Analisis Energetico por calor entre España e India

Adilene Calderon G.

12/12/2021

Introducción

En este trabajo aplicaremos a 2 base de datos el proceso de ingeniería de caracteristcias: Un paso fundamental al momento de trabajar una base de datos con Machine Learning.

Problema

En este caso abordaremos el tema de la energía.

Analizando como afecta la localización de un pais a la producción de esta. Sabiendo que India se encuentra en zona cercana al desierto mientras que España se encuentra en climas más templados.

Para la primera base de datos intentaremos usar los datos para predecir el precio de la energia según la producción de España.

Viendo entonces la primera baase de datos, la cual llamaremos $Base1$ Haciendo referencia a la base de origen, pues esta sera transformada para poder concluir la preparación de la base de datos.

```
Base1=read.csv("energy_dataset.csv")
```

Esta base recopilo los datos de la energia producida en España mediante distintas fuentes de generación durante un intervalo de tiempo (Aprox. 4 años). Algunas son energias renovables, otras no. Además que incluye la demanda de energia y su costo de la energia producida. Las variables se veran a detalle más adeltante: Algunas que no necesitamos seran eliminadas y las relevantes veremos que tan relacionadas estan con

Luego tenemos una base de datos similar: Se trata entonces de una base de datos sobre la energía en la India durante 2017 hasta 2020. Revisando la producción de las diferentes fuentes de energía, aqui no viene el precio. Pero podemos intentar predecir el consumo de cierta región de la India.

```
Base2=read.csv("file_02.csv",dec=".")
```

Analisis de Datos

Ahora partiremos de una revisión más detallada de las bases de datos. Viendo las variables con mayor detenimiento que en la introducción, a modo de ver alguna problema con estas que podamos resolver antes de la predicción.

Base 1: Energia en España

De primera instancia vemos que existen columnas poseen datos faltantes, estas las trataremos más adelante. Sin embargo tambien hay columnas que se encuentran vacias por lo que no aportan nada a la base de datos ni a nuestro trabajo, entonces las eliminaremos, usando R.

```
#Detección de columnas vacias.
BCol=length(Base1[1,])
BRow=length(Base1[,1])
Empty=array(0,dim=c(BRow,BCol))
for(i in 1:BCol){
    for(j in 1:BRow){

      if(is.na(Base1[j,i])==T){
        Empty[j,i]=1

      }
    }
}

VSum=rep(0,BCol)

for(i in 1:BCol){

  VSum[i]=sum(Empty[,i])

}
VSum
```

```
##  [1]     0    19    18    18    18    18    19    18    18    18 35064    19
## [13]    19    18    19    17    18    18    18    19    18    18     0 35064
## [25]     0     0    36     0     0
```

Con $VSum$ vemos que la base de datos SI posee datos faltantes pero estos se trataran más adelante. Ahora lo que haremos será eliminar las 2 columnas vacias que se pueden apreciar. Con el vector es claro que las columnas 11 y 24 estan vacias, por lo tanto las eliminaremos

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
EnergySpain=select(Base1,-colnames(Base1)[c(11,24)])

BCol2=length(EnergySpain[1,])
Empty=array(0,dim=c(BRow,BCol))
for(i in 1:BCol2){
    for(j in 1:BRow){
      if(is.na(EnergySpain[j,i])==T){
        Empty[j,i]=1
      }
    }
}

VSum2=rep(0,BCol)
```

```
for(i in 1:BCol){

  VSum2[i]=sum(Empty[,i])

}
VSum2
```

```
##  [1]  0 19 18 18 18 18 19 18 18 18 19 19 18 19 17 18 18 18 19 18 18  0  0  0 36
## [26]  0  0  0  0
```

Ahora en $VSum2$ vemos que ya no hay columnas vacias en nuestra base de datos, recordando que los datos faltantes en columnas no vacias se verán más adelante.

También eliminaremos las columnas que no aporten nada, es decir, las columnas con puros ceros.

```
Prueba=na.omit(EnergySpain)

Z=length(Prueba[1,])
Deletezeros=rep(0,Z)
for(i in 2:Z){

  if( sum(Prueba[,i])==0 ){

    Deletezeros[i]=i

    }
}
Deletezeros
```

```
##  [1]  0  0  0  4  0  0  0  8  9 10  0  0  0 14  0  0  0  0  0 20  0  0  0  0  0
## [26]  0  0
```

Entonces las columnas $4, 8, 9, 10, 14, 20$ son puros ceros, entonces no afectaron al precio. Entonces podemos eliminarlas.

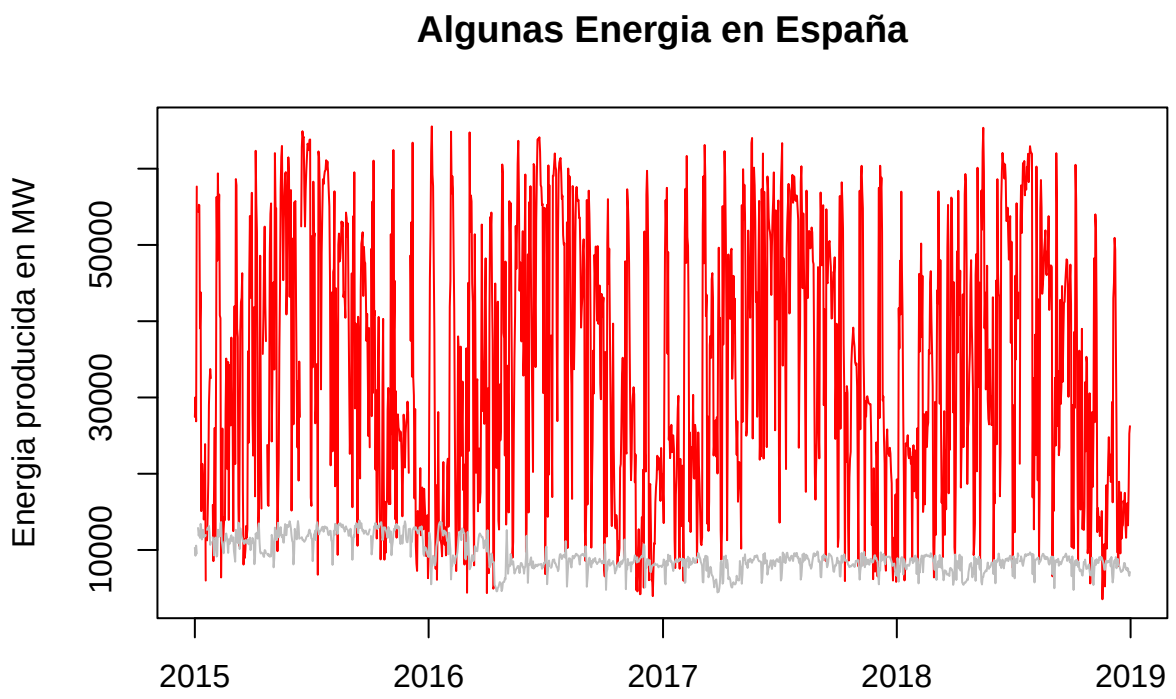
```
EnergySpain=select(EnergySpain,-colnames(EnergySpain)[c(4,8,9,10,14,20)])
```

Como la base de datos se trata de un registro de la energía generada en cada tipo de energía, teniendo desconocimiento del encargado del registro de datos supondremos que son coherentes (Aun con el pendiente de los datos faltantes). Entonces procedemos a ver un poco más a detalle las variables.

Como queremos predecir el precio de la energía eléctrica podemos prescidir de la hora y agruparemos las producciones por fechas.

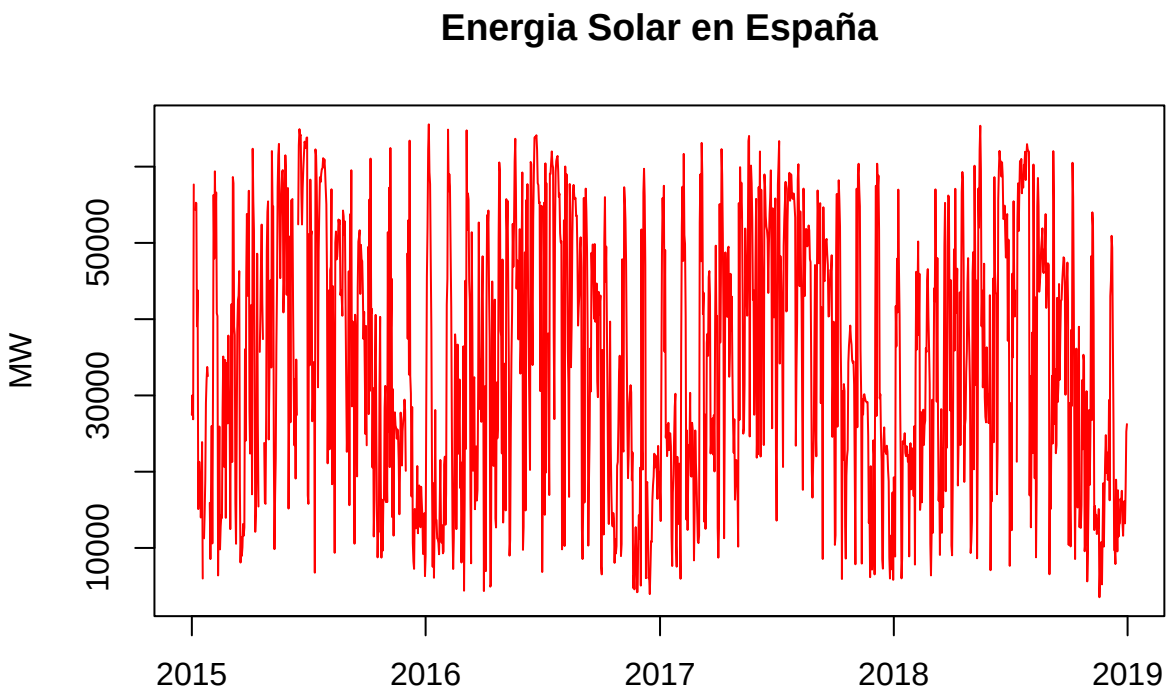
```
EnergySpain$time=as.Date(EnergySpain$time)
SpainDate=unique(EnergySpain$time)
NDSpain=length(SpainDate)    #Número de dias vistos en la base de datos.

#Energia producida por biomasa

Biomass=rep(0,NDSpain)
Solar=rep(0,NDSpain)
PricesA=rep(0,NDSpain)
for(i in 1:NDSpain){
  Q=filter(EnergySpain,time==SpainDate[i])
  Biomass[i]=sum(Q$generation.biomass)
  Solar[i]=sum(Q$generation.solar)
```

```
  PricesA[i]=mean(Q$price.actual)
}

plot(SpainDate,Solar,type="l",col="red",main = "Algunas Energia en España",ylab="Energia producida en M
lines(SpainDate,Biomass,col="gray")
```

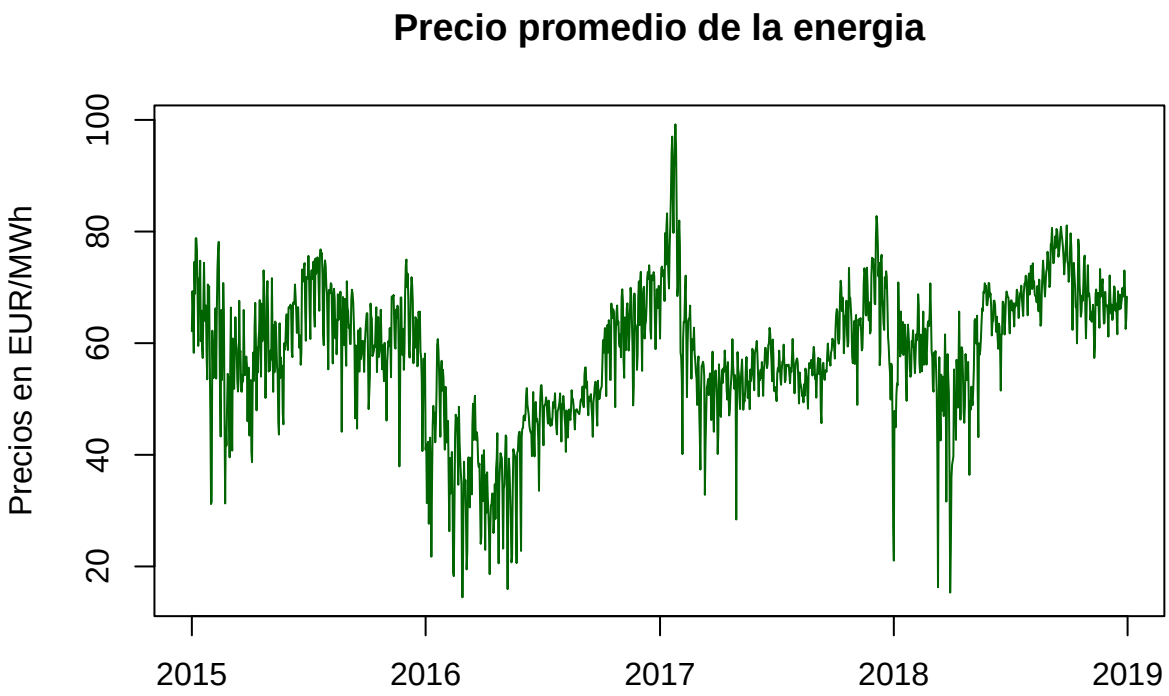


Aqui podemos ver, por ejemplo que en España se produce en promedio mucha más energía solar que Biomasa. Ahora veremos la energía solar con respecto al costo promedio por dia de la energia.

```
plot(SpainDate,Solar,col="red",type="l",main="Energia Solar en España",ylab="MW",xlab="")
```




```
plot(SpainDate,PricesA,col="darkgreen",type="l",ylab="Precios en EUR/MWh",xlab="",main="Precio promedio
```



Como podemos ver en las graficas anteriores cuando la producción de energia solar disminuyo el precio de la energía aumento. ¿Significa que estan relacionadas? Más adelante verificaremos que energias influyen más en el precio actual, esto con la finalidad de elegir la mejor para la predicción del precio de la energia promedio.

Tidy data

La base de datos solo tiene un punto de vista: La producción de energia durante cada hora del dia, durante 4 años. Mostrando que tanta energía en MW es producida en España distinguiendo entre fuentes. Además del precio de la energía y una predicción realizada por los encargados de la base de datos. Entonces por la base de datos solo tenemos la producción de energía como dato además de los precios para predecir el precio a futuro de la energía electrica.

Entonces retiraremos las columnas que no serán utilizadas para la predicción.

```
#Eliminamos la demanda y las observaciones del sol y el viento
E.Spain=select(EnergySpain,-colnames(EnergySpain)[16:19])
```

Quedandonos con el siguiente listado de variables

- *time*: La fecha en la que se tomo el registro (Para el trabajo se omitiran las horas)
- *generation.biomass*: Energía generada por biomasa en Megavatios MW
- *generation.fossil.brown.coal.lignite*: Energía generada por la quema de fosiles tipo lignite (carbon marrón) en Megavatios MW
- *generation.fossil.gas*: Energía generada por gas de carbon en Megavatios MW
- *generation.fossil.hard.coal*: Energía generada por carbon en Megavatios MW
- *generation.fossil.oil*: Energía generada por aceites fosiles en Megavatios MW

- *generation.hydro.pumped.storage.consumption*: Energía generada por bombeo de agua en Megavatios MW
- *generation.hydro.run.of.river.and.poundage*: Energía generada por los rios en Megavatios MW
- *generation.hydro.water.reservoir*: Energía generada por reservas de centrales hidroelécricas en Megavatios MW
- *generation.nuclear*: Energía generada por las plantas nucleares en Megavatios MW
- *generation.other*: Energía generada por otras fuentes no renovables en Megavatios MW
- *generation.other.renewable*: Energía generada por otras fuentes renovables en Megavatios MW
- *generation.solar*: Energía generada por el sol en Megavatios MW
- *generation.waste*: Energía generada por basura en Megavatios MW
- *generation.wind.onshore*: Energía generada por el viento en Megavatios MW
- *price.day.ahead*:Precio previsto en EUR/MWh
- *price.actual*: Precio de la energía en EUR/MWh

Limpieza de datos

Ya comenzamos en el analisis con la limpieza, corrigiendo el formato de fechas (a costo de la horas), eliminando columnas vacias o nulas (todo ceros). Ahora nos queda una de las cosas mas mencionadas durante el trabajo: Los datos faltantes, además de los valores extremos.

Con los datos faltantes veremos primeros cuantos hay por columna. Para eso recordemos $VSum2$.

```
VSum2
```

```
##  [1]  0 19 18 18 18 18 19 18 18 18 19 19 18 19 17 18 18 18 19 18 18  0  0  0 36
## [26]  0  0  0  0
```

```
Question=VSum2/length(E.Spain[,4])
Question
```

```
##  [1] 0.0000000000 0.0005418663 0.0005133470 0.0005133470 0.0005133470
##  [6] 0.0005133470 0.0005418663 0.0005133470 0.0005133470 0.0005133470
## [11] 0.0005418663 0.0005418663 0.0005133470 0.0005418663 0.0004848277
## [16] 0.0005133470 0.0005133470 0.0005133470 0.0005418663 0.0005133470
## [21] 0.0005133470 0.0000000000 0.0000000000 0.0000000000 0.0010266940
## [26] 0.0000000000 0.0000000000 0.0000000000 0.0000000000
```

Entonces observamos Question. Llamada así porque aqui entra la duda de que metodo usar para tratar los datos faltantes. Notemos primero que al ser un registro de producción de energía los datos faltantes son totalmente aleatorios. Luego para todas las columnas.

```
Percentage=round(Question*100,4)
Percentage
```

```
##  [1] 0.0000 0.0542 0.0513 0.0513 0.0513 0.0513 0.0542 0.0513 0.0513 0.0513
## [11] 0.0542 0.0542 0.0513 0.0542 0.0485 0.0513 0.0513 0.0513 0.0542 0.0513
## [21] 0.0513 0.0000 0.0000 0.0000 0.1027 0.0000 0.0000 0.0000 0.0000
```

Vemos que el porcentaje de datos faltantes es menor del 0.11%, es decir demasiado poco en relación con el tamaño de la base de datos. Por lo tanto podemos eliminar las filas que contengan datos faltantes con la certeza de no afectar la distribución de los datos.

Para eso checaremos los datos faltantes para eliminar las filas mientras recorremos las columnas.

```
E.Col=length(colnames(E.Spain))
for(j in 2:E.Col){
for(i in 1:BRow){
  if(is.na(E.Spain[i,j])==T){
    E.Spain=E.Spain[-i,]
  }
}
}
NewRow=length(E.Spain$time)
Empty2=array(0,dim=c(NewRow,E.Col))
Prueba2=rep(0,E.Col)
ENA.Spain=is.na(E.Spain)

for(i in 1:E.Col){
  Prueba2[i]=sum(as.numeric(ENA.Spain[,i]))
}

Prueba2
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Por Prueba2 podemos ver que ya no tenemos valores faltantes. Entonces podemos seguir con los valores valores extremos y la correlación de las variables. Procedemos primero con summarys de cada variable.

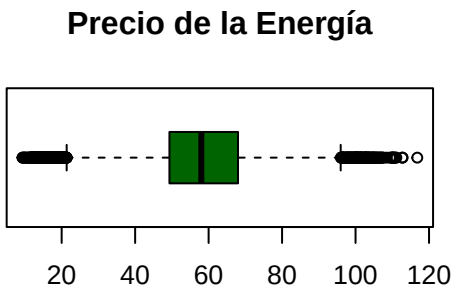
```
summary(E.Spain)
```

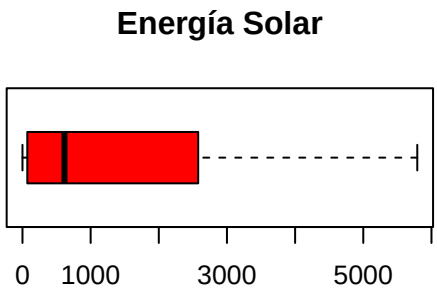
```
##       time            generation.biomass generation.fossil.brown.coal.lignite
##  Min.   :2015-01-01   Min.   :  0.0      Min.   :  0
##  1st Qu.:2016-01-01   1st Qu.:333.0      1st Qu.:  0
##  Median :2016-12-31   Median :367.0      Median :509
##  Mean   :2016-12-31   Mean   :383.5      Mean   :448
##  3rd Qu.:2017-12-31   3rd Qu.:433.0      3rd Qu.:757
##  Max.   :2018-12-31   Max.   :592.0      Max.   :999
##  generation.fossil.gas generation.fossil.hard.coal generation.fossil.oil
##  Min.   :    0         Min.   :   0               Min.   :  0.0
##  1st Qu.: 4126         1st Qu.:2527               1st Qu.:263.0
##  Median : 4969         Median :4474               Median :300.0
##  Mean   : 5623         Mean   :4256               Mean   :298.3
##  3rd Qu.: 6429         3rd Qu.:5838               3rd Qu.:330.0
##  Max.   :20034         Max.   :8359               Max.   :449.0
##  generation.hydro.pumped.storage.consumption
##  Min.   :   0.0
##  1st Qu.:   0.0
##  Median :  68.0
##  Mean   : 475.5
##  3rd Qu.: 616.0
##  Max.   :4523.0
##  generation.hydro.run.of.river.and.poundage generation.hydro.water.reservoir
##  Min.   :   0.0                             Min.   :   0
##  1st Qu.: 637.0                             1st Qu.:1077
##  Median : 906.0                             Median :2164
##  Mean   : 972.1                             Mean   :2605
##  3rd Qu.:1250.0                             3rd Qu.:3757
##  Max.   :2000.0                             Max.   :9728
##  generation.nuclear generation.other generation.other.renewable
```

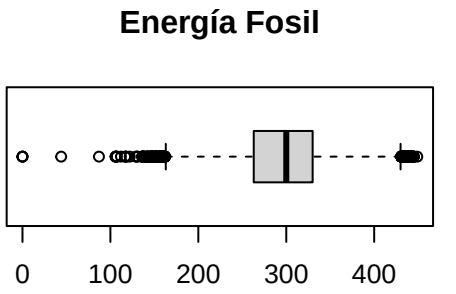
```
##  Min.   :   0       Min.   :  0.00   Min.   :  0.00
##  1st Qu.:5761       1st Qu.: 53.00   1st Qu.: 73.00
##  Median :6566       Median : 57.00   Median : 88.00
##  Mean   :6264       Mean   : 60.23   Mean   : 85.64
##  3rd Qu.:7025       3rd Qu.: 80.00   3rd Qu.: 97.00
##  Max.   :7117       Max.   :106.00   Max.   :119.00
##  generation.solar generation.waste generation.wind.onshore price.day.ahead
##  Min.   :   0     Min.   :  0.0    Min.   :    0           Min.   :  2.06
##  1st Qu.:  71     1st Qu.:240.0    1st Qu.: 2933           1st Qu.: 41.49
##  Median : 616     Median :279.0    Median : 4849           Median : 50.52
##  Mean   :1433     Mean   :269.5    Mean   : 5465           Mean   : 49.87
##  3rd Qu.:2578     3rd Qu.:310.0    3rd Qu.: 7398           3rd Qu.: 60.53
##  Max.   :5792     Max.   :357.0    Max.   :17436           Max.   :101.99
##   price.actual
##  Min.   :  9.33
##  1st Qu.: 49.34
##  Median : 58.01
##  Mean   : 57.88
##  3rd Qu.: 68.00
##  Max.   :116.80
```

y boxplot de algunas.

```
par(mfrow=c(2,2))
boxplot(E.Spain$price.actual,horizontal = T,main="Precio de la Energía",col="darkgreen")
boxplot(E.Spain$generation.solar,horizontal = T,main="Energía Solar",col="red")
boxplot(E.Spain$generation.fossil.oil,horizontal = T,main="Energía Fosil")
```







Vemos que tenemos muchos valores extremos en algunos de los diagramas de caja. La energía solar no posee valores extremos, sin embargo los datos tienen un sesgo a la izquierda.

Finalmente con la base de datos de energía española, veremos cuales de las 14 fuentes de energía tienen mayor relación con el precio de la energía, siendo eso respondido por la Covarianza.

```
Covar=array(NA,dim=c(1,14))
colnames(Covar)=colnames(E.Spain)[2:15]

for(i in 1:14){
  Covar[1,i]=round(cor(E.Spain$price.actual,E.Spain[,i+1]),4)
}

Covar
```

```
##      generation.biomass generation.fossil.brown.coal.lignite
## [1,]             0.1423                                0.3641
##      generation.fossil.gas generation.fossil.hard.coal generation.fossil.oil
## [1,]                0.4617                     0.4657                0.2846
##      generation.hydro.pumped.storage.consumption
## [1,]                                     -0.4263
##      generation.hydro.run.of.river.and.poundage
## [1,]                                     -0.137
##      generation.hydro.water.reservoir generation.nuclear generation.other
## [1,]                           0.0716            -0.0526              0.1
##      generation.other.renewable generation.solar generation.waste
## [1,]                     0.2561           0.0984           0.1696
##      generation.wind.onshore
## [1,]                 -0.2208
```

Como la correlación es diferente de cero, existe una relación entre las diferentes energias con respecto al precio. Algo a considerar para una futura predicción seria usar las energías con mayor coeficiente de correlación en valor absoluto.

```
Plus=max(Covar)
Minus=min(Covar)

Plus
```

```
## [1] 0.4657
```

```
Minus
```

```
## [1] -0.4263
```

```
colnames(Covar)[c(4,6)]
```

```
## [1] "generation.fossil.hard.coal"
## [2] "generation.hydro.pumped.storage.consumption"
```

Teniendo que las energias que mayor peso tienen con respecto al precio son la de carbon. Por lo tanto serian las variables más adecuadas para usarse para una predicción sobre el precio de la energía.

Finalmente, de haber partido de la base de datos $Base1$ terminamos en la base de datos $E.Spain$ a la cual se anexara al diccionario antes mencionado (Este se anexara fuera de codigo) y se exportara en un archivo .csv

```
write.csv(E.Spain,"Predicción de precio de energía electrica.csv")
```

Base 2: Energía en la India

Ahora cambiamos a la India, en este caso no tenemos tantas fuentes de energía.

```
colnames(Base2)
```

```
## [1] "index"
## [2] "Date"
## [3] "Region"
## [4] "Thermal.Generation.Actual..in.MU."
## [5] "Thermal.Generation.Estimated..in.MU."
## [6] "Nuclear.Generation.Actual..in.MU."
## [7] "Nuclear.Generation.Estimated..in.MU."
## [8] "Hydro.Generation.Actual..in.MU."
## [9] "Hydro.Generation.Estimated..in.MU."
```

Siendo unicamnete 3, la termica, la nuclear y la hidroelecrica. Sin embargo la diferencia esta en qu la base de datos divide a la India en regiones.

```
Regions=unique(Base2$Region)
Regions
```

```
## [1] "Northern"     "Western"      "Southern"     "Eastern"      "NorthEastern"
```

Entonces en este caso podemos focalizarnos en las regiones y preparar la base de datos para predecir cual de las regiones tiene mayor peso en la energia eléctrica total de la India.

Analisis y Limpieza de Datos

En este caso tenemos un problema, vease que la base de datos posee p

```
S=Base2$Thermal.Generation.Actual..in.MU.[2]
S
```

```
## [1] "1,106.89"
```

Aqui tenemos que los números estan en forma de texto, esto puede resolverse con *as.numeric*, pero la coma (,) hace que esto no sea efectivo, sin embargo esto se puede resolver con la libreria *stringr*.

```
library(stringr)

str_remove_all(S,",")
```

```
## [1] "1106.89"
```

Entonces la coma desaparece y podemos trabajar con *as.numeric*

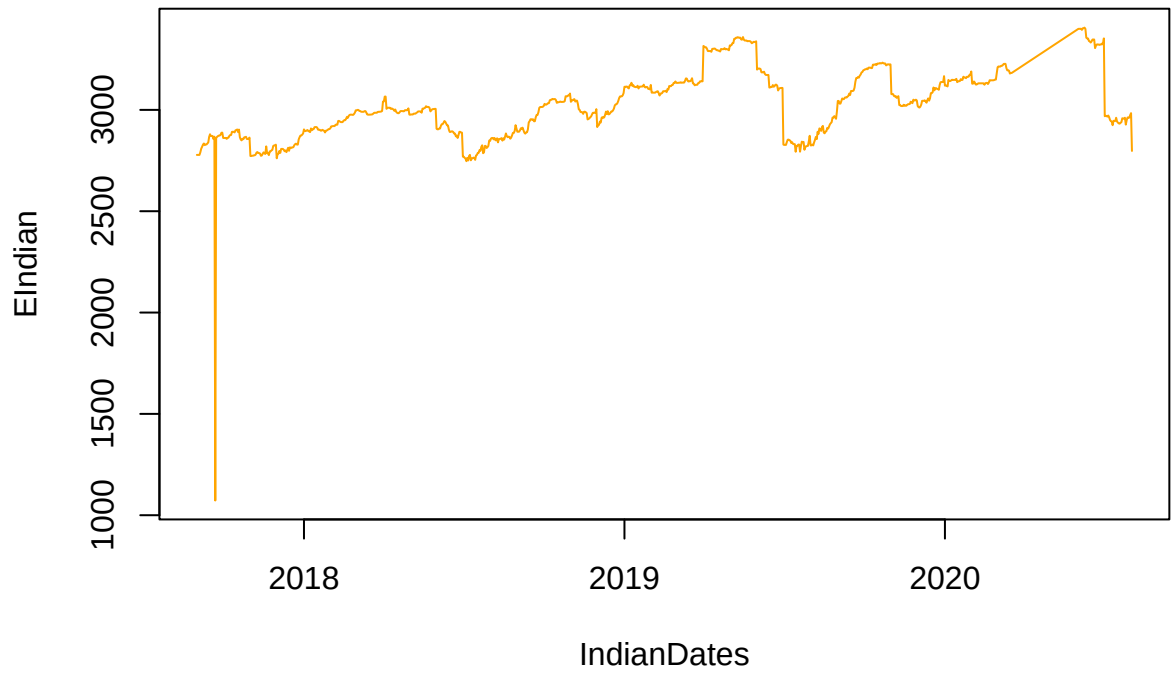
```
N2Row=length(Base2$Region)
N2Col=length(colnames(Base2))

for(i in 1:6){
  for(j in 1:N2Row){
    Base2[j,i+3]=str_remove_all(Base2[j,i+3],",")
  }
  Base2[,i+3]=as.numeric(Base2[,i+3])
}
```

Ahora, como buscamos el aporte real de cada estado, eliminaremos las columnas de aportaciones estimadas

```
IndianEnergy=Base2[,-c(5,7,9)]
IndianEnergy$Date=as.Date(IndianEnergy$Date)
```

Ahora veremos la generación de la India por región. Para ver el comportamiento de las fuentes de energia por estado

```
IndianDates=unique(IndianEnergy$Date)
nID=length(IndianDates)
EIndian=rep(0,nID)
GIndian=EIndian
FIndian=EIndian
for(i in 1:nID){

  QQ=filter(IndianEnergy,Date==IndianDates[i])
  EIndian[i]=sum(QQ$Thermal.Generation.Actual..in.MU)
  GIndian[i]=sum(QQ$Nuclear.Generation.Actual..in.MU)
  FIndian[i]=sum(QQ$Hydro.Generation.Actual..in.MU)
}

plot(IndianDates,EIndian,type="l",col="orange")
lines(IndianDates,GIndian)
lines(IndianDates,FIndian)
```



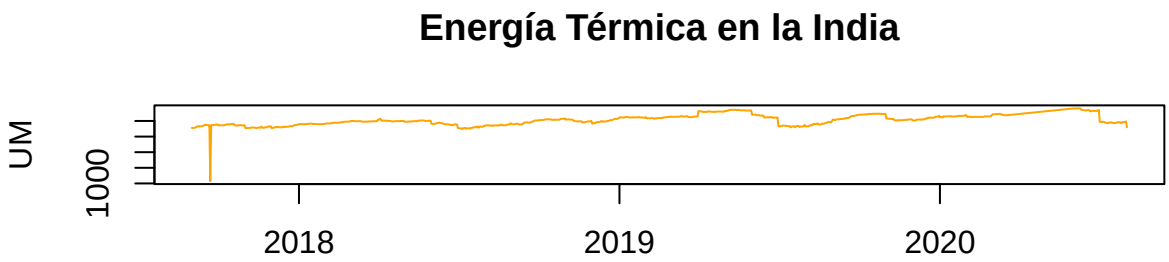
Notamos que dos de las graficas no aparecen. Esto es porque en las generación Nuclear hay muchos valores faltantes

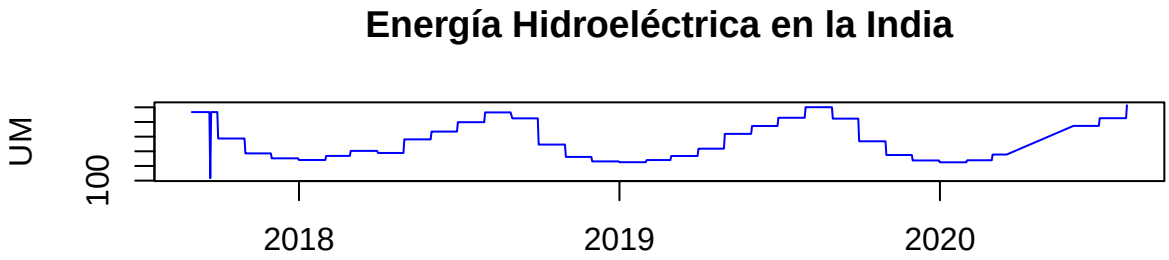
```
X=as.numeric(is.na(IndianEnergy$Nuclear.Generation.Actual..in.MU.))

sum(X)*100/N2Row
```

```
## [1] 40
```

Tenemos que el 40% de los valores de la energia nuclear son faltantes, al ser demasiados puede esta darnos muy poca información, por lo tanto optaremos por descartarla del analisis. Entonces checamos la energía termica e hidroelectrica.

```
par(mfrow=c(2,1))
plot(IndianDates,EIndian,type="l",col="orange",main="Energía Térmica en la India",ylab="UM",xlab="")
plot(IndianDates,FIndian,type="l",main="Energía Hidroeléctrica en la India",ylab="UM",xlab="",col="blue
```





Como se puede apreciar la energía termica es significativamente mayor. Entonces podemos concluir que la energia principal en la India seria la termica, lo cual tiene sentido por su ubicación.

tidy data

La base de datos es más pequeña que la anterior, entonces la transformación que recibio fue la vista en el Analisis y Limpieza de Datos, ya que la energia nuclear no nos podria dar mucha información debido a la gran cantidad de datos faltantes. Entonces la base de datos quedaria como sigue.

```
E.Indian=IndianEnergy[,-c(1,5)]
colnames(E.Indian)
```

```
## [1] "Date"                                "Region"
## [3] "Thermal.Generation.Actual..in.MU." "Hydro.Generation.Actual..in.MU."
```

Donde

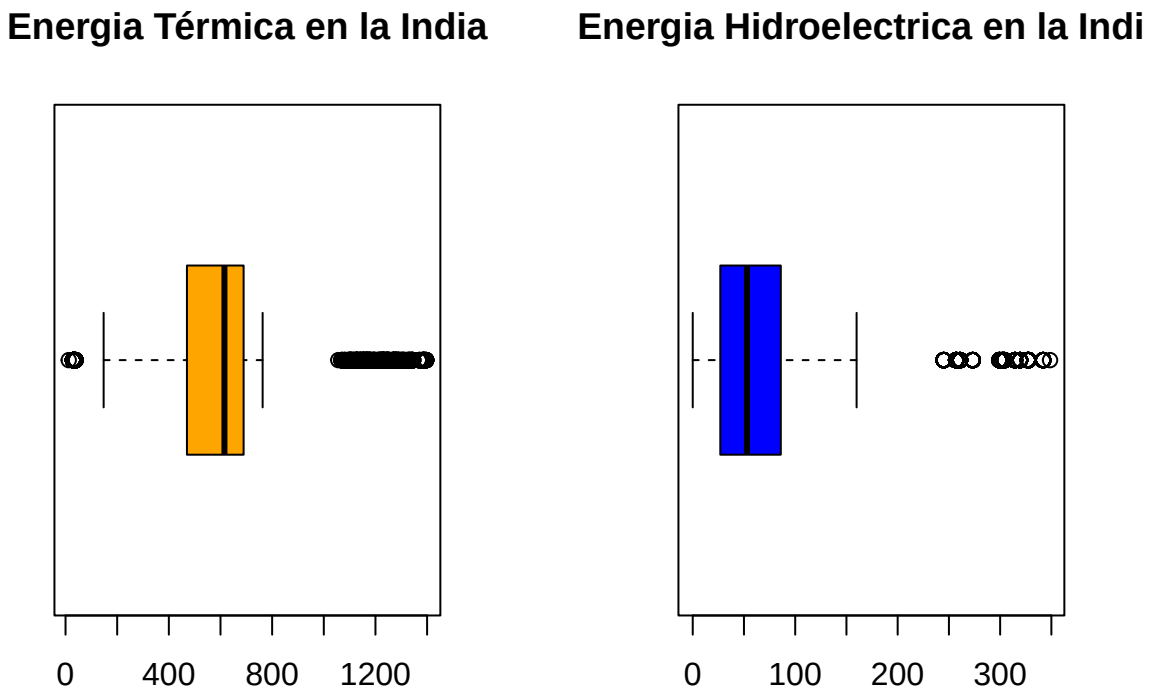
- *Date*: La fecha de registro del dato
- *Region*: La región de la India donde se registro.
- *Thermal.Generation.Actual..in.MU.*: La energía térmica generada en MU

- *Hydro.Generation.Actual..in.MU* La energía hidroeléctrica generada en MU

Ya con las fechas en el formato adecuado, al igual que la energia generada representada como un número, la base de datos estaria preparada para predecir la producción de cada región en un tiempo determinado sobre una energia en particular o la total (la suma de las 2).

Finalmente veremos los puntos extremos de ambas Energias.

```
par(mfrow=c(1,2))
boxplot(E.Indian$Thermal.Generation.Actual..in.MU.,horizontal = T,
        main="Energia Térmica en la India",col="orange")
boxplot(E.Indian$Hydro.Generation.Actual..in.MU.,horizontal = T,
        main="Energia Hidroelectrica en la India",col="blue")
```



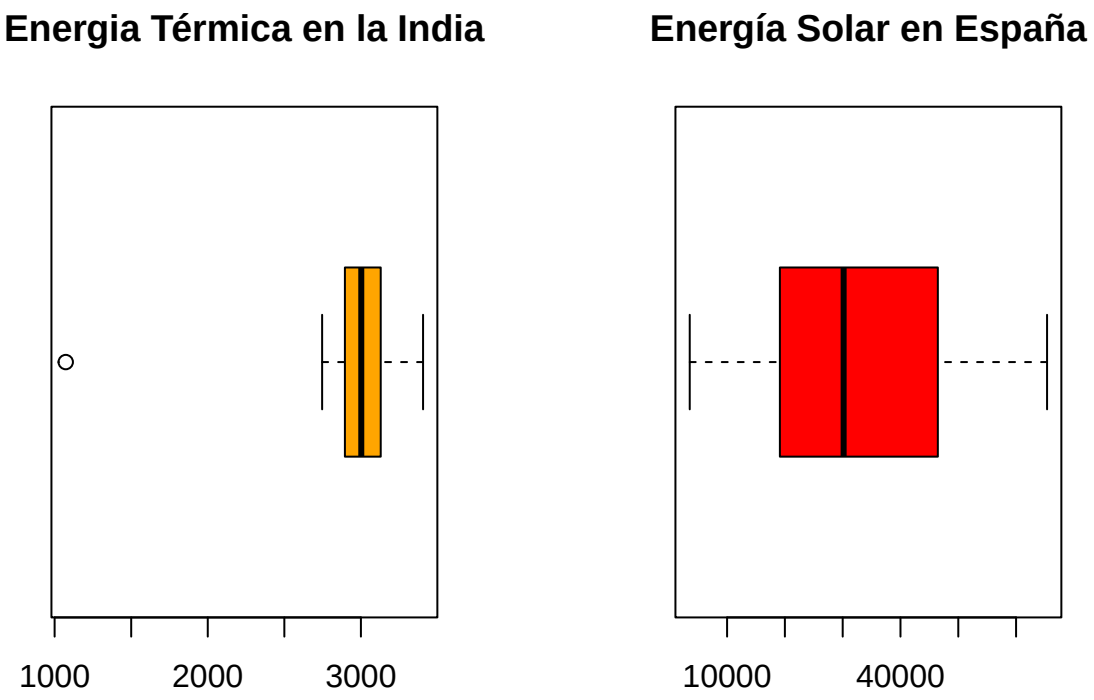
Entonces vemos que existen muchos más valores extremos en la energia Térmica que en la Hidroeléctrica. Para concluir con el analisis exportaremos la base de datos modificada.

```
write.csv(E.Indian,"Energia en India.csv")
```

Ahora podemos hacer la comparativa entre Esapaña e India, pero antes debemos de restringir el periodo de tiempo. viendo lo siguiente

```
Solar2=rep(0,nID)
for(i in 1:nID){

  Q=filter(E.Spain,time==IndianDates[i])
  Solar2[i]=sum(Q$generation.solar)
}
```

```r
Solar2=na.omit(Solar2)
for(i in 1:length(Solar2)){
  if(Solar2[i]==0){
    Solar2[i]=NA
  }
}
Solar2=na.omit(Solar2)


par(mfrow=c(1,2))
boxplot(EIndian,horizontal = T,
        main="Energia Térmica en la India",col="orange")
boxplot(Solar2,horizontal = T,main="Energía Solar en España",col="red")
```



Entonces podemos ver que España con la energia solar aprovecha el calor de una manera más eficiente que la India. Haciendo claro que más que la ubicación del pais es la organización del mismo quien decidira la producción de cierta energía.

Reducción Dimensional

Para finaliza este trabajo trataremos de aplicar algun metodo de reducción de caracteristicas

Base de Datos 1

Ahora veremos si podemos reducir las dimensiones de las fuentes de energia de España.

Para este caso aplicaremos el metodo PCA o analisis de componentes principales, a modo de agrupar las fuentes de energia en menos variables. Por lo tanto usaremos una parte de la base de datos transformada para aplicar el metodo.

```
PCAdf=select(E.Spain,seq(2,15))
```

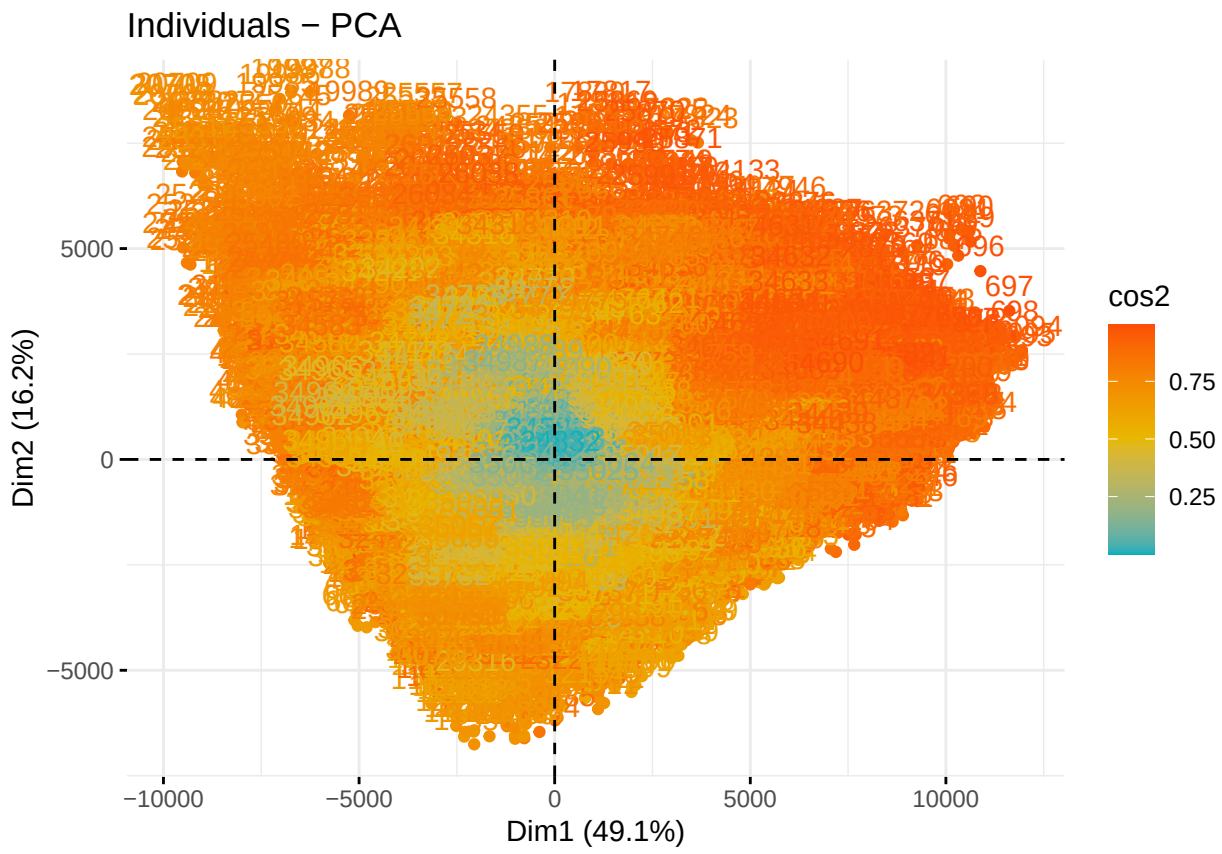
Ahora si, aplicaremos PCA sobre las fuentes de energía y trataremos de visualizar las variables en terminos de menos variables.

```
prSpain<-prcomp(PCAdf, scale = FALSE)




library(FactoMineR)
library(stats)
library(factoextra)
```

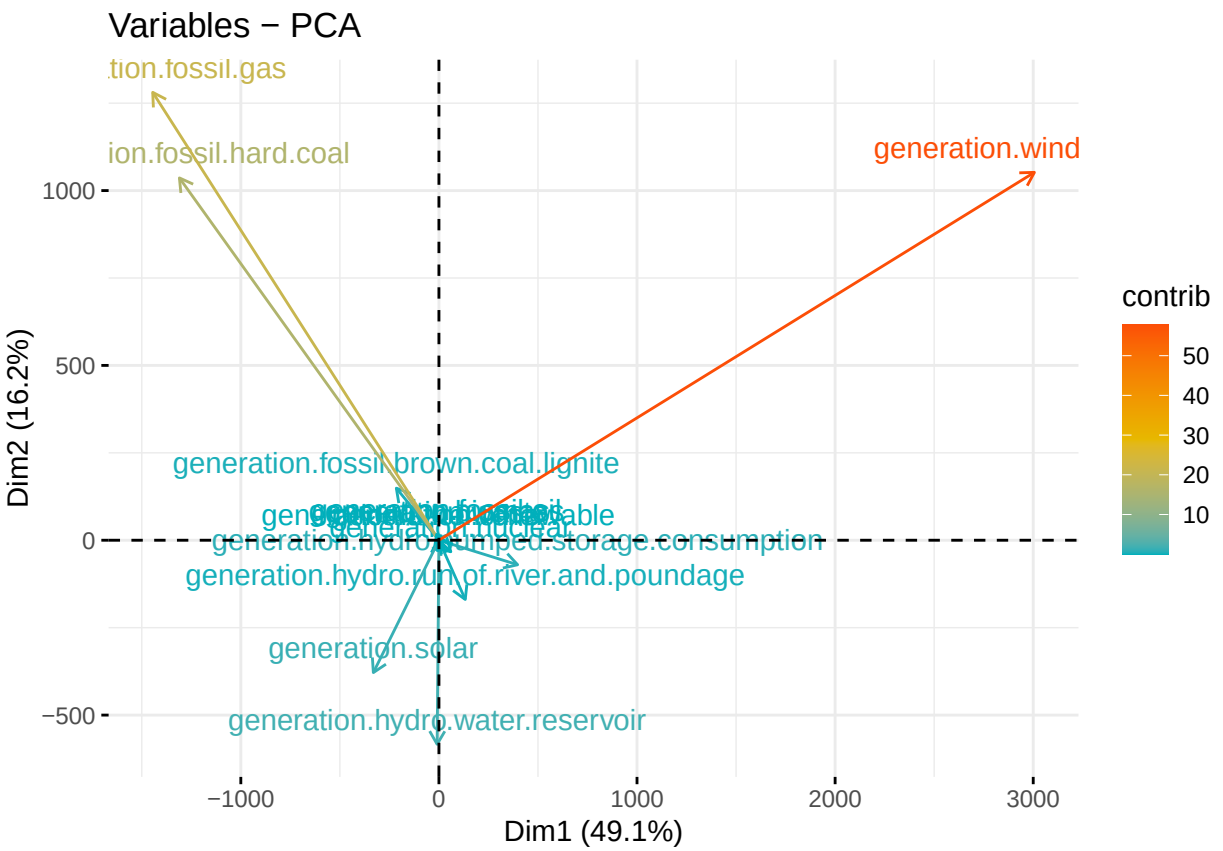
```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ggplot2)

Vis.Spain=PCA(X=PCAdf,scale.unit = FALSE,ncp=14,graph = F)
par(mfrow=c(1,2))
fviz_pca_ind(Vis.Spain,col.ind = "cos2", gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = FALSE)
```

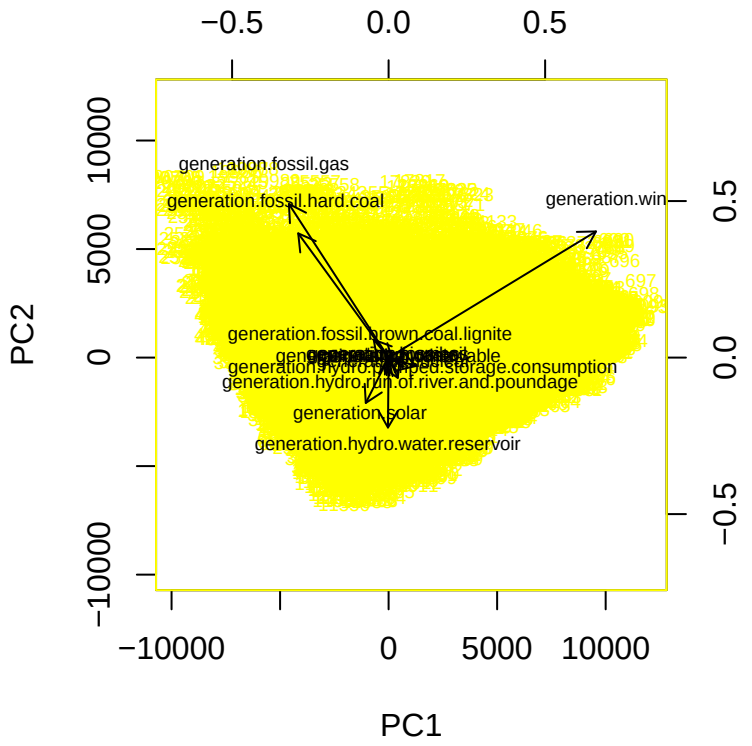



```
fviz_pca_var(Vis.Spain,col.var = "contrib",gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = FALSE)
```



Aqui vemos como las 14 variables son usadas como "ejes" para combinarlas dejandolas en solo 2 variables.

```
biplot(x = prSpain, scale = 0, cex = 0.6, col = c("yellow", "black"))
```



Quedando esto, las 14 fuentes de energía combinadas en 2 variables $PC1$ Y $PC2$. Además que estas 2 nuevas vaariables tienen una relación del 50 con las 14 variables.

Base de Datos 2

En el caso de la Base de datos sobre la india tenemos que solamente tenemos 2 dimensiones númericas, por lo tanto se piensa que el PCA no es necesario para esta base de datos en particular, menos aun que se elimino la energía nuclear por su alto radio de valores faltantes.

Conclusion

En este trabajo se hizo claro la importancia del proceso de ingenieria de caracteristicas, siendo adecuada una preparación de la base de datos a usar para algun metodo Machine Learning.

Con respecto a la comparativa entre España a India, cabe destacar la falta de detalle en cuanto a las variables a evaluar, ya que a pesar que se tomaron las mismas fechas, carecemos de información con respecto a la tecnologia que usa cada pais para la generación y almacenación, ignorando tambien su demanda de energía.

Algunos puntos a resaltar pudiera ser la falta de contextualizacion sobre los problemas a la hora de elegir el metodo de reducción de caractéristicas, pues la problematica principal era la predicción del costo de la energia (Base 1) y el desarrollo energetico de las regiones en la India (Base 2). Pero a fin de cuentas esto es una preparación, no se ha probado nada. Y no hay metodos para garantizar la elección correcta del metodo hasta que se pruebe un algortimo sobre la base de datos para que el tiempo de ejecución y los resultados obtenidos sean la evidencia si se utilizo el metodo adecuado, o en algunos casos no debio haberse usado algun metodo de reducción de caracterisitcas.

Referencias

Base de Datos

1. Navin Mundhra. Daily Power Generation in India (2017-2020). Kaggle.com. Published 2017. Accessed December 14, 2021. $https://www.kaggle.com/navinmundhra/daily-power-generation-in-india-20172020?select=file_02.csv$.

2. Jhana N. Hourly energy demand generation and weather. Kaggle.com. Published 2019. Accessed December 14, 2021. $https://www.kaggle.com/nicholasjhana/energy-consumption-generation-prices-and-weather$.