18201501_Final_Project

Azam Jainullabudin Mohamed & 18201501

9/1/2019

Question 1

```
# Importing the exoplanet dataset
exodata <- read.csv("C:\\Users\\DELL\\Downloads\\exo_data.csv")
# class(exodata)

# Typcasting the data frame in to tibble to make it more
# convinient for large dataset
exodata <- exodata %>% as_tibble()

# Modifying the datatype to character format
exodata$id      <-as.character(exodata$id)
exodata$recency <-as.character(exodata$recency)
exodata$r_asc   <-as.character(exodata$r_asc)
exodata$decl    <-as.character(exodata$decl)
exodata$lists   <-as.character(exodata$lists)

# Modifying the datatype to factor format
exodata$flag    <-as.factor(exodata$flag)
exodata$meth    <-as.factor(exodata$meth)

# Modifying the datatype to integer format
exodata$year    <-as.integer(exodata$year)
```

Question 2

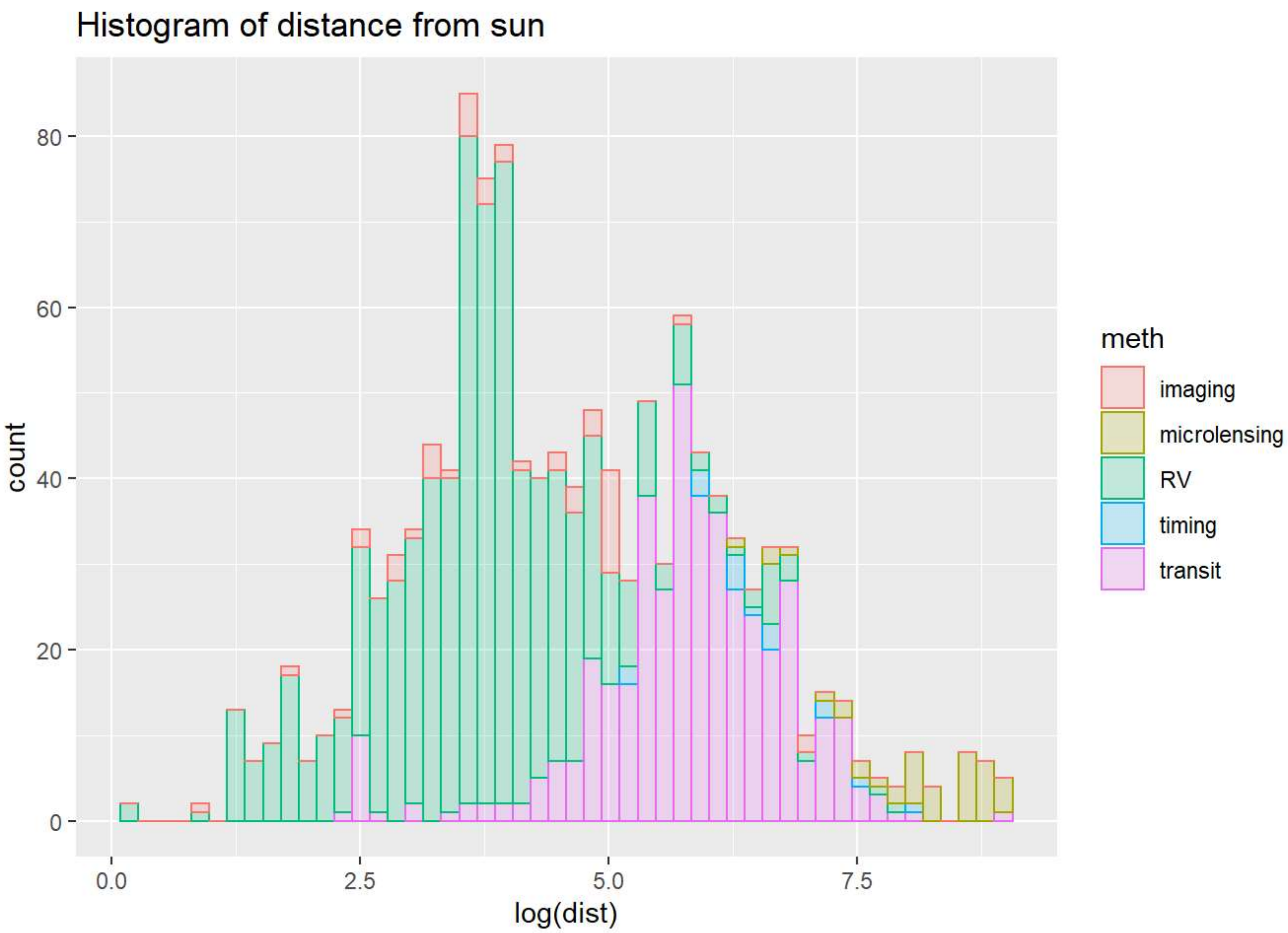
```
# Locating the whitespace characters in methodology column
# and updating it as NA
for (i in 1:nrow(exodata)) {
  if (exodata$meth[i] == "") exodata$meth[i] = NA
}

# length(exodata$meth)
# Dropping the NA's from methodology column
exodata <- exodata %>% drop_na(meth)
# length(exodata$meth)
```

Question 3

```
# Dropping the NA's from column distance from the sun
exodata <- exodata %>% drop_na(dist)
exodata <- exodata %>% drop_na(mass)

# Histogram plot of log distance from the sun differentiating
# using methodology
ggplot(data=exodata, aes(x= log(dist), fill=meth, color=meth)) +
  ggtitle("Histogram of distance from sun") +
  geom_histogram(alpha=I(.2), stat = "bin", bins = 50)
```

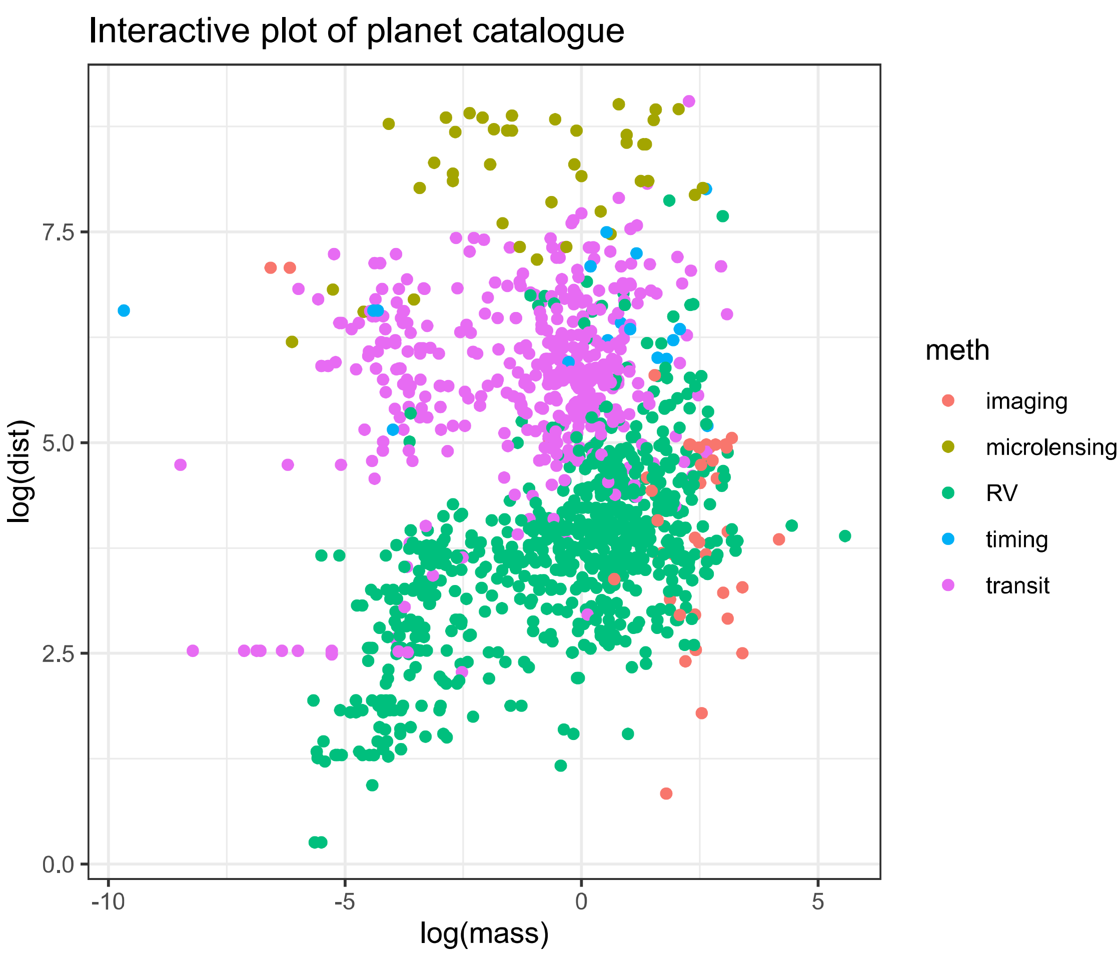


Question 4

```
# Dropping the NA's from mass column
exodata <- exodata %>% drop_na(mass)

# Updating the open planet catalogue page based upon its id column
exodata$onclick <- sprintf("window.open(\"%s%s\")",
                           "http://www.openexoplanetcatalogue.com/planet/",
                           as.character(exodata$id))

# Creating an interactive plot that highlights the details on hovering
# through the plot
p = ggplot(exodata,aes(x=log(mass), y=log(dist), color=meth))+
  ggtitle("Interactive plot of planet catalogue") +
  geom_point_interactive(aes(data_id=id, tooltip=id,onclick=onclick)) +
  theme_bw()
# To create an interactive grahics in the web browser
ggiraph(code = print(p), width = 0.65)
```



Question 5

```
# 5. Rename the radius into jupiter and create new column called earth_radius
# which is 11.2 times the jupiter radius

# Renaming the radius column to jupiter radius
exodata <- exodata %>% rename(jupiter_radius = radius)
# glimpse(exodata)

# Updating new column earth radius using the jupiter radius
exodata <- exodata %>% mutate(earth_radius = jupiter_radius / 11.2)
# glimpse(exodata)
```

Question 6

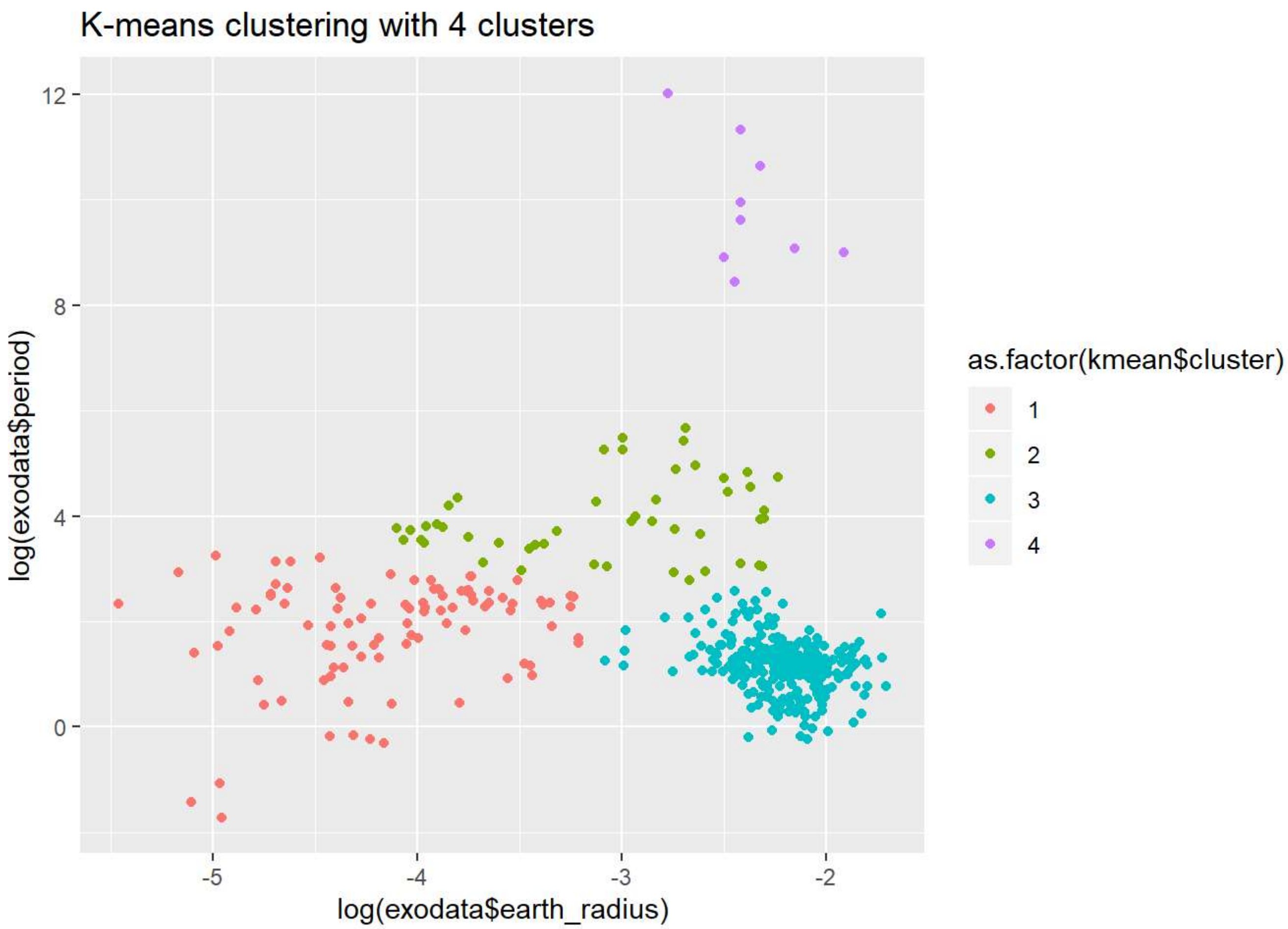
```
# 6) Focus only on the rows where log-radius and log-period have no missing values, and perform
 kmeans with four clusters on these two columns.

# Dropping the NA's from earth radius and period column
exodata <- exodata %>% drop_na(earth_radius)
exodata <- exodata %>% drop_na(period)

# Constructing a matrix using earth radius and period column to perform
# k-means clustering
x = cbind(log(exodata$earth_radius), log(exodata$period))
# Performing k-means with 4 clusters
kmean <- kmeans(x, 4)

# plot(x, col=kmean$cluster, main="kmeans clustering", xlab="Earth radius", ylab="Period (Day
s)")

# K-means clustering plot differentiating the 4 clusters.
ggplot(data=exodata, aes(x=log(exodata$earth_radius), y=log(exodata$period), color=as.factor(kme
an$cluster))) +
  ggtitle("K-means clustering with 4 clusters") +
  geom_point()
```



Question 7

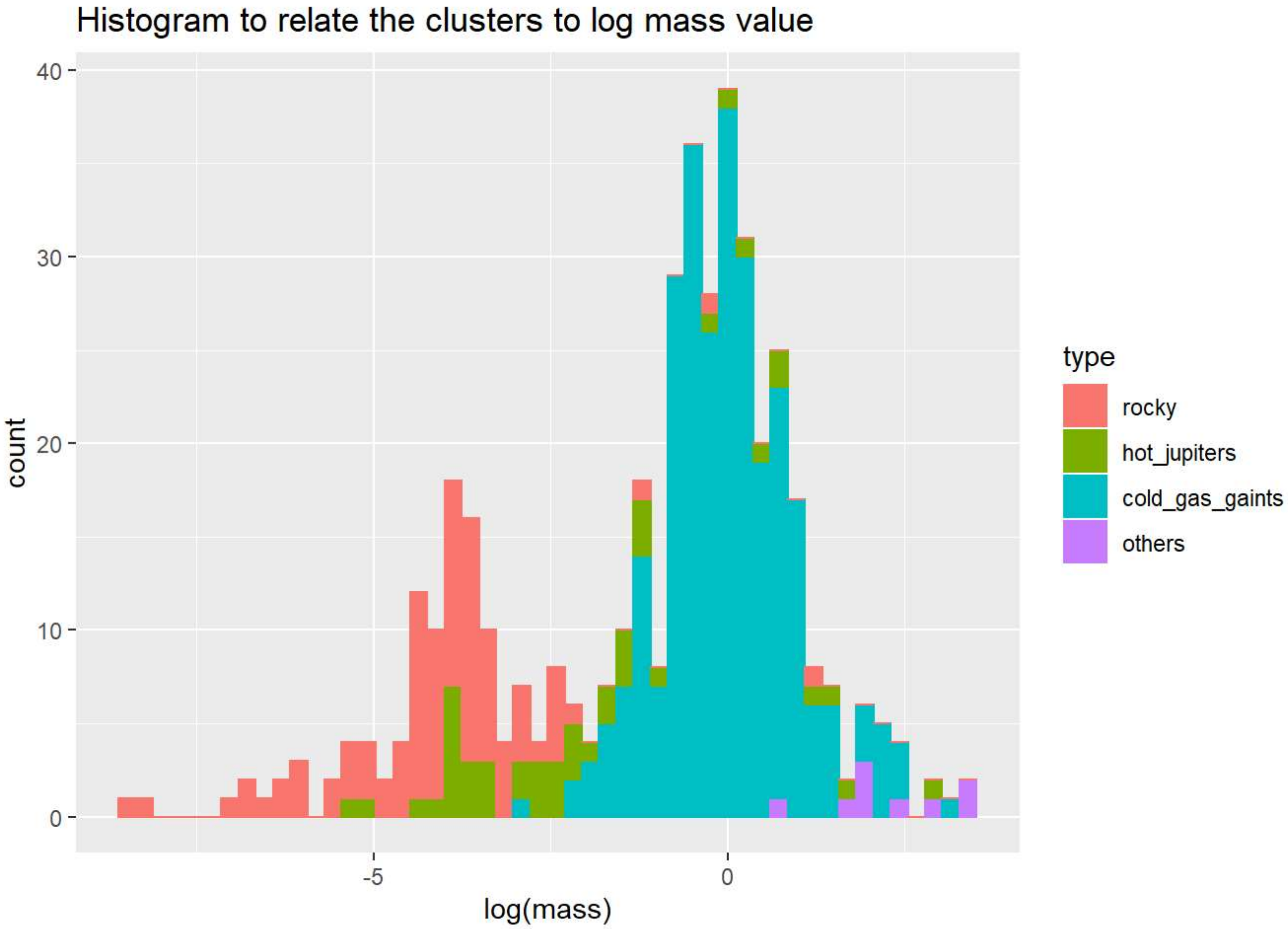
```
# Updating a new column type to exoplanet data with the k-means cluster values
exodata <- exodata %>% mutate(type = kmean$cluster)
# glimpse(exodata)

# Modifying the cluster labels with the following characters
exodata$type <- factor(exodata$type, labels=c("rocky", "hot_jupiters", "cold_gas_gaints", "other
s"))
head(exodata$type)
```

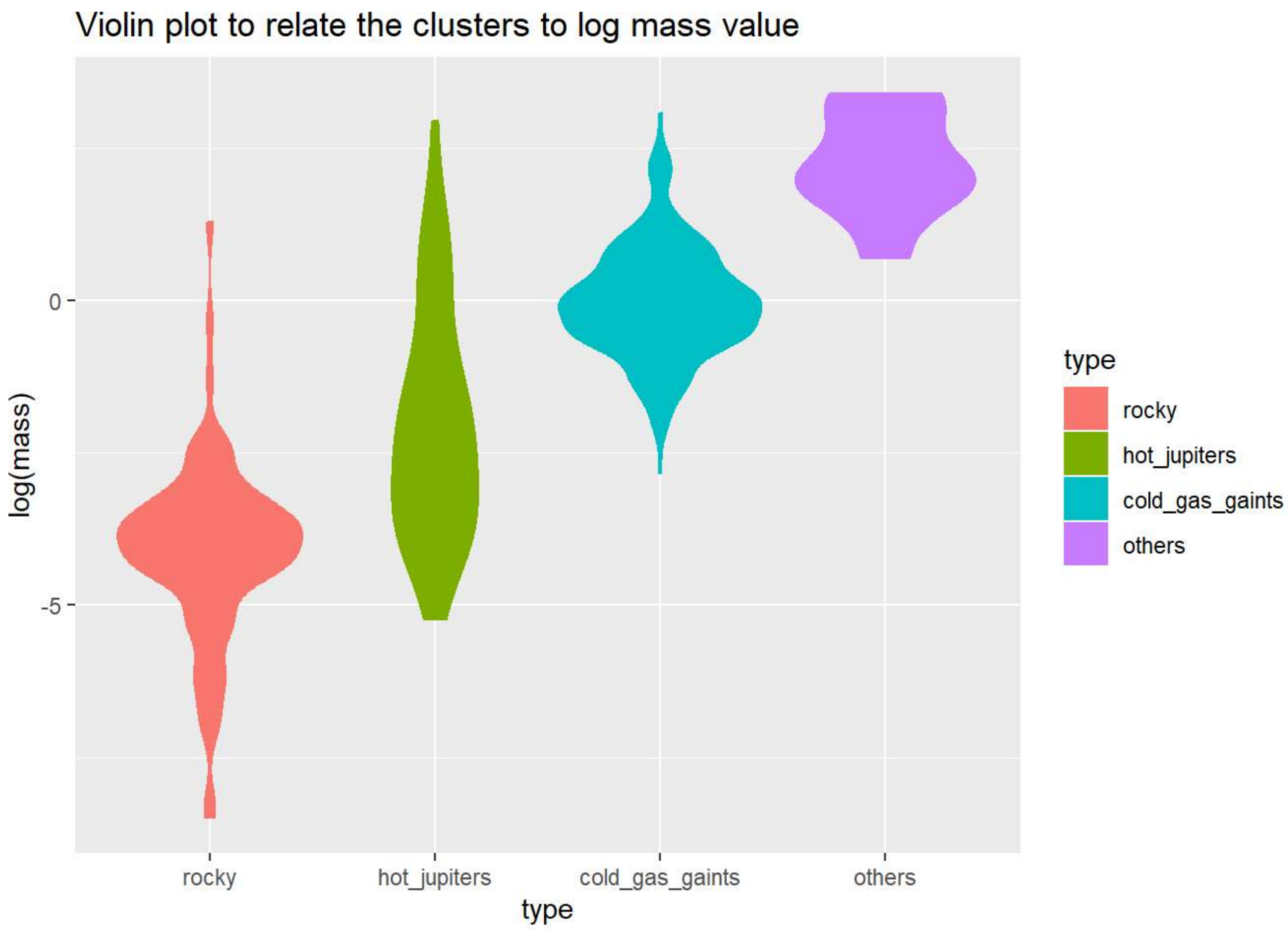
```
## [1] hot_jupiters    hot_jupiters    rocky           cold_gas_gaints
## [5] cold_gas_gaints rocky
## Levels: rocky hot_jupiters cold_gas_gaints others
```

Question 8

```
# Histogram plot to highlight the relationship of k-means cluster to the log mass value.
ggplot(exodata, aes(x= log(mass), color= type, fill=type)) +
  ggtitle("Histogram to relate the clusters to log mass value") +
  geom_histogram(stat = "bin", bins = 50)
```




```
# Highlighting the relationship of k-means cluster to the log mass value using violin plot
ggplot(exodata, aes(x=type, y=log(mass), fill=type, color=type)) +
  ggtitle("Violin plot to relate the clusters to log mass value") +
  geom_violin()
```



Question 9

```
# Transforming the characters representation of time into seconds using hms package
exodata$r_asc <- period_to_seconds(hms(exodata$r_asc))
exodata$decl <- period_to_seconds(hms(exodata$decl))

# Dropping the NA's from earth radius and period column
# exodata <- exodata %>% drop_na(exodata$r_asc)
# exodata <- exodata %>% drop_na(exodata$decl)

# Plot pending
# ggplot(data=exodata, mapping= aes(x=r_asc, y=decl)) +
 # geom_point()
```

Question 10

```
exodata <- exodata %>% drop_na(exodata$year)
exodata <- exodata %>% drop_na(exodata$meth)

counts <- exodata %>% count(year, meth)

# Time series plot
p <- ggplot(data= counts, mapping = aes(x= year, y=meth, color=meth)) +
         geom_line() +
  geom_segment(aes(xend = 31, yend = meth), linetype = 2, colour = 'grey') +
  geom_point(size = 2) +
  transition_reveal(year) +
  coord_cartesian(clip = 'off') +
  labs(x = "year", y="discovered") +
   theme_minimal() +
  theme(plot.margin = margin(5.5, 40, 5.5, 5.5))
```

Question 11

```
# Define UI for application that draws a Scatterplot
ui <- fluidPage(

  # Application title
  titlePanel("Scatterplot of exoplanet"),

  # Sidebar with a slider input for Discovery years and exoplanet type
  sidebarLayout(
    sidebarPanel(
      sliderInput("year",
                  "Discovery years",
                  min = 2009,
                  max = 2017,
                  value = 2015),
      selectInput(inputId = "type",
                  label= "Exoplanet Type",
                  choices = c("rocky", "hot_jupiters", "cold_gas_gaints", "others", "all"),
                  selected = "all")
    ),

    # Show a plot of the generated distribution
    mainPanel(
      plotOutput("ScatterPlot")
    )
  )
)

# Define server logic required to draw a Scatter plot
server <- function(input, output) {

  output$ScatterPlot <- renderPlot({
    # generate scatterplot based on input$year and input$type from ui.R
    if(input$type == "all"){
      ggplot(exodata %>% filter(year == input$year), aes(x=log(mass), y= log(dist))) +
        geom_point(aes(color=meth))
    }

    else{
      ggplot(exodata %>% filter(year == input$year & type == input$type), aes(x=log(mass), y= lo
g(dist), col=meth)) +
        ggtitle("Interactive scatter plot with sliding widget") +
        geom_point()
    }
    # Plotting the scatterplot based on the input from ui.R
    ggplot(exodata, aes(x=log(mass), y=log(dist), color=meth)) +ggtitle("Histogram to relate the
clusters to log mass value") +
      geom_point()
  })
}

# Run the application
shinyApp(ui = ui, server = server)
```

Scatterplot of exoplanet

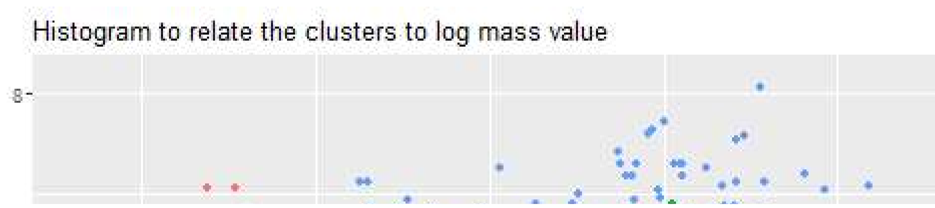
Discovery years

2,009 2,015 2,017

2,009 2,010 2,011 2,012 2,013 2,014 2,015 2,017

Exoplanet Type

all



Question 12

```
rstan_options(auto_write= TRUE)
options(mc.cores = parallel::detectCores())

# Dropping the na's before preparing the stan list
exodata <- exodata %>% drop_na(period)
exodata <- exodata %>% drop_na(host_mass)
exodata <- exodata %>% drop_na(host_temp)
exodata <- exodata %>% drop_na(axis)

# Extracting only period, host mass, host temp and axis columns
exodata_rstan <- exodata[,c(5,6,20,23)]

# Constructing the list as input to the stan
x1 = scale(log(exodata_rstan$host_mass))[,1]
x2 = scale(log(exodata_rstan$host_temp))[,1]
x3 = scale(log(exodata_rstan$axis))[,1]
exo_data_lr = list(N = nrow(exodata_rstan),
                   x = cbind(x1,x2,x3),
                   y = scale(log(exodata_rstan$period))[,1])

# Maximum likelehood estimation using optimizing function
exo_data_lr$K = 3
exo_data_lr$x = matrix(exo_data_lr$x, ncol = 3)
# Calculating the maximum likelehood from the regression model stran file
stan_model_lr = stan_model("regression_model.stan")
stan_run_lr_ml = optimizing(stan_model_lr, data=exo_data_lr)
print(stan_run_lr_ml)
```

```
## $par
##         alpha       beta[1]       beta[2]       beta[3]         sigma
## -9.290533e-07 -1.443336e-01 -4.293221e-03  9.932219e-01  3.833848e-02
##
## $value
## [1] 1046.537
##
## $return_code
## [1] 0
##
## $theta_tilde
##              alpha    beta[1]      beta[2]   beta[3]      sigma
## [1,] -9.290533e-07 -0.1443336 -0.004293221 0.9932219 0.03833848
```

```
summary(stan_run_lr_ml)
```

```
##             Length Class  Mode
## par         5      -none- numeric
## value       1      -none- numeric
## return_code 1      -none- numeric
## theta_tilde 5      -none- numeric
```

Question 13

```
# Calculating the posterier values from the regression model stran file
stan_model_lr_post = stan_model("posterier_model.stan")
# stan_run_lr_ml = optimizing(stan_model_lr, data=exo_data_lr)
stan_run_lr_post = sampling(stan_model_lr_post, data=exo_data_lr)
print(stan_run_lr_post)
```

```
## Inference for Stan model: posterier_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean   sd    2.5%     25%     50%     75%   97.5% n_eff
## alpha       0.00    0.00 0.00    0.00    0.00    0.00    0.00    0.00  5249
## beta[1]    -0.14    0.00 0.00   -0.15   -0.15   -0.14   -0.14   -0.14  3042
## beta[2]     0.00    0.00 0.00   -0.01   -0.01    0.00    0.00    0.00  3216
## beta[3]     0.99    0.00 0.00    0.99    0.99    0.99    0.99    1.00  5300
## sigma       0.04    0.00 0.00    0.04    0.04    0.04    0.04    0.04  2133
## lp__     1040.23    0.04 1.55 1036.49 1039.40 1040.52 1041.37 1042.32  1663
##          Rhat
## alpha       1
## beta[1]     1
## beta[2]     1
## beta[3]     1
## sigma       1
## lp__        1
##
## Samples were drawn using NUTS(diag_e) at Sun Sep 01 23:06:56 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

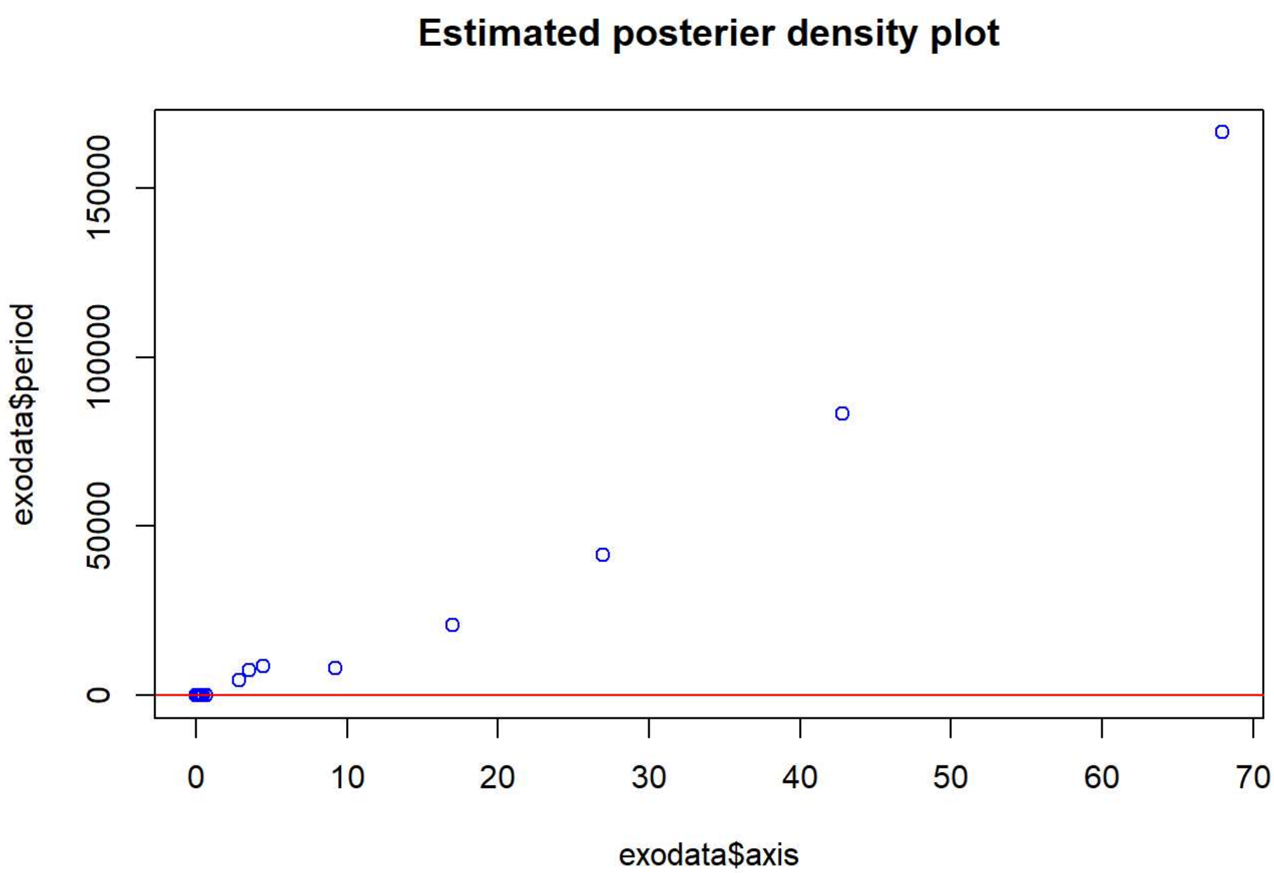
```
# summary(stan_run_lr_post)

# Plotting the intercept against the slope
# plot(exodata$period, exodata$mass, col=3)
# abline(a = stan_run_lr_ml$par[['alpha']], b= stan_run_lr_ml$par[['beta[1]']])

# plot(exodata$period, exodata$temp, col=2)
# abline(a=stan_run_lr_ml$par[['alpha']], b=stan_run_lr_ml$par[['beta[2]']], col=3)
```

Question 14

```
# Estimated Posterier density plot
plot(exodata$axis, exodata$period, col=4, main="Estimated posterier density plot")
abline(a=stan_run_lr_ml$par[['alpha']], b=stan_run_lr_ml$par[['beta[3]']], col=2)
```



Question 15 Embedded the Shiny app to R Markdown document using Shiny document option.