STAT495 (Advanced Data Analysis) HW#8

Tam Tran The

November 16, 2016

Exercise 9.2

Carry out and interpret a clustering of vehicles from another manufacturer using the approach outlined in section 9.1.1. (Hint: be sure that you have wrangled the data correctly by replicating the analyses from the book for Toyota vehicles.)

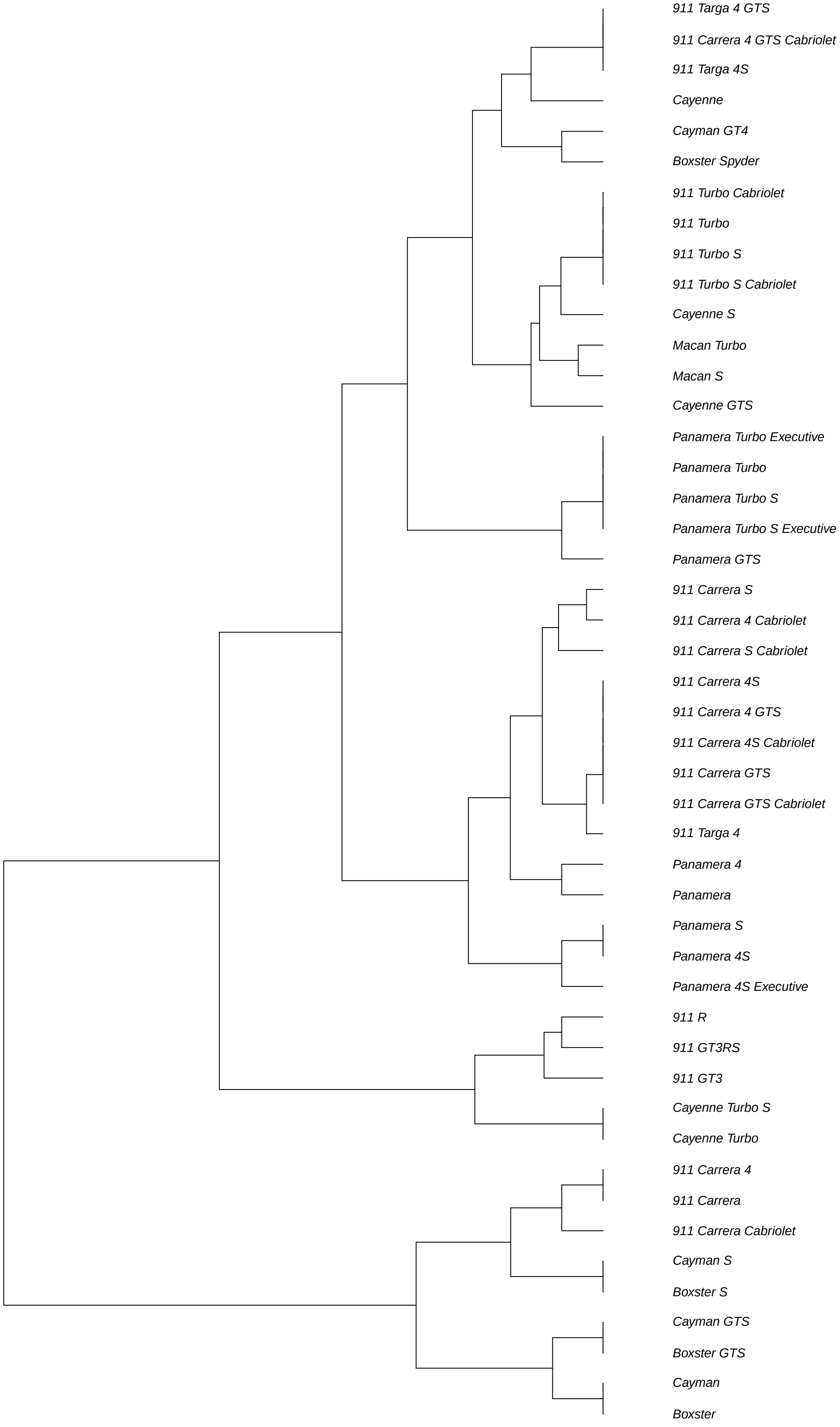
SOLUTION:

```
download.file("https://fueleconomy.gov/feg/epadata/16data.zip",
              destfile = "HW8/fueleconomy.zip")
unzip("HW8/fueleconomy.zip", exdir = "HW8/fueleconomy/")
```

```
library(mdsr)
library(readxl)
filename <- "2016 FE Guide for DOE-release dates before 8-10-2016-no-sales-8-9-2016FCVforpublicVW.xlsx"
cars <- read_excel(paste0("fueleconomy/", filename)) %>%
  data.frame()
cars <- cars %>%
  rename(make = Mfr.Name, model = Carline, displacement = Eng.Displ,
         cylinders = X..Cyl, city_mpg = City.FE..Guide....Conventional.Fuel,
         hwy_mpg = Hwy.FE..Guide....Conventional.Fuel, gears = X..Gears) %>%
  select(make, model, displacement, cylinders, gears, city_mpg, hwy_mpg) %>%
  distinct(model, .keep_all = TRUE) %>%
  filter(make == "Porsche")
rownames(cars) <- cars$model
glimpse(cars)
```

```
## Observations: 47
## Variables: 7
## $ make         <chr> "Porsche", "Porsche", "Porsche", "Porsche", "Pors...
## $ model        <chr> "911 GT3", "911 GT3RS", "Boxster", "Boxster GTS",...
## $ displacement <dbl> 3.8, 4.0, 2.7, 3.4, 3.4, 3.8, 2.7, 3.8, 3.4, 3.4,...
## $ cylinders    <dbl> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6...
## $ gears        <dbl> 7, 7, 7, 7, 7, 6, 7, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7...
## $ city_mpg     <dbl> 15, 14, 22, 22, 21, 18, 22, 18, 22, 21, 20, 20, 2...
## $ hwy_mpg      <dbl> 20, 20, 32, 31, 30, 24, 32, 23, 31, 30, 28, 28, 2...
```

```
car_diffs <- dist(cars)
library(ape)
p <- car_diffs %>%
  hclust() %>%
  as.phylo() %>%
  plot(cex = 0.9, label.offset = 1)
```


911 Targa 4 GTS
911 Carrera 4 GTS Cabriolet
911 Targa 4S
Cayenne
Cayman GT4
Boxster Spyder
911 Turbo Cabriolet
911 Turbo
911 Turbo S
911 Turbo S Cabriolet
Cayenne S
Macan Turbo
Macan S
Cayenne GTS
Panamera Turbo Executive
Panamera Turbo
Panamera Turbo S
Panamera Turbo S Executive
Panamera GTS
911 Carrera S
911 Carrera 4 Cabriolet
911 Carrera S Cabriolet
911 Carrera 4S
911 Carrera 4 GTS
911 Carrera 4S Cabriolet
911 Carrera GTS
911 Carrera GTS Cabriolet
911 Targa 4
Panamera 4
Panamera
Panamera S
Panamera 4S
Panamera 4S Executive
911 R
911 GT3RS
911 GT3
Cayenne Turbo S
Cayenne Turbo
911 Carrera 4
911 Carrera
911 Carrera Cabriolet
Cayman S
Boxster S
Cayman GTS
Boxster GTS
Cayman
Boxster

The first branch in the tree is between small-engine vehicles (with displacement lower than 3.4) and all others. This is expected since small-engine cars burn less gas, which is more efficient in terms of fuel economy.

The first branch among small-engine vehicles divides cars that are rated less than 20 mpg on city streets and less than 32 mpg on the highway, and cars with higher mpg rates. The inner branches keep separating vehicles based on their city mpg and highway mpg until all models have exactly the same combined fuel economy.

The first branch among larger-engine vehicles divides cars based on their displacement: the lower branch includes cars with displacement higher than 3.8, and the higher branch includes the rest with smaller engines. The inner branches separate cars in the same manner as mentioned above.

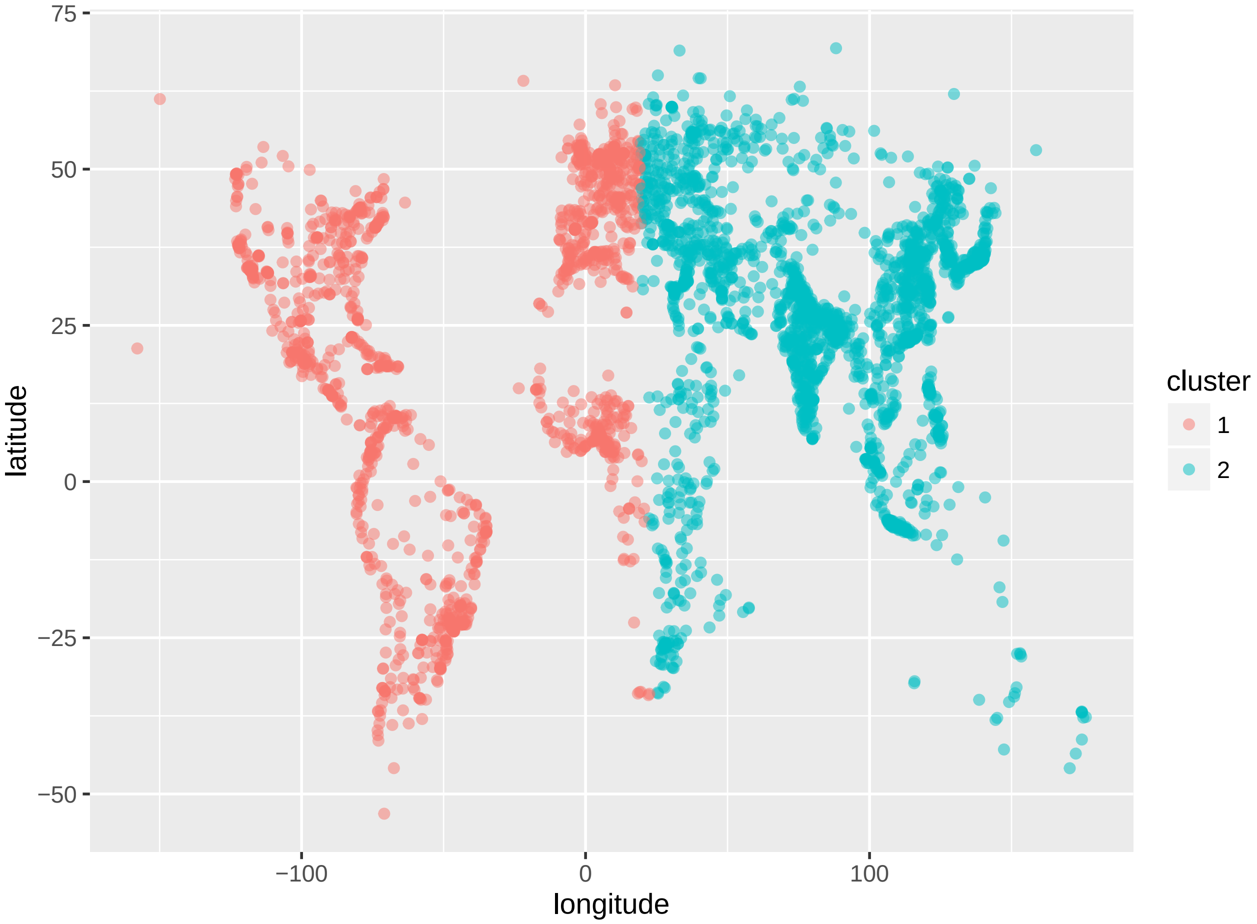
Because the resulting tree is pretty complex, there stands a chance that the inner branches are likely to overfit the data and only the first branch in the tree produces true signal.

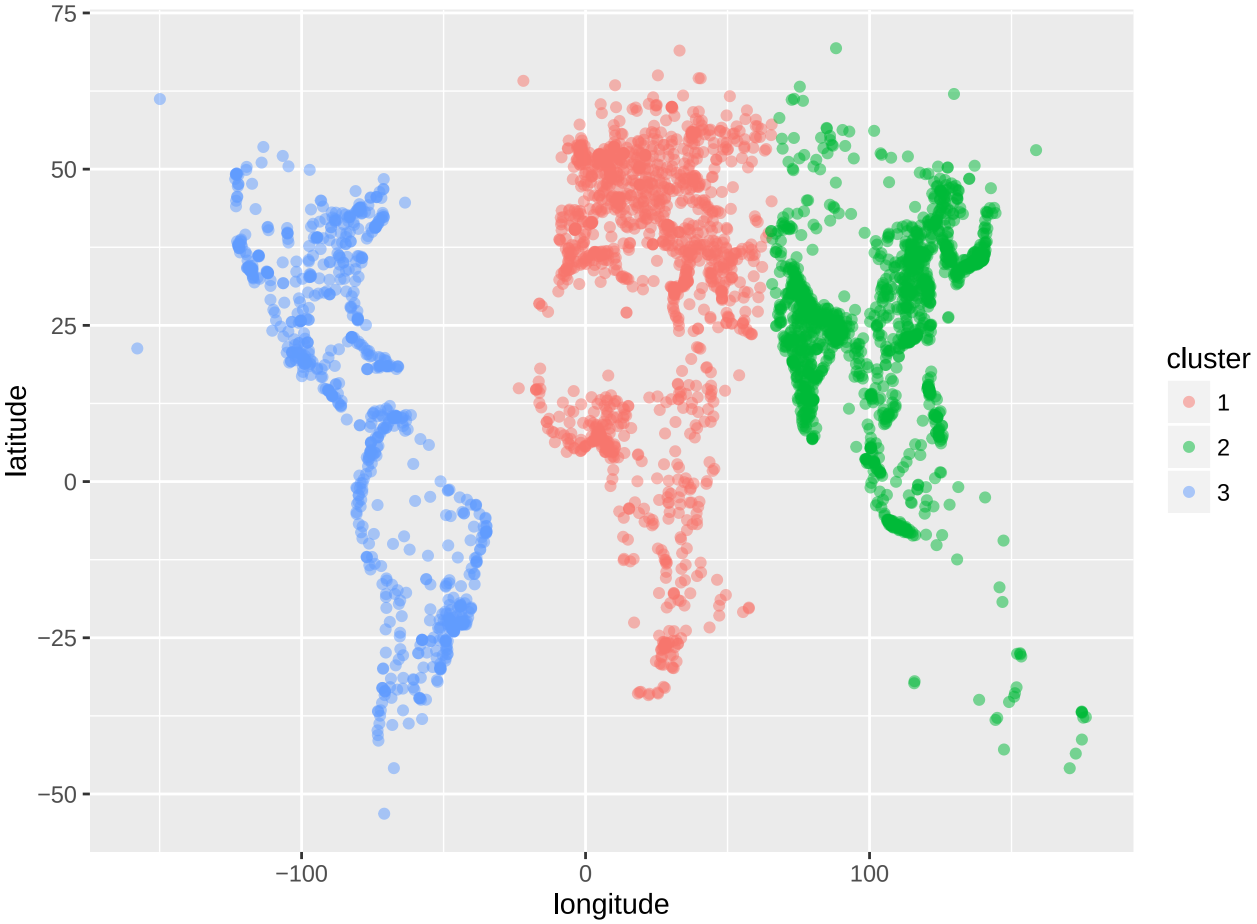
Exercise 9.4 Re-fit the k–means algorithm on the `BigCities` data with a different value of k (i.e., not six). Experiment with different values of k and report on the sensitivity of the algorithm to changes in this parameter.

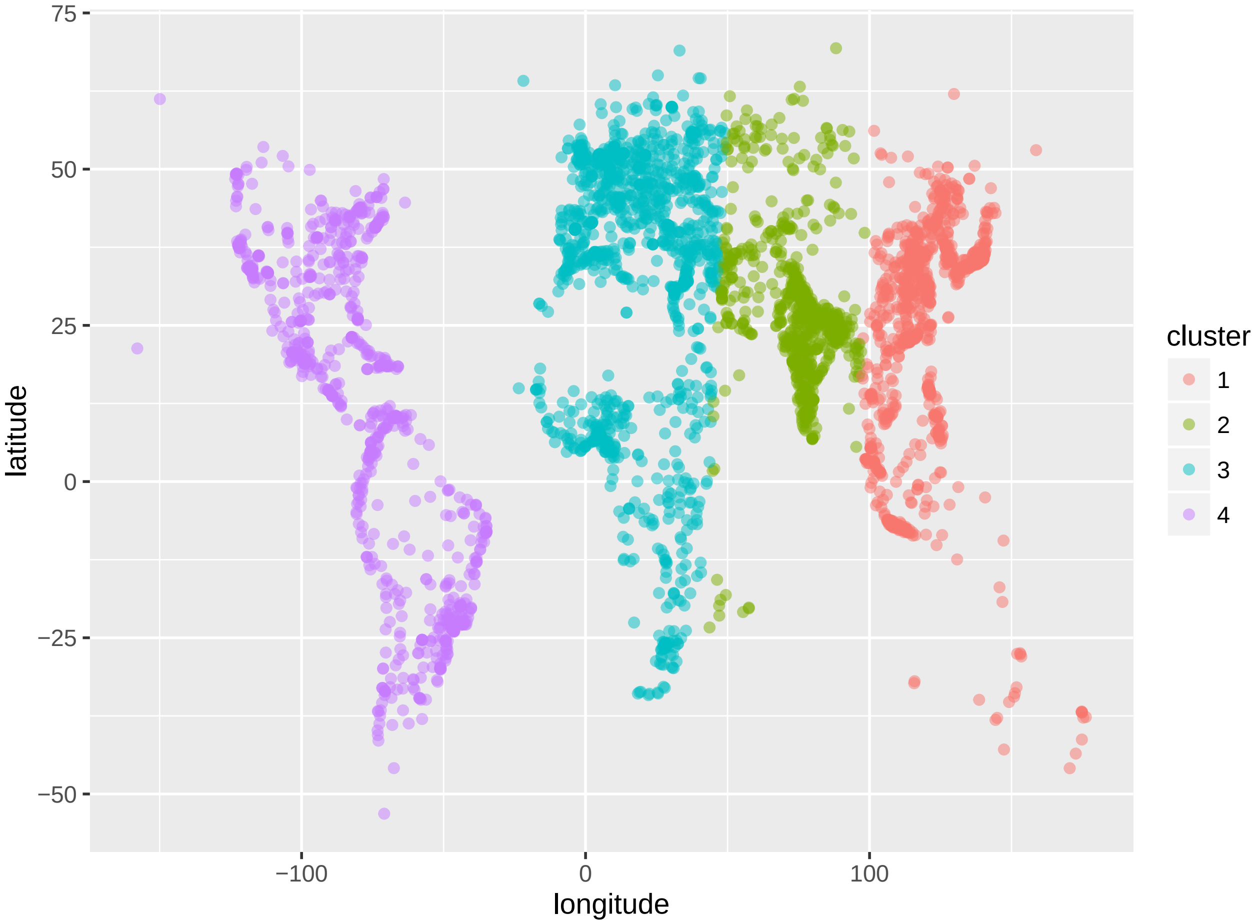
SOLUTION:

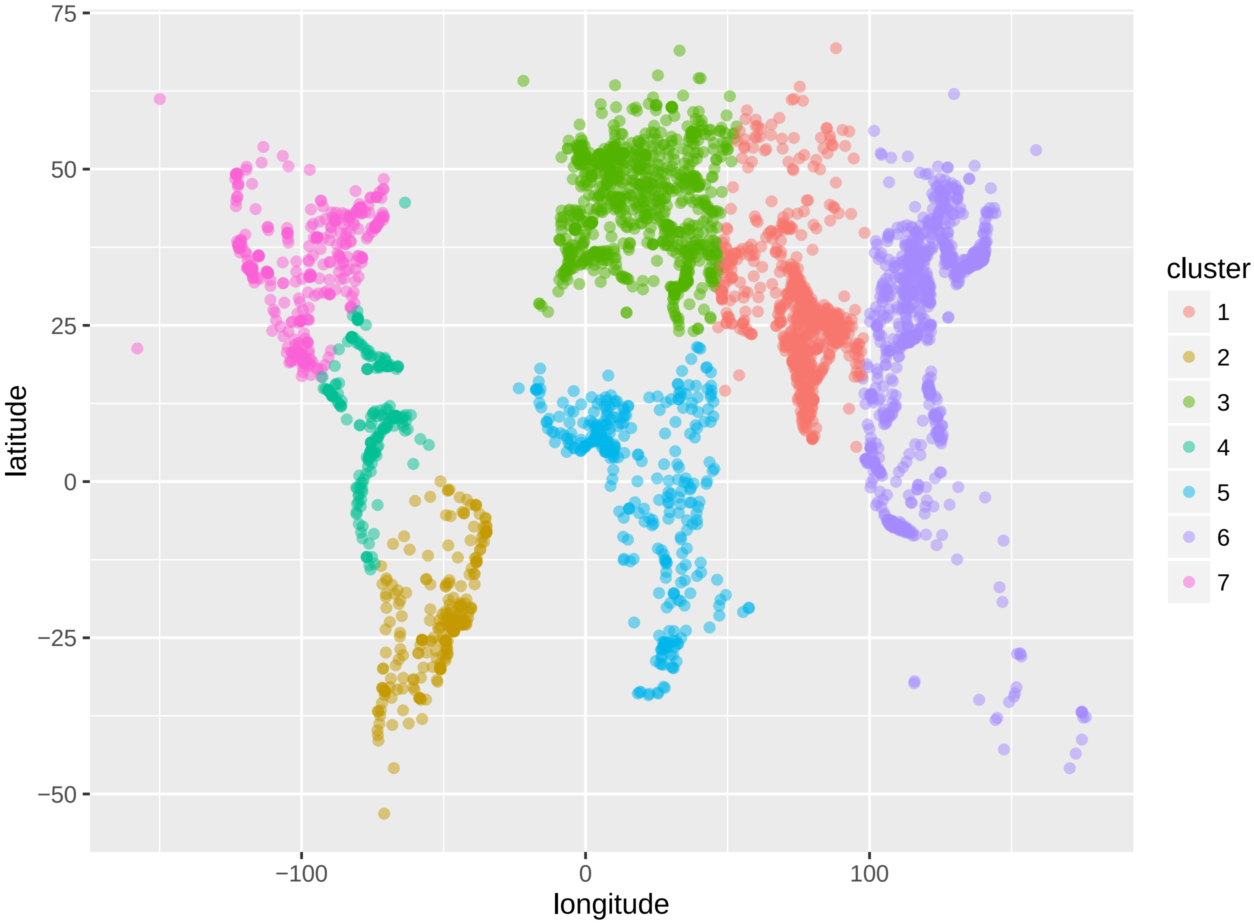
```
BigCities <- WorldCities %>%
  arrange(desc(population)) %>%
  head(4000) %>%
  select(longitude, latitude)
```

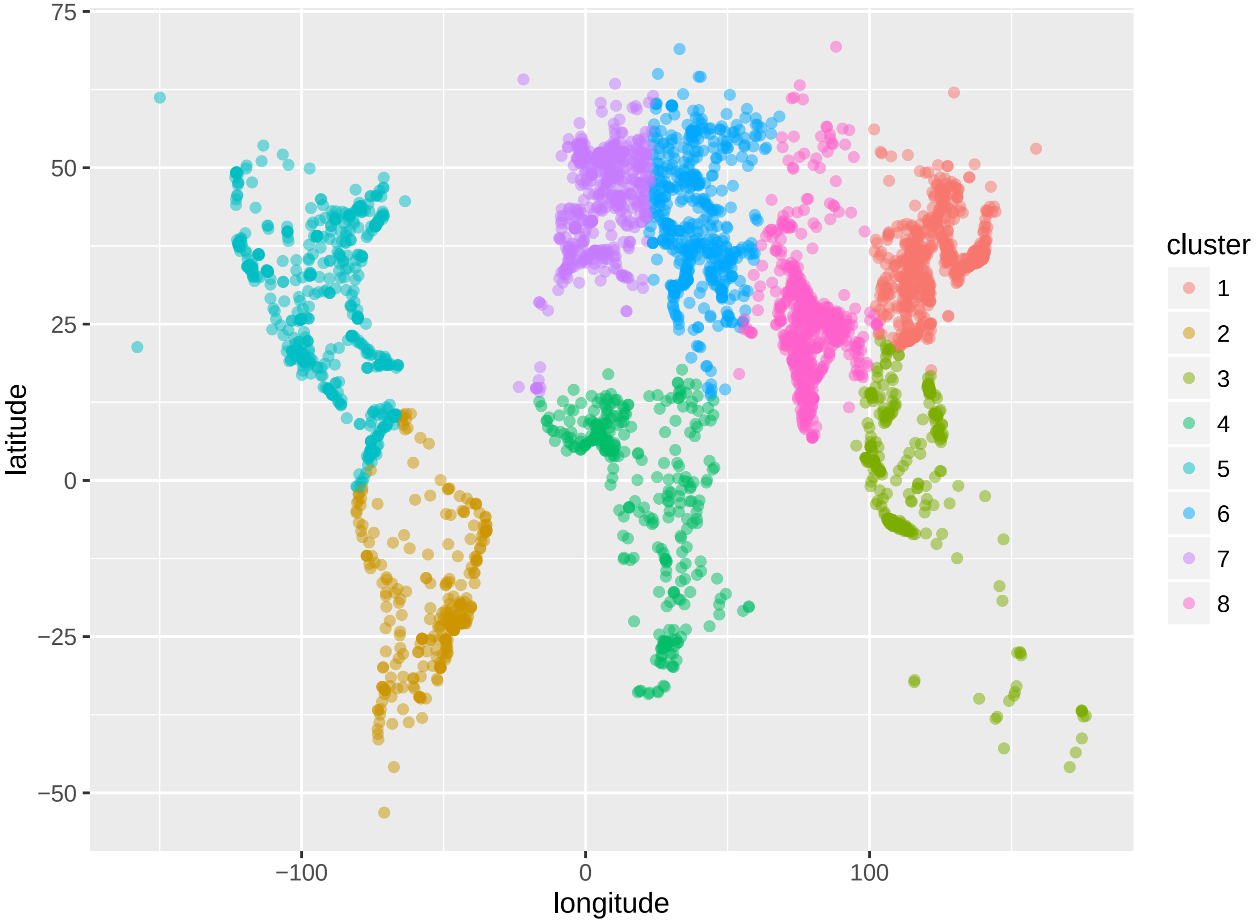
```
set.seed(10)
library(mclust)
plot <- function(i) {
  for (i in c(2, 3, 4, 7, 8)) {
    city_clusts <- BigCities %>%
      kmeans(centers = i) %>%
      fitted("classes") %>%
      as.character()
  BigCities <- BigCities %>% mutate(cluster = city_clusts)
  plots <- BigCities %>% ggplot(aes(x = longitude, y = latitude)) +
    geom_point(aes(color = cluster), alpha = 0.5)
  print(plots)
  }
}
plot()
```


75
50
25
0
−25
−50
latitude
−100
0
100
longitude
cluster
● 1
● 2


75
50
25
0
−25
−50
latitude
−100
0
100
longitude
cluster
1
2
3


75
50
25
0
−25
−50
latitude
−100
0
100
longitude
cluster
1
2
3
4


75
50
25
0
−25
−50
latitude
−100
0
100
longitude
cluster
1
2
3
4
5
6
7



With $k = 2$, the algorithm cannot wholly identify Europe and Africa. Instead, it separates the two continents into halves.

With $k = 3$, the algorithm groups North and South America into one cluster, Europe and Africa together, and accurately identify Asia.

With $k = 4$, while North and South America, and Europe and Africa are grouped together, East Asia and Central Asia are marked as distinct.

With $k = 7$, the algorithm separates Mexico, Cuba and the Caribbean islands from North America. It might be because these countries have significantly smaller population and area than the US and Canada in the same continent. East Asia and Central Asia are also separated.

With $k = 8$, the algorithm divides Europe into halves, Asia into three clusters, and correctly recognize South America, North America, and Africa.

As we can see, k–means clustering is very sensitive to the cluster centers that we choose. Poor choice of k may lead to incorrect clustering.