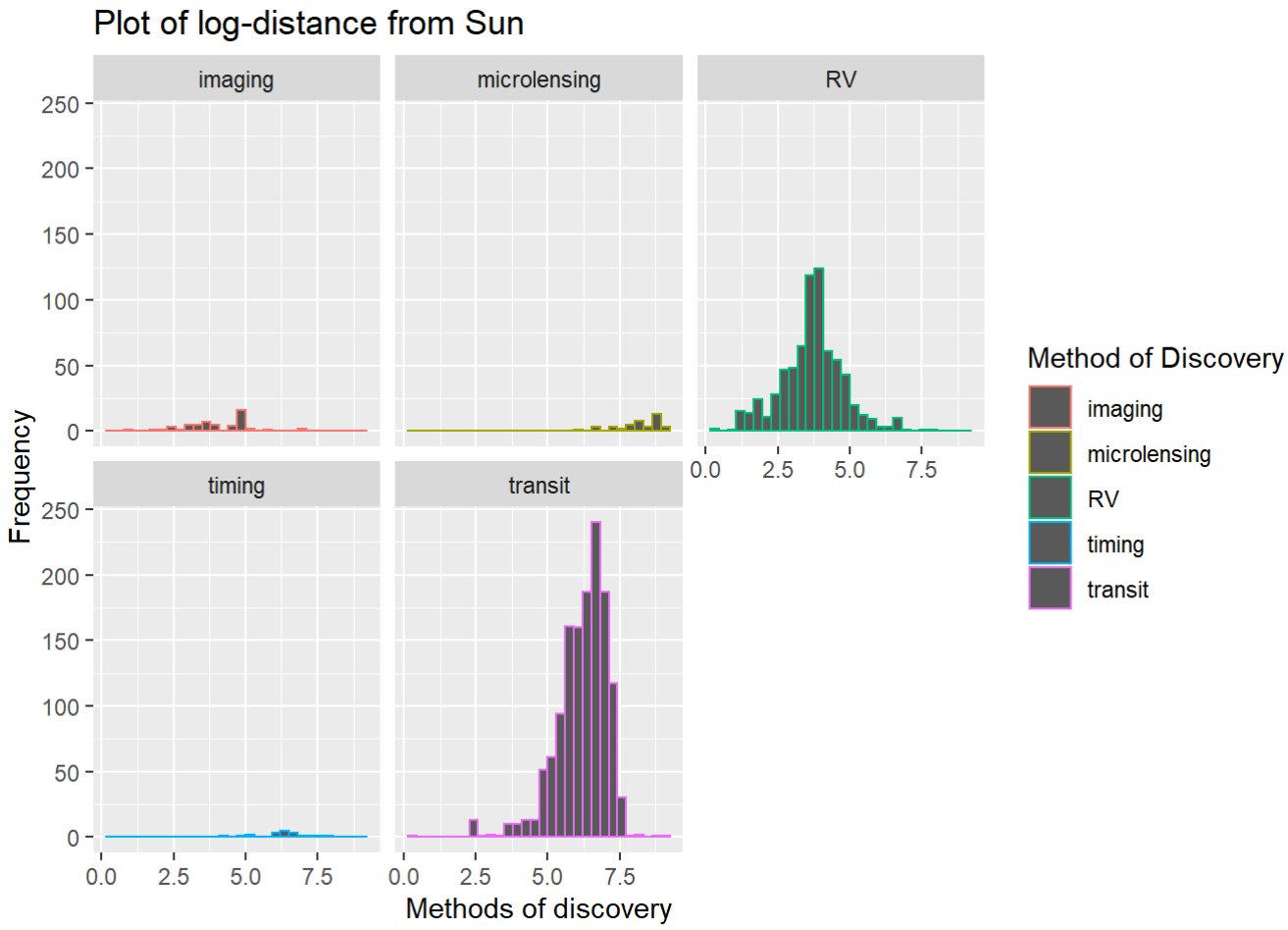
We will create the graphics for illustrating the relationship between the distance of the exoplanets from sun and the methods used for the discovery.

```
# Box plot of the dataset
ggplot(expodataset, aes(meth,log10(dist),col=meth)) +
  geom_boxplot(na.rm = T) +
  facet_wrap(~meth) +
  labs(
    title = "Plot of log-distance from Sun",
    x = 'Methods of discovery',
    y = 'Log-distances from the Sun') +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  scale_color_discrete(name = "Methods of Discovery")
```



The plot shows that relationship of methods such as imaging, microlensing, transit, etc. with the log distance of the exoplanets from the sun. The methods such as microlensing, transit, RV have a very clear relationship with distances from sun. We can check the relationships using histogram graphics aswell.

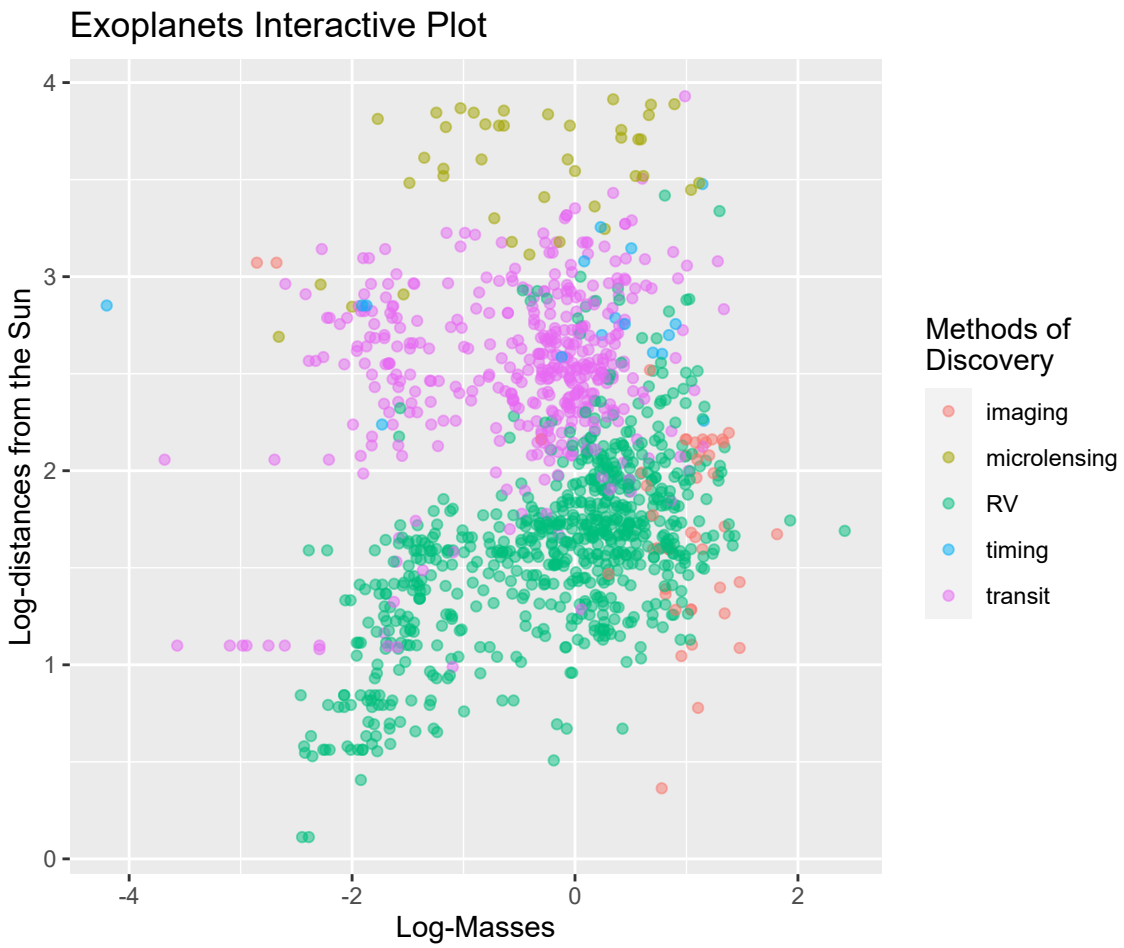
```
# Histogram of the dataset
ggplot(expodataset, aes(x = log(dist),col = meth)) +
  geom_histogram(position = "identity",na.rm = T,bins = 30) +
  facet_wrap(~meth) +
  labs(
    title = "Plot of log-distance from Sun",
    x = 'Methods of discovery',
    y = 'Frequency') +
   scale_color_discrete(name = "Method of Discovery")
```



4. Create scatterplots of the log-mass versus log-distances, separating by methods of discovery. Hovering with the cursor highlights the point and displays its name, and, if you click, the exoplanet's page on the Open Exoplanet Catalogue will be opened. (paste the id after http://www.openexoplanetcatalogue.com/planet/ (http://www.openexoplanetcatalogue.com/planet/)).

In this plot, each scattered points have a link which shows the details of the particular exoplanet. This is done using interactive function in ggplot.

```
expodataset$link <- sprintf("window.open(\"%s%s\")",
                            "http://www.openexoplanetcatalogue.com/planet/",
                            as.character(expodataset$id))

g2 <- ggplot(expodataset, aes(log10(mass),log10(dist), colour = as.factor(meth))) +
  labs(
    title = "Exoplanets Interactive Plot",
    x = 'Log-Masses',
    y = 'Log-distances from the Sun') +
    scale_color_discrete(name="Methods of \nDiscovery") +
    geom_point_interactive(aes(tooltip = id,
                               data_id = id,
                               onclick = link),alpha=0.5,
                           size = 1.5,na.rm=T)

ggiraph(code = print(g2), width = 0.8,tooltip_opacity = 0.5)
```



5. Rename the radius into jupiter_radius, and create a new column called earth_radius which is 11.2 times the Jupiter radius.

```
#checking the column name of the fourth variable
colnames(expodataset[,4])
```

```
## [1] "radius"
```

```
expodataset <- expodataset %>% rename(jupiter_radius = radius)

#checking the column name of the fourth variable after change
colnames(expodataset[,4])
```

```
## [1] "jupiter_radius"
```

```
expodataset <- expodataset %>% mutate(earth_radius = 11.2 * jupiter_radius)
```

6. Focus only on the rows where log-earth radius and log-period have no missing values, and perform kmeans with four clusters on these two columns.

The required data are saved to a new dataset which is then used for clustering the data. The model creates clusters which is stored in ‘kmeansmodel’. This model is used for understanding the clusters of the exoplanets dataset provided.

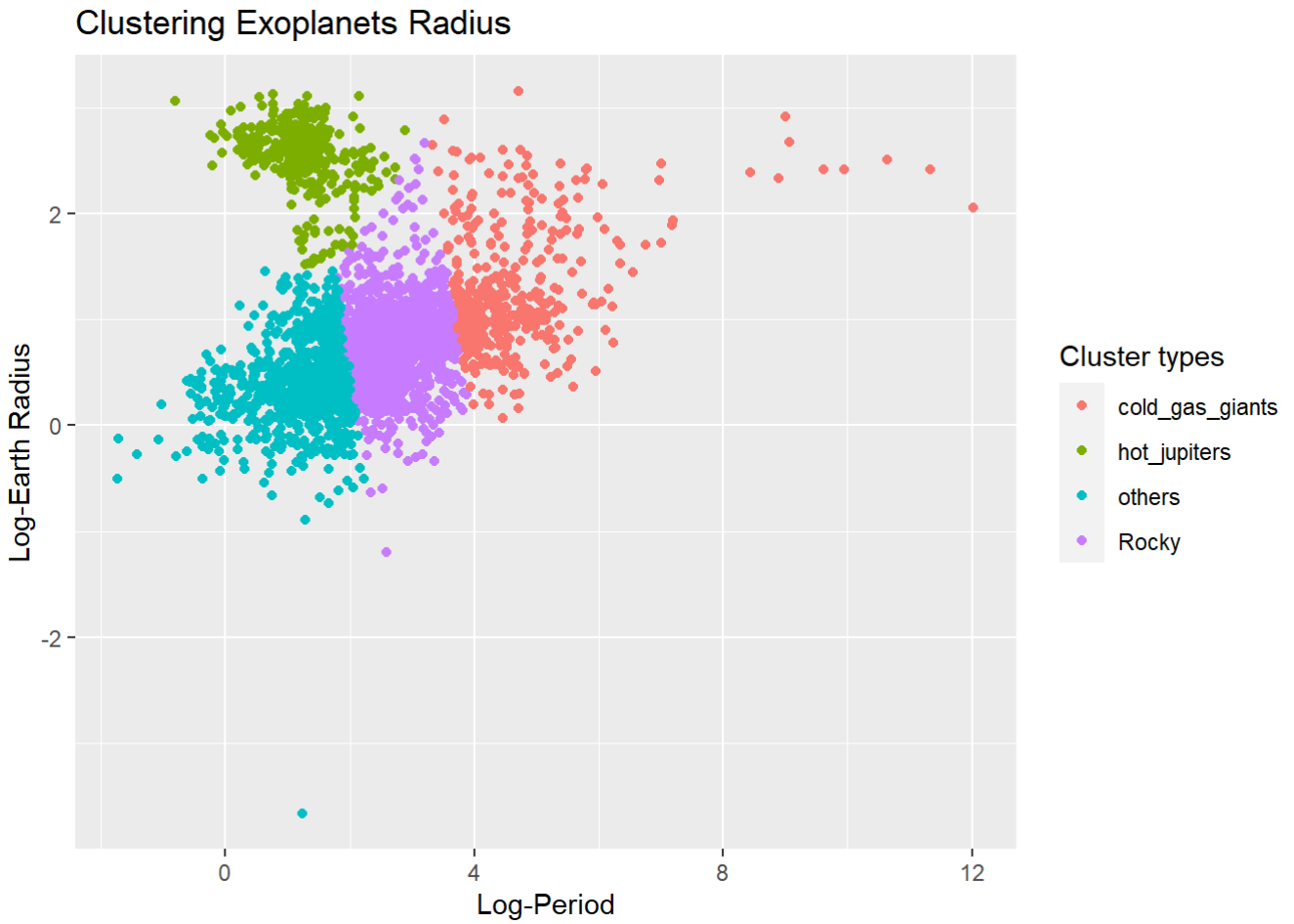
```
#Removing the null values before clustering
expodataset1 <- expodataset %>% drop_na(earth_radius, period)

# New dataset is created required for clustering
radius_dataset <- expodataset1 %>% transmute(period = log(period),earth_radius = log(earth_ra
dius))

#Clusters on the data is created using kmeans function with four centers.
kmeansmodel <- kmeans(radius_dataset, centers = 4)

# Table of Clusters
table(kmeansmodel$cluster)
```

```
##
##    1    2    3    4
##  798  385  416 1133
```

7. Add the clustering labels to the dataset through a new factor column called type, with levels rocky, hot_jupiters, cold_gas_giants, others; similarly to https: //en.wikipedia.org/wiki/Exoplanet#/media/File:ExoplanetPopulations-20170616 (File:ExoplanetPopulations-20170616). png and produce the scatterplot highlighting these clusters.

We can use the output of the dataset for interpretting the levels of the exoplanets. The output is used for creating a new label for the exoplanets based on the clusters and a graphic representation is done. This plot shows the spread of data among the four clusters.

```
# the cluster values are converted to a factor
kmeansmodel$cluster <- as.factor(kmeansmodel$cluster)

# this factor is then updated in the dataset each each exoplanets.
expodataset1$type <- kmeansmodel$cluster
expodataset1$type <- as.numeric(expodataset1$type)

# we name each cluster based on the wikipedia link provided
expodataset1$type[expodataset1$type == 1] <- "others"
expodataset1$type[expodataset1$type == 2] <- "hot_jupiters"
expodataset1$type[expodataset1$type == 3] <- "cold_gas_giants"
expodataset1$type[expodataset1$type == 4] <- "Rocky"

# plot of the dataset
ggplot(expodataset1, aes(log(period), log(earth_radius), color = type)) +
  geom_point() +
  xlab('Log-Period') +
  ylab('Log-Earth Radius') +
  labs(title="Clustering Exoplanets Radius") +
  scale_color_discrete(name="Cluster types")
```



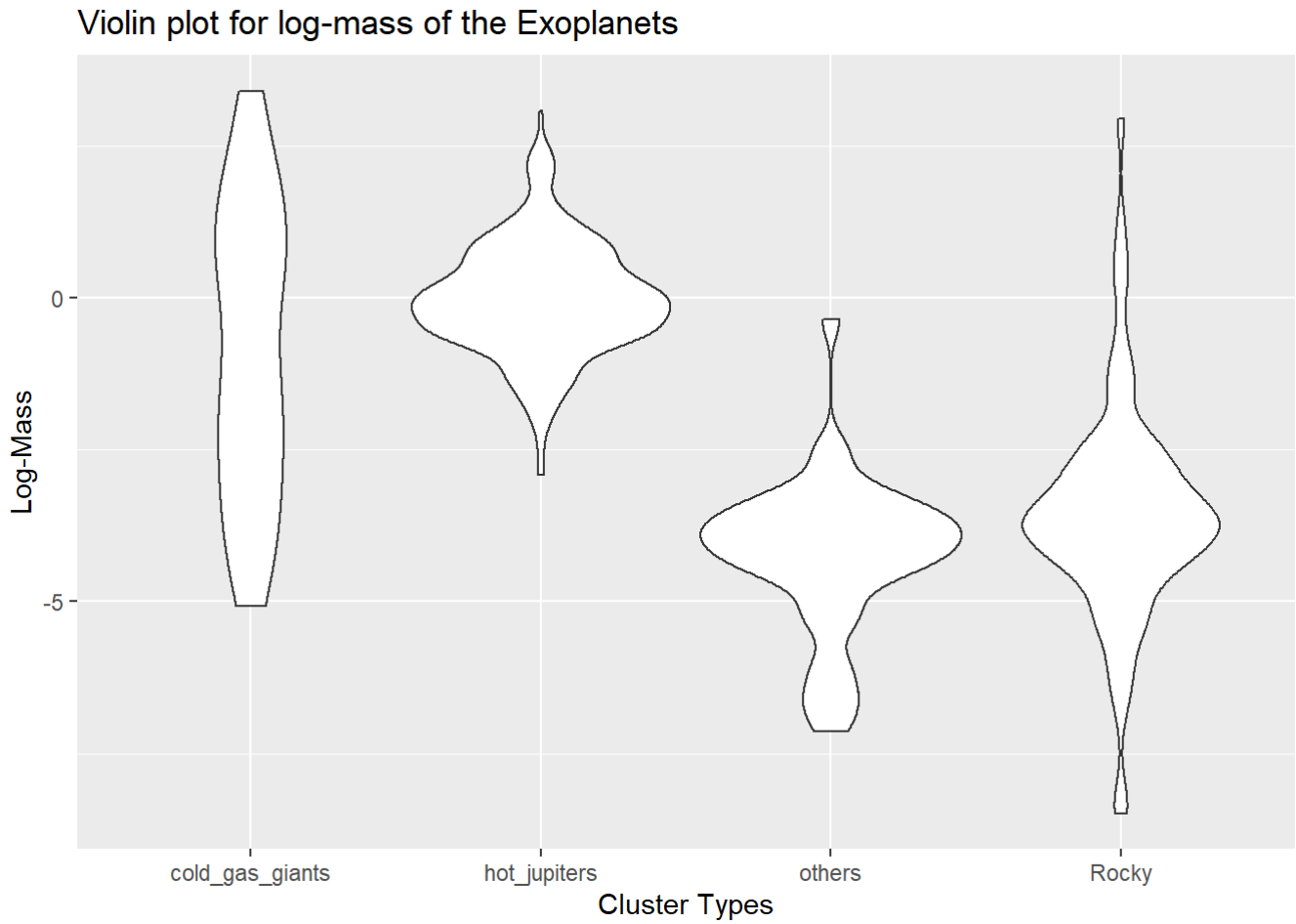

```
# CLustering Labels
table(expodataset1$type)
```

```
##
## cold_gas_giants    hot_jupiters          others           Rocky
##             416             385             798            1133
```

8. Use a violin plot to illustrate how these clusters relate to the log-mass of the exoplanet.

The violin plot is similar to the box plot whcih we came across in part 3. This plot describes about the median and quantiles of the cluster types of teh dataset. From this output, we see that most of the exoplanets fall under rocky type and it has the maximum count of the exoplanets.

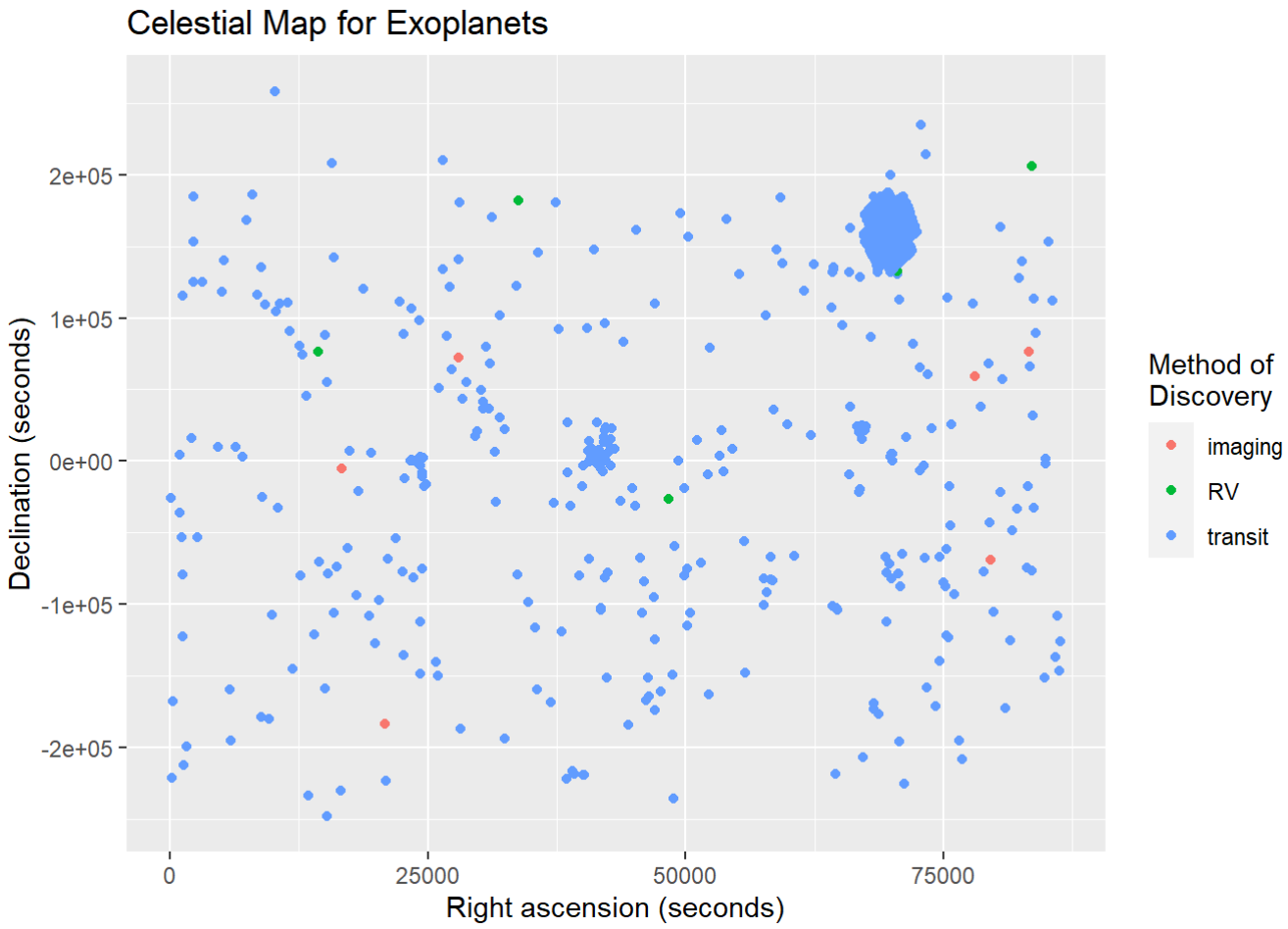
```
#violin plot for the dataset
ggplot(expodataset1, aes(x = type, y = log(mass))) +
      geom_violin(na.rm=TRUE) +
  xlab('Cluster Types') +
  ylab('Log-Mass') +
      labs(title = "Violin plot for log-mass of the Exoplanets")
```



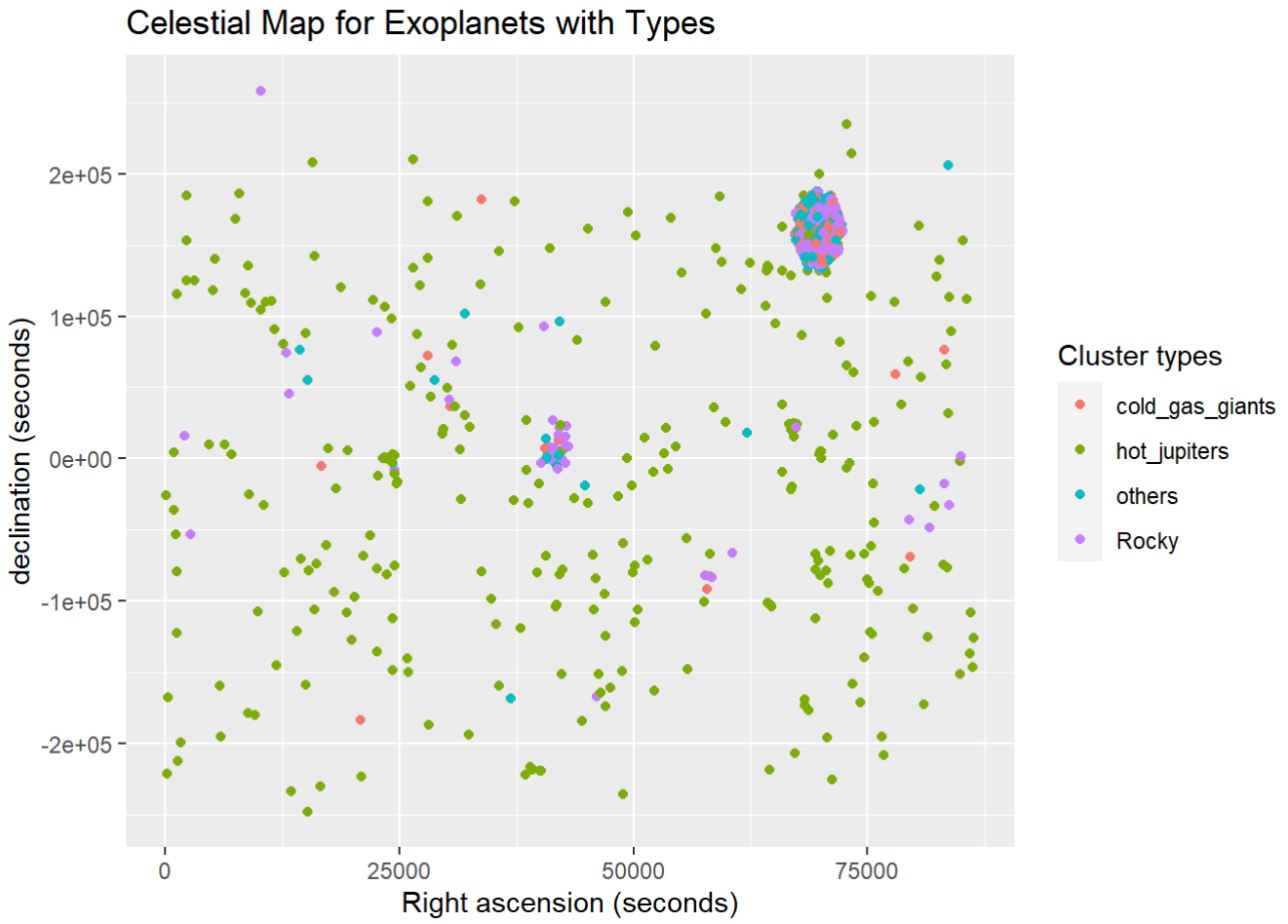
9. Transform r_asc and decl into two new variables that are the same variables but in values of seconds. Use these as coordinates to represent a celestial map for the exoplanets.

The Right ascension and declination variables contains the degrees in characters. So, it is converted to time format from character format and then converted to seconds. Once both the variables are converted to seconds, it is used for representing the celestrial map for the exoplanets.

```
#We would remove the empty rows from the dataset
expodataset1 <- expodataset1 %>% drop_na(r_asc,decl)

# data is splitted using gsub function and then converted to time format and then converted t
o seconds using period_to_seconds.
expodataset1$r_asc1 <- gsub(" ", ":", expodataset1$r_asc, fixed = TRUE)
expodataset1$r_asc1 <- hms(expodataset1$r_asc1)
expodataset1$r_asc1 <- period_to_seconds(expodataset1$r_asc1)

# data is splitted using gsub function and then converted to time format and then converted t
o seconds using period_to_seconds.
expodataset1$decl1 <- gsub(" ", ":", expodataset1$decl, fixed = TRUE)
expodataset1$decl1 <- hms(expodataset1$decl1)
expodataset1$decl1 <- period_to_seconds(expodataset1$decl1)

# The calculated variables are then used for representing the celestrial map for the exoplane
ts
ggplot(expodataset1, aes(r_asc1, decl1, color= meth)) +
  geom_point() +
  labs(title="Celestial Map for Exoplanets",
       x="Right ascension (seconds)",
       y="Declination (seconds)") +
  scale_color_discrete(name="Method of \nDiscovery")
```

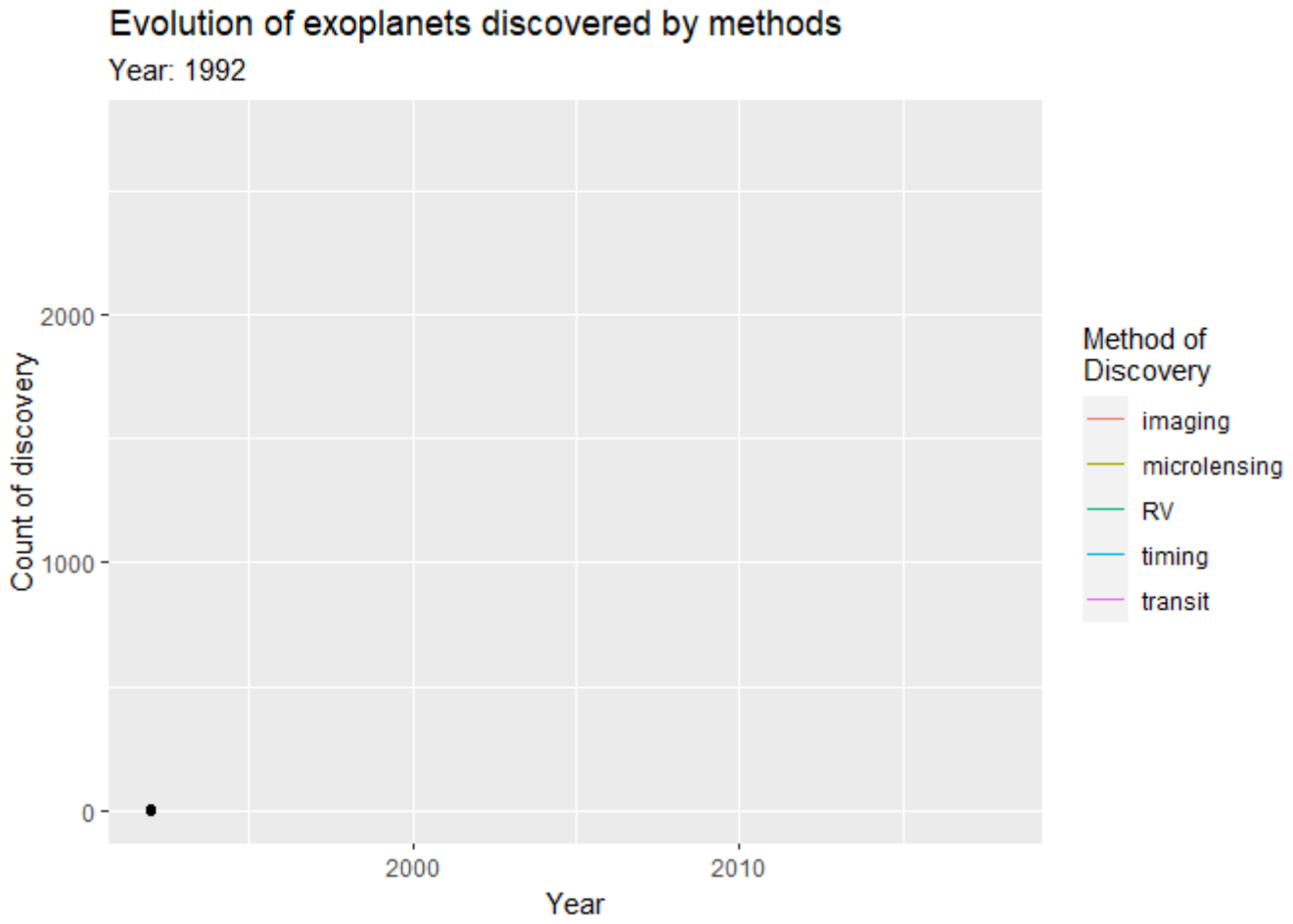



```
#plotting with cluster types
ggplot(expodataset1, aes(r_asc1, decl1, color= type)) +
  geom_point() +
  labs(title="Celestial Map for Exoplanets with Types",
       x="Right ascension (seconds)",
       y="declination (seconds)") +
  scale_color_discrete(name="Cluster types")
```

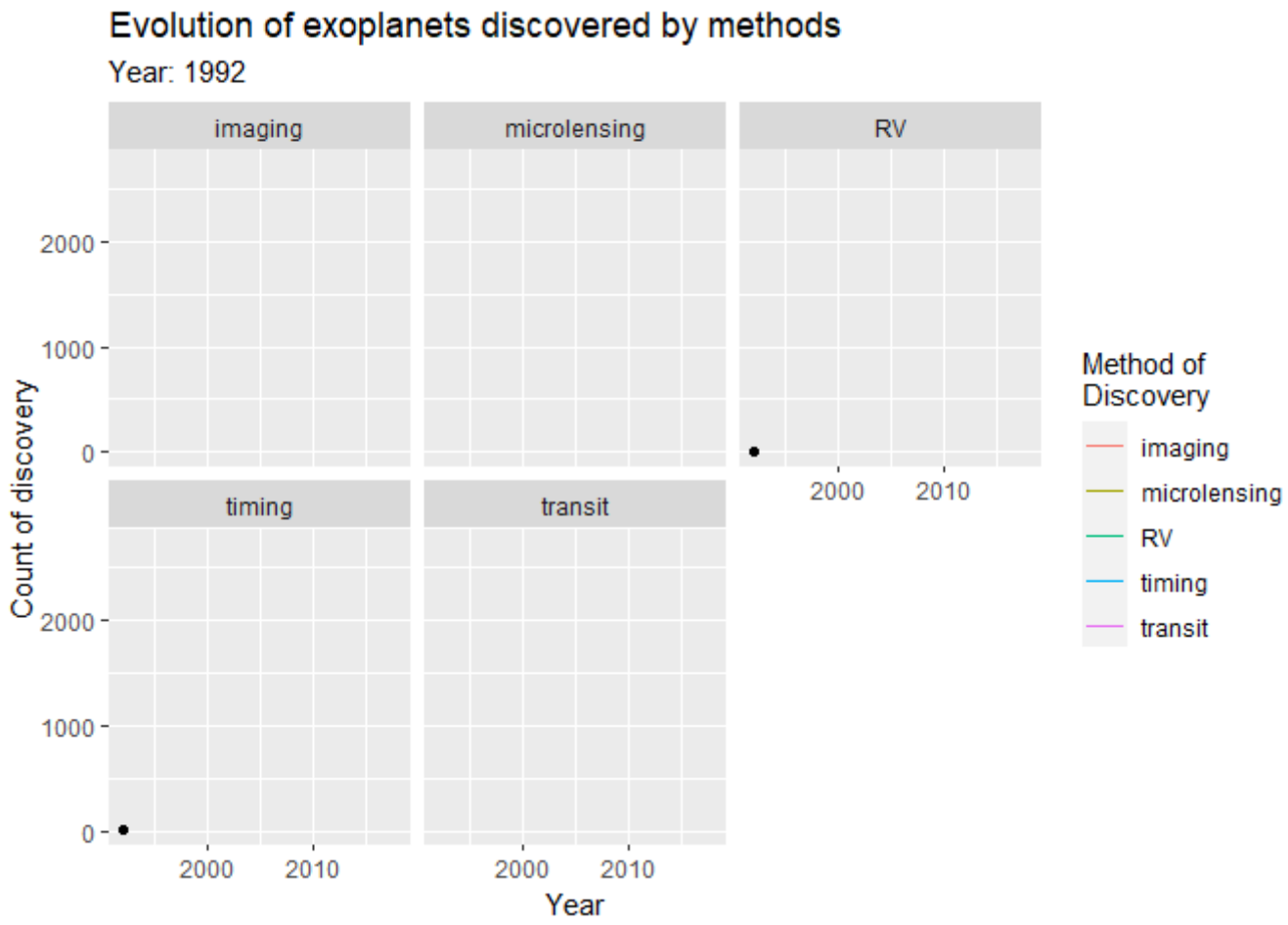


10. Create an animated time series where multiple lines illustrate the evolution over time of the total number of exoplanets discovered for each method up to that year.

```
# A new variable is created which would be used in animating the values
exoplanets_count <- expodataset %>%
            group_by(meth, year) %>%
            summarise(Count = length(meth)) %>%
            mutate(Count = cumsum(Count))

# we remove any empty values in the cariable
exoplanets_count <- na.omit(exoplanets_count)

# the time series plot is created with the new variable
time_series<-ggplot(exoplanets_count, aes(x = year, y = Count, group = meth)) +
  geom_point() +
  geom_line(aes(color = meth)) +
  transition_reveal(year) +
  labs(title = 'Evolution of exoplanets discovered by methods',
       subtitle = 'Year: {frame_along}',
       x = 'Year',
       y = 'Count of discovery') +
  scale_color_discrete(name="Method of \nDiscovery")

# the plot is rendered to create animation
animation <- animate(time_series, renderer = gifski_renderer(loop = T))
animation
```




```
# The individual method animation plot can be creating by adding facet_wrap to the ggplot
time_series1<-ggplot(exoplanets_count, aes(x = year, y = Count, group = meth)) +
  geom_point() +
  geom_line(aes(color = meth)) +
  facet_wrap(~meth) +
  transition_reveal(year) +
  labs(title = 'Evolution of exoplanets discovered by methods',
       subtitle = 'Year: {frame_along}',
       x = 'Year',
       y = 'Count of discovery') +
  scale_color_discrete(name="Method of \nDiscovery")

# the plot is rendered to create animation
animation1 <- animate(time_series1, renderer = gifski_renderer(loop = T))
animation1
```

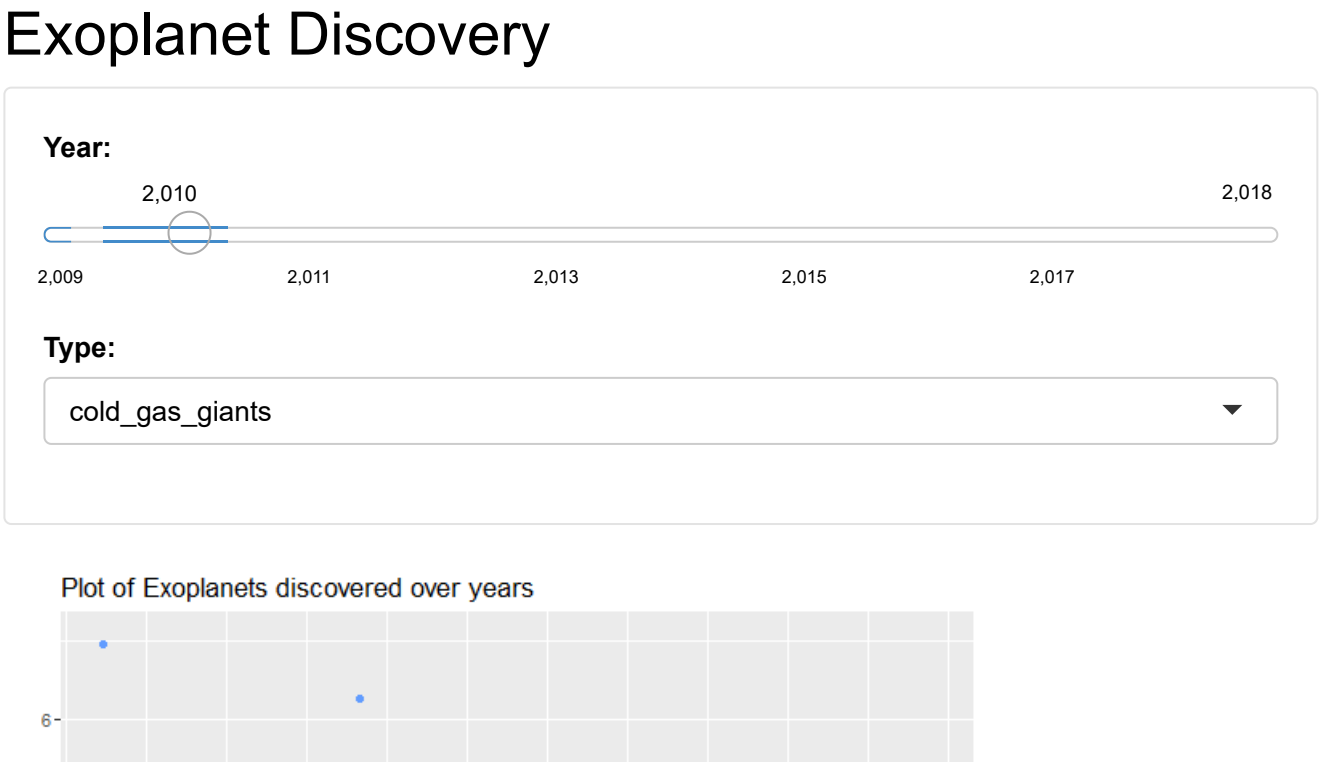


11. Create an interactive plot with Shiny where you can select the year (slider widget, with values ≥ 2009) and exoplanet type. Exoplanets appear as points on a scatterplot (log-mass vs log-distance coloured by method) only if they have already been discovered. If type is equal to all all types are plotted together.

Shiny can be used for creating an interactive plot with default values in the plot. In this plot, default values of year and type is added. Once the plot is created, we can update the plot by updating the value of year or cluster type in the interactive mode and update the plot as required.

```
#max year value
maxyr <-  max(expodataset1$year,na.rm = T)

ui <- shinyUI(  #UI function for front end
    fluidPage(
      # Application title
    titlePanel("Exoplanet Discovery"),

    # Sidebar with a slider input for Year and Type of exoplanet
    sidebarLayout(
        sidebarPanel(
            sliderInput("year",       # Name of the input slider
                        "Year:",      # Label
                        min = 2009,   # minimum value of the slider
                        max = maxyr,  # max value of the slider
                        value = 2010, # initial value
                        step = 1),    # increment of the input by 1

            selectInput("type",       # Name of input choice
                        label = "Type:",    # Label
                        choices = c('hot_jupiters', 'cold_gas_giants', 'Rocky', 'others','Al
l'),   # Combination of choices
                        selected = 'cold_gas_giants')     # default value
        ),

        # Show a plot of the generated scatter points
        mainPanel(
           plotOutput("scatterplot")   # scatterplot display
        )
    )
  )
)

server <- function(input, output) {          #server function for backend
    output$scatterplot <- renderPlot({

        # generating dataset based on inputs from ui block
        x <- reactive({
                      if(input$type != 'All'){
                        expodataset1[((expodataset1$year <= input$year) & (expodataset1$type
 == input$type)),]
                      }
                      else {
                        expodataset1[(expodataset1$year <= input$year),]
                      }

                      })

        # plotting function based on the dataset
        ggplot(x(), aes(x= log(mass),y = log(dist), color = meth)) +
          geom_point(na.rm=TRUE) +
            labs(
                 title = "Plot of Exoplanets discovered over years",
                 x = 'Log - Mass',
                 y = 'Log - Distance') +
          scale_color_discrete(name="Method of \nDiscovery")
```

```
    })
}
# function for running the shiny app
shinyApp(ui, server)
```



12. Fit a linear regression model where log period is the response variable and the logs of host_mass, host_temp and axis are the covariates (exclude rows that contain at least one missing value). Include an intercept term in the regression model.

```
# dropping empty values
expodataset2 <- expodataset1 %>% drop_na(mass,temp,axis)

#fitting the linear regression model
fitlm <-lm(log(period) ~ log(mass) + log(temp) + log(axis), data = expodataset2)
#printing the summary of the fit
summary(fitlm)
```

```
##
## Call:
## lm(formula = log(period) ~ log(mass) + log(temp) + log(axis),
##     data = expodataset2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28421 -0.06576 -0.02123  0.02683  0.60051
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.795781   0.159526  48.869  < 2e-16 ***
## log(mass)   -0.031775   0.005475  -5.804 1.77e-08 ***
## log(temp)   -0.307241   0.023473 -13.089  < 2e-16 ***
## log(axis)    1.406615   0.008817 159.532  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1468 on 277 degrees of freedom
## Multiple R-squared:  0.9919, Adjusted R-squared:  0.9918
## F-statistic: 1.127e+04 on 3 and 277 DF,  p-value: < 2.2e-16
```

From the summary statistics of the fitted model on log-period with covariates of log-mass,log-temp and log-axis, we see that all the three covariates are highly significant as 99% confidence level. A change in any of these three variables will impact the response variable period of the exoplanets.

The linear model expression of the given model is,

```
#code block for printing the fit expression
  cat("log(period) =",fitlm$coefficients[1],fitlm$coefficients[2],"*log(mass)",fitlm$coeffici
ents[3],"*log(temp) + ",fitlm$coefficients[4],"*log(axis)")
```

```
## log(period) = 7.795781 -0.03177475 *log(mass) -0.3072415 *log(temp) +  1.406615 *log(axis)
```

So, one unit increase in mass or temp will have a negative impact on the period of the exoplanets,keeping all other elements constant. With one unit increase in axis variable will have a positive impact on the period response variable,keeping all other variables constant.

The period variable in this dataset depicts the orbital period of the planet. From this model fit, we see that the covariates - mass, temp and axis are highly significant to the response variable period. This depicts that the orbital period of the planet will be affected if there are any change in the mass or temperature or axis of the planet.

This model has an adjusted R-squared value as 0.9918. This shows that model is fitted very well and all the three covariates contribute very much to the response variable.

13. Include in your RMarkdown document some model summaries and an interpretation of the model you have fit.

```
#Anova table
anova(fitlm)
```

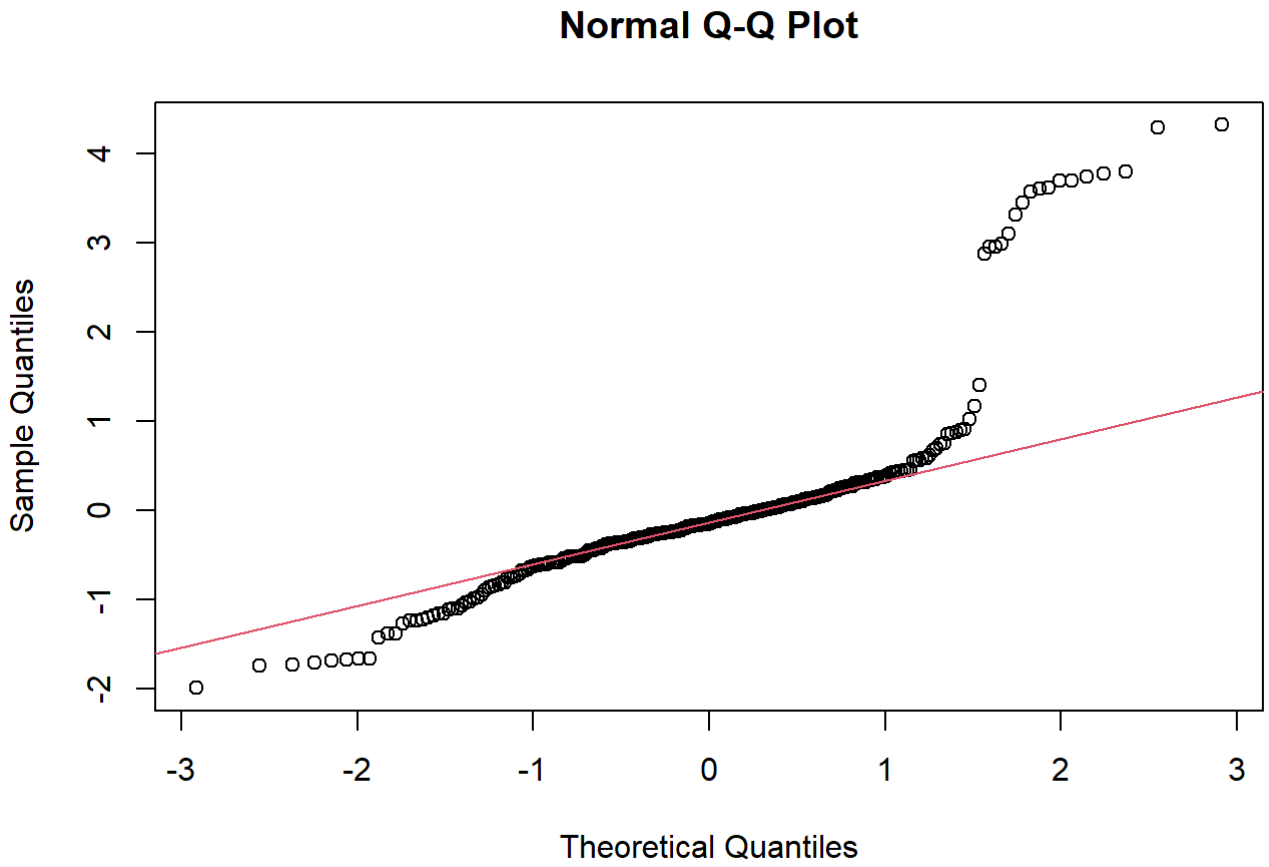
```
## Analysis of Variance Table
##
## Response: log(period)
##            Df Sum Sq Mean Sq   F value    Pr(>F)
## log(mass)   1   0.43    0.43    19.979 1.142e-05 ***
## log(temp)   1 179.75  179.75  8343.827 < 2.2e-16 ***
## log(axis)   1 548.26  548.26 25450.392 < 2.2e-16 ***
## Residuals 277   5.97    0.02
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type 1 Analysis of Variance determines whether there is any statistically significant difference in the model by introducing new variables one by one to the model. The anova table shows that there is high significance by adding log - mass to the model. So it can be added to the model. Now, the second variable log - temp is added to the existing model and the table is compared to check if the new variable fits in the model. If the variable fits, then the next variable until last variable is added to the existing model and compared. So, it works in a sequential way with one variable at a time.

From the above table, all the three covariates have a higher significance on the response variable. So, these covariates can be used to model the dataset.

We can have a look at the qqplot to check the linear regression fit of the covariates on the response variable log - period.

```
#this can be used for representing several plots related to residuals and diagnostics of the
 model.
#plot(fitlm)

#QQ Plot
qqnorm(rstudent(fitlm))
qqline(rstudent(fitlm),col=2)
```



From the QQ plot, we see that the model is fit linearly and it is a best fit with the residuals from the provided covariates. But, it is also noted that there are few outliers in the dataset which can be removed if required. There points can be verified using the Cook's distance method. If there are outliers in the dataset, it can be removed and the model can be refitted with the updated values of the dataset.

We can also produce different models with more covariates and check the efficiency of this model using AIC,BIC or Chi squared tests and find the best model.

14. Embed the Shiny app from (11) in your RMarkdown document.

The Shiny app is created using interactive plots is embedded in the part 11. This plot can be visualized by running the rmd document in the html format and relevant values for years and exoplanets type can be adjusted in the interactive version of the plot.