task2

data cleaning

```
df.raw = read.csv("covid19-1.csv")
df = df.raw %>%
  janitor::clean_names() %>%
  dplyr::select(country_region, province_state, date, confirmed_cases) %>%
  filter(confirmed_cases != 0)

# region with confirm case > 20
df_country = df %>%
  group_by(country_region) %>%
  summarise(max=max(confirmed_cases)) %>%
  filter(max > 20)

region_index = as.character(unique(df_country$country_region))

df.region = function(df, region) {
  df.r = df %>%
    filter(country_region == region) %>%
    group_by(country_region, date) %>%
    summarise(cases = sum(confirmed_cases)) %>%
    mutate(formal_date = as.Date(date, '%m/%d/%Y')) %>%
    mutate(time = as.numeric(formal_date-min(formal_date)))%>%
    arrange(time)  %>%
    dplyr::select(region = country_region, date, time, cases)
  df.r
}

i= 1
df_list=vector("list", length = length(region_index))
while(i < length(region_index)+1){
  df_list[[i]]   = df.region(df, region_index[i])
  i = i+1
}

for (i in 1:length(df_list)){
names(df_list)[i] <- region_index[i]
}
```

```
res = read_csv("./abc_values") %>%
  dplyr::select(-X1) %>%
  mutate(
    a_value = round(a_value,0),
    b_value = round(b_value,3),
    c_value = round(c_value,0)
  )
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   country_region = col_character(),
##   a_value = col_double(),
##   b_value = col_double(),
##   c_value = col_double()
## )
```

```
res[which(res$country_region == "China"), 2] = 7.873205*10^4
res[which(res$country_region == "China"), 3] = 0.22511
res[which(res$country_region == "China"), 4] = 17.91412

res[which(res$country_region == "Korea, South"), 2] = 8.801392e+03
res[which(res$country_region == "Korea, South"), 3] = 2.836325e-01
res[which(res$country_region == "Korea, South"), 4] = 4.038837e+01


all_t=NULL
for(c in 1:length(df_list))
all_t=rbind(all_t,df_list[[c]][nrow(df_list[[c]]),3])

sum((res[,4])<all_t[,1])# total country pass the mid point
```

```
## [1] 27
```

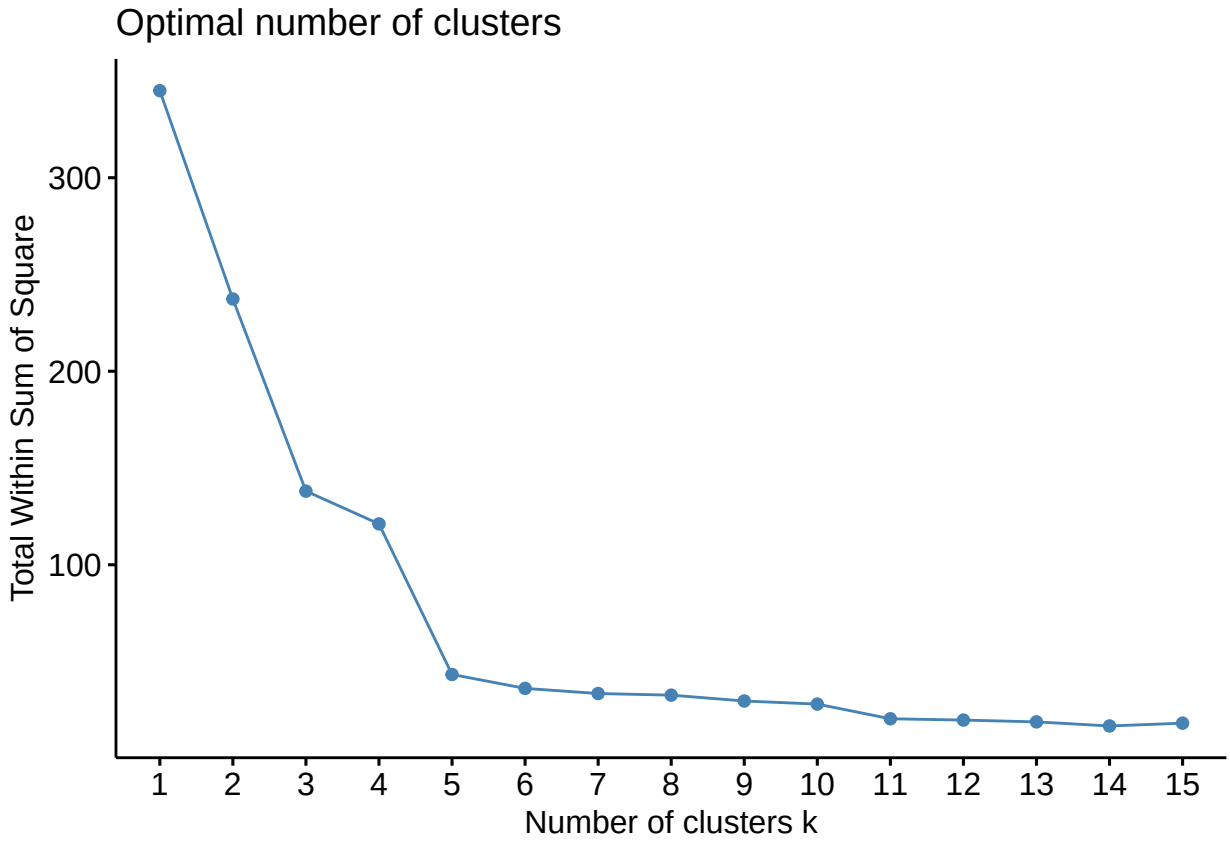
```
names=res[which((res[,4])<all_t[,1]),1]# names of those country
```

Task2

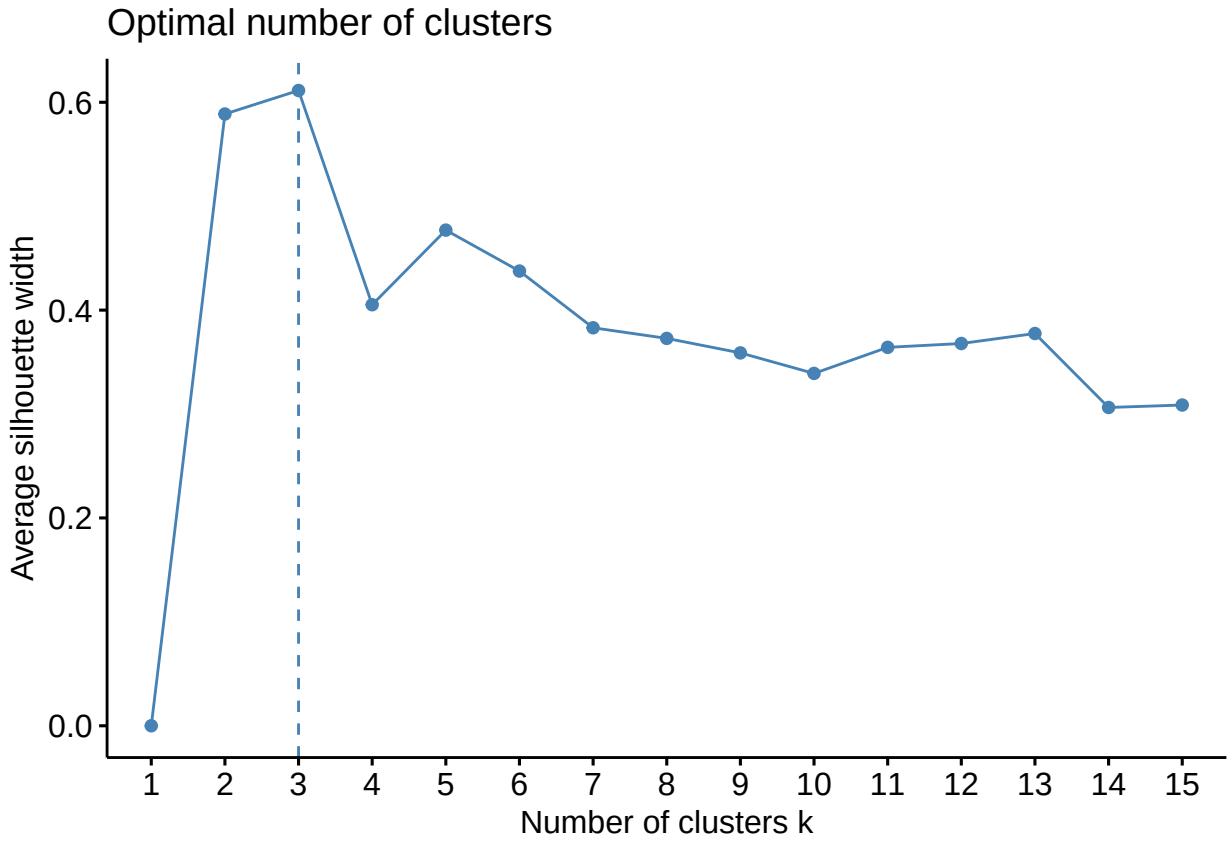
```
EM = function(data,ncluster){
  data = as.matrix(data) %>% scale()
  n = nrow(data)
  q = ncol(data)
  p_j = rep(1/ncluster,ncluster)
  mu = data[sample(n,ncluster),] %>% as.matrix()
  covmat = diag(ncol(data))
  covlist = list()
  for(i in 1:ncluster){
    covlist[[i]] = covmat
  }

count = 1
while(count <100){
  mu0 <- mu

  # E-step: Evaluate posterior probability, gamma
  gamma <- c()
  for(j in 1:ncluster){
    gamma2 <- apply(data,1, dmvnorm, mean = mu[j,], sigma = covlist[[j]])
    gamma <- cbind(gamma, gamma2)
  }
```

```
  # M- step: Calculate mu
 tempmat <- matrix(rep(p_j,n),nrow=n,byrow = T)
 r <- (gamma * tempmat) / rowSums(gamma * tempmat)
 mu <- t(r) %*% data / colSums(r)


 # M- step: Calculate Sigma and p
 for(j in 1:ncluster){
   sigma <- matrix(rep(0,q^2),ncol=q)
   for(i in 1:n){
     sigma = sigma + r[i,j] * (data[i,]-mu0[j,]) %*% t(data[i,]-mu0[j,])       }
   covlist[[j]] <- sigma/sum(r[,j])     }
 p_j <- colSums(r)/n
 count = count + 1  }

 cluster <- which(r == apply(r, 1, max), arr.ind = T)
 cluster <- cluster[order(cluster[,1]),]
 return(list(mu = mu,covlist = covlist, p_j = p_j,cluster = cluster)) }
```

```
em_dat = res %>% dplyr::select(-country_region)
em_dat_scaled <- scale(em_dat)

## use wss
fviz_nbclust(em_dat_scaled, kmeans, method = "wss",k.max = 15)
```




```
## use silhouette
fviz_nbclust(em_dat_scaled, kmeans, method = "silhouette",k.max=15)
```

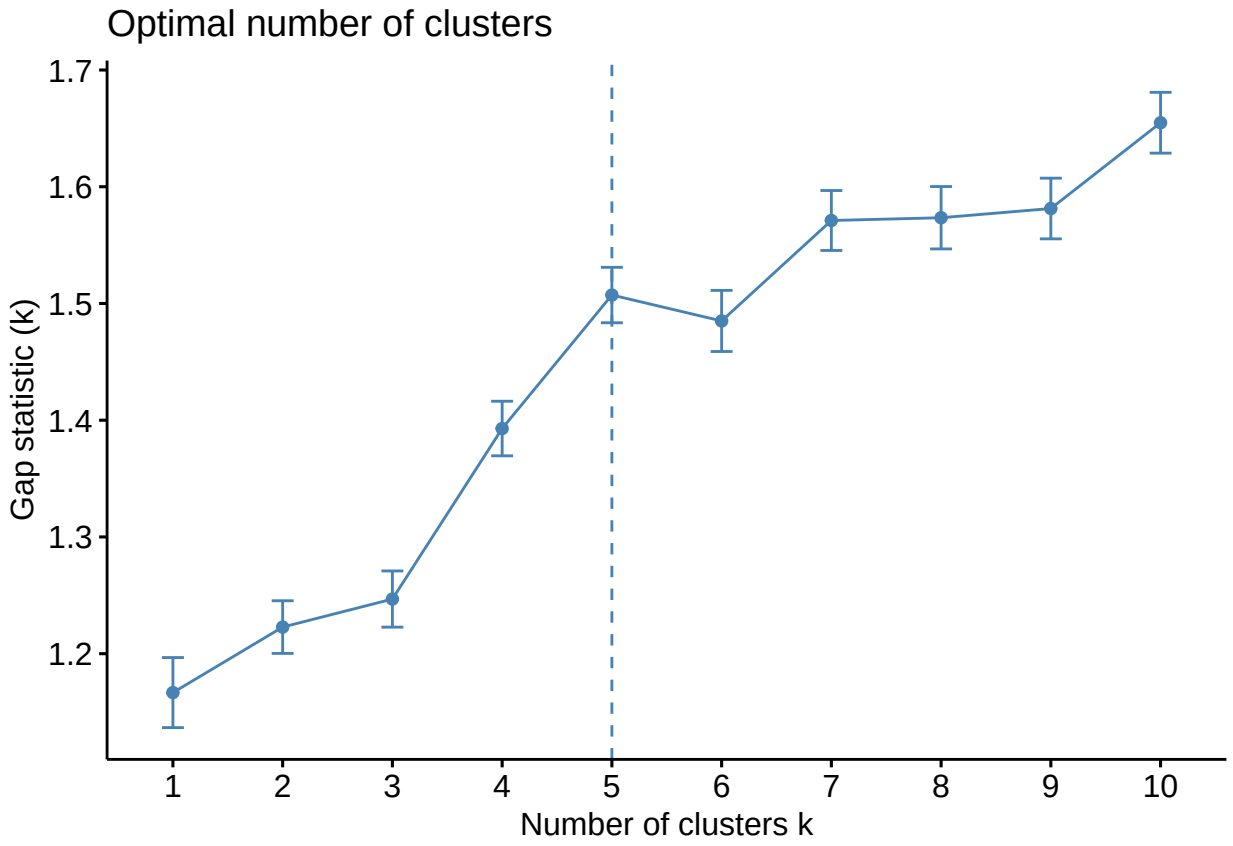



```
## use Gap Statistic Method
library(cluster)
```

```
##
## Attaching package: 'cluster'
```

```
## The following object is masked from 'package:maps':
##
##     votes.repub
```

```
set.seed(123)
gap_stat <- clusGap(em_dat_scaled, FUN = kmeans, nstart = 25,
                    K.max = 10, B = 50)
fviz_gap_stat(gap_stat)
```




```
set.seed(1)
res2 = EM(em_dat,3)
res3=res2$mu %>%
  as.data.frame()
res3
```

```
##             a_value    b_value    c_value
## gamma2   -0.2992109  0.4627679 -0.8220418
## gamma2.1  1.8053018 -0.2416451  0.7382552
## gamma2.2 -0.2476236 -0.1832539  0.2569219
```

```
set.seed(1)
clusters = kmeans(em_dat,3)
clusternumbers = as.factor(clusters$cluster)
```

```
## res4 is the classification result of em and kmeans
res4 = res2$cluster %>%
  as.data.frame() %>%
  dplyr::select(-1) %>%
  mutate(country = region_index ) %>%
  dplyr::select(2,1) %>%
  rename(GMM_class = col) %>%
  mutate(
    kmeans_class = clusters$cluster
  ) %>%
  mutate(
    a_value = res$a_value,
    b_value = res$b_value,
```

```
    c_value = res$c_value
  )

res4[1:50,] %>%
  knitr::kable(booktabs = T, align = 'c')
```

country	GMM_class	kmeans_class	a_value	b_value	c_value
Afghanistan	3	1	342.00	0.20200	37.00000
Albania	1	1	269.00	0.17300	17.00000
Algeria	3	1	723.00	0.25800	30.00000
Andorra	1	1	345.00	0.34400	22.00000
Argentina	3	1	970.00	0.31500	23.00000
Armenia	3	1	514.00	0.28800	23.00000
Australia	3	1	4072.00	0.29300	58.00000
Austria	2	1	10760.00	0.27500	28.00000
Azerbaijan	3	1	365.00	0.18400	30.00000
Bahrain	3	1	795.00	0.11800	29.00000
Bangladesh	3	1	99.00	0.24400	18.00000
Belarus	3	1	102.00	0.27600	19.00000
Belgium	2	1	8530.00	0.25400	49.00000
Bolivia	1	1	81.00	0.19200	16.00000
Bosnia and Herzegovina	1	1	352.00	0.29200	19.00000
Brazil	3	1	4507.00	0.38000	27.00000
Brunei	1	1	98.00	0.38100	7.00000
Bulgaria	3	1	459.00	0.25300	16.00000
Burkina Faso	1	1	252.00	0.36300	14.00000
Cambodia	3	1	168.00	0.31700	56.00000
Canada	3	1	5462.00	0.33800	58.00000
Chile	3	1	1862.00	0.31800	21.00000
China	2	3	78732.05	0.22511	17.91412
Colombia	3	1	777.00	0.33500	18.00000
Congo (Kinshasa)	1	1	115.00	0.36000	14.00000
Costa Rica	1	1	375.00	0.26800	18.00000
Cote d'Ivoire	1	1	342.00	0.85700	15.00000
Croatia	3	1	958.00	0.31000	29.00000
Cuba	1	1	122.00	0.36300	13.00000
Cyprus	1	1	272.00	0.23400	15.00000
Denmark	3	1	3258.00	0.17000	24.00000
Dominican Republic	3	1	640.00	0.49800	23.00000
Ecuador	3	1	2180.00	0.44900	23.00000
Egypt	3	1	806.00	0.19300	39.00000
Estonia	3	1	569.00	0.23500	22.00000
Finland	3	1	1570.00	0.21600	55.00000
France	2	3	39932.00	0.14800	64.00000
Georgia	3	1	151.00	0.14000	29.00000
Germany	2	3	65957.00	0.25900	57.00000
Ghana	1	1	300.00	0.33200	15.00000
Greece	3	1	1499.00	0.18200	27.00000
Guatemala	1	1	23.00	0.58900	6.00000
Honduras	1	1	32.00	0.54900	8.00000
Hungary	1	1	393.00	0.26600	20.00000
Iceland	3	1	1311.00	0.21300	25.00000
India	3	1	1060.00	0.25300	54.00000
Indonesia	3	1	1389.00	0.26600	22.00000
Iran	2	3	49441.00	0.13100	33.00000
Iraq	3	1	642.00	0.14300	30.00000
Ireland	3	1	2673.00	0.30900	24.00000

```r
res4[51:100,] %>%
  knitr::kable(booktabs = T, align = 'c')
```

	country	GMM_class	kmeans_class	a_value	b_value	c_value
51	Israel	3	1	4055.000	0.3040000	33.00000
52	Italy	2	2	138340.000	0.1830000	53.00000
53	Jamaica	1	1	20.000	0.3310000	5.00000
54	Japan	3	1	2195.000	0.0940000	60.00000
55	Jordan	1	1	326.000	0.3020000	21.00000
56	Kazakhstan	1	1	69.000	0.5290000	5.00000
57	Kenya	1	1	237.000	0.3200000	18.00000
58	Korea, South	2	1	8801.392	0.2836325	40.38837
59	Kuwait	3	1	564.000	0.0880000	36.00000
60	Kyrgyzstan	1	1	279.000	0.5460000	9.00000
61	Latvia	1	1	411.000	0.2700000	22.00000
62	Lebanon	3	1	829.000	0.1690000	35.00000
63	Liechtenstein	1	1	55.000	0.5000000	15.00000
64	Lithuania	1	1	432.000	0.4510000	25.00000
65	Luxembourg	3	1	2213.000	0.3540000	24.00000
66	Malaysia	3	1	3231.000	0.2220000	59.00000
67	Malta	1	1	242.000	0.2480000	17.00000
68	Martinique	1	1	135.000	0.2510000	18.00000
69	Mauritius	1	1	115.000	0.4920000	7.00000
70	Mexico	3	1	748.000	0.3170000	25.00000
71	Moldova	1	1	273.000	0.2850000	16.00000
72	Monaco	3	1	60.000	0.2720000	25.00000
73	Montenegro	1	1	124.000	0.5070000	8.00000
74	Morocco	1	1	357.000	0.2910000	22.00000
75	Netherlands	2	1	11170.000	0.2390000	26.00000
76	New Zealand	1	1	505.000	0.4200000	27.00000
77	Nigeria	3	1	102.000	0.4070000	25.00000
78	North Macedonia	3	1	309.000	0.3250000	27.00000
79	Norway	2	1	5557.000	0.1750000	26.00000
80	Oman	3	1	361.000	0.1250000	40.00000
81	Pakistan	3	1	1774.000	0.3260000	26.00000
82	Panama	3	1	715.000	0.3210000	14.00000
83	Paraguay	3	1	74.000	0.1950000	19.00000
84	Peru	3	1	678.000	0.3220000	16.00000
85	Philippines	3	1	1091.000	0.2400000	54.00000
86	Poland	3	1	1821.000	0.2830000	20.00000
87	Portugal	3	1	4741.000	0.3350000	22.00000
88	Qatar	3	1	889.000	0.1750000	19.00000
89	Romania	3	1	1783.000	0.2560000	29.00000
90	Russia	3	1	979.000	0.2910000	53.00000
91	Rwanda	1	1	107.000	0.3560000	11.00000
92	San Marino	1	1	230.000	0.1910000	19.00000
93	Saudi Arabia	3	1	1551.000	0.2880000	23.00000
94	Senegal	3	1	357.000	0.2170000	27.00000
95	Serbia	3	1	627.000	0.2860000	18.00000
96	Singapore	3	1	1262.000	0.0850000	67.00000
97	Slovakia	1	1	254.000	0.3320000	13.00000
98	Slovenia	3	1	805.000	0.2000000	16.00000
99	South Africa	3	1	1303.000	0.3430000	20.00000
100	Spain	2	3	79759.000	0.2570000	52.00000

```
res4[101:116,] %>%
  knitr::kable(booktabs = T, align = 'c')
```

	country	GMM_class	kmeans_class	a_value	b_value	c_value
101	Sri Lanka	3	1	105	0.459	51
102	Sweden	3	1	4381	0.171	52
103	Switzerland	2	1	19766	0.261	28
104	Taiwan*	3	1	576	0.097	70
105	Thailand	3	1	1634	0.306	62
106	Trinidad and Tobago	1	1	53	3.857	6
107	Tunisia	1	1	419	0.242	24
108	Turkey	3	1	3770	0.537	13
109	Ukraine	1	1	212	0.395	21
110	United Arab Emirates	3	1	652	0.114	62
111	United Kingdom	2	1	16258	0.279	53
112	Uruguay	1	1	184	0.548	6
113	US	2	2	106991	0.389	29
114	Uzbekistan	1	1	50	0.729	4
115	Venezuela	1	1	95	0.426	5
116	Vietnam	3	1	418	0.102	69

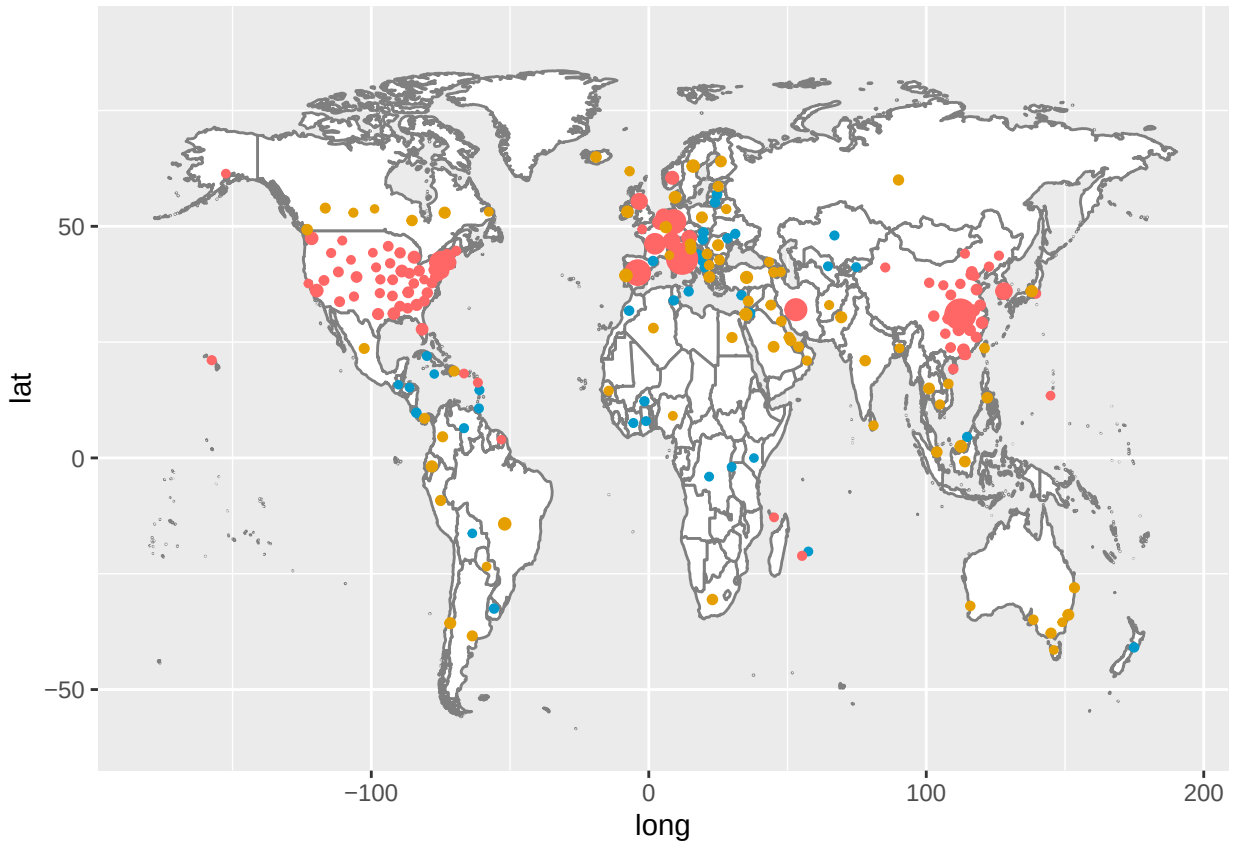
```
kmean_mean = clusters$centers %>%
  as.data.frame()

a_mean = mean(res$a_value)
a_sd = sd(res$a_value)
b_mean = mean(res$b_value)
b_sd = sd(res$b_value)
c_mean = mean(res$c_value)
c_sd = sd(res$c_value)
em_mean = res3 %>%
  mutate(
    a_value = a_value*a_sd+a_mean,
    b_value = b_value*b_sd+b_mean,
    c_value = c_value*c_sd+c_mean,
  )

mean_value = rbind(em_mean,kmean_mean) %>%
  mutate(method = c("GMM","GMM","GMM","Kmeans","Kmeans","Kmeans")) %>%
  dplyr::select(c(4,1,2,3))

mean_value %>%
  knitr::kable(booktabs = T, align = 'c')
```

method	a_value	b_value	c_value
GMM	212.797	0.4922645	14.14861
GMM	43289.675	0.2424189	39.97001
GMM	1268.728	0.2631294	32.00442
Kmeans	1614.435	0.3345930	26.72833
Kmeans	122665.500	0.2860000	41.00000
Kmeans	62764.210	0.2040220	44.78282

```
df_map = df.raw %>%
  janitor::clean_names() %>%
  dplyr::select(country_region, province_state, date, confirmed_cases, lat,long) %>%
  filter(confirmed_cases != 0) %>%
  group_by(country_region,lat,long) %>%
  summarise(max=max(confirmed_cases)) %>%
  filter(max > 20) %>%
  as.data.frame() %>%
  mutate(
    country_region = as.character(country_region)
  )

GMM_class = NULL
kmeans_class = NULL
for(i in 1:212){
  GMM_class[i] = res4[which(res4$country == df_map[i,1]),2]
  kmeans_class[i] = res4[which(res4$country == df_map[i,1]),3]
}

df_map = df_map %>%
  mutate(
    GMM_class = GMM_class,
    kmeans_class = kmeans_class
  )

mp<-NULL

mapworld<-borders("world",colour = "gray50",fill="white") #

mp<-ggplot()+mapworld+ylim(-60,90)

df_map_1 = df_map %>% filter(GMM_class ==1)
df_map_2 = df_map %>% filter(GMM_class ==2)
df_map_3 = df_map %>% filter(GMM_class ==3)

mp2<-mp+geom_point(aes(x=long,y=lat,size=max),data = df_map_1,color="#0099CC")+scale_size(range=c(1,5))
geom_point(aes(x=long,y=lat,size=max),data = df_map_2,color="#FF6666")+scale_size(range=c(1,5)) +
geom_point(aes(x=long,y=lat,size=max),data = df_map_3,color="#E69F00")+scale_size(range=c(1,5)) +
theme(legend.position = "none")
```

```
## Scale for 'size' is already present. Adding another scale for 'size', which
## will replace the existing scale.
## Scale for 'size' is already present. Adding another scale for 'size', which
## will replace the existing scale.
```

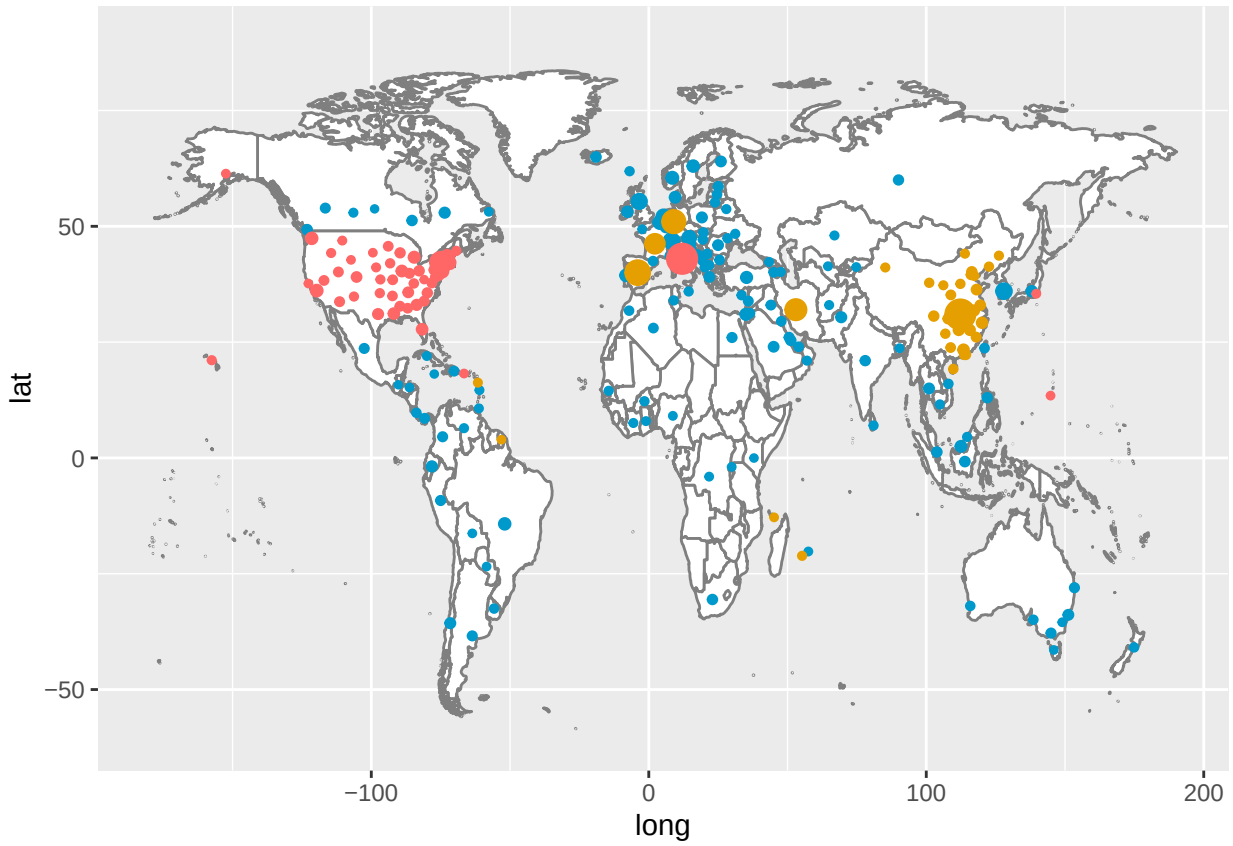
```
## blue class1   red class2  yellow class3
mp2
```



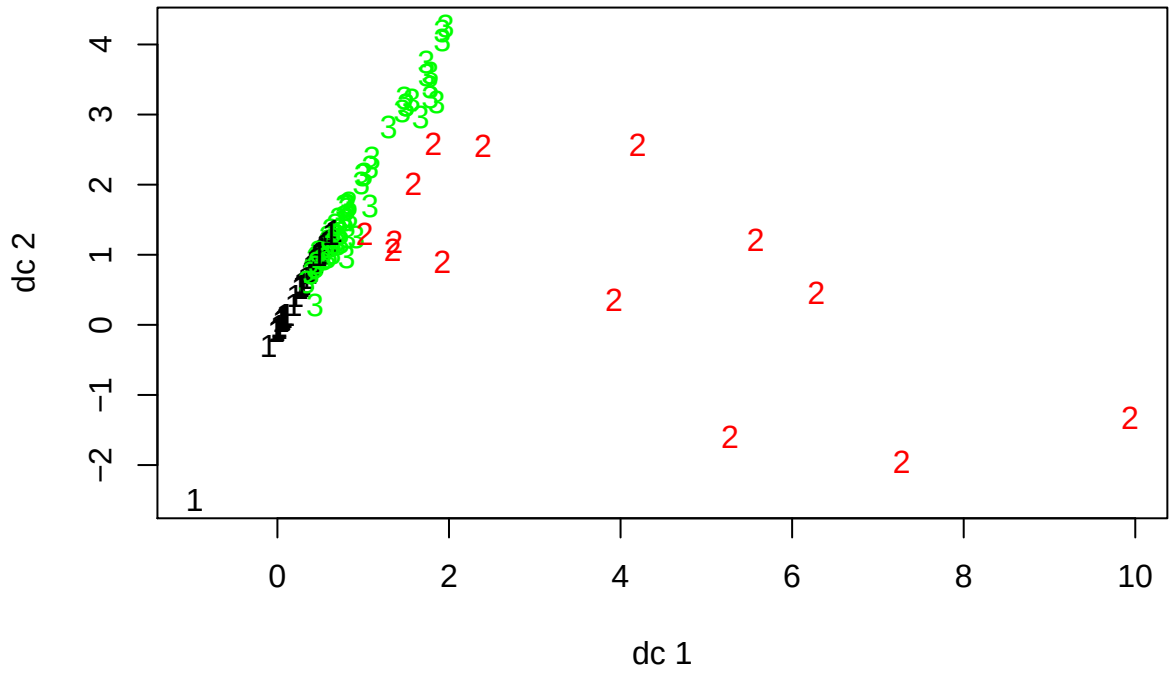

```
df_map_5 = df_map %>% filter(kmeans_class ==1)
df_map_6 = df_map %>% filter(kmeans_class ==2)
df_map_7 = df_map %>% filter(kmeans_class ==3)

mp3<-mp+geom_point(aes(x=long,y=lat,size=max),data = df_map_5,color="#0099CC")+scale_size(range=c(1,5))+
geom_point(aes(x=long,y=lat,size=max),data = df_map_7,color="#E69F00")+scale_size(range=c(1,5))+
theme(legend.position = "none")
```

```
## Scale for 'size' is already present. Adding another scale for 'size', which
## will replace the existing scale.
## Scale for 'size' is already present. Adding another scale for 'size', which
## will replace the existing scale.
```

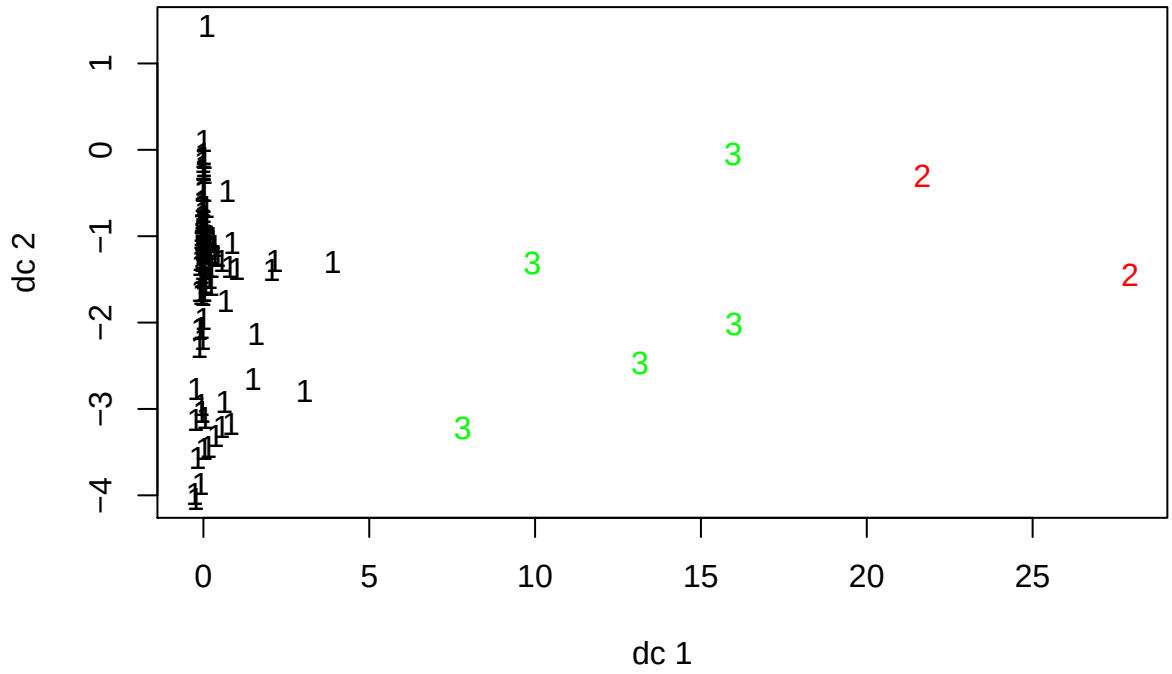
```
mp3 #
```




```
d <- dist(em_dat_scaled, method = "euclidean")

plotcluster(em_dat, res2$cluster[,2])
```



```
plotcluster(em_dat, clusters$cluster)
```




```
km_stats <- cluster.stats(d,  clusters$cluster)
km_stats$dunn
```

```
## [1] 0.122663
```

```
gmm_stats <- cluster.stats(d,  res2$cluster[,2])
gmm_stats$dunn
```

```
## [1] 0.00130779
```

```
method = c("Kmeans", "GMM")
Dunn_index = round(c(km_stats$dunn,gmm_stats$dunn),4)
cbind(method,Dunn_index) %>% as.data.frame() %>%
  knitr::kable(booktabs = T, align = 'c')
```

method	Dunn_index
Kmeans	0.1227
GMM	0.0013