20BDS0146

VENNELA G

DATA VISUALIZATION AND PRESENTATION

LAB SLOT: L31+32

LAB ASSIGNMENT 4

1. Create a Text Analytics with Word Cloud using Shakespeare dataset.

R CODE:

```
library(tidytext)

library(dplyr)

install.packages("wordcloud")

library(wordcloud)

shakespeare %>% count(title, type)

library(tidytext)

tidy_shakespeare <- shakespeare %>%group_by(title,type)
%>%mutate(linenumber = row_number()) %>%unnest_tokens(word, text)
%>%ungroup()

tidy_shakespeare

words <- shakespeare %>%select(text) %>%unnest_tokens(word, text)


word_freq <- words %>%count(word, sort = TRUE)

set.seed(123)

wordcloud(words = word_freq$word, freq = word_freq$n, min.freq =
10,max.words = 200, random.order = FALSE, rot.per = 0.35,colors = brewer.pal(8,
"Dark2"))

install.packages("textdata")

library(textdata)
```

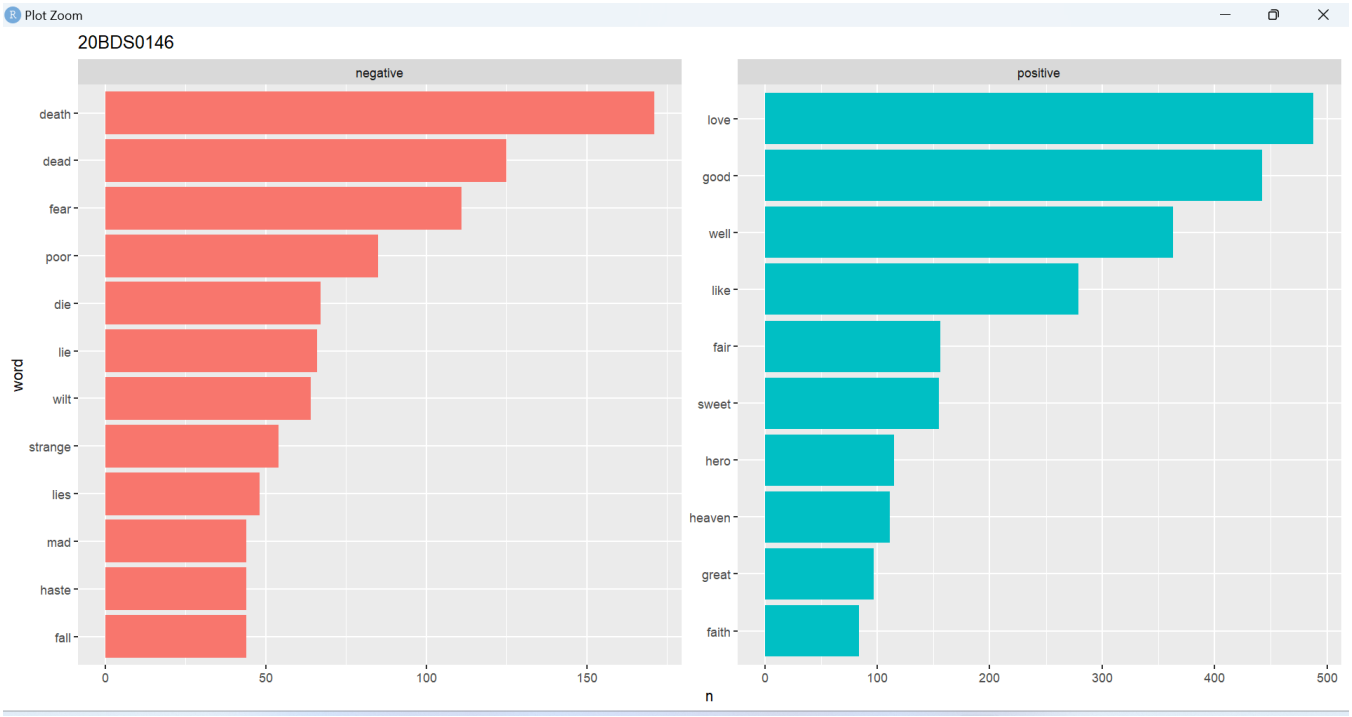
```
shakespeare_sentiment <- tidy_shakespeare
%>%inner_join(get_sentiments("bing"))

shakespeare_sentiment %>%count(title, sentiment)

sentiment_counts <- tidy_shakespeare %>%inner_join(get_sentiments("bing"))
%>%count(title, type, sentiment)

sentiment_counts %>%group_by(title) %>% mutate(total = sum(n), percent = n /
total) %>% filter(sentiment == "negative") %>% arrange(percent)

word_counts <- tidy_shakespeare %>%inner_join(get_sentiments("bing"))
%>%count(word, sentiment)

top_words <- word_counts %>%group_by(sentiment) %>% top_n(10) %>%
ungroup() %>% mutate(word = reorder(word, n))

library(ggplot2)

ggplot(top_words, aes(x = word, y = n, fill = sentiment)) +geom_col(show.legend =
FALSE) +facet_wrap(~sentiment, scales = "free") +
coord_flip()+ggtitle('20BDS0146')

tidy_shakespeare %>%count(title, word, sort = TRUE)
%>%inner_join(get_sentiments("afinn")) %>%filter(title == "The Tragedy of
Macbeth", value < 0)

sentiment_contributions <- tidy_shakespeare %>%count(title, word, sort = TRUE)
%>%  inner_join(get_sentiments("afinn")) %>% group_by(title) %>%
mutate(contribution = (n * value) / sum(n)) %>% ungroup()

sentiment_contributions
```

OUTPUT:

```
> tidy_shakespeare
# A tibble: 141,067 × 3
   title                               type     word
   <chr>                               <chr>    <chr>
 1 The Tragedy of Romeo and Juliet     Tragedy  the
 2 The Tragedy of Romeo and Juliet     Tragedy  complete
 3 The Tragedy of Romeo and Juliet     Tragedy  works
 4 The Tragedy of Romeo and Juliet     Tragedy  of
 5 The Tragedy of Romeo and Juliet     Tragedy  william
 6 The Tragedy of Romeo and Juliet     Tragedy  shakespe
 7 The Tragedy of Romeo and Juliet     Tragedy  the
 8 The Tragedy of Romeo and Juliet     Tragedy  tragedy
 9 The Tragedy of Romeo and Juliet     Tragedy  of
10 The Tragedy of Romeo and Juliet     Tragedy  romeo
# … with 141,057 more rows
# ℹ Use `print(n = ...)` to see more rows
> 20BDS0146
```

```
> shakespeare_sentiment %>%count(title, sentiment)
# A tibble: 12 × 3
   title                               sentiment     n
   <chr>                               <chr>     <int>
 1 A Midsummer Night's Dream           negative    681
 2 A Midsummer Night's Dream           positive    773
 3 Hamlet, Prince of Denmark           negative   1323
 4 Hamlet, Prince of Denmark           positive   1223
 5 Much Ado about Nothing              negative    767
 6 Much Ado about Nothing              positive   1127
 7 The Merchant of Venice              negative    740
 8 The Merchant of Venice              positive    962
 9 The Tragedy of Macbeth              negative    914
10 The Tragedy of Macbeth              positive    749
11 The Tragedy of Romeo and Juliet     negative   1235
12 The Tragedy of Romeo and Juliet     positive   1090
> 20BDS0146
```

```
> sentiment_counts %>%group_by(title) %>% mutate(total = sum(n), percent = n
tal) %>% filter(sentiment == "negative") %>% arrange(percent)
# A tibble: 6 x 6
# Groups:   title [6]
  title                          type    sentiment     n total percent
  <chr>                          <chr>   <chr>     <int> <int>   <dbl>
1 Much Ado about Nothing         Comedy  negative    767  1894   0.405
2 The Merchant of Venice         Comedy  negative    740  1702   0.435
3 A Midsummer Night's Dream      Comedy  negative    681  1454   0.468
4 Hamlet, Prince of Denmark      Tragedy negative   1323  2546   0.520
5 The Tragedy of Romeo and Juliet Tragedy negative  1235  2325   0.531
6 The Tragedy of Macbeth         Tragedy negative    914  1663   0.550
> 20BDS0146
```



```
> tidy_shakespeare %>%count(title, word, sort = TRUE) %>%inner_join(get_sentiment
s("afinn")) %>%filter(title == "The Tragedy of Macbeth", value < 0)
Joining, by = "word"
# A tibble: 237 × 4
   title                  word        n value
   <chr>                  <chr>   <int> <dbl>
 1 The Tragedy of Macbeth no         73    -1
 2 The Tragedy of Macbeth fear       35    -2
 3 The Tragedy of Macbeth death      20    -2
 4 The Tragedy of Macbeth bloody     16    -3
 5 The Tragedy of Macbeth poor       16    -2
 6 The Tragedy of Macbeth strange    16    -1
 7 The Tragedy of Macbeth dead       14    -3
 8 The Tragedy of Macbeth leave      14    -1
 9 The Tragedy of Macbeth fight      13    -1
10 The Tragedy of Macbeth charges    11    -2
# … with 227 more rows
# ℹ Use `print(n = ...)` to see more rows
> 20BDS0146
```

```
> sentiment_contributions
# A tibble: 2,366 × 5
   title                           word      n value contribution
   <chr>                           <chr> <int> <dbl>        <dbl>
 1 Hamlet, Prince of Denmark       no      143    -1      -0.0652
 2 The Tragedy of Romeo and Juliet love    140     3       0.213
 3 Much Ado about Nothing          no      132    -1      -0.0768
 4 Much Ado about Nothing          hero    114     2       0.133
 5 A Midsummer Night's Dream       love    110     3       0.270
 6 Hamlet, Prince of Denmark       good    109     3       0.149
 7 The Tragedy of Romeo and Juliet no      102    -1      -0.0518
 8 Much Ado about Nothing          good     93     3       0.162
 9 The Merchant of Venice          no       92    -1      -0.0630
10 Much Ado about Nothing          love     91     3       0.159
# … with 2,356 more rows
# ℹ Use `print(n = ...)` to see more rows
> 20BDS0146
```

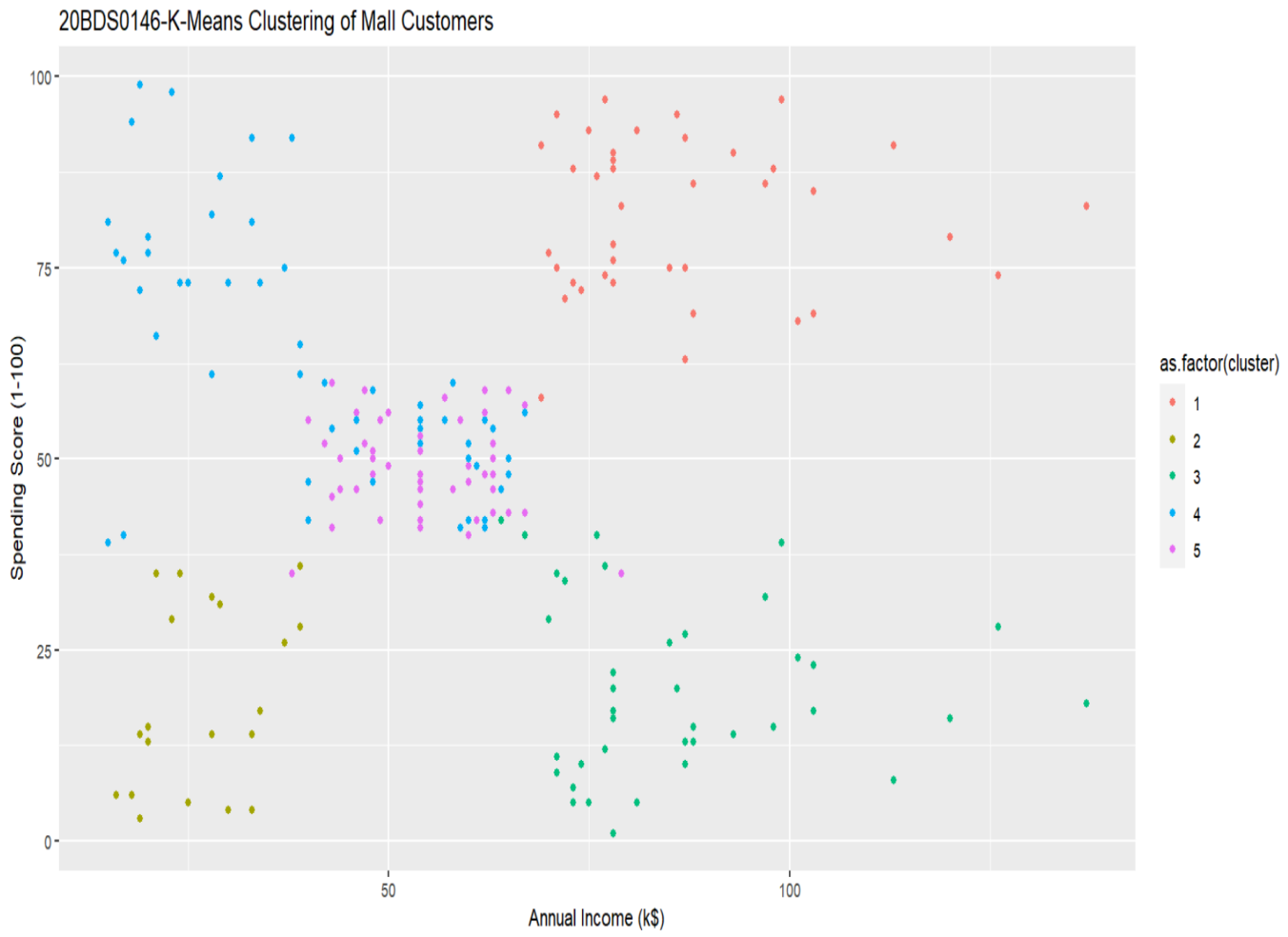

Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
20bds0146
down two father day did think give exeunt god life dead
eyes hear never can may upon had pedro first queen
thus say go lord enter would one too fair
most their man thy are when mine
any tell am him her he do thou here true
know up where as all is in not o which should look friar
lady at your you it now my have they take macbeth
portia shall if be or the of but our
make hath with so i this till there
some more hor what me a to for let must
ham scene will she
out an was we way that good time
heart night and art beat
nor then his how such very
romeo were why on tis love like much pray hero
sweet than thee no by us well exit nurse
juli ill come from them sir these doth hand
death speak king yet old see wife bene leon
away who rom bassanio

2. Perform K-means Cluster using Mall Customer dataset

R CODE:

```
library(dplyr)

library(ggplot2)

library(cluster)

head(Mall_Customers)

features <- select(Mall_Customers, -c(CustomerID, Genre))

scaled_features <- scale(features)

set.seed(123)

wss <- numeric(10)

for (i in 1:10) {

  kmeans_model <- kmeans(scaled_features, centers = i, nstart = 25)

  wss[i] <- kmeans_model$tot.withinss

}

plot(1:10, wss, type = "b", xlab = "Number of Clusters", ylab = "Within-Cluster Sum of Squares")

abline(v = 5, lty = 2)

kmeans_model <- kmeans(scaled_features, centers = 5, nstart = 25)

Mall_Customers$cluster <- kmeans_model$cluster

ggplot(Mall_Customers, aes(x = Annual.Income..k.., y = Spending.Score..1.100., color = as.factor(cluster))) +geom_point() +xlab("Annual Income (k$)") +ylab("Spending Score (1-100)") +ggtitle("20BDS0146-K-Means Clustering of Mall Customers")
```

OUTPUT:


20BDS0146-K-Means Clustering of Mall Customers
Spending Score (1-100)
Annual Income (k$)
100
75
50
25
0
50
100
as.factor(cluster)
1
2
3
4
5

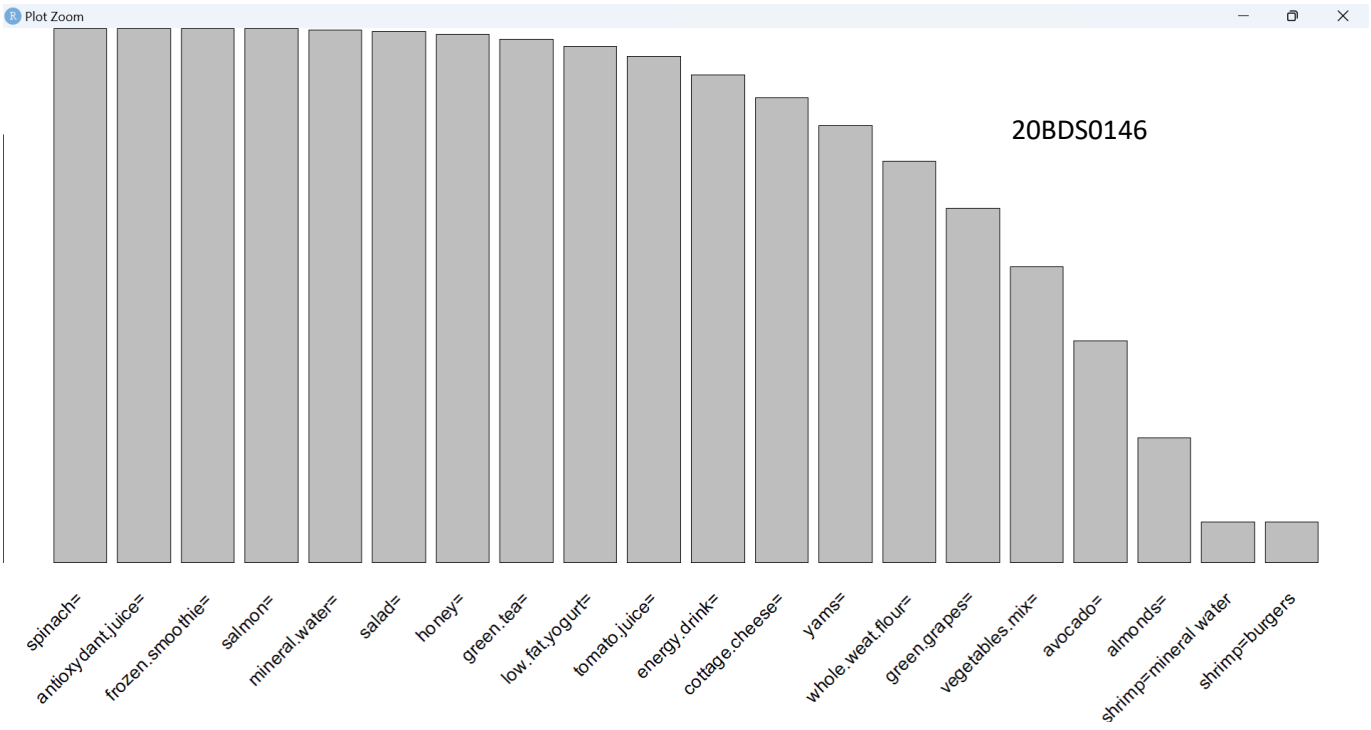
3. Perform Market basket analysis

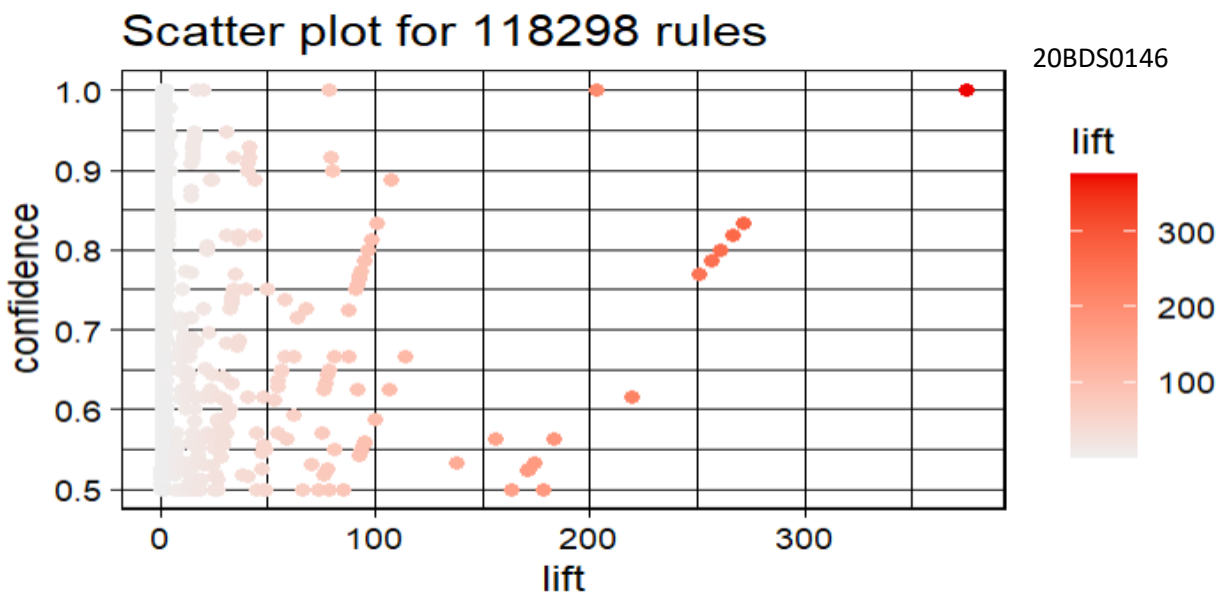
RCODE:

```
data <- read.csv("C:/Users/HP/Downloads/Market_Basket_Optimisation.csv")

transactions <- as(data, "transactions")

summary(transactions)

title(".")

title("VENNELA G-20BDS0146")

itemFrequencyPlot(transactions, topN = 20)

rules <- apriori(transactions, parameter = list(support = 0.001, confidence = 0.5,minlen = 2, maxlen = 3))

top_rules <- head(sort(rules, by = "lift"), 10)

inspect(top_rules)
```

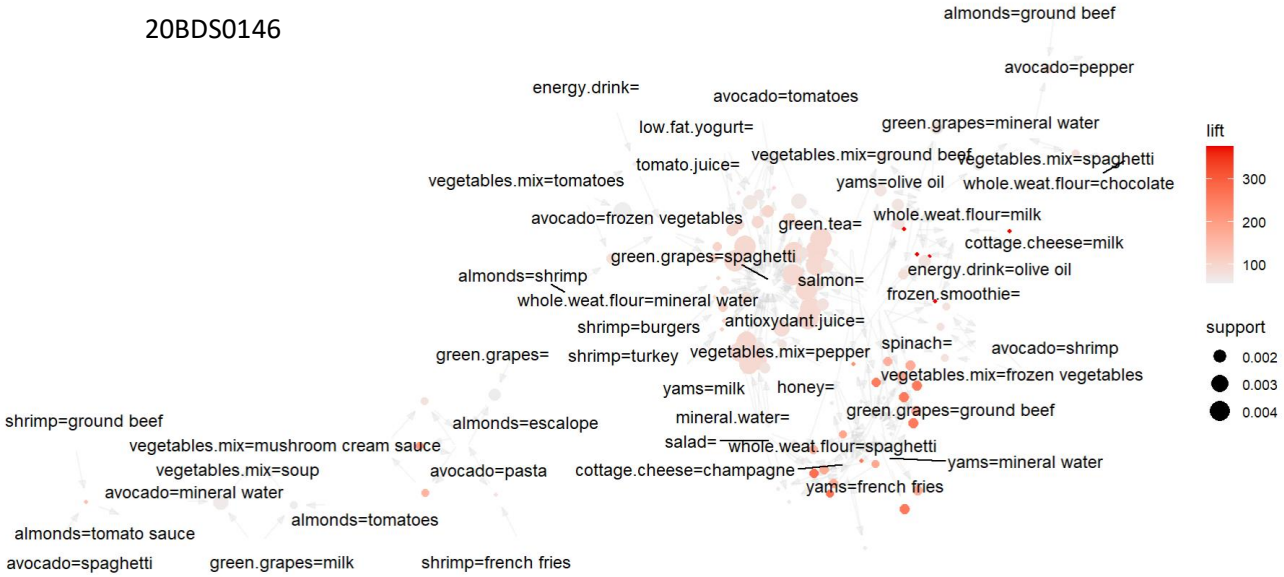
OUTPUT:

```
> inspect(top_rules)
     lhs                                rhs                          support confidence coverage lift count
[1]  {energy.drink=olive oil}        => {cottage.cheese=milk}         0.0011       1.00   0.0011  375     8
[2]  {energy.drink=olive oil,
      salmon=}                       => {cottage.cheese=milk}         0.0011       1.00   0.0011  375     8
[3]  {energy.drink=olive oil,
      antioxydant.juice=}            => {cottage.cheese=milk}         0.0011       1.00   0.0011  375     8
[4]  {energy.drink=olive oil,
      frozen.smoothie=}              => {cottage.cheese=milk}         0.0011       1.00   0.0011  375     8
[5]  {energy.drink=olive oil,
      spinach=}                      => {cottage.cheese=milk}         0.0011       1.00   0.0011  375     8
[6]  {yams=mineral water,
      mineral.water=}                => {whole.weat.flour=spaghetti}  0.0013       0.83   0.0016  272    10
[7]  {yams=mineral water,
      salad=}                        => {whole.weat.flour=spaghetti}  0.0012       0.82   0.0015  267     9
[8]  {yams=mineral water,
      honey=}                        => {whole.weat.flour=spaghetti}  0.0011       0.80   0.0013  261     8
[9]  {yams=mineral water}            => {whole.weat.flour=spaghetti}  0.0015       0.79   0.0019  256    11
[10] {yams=mineral water,
      antioxydant.juice=}            => {whole.weat.flour=spaghetti}  0.0015       0.79   0.0019  256    11
> 20BDS0146
```


Plot Zoom
20BDS0146
spinach=
antioxydant.juice=
frozen.smoothie=
salmon=
mineral.water=
salad=
honey=
green.tea=
low.fat.yogurt=
tomato.juice=
energy.drink=
cottage.cheese=
yams=
whole.weat.flour=
green.grapes=
vegetables.mix=
avocado=
almonds=
shrimp=mineral water
shrimp=burgers


Scatter plot for 118298 rules
20BDS0146
confidence
1.0
0.9
0.8
0.7
0.6
0.5
0
100
200
300
lift
lift
300
200
100

20BDS0146


Plot Zoom
almonds=ground beef
avocado=pepper
energy.drink=
avocado=tomatoes
low.fat.yogurt=
green.grapes=mineral water
tomato.juice=
vegetables.mix=ground beef
vegetables.mix=spaghetti
vegetables.mix=tomatoes
yams=olive oil
whole.weat.flour=chocolate
avocado=frozen vegetables
green.tea=
whole.weat.flour=milk
cottage.cheese=milk
green.grapes=spaghetti
energy.drink=olive oil
almonds=shrimp
salmon=
frozen.smoothie=
whole.weat.flour=mineral water
shrimp=burgers
antioxydant.juice=
green.grapes=
shrimp=turkey
vegetables.mix=pepper
spinach=
avocado=shrimp
vegetables.mix=frozen vegetables
yams=milk
honey=
green.grapes=ground beef
shrimp=ground beef
almonds=escalope
mineral.water=
vegetables.mix=mushroom cream sauce
salad=
whole.weat.flour=spaghetti
vegetables.mix=soup
avocado=pasta
cottage.cheese=champagne
yams=mineral water
avocado=mineral water
yams=french fries
almonds=tomatoes
almonds=tomato sauce
avocado=spaghetti
green.grapes=milk
shrimp=french fries
lift
300
200
100
support
0.002
0.003
0.004

4. Create a Shiny Application using gapminder dataset

R CODE

ui.R

```
library(shiny)
library(gapminder)
library(ggplot2)
ui <- fluidPage(
  titlePanel("Gapminder Data Exploration-20BDS0146-VENNELA G"),
  sidebarLayout(
    sidebarPanel(
      selectInput("x", "Select X-axis variable:", choices = c("lifeExp", "gdpPercap", "pop")),
      selectInput("y", "Select Y-axis variable:", choices = c("lifeExp", "gdpPercap", "pop")),
      sliderInput("year", "Select year:", min = 1952, max = 2007, step = 5, value = 1952)
    ),
    mainPanel(
      plotOutput("scatterplot")
    )
  )
)
```

server.R

```
server <- function(input, output) {
  output$scatterplot <- renderPlot({
    ggplot(filter(gapminder, year == input$year), aes_string(x = input$x, y = input$y)) +
      geom_point() +
      scale_x_log10() +
      scale_y_log10() +
      labs(x = input$x, y = input$y)
  })
}

shinyApp(ui, server)
```

OUTPUT

~ - Shiny

http://127.0.0.1:5081 | Open in Browser | Publish

Gapminder Data Exploration-20BDS0146-VENNELA G

Select X-axis variable:

gdpPercap

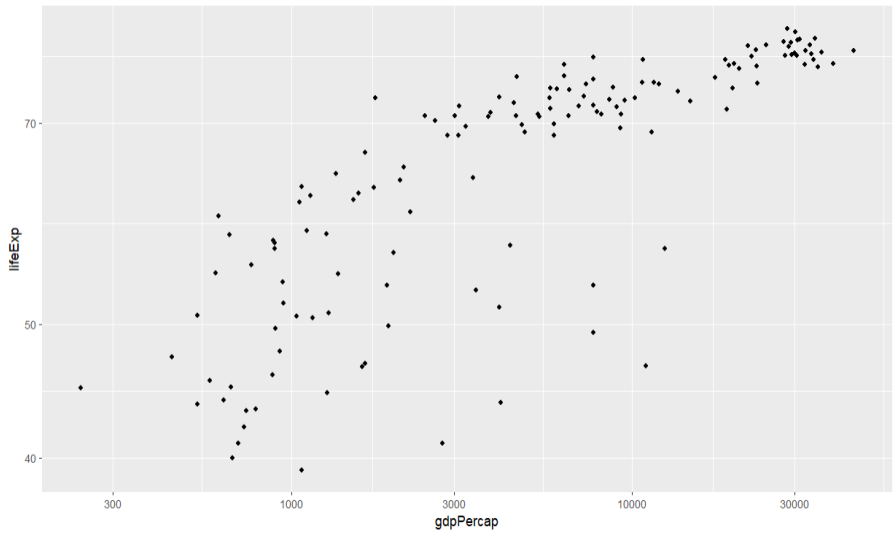
Select Y-axis variable:

lifeExp

Select year:

1,952 — 2,002

1,952 1,962 1,972 1,982 1,992 2,002 2,007



Gapminder Data Exploration-20BDS0146-VENNELA G

Select X-axis variable:

pop

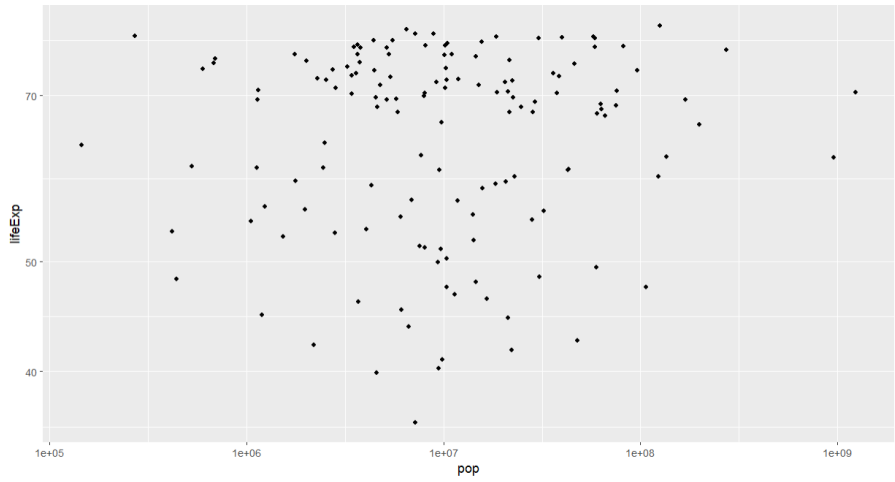
Select Y-axis variable:

lifeExp

Select year:

1,952 — 1,997 — 2,007

1,952 1,962 1,972 1,982 1,992 2,002 2,007



Gapminder Data Exploration-20BDS0146-VENNELA G

Select X-axis variable:

gdpPercap

Select Y-axis variable:

pop

Select year:

1,952 1,977 2,007

1,952 1,962 1,972 1,982 1,992 2,002 2,007