predict the optimum number of clusters

surabhi

06/02/2021

```
#to import the data set
df=iris
#to check any outliers and missing values are present
summary(df)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
## 
```

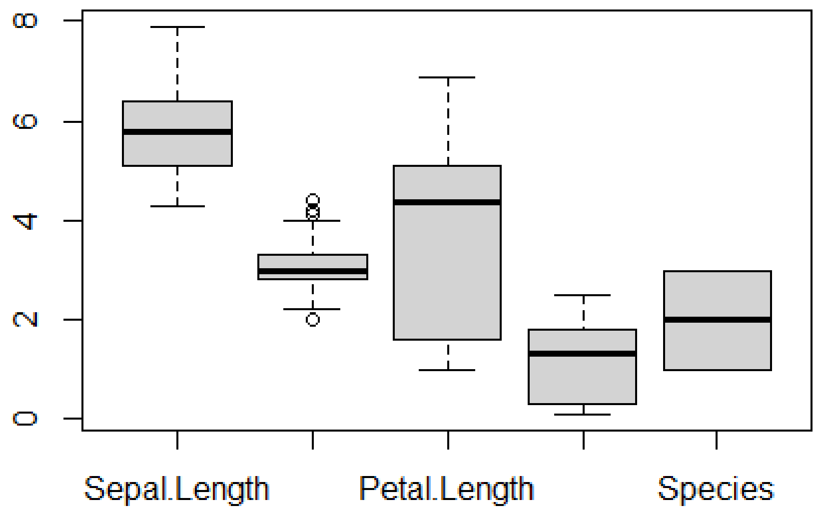
```
boxplot(df)
#to remove the non numeric feature
df <- subset (df, select = -Species)
df
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1            5.1         3.5          1.4         0.2
## 2            4.9         3.0          1.4         0.2
## 3            4.7         3.2          1.3         0.2
## 4            4.6         3.1          1.5         0.2
## 5            5.0         3.6          1.4         0.2
## 6            5.4         3.9          1.7         0.4
## 7            4.6         3.4          1.4         0.3
## 8            5.0         3.4          1.5         0.2
## 9            4.4         2.9          1.4         0.2
## 10           4.9         3.1          1.5         0.1
## 11           5.4         3.7          1.5         0.2
## 12           4.8         3.4          1.6         0.2
## 13           4.8         3.0          1.4         0.1
## 14           4.3         3.0          1.1         0.1
## 15           5.8         4.0          1.2         0.2
## 16           5.7         4.4          1.5         0.4
```

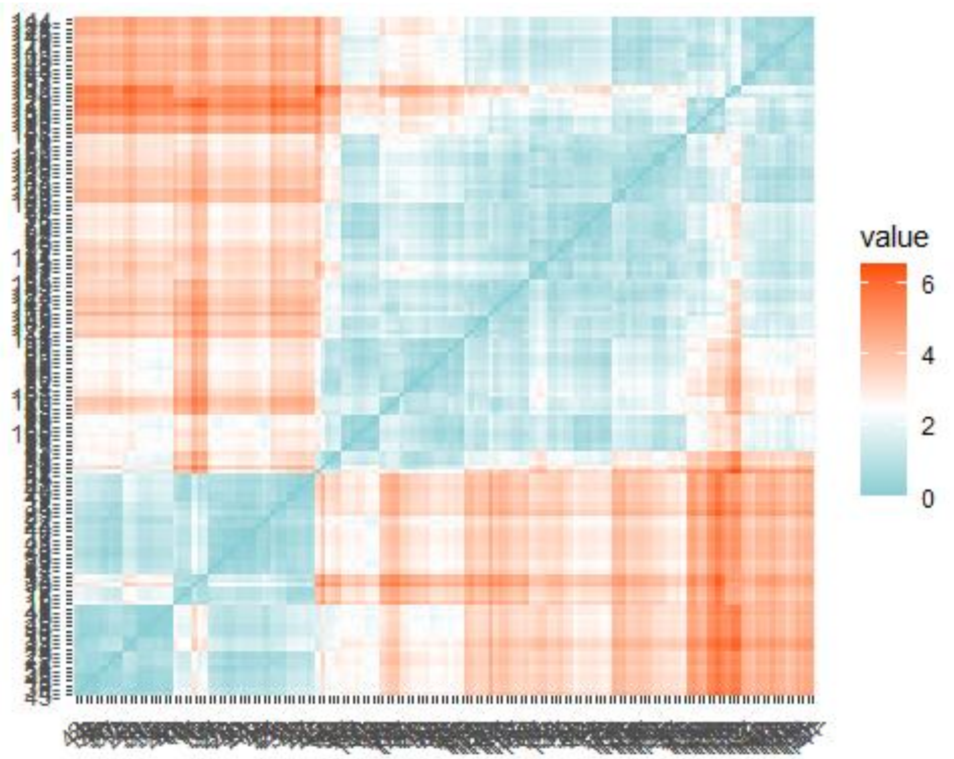
```
## 17          5.4         3.9          1.3         0.4
## 18          5.1         3.5          1.4         0.3
## 19          5.7         3.8          1.7         0.3
## 20          5.1         3.8          1.5         0.3
## 21          5.4         3.4          1.7         0.2
## 22          5.1         3.7          1.5         0.4
## 23          4.6         3.6          1.0         0.2
## 24          5.1         3.3          1.7         0.5
## 25          4.8         3.4          1.9         0.2
## 26          5.0         3.0          1.6         0.2
## 27          5.0         3.4          1.6         0.4
## 28          5.2         3.5          1.5         0.2
## 29          5.2         3.4          1.4         0.2
## 30          4.7         3.2          1.6         0.2
## 31          4.8         3.1          1.6         0.2
## 32          5.4         3.4          1.5         0.4
## 33          5.2         4.1          1.5         0.1
## 34          5.5         4.2          1.4         0.2
## 35          4.9         3.1          1.5         0.2
## 36          5.0         3.2          1.2         0.2
## 37          5.5         3.5          1.3         0.2
## 38          4.9         3.6          1.4         0.1
## 39          4.4         3.0          1.3         0.2
## 40          5.1         3.4          1.5         0.2
## 41          5.0         3.5          1.3         0.3
## 42          4.5         2.3          1.3         0.3
## 43          4.4         3.2          1.3         0.2
## 44          5.0         3.5          1.6         0.6
## 45          5.1         3.8          1.9         0.4
## 46          4.8         3.0          1.4         0.3
## 47          5.1         3.8          1.6         0.2
## 48          4.6         3.2          1.4         0.2
## 49          5.3         3.7          1.5         0.2
## 50          5.0         3.3          1.4         0.2
## 51          7.0         3.2          4.7         1.4
## 52          6.4         3.2          4.5         1.5
## 53          6.9         3.1          4.9         1.5
## 54          5.5         2.3          4.0         1.3
## 55          6.5         2.8          4.6         1.5
## 56          5.7         2.8          4.5         1.3
## 57          6.3         3.3          4.7         1.6
## 58          4.9         2.4          3.3         1.0
## 59          6.6         2.9          4.6         1.3
## 60          5.2         2.7          3.9         1.4
## 61          5.0         2.0          3.5         1.0
## 62          5.9         3.0          4.2         1.5
## 63          6.0         2.2          4.0         1.0
## 64          6.1         2.9          4.7         1.4
## 65          5.6         2.9          3.6         1.3
## 66          6.7         3.1          4.4         1.4
```

```
## 67          5.6         3.0          4.5         1.5
## 68          5.8         2.7          4.1         1.0
## 69          6.2         2.2          4.5         1.5
## 70          5.6         2.5          3.9         1.1
## 71          5.9         3.2          4.8         1.8
## 72          6.1         2.8          4.0         1.3
## 73          6.3         2.5          4.9         1.5
## 74          6.1         2.8          4.7         1.2
## 75          6.4         2.9          4.3         1.3
## 76          6.6         3.0          4.4         1.4
## 77          6.8         2.8          4.8         1.4
## 78          6.7         3.0          5.0         1.7
## 79          6.0         2.9          4.5         1.5
## 80          5.7         2.6          3.5         1.0
## 81          5.5         2.4          3.8         1.1
## 82          5.5         2.4          3.7         1.0
## 83          5.8         2.7          3.9         1.2
## 84          6.0         2.7          5.1         1.6
## 85          5.4         3.0          4.5         1.5
## 86          6.0         3.4          4.5         1.6
## 87          6.7         3.1          4.7         1.5
## 88          6.3         2.3          4.4         1.3
## 89          5.6         3.0          4.1         1.3
## 90          5.5         2.5          4.0         1.3
## 91          5.5         2.6          4.4         1.2
## 92          6.1         3.0          4.6         1.4
## 93          5.8         2.6          4.0         1.2
## 94          5.0         2.3          3.3         1.0
## 95          5.6         2.7          4.2         1.3
## 96          5.7         3.0          4.2         1.2
## 97          5.7         2.9          4.2         1.3
## 98          6.2         2.9          4.3         1.3
## 99          5.1         2.5          3.0         1.1
## 100         5.7         2.8          4.1         1.3
## 101         6.3         3.3          6.0         2.5
## 102         5.8         2.7          5.1         1.9
## 103         7.1         3.0          5.9         2.1
## 104         6.3         2.9          5.6         1.8
## 105         6.5         3.0          5.8         2.2
## 106         7.6         3.0          6.6         2.1
## 107         4.9         2.5          4.5         1.7
## 108         7.3         2.9          6.3         1.8
## 109         6.7         2.5          5.8         1.8
## 110         7.2         3.6          6.1         2.5
## 111         6.5         3.2          5.1         2.0
## 112         6.4         2.7          5.3         1.9
## 113         6.8         3.0          5.5         2.1
## 114         5.7         2.5          5.0         2.0
## 115         5.8         2.8          5.1         2.4
## 116         6.4         3.2          5.3         2.3
```

```
## 117          6.5         3.0          5.5         1.8
## 118          7.7         3.8          6.7         2.2
## 119          7.7         2.6          6.9         2.3
## 120          6.0         2.2          5.0         1.5
## 121          6.9         3.2          5.7         2.3
## 122          5.6         2.8          4.9         2.0
## 123          7.7         2.8          6.7         2.0
## 124          6.3         2.7          4.9         1.8
## 125          6.7         3.3          5.7         2.1
## 126          7.2         3.2          6.0         1.8
## 127          6.2         2.8          4.8         1.8
## 128          6.1         3.0          4.9         1.8
## 129          6.4         2.8          5.6         2.1
## 130          7.2         3.0          5.8         1.6
## 131          7.4         2.8          6.1         1.9
## 132          7.9         3.8          6.4         2.0
## 133          6.4         2.8          5.6         2.2
## 134          6.3         2.8          5.1         1.5
## 135          6.1         2.6          5.6         1.4
## 136          7.7         3.0          6.1         2.3
## 137          6.3         3.4          5.6         2.4
## 138          6.4         3.1          5.5         1.8
## 139          6.0         3.0          4.8         1.8
## 140          6.9         3.1          5.4         2.1
## 141          6.7         3.1          5.6         2.4
## 142          6.9         3.1          5.1         2.3
## 143          5.8         2.7          5.1         1.9
## 144          6.8         3.2          5.9         2.3
## 145          6.7         3.3          5.7         2.5
## 146          6.7         3.0          5.2         2.3
## 147          6.3         2.5          5.0         1.9
## 148          6.5         3.0          5.2         2.0
## 149          6.2         3.4          5.4         2.3
## 150          5.9         3.0          5.1         1.8

#importing the essential packages
library(tidyverse)  # data manipulation

## Warning: package 'tidyverse' was built under R version 4.0.3

## -- Attaching packages ------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ---------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```



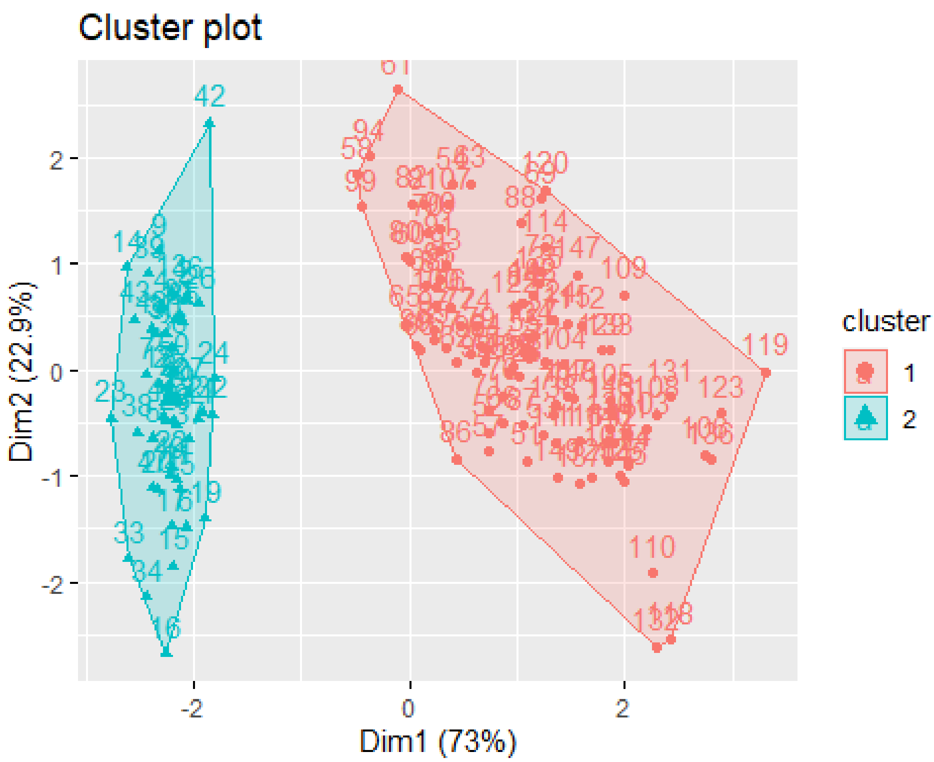

```
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization

## Warning: package 'factoextra' was built under R version 4.0.3

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

#to standardizing the data
df <- scale(df)
head(df)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1   -0.8976739  1.01560199    -1.335752   -1.311052
## 2   -1.1392005 -0.13153881    -1.335752   -1.311052
## 3   -1.3807271  0.32731751    -1.392399   -1.311052
## 4   -1.5014904  0.09788935    -1.279104   -1.311052
## 5   -1.0184372  1.24503015    -1.335752   -1.311052
## 6   -0.5353840  1.93331463    -1.165809   -1.048667

#to visualize the distance matrix
distance <- get_dist(df)
fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white",
                                    high = "#FC4E07"))
```



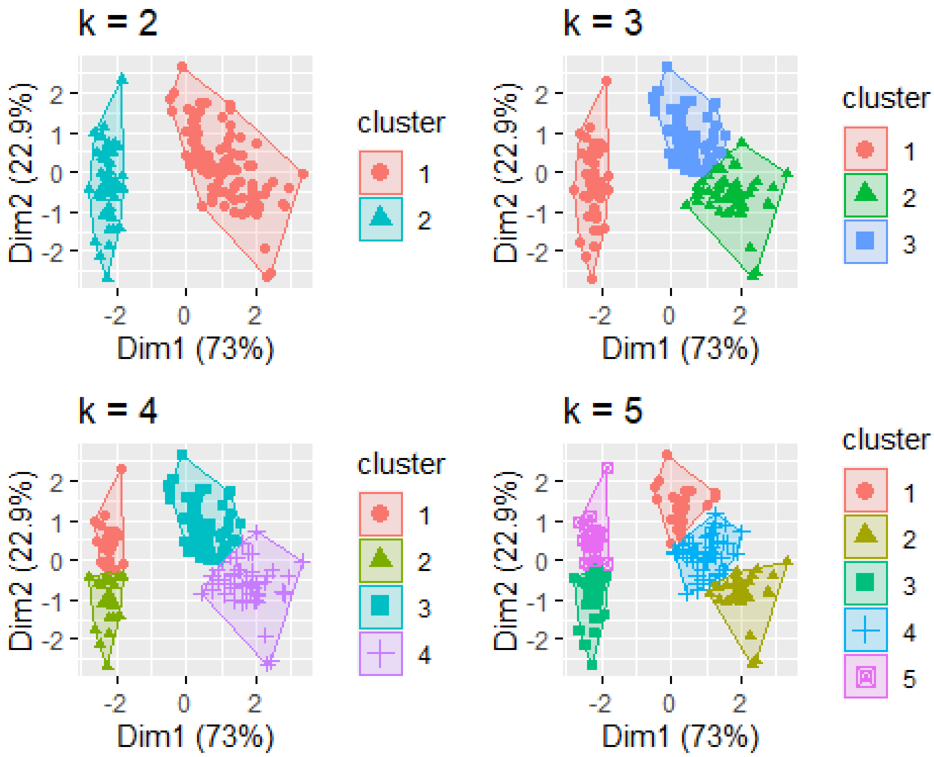

```
#k means clustering with 2 clusters
k2 <- kmeans(df, centers = 2, nstart = 25)
str(k2)

## List of 9
##  $ cluster     : Named int [1:150] 2 2 2 2 2 2 2 2 2 2 ...
##   ..- attr(*, "names")= chr [1:150] "1" "2" "3" "4" ...
##  $ centers     : num [1:2, 1:4] 0.506 -1.011 -0.425 0.85 0.65 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length"
"Petal.Width"
##  $ totss       : num 596
##  $ withinss    : num [1:2] 173.5 47.4
##  $ tot.withinss: num 221
##  $ betweenss   : num 375
##  $ size        : int [1:2] 100 50
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"

k2

## K-means clustering with 2 clusters of sizes 100, 50
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
```

```
## 1    0.5055957  -0.4252069     0.650315   0.6253518
## 2   -1.0111914   0.8504137    -1.300630  -1.2507035
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
19  20
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
2   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38
39  40
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
2   2
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58
59  60
##   2   2   2   2   2   2   2   2   2   2   1   1   1   1   1   1   1   1
1   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78
79  80
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
1   1
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98
99 100
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
1   1
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
119 120
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
1   1
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138
139 140
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
1   1
## 141 142 143 144 145 146 147 148 149 150
##   1   1   1   1   1   1   1   1   1   1
##
## Within cluster sum of squares by cluster:
## [1] 173.52867  47.35062
##  (between_SS / total_SS =  62.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

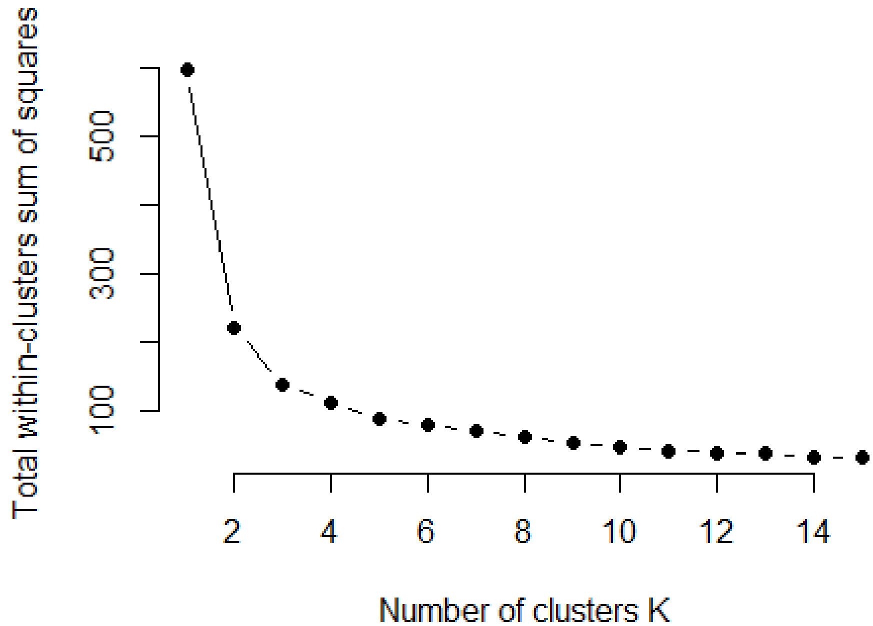
```r
#illustration of the clusters.
fviz_cluster(k2, data = df)
```




```
#execute the same process for 3, 4, and 5 clusters
k3 <- kmeans(df, centers = 3, nstart = 25)
k4 <- kmeans(df, centers = 4, nstart = 25)
k5 <- kmeans(df, centers = 5, nstart = 25)

# plots to compare
p1 <- fviz_cluster(k2, geom = "point", data = df) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = df) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",  data = df) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",  data = df) + ggtitle("k = 5")

library(gridExtra)

##
## Attaching package: 'gridExtra'

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

grid.arrange(p1, p2, p3, p4, nrow = 2)
```




```
#Determining Optimal Clusters
set.seed(123)

# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(df, k, nstart = 10 )$tot.withinss
}

# Compute and plot wss for k = 1 to k = 15
k.values <- 1:15

# extract wss for 2-15 clusters
wss_values <- map_dbl(k.values, wss)

plot(k.values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")
```




```r
set.seed(123)

fviz_nbclust(df, kmeans, method = "wss")
```