Homework 4

Anish Bhurtyal

2022-11-21

```
library(dplyr)
library(scatterplot3d)
library(magrittr)
library(ggplot2)
```

Problem 1

The Data. The 'maps' package contains a database of world cities i.e. 'world.cities' of population greater than about 40,000. Also included are capital cities of any population size, and many smaller towns.

The database constitutes a list with 6 components, namely "name", "country.etc", "pop", "lat", "long", and "capital", containing the city name, the country name, approx- imate population (as at January 2006), latitude, longitude and capital status indication (0 for non-capital, 1 for capital, 2 for China Municipalities, and 3 for China Provincial capitals).

Consider the 4,000 biggest cities in the world and their longitudes and latitudes:

```
library(maps)
big_cities <- world.cities %>%
  arrange(desc(pop)) %>%
  head(4000) %>%
  select(long, lat)

glimpse(big_cities)
```

```
## Rows: 4,000
## Columns: 2
## $ long <dbl> 121.47, 72.82, 67.01, -58.37, 77.21, 120.97, 37.62, 126.99, -46.6~
## $ lat  <dbl> 31.23, 18.96, 24.86, -34.61, 28.67, 14.62, 55.75, 37.56, -23.53, ~
```

Exploration Data Analysis (EDA)

Are there any missing data?

```
colSums(is.na(big_cities))
```

```
## long  lat
##    0    0
```

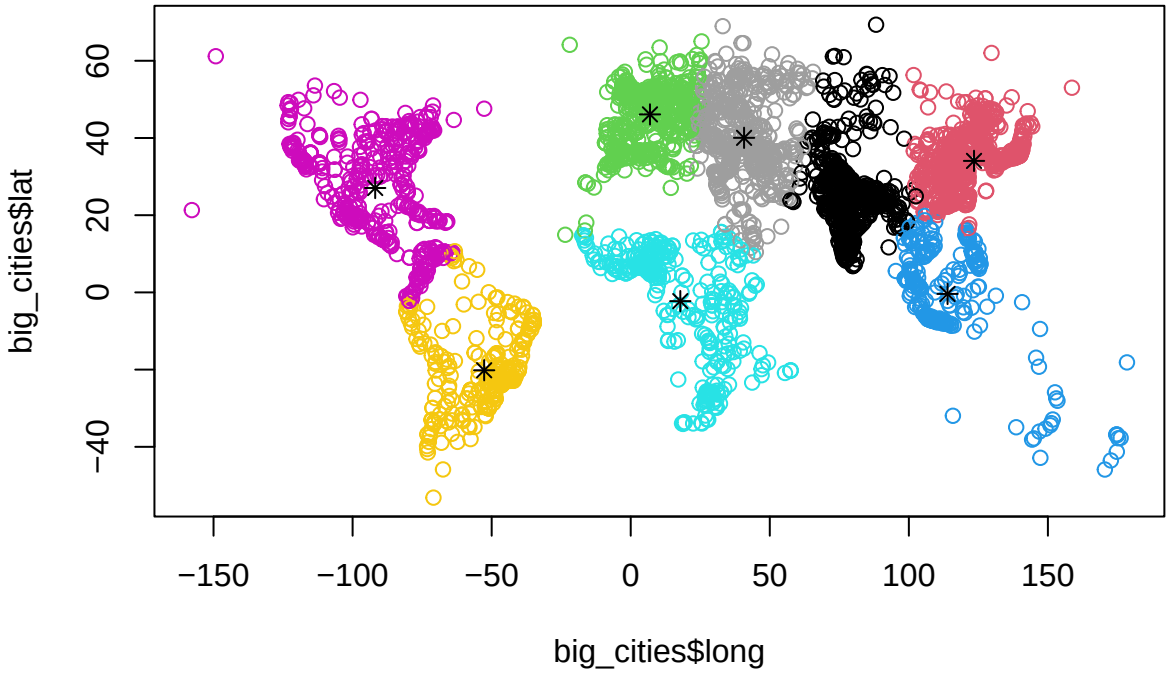
We dont have any missing values in the dataframe

```
str(big_cities)
```

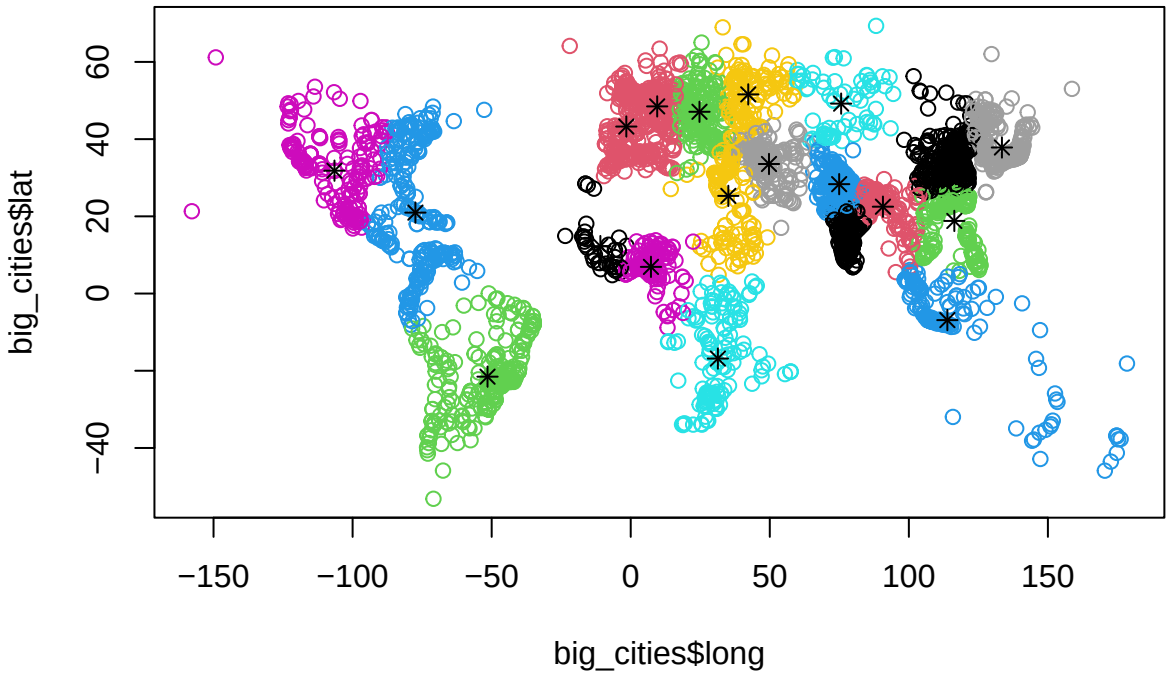
```
## 'data.frame':    4000 obs. of  2 variables:
##  $ long: num  121.5 72.8 67 -58.4 77.2 ...
```

```
##  $ lat : num  31.2 19 24.9 -34.6 28.7 ...
```

```
cl = kmeans(big_cities,8)
 #cl$cluster
 plot(big_cities$long, big_cities$lat,col=cl$cluster)
 points(cl$centers, pch=8)
```




```
cl = kmeans(big_cities,20)
 #cl$cluster
 plot(big_cities$long, big_cities$lat,col=cl$cluster)
 points(cl$centers, pch=8)
```



-Here

taking a different value of k in k-means clustering chaanges how clusters are formed around the centroids. As the value of K increases, there will be fewer elements in the cluster and cover smaller groups.

-Here when we first took k=8, we got bigger cluster size almost resembling those of continents, which are large in area

-When we then took k=20, we got smaller cluster size almost resembling those of sub-continents or geographical regions, which are smaller in area as compared to continents

```
#find optimal no of clusters
# library(cluster)
# gap.stat <- clusGap(big_cities, FUNcluster = kmeans, K.max = 25)
# #gap.stat
# library(factoextra)
# fviz_gap_stat(gap.stat)
```

Problem 2

Baseball players are voted into the Hall of Fame by the members of the Baseball Writers of America Association. Quantitative criteria are used by the voters, but they are also allowed wide discretion.The following code identifies the position players who have been elected to the Hall of Fame and tabulates a few basic statistics,including their number of career hits(H), home runs(HR), and stolen bases(SB).

Use the k-means algorithm to perform a cluster analysis on these players.Describe the properties that seem common to each cluster.

```
library(Lahman)
hof <- Batting %>%
  group_by(playerID) %>%
  inner_join(HallOfFame, by = c("playerID" = "playerID")) %>%
  filter(inducted == "Y" & votedBy == "BBWAA") %>%
  summarize(tH = sum(H), tHR = sum(HR), tRBI = sum(RBI), tSB = sum(SB)) %>%
  filter(tH > 1000)
```

```
head(hof)
```

```
## # A tibble: 6 x 5
##   playerID     tH   tHR  tRBI   tSB
##   <chr>     <int> <int> <int> <int>
## 1 aaronha01  3771   755  2297   240
## 2 alomaro01  2724   210  1134   474
## 3 aparilu01  2677    83   791   506
## 4 bagweje01  2314   449  1529   202
## 5 bankser01  2583   512  1636    50
## 6 benchjo01  2048   389  1376    68
```

```
clust <- hof %>%
  select(2:5)%>%
  kmeans(7)
clust$cluster
```

```
##  [1] 1 2 2 7 6 7 7 4 4 3 6 2 3 4 7 1 3 4 7 6 3 7 7 6 4 3 6 6 4 3 6 2 6 6 6 6 2 7
## [39] 3 4 5 7 5 7 1 7 7 4 2 1 1 6 6 7 5 2 7 6 6 6 3 6 6 5 7 6 4 2 7 4 7 5 7 7 5 1
## [77] 4 6 6 6 1 4
```

```
corrr::correlate(hof, method = "pearson", quiet=T)
```

```
## # A tibble: 4 x 5
##   term        tH      tHR   tRBI    tSB
```

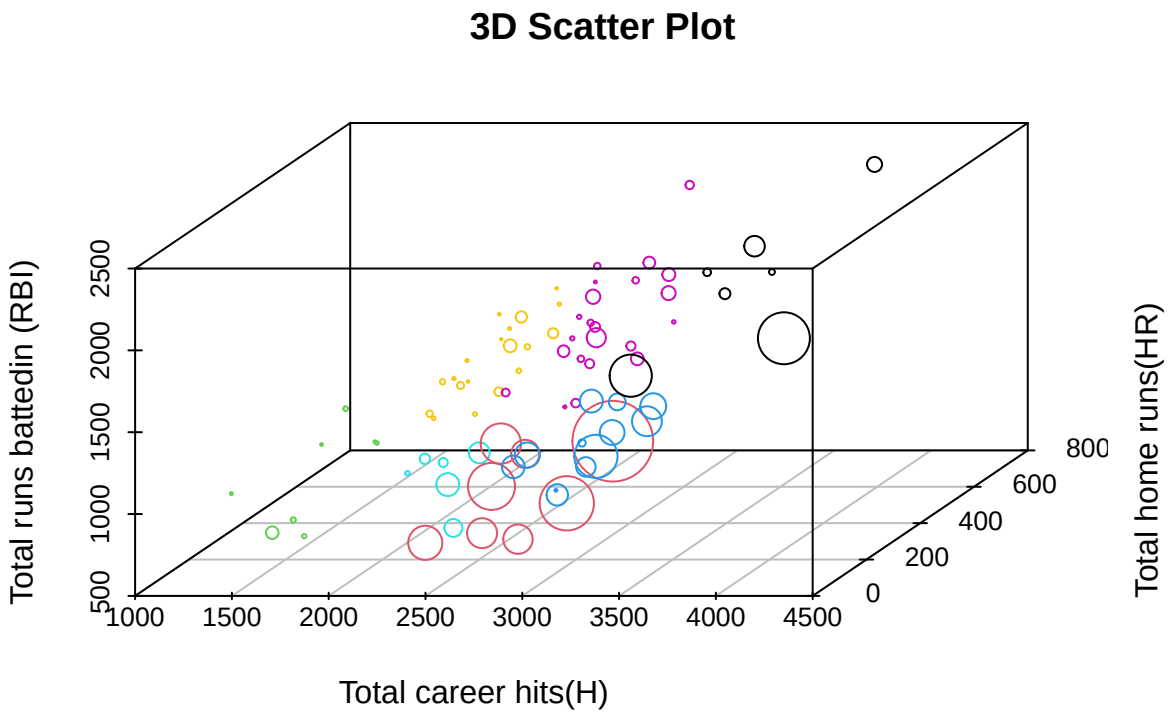
```
##   <chr>    <dbl>    <dbl>  <dbl>  <dbl>
## 1 tH    NA         0.00737  0.486  0.451
## 2 tHR    0.00737 NA        0.738 -0.393
## 3 tRBI   0.486    0.738   NA     -0.235
## 4 tSB    0.451   -0.393   -0.235 NA
```

-As we see we have 4 numerical variables, a 2D scatterplot could be used where we use 2 variables as axes and a third variable as bubble chart. We could use the fourth variable as color but it would be harder to visualize the cluster if we do so

-3D-scatter plot would be one suitable alternative for easy visualization of clusters

```
#normalizing the tSB to rescale to fit in graph
#hof$tSB <- (hof$tSB-min(hof$tSB))/(max(hof$tSB)-min(hof$tSB))
hof$tSB <- scale(hof$tSB)+1

hof  %>%
  select(2:4) %>%
  scatterplot3d(  color=clust$cluster,
                  cex.symbols = hof$tSB,
              main="3D Scatter Plot",
              sub = "bubble size corresponds to Total stolen bases(SB)",
              xlab = "Total career hits(H)",
              ylab = "Total home runs(HR)",
              zlab = "Total runs battedin (RBI)")
```



Looking at the clusters we see some good similarities between observations in the dataset. It gives us idea about which baseball players are similar in performance in terms of their attributes. Players within the same cluster with similar bubble size mean that the observation are identical to each others.

This can be helpful for team managers to replace players due to injury or for any other tactical and strategical

reasons using the cluster analysis in their team

-Another possible way, we could use PCA

```
hoff <- hof  %>% select(2:4)
require(graphics)
pca = princomp(hoff, cor=T) # principal components analysis using correlation matrix
pc.comp = pca$scores
pc.comp1 = -1*pc.comp[,1] # principal component 1 scores (negated for convenience)
pc.comp2 = -1*pc.comp[,2] # principal component 2 scores (negated for convenience)
```

```
X = cbind(pc.comp1, pc.comp2)
cl = kmeans(X,7)
cl$cluster
```

```
##  [1] 2 6 6 5 5 5 5 7 6 4 7 6 4 6 1 3 4 3 1 7 1 5 5 2 6 1 2 5 6 1 7 7 7 5 5 7 6 5
## [39] 4 3 1 5 6 5 2 5 1 3 1 2 2 2 7 5 1 6 5 7 7 2 4 7 2 1 5 7 6 6 5 3 5 1 5 5 6 3
## [77] 3 7 2 2 2 7
```

```
plot(pc.comp1, pc.comp2,col=cl$cluster)
points(cl$centers, pch=16)
```