Minería de datos y modelización predictiva

Profesora: Juana María Alonso

Alumno: Guillermo Sánchez-Mariscal

EJERCICIO DE EVALUACIÓN I

ANÁLISIS DE COMPONENTES PRINCIPALES Y CLUSTER

El fichero Provincias contiene información socio-económica de las provincias españolas. (Se han modificado los nombres de las variables). Para reducir el número de variables e intentar encontrar relaciones, tanto entre variables como entre provincias, realizar los siguientes apartados.

Cargo las librerías que voy a utilizar:

```
library(psych)
library(tidyverse)
library(nortest)
library(readxl)
library(pastecs)
library(corrplot)
library(factoextra)
library(FactoMineR)
library(lattice)
library(cluster)
library("heatmaply")
library(NbClust)
```

Cargo en un dataframe los datos del fichero provincias y cambio los índices por el nombre de las provincias:

```
datos <- as.data.frame(read_excel("./Provincias.xlsx"))
rownames(datos) <- datos[,1]
datos <- datos[,-1]
```

A continuación, mostraré un pequeño resumen de las variables que tenemos.

Variables	Significado
Pobl Total	Población total
Mortalidad	Mortalidad (defunciones por mil habitantes)
Natalidad	Tasa Bruta de Natalidad (nacidos por mil habitantes)
IPC	Índice de precios de consumo
Número de empresas	
Industria	Industria (nº empresas)
Construcción	Construcción (nº empresas)
CTH	Comercio, transporte y hostelería (nº empresas)
Infor.	Información y comunicaciones (nº empresas)
AFS.	Actividades financieras y de seguros (nº empresas)
APT	Actividades profesionales y técnicas (nº empresas)
Tasa Actividad	Cociente entre la población activa y la población en edad de trabajar
Tasa de paro	Mide el nivel de paro de un país
Ocupados	Número de personas con empleo en España
PIB	PIB a precios de mercado (miles de euros)
CANE	Censo Agrario Número de Explotaciones
TVF	Censo 2011: Total viviendas familiares
VS	Censo 2011: Viviendas secundarias

1. Calcular la matriz de correlaciones, y su representación gráfica ¿Cuáles son las variables más correlacionadas de forma inversa?

Matriz de correlaciones

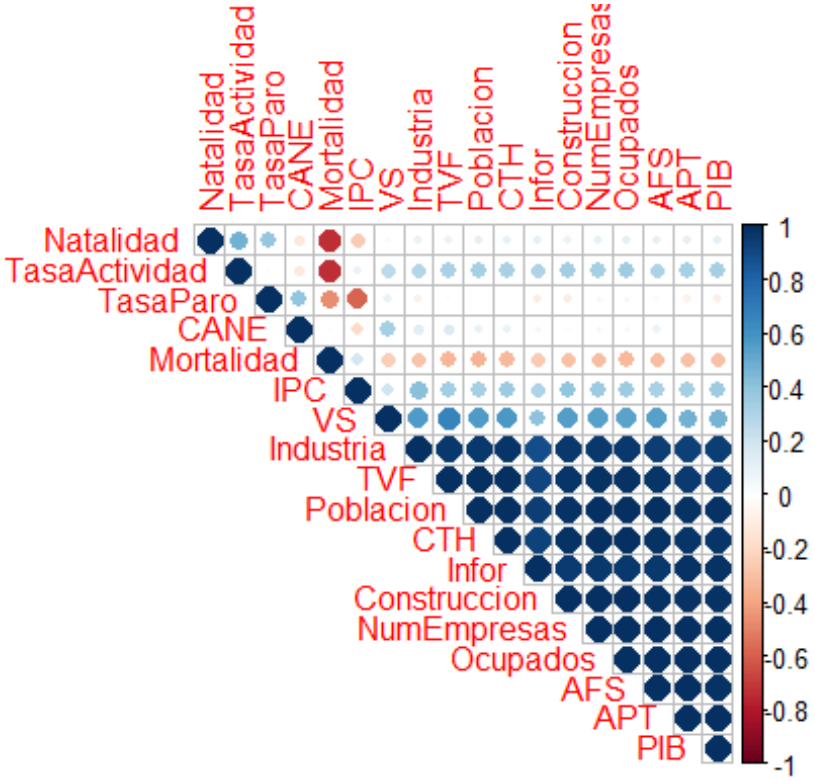
```
R <- cor(datos)
knitr::kable(R, digits =2,caption = "Correlaciones")
```

	Poblacion	Mortalidad	Natalidad	IPC	NumEmpresas	Industria	Construccion	CTH	Infor	AFS	APT	TasaActividad	TasaParo	Ocupados	PIB	CANE	TVF	VS
Poblacion	1.00	-0.34	0.11	0.33	0.99	0.96	0.98	1.00	0.94	0.99	0.98	0.33	0.01	1.00	0.98	0.10	0.99	0.57
Mortalidad	-0.34	1.00	-0.74	0.19	-0.31	-0.28	-0.30	-0.33	-0.26	-0.31	-0.30	-0.73	-0.46	-0.33	-0.30	0.02	-0.33	-0.25
Natalidad	0.11	-0.74	1.00	-0.25	0.11	0.09	0.09	0.10	0.11	0.10	0.11	0.47	0.38	0.11	0.11	-0.12	0.08	-0.03
IPC	0.33	0.19	-0.25	1.00	0.36	0.42	0.40	0.36	0.30	0.32	0.33	0.09	-0.58	0.36	0.36	-0.19	0.34	0.19
NumEmpresas	0.99	-0.31	0.11	0.36	1.00	0.97	0.99	0.99	0.96	0.99	0.99	0.33	-0.06	1.00	0.99	0.04	0.98	0.54
Industria	0.96	-0.28	0.09	0.42	0.97	1.00	0.97	0.98	0.89	0.95	0.93	0.29	-0.08	0.96	0.94	0.12	0.97	0.57
Construccion	0.98	-0.30	0.09	0.40	0.99	0.97	1.00	0.99	0.96	0.98	0.98	0.34	-0.11	0.99	0.99	0.03	0.98	0.56

CTH	1.00	-0.33	0.10	0.36	0.99	0.98	0.99	1.00	0.93	0.98	0.97	0.33	-0.01	0.99	0.97	0.09	0.99	0.58
Infor	0.94	-0.26	0.11	0.30	0.96	0.89	0.96	0.93	1.00	0.97	0.99	0.31	-0.11	0.96	0.99	-0.07	0.91	0.41
AFS	0.99	-0.31	0.10	0.32	0.99	0.95	0.98	0.98	0.97	1.00	0.99	0.32	-0.03	0.99	0.99	0.09	0.98	0.54
APT	0.98	-0.30	0.11	0.33	0.99	0.93	0.98	0.97	0.99	0.99	1.00	0.33	-0.08	0.99	1.00	-0.01	0.96	0.48
TasaActividad	0.33	-0.73	0.47	0.09	0.33	0.29	0.34	0.33	0.31	0.32	0.33	1.00	0.03	0.35	0.33	-0.12	0.33	0.26
TasaParo	0.01	-0.46	0.38	-0.58	-0.06	-0.08	-0.11	-0.01	-0.11	-0.03	-0.08	0.03	1.00	-0.05	-0.10	0.39	0.01	0.10
Ocupados	1.00	-0.33	0.11	0.36	1.00	0.96	0.99	0.99	0.96	0.99	0.99	0.35	-0.05	1.00	0.99	0.05	0.98	0.54
PIB	0.98	-0.30	0.11	0.36	0.99	0.94	0.99	0.97	0.99	0.99	1.00	0.33	-0.10	0.99	1.00	-0.01	0.96	0.47
CANE	0.10	0.02	-0.12	-0.19	0.04	0.12	0.03	0.09	-0.07	0.09	-0.01	-0.12	0.39	0.05	-0.01	1.00	0.15	0.34
TVF	0.99	-0.33	0.08	0.34	0.98	0.97	0.98	0.99	0.91	0.98	0.96	0.33	0.01	0.98	0.96	0.15	1.00	0.67
VS	0.57	-0.25	-0.03	0.19	0.54	0.57	0.56	0.58	0.41	0.54	0.48	0.26	0.10	0.54	0.47	0.34	0.67	1.00

Representación gráfica

```
corrplot(R, type="upper", order="hclust")
```



La mayor correlación negativa la encontramos entre mortalidad y natalidad (-0.74), y también en tasa de actividad y mortalidad (-0.73). En general, las correlaciones entre mortalidad y las demás variables son negativas, excepto para el IPC y el censo agrario.

Además hay otra correlación negativa que destacar, no relacionada con la mortalidad que es la de tasa de paro con IPC (-0.58)

2. Realizar un análisis de componentes principales sobre la matriz de correlaciones, calculando 7 componentes. Estudiar los valores de los autovalores obtenidos y las gráficas que los resumen. ¿Cuál es el número adecuado de componentes?

Si tomamos demasiadas variables es difícil visualizar relaciones entre ellas, como hemos podido ver en el gráfico anterior. Por lo que necesitamos reducir el máximo número de variables sin perder información. Se buscan variables que sean combinaciones lineales de las originales y que no estén relacionadas, recogiendo la mayor parte de la información o variabilidad de los datos.

Analizamos los componentes principales, calculando 7 componentes sobre la matriz de correlaciones (aunque en los gráficos podemos ver el total de las variables)

```
fit <- PCA(datos, scale.unit = TRUE, graph = TRUE, ncp = 7)
```

En este gráfico podemos ver por ejemplo como mortalidad y natalidad son vectores con dirección contraria.

A continuación, mostraremos la proporción de variabilidad explicada por la componente.

```
eig<-get_eigenvalue(fit)
knitr::kable(eig, digits =2,caption = "Autovalores")
```

Autovalores

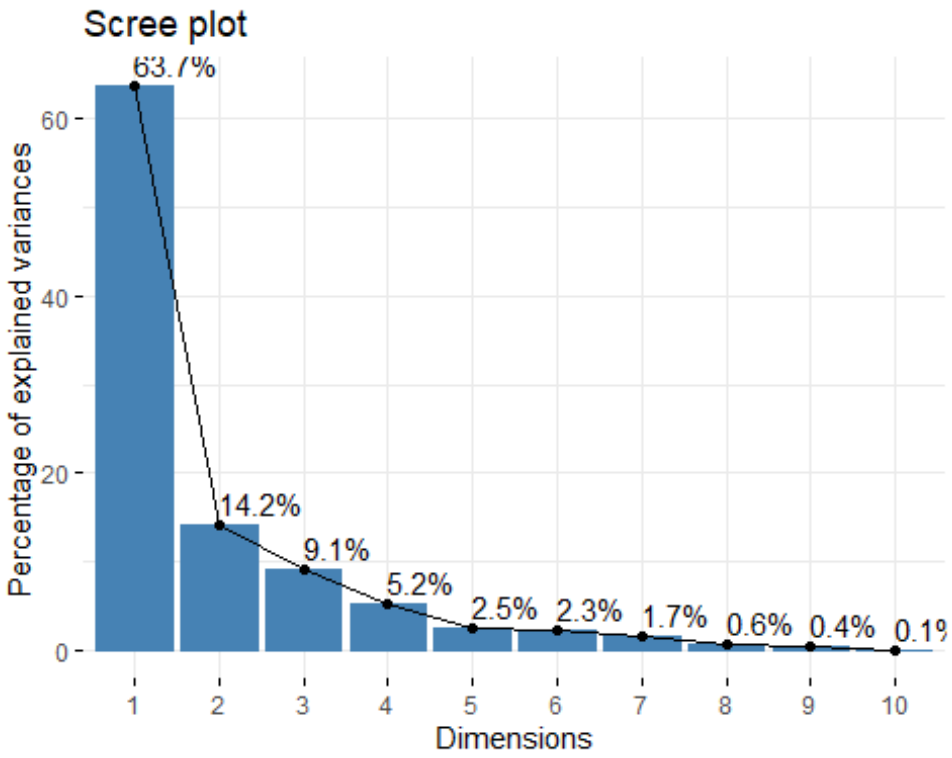
	eigenvalue	variance.percent	cumulative.variance.percent
Dim.1	11.47	63.70	**63.70**
Dim.2	2.56	14.23	**77.93**
Dim.3	1.63	9.08	**87.01**
Dim.4	0.93	5.19	**92.19**
Dim.5	0.46	2.54	**94.73**
Dim.6	0.41	2.30	**97.03**
Dim.7	0.31	1.71	**98.74**
Dim.8	0.12	0.65	**99.39**
Dim.9	0.07	0.41	**99.79**
Dim.10	0.02	0.11	**99.91**
Dim.11	0.01	0.05	**99.96**
Dim.12	0.00	0.02	**99.98**
Dim.13	0.00	0.01	**99.99**
Dim.14	0.00	0.00	**99.99**
Dim.15	0.00	0.00	**100.00**

Dim.16	0.00	0.00	**100.00**
Dim.17	0.00	0.00	**100.00**
Dim.18	**0.00**	**0.00**	**100.00**

Lo ideal es explicar aproximadamente el 90% de la variabilidad total (columna 3), en nuestro caso sería quedarnos entre tres y cuatro componentes.

Representación gráfica de la proporción de varianza explicada por cada componente

```
fviz_eig(fit, addlabels = TRUE)
```



3. Hacer de nuevo el análisis sobre la matriz de correlaciones, pero ahora indicando el número de componentes principales que hemos decidido retener (Que expliquen aproximadamente el 90%). Sobre este análisis contestar los siguientes apartados.

Decidimos hacerlo con 3 componentes

```
fit <- PCA(datos, scale.unit = TRUE, graph = TRUE, ncp = 3)
```

a. Mostrar los coeficientes para obtener las componentes principales ¿Cuál es la expresión para calcular la primera Componente en función de las variables originales?

```
knitr::kable(fit$svd$V, digits =3,caption = "Autovectores")
```

Autovectores

0.294	0.002	**0.050**
-0.106	-0.527	**0.189**
0.041	0.495	**-0.271**
0.110	-0.365	**-0.262**
0.294	-0.026	**0.008**
0.286	-0.045	**0.046**
0.293	-0.045	**-0.012**
0.293	-0.011	**0.049**
0.282	-0.042	**-0.065**
0.292	-0.016	**0.040**
0.291	-0.029	**-0.028**
0.114	0.331	**-0.363**
-0.014	0.462	**0.387**
0.294	-0.017	**0.002**
0.291	-0.036	**-0.037**
0.018	0.096	**0.657**
0.292	-0.002	**0.100**
0.172	0.048	**0.290**

$$CP_1 \coloneqq 0.294 Poblacion^* - 0.106 Mortalidad^* + 0.041 Natalidad^* + 0.110 IPC^* + \cdots + 0.172 VS^*$$

b. Mostar una tabla con las correlaciones de las Variables con las Componentes Principales. Para cada Componente indicar las variables con las que está más correlacionada

```
var<-get_pca_var(fit)
knitr::kable(var$cor, digits =2,caption = "Correlaciones de la CP con las variables")
```

Correlaciones de la CP con las variables

	Dim.1	**Dim.2**	***Dim.3***
Poblacion	0.99	0.00	*0.06*
Mortalidad	-0.36	-0.84	*0.24*
Natalidad	0.14	0.79	*-0.35*
IPC	0.37	-0.58	*-0.34*
NumEmpresas	1.00	-0.04	*0.01*
Industria	0.97	-0.07	*0.06*

Construccion	0.99	-0.07	*-0.02*
CTH	0.99	-0.02	*0.06*
Infor	0.95	-0.07	*-0.08*
AFS	0.99	-0.03	*0.05*
APT	0.98	-0.05	*-0.04*
TasaActividad	0.39	0.53	*-0.46*
TasaParo	-0.05	0.74	*0.50*
Ocupados	1.00	-0.03	*0.00*
PIB	0.99	-0.06	*-0.05*
CANE	0.06	0.15	*0.84*
TVF	0.99	0.00	*0.13*
VS	0.58	0.08	*0.37*

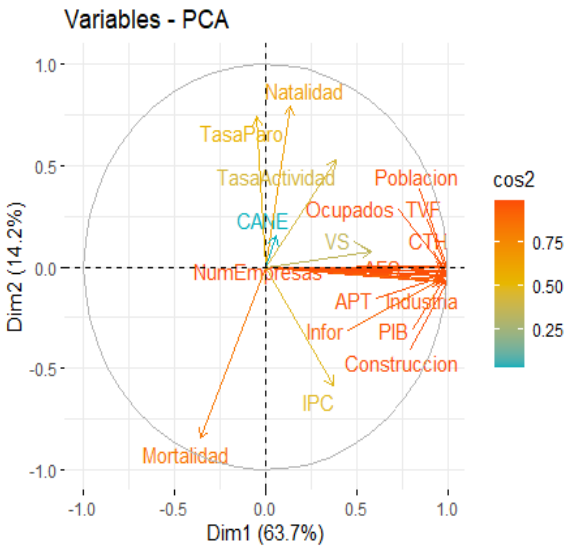
La componente uno tiene mayores correlaciones con las variables relacionadas con el número de empresas (Num empresas, Industria, Construcción, CTH, Infor, AFS, APT), población (TVF, ocupados) y PIB.

La componente dos tiene mayores correlaciones con las variables Natalidad, Tasa de actividad y Tasa de Paro. Además, también he marcado mortalidad porque está muy correlacionado negativamente.

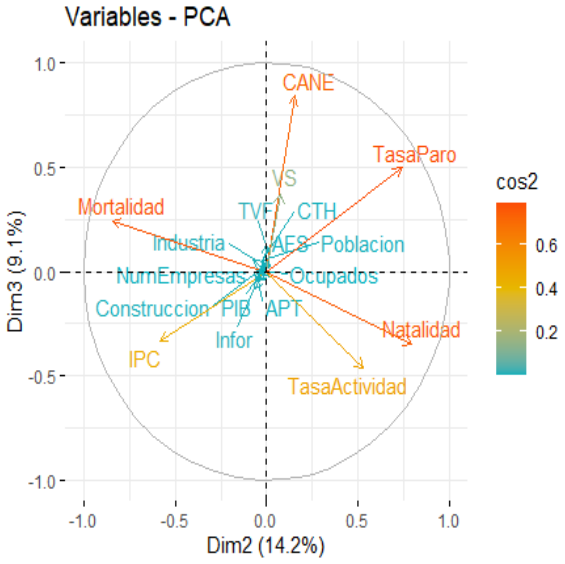
Por último la componente 3 tiene una mayor correlación con el censo Agrario Número de Explotaciones y el censo de viviendas secundarias. También podemos destacar la correlación negativa con la tasa de actividad que ya se correlacionaba en la componente anterior.

c. Comentar los gráficos que representan las variables en los planos formados por las componentes, intentando explicar lo que representa cada componente

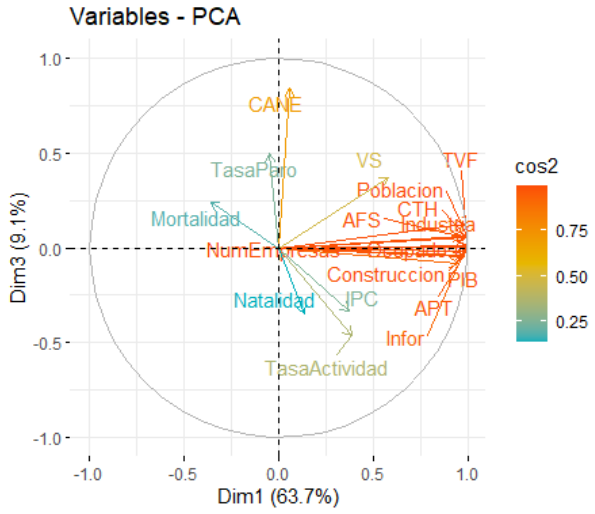
```
fviz_pca_var(fit, axes = c(1, 2), col.var="cos2", gradient.cols =
c("#00AFBB", "#E7B800", "#FC4E07"),repel = TRUE )

fviz_pca_var(fit, axes = c(1, 3), col.var="cos2", gradient.cols =
c("#00AFBB", "#E7B800", "#FC4E07"),repel = TRUE )

fviz_pca_var(fit, axes = c(2, 3), col.var="cos2", gradient.cols =
c("#00AFBB", "#E7B800", "#FC4E07"),repel = TRUE )
```



En este gráfico podemos ver mejor como la componente uno representa a las variables relacionadas con el número de empresas (Num empresas, Industria, Construcción, CTH, Infor, AFS, APT), población, TVF, ocupados y PIB. Parece que la componente uno refleja provincias con mejor economía y una mayor población.



La componente dos tiene mayores correlaciones con las variables Natalidad, Tasa de actividad y Tasa de Paro. Además, la mortalidad está muy correlacionada negativamente.



La componente 3 tiene una mayor correlación con el censo Agrario Número de Explotaciones y el censo de viviendas secundarias. También podemos destacar la correlación negativa con la tasa de actividad que ya se correlacionaba en la componente anterior.

d. Mostrar la tabla y los gráficos que nos muestran la proporción de la varianza de cada variable que es explicado por cada componente. ¿Cuál de las variables es la que está peor explicada?

Los cosenos al cuadrado son las correlaciones al cuadrado que expresan la proporción de la varianza de cada variable que es explicada por cada componente.

```
knitr::kable(var$cos2, digits =2,caption = "Cosenos al cuadrado")
```

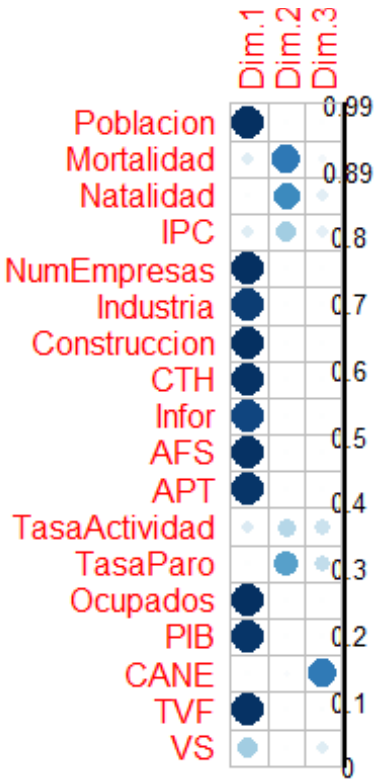
Cosenos al cuadrado

	Dim.1	Dim.2	Dim.3	Suma
Poblacion	0.99	0.00	0.00	**0,99**
Mortalidad	0.13	0.71	0.06	**0,9**
Natalidad	0.02	0.63	0.12	**0,77**
IPC	0.14	0.34	0.11	**0,59**
NumEmpresas	0.99	0.00	0.00	**0,99**
Industria	0.94	0.01	0.00	**0,95**
Construccion	0.99	0.01	0.00	**1**
CTH	0.98	0.00	0.00	**0,98**
Infor	0.91	0.00	0.01	**0,92**
AFS	0.98	0.00	0.00	**0,98**
APT	0.97	0.00	0.00	**0,97**
TasaActividad	0.15	0.28	0.22	**0,65**
TasaParo	0.00	0.55	0.25	**0,8**
Ocupados	0.99	0.00	0.00	**0,99**
PIB	0.97	0.00	0.00	**0,97**
CANE	0.00	0.02	0.70	**0,72**
TVF	0.97	0.00	0.02	**0,99**
VS	0.34	0.01	0.14	**0,49**

El censo de las viviendas secundarias es la variable menos representada.

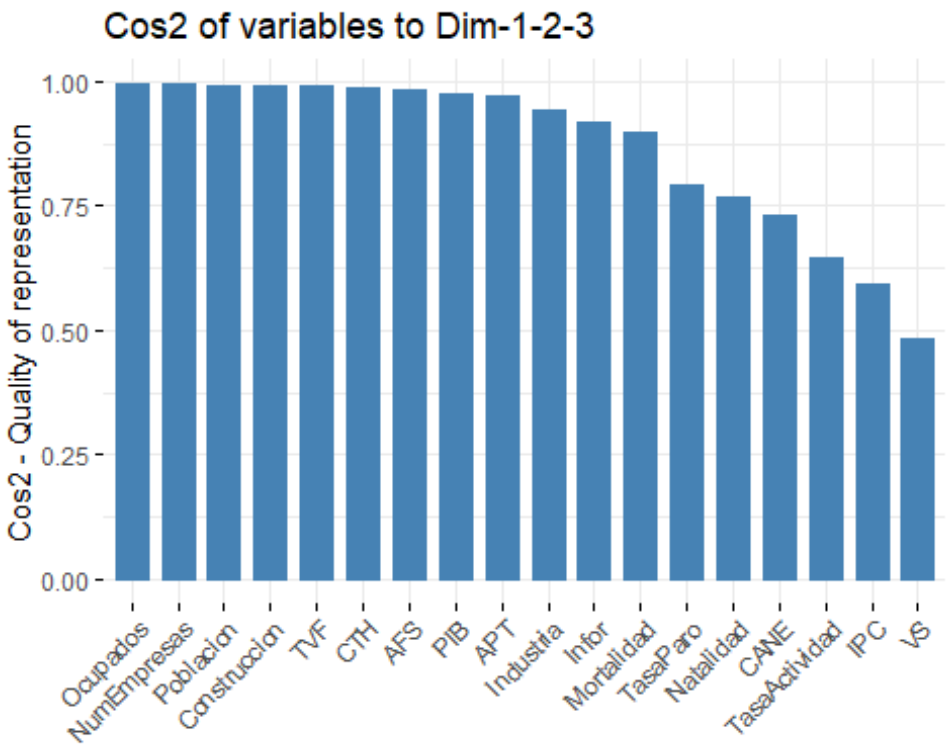
Representación gráfica de los cosenos

```
corrplot(var$cos2,is.corr=FALSE)
```



Porcentaje de variabilidad explicada por las tres CP

```
fviz_cos2(fit,choice="var",axes=1:3)
```



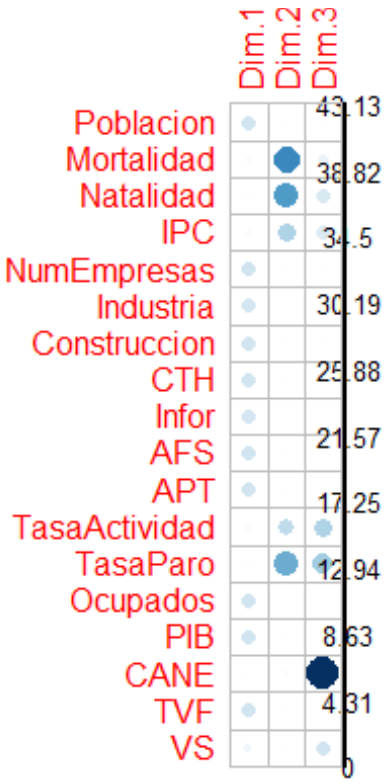
e. Mostrar la tabla y los gráficos que nos muestran el porcentaje de la varianza de cada Componente que es debido a cada variable. ¿Que variables contribuyen más a cada Componente?

```
knitr::kable(var$contrib, digits =2,caption = "Contribuciones")
```

Contribuciones

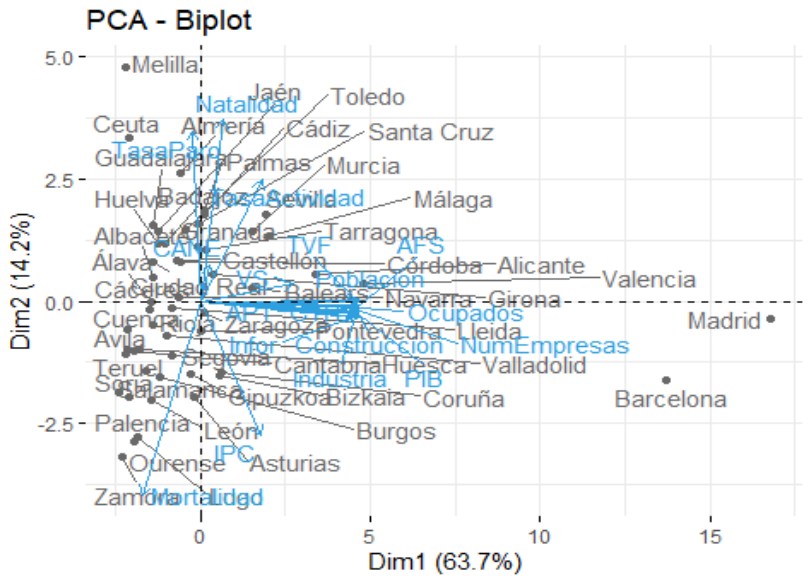
	Dim.1	Dim.2	Dim.3
Poblacion	8.62	0.00	0.25
Mortalidad	1.13	27.79	3.57
Natalidad	0.17	24.54	7.33
IPC	1.21	13.35	6.88
NumEmpresas	8.65	0.07	0.01
Industria	8.16	0.20	0.22
Construccion	8.60	0.21	0.01
CTH	8.58	0.01	0.24
Infor	7.92	0.18	0.42
AFS	8.55	0.03	0.16
APT	8.45	0.09	0.08
TasaActividad	1.31	10.93	13.16
TasaParo	0.02	21.30	15.00
Ocupados	8.67	0.03	0.00
PIB	8.46	0.13	0.14
CANE	0.03	0.93	43.13
TVF	8.50	0.00	1.00
VS	2.97	0.23	8.42

```
corrplot(var$contrib,is.corr=FALSE)
```



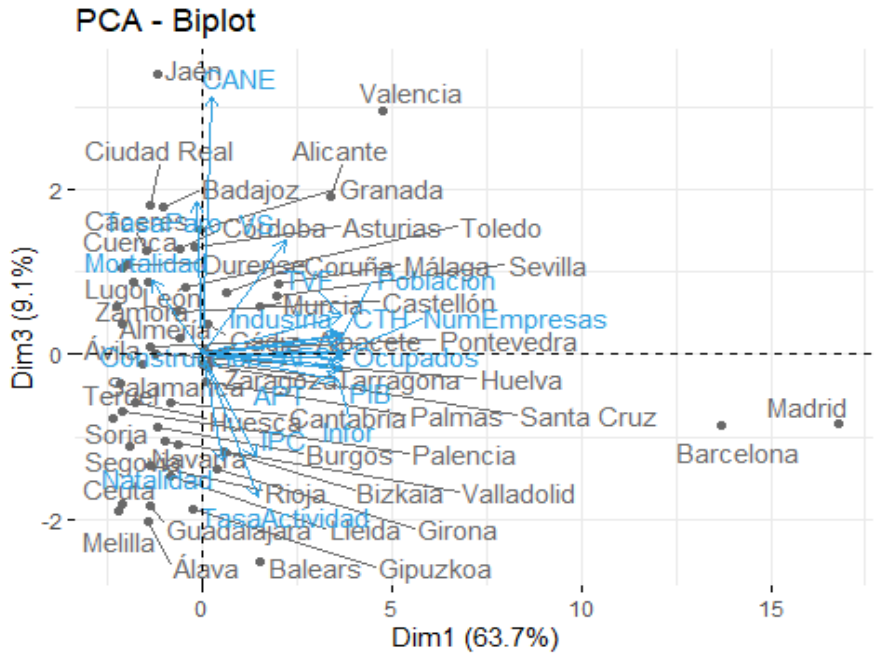
f. Sobre los gráficos que representan las observaciones en los nuevos ejes y el gráfico Biplot, teniendo en cuenta la posición de las provincias en el gráfico. Comentar las provincias que tienen una posición más destacada en cada componente, en positivo o negativo, ¿Qué significa esto en términos socioeconómicos para estas provincias?

```
fviz_pca_biplot(fit, repel = TRUE,
 col.var = "#2E9FDF", # Variables color
 col.ind = "#696969") # Individuals color
```



Del gráfico podemos destacar como Madrid y Barcelona, tienen un valor muy alto en la componente 1, que era la componente más correlacionada con el número de empresas y población. En la componente 2 vemos como provincias Palencia, Lugo, Zamora y Ourense tienen un comportamiento parecido, en este caso la componente 2 significa para estas provincias mayor tasa de mortalidad. En cambio, para las ciudades autónomas de Ceuta y Melilla que tienen un comportamiento parecido, reflejan una mayor natalidad, tasa de actividad y tasa de paro.

```
fviz_pca_biplot(fit, repel = TRUE,axes = c(1, 3), col.var = "#2E9FDF",
  col.ind = "#696969")
```



Jaén por otra parte es la provincia con el valor más alto en la componente 3 que refleja mayormente el censo Agrario Número de Explotaciones

g. Si tuviéramos que construir un índice que valore de forma conjunta el desarrollo económico de una provincia, como se podría construir utilizando una combinación lineal de todas las variables. ¿Cuál sería el valor de dicho índice en Madrid? ¿Cual sería su valor en Melilla?

El desarrollo económico de una provincia lo resume el componente 1, que no es una combinación lineal de todas las variables, pero si una combinación lineal de las variables que tienen una mayor correlación con el desarrollo económico.

```
ind<-get_pca_ind(fit)
knitr::kable(ind$coord, digits =3,caption = "Valores de los individuo
s en las Cp")
```

Valores de los individuo s en las Cp

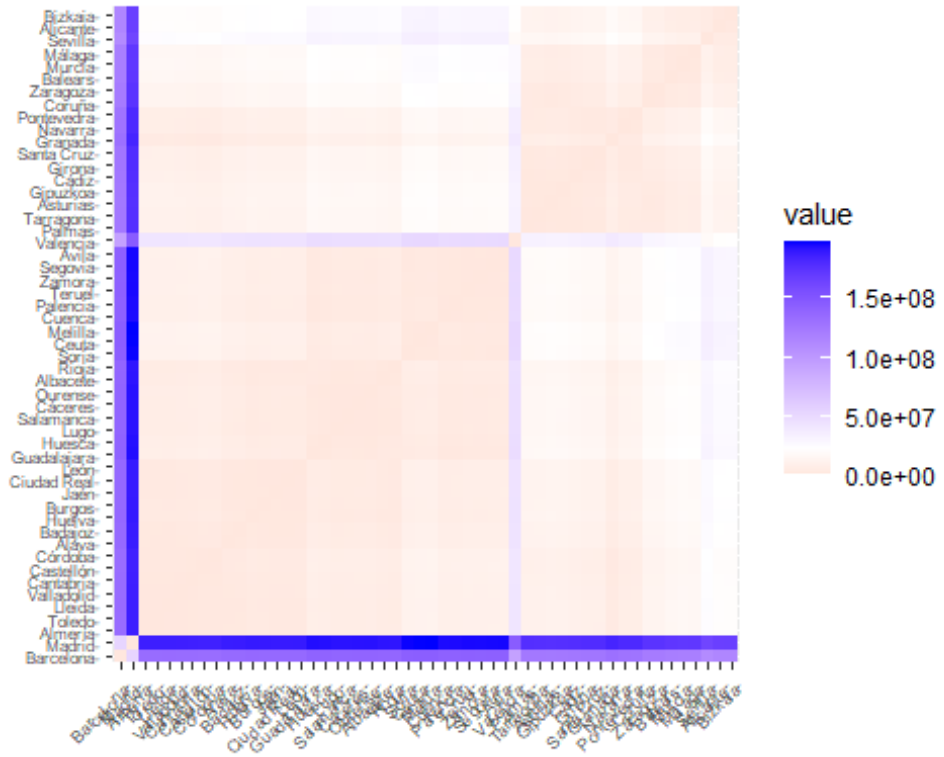
Provincia	Dim.1	Dim.2	Dim.3	Provincia	Dim.1	Dim.2	Dim.3
Albacete	-1.410	0.473	0.096	**León**	-1.464	-2.016	0.877
Alicante	3.384	0.540	1.919	**Lleida**	-0.835	-0.136	-1.472
Almería	-0.617	2.614	0.208	**Lugo**	-1.826	-2.785	0.871
Álava	-1.444	-0.001	-2.032	**Madrid**	16.778	-0.366	-0.849
Asturias	-0.204	-1.953	1.298	**Melilla**	-2.218	4.782	-1.905
Badajoz	-1.048	1.168	1.796	**Murcia**	1.522	1.442	0.580
Balears	1.526	0.260	-2.519	**Málaga**	2.006	1.325	0.869
Barcelona	13.683	-1.612	-0.867	**Navarra**	-0.653	0.078	-1.096
Bizkaia	0.576	-1.508	-1.180	**Ourense**	-1.965	-2.858	1.098
Burgos	-1.202	-1.550	-0.892	**Palencia**	-2.122	-1.951	-0.695
Cantabria	-0.849	-1.126	-0.594	**Palmas**	0.092	1.857	-0.330
Castellón	-0.690	0.821	0.518	**Pontevedra**	0.036	-0.607	0.052
Ceuta	-2.125	3.326	-1.811	**Rioja**	-1.383	-0.484	-1.354
Ciudad Real	-1.392	0.815	1.819	**Salamanca**	-1.612	-1.425	-0.121
Coruña	0.635	-1.442	0.759	**Santa Cruz**	-0.029	1.573	-0.126
Cuenca	-2.132	-0.569	1.048	**Segovia**	-1.931	-1.015	-1.110
Cáceres	-1.503	-0.180	1.269	**Sevilla**	1.948	1.775	0.712
Cádiz	0.128	1.771	0.369	**Soria**	-2.399	-1.857	-0.778
Córdoba	-0.594	0.811	1.292	**Tarragona**	0.175	1.040	-0.102
Gipuzkoa	-0.281	-1.485	-1.879	**Teruel**	-2.185	-1.082	-0.351
Girona	0.388	0.544	-1.387	**Toledo**	-0.461	1.449	0.812
Granada	-0.072	1.124	1.511	**Valencia**	4.770	0.360	2.961
Guadalajara	-1.408	1.542	-1.849	**Valladolid**	-1.007	-0.704	-1.052
Huelva	-1.265	1.168	0.013	**Zamora**	-2.287	-3.169	0.578
Huesca	-1.776	-0.984	-0.587	**Zaragoza**	0.115	-0.237	-0.156
Jaén	-1.221	1.424	3.407	**Ávila**	-2.152	-0.982	0.365

Para Madrid el valor de dicho indece es decir el valor de la componente 1 sería de 16.778 y el de Melilla -2.218

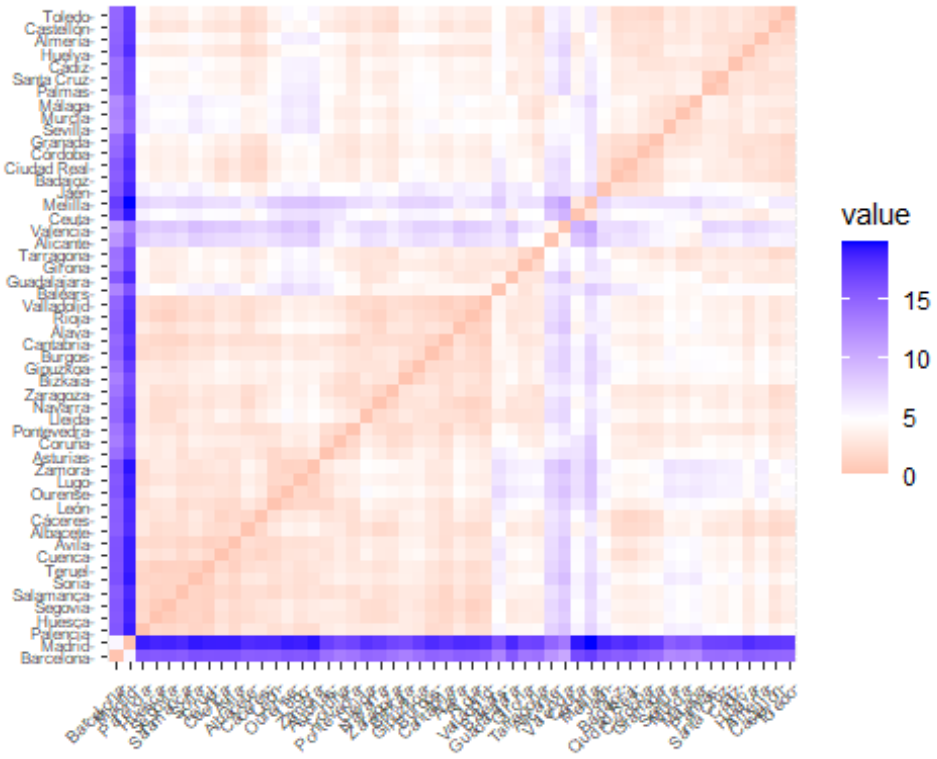
4. Representar un mapa de calor de la matriz de datos, estandarizado y sin estandarizar para ver si se detectan inicialmente grupos de provincias.

Mostramos el mapa de calor con los valores de distancias sin estandarizar

```
#Calculamos las distancias con los valores sin estandarizar
d <- dist(datos, method = "euclidean")
fviz_dist(d,lab_size = 6)
```




```
#ggheatmap(as.matrix(d), seriate="mean", cexRow=0.5, cexCol =0.5)

# Estandarizamos los dattos
datos_ST <- scale(datos)
#Calculamos las distancias con los valores estandarizados
d_st <- dist(datos_ST, method = "euclidean")
fviz_dist(d_st,lab_size = 6)
```



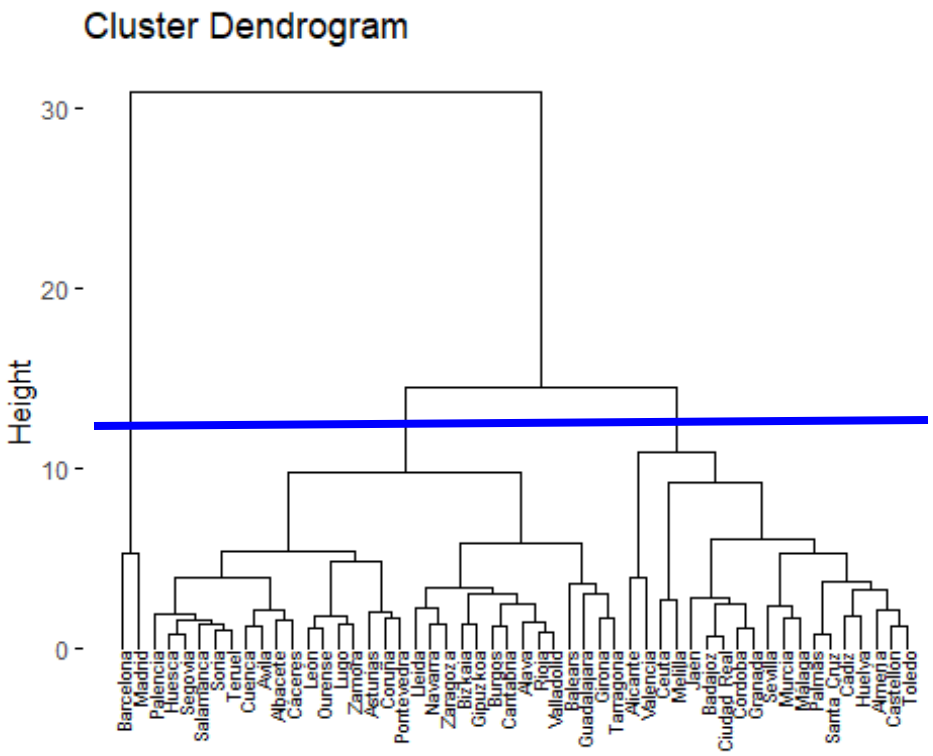
Puesto que el color es el valor de cada variable , el mismo color en una variable indica grupos.

Tenemos una primera aproximación de las provincias que tienen valores de las variables más parecidos, en este caso Madrid y Barcelona, y Ceuta, Melilla, Jaén y Alicante

5. Realizar un análisis Jerárquico de clusters para determinar si existen grupos de provincias con comportamiento similar.

a. A la vista del dendrograma ¿Cuántos clusters recomendarías?

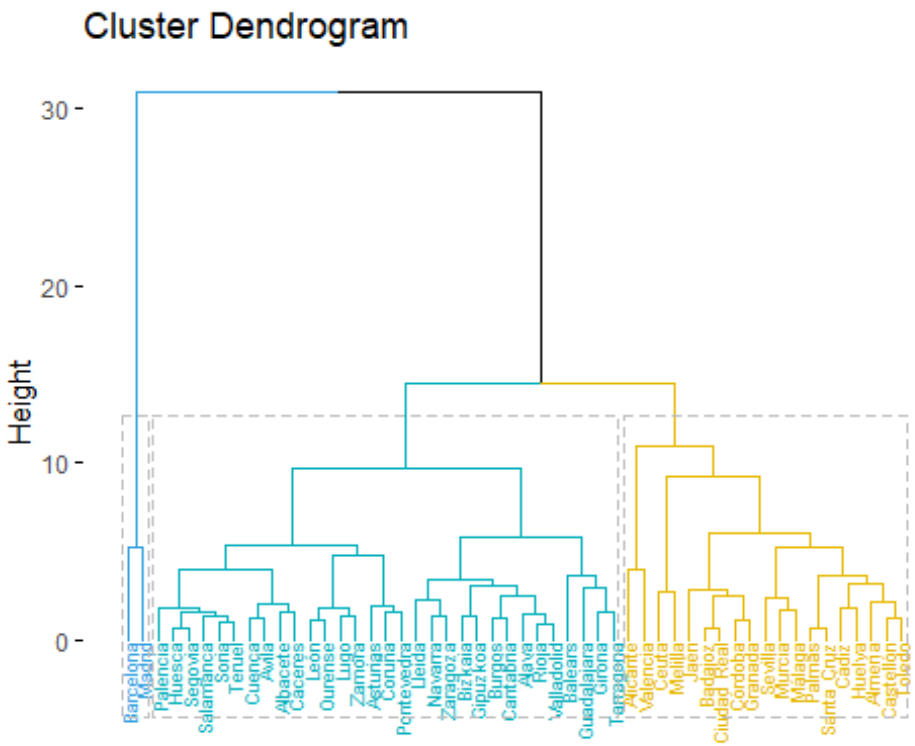
```
res.hc_st <- hclust(d_st, method="ward.D2")
fviz_dend(res.hc_st, cex = 0.5)
```



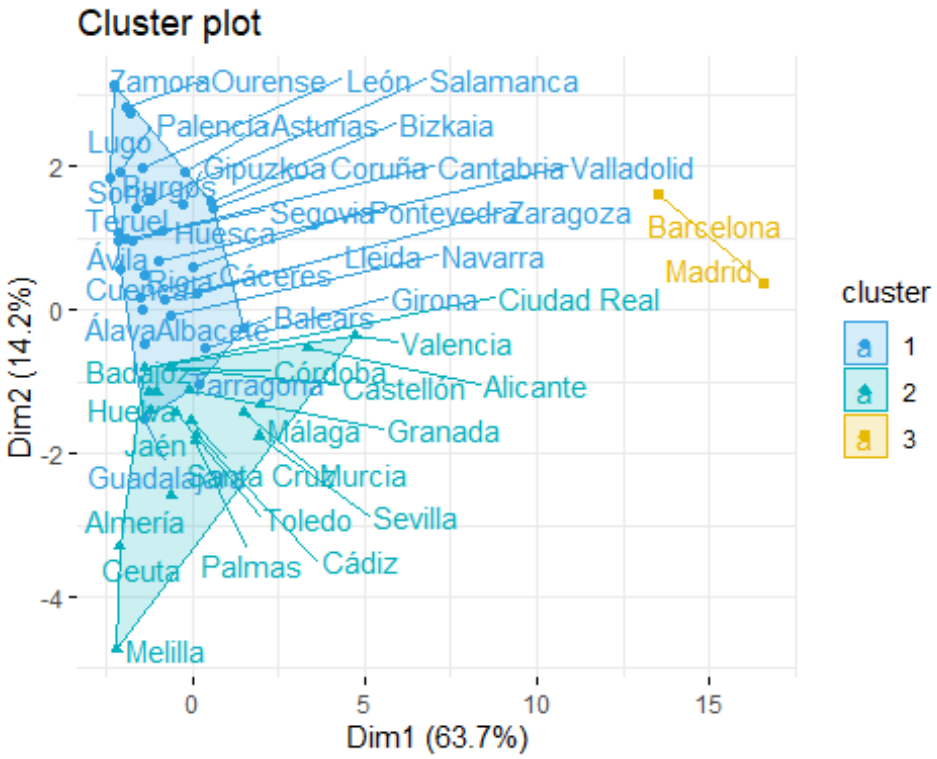
Observando la estructura del dendrograma podemos hacernos una idea de cuál podría ser el número más adecuado de cluster. Por ejemplo, en nuestro caso podría ser 3, como aparece en la imagen en donde hemos dibujado una línea.

b. Representar los individuos agrupados según el número de clusters elegido.

```
grp <- cutree(res.hc_st, k = 3)
fviz_dend(res.hc_st, k = 3, # Cut in three groups
 cex = 0.5, # label size
 k_colors = c("#2E9FDF", "#00AFBB", "#E7B800"),
 color_labels_by_k = TRUE, # color labels by groups
 rect = TRUE) # Add rectangle around groups
```



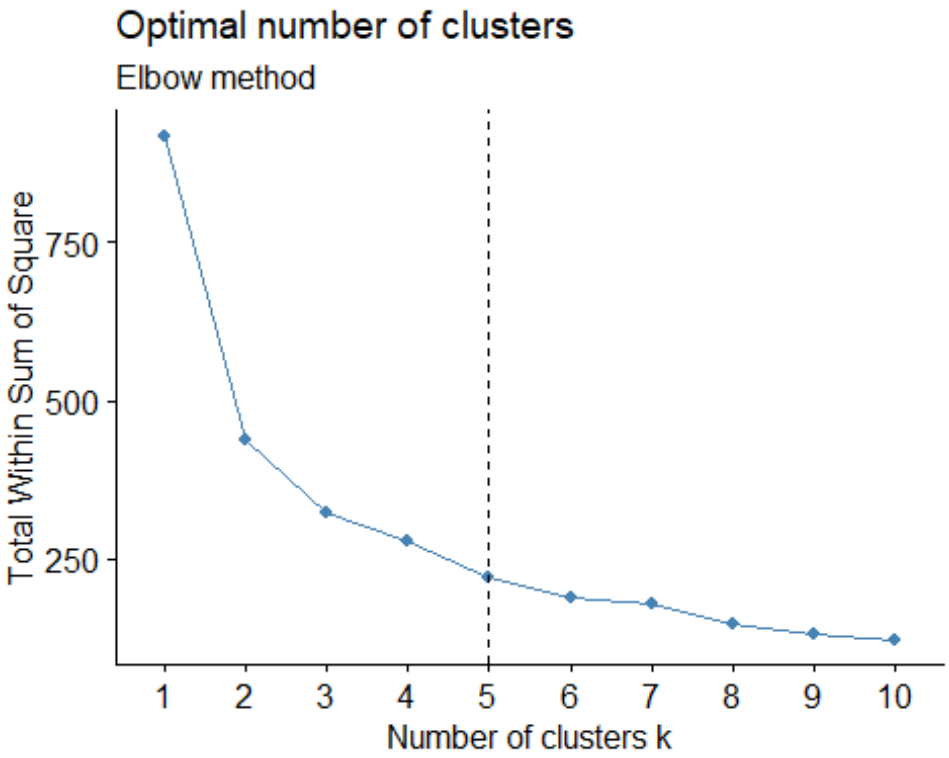

```
fviz_cluster(list(data = datos_ST, cluster = grp), palette = c("#2E9FDF",
"#00AFBB", "#E7B800"), ellipse.type = "convex", repel = TRUE,
show.clust.cent = FALSE, ggtheme = theme_minimal())
```



c. ¿Qué número óptimo de clusters nos indican los criterios Silhoutte y de Elbow?

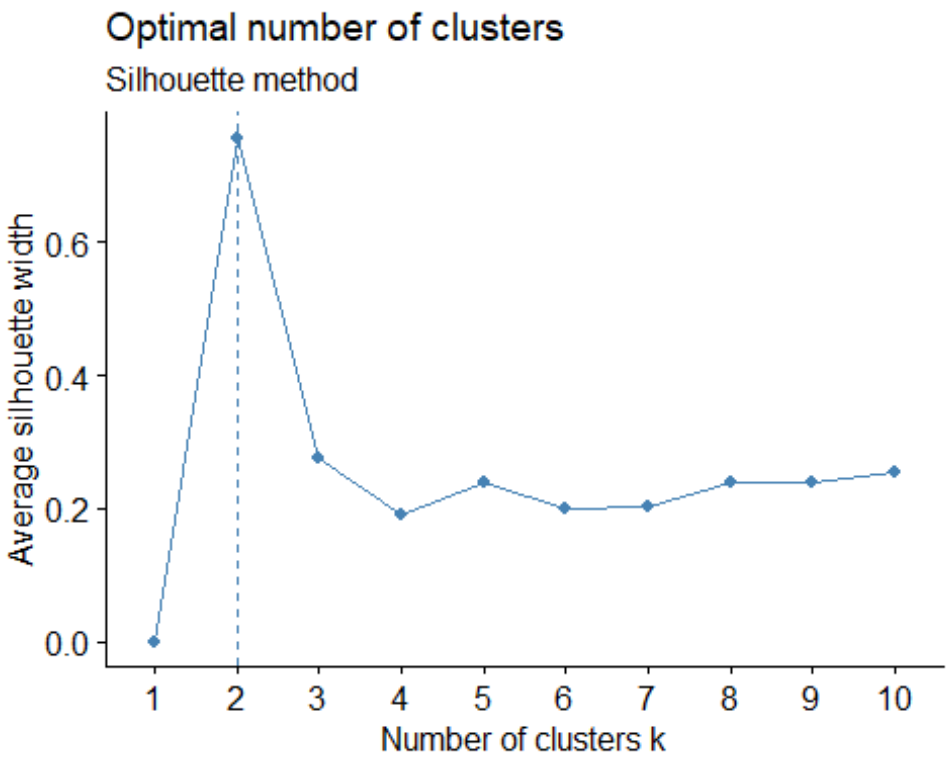
Método elbow

```
fviz_nbclust(datos_ST, kmeans, method = "wss") + geom_vline(xintercept
=5, linetype = 2)+ labs(subtitle = "Elbow method")
```



El criterio de Elbow elige como número óptimo de clusters aquel número de clusters en el que la Variabilidad total intra-clústeres ya no se reduce de forma significativa al aumentar uno más. Me recomienda 5 clusters como el número óptimo de clusters

```
fviz_nbclust(datos_ST, kmeans, method = "silhouette") +
 labs(subtitle = "Silhouette method")
```



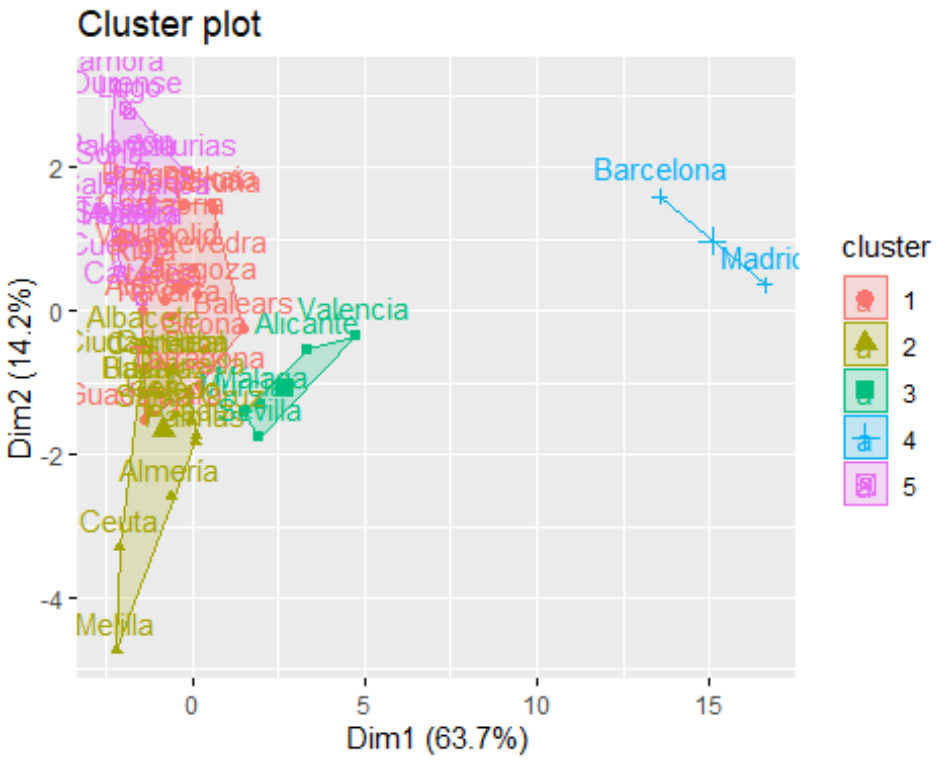
Me recomienda 2 clusters como el número óptimo, pero no veo mucho sentido en realizar sólo dos agrupaciones puesto que no nos daría mucha información, por lo que me quedaría con el recomendado con el criterio de elbow (5).

d. Con el número de clusters decidido en el apartado anterior realizar un agrupamiento no jerárquico.

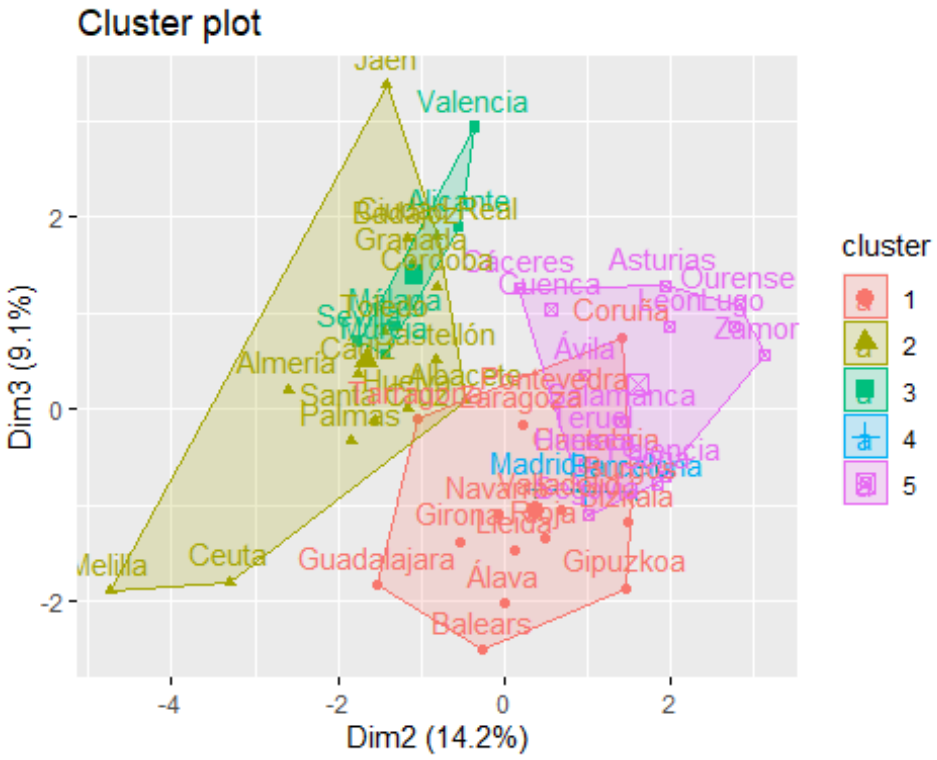
```
set.seed(1234)
km.res <- kmeans(datos_ST, 5)
```

i. Representar los clusters formados en los planos de las Componentes principales. Relacionar la posición de cada cluster en el plano con lo que representa cada componente principal.

```
fviz_cluster(km.res, datos_ST)
```




```
fviz_cluster(km.res, datos_ST, axes = c(2,3))
```



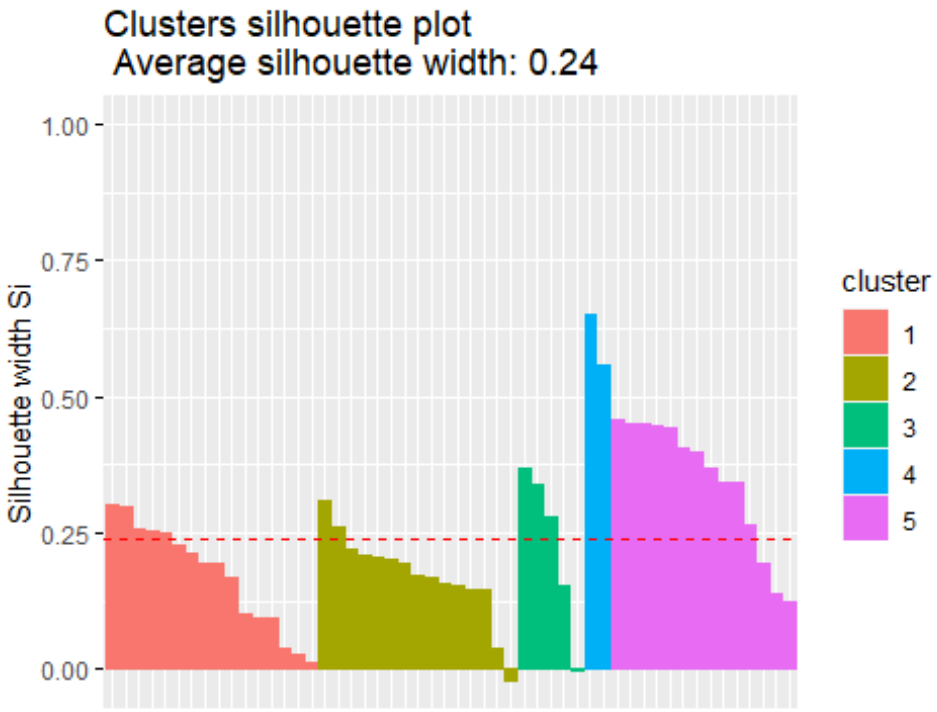
ii. Evaluar la calidad de los clusters

Silohuette es una medida de como de compactos son los clusters y cuanto de separados están unos de otros.

```
sil <- silhouette(km.res$cluster, dist(datos_ST))
rownames(sil) <- rownames(datos)
fviz_silhouette(sil)
```

```
##   cluster size ave.sil.width
## 1       1   16          0.17
## 2       2   15          0.17
## 3       3    5          0.23
## 4       4    2          0.60
## 5       5   14          0.34
```

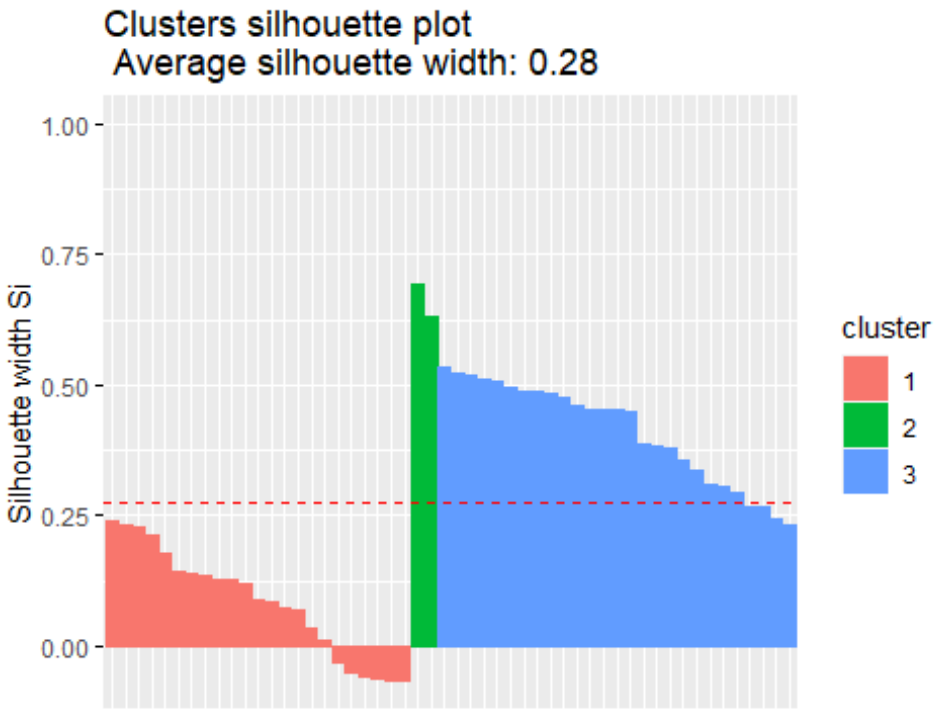
	cluster	neighbor	sil_width
Albacete	2	5	-0.02095427
Alicante	3	1	0.33921400
Almería	2	1	0.31029613
Álava	1	5	0.22812174
Asturias	5	1	0.26330217
Badajoz	2	5	0.26174926
Balears	1	3	0.25027123
Barcelona	4	3	0.55872598
Bizkaia	1	5	0.21354124
Burgos	1	5	0.03868150
Cantabria	1	5	0.02721981
Castellón	2	1	0.03869451
Ceuta	2	1	0.15779037
Ciudad Real	2	5	0.17230142
Coruña	1	5	0.01414160
Cuenca	5	2	0.36818190
Cáceres	5	2	0.13852320
Cádiz	2	1	0.19334458
Córdoba	2	1	0.21954407
Gipuzkoa	1	5	0.25307731
Girona	1	2	0.30039754
Granada	2	1	0.20947522
Guadalajara	1	2	0.10330459
Huelva	2	1	0.14679175
Huesca	5	1	0.19371363
Jaén	2	5	0.20039068
León	5	1	0.45171045
Lleida	1	5	0.29827195
Lugo	5	1	0.44650600
Madrid	4	3	0.64909131
Melilla	2	1	0.15245784
Murcia	3	2	0.15378142
Málaga	3	2	0.27821418
Navarra	1	5	0.25870017
Ourense	5	1	0.45646137
Palencia	5	1	0.34284470
Palmas	2	1	0.16860980
Pontevedra	1	5	0.09476070
Rioja	1	5	0.16856483
Salamanca	5	1	0.34226221
Santa Cruz	2	1	0.14558058
Segovia	5	1	0.12551947
Sevilla	3	2	-0.00251291
Soria	5	1	0.40454914
Tarragona	1	2	0.09456422
Teruel	5	1	0.39939122
Toledo	2	1	0.20544284
Valencia	3	1	0.36746355
Valladolid	1	5	0.19469075
Zamora	5	1	0.44197619
Zaragoza	1	5	0.19534258
Ávila	5	1	0.44944150



Valores negativos indican que esa observación no está bien clasificada, por lo que voy a probar otras cantidades de clusters por si hay alguno mejor, aunque este caso sólo hay un valor negativo para Albacete y es muy bajo.

```
km.res3 <- kmeans(datos_ST, 3)
sil3 <- silhouette(km.res3$cluster, dist(datos_ST))
rownames(sil3) <- rownames(datos)
fviz_silhouette(sil3)
```

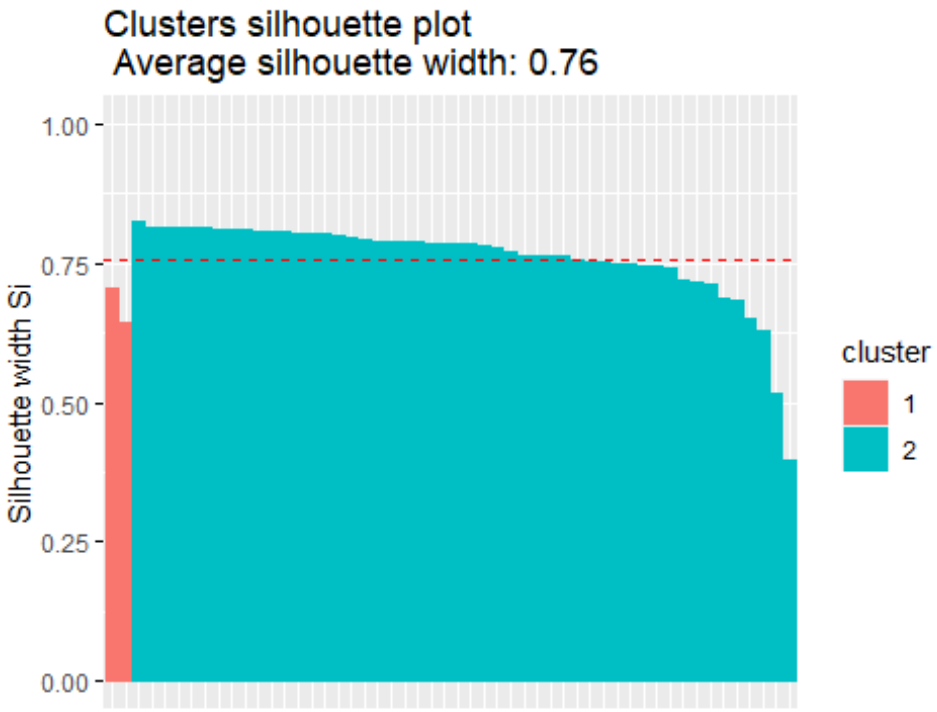
```
##   cluster size ave.sil.width
## 1       1   23          0.08
## 2       2    2          0.66
## 3       3   27          0.41
```



Ahora hay más valores negativos por lo que aunque tenga un valor medio de Silhoutte mejor nos quedamos con 5

```
km.res2 <- kmeans(datos_ST, 2)
sil2 <- silhouette(km.res2$cluster, dist(datos_ST))
rownames(sil2) <- rownames(datos)
fviz_silhouette(sil2)
```

```
##   cluster size ave.sil.width
## 1       1    2          0.67
## 2       2   50          0.76
```



Para dos agrupaciones si que mejora, y ya no tenemos valores negativos, pero por lo comentado anteriormente no es muy interesante hacer sólo dos agrupaciones de provincias.

e. Explicar las provincias que forman cada uno de los clusters y comentar cuales son las características socioeconómicas que las hacen pertenecer a dicho cluster.

```
ordenado<-sort(km.res$cluster)
ordenado

EsT_Clus<-aggregate(datos, by=list(km.res$cluster),mean)
knitr::kable(EsT_Clus, digits =2,caption = "Estadísticos de los
clusters")
```

Provincia	Cluster	Provincia2	Cluster2
Álava	1	Alicante	3
Balears	1	Murcia	3
Bizkaia	1	Málaga	3
Burgos	1	Sevilla	3
Cantabria	1	Valencia	3
Coruña	1	Barcelona	4
Gipuzkoa	1	Madrid	4

Girona	1	Asturias	5
Guadalajara	1	Cuenca	5
Lleida	1	Cáceres	5
Navarra	1	Huesca	5
Pontevedra	1	León	5
Rioja	1	Lugo	5
Tarragona	1	Ourense	5
Valladolid	1	Palencia	5
Zaragoza	1	Salamanca	5
Albacete	2	Segovia	5
Almería	2	Soria	5
Badajoz	2	Teruel	5
Castellón	2	Zamora	5
Ceuta	2	Ávila	5
Ciudad Real	2		
Cádiz	2		
Córdoba	2		
Granada	2		
Huelva	2		
Jaén	2		
Melilla	2		
Palmas	2		
Santa Cruz	2		
Toledo	2		

Estadísticos de los clusters

Group.1	Poblacion	Mortalidad	Natalidad	IPC	NumEmpresas	Industria	Construccion	PIB	CANE
1	689452.6	8.97	8.91	102.91	47950.69	3259.56	7048.00	17092150	13214.81
2	667162.8	7.99	10.09	101.58	38314.87	2500.93	4507.07	11676712	24147.80
3	1889495.4	7.75	9.75	102.43	122866.60	7704.60	14389.00	34927844	36462.20
4	5989112.0	7.46	9.94	103.73	474865.50	25012.00	55205.50	172165306	9156.00
5	307639.7	12.18	6.94	102.31	20062.86	1413.57	3079.93	6193759	15394.21

Group.1	CTH	Infor	AFS	APT	TasaActividad	TasaParo	Ocupados	TVF	VF
1	18535.50	674.50	1022.50	7759.56	59.57	15.83	285.29	380657.9	58739.12
2	16988.93	412.20	875.47	5778.47	58.33	29.53	225.63	358378.8	54634.47
3	49801.60	1839.40	2955.80	21007.80	59.52	25.81	685.20	1056591.8	182003.60
4	157055.00	15096.00	10593.00	105424.00	62.86	16.75	2542.35	2748888.0	156678.50
5	8573.07	178.14	443.93	2690.36	54.02	17.30	113.49	211548.5	49917.71

Comenzamos hablando del clúster número uno, económicamente representa a un grupo que se encuentra en la mitad de los otros clusters, no tiene muchas empresas ni mucha población, pero no es la peor agrupación. Su tasa de mortalidad es alta, no las más alta, pero representa a una población envejecida. Sus provincias son Álava, Balears,Bizkaia,Burgos,Cantabria,Coruña,Gipuzkoa,Girona,Guadalajara,Lleida,Navarra, Pontevedra,Rioja,Tarragona,Valladolid y Zaragoza.

El clúster número 2 representa a un grupo que económicamente de los peores y con la mayor tasa de paro. Además de este grupo podemos destacar que tiene la tasa de natalidad más alta.

Clúster número 3, económicamente el segundo mejor grupo, en número de empresas se encuentra en segundo lugar, aun así, tiene una tasa de paro alta. También tiene una población alta y el mayor censo agrario.

El clúster 4, representa a las ciudades con mayor población , aquellas que económicamente van mejor, con mayor número de empresas, mayor PIB y menor tasa de paro, estamos hablando las provincias más fuertes que son Madrid y Barcelona.

Por último, el clúster 5, representa a las provincias con una población envejecida, aquellas que tienen una mayor tasa de mortalidad y menor tasa de natalidad. Además son las provincias con menor número de empresas y poca tasa de actividad

EJERCICIO DE EVALUACIÓN II

ANÁLISIS YPREDICCIÓN DE SERIES TEMPORALES

1. Introducción: Presentación de la serie a analizar.

Después de mucho buscar en distintas fuentes, encontré en el INE (https://www.ine.es/consul/serie.do?s=168-1510&c=2&nult=50) datos sobre compraventa de viviendas totales en España por mes, desde 2007 hasta 2019.

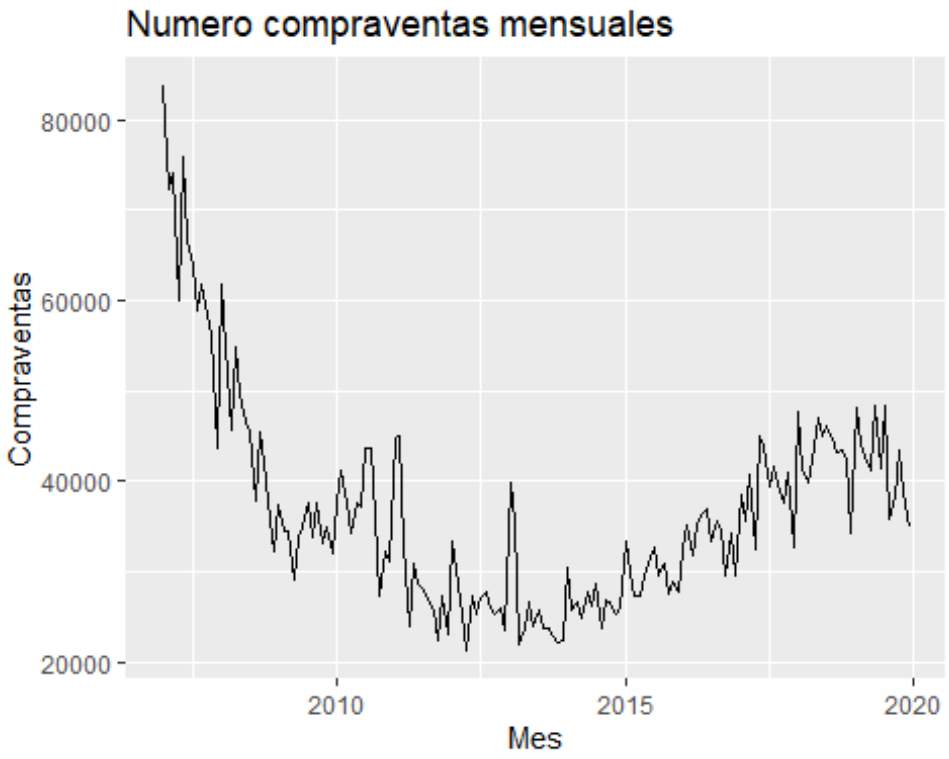
2. Representación gráfica y descomposición estacional (si tuviera comportamiento estacional).

Primero cargo las librerías que voy a utilizar

```
library(readxl)
library(zoo)
library(forecast)
library(ggplot2)
```

Para representarla cargo los datos del excel, creo la serie temporal con la función ts y por último utilizo autoplot para mostrarla.

```
compraventaViviendas <- read_excel("./compraventaViviendas.xlsx")
viv_seq <- ts(compraventaViviendas[,2], start=c(2007,1), frequency=12)
autoplot(viv_seq) + ggtitle("Numero compraventas mensuales") +
xlab("Mes") +    ylab("Compraventas")
```



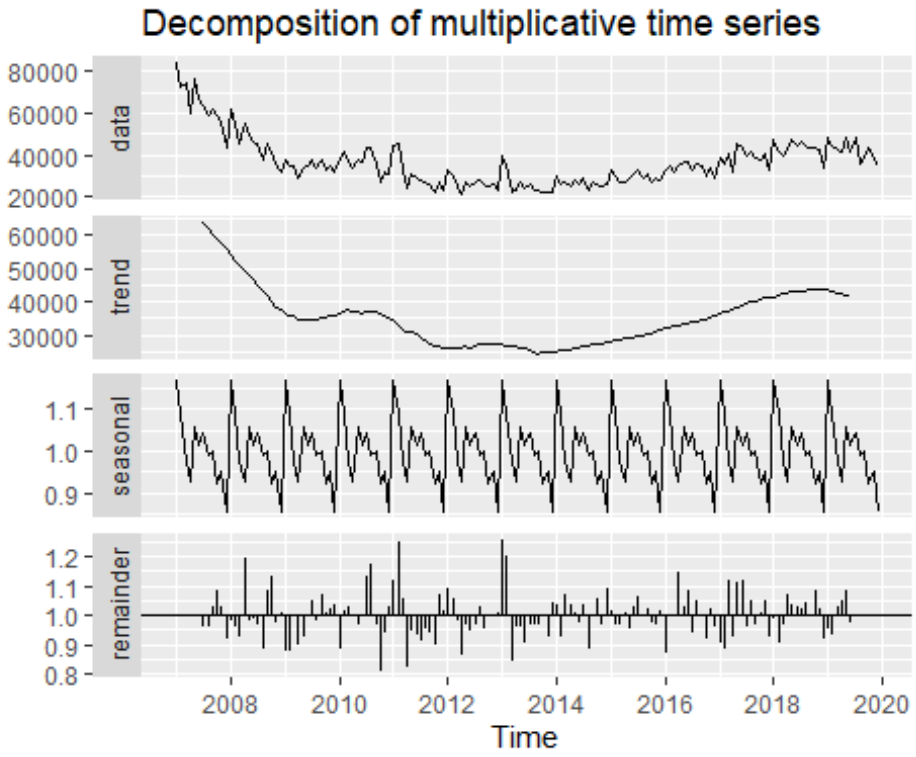
```
viv_seq_dec <- decompose(viv_seq,type=c("multiplicative"))

knitr::kable(viv_seq_dec$figure,    digits =2,caption = "Coef
Estacionalidad")
```

Coef Estacionalidad

x
1.17
1.09
0.98
0.93
1.06
1.01
1.04
0.99
1.00
0.92
0.95
0.86

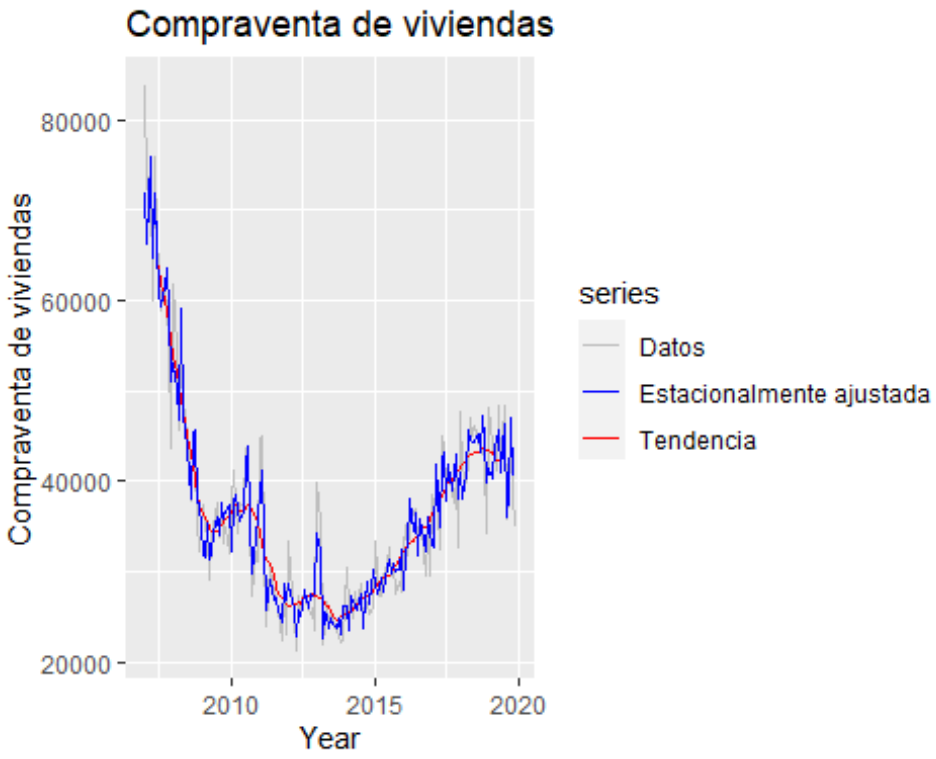
Vemos como en enero hay un 17% más de compraventa que la media del año, por eso hemos elegido el tipo mutiplicativo al hacer la descomposición estacional, para poder comparar los coeficientes.

```
viv_seq_dec <- decompose(viv_seq,type=c("multiplicative"))
autoplot(viv_seq_dec)
```



Se puede ver el impacto de la crisis en la compraventa de viviendas, a partir de 2008 baja y es en 2015 cuando empieza a ascender.

```
autoplot(viv_seq, series="Datos") +
autolayer(trendcycle(viv_seq_dec), series="Tendencia") +
autolayer(seasadj(viv_seq_dec), series="Estacionalmente ajustada") +
xlab("Year") + ylab("Compraventa de viviendas") +
ggtitle("Compraventa de viviendas") +
scale_colour_manual(values=c("gray","blue","red"),
breaks=c("Datos","Estacionalmente ajustada","Tendencia"))
```



3. Para comprobar la eficacia de los métodos de predicción que vamos a hacer en los siguientes apartados reservamos los últimos datos observados (un periodo en las series estacionales o aproximadamente 10 observaciones) para comparar con las predicciones realizadas por cada uno de los métodos. Luego ajustamos los modelos sobre la serie sin esos últimos datos en los siguientes apartados.

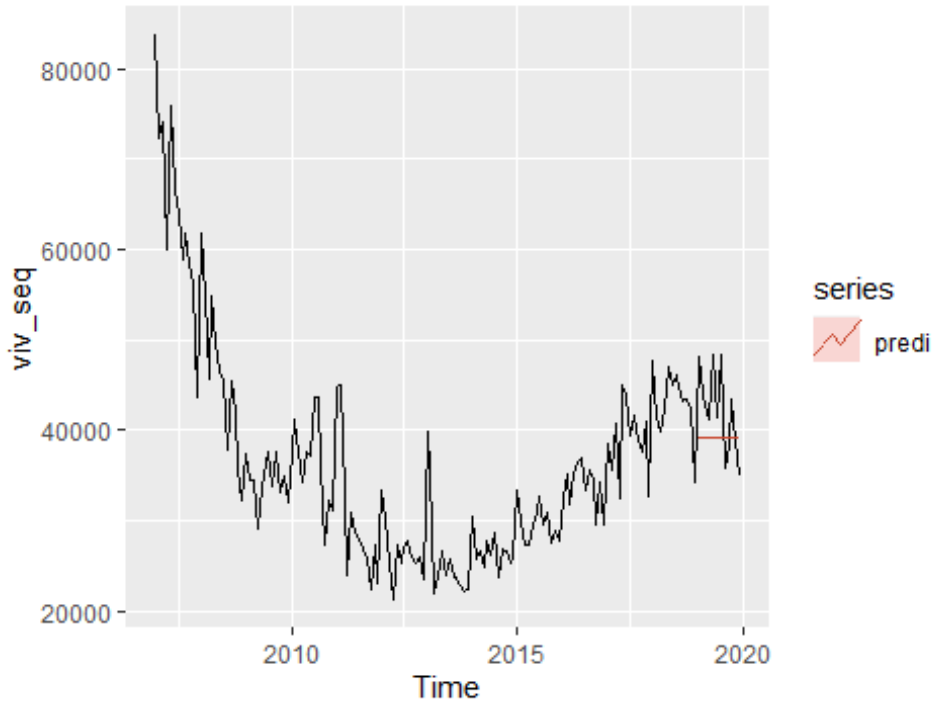
```
viv_train <- window(viv_seq, end=c(2018,12))
```

4. Encontrar el modelo de suavizado exponencial más adecuado. Para dicho modelo, representar gráficamente la serie observada y la suavizada con las predicciones para un periodo que se considere adecuado. (2)

Modelo de alisado simple

Empiezo utilizando el modelo de alisado simple, aunque sé que no va a ser el mejor la predicción permanece constante para todos los meses, y no se utiliza.

```
suavizado_simple =  ses(viv_train, h=12)
autoplot(viv_seq) +
  autolayer(forecast(suavizado_simple), series="predi", PI=FALSE)
```



El método de suavizado no nos proporciona buenos resultados cuando los datos tienen una tendencia que varía rápidamente.

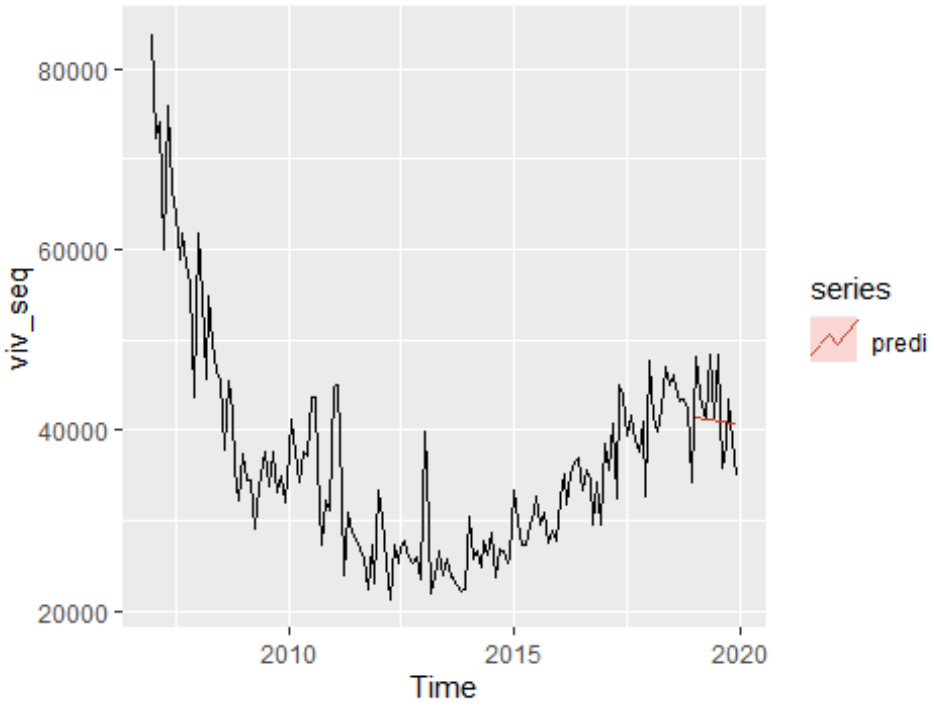
Método de alisado doble de Holt

Este método supone que la tendencia es lineal pero su pendiente va variando en el tiempo

```
holt <- holt(viv_train, h=12)
print(holt)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2019       41395.70 35103.01 47688.40 31771.85 51019.56
## Feb 2019       41335.17 34685.61 47984.72 31165.55 51504.79
## Mar 2019       41274.63 34223.31 48325.95 30490.57 52058.69
## Apr 2019       41214.09 33718.15 48710.03 29750.04 52678.15
## May 2019       41153.56 33172.42 49134.69 28947.47 53359.64
## Jun 2019       41093.02 32588.51 49597.53 28086.49 54099.55
## Jul 2019       41032.48 31968.73 50096.24 27170.67 54894.30
## Aug 2019       40971.95 31315.30 50628.59 26203.38 55740.51
## Sep 2019       40911.41 30630.26 51192.56 25187.75 56635.07
## Oct 2019       40850.87 29915.48 51786.27 24126.63 57575.12
## Nov 2019       40790.33 29172.63 52408.04 23022.59 58558.08
## Dec 2019       40729.80 28403.23 53056.37 21877.94 59581.66
```

```
autoplot(viv_seq) +
  autolayer(forecast(holt), series="predi", PI=FALSE)
```



Este método da malas predicciones, a simple vista no sabría decir si es mejor que el de suavizado simple. Voy a utilizar la función accuracy para comparar los resultados.

```
accuracy(holt)
```

```
##                     ME     RMSE      MAE          MPE     MAPE      MASE
## Training set 407.015 4841.541 3574.157 -0.01874641 10.09799 0.5549359
##                      ACF1
## Training set 0.04862283
```

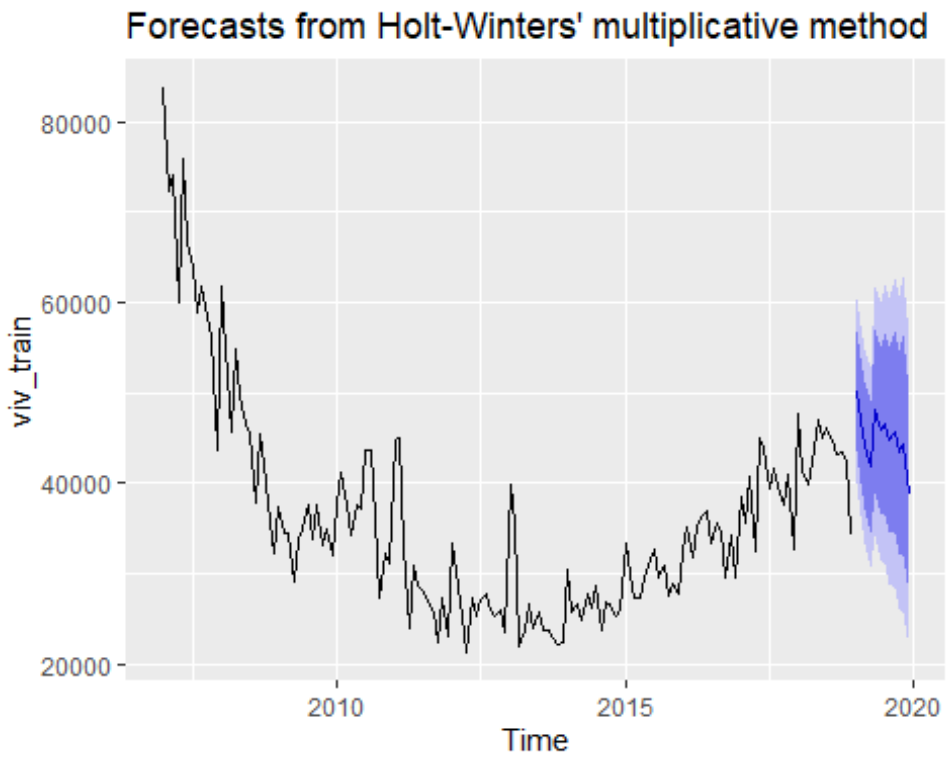
```
accuracy(suavizado_simple)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -571.9613 4992.172 3662.146 -2.448583 10.19955 0.5685973
##                       ACF1
## Training set -0.008018678
```

Si nos fijamos por ejemplo en el RMSE (raíz del error cuadrático medio), podemos ver que el holt es algo mejor.

Método de suavizado para series con estacionalidad: Holt-Winters.

```
viv_holt_winters <- hw(viv_train, h =12, seasonal = "multiplicative",
level = c(80,95))
autoplot(viv_holt_winters)
```



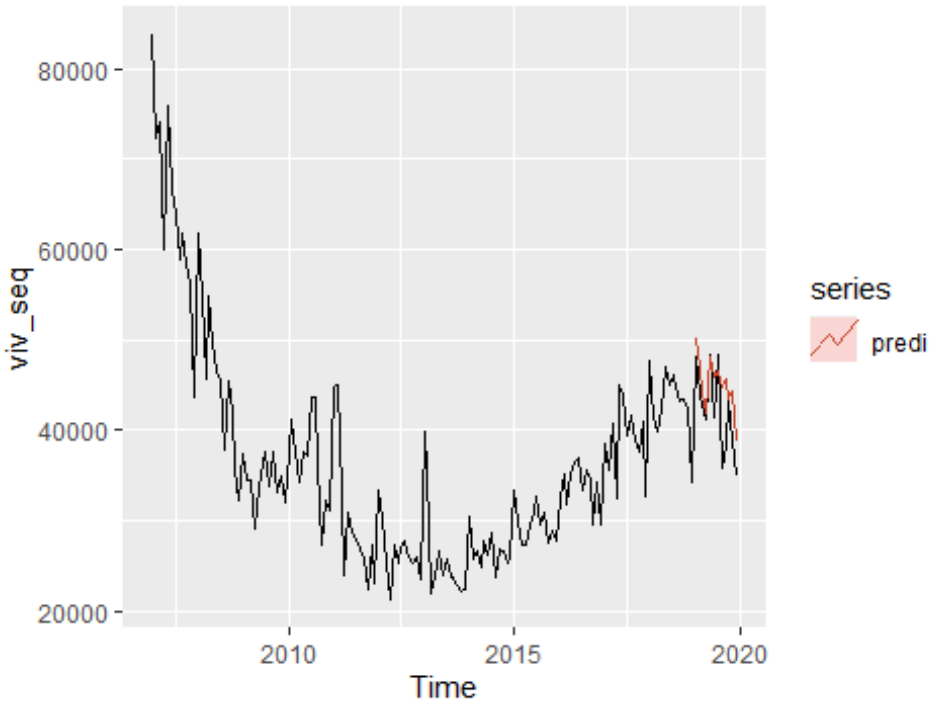

```
accuracy(viv_holt_winters)
```

```
##                     ME     RMSE      MAE       MPE   MAPE      MASE
ACF1
## Training set 309.5515 3420.185 2629.631 0.5470577 7.4668 0.4082856
0.02139528
```

```
print(viv_holt_winters)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2019       50261.08 43586.46 56935.70 40053.13 60469.03
## Feb 2019       46979.64 40097.59 53861.70 36454.44 57504.84
## Mar 2019       44140.20 37067.38 51213.02 33323.26 54957.14
## Apr 2019       41762.16 34492.75 49031.58 30644.55 52879.78
## May 2019       48059.76 39024.13 57095.38 34240.96 61878.55
## Jun 2019       45799.36 36544.52 55054.19 31645.31 59953.41
## Jul 2019       46598.29 36520.30 56676.28 31185.33 62011.24
## Aug 2019       44725.77 34411.27 55040.27 28951.11 60500.43
## Sep 2019       45611.26 34431.67 56790.85 28513.55 62708.97
## Oct 2019       43374.66 32108.28 54641.03 26144.23 60605.09
## Nov 2019       44285.85 32127.74 56443.96 25691.62 62880.08
## Dec 2019       38726.66 27516.00 49937.33 21581.43 55871.89
```

```
autoplot(viv_seq) +
  autolayer(forecast(viv_holt_winters), series="predi", PI=FALSE)
```

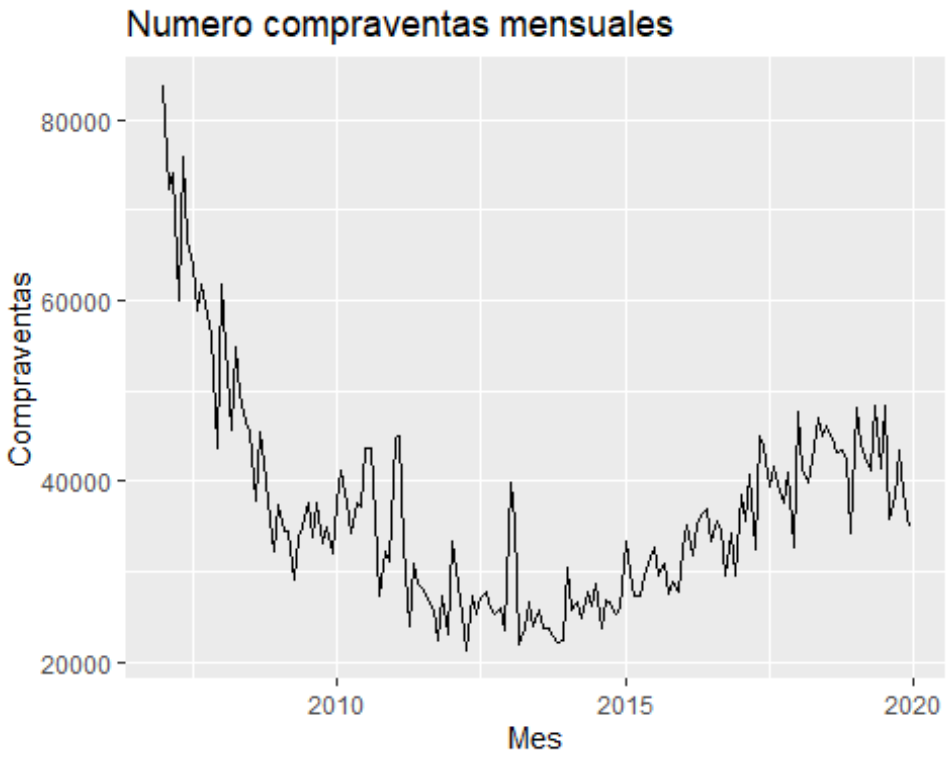


Como podemos ver en el gráfico el método de suavizado de Holt-Winters para series con estacionalidad proporciona unas predicciones muy buenas dado que sus intervalos de predicción son muy ajustados. Si nos fijamos en el MAPE podemos comprobar que lo que se ve en el gráfico es cierto se ajusta mucho mejor que los anteriores 7.4668

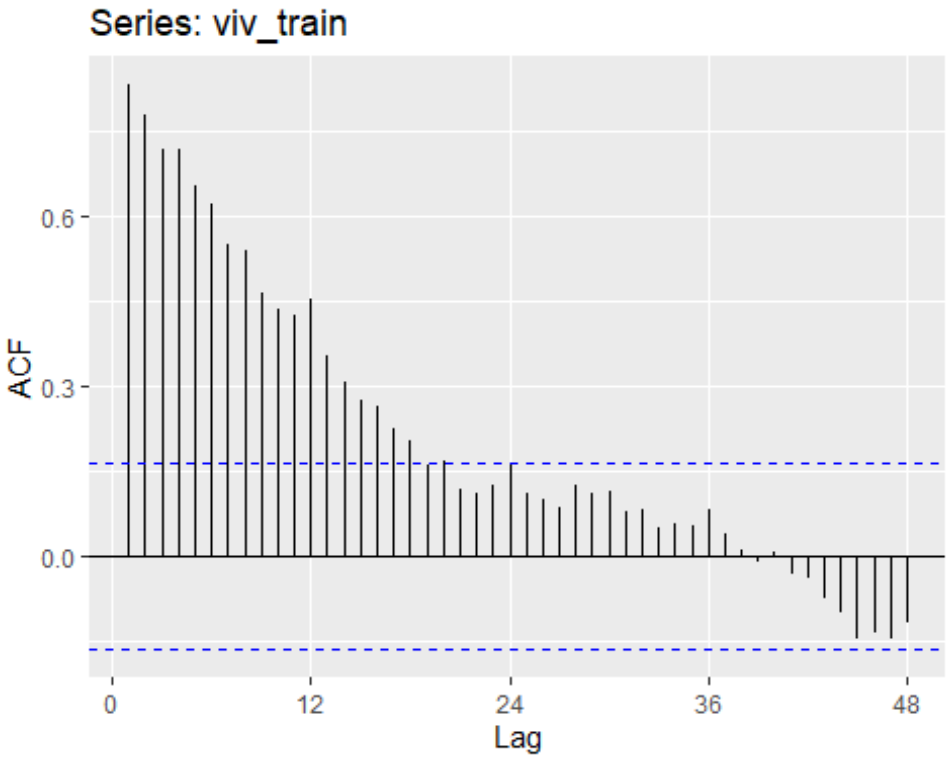
5. Representar la serie y los correlogramas. Decidir qué modelo puede ser ajustado. Ajustar el modelo adecuado comprobando que sus residuales están incorrelados. (Sintaxis, tablas de los parámetros estimados y gráficos) (3)

Represento la serie de nuevo:

```
autoplot(viv_seq) + ggtitle("Numero compraventas mensuales") +
xlab("Mes") +    ylab("Compraventas")
```

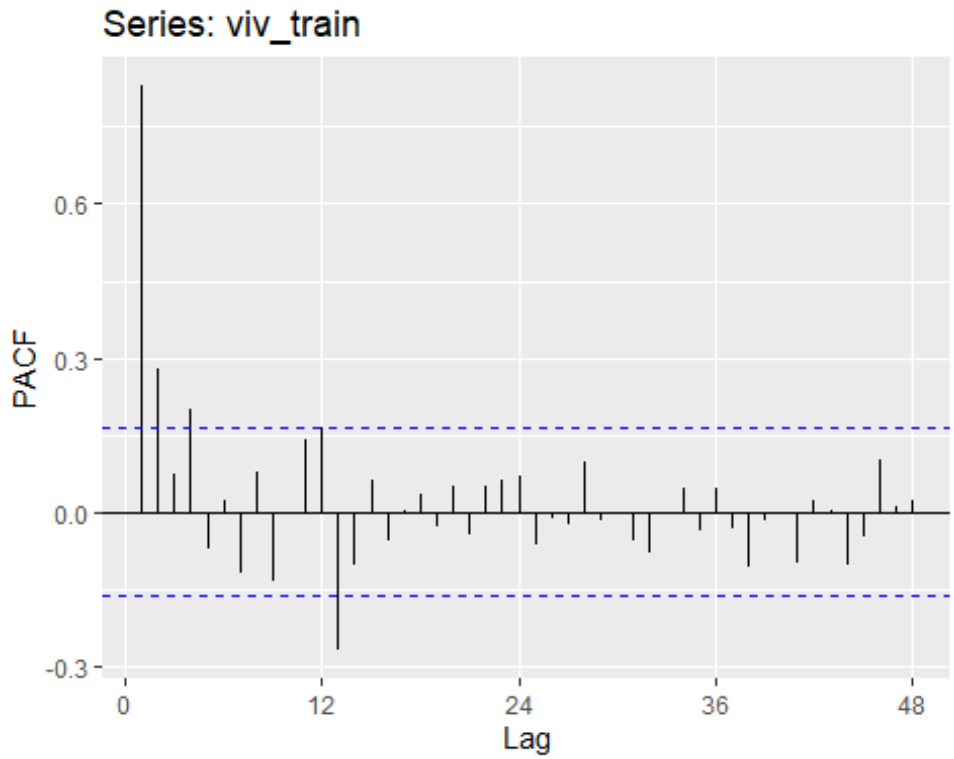



```
ggAcf(viv_train, lag=48)
```



Nuestro correlograma decrece de forma exponencial por lo que podemos decir que nuestra serie es estacionaria. Y cada 12 meses tiene una correlación alta.

```
ggPacf(viv_train, lag=48)
```



La función de correlación parcial nos va a servir para determinar que instantes tienen correlación fuerte porque no tienen el efecto acumulado que tiene la función de correlación simple. Por lo que analizando el correlograma vemos como los instantes 1, 2 y 4 son los que tienen correlaciones más significativas. Cuando hay correlaciones significativas en el primer o segundo desfases, seguidas de correlaciones que no son significativas, nos indica que existe un término autorregresivo en los datos. El número de correlaciones significativas indica el orden del término autorregresivo.

Utilizamos el modelo arima estacional que se modeliza como ARIMA(p,d,q)x(P,D,Q)S.

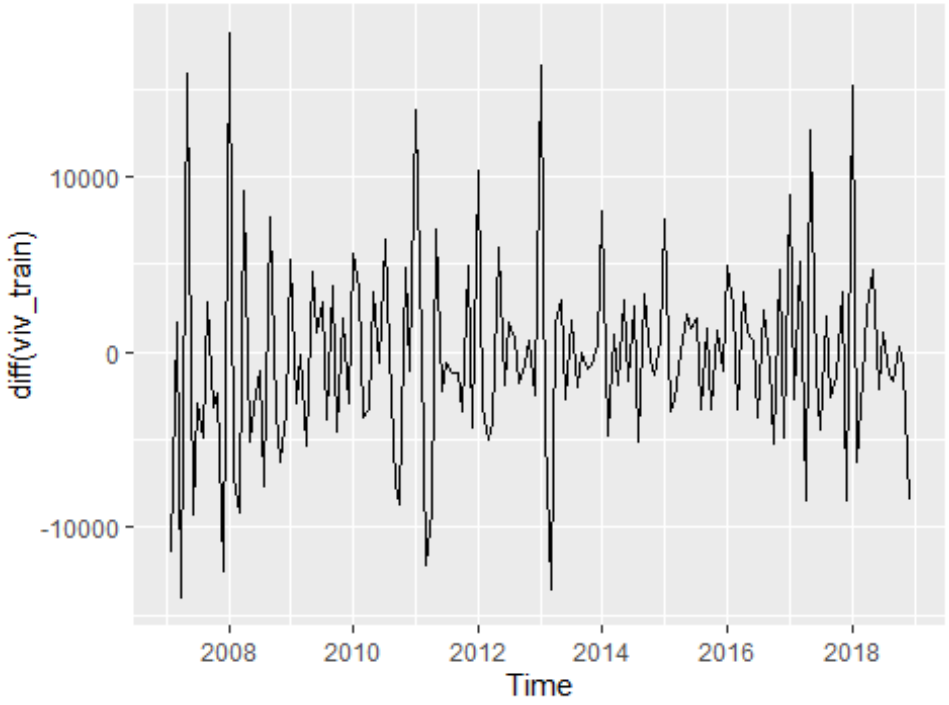
- p es el orden del polinomio auto-regresivo
- d es el numero de diferenciaciones
- q es el orden del polinomio de medias móviles
- P orden del polinomio auto-regresivo estacional
- D es el numero de diferenciaciones estacional
- Q es el orden del polinomio de medias móviles estacional
- S = espacio de tiempo de repetición del patron estacional

Los ajustes los haré de forma manual, automática y logarítmica, y los compararé.

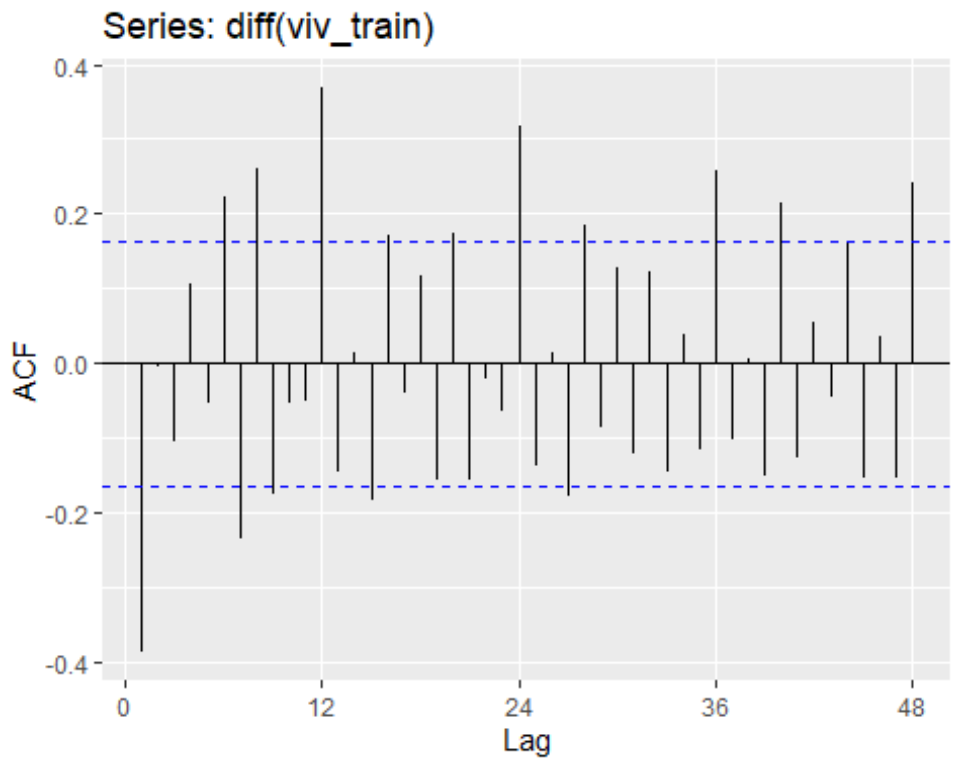
Ajuste manual

Puesto que el gráfico de la serie no tiene media costante por que la serie tiene tendencia es necesario hacer una diferenciación de orden 1.

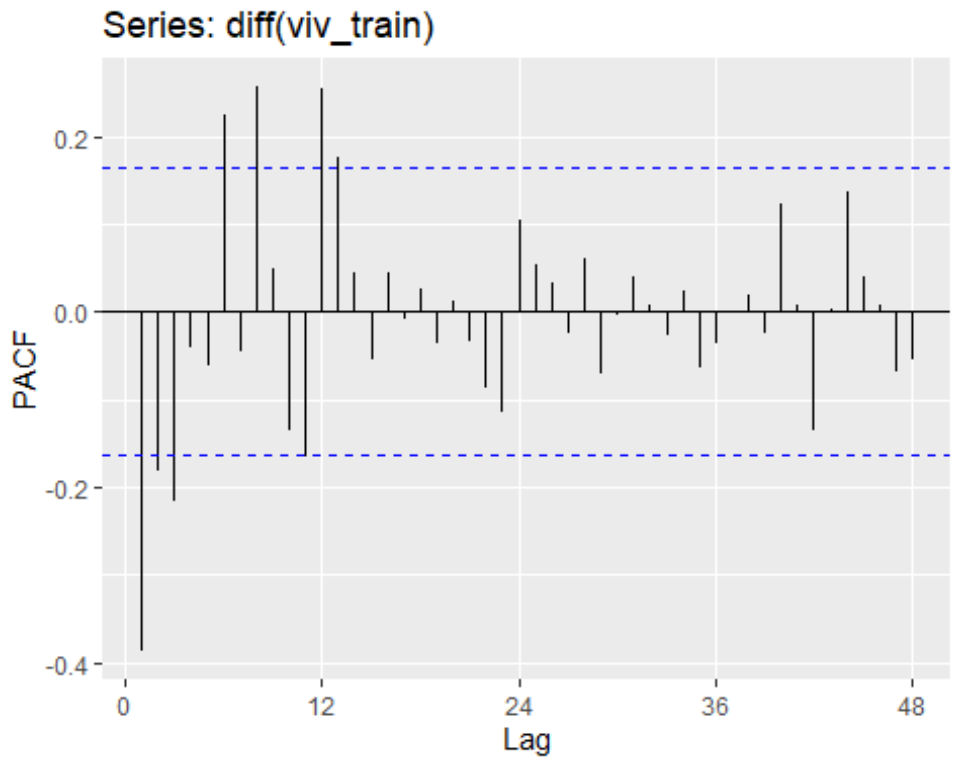
```
autoplot(diff(viv_train))
```




```
ggAcf(diff(viv_train), lag=48)
```

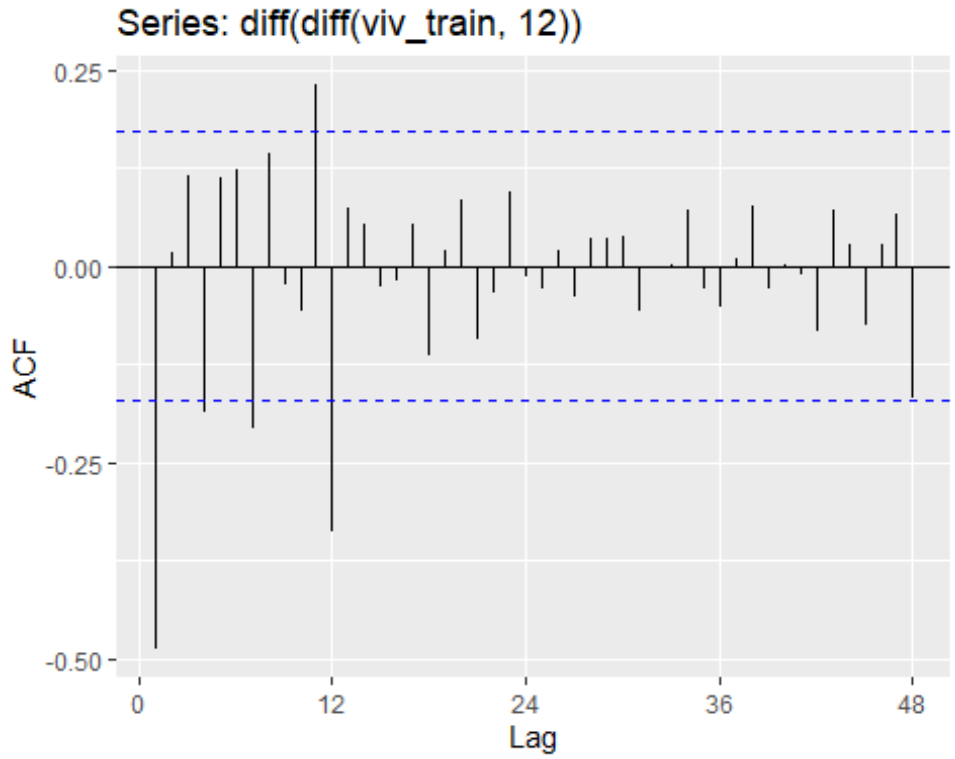



```r
ggPacf(diff(viv_train), lag=48)
```

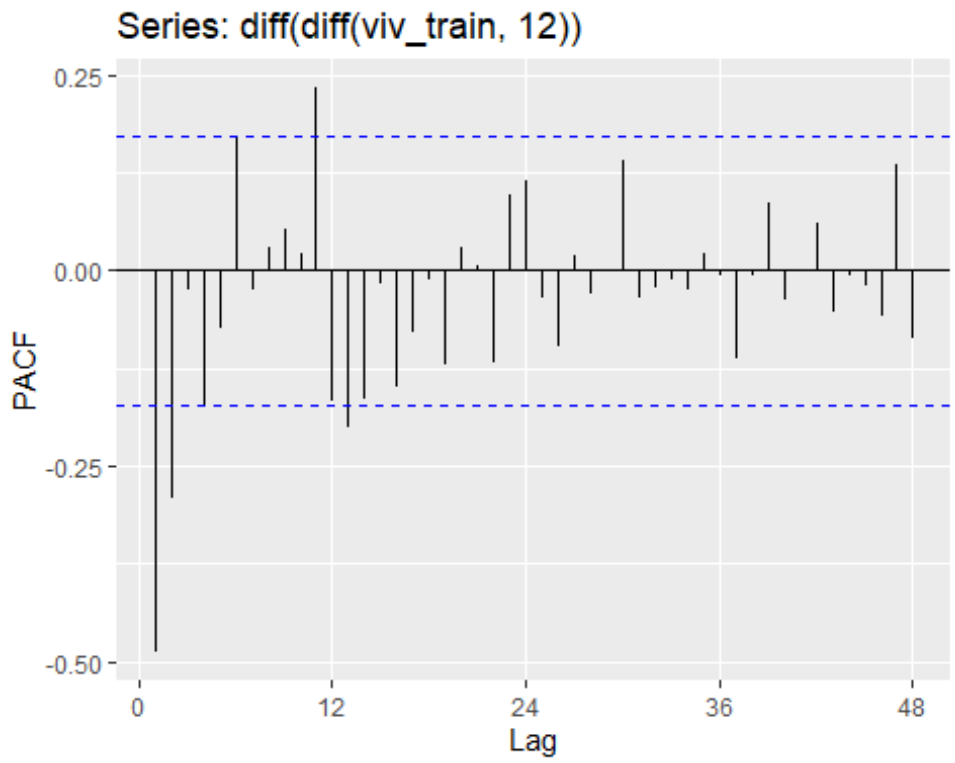


Hemos conseguido eliminar la tendencia de la serie y además los autocorrelogramas ya decrecen de forma más rápida. Aún así, los retardos multiplos de 12 siguen tinendo una correlación muy alta por lo que es necesario realizar una diferenciación de orden 12.

```
ggAcf(diff(diff(viv_train,12)), lag=48)
```




```
ggPacf(diff(diff(viv_train,12)), lag=48)
```



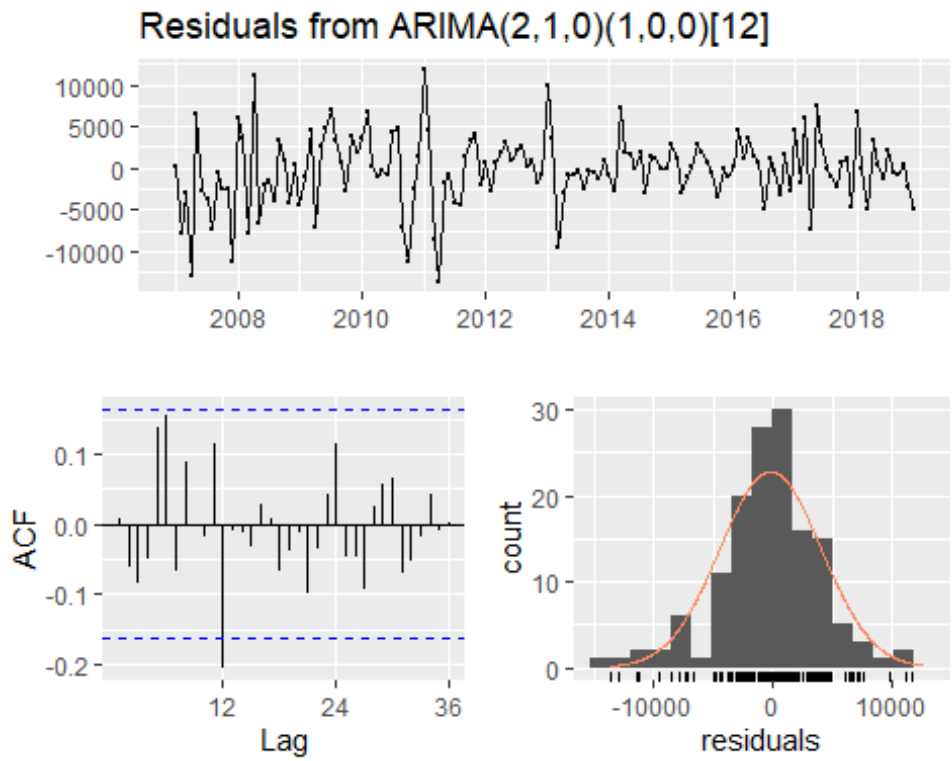
Con la serie doblemente diferenciada vemos, en el PACF, que la autocorrelación de orden 1, sigue siendo significativa, pero la de orden 12 no lo es en el ACF, y quitamos lam estacionalidad, por lo que nos quedamos con la primera diferenciación

Procedemos a ajustar el modelo Arima:

- p es el orden del polinomio auto-regresivo, tiene un ACF sinusoidal que converge hacia 0, por lo que es de orden 2
- d es el numero de diferenciaciones, en nuestro caso 1.
- q es el orden del polinomio de medias móviles, en nuestro caso 0
- P es el orden del polinomio auto-regresivo estacional, en nuestro caso hemos elegido 1(usar la anterior).
- D en este caso no hacemos diferenciación para el perido estacional, es 0, porque al final no nos hemos quedado con la diferencicación de de orden 12
- Q es el orden del polinomio de medias móviles estacional, en nuestro caso 0
- S en nuestro caso el periodo es de 12.

Probamos nuestro modelo y vemos si sus residuales estan incorrelados:

```
viv_arima <- Arima((viv_train), c(2,1,0), seasonal = c(1,0,0))
checkresiduals(viv_arima)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0)(1,0,0)[12]
## Q* = 25.203, df = 21, p-value = 0.2385
##
## Model df: 3.   Total lags used: 24
```

```
print(viv_arima)
```

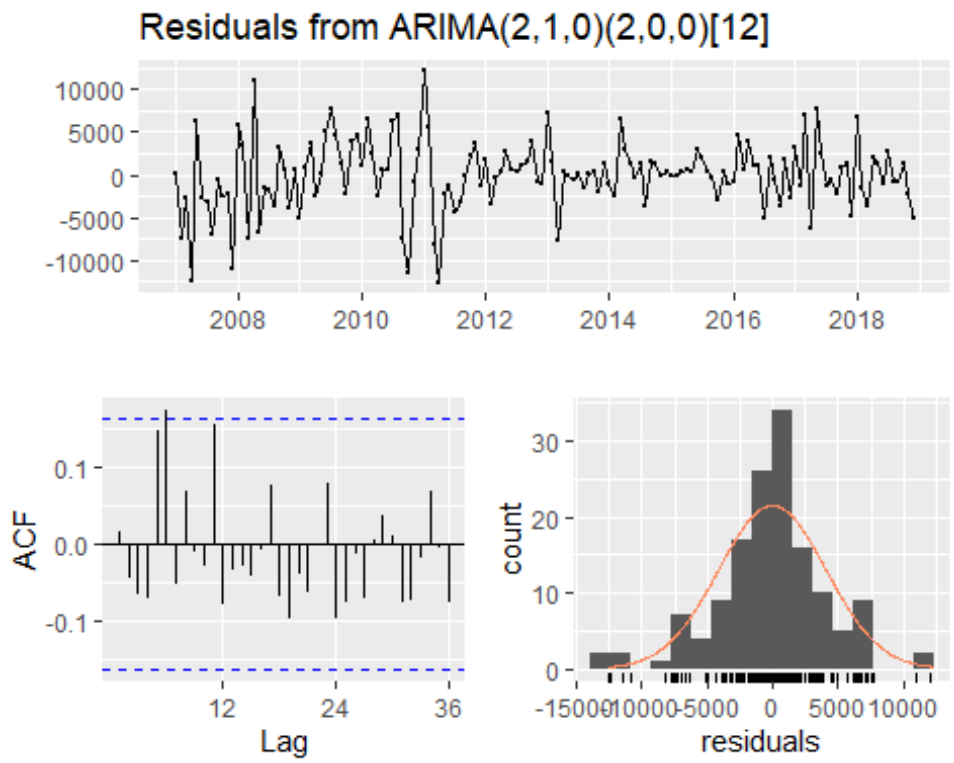
```
## Series: (viv_train)
## ARIMA(2,1,0)(1,0,0)[12]
##
## Coefficients:
##           ar1      ar2    sar1
##       -0.6000  -0.2534  0.5864
## s.e.   0.0821   0.0820  0.0737
##
## sigma^2 estimated as 18588614:  log likelihood=-1400.89
## AIC=2809.78   AICc=2810.07   BIC=2821.63
```

Puesto que el pvalor es mayor de 0.05 podemos decir que los residuos estan incorrelados, lo que implica que el modelo explica la dependencia de la serie. Sin embargo, en el gráfico de autocorrelaciones de los residuos vemos que el instante 12 no es 0.

Auto ajuste

Probamos la función autoArima para ver cual nos recomienda.

```
viv_auto_arima <- auto.arima(viv_train)

checkresiduals(viv_auto_arima)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0)(2,0,0)[12]
## Q* = 23.38, df = 20, p-value = 0.2705
##
## Model df: 4.   Total lags used: 24
```

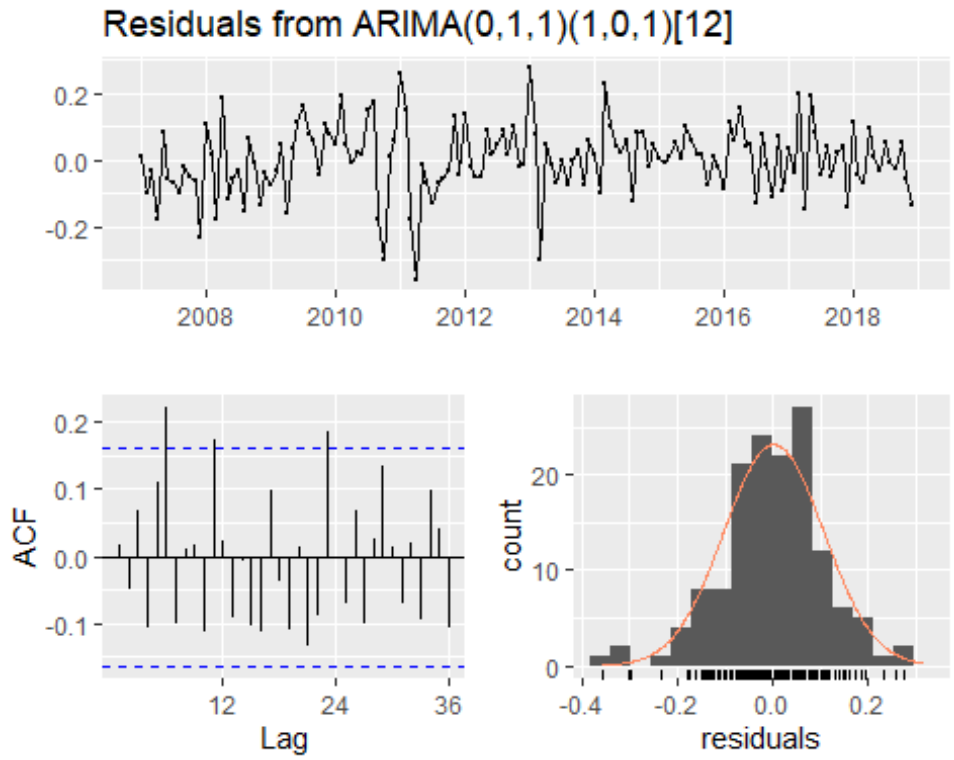
Puesto que el pvalor es mayor de 0.05 podemos decir que los residuos están incorrelados, lo que implica que el modelo explica la dependencia de la serie. El p-valor es algo mejor que el anterior que ajustamos anteriormente, y además ya no hay ninguno que esté igual de lejos que el instante 12 del anterior.

```
print(viv_auto_arima)

## Series: viv_train
## ARIMA(2,1,0)(2,0,0)[12]
##
## Coefficients:
##           ar1      ar2    sar1    sar2
```

```
##       -0.5946  -0.2658  0.4306  0.2772
## s.e.   0.0822   0.0818  0.0867  0.0933
##
## sigma^2 estimated as 17408514:  log likelihood=-1396.75
## AIC=2803.5   AICc=2803.93   BIC=2818.31
```

Auto ajuste logarítmico

```
viv_log_auto_arima <- auto.arima(log(viv_train))
checkresiduals(viv_log_auto_arima)
```



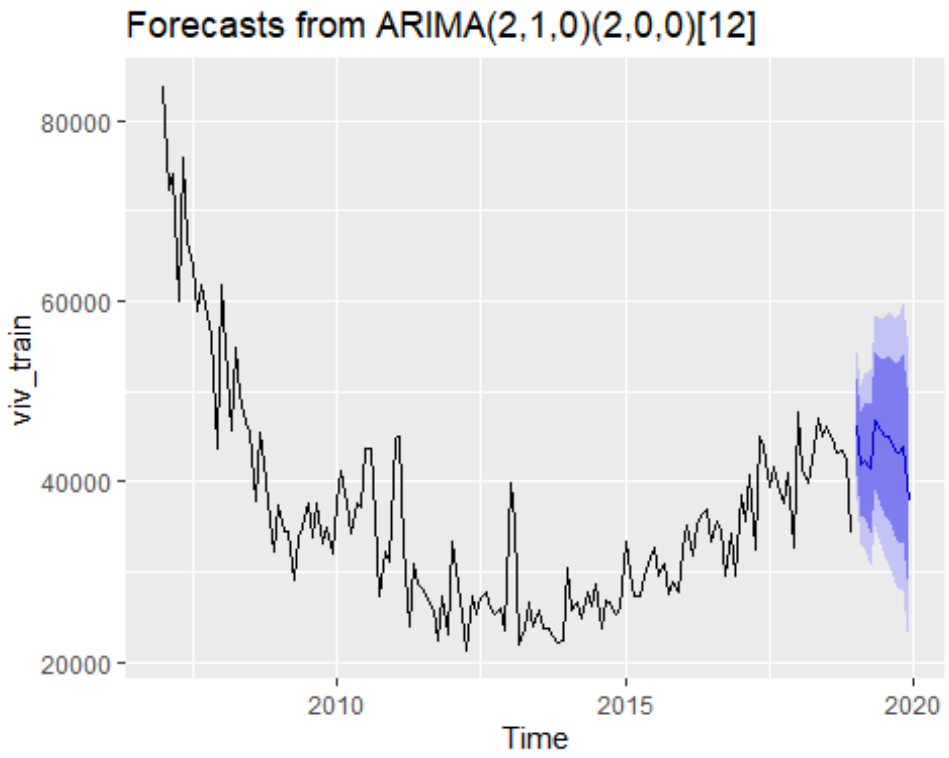

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(1,0,1)[12]
## Q* = 40.452, df = 21, p-value = 0.00655
##
## Model df: 3.   Total lags used: 24
```

Puesto que el pvalor es menor de 0.05 no podemos decir que los residuos están incorrelados, lo que implica que el modelo no explica la dependencia de la serie.

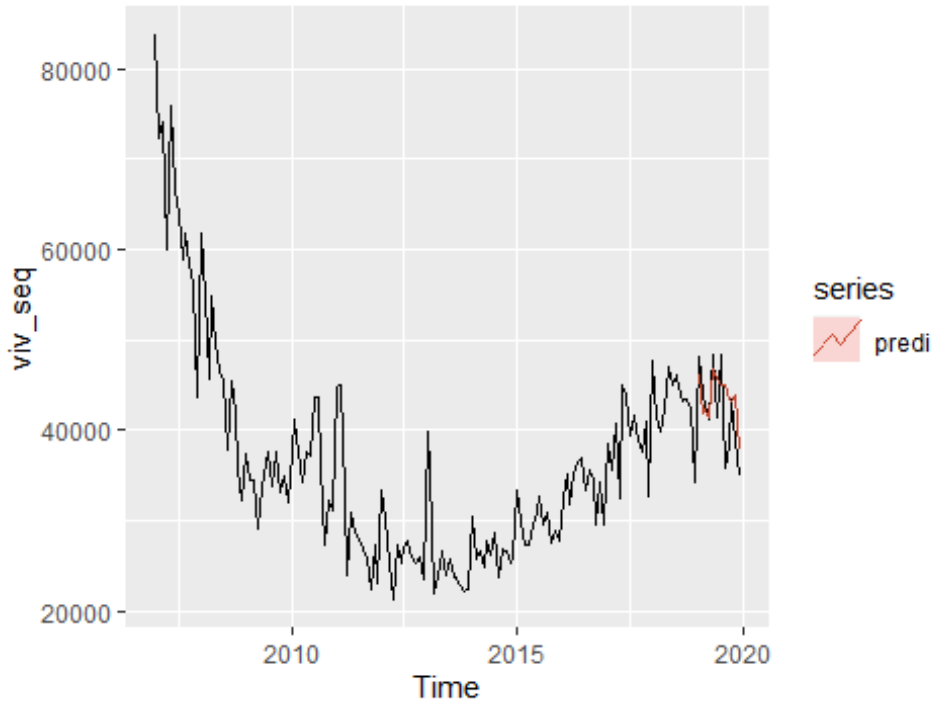
6. Escribir la expresión algebraica del modelo ajustado con los parámetros estimados. (1)

7. Calcular las predicciones y los intervalos de confianza para las unidades de tiempo que se considere oportuno, dependiendo de la serie, siguientes al último valor observado. Representarlas gráficamente. (1)

```
autoplot(forecast(viv_auto_arima, h=12))
```




```
autoplot(viv_seq) +
  autolayer(forecast(viv_auto_arima, h=12), series="predi", PI=FALSE)
```



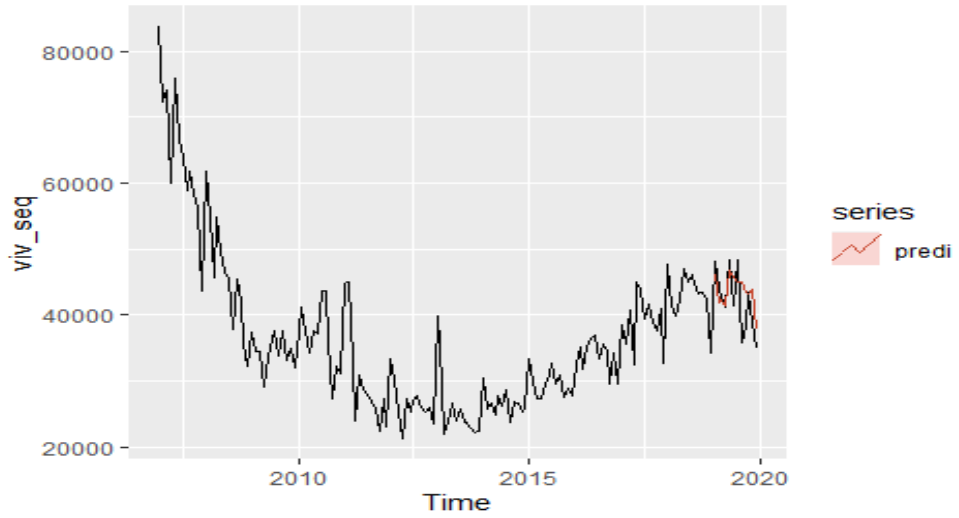

```
predi_auto <- forecast(viv_auto_arima,h=12)
accuracy(viv_auto_arima)
```

```
##                     ME     RMSE      MAE        MPE     MAPE      MASE
ACF1
## Training set -40.1272 4099.275 2964.97 -0.4039865 8.202484 0.4603514
0.0158448
```

```
predi_auto
```

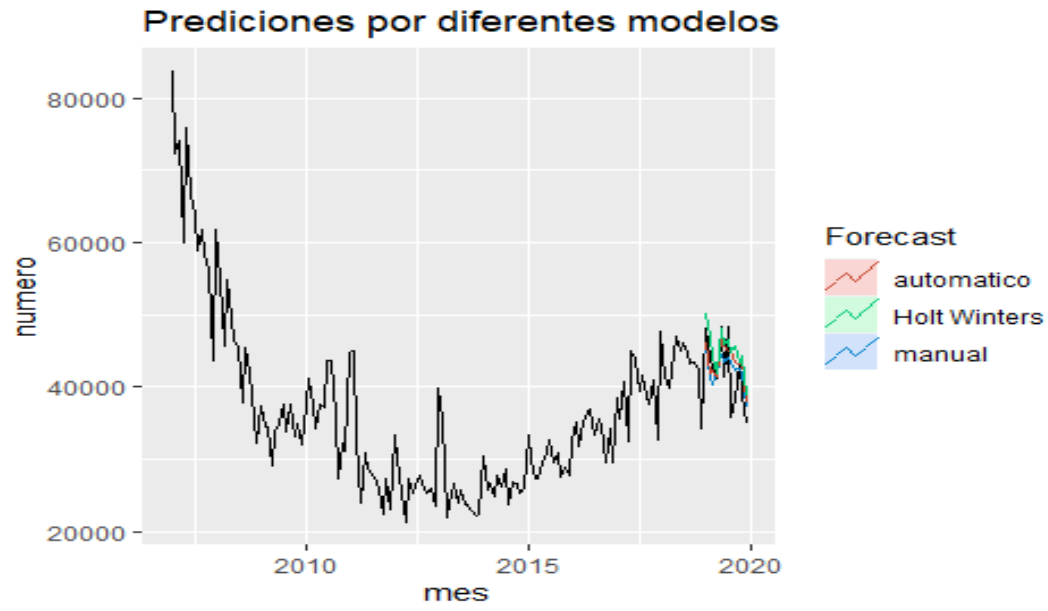
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2019       46189.51 40842.43 51536.60 38011.86 54367.17
## Feb 2019       41868.48 36098.62 47638.33 33044.24 50692.71
## Mar 2019       42337.77 35993.85 48681.69 32635.58 52039.96
## Apr 2019       41443.95 34337.27 48550.62 30575.23 52312.66
## May 2019       46814.44 39197.18 54431.69 35164.85 58464.03
## Jun 2019       45595.96 37457.50 53734.41 33149.26 58042.65
## Jul 2019       44918.54 36273.92 53563.16 31697.74 58139.34
## Aug 2019       44901.52 35798.40 54004.64 30979.50 58823.54
## Sep 2019       43473.13 33926.37 53019.90 28872.61 58073.65
## Oct 2019       43223.76 33252.40 53195.11 27973.89 58473.63
## Nov 2019       43798.31 33421.85 54174.76 27928.89 59667.72
## Dec 2019       37821.50 27054.12 48588.89 21354.21 54288.79
```

```
autoplot(viv_seq) +
  autolayer(forecast(viv_auto_arima, h=12), series="predi", PI=FALSE)
```



8. Comparar las predicciones obtenidas con cada uno de los métodos

```
autoplot(viv_seq) +
 autolayer(forecast(viv_arima,h=12), series="manual", PI=FALSE) +
 autolayer(forecast(viv_auto_arima,h=12), series="automatico", PI=FALSE)
+
  autolayer(forecast(viv_holt_winters,h=12), series="Holt Winters",
PI=FALSE) +
 ggtitle("Prediciones por diferentes modelos ") + xlab("mes") +
 ylab("numero") +
 guides(colour=guide_legend(title="Forecast"))
```



Podemos comprobar como el modelo del autoarima es algo mejor que el que he ajustado manualmente viendo el MAPE 8.20 frente a 8.70, además el modelo del autoarima sus residuos estaban más incorrelados, por lo que por doble motivo, nos quedamos con este.

```
accuracy(viv_auto_arima)
```

```
##                     ME     RMSE      MAE        MPE     MAPE      MASE
ACF1
## Training set -40.1272 4099.275 2964.97 -0.4039865 8.202484 0.4603514
0.0158448
```

```
accuracy(viv_arima)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -187.5408 4251.148 3097.418 -0.901882 8.707027 0.4809157
##                      ACF1
## Training set 0.007830328
```