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
```

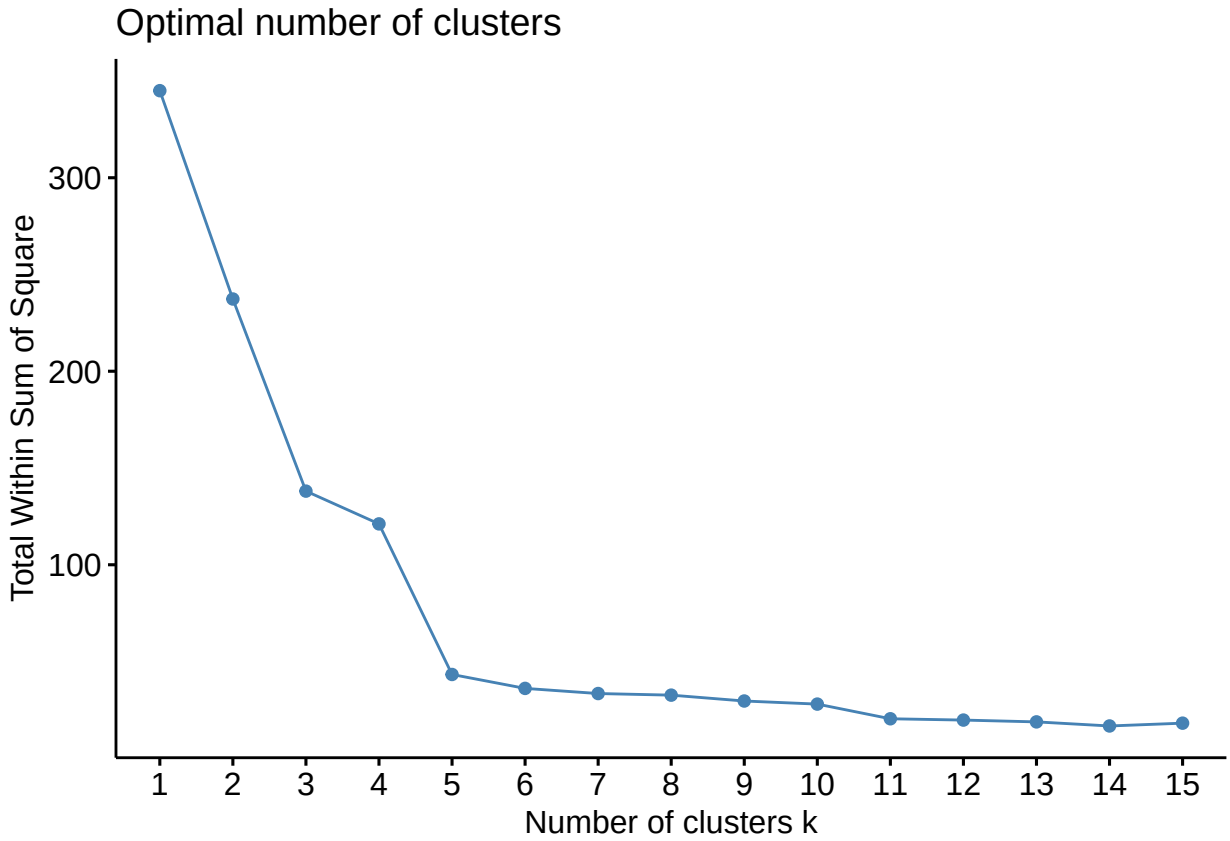
```
names=res[which((res[,4])<all_t[,1]),1]# names of those country
```

Task2

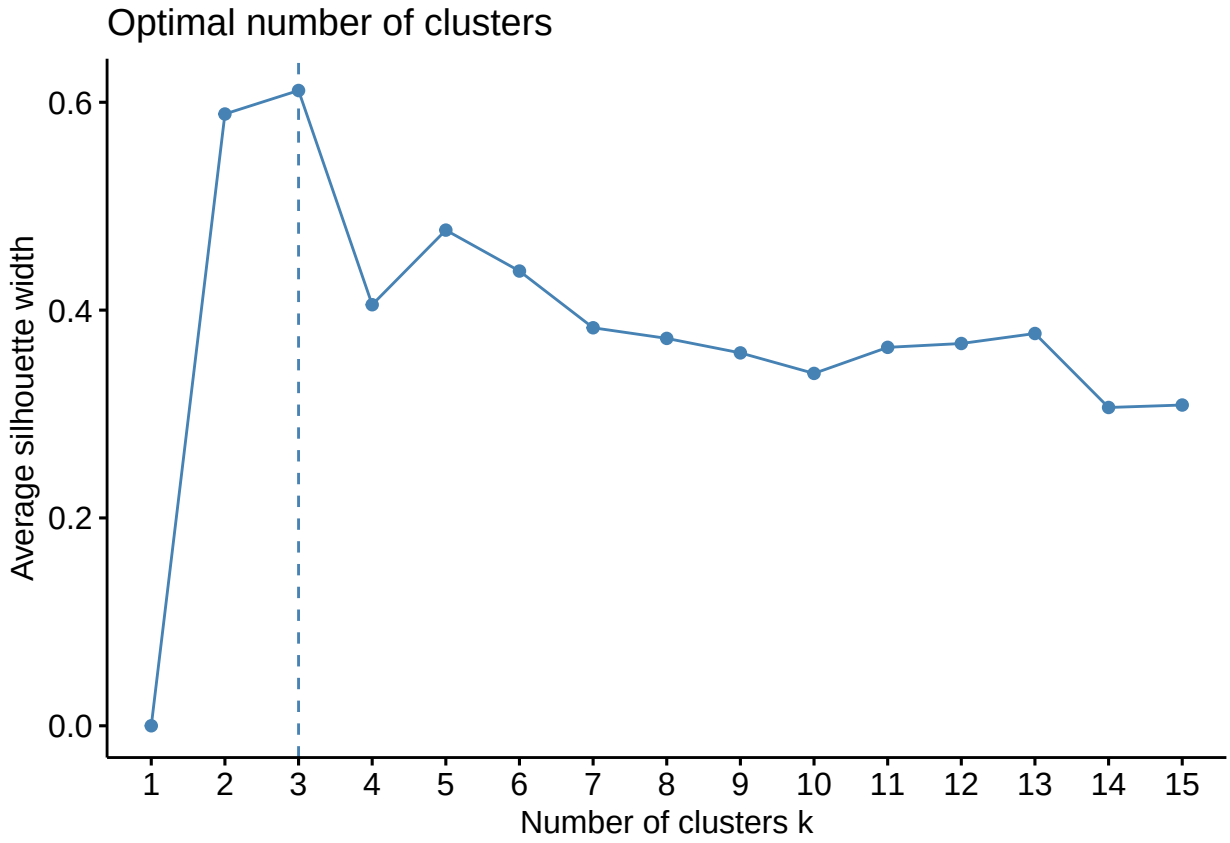
```
EM = function(data,ncluster){
  data = as.matrix(data) %>% scale()
  n = nrow(data)
  q = ncol(data)
  p_j = rep(1/ncluster,ncluster)
  mu = data[sample(n,ncluster),] %>% as.matrix()
  covmat = diag(ncol(data))
  covlist = list()
  for(i in 1:ncluster){
    covlist[[i]] = covmat
  }

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)
```

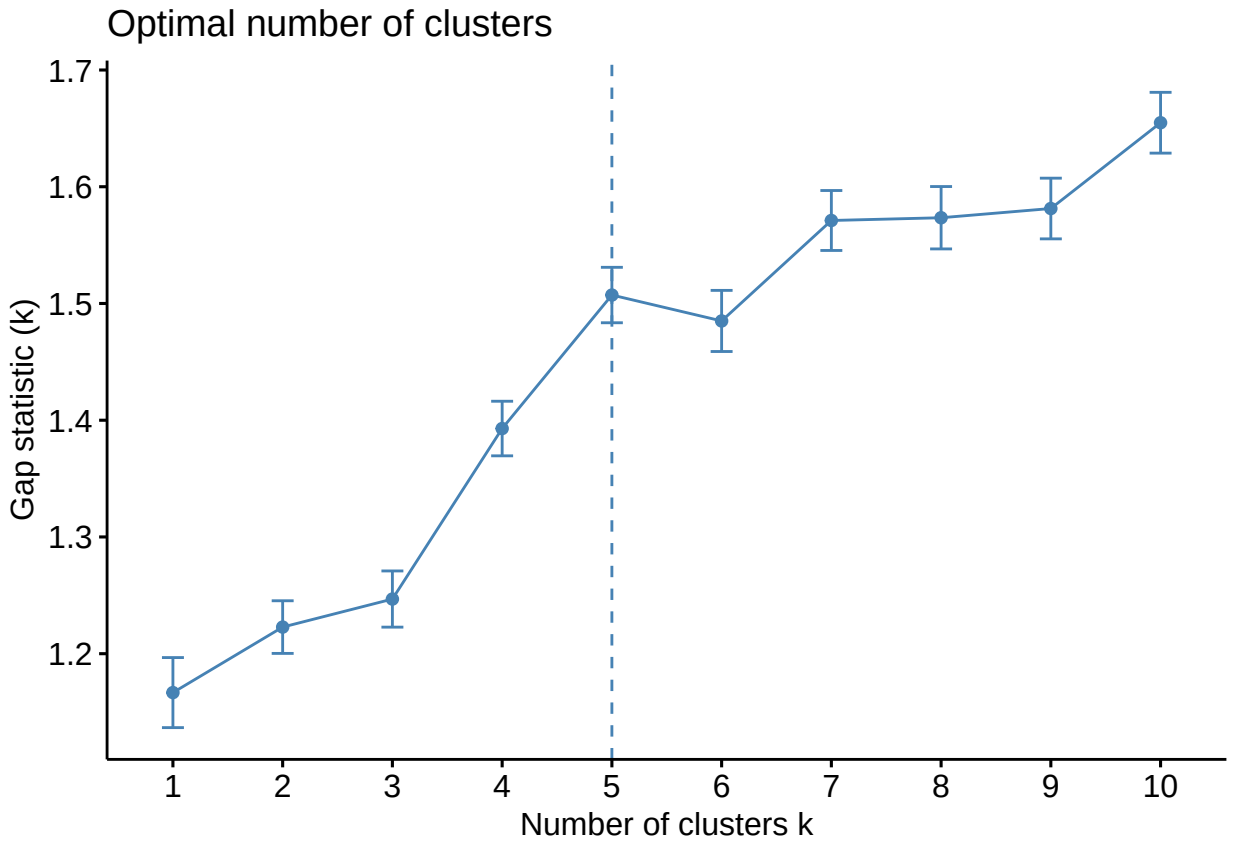



```
## use Gap Statistic Method
library(cluster)
```

```
##
## 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
  )

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

country	GMM_class	kmeans_class
Afghanistan	3	1
Albania	1	1
Algeria	3	1
Andorra	1	1
Argentina	3	1
Armenia	3	1
Australia	3	1
Austria	2	1
Azerbaijan	3	1
Bahrain	3	1
Bangladesh	3	1
Belarus	3	1
Belgium	2	1
Bolivia	1	1
Bosnia and Herzegovina	1	1
Brazil	3	1
Brunei	1	1
Bulgaria	3	1
Burkina Faso	1	1
Cambodia	3	1
Canada	3	1
Chile	3	1
China	2	3
Colombia	3	1
Congo (Kinshasa)	1	1
Costa Rica	1	1
Cote d'Ivoire	1	1
Croatia	3	1
Cuba	1	1
Cyprus	1	1
Denmark	3	1
Dominican Republic	3	1
Ecuador	3	1
Egypt	3	1
Estonia	3	1
Finland	3	1
France	2	3
Georgia	3	1
Germany	2	3
Ghana	1	1
Greece	3	1
Guatemala	1	1
Honduras	1	1
Hungary	1	1
Iceland	3	1
India	3	1
Indonesia	3	1
Iran	2	3
Iraq	3	1
Ireland	3	1

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

	country	GMM_class	kmeans_class
51	Israel	3	1
52	Italy	2	2
53	Jamaica	1	1
54	Japan	3	1
55	Jordan	1	1
56	Kazakhstan	1	1
57	Kenya	1	1
58	Korea, South	2	1
59	Kuwait	3	1
60	Kyrgyzstan	1	1
61	Latvia	1	1
62	Lebanon	3	1
63	Liechtenstein	1	1
64	Lithuania	1	1
65	Luxembourg	3	1
66	Malaysia	3	1
67	Malta	1	1
68	Martinique	1	1
69	Mauritius	1	1
70	Mexico	3	1
71	Moldova	1	1
72	Monaco	3	1
73	Montenegro	1	1
74	Morocco	1	1
75	Netherlands	2	1
76	New Zealand	1	1
77	Nigeria	3	1
78	North Macedonia	3	1
79	Norway	2	1
80	Oman	3	1
81	Pakistan	3	1
82	Panama	3	1
83	Paraguay	3	1
84	Peru	3	1
85	Philippines	3	1
86	Poland	3	1
87	Portugal	3	1
88	Qatar	3	1
89	Romania	3	1
90	Russia	3	1
91	Rwanda	1	1
92	San Marino	1	1
93	Saudi Arabia	3	1
94	Senegal	3	1
95	Serbia	3	1
96	Singapore	3	1
97	Slovakia	1	1
98	Slovenia	3	1
99	South Africa	3	1
100	Spain	2	3

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

	country	GMM_class	kmeans_class
101	Sri Lanka	3	1
102	Sweden	3	1
103	Switzerland	2	1
104	Taiwan*	3	1
105	Thailand	3	1
106	Trinidad and Tobago	1	1
107	Tunisia	1	1
108	Turkey	3	1
109	Ukraine	1	1
110	United Arab Emirates	3	1
111	United Kingdom	2	1
112	Uruguay	1	1
113	US	2	2
114	Uzbekistan	1	1
115	Venezuela	1	1
116	Vietnam	3	1

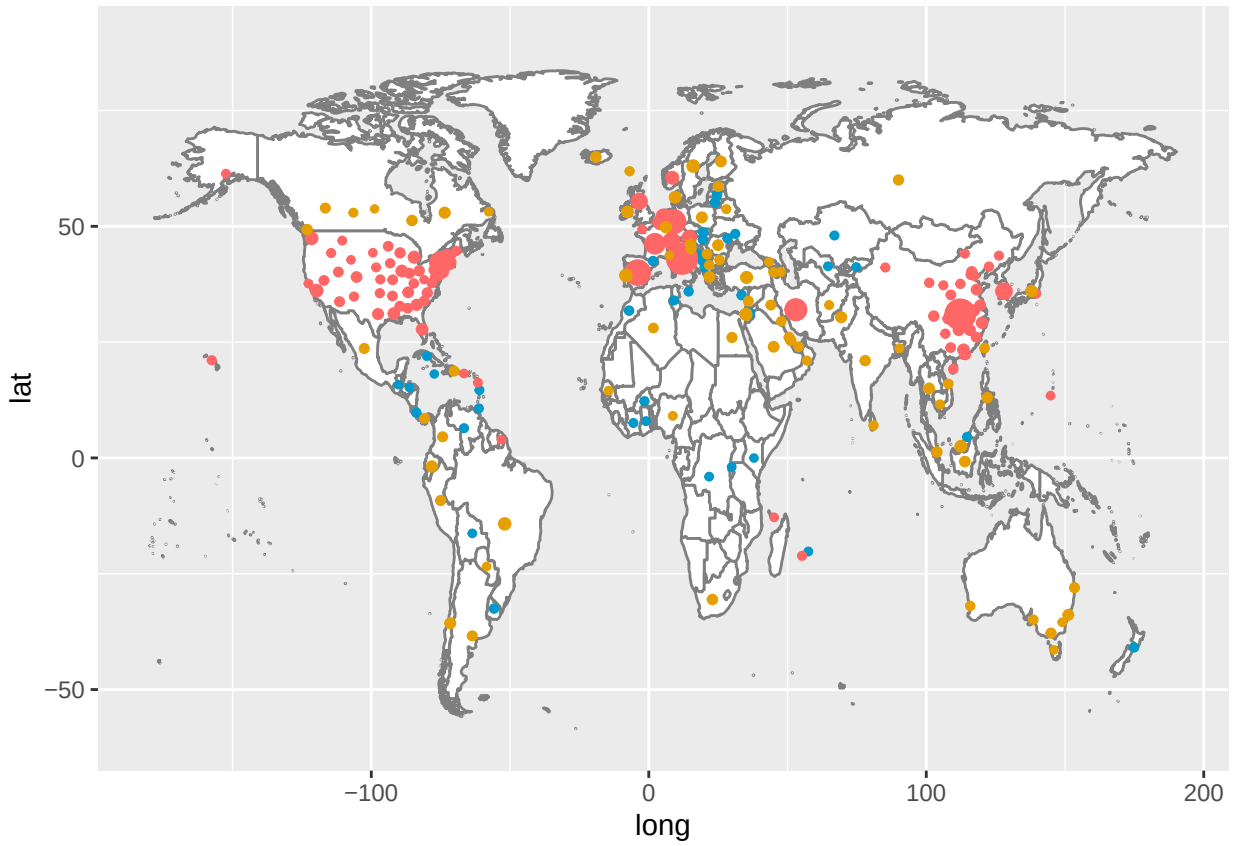
```
kmean_mean = clusters$centers %>%
  as.data.frame()

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.
```

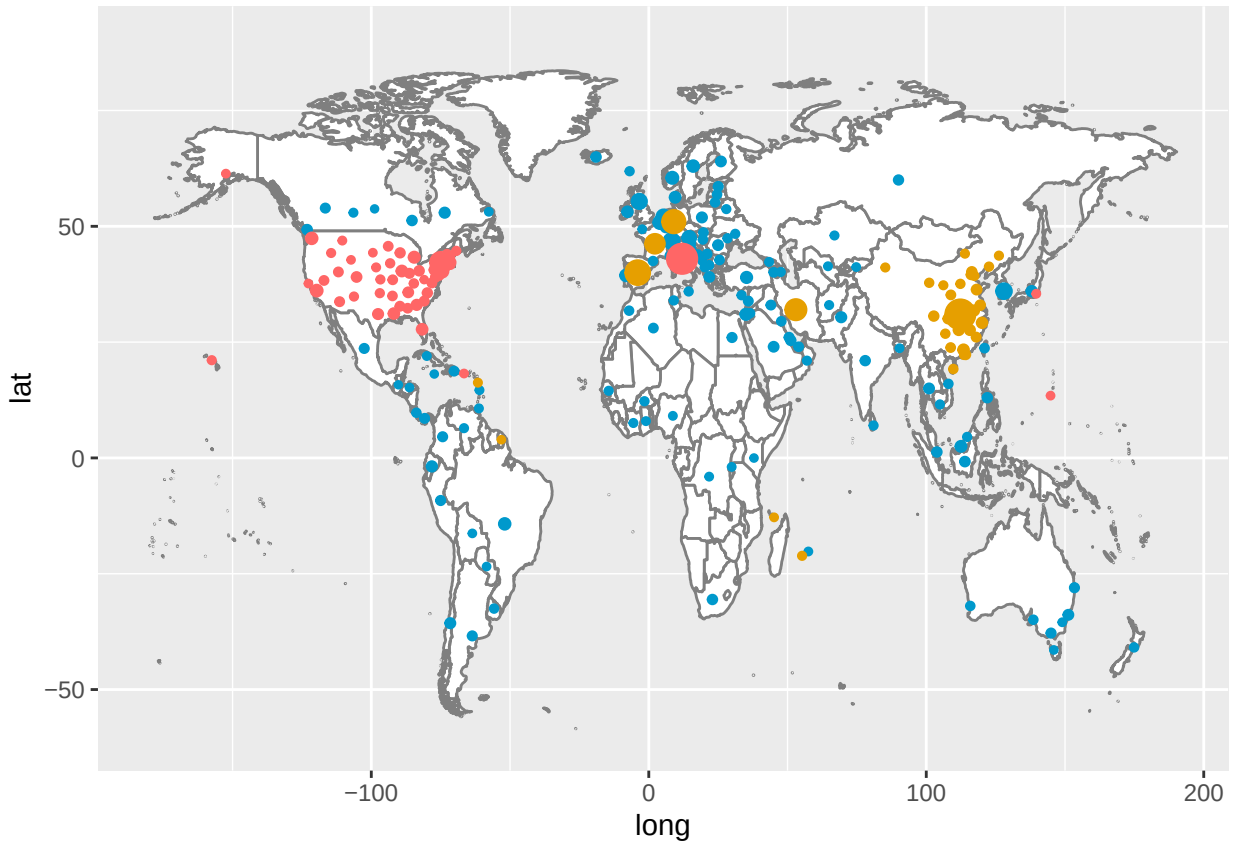
```
## blue class1   red class2  yellow class3
mp2
```



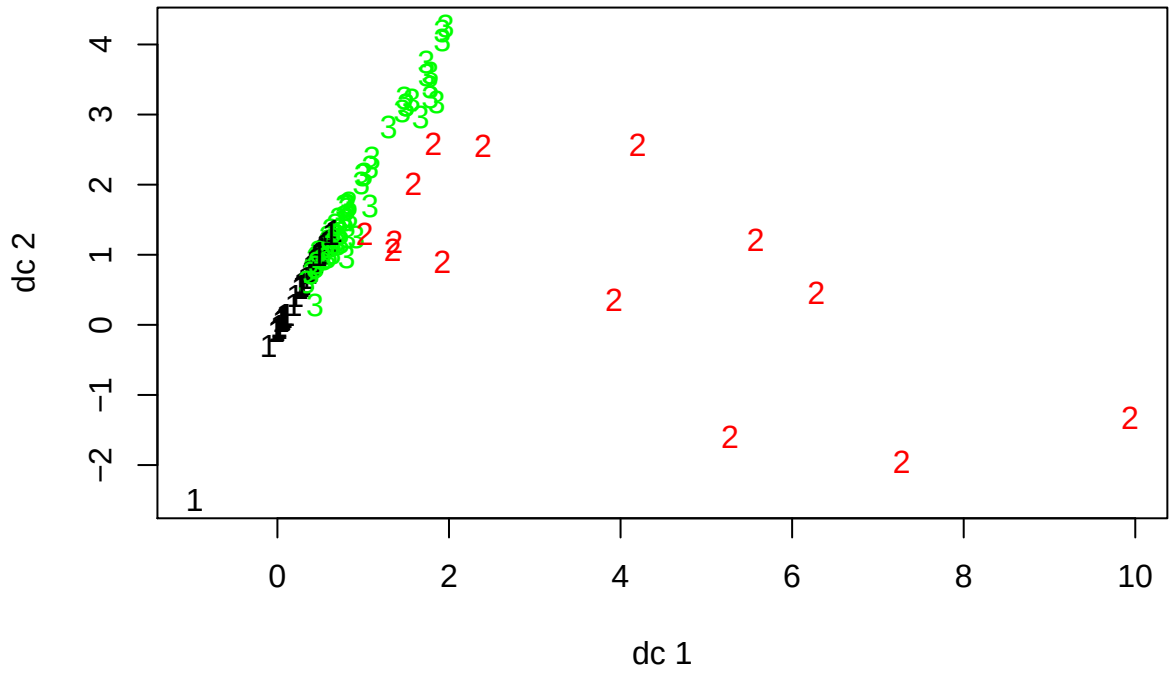

```
df_map_5 = df_map %>% filter(kmeans_class ==1)
df_map_6 = df_map %>% filter(kmeans_class ==2)
df_map_7 = df_map %>% filter(kmeans_class ==3)

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.
```

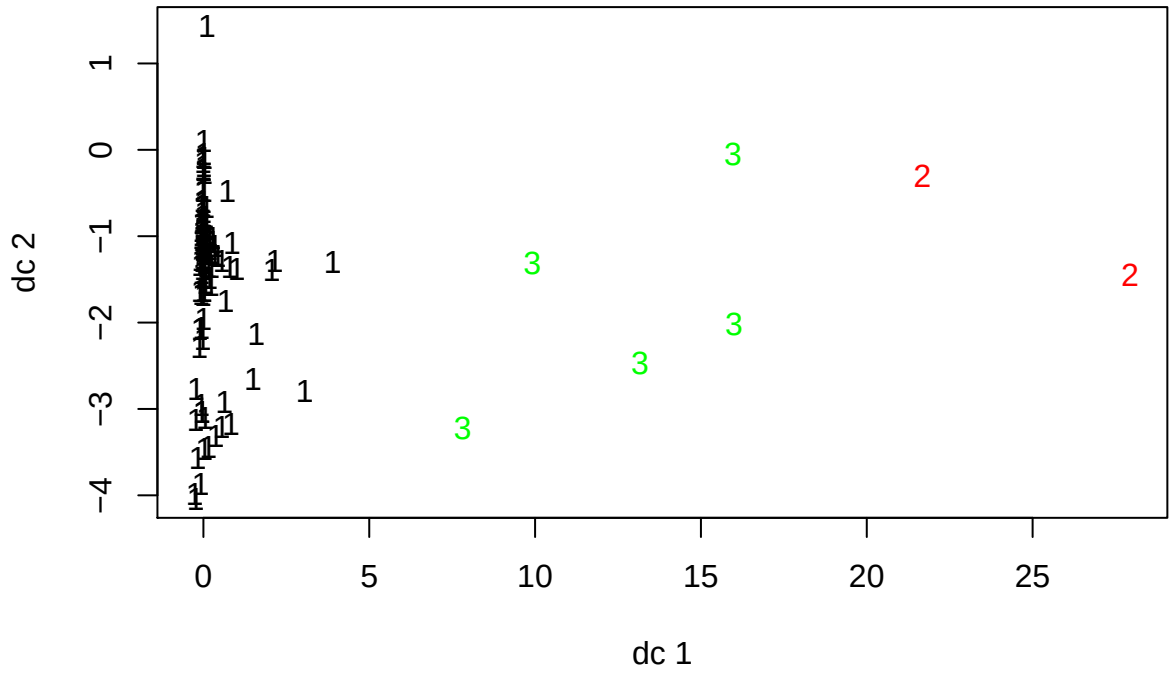
```
mp3 #
```




```
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