Universidad del Desarrollo
Facultad de Ingeniería

Examen

Taller de Inteligencia de Negocios

Docente:

Mauricio Herrera

Integrantes:

Jan Frese

Jorge Ramírez

Marcelo Cáceres

Problema 1

Para el desarrollo de este problema ocuparemos los dataset:

- “movilidad_todas_regiones.csv”
- “IDS_IndiceDeMovilidad.csv”
- “movilidad_covid_region_metropolitna.csv”

Se limpian todas las variables y eliminan gráficos existentes para comenzar a trabajar.

```
# Problema 1
#########################################################################
rm(list=ls()) # Para limpiar todas las variables
graphics.off() # Borra todos los gráficos
setwd("C:/Users/pc/Desktop/bases de datos csv y excel/")
datos=read.csv('movilidad_todas_regiones.csv',header=T,dec='.',sep=',')
head(datos)
```

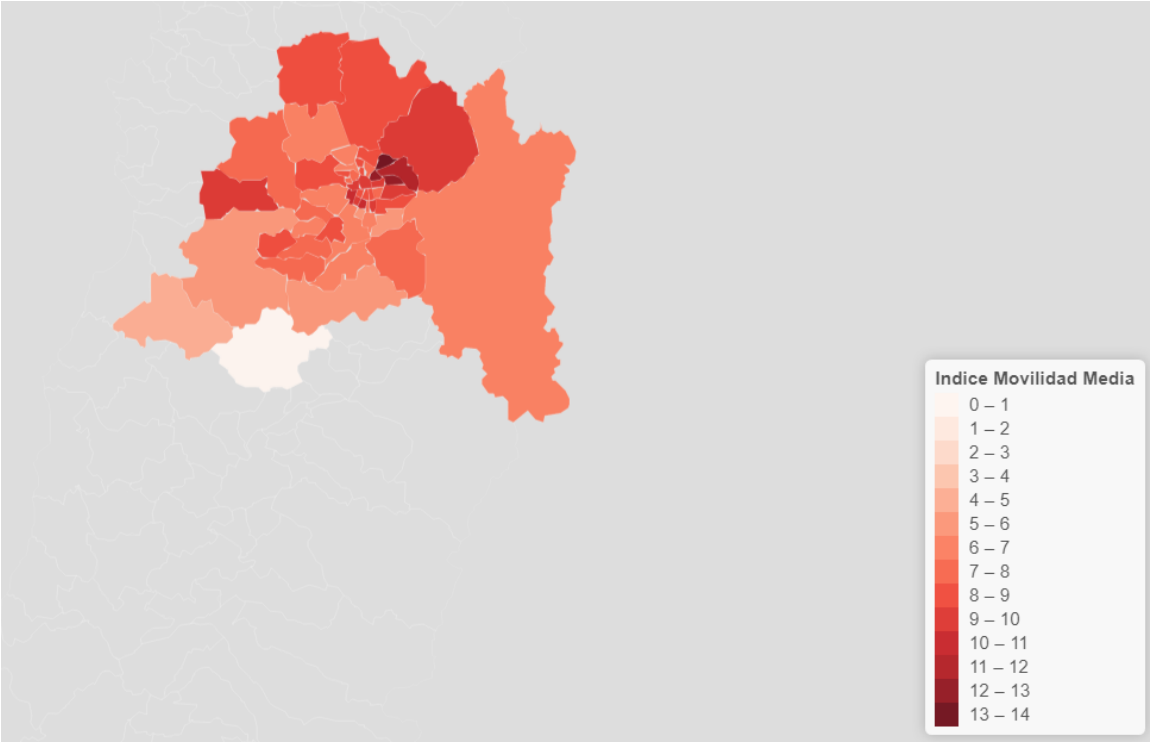
Pregunta 1:

Se solicita hacer un mapa con las comunas de la región metropolitana con los índices medios de movilidad dados por la variable “AVind” contenidos en el dataset “movilidad_todas_regiones.csv”. Para construir este mapa se utilizó como referencia el código entregado por el profesor con cambios en los datos seleccionados y en los intervalos seleccionados para notar con mayor claridad los datos.

```
#Pregunta 1

datos <- filter(datos,REGION=="RegiÃ³n Metropolitana de Santiago")
max(datos$AVind,na.rm=T)
min(datos$AVind,na.rm=T)
```

```
l=getwd()
l1=paste0(l,"/comunas")
Comunas <- readOGR(
  dsn= l1, # debe apuntar a donde está el archivo comunas.shp
  layer="comunas",
  verbose=FALSE
)
data=Comunas@data
glimpse(data)
dat=datos%>%dplyr::select(-c(Comuna,REGION,PROV))
P=left_join(data, dat, by = "CODIGO")
Comunas@data$Casos=P$Casos
Comunas@data$Poblacion=P$Poblacion
Comunas@data$AVind=P$AVind
mis_bins <- c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14) # la partición
pal <-colorNumeric("viridis",NULL,reverse = T)
mi_paleta <- colorBin( palette= "Reds", domain=Comunas@data$AVind,
                       na.color="transparent", bins=mis_bins)
# Preparación del texto que saldrá en los popout
mi_texto <- paste(
  "Número de Casos: ", Comunas@data$Casos, "<br/>",
  "Población: ", Comunas@data$Poblacion, "<br/>",
  "Comuna: ", Comunas@data$NOM_COM, "<br/>",
  "Provincia: ", Comunas@data$PROV, "<br/>",
  "Indice Movilidad Media: ", Comunas@data$AVind, "<br/>",
  "Región: ", Comunas@data$REGION,
  sep="") %>%
  lapply(htmltools::HTML)

# El mapa (Puede demorar hasta un par de minutos. Depende del computador)
leaflet(Comunas) %>%
```

```
# addTiles() %>% # Se puede descomentar si se quiere insertar el mapa en un mapa del mundo
setView(lng = -71.542969, lat = -35.675147, zoom = 3)%>%
addPolygons(
  fillColor = ~mi_paleta(Comunas@data$AVind),
  stroke=TRUE,
  fillOpacity = 0.9,
  color="white",
  weight=0.3,
  label = mi_texto,
  labelOptions = labelOptions(
    style = list("font-weight" = "normal", padding = "3px 8px"),
    textsize = "13px",
    direction = "auto"
  )
) %>%
addLegend( pal=mi_paleta,
          values=~Casos, opacity=0.9,
          title = "Indice Movilidad Media", position = "bottomright" )
```



Del gráfico, podemos observar que las comunas del centro de la región metropolitana tienen un mayor índice de movilidad media en comparación a las más alejadas del centro.

La comuna de Vitacura presenta el mayor índice de movilidad con un valor de 13.08.

Pregunta 2:

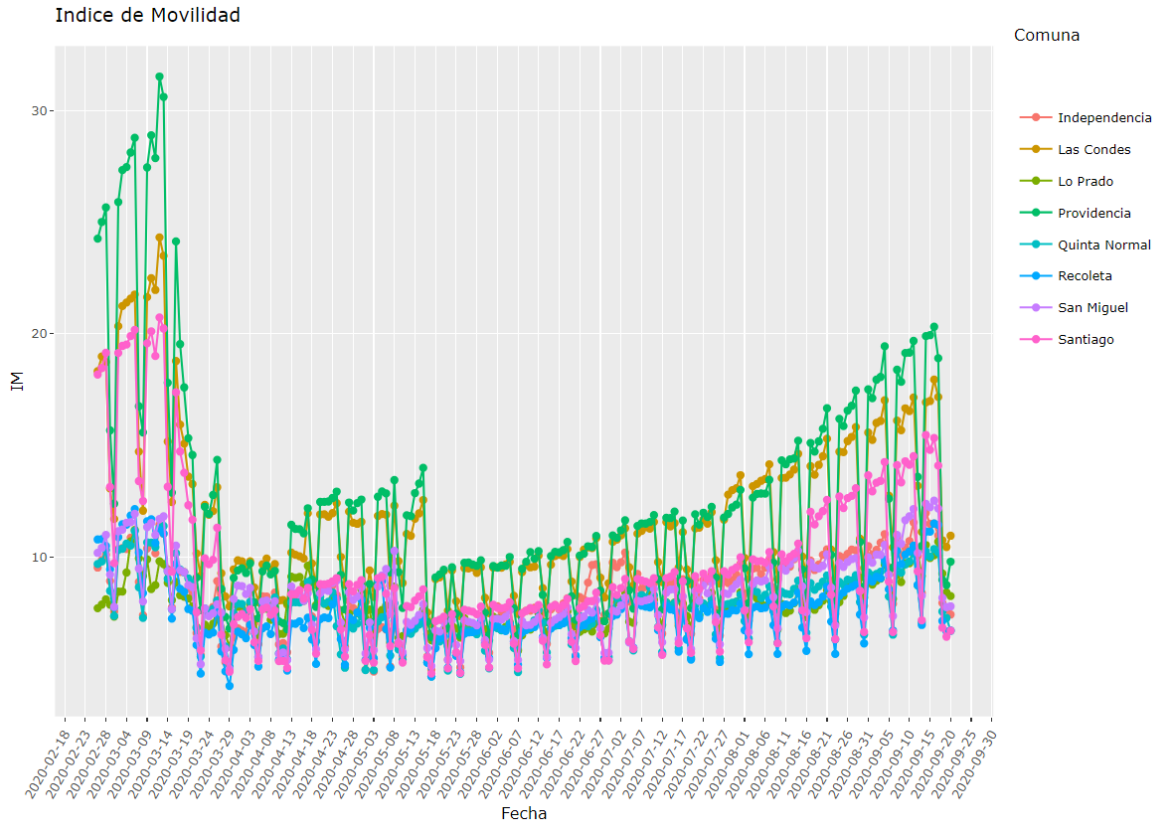
Se solicita estudiar los cambios de movilidad durante la pandemia mediante gráficos. Considerando los efectos de la cuarentena para dos conjuntos “q1” y “q2” con diferentes comunas de la región metropolitana.

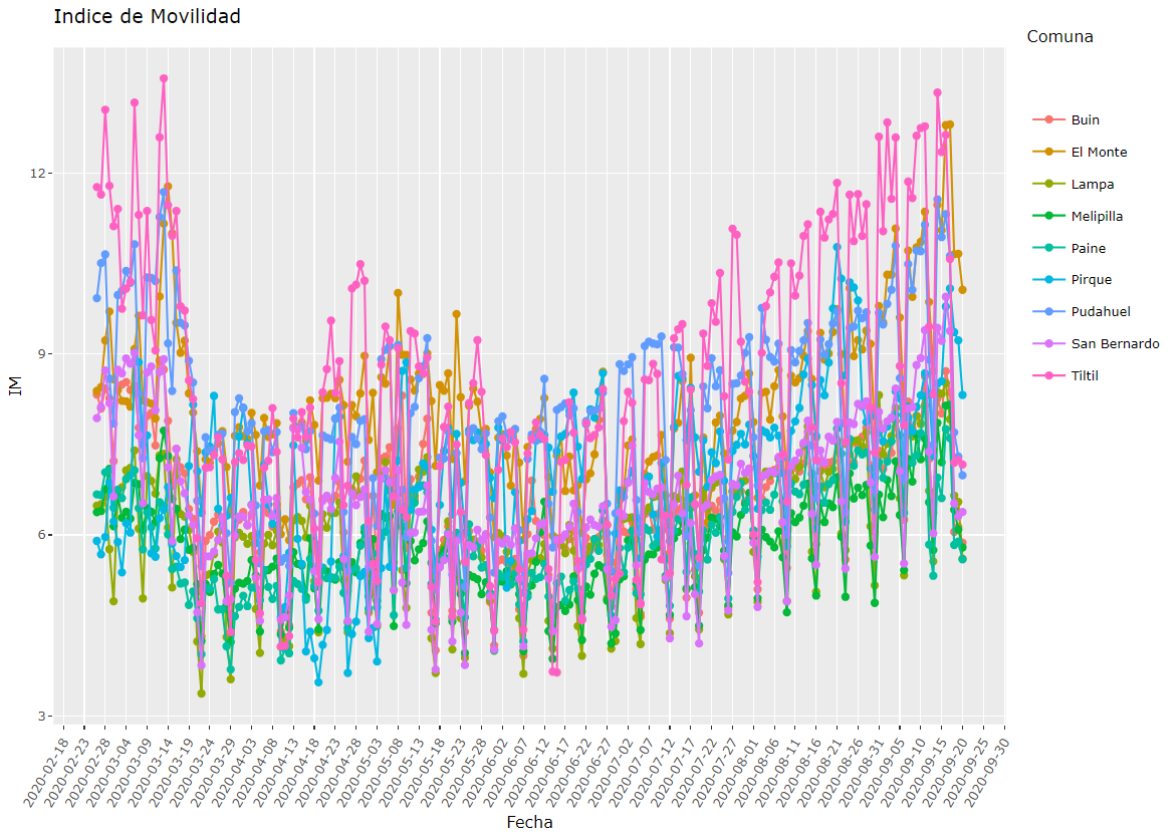
Definimos q1 y q2 como:

q1 = Las Condes, Independencia, Estación Central, Lo Prado, Ñuñoa, San Miguel, Providencia, Santiago, Quinta Normal, Recoleta.

q2 = Melipilla, Tiltil, Lampa, Buin, Pirque, San Bernardo, Paine, Peñaflor, Pudahuel, María Pinto, El Monte.

```
#Pregunta 2

datosM=read.csv("IDS_IndiceDeMovilidad.csv", sep=',',dec='.',header=T)
datosM=datosM[-c(71553,71554)] # Eliminamos estas comunas que estan sin nombres

datosM$Fecha=ymd(datosM$Fecha)
datosM

q1=c('Las Condes', 'Independencia','Estación Central', 'Lo Prado','Ñuñoa',
    'San Miguel', 'Providencia','Santiago','Quinta Normal', 'Recoleta')

q2=c('Melipilla','Tiltil', 'Lampa','Buin', 'Pirque', 'San Bernardo', 'Paine',
    'Peñaflor', 'Pudahuel', 'María Pinto','El Monte')

g1 <- ggplot(data=filter(datosM,Comuna%in%q1),
           aes(x=Fecha,y= IM ,color=Comuna))+
  ggtitle('Indice de Movilidad')+
  geom_line()+
  geom_point()+
  scale_x_date(date_breaks = "5 day")+
  theme(axis.text.x=element_text(angle=60, hjust=1))
plot(g1)

g2 <- ggplot(data=filter(datosM,Comuna%in%q2),
           aes(x=Fecha,y= IM ,color=Comuna))+
  ggtitle('Indice de Movilidad')+
  geom_line()+
  geom_point()+
  scale_x_date(date_breaks = "5 day")+
  theme(axis.text.x=element_text(angle=60, hjust=1))
plot(g2)
```


Indice de Movilidad
Comuna
Independencia
Las Condes
Lo Prado
Providencia
Quinta Normal
Recoleta
San Miguel
Santiago
IM
30
20
10
Fecha


Indice de Movilidad
Comuna
Buin
El Monte
Lampa
Melipilla
Paine
Pirque
Pudahuel
San Bernardo
Tiltil
IM
12
9
6
3
Fecha

Para ambos gráficos, podemos notar una disminución en los índices de movilidad a partir de la segunda semana de marzo, lo que está directamente relacionado con el inicio de las primeras cuarentenas en la región metropolitana.

Además, podemos notar que las comunas de q2 volvieron casi a la normalidad según los índices mientras que las comunas de q1 siguen aumentando, pero aún no se llega a recuperar la normalidad que se tenía en sus índices de movilidad de antes que se decretara la cuarentena.

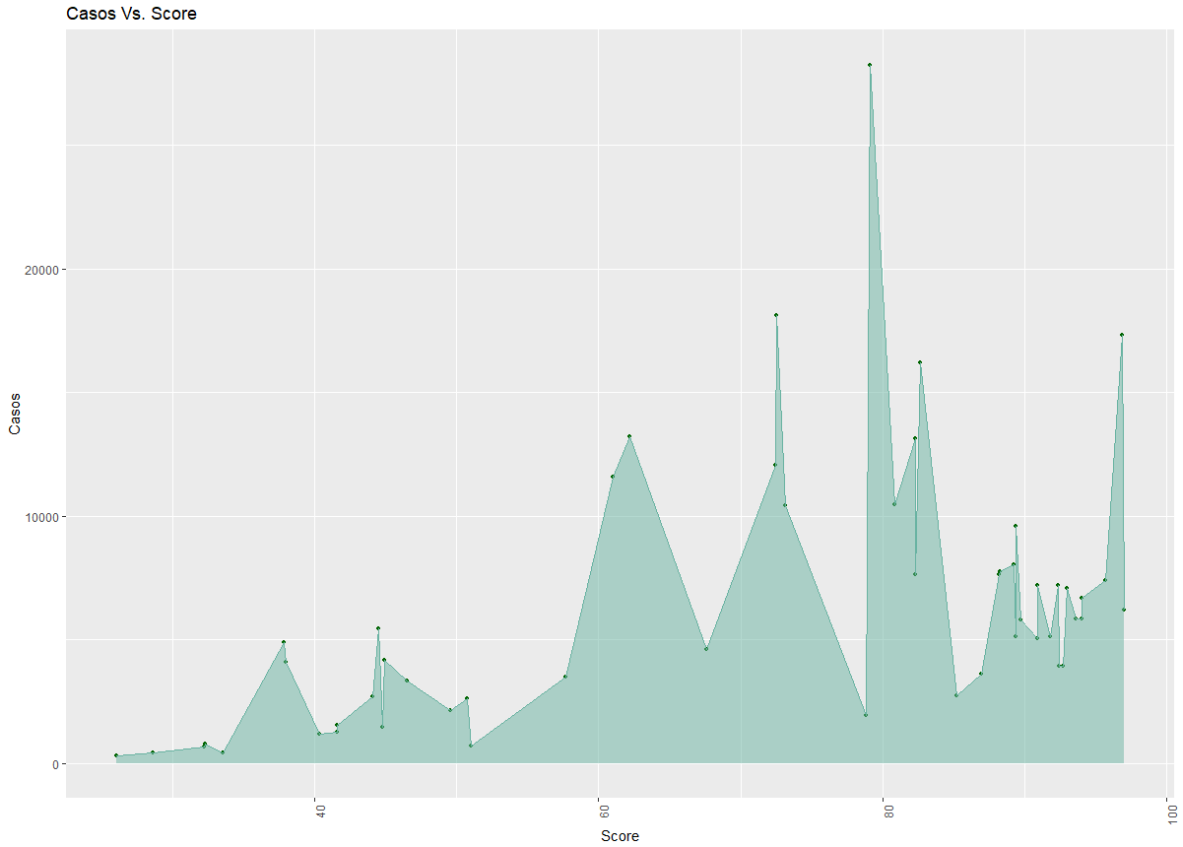
Pregunta 3:

Utilizando la base de datos “movilidad_covid_region_metropolitna.csv” se pide lo siguiente:

```
# Pregunta 3


datosA=read.csv("movilidad_covid_region_metropolitna.csv", sep=',',dec='.',header=T)

datosA
```

(A) Se solicita graficar la relación que existe entre la variable “score” y el número de casos totales en las comunas de la región metropolitana

```
p <- data.frame(datosA) %>%
  ggplot( aes(x=score  , y= Casos))+
  ylab("Casos")+xlab("Score")+
  ggtitle('Casos Vs. Score')+
  theme(axis.text.x=element_text(angle=90, hjust=1),legend.position ='none')+
  geom_point(size=1,color='Darkgreen') + geom_area(fill="#69b3a2", alpha=0.5) +
  geom_line(color="#69b3a2")
plot(p)
```



De acuerdo al gráfico presentado, no podemos afirmar de manera significativa que existe una relación directa entre estas dos variables, ya que la variación presentada en el gráfico es bastante irregular y variable.

(B) Se solicita crear un modelo de regresión lineal utilizando las variables “score” como variable independiente y la variable casos como la variable dependiente.

```
regresion <- lm(Casos ~ score, data = datosA)
summary(regresion)

plot(x = datosA$score, y = datosA$Casos,
    xlab='score',ylab='Casos', main = "Casos vs Score", pch = 20, col = "grey30")
abline(regresion, col = "red")
```

Summary:

Call:

lm(formula = Casos ~ score, data = datosA)

Residuals:

Min	1Q	Median	3Q	Max
-5415	-2396	-1566	1125	20895

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-1112.44	2122.57	-0.524	0.602524	
score	107.19	29.07	3.687	0.000559	***

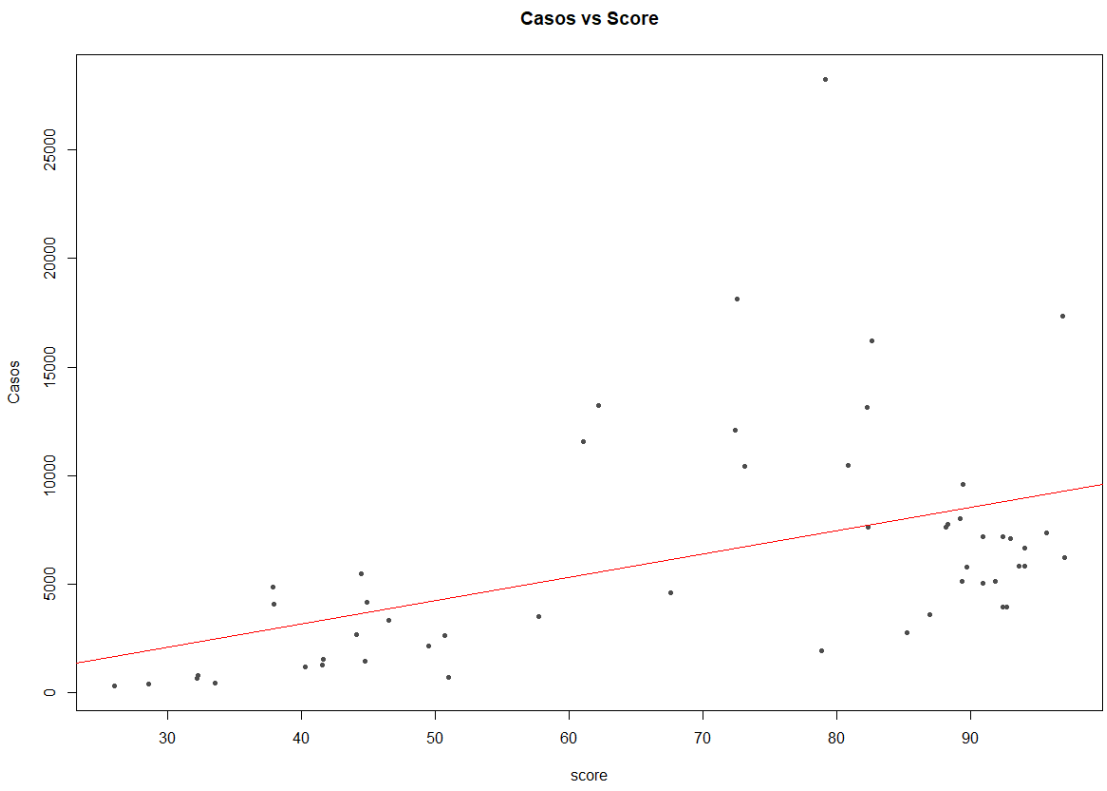
---

*Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1*

Residual standard error: 4829 on 50 degrees of freedom

Multiple R-squared: 0.2138, Adjusted R-squared: 0.1981

F-statistic: 13.6 on 1 and 50 DF, p-value: 0.0005585



Según lo obtenido, Podemos concluir que score es una variable significativa para explicar el modelo dado que su p-valor es cercano al cero (0.0005585), además, como el R-cuadrado y el R-cuadrado ajustado son cercanos a 0.2, se puede concluir que la recta ajustada contempla el 20% de la varianza de los datos, lo cual hace referencia que el modelo de regresión lineal creado es deficiente para predecir los casos de contagios de acuerdo con el puntaje “score”.

En resumen, no se puede afirmar que la movilidad definida por “score” explica todo el proceso de contagio, debido a que la recta regresión ajustada no contempla la mayoría de las observaciones.

Pregunta 4:

Se procede a trabajar con los datos “movilidad_covid_region_metropolitna.csv” y se seleccionan las variables “tasa” y “score”. Después, se determina la distancia que hay entre las observaciones según el método euclidiano. Finalmente, se define la composición de los clústeres según el método “Ward”.

```
#Pregunta 4

datosA=read.csv("movilidad_covid_region_metropolitna.csv", sep=',',dec='.',header=T)

datosA

rownames(datosA) <- datosA$Comuna

datosA$Comuna <- NULL

datosA1 <- datosA[,-(1:3)]

datosA2 <- datosA1[,-2]

datosA <- scale(datosA2)


d=dist(datosA,method="euclidean",p=2)

ward  = hclust(d,method="ward.D")
```

(A) Se procede a construir la matriz “cophenetic” de Ward y se evalúa la correlación de esta con la distancia. Dado esto nos podemos dar cuenta que la correlación de la matriz “cophenetic” con la distancia es de 0.678 aprox. Esto quiere decir que es una correlación alta.

```
W=cophenetic(ward)

W

cor(W,d)
```

(B) Se distribuyen los clústeres de Ward, según 5 tipos de particiones. En el primer clúster se observan 15 observaciones, en el segundo 12 observaciones, en el tercero 10 observaciones, en el cuarto 9 observaciones y en el quinto 5 observaciones.

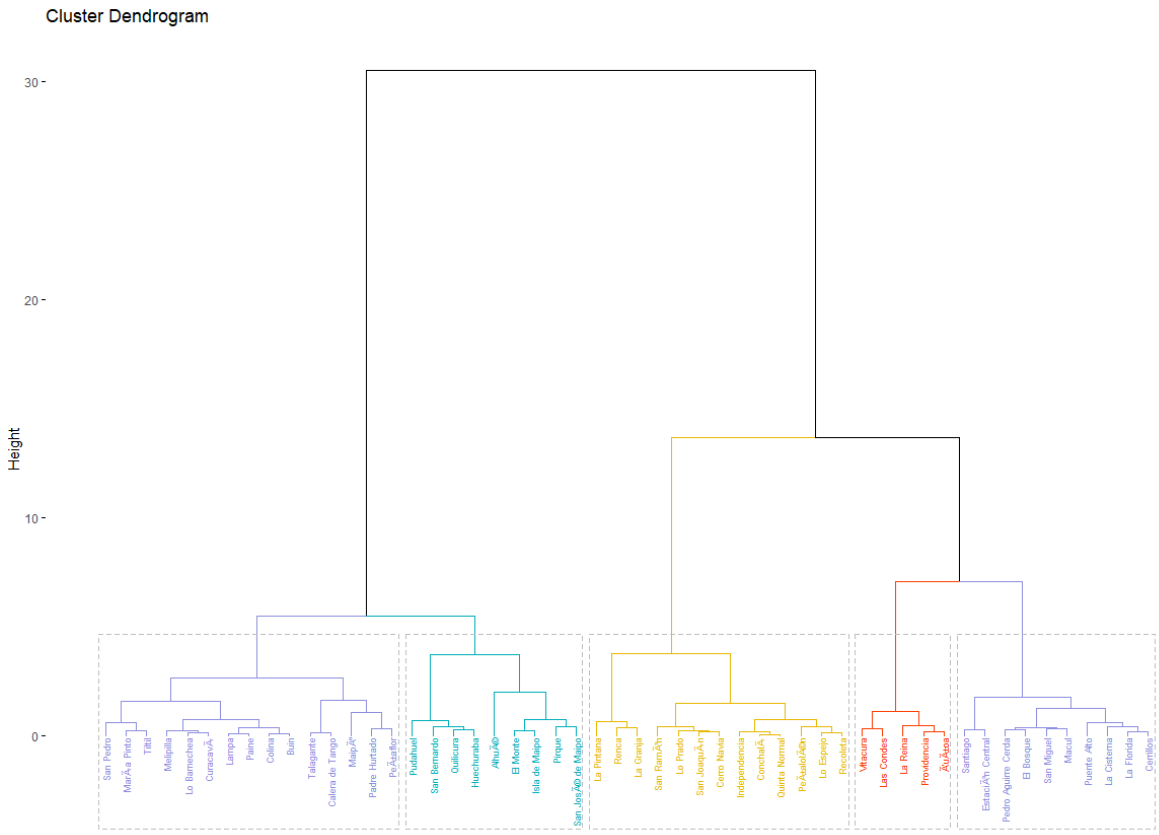
```
group5=cutree(ward, k = 5)

table(group5)


#Visualizacion del corte en 5 clusters


fviz_dend(ward, k = 5, # Numero del corte

      cex = 0.5, # el tamaño de la etiqueta
```

k_colors = c("#8E8FDF", "#00AFBB", "#E7B800", "#FC3E01"),

color_labels_by_k = TRUE, # colorear por valor de la etiqueta

rect = TRUE # Adiciona rectangulos

)



Visualización de clúster.

(C) En esta pregunta se pide mostrar los valores medios de la “Tasa” y “score” en cada clúster.

Para realizar este inciso, se procede a utilizar el código de referencia designado y se procedió a adaptarlo al código. Luego se presenta cada valor medio mediante una tabla.

grupos.5 = cutree(ward,5)

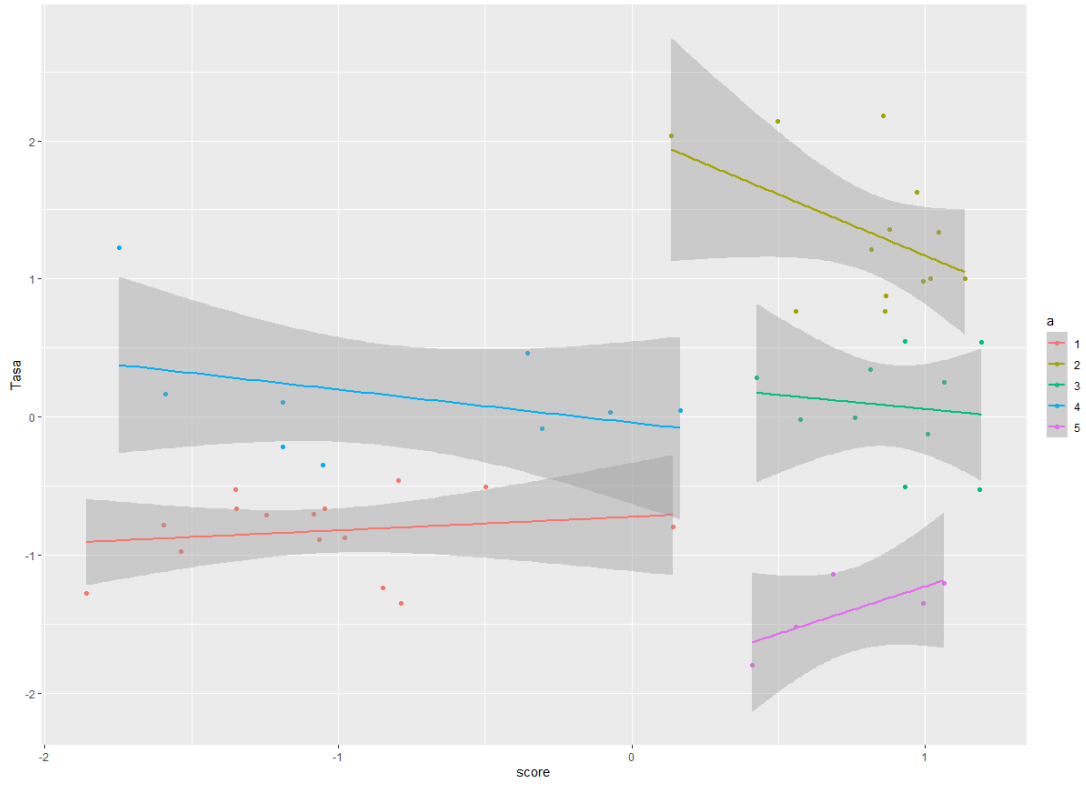
aggregate(datosA,list(grupos.5),mean)

sapply(unique(grupos.5),function(g)row.names(datosA)[grupos.5 == g])

	Group.1	Tasa	score
1	1	-0.82881335	-1.0590893
2	2	1.32908127	0.8179009
3	3	0.07868397	0.8876655
4	4	0.15415326	-0.8151539
5	5	-1.40401505	0.7426716

(D) Para esta pregunta se agregó la etiqueta de membresía que el análisis de clúster otorgó a cada comuna y se creó un gráfico de dispersión.

```
df <- as.data.frame(datosA)
a=as.factor(as.vector(grupos.5))
df$a <- a
ggplot(data=df,aes(x=score,y=Tasa,color=a))+
 geom_point()+geom_smooth(method="lm")
```



Aquí se puede visualizar la regresión lineal de cada uno de los 5 clúster.

1er clúster tiene tendencia ascendente.

2do clúster tiene tendencia descendente.

3er clúster tiene tendencia descendente.

4to clúster tiene tendencia descendente.

5to clúster tiene tendencia ascendente.

Por lo que se puede inferir que en algunos clústeres la movilidad se relaciona de manera descendente con el contagio y en otra de manera ascendente. Por lo que podemos ver que tan segura es la movilidad con respecto a los contagios según cada clúster.

Problema 2

Para el desarrollo de este problema se ocupará la base de datos “NCI60” correspondiente al paquete “ISLR”.

Pregunta 1:

Se solicita hacer un escalado de los datos de la siguiente manera:

```
data("NCI60", package = "ISLR")

# se separa la data
nci.labs=NCI60$labs
nci.data=NCI60$data

# 1) Antes de hacer el analisis escale (estandarice) los datos
nci.data <- scale(nci.data, center = TRUE, scale = TRUE)
head(nci.data)
```

Pregunta 2:

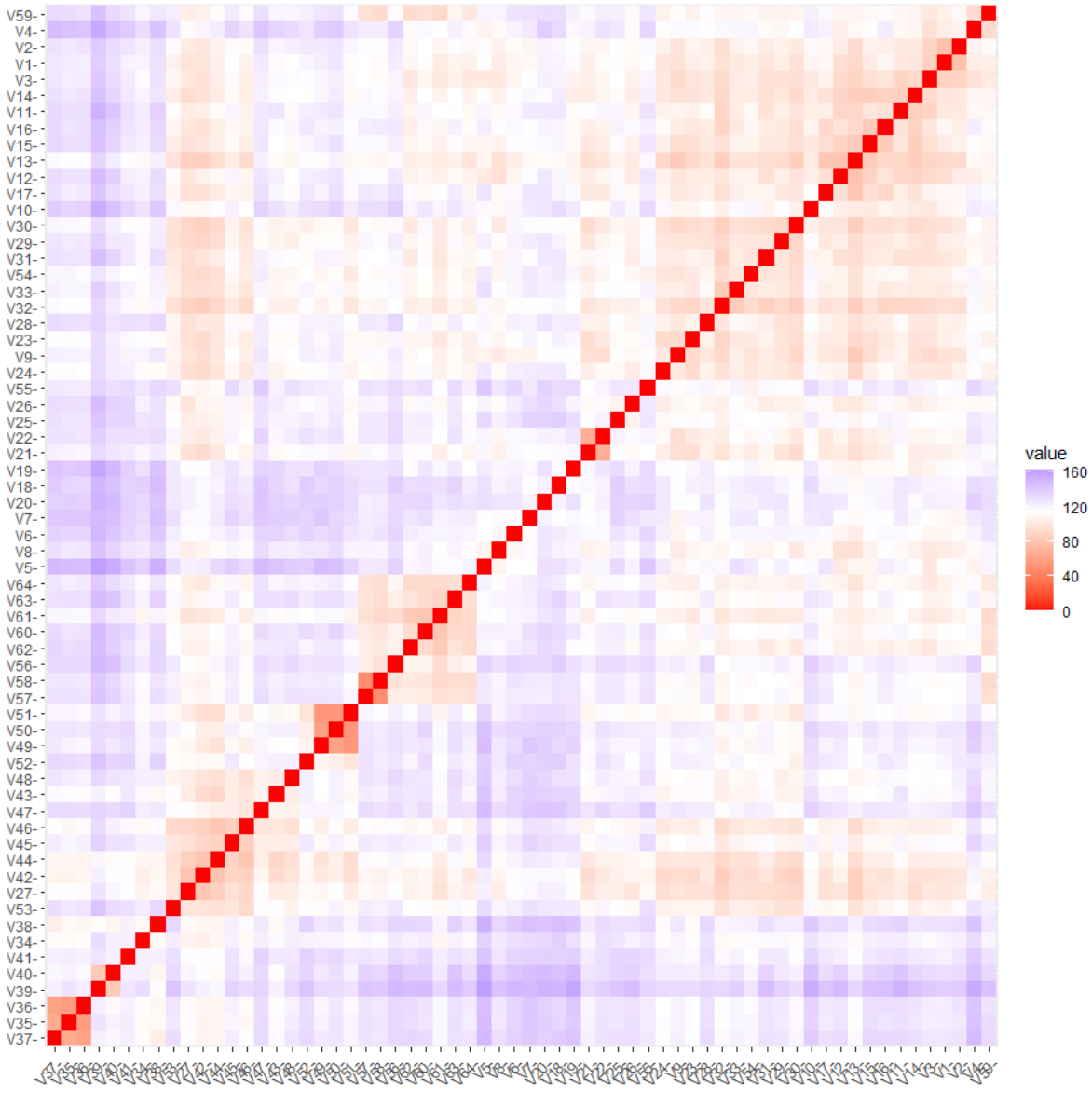
Se solicita determinar cuántas observaciones por tipos de cáncer hay en el dataset, para esto ocupamos una tabla en “nci.labs”, en donde se pueden observar los siguientes datos:

- Breast: 7
- CNS: 5
- Colon: 7
- K562A-repro: 1
- K562B-repro:1
- Leucemia: 6
- MCF7A-repro: 1
- MCF7D-repro: 1
- Melanoma: 8
- NSCLC: 9
- Ovarian: 6
- Prostate: 2
- Renal: 9
- Unknown: 1

Pregunta 3:

Se solicita construir una matriz de distancia entre observaciones empleando la distancia euclidiana con los datos estandarizados, por lo que se realizó de la siguiente manera:

```
# 3) Construya la matriz de distancia entre observaciones empleando,
#     la distancia euclidiana con los datos estandarizados.
d = dist(nci.data,method="euclidean")
fviz_dist(d)
```



Como se puede apreciar en la matriz, los datos no poseen una correlación muy fuerte entre sí dado que los colores son tenues, no obstante, existen algunas correlaciones fuertes, pero son difícil de diferenciar.

Pregunta 4:

Para utilizar el método jerárquico con distintos tipos de “linkage” y luego determinar la correlación, se ocupa el código empleado en las instrucciones del trabajo.

```
# 4) Use el metodo jerarquico con distintos tipos de "linkage" y determine la correlacion
#   entre ellos. Comente. Sugerencia: Use el siguiente codigo:
dend1=nci.data %>% dist %>% hclust("complete") %>% as.dendrogram
dend2=nci.data %>% dist %>% hclust("single") %>% as.dendrogram
dend3=nci.data %>% dist %>% hclust("average") %>% as.dendrogram
dend4=nci.data %>% dist %>% hclust("centroid") %>% as.dendrogram
dend_list <- dendlist("Complete" = dend1, "Single" = dend2,
                      "Average" = dend3, "Centroid" = dend4)
# Calcule la matriz de correlacion usando:
cors <- cor.dendlist(dend_list)
# Muestre la matriz mediante:
round(cors, 2)
```

Obteniendo de esto la siguiente tabla con las correlaciones:

	Complete	**Single**	**Average**	**Centroid**
Complete	1	0.5	0.75	0.31
Single	0.5	1	0.85	0.68
Average	0.75	0.85	1	0.55
Centroid	0.31	0.68	0.55	1

Como se puede observar, lógicamente la mayor correlación entre los tipos de “Linkage” son cada uno con sí mismo, la cual presenta la diagonal de la tabla. Sin embargo, si no se llega a considerar esa relación, se puede observar que la correlación más alta que se tiene con los diversos “Linkage”, son el “Linkage Average” con el “Linkage Single” con una correlación del 85%. Por otra parte, se puede apreciar que la menor correlación que se posee, son los “Linkage Centroid”, junto con el “Linkage Complete”, representando una correlación de 31%.

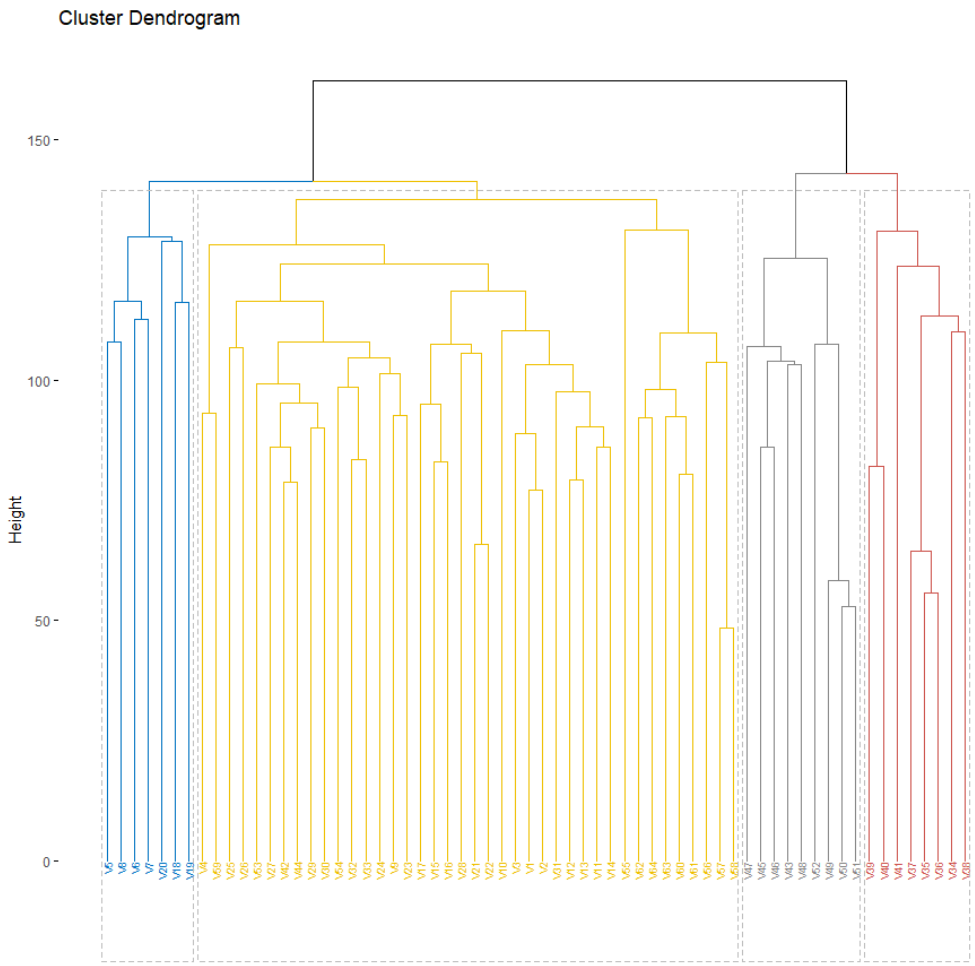
Pregunta 5:

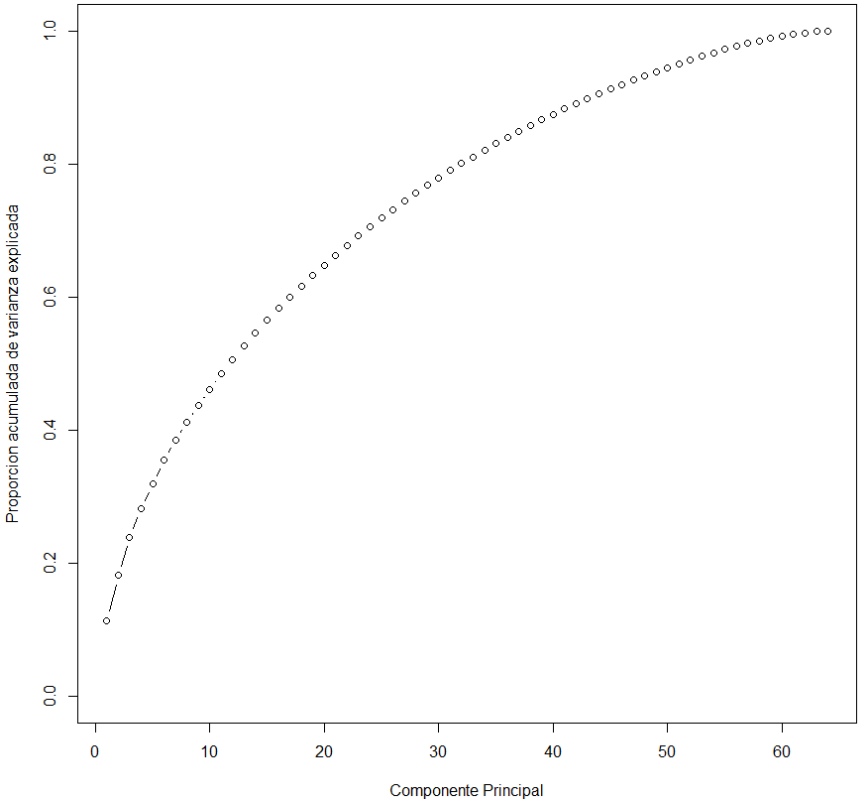
Se selecciona el método complete para formar los clústeres con 4 cortes, luego se ejecuta un análisis de componentes principales, obteniendo que los primeros dos componentes principales no son suficientes, necesitando de esta manera los primeros 25 componentes principales para recién poder explicar de buena manera la base de datos. Finalmente se genera un diagrama de dispersión para los dos

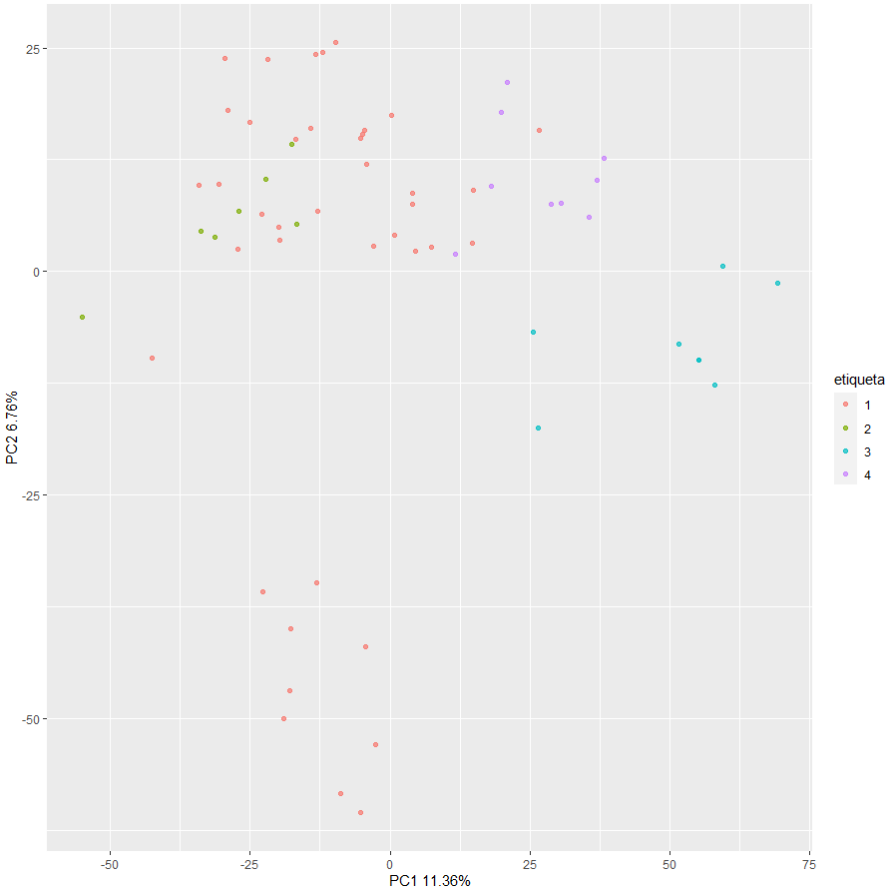
primeros componentes principales, pudiendo apreciar una homogeneidad en los clústeres, esto debido a que ambos componentes principales no son suficientes para explicar la base de datos.

```
# 5) Elija un metodo de “linkage” para continuar el analisis. Construya y visualice graficamente
# el dendrograma. Identifique un numero apropiado de clusters (por estimacion y/o de
# acuerdo al problema tratado) y muestrelos en el dendrograma. Visualice ademas los
# cluster en un diagrama 2D usando componentes principales.

# Dendograma
par(mfrow = c(1,1))
c = hclust(d,method="complete")
fviz_dend(c, k = 4, # Número del corte
        cex = 0.5, # el tamaño de la etiqueta
        k_colors = "jco",
        color_labels_by_k = TRUE, # color por valor de la etiqueta
        rect = TRUE # Adiciona retángulos
)

# componentes principales

# PCA
out=prcomp(nci.data,scale=T)
y=out$x
ym   = y[,1:2] # primeros dos componentes principales

pr.var=out$sdev^2
pr.var
pve=pr.var/sum(pr.var)
pve
plot(cumsum(pve), xlab="Componente Principal", ylab="Proporcion acumulada de varianza
explicada", ylim=c(0,1),type='b')
```

```
grupos.4 <- cutree(c,4)
table(grupos.4)
a1=round(out$sdev^2/sum(out$sdev^2)*100,2)
show(a1)
df=data.frame(PC1=ym[,1],PC2=ym[,2],etiqueta=as.factor(grupos.4))
ggplot(df,aes(x=PC1,y=PC2,color=etiqueta))+
  geom_point(alpha=0.7)+
  xlab(paste('PC1',paste0(a1[1],'%'),sep=' '))+
  ylab(paste('PC2',paste0(a1[2],'%'),sep=' '))
```




Proporcion acumulada de varianza explicada
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
50
60
Componente Principal


PC2 6.76%
25
0
-25
-50
-50
-25
0
25
50
75
PC1 11.36%
etiqueta
1
2
3
4

Pregunta 6:

Se pide determinar qué tipo de cáncer y número de observaciones están en cada clúster y si existe algún clúster que contenga un solo tipo de cáncer.

Para esto se hizo una división de las observaciones mediante el comando “hclust” para 4 clúster dentro de “ncilabs”, obteniendo los datos en la siguiente tabla.

```
# 6) Determine que tipo de cancer y numero de observaciones contenidos en cada cluster.
#    ¿Existe algun cluster que contenga solo un tipo de cancer?. ¿Cual es este cluster?.

clust <- hclust(dist(nci.data))
clust
clust.hc <- cutree(tree = clust, k = 4)
clust.hc
table(clust.hc, nci.labs)
```

```
> table(clust.hc, nci.labs)
        nci.labs
clust.hc BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro MCF7D-repro MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
       1      2   3     2          0          0        0          0          0        8     8       6        2     8       1
       2      3   2     0          0          0        0          0          0        0     1       0        0     1       0
       3      0   0     0          1          1        6          0          0        0     0       0        0     0       0
       4      2   0     5          0          0        0          1          1        0     0       0        0     0       0
>
```

Tras observar los datos obtenidos de la tabla, se diferencia que en el clúster 3 contiene solamente cáncer derivados de leucemia, existiendo entonces un clúster que contiene solo un tipo de cáncer.

Pregunta 7:

Se solicita calcular la correlación entre la matriz “cophenetic” y la matriz de distancias.

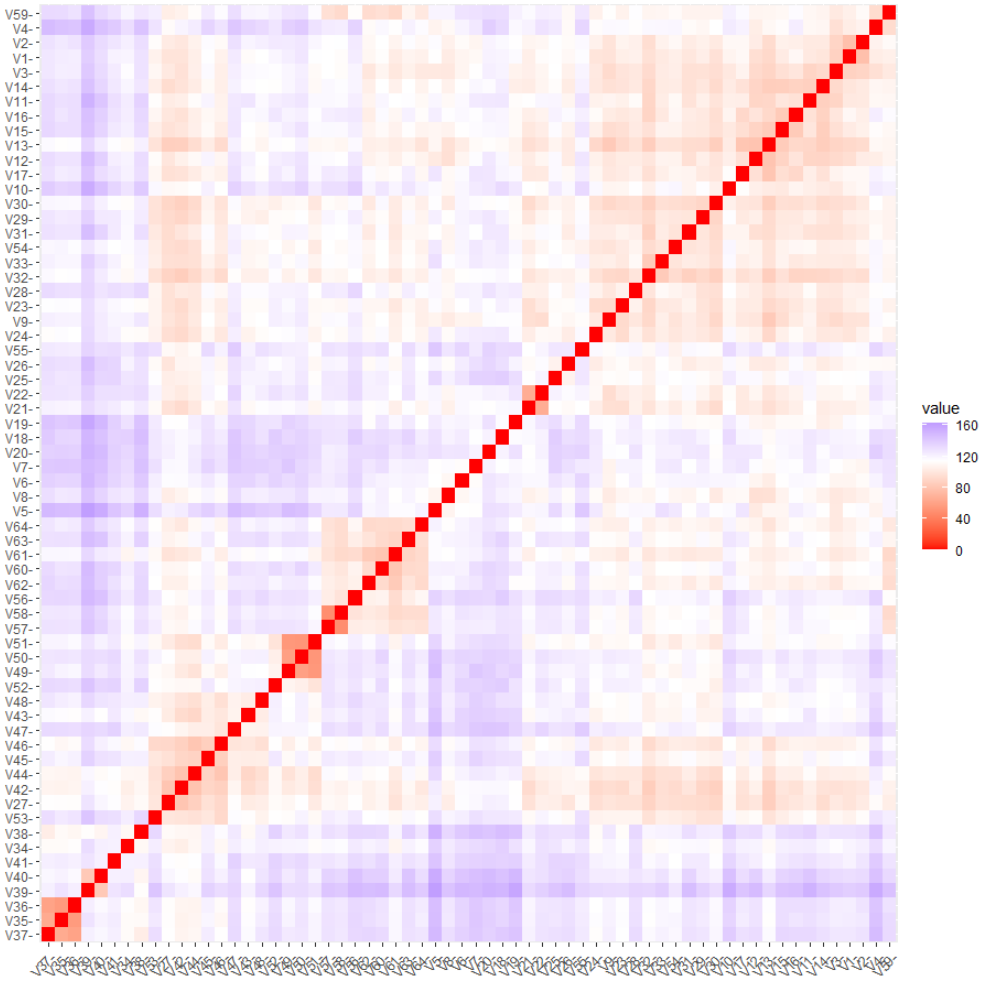
 # 7) Calcule la correlación entre la matriz "cophenetic" y la matriz de distancias. Comente.

z=cophenetic(c)

cor(z,d)

fviz_dist(d)

Obteniendo un 66% de correlación aproximadamente entre las matrices se puede afirmar que es un buen valor pero no significativo para determinar la correlación entre ambas.



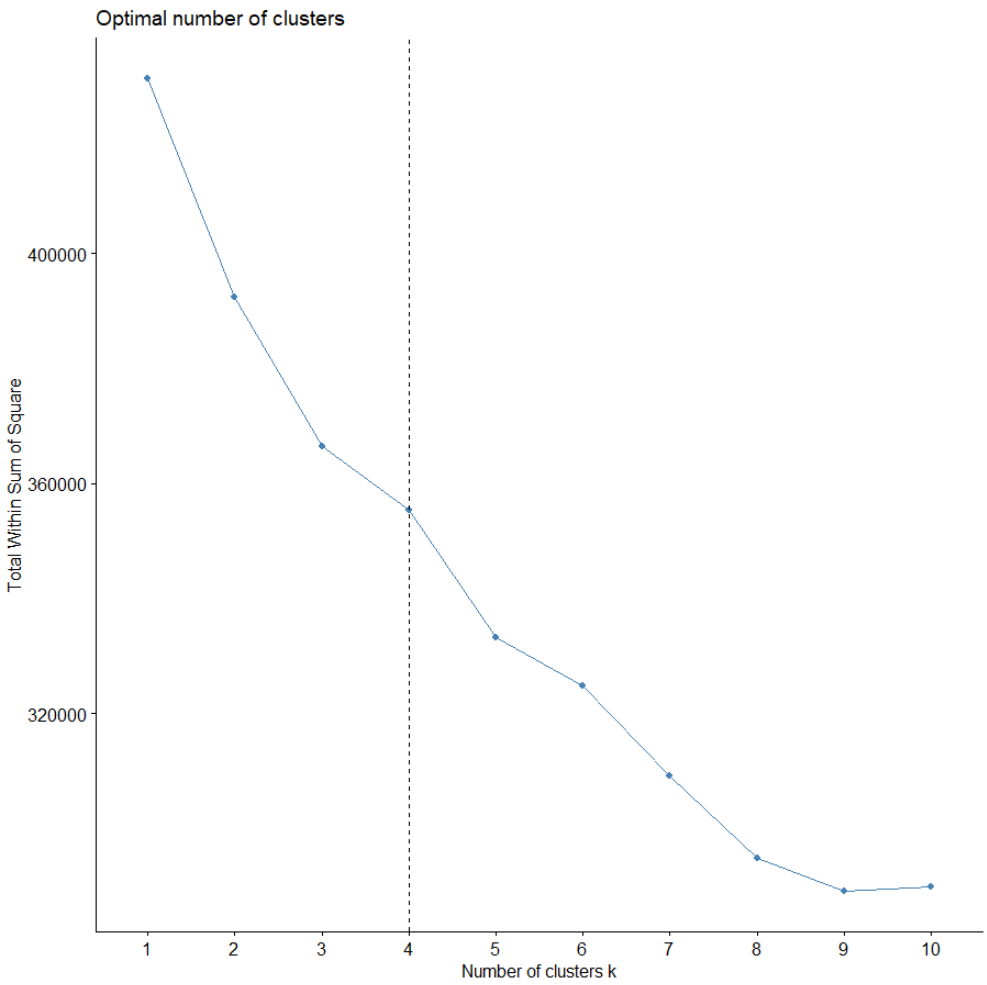
Observando la matriz de distancias, los colores son tenues por lo que no podemos afirmar una correlación muy significativa.

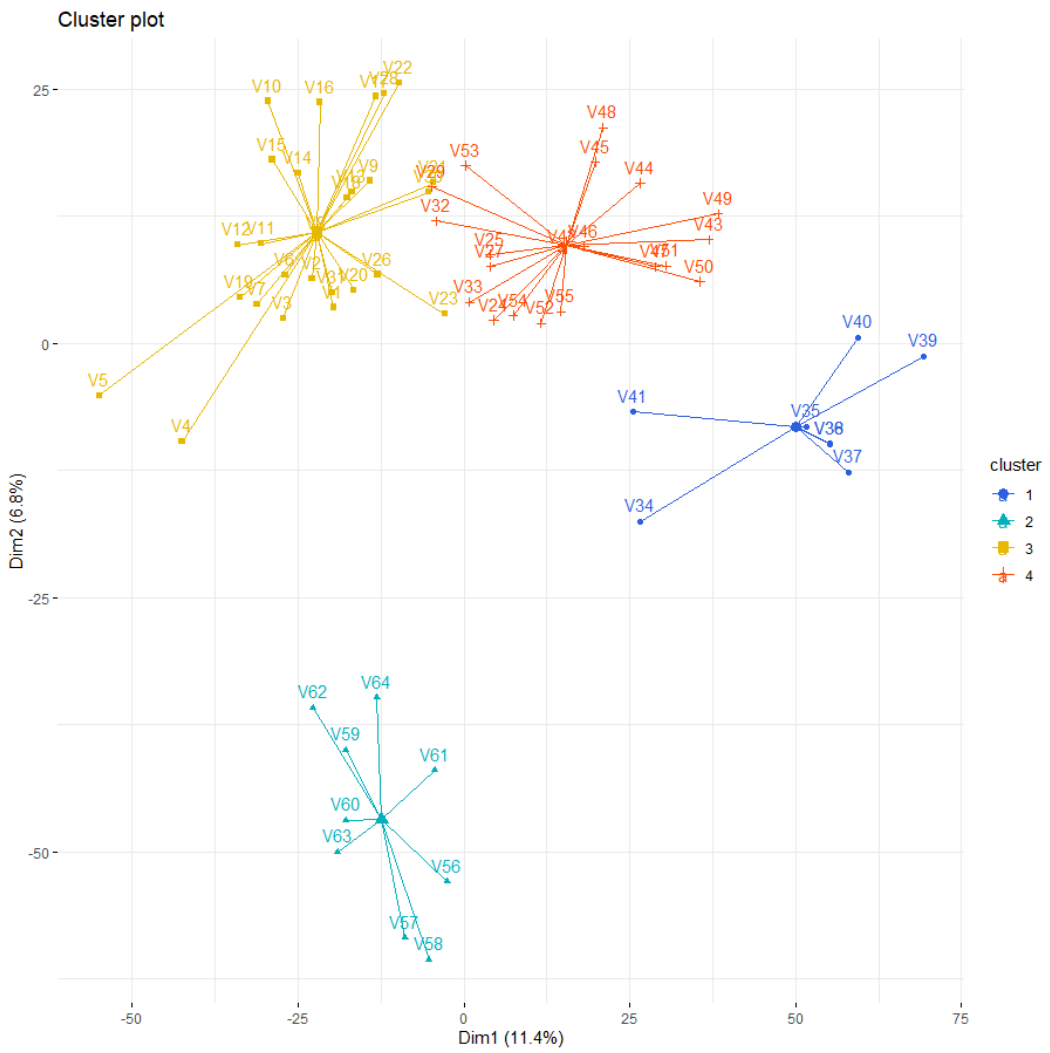
Pregunta 8:

Utilizando el método para construir clúster “K-means”, se procede a escalar los datos del dataset “ncilabs”. Dado esto, se puede apreciar que el número óptimo de cortes determinado de acuerdo a este método son 4. Gracias a esto se puede concluir que el método jerárquico es similar al “K-means”, dado que se ocupa el mismo valor óptimo de cortes (4).

```
# 8) Utilice otro metodo para construir los cluster, por ejemplo,"PAM"y/o "K-means" y
#     compare la estructura obtenida con el analisis realizado mediante el metodo jerarquico

# se realizara mediante el metodo de k-means
nci.datas <- scale(nci.data)
set.seed(123)
res=kmeans(nci.datas, 4, iter.max = 20, nstart = 25)
table(rownames(nci.datas),res$cluster)
sapply(unique(res$cluster),function(g)rownames(nci.datas)[res$cluster == g])
aggregate(nci.datas,list(res$cluster),median)
aggregate(nci.datas,list(res$cluster),mean)
print(res)
fviz_nbclust(nci.datas, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2)
# Sin usar datos escalados
aggregate(nci.data, by=list(cluster=res$cluster), mean)

#Visualizando los clusters (usa PCA para p>2)
fviz_cluster(res, data = nci.data,
             palette = c("#2E5FDF", "#02AFBB", "#E7B800", "#FC4E07"),
             ellipse.type = "jerarquic",
             star.plot = TRUE, # Adiciona segmentos desde el centroide
             ggtheme = theme_minimal()
```

)


Optimal number of clusters
Total Within Sum of Square
400000
360000
320000
1
2
3
4
5
6
7
8
9
10
Number of clusters k



De acuerdo a este análisis de los dos primeros componentes, se puede apreciar que existe una cierta caracterización de cada clúster, ya que los datos no están distribuidos de manera homogénea y, por lo tanto, se puede concluir qué observación corresponde a cada clúster.

Pregunta 9:

A modo de comentario, cabe destacar lo señalado en la pregunta 8, la división de 4 clúster es óptima y se puede diferenciar de buena manera (valores arrojados tras desarrollo de método jerárquico e “K-means”). Existiendo también clúster que contienen 1 solo tipo de cáncer.

Por lo que se puede concluir que el método de creación de clústeres “K-means” es similar al método jerárquico.

Problema 3

Para el desarrollo de este problema utilizaremos el dataset “Frutas” indicado en la plataforma “Canvas” el cual fue descargado y descomprimido en la carpeta de trabajo.

Se ejecutó el código mostrado en el anuncio del trabajo para permitir leer las imágenes y convertirlas a un data frame.

```
# Problema 3
#########################################################################
# Para instalar el paquete EBImage que permite leer im´agenes.
#Utilice primero install.packages('EBImage'). Si no funciona
#descomente los siguientes comandos:
if (!requireNamespace("BiocManager", quietly = TRUE))
install.packages("BiocManager")
BiocManager::install()
BiocManager::install(c("EBImage"))
setwd("C:/Users/jorge/OneDrive/Escritorio/SEM 6/TIN/Examen/")# colocar el directorio de trabajo
rm(list=ls(all=TRUE)) #Elimia variables
graphics.off() # Borra los gr´aficos
#install.packages("EBImage")
library(EBImage)
library(stringr)
# Carga de los datos
l="Frutas" # Carpeta con las im´agenes:
# Funci´on para convertir las im´agenes en data.frame
image2frame=function(l,H=100){
  cname<-file.path(".",l)
  A=c(dir(cname))
  n_var=str_split(cname,"/")
  (W=n_var[[1]][2])
  (K=length(dir(cname)))
```

```
  label=c()
  K=min(c(18,K))
  ss=matrix(0,H*K,45*45*3)
  for (r in 1:K){
    cname_1<-file.path(".",W,A[r][1]) # subcarpetas
    (B=c(dir(cname_1)))
    (M=length(dir(cname_1)))
    S=min(M,H)
    image1=list()
    t=0
    for (j in 1:S){
      cname_2<-file.path(".",W,A[r][1],B[j][1]) # archivos
      s2=readImage(cname_2)
      s2=resize(s2,45,45)
      red=s2[,,1]
      g=s2[,,2]
      b=s2[,,3]
      ss[j+(r-1)*S,]= unlist(list(red,g,b))
      7
      label[j+(r-1)*S]=A[r][1]
    }
  }
  #h=length(names(ss))
  ss=data.frame(ss)
  ss$label=as.factor(label)
  ss=dplyr::distinct(ss)
  return(ss)
}
# Datos
datos=image2frame(l)
h=length(names(datos))
# Analisis de componentes principales (PCA):
```

```
out=prcomp(datos[,-h],scale=F)
#Sugerencias adicionales. Para la visualizaci´on 3D con plotly7
#tendra que dar colores a 18
#tipos de puntos que corresponden a la misma cantidad de tipos de frutas. Para esto, puede
#usar la siguiente funci´on de usuario:
  Cols=function(vec){
    cols=rainbow(length(unique(vec)))
    return(cols[as.numeric(as.factor(vec))])
  }
```

Pregunta A:

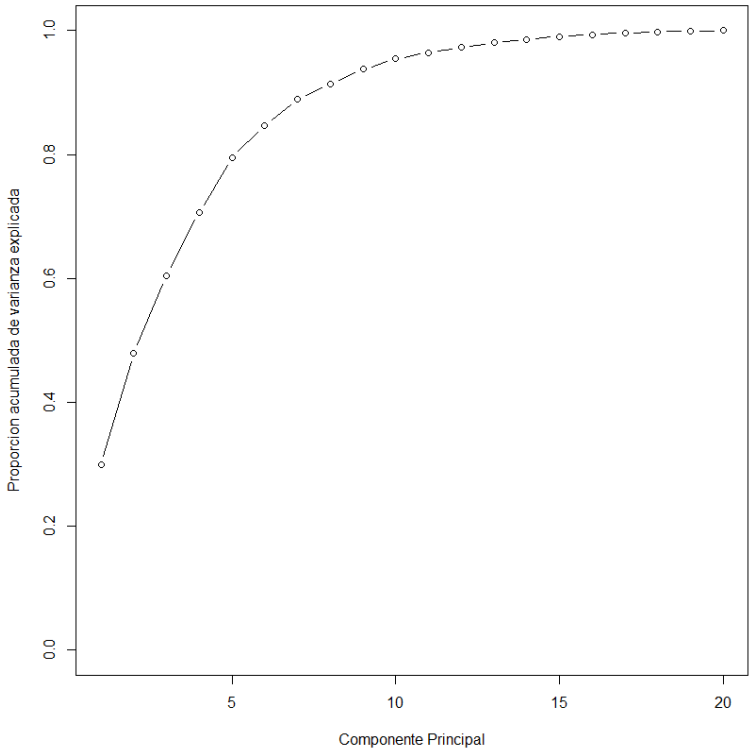
Se requiere hacer un análisis de componentes principales (PCA) y una visualización 2D con ggplot2 y 3D con plotly. Primero se visualiza el porcentaje de varianza explicada por los primeros 20 PC y podemos ver que basta con los primeros 5 PC para explicar la base de datos dado que explica más del 60%. Se generan 3 gráficos ggplot2, barras y torta (plotly) mostrados a continuación:

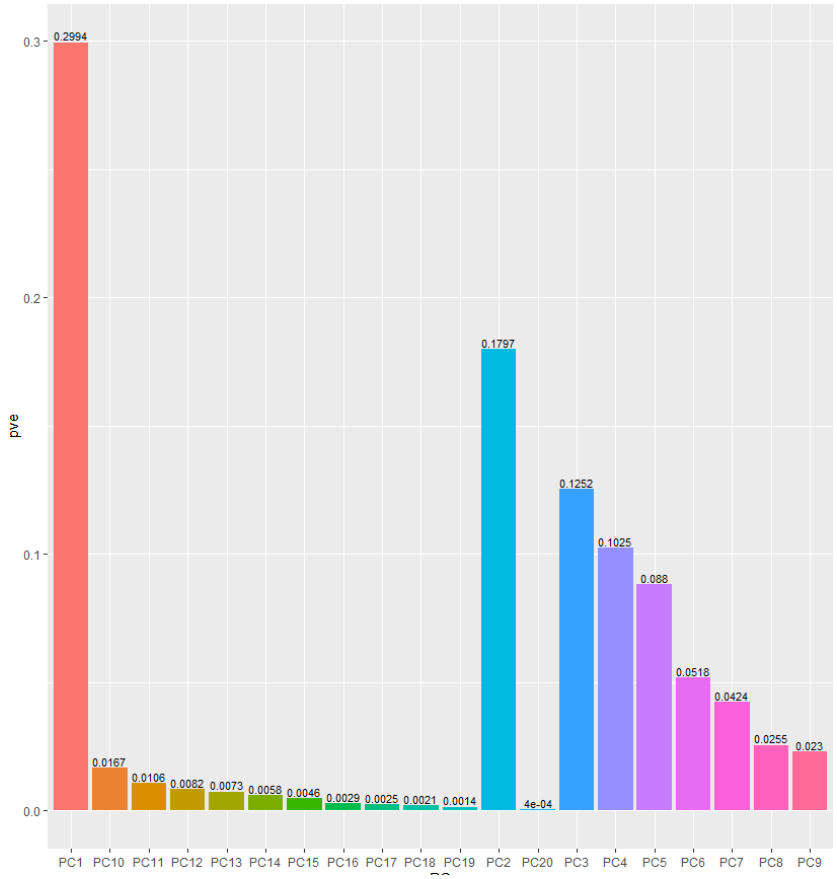
```
# se decidira ocupar los primeros 20 PC
datoss <- datos[,1:20]

# Análisis de componentes principales (PCA):
out=prcomp(datoss[,-h],scale=F)
library(ggplot2)
library(dplyr)
library(httr)
library(lubridate)
library(ggplot2)
library(RColorBrewer)
library(rgdal)
library(leaflet)
library(factoextra)
library(dendextend)
library(cluster)
```

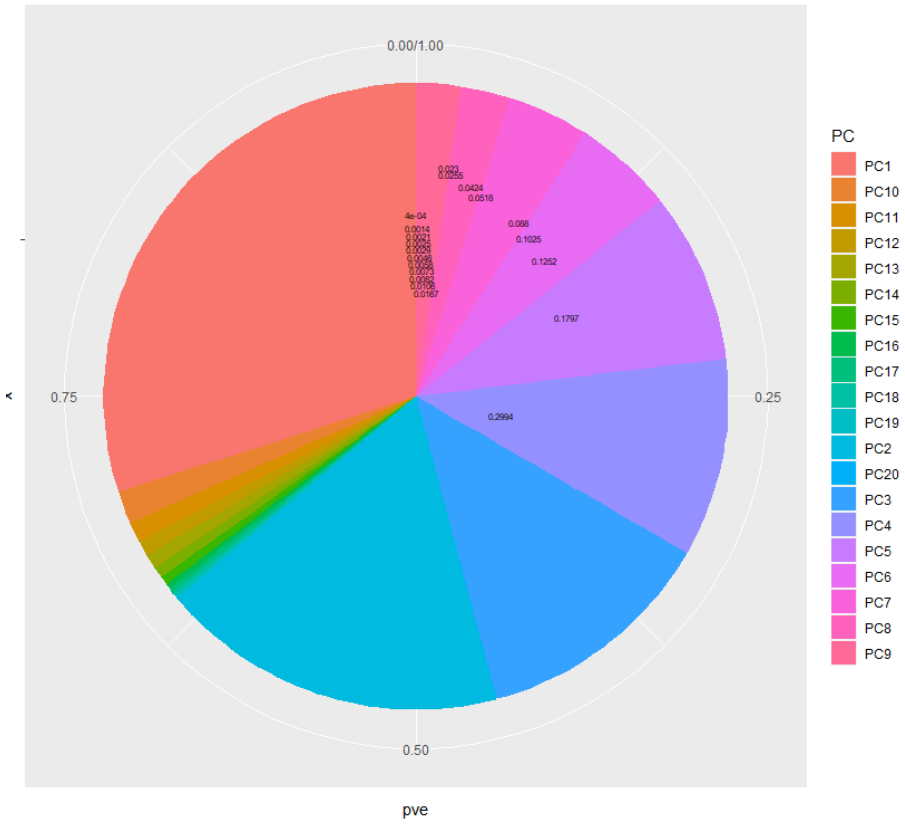
```
# A) Hacer un analisis de componentes principales (PCA) y una visualizacion 2D (con
#    ggplot2) y 3D (con plotly). Inserte las imagenes en el informe y comente los resultados.

pcfrutas <- prcomp(datoss,scale = TRUE)


# analisis de componentes principales
names(pcfrutas)
pcfrutas$center # valores promedios de los componentes principales
pcfrutas$rotation # vectores propios colgados para cada componente (matriz gamma) / se diseña la ecuacion
pcfrutas$x # nueva tabla evaluado con los valores (sirve para crear los diagramas) / teorema espectral
pcfrutas$sdev # desviacion estandar de los componentes principales
pcfrutas$scale # valores promedios divididos por las desviaciones estandar (variables escaladas)


# porcentajes de varianza explicada por cada componente        principal
pcfrutas.var <- (pcfrutas$sdev)^2
pve <- pcfrutas.var/sum(pcfrutas.var)
pve <- round(pve,4)
plot(cumsum(pve), xlab="Componente Principal", ylab="Proporcion acumulada de varianza explicada", ylim=c(0,1),type='b')


library(ggplot2)
df <- data.frame(PC=c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8", "PC9", "PC10",
                      "PC11", "PC12", "PC13", "PC14", "PC15", "PC16", "PC17", "PC18", "PC19", "PC20"),
                 pve=pve)
ggplot(data=df,aes(x=PC,y=pve,fill=PC))+
  geom_bar(stat = "identity", show.legend = F)+
  geom_text(aes(label=pve), position=position_dodge(width=0.9), vjust=-0.25,size=3)
```

```
P <- ggplot(data=df,aes(x='',y=pve,fill=PC))+geom_bar(stat = "identity",width=2)
P <- P+geom_text(aes(y=pve,label = pve), position=position_dodge(width=0.9),
            vjust=-0.5,size=2)
P+coord_polar("y", start=0)

library(plotly)
plot_ly(df, labels = ~PC, values = ~pve, type = "pie") %>%
  layout(title = "PCA para Frutas",
       xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
       yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

# reconocimiento de patrones separando los grupos en clusters
# determina la distancia entre las observaciones
y <- pcfrutas$x
pcfrutas.dist <- dist(y)
# aplica un método jerárquico para construir clusters:
pcfrutas.hclust <- hclust(pcfrutas.dist,method='ward.D')
# indica que usaremos 4 clusters
grupos.4 <- cutree(pcfrutas.hclust,4)
datos1 <- data.frame(datoss,grupos.4=grupos.4)
library(ggfortify)
autoplot(prcomp(datos1,scale=T), data = datos1,
       loadings = TRUE, loadings.colour = 'brown',
       loadings.label = TRUE, loadings.label.size = 4,label = TRUE,
       label.colour = grupos.4,shape = FALSE,label.size=3)

# correlacion las variables antiguas x1, x2, con los dos primeros componentes principales PC1,
PC2
x <- datoss
(n = nrow(x))
(h=ncol(datoss))
```

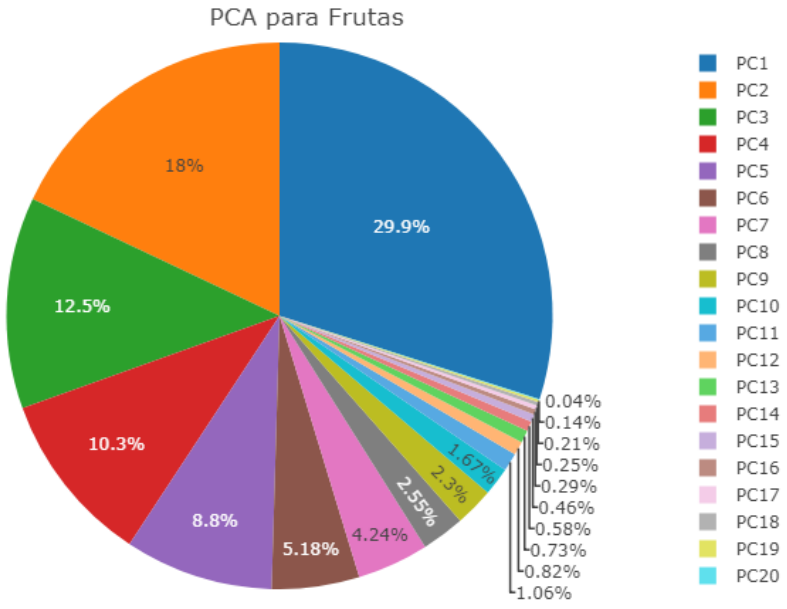
```
# calculates eigenvalues and eigenvectors and sorts them by size
e <- eigen((n-1)*cov(x)/n)
e1 <- e$values/sum(e$values)
m <- apply(as.matrix(x),2,mean)
temp <- as.matrix(x-matrix(m,n,ncol(x),byrow=T))
r <- temp%*%e$vectors
# Calcula la correlacion entre los PC y las variables originales
r <- cor(cbind(r,x))
a=h+1
b=2*h
# La correlacion entre los dos PC mas importantes y las variables
r1 <- r[a:b,1:2]
r1
# grafico de las correlaciones entre los dos primeros PC y las variables originales
ucircle=cbind(cos((0:360)/180*pi),sin((0:360)/180*pi))
plot(ucircle,type="l",lty="solid",col="blue",xlab="Primer PC",ylab="Segundo PC",
    main="Análisis Frutas",cex.lab=1.2,cex.axis=1.2,cex.main=1.8,lwd=2)
abline(h=0.0,v=0.0)
label=names(x)
text(r,label,cex=0.7)
```


Proporcion acumulada de varianza explicada
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
Componente Principal


pve
0.0
0.1
0.2
0.3
0.2994
0.0167
0.0106
0.0082
0.0073
0.0058
0.0046
0.0029
0.0025
0.0021
0.0014
0.1797
4e-04
0.1252
0.1025
0.088
0.0518
0.0424
0.0255
0.023
PC1
PC10
PC11
PC12
PC13
PC14
PC15
PC16
PC17
PC18
PC19
PC2
PC20
PC3
PC4
PC5
PC6
PC7
PC8
PC9

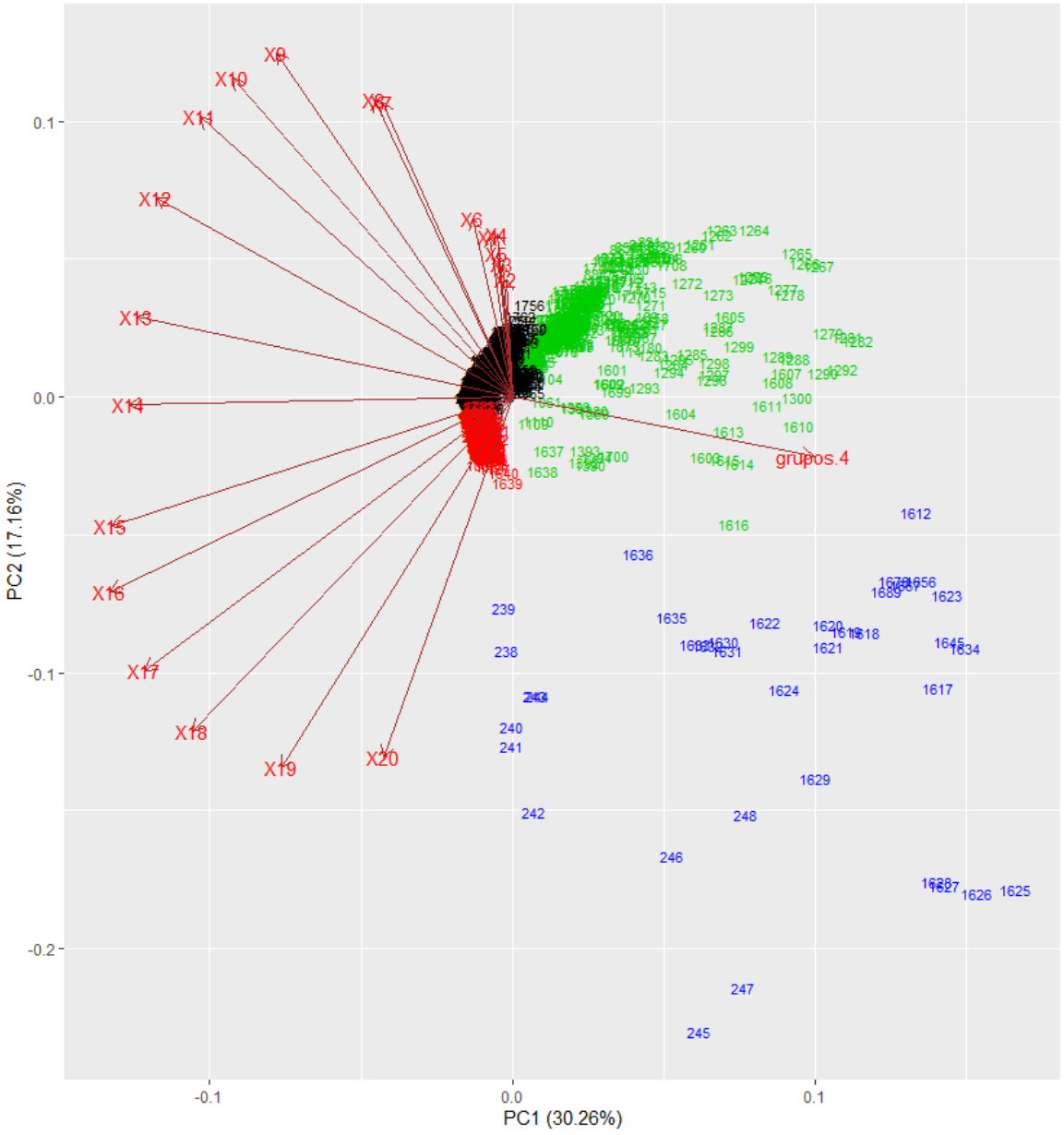


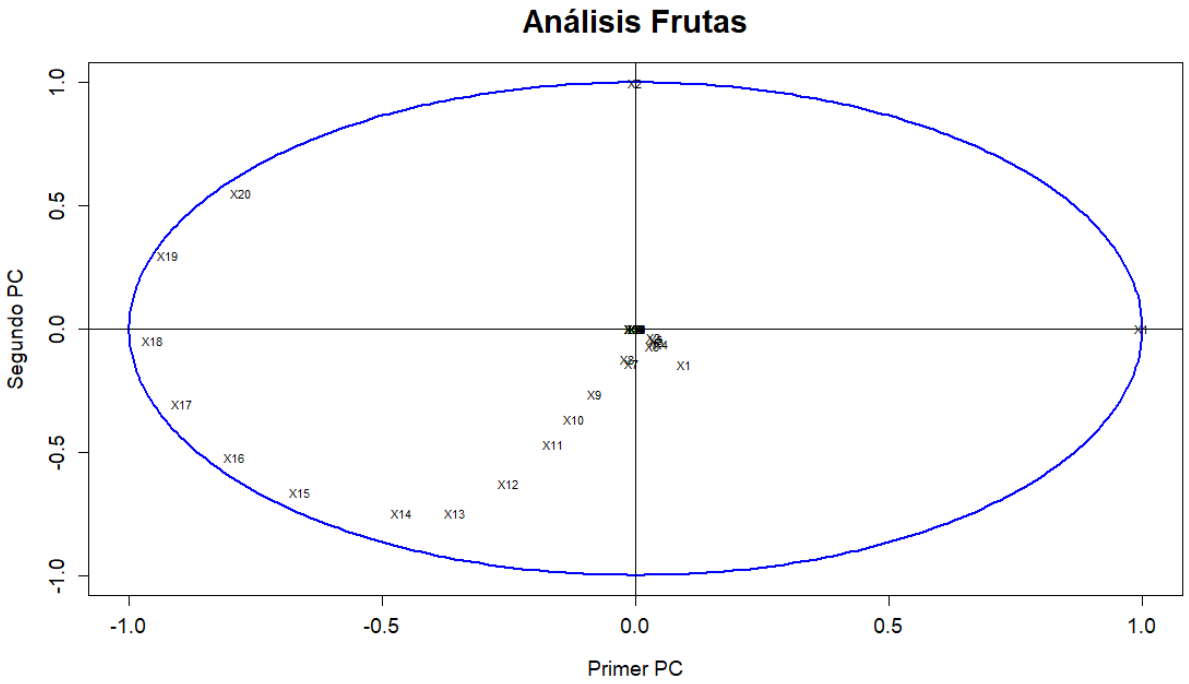
Visualización 2D de los componentes principales solicitada



Visualización 3D de los componentes principales solicitada

Finalmente se realiza un reconocimiento de patrones separando los grupos en clúster para determinar la distancia entre observaciones y se genera un “autoplot”, junto con un gráfico circular para el análisis.





Tras un análisis del gráfico circular, se puede observar que, horizontalmente, las variables x18, x17 y x16 del lado izquierdo están en el borde del circulo y, por la contraparte, la variable x1; dado esto, la resta de estas variables, debido a que están alejadas unas de otras y a la vez pegadas al eje del primer componente principal, tienen un coeficiente grande (cercano a uno) y las que más contribuyen a la separación de los datos de este componente principal. Al ser el primer componente principal el más importante (poco más del 25%), estas variables son las más importantes que existen.

Verticalmente (segundo componente principal), encontramos la variable x2 y x7, por tanto, la resta de estas dos variables será la con mayor relevancia para el segundo componente principal.

Gracias al gráfico “autoplot”, se puede confirmar el análisis realizado para el gráfico circular, ya que hace referencia a un análisis similar.

Sin embargo, ya que estos dos componentes principales no alcanzan a representar más de la mitad de las observaciones, no hay suficiente evidencia para afirmar que estos análisis son representativos para todas las observaciones.

Pregunta B:

Para construir el nuevo dataset a partir de los primeros 20 PC y adicionar una columna con “label” que contenga el nombre de las frutas se utilizaron las siguientes líneas de código:

```
# B) Construya un nuevo dataset a partir de los primeros 20 componentes principales
#    obtenidos. Adicione una columna con "label", que contiene los nombres de las frutas.
#    (incluya los codigos)

dataset <- as.data.frame(datoss)
label <- datos$label
dataset$label <- label
```

Pregunta C y D:

Se procede a dividir el dataset creado anteriormente en un conjunto “train” 80% y otro “test” 20% para construir un clasificador SVM con kernel lineal. Primero se define el mejor costo para nuestro clasificador el cual es costo 10. Luego se define la tabla de confusión obteniendo los siguientes niveles de Especificidad y Sensibilidad de todas las clases de Frutas.

```
# C) Divida el dataset, creado en el punto anterior, en un conjunto "train" y otro "test"
#    (80/20) y construya un clasificador SVM con kernel lineal.
library(e1071)

set.seed(123)
train <- sample(1:nrow(dataset),0.8*nrow(dataset),replace=F)
Train.data<- dataset[train,]
Test.data<-dataset[-train,]
#install.packages("caret")
library(caret)
library(klaR)
tune.out=tune(svm,label~.,data=Train.data,kernel="linear",
              ranges=list(cost=c(0.01,0.1,1.0,10)))
summary(tune.out)
bestmod=tune.out$best.model
summary(bestmod)
```

```
svmfit=svm(label~., data=Train.data, kernel="linear",cost=10)
summary(svmfit)

# D) Determine el desempeño del clasificador usando los datos "test".
p=predict(svmfit,Test.data[,-21])
tabla <- table(predichos=p,reales=Test.data$label)
tabla
confusionMatrix(tabla)
```

CLASE	ESPECIFICIDAD	SENSIBILIDAD
Apple Red	0.97384	0.43750
Apricot	0.95870	0.61905
Banana	0.95308	0.84211
Cocos	0.99110	0.26087
Kiwi	0.95906	0.77778
Lemon	0.98503	0.42308
Limes	0.94798	0.92857
Mandarine	0.94493	0.86667
Mango	0.98534	0.21053
Maracuja	0.96532	0.50000
Nectarine	0.97917	0.33333
Orange	0.96471	0.60000
Papaya	0.97935	0.57143
Peach	0.94012	0.57692
Pear	0.98802	0.30769
Pineapple	0.97947	0.47368
Raspberry	0.98824	0.40000
Strawberry	0.982405	0.105263

A través de la tabla de confusión mostrada, el mejor desempeño del clasificador usando los datos “test” se obtiene para las clases: Banana, Kiwi, Limes y Mandarine, con especificidades sobre el 90% y sensibilidad sobre 75%. Por la contraparte, las clases Strawberry, Mango y Cocos tienen una sensibilidad menor al 30%, por lo que el clasificador creado no sería el más recomendable para determinar este tipo de frutas, si se busca con alta probabilidad todos los verdaderos positivos de este tipo de frutas.