basico.R

nialv_000

Fri Jun 29 14:26:53 2018

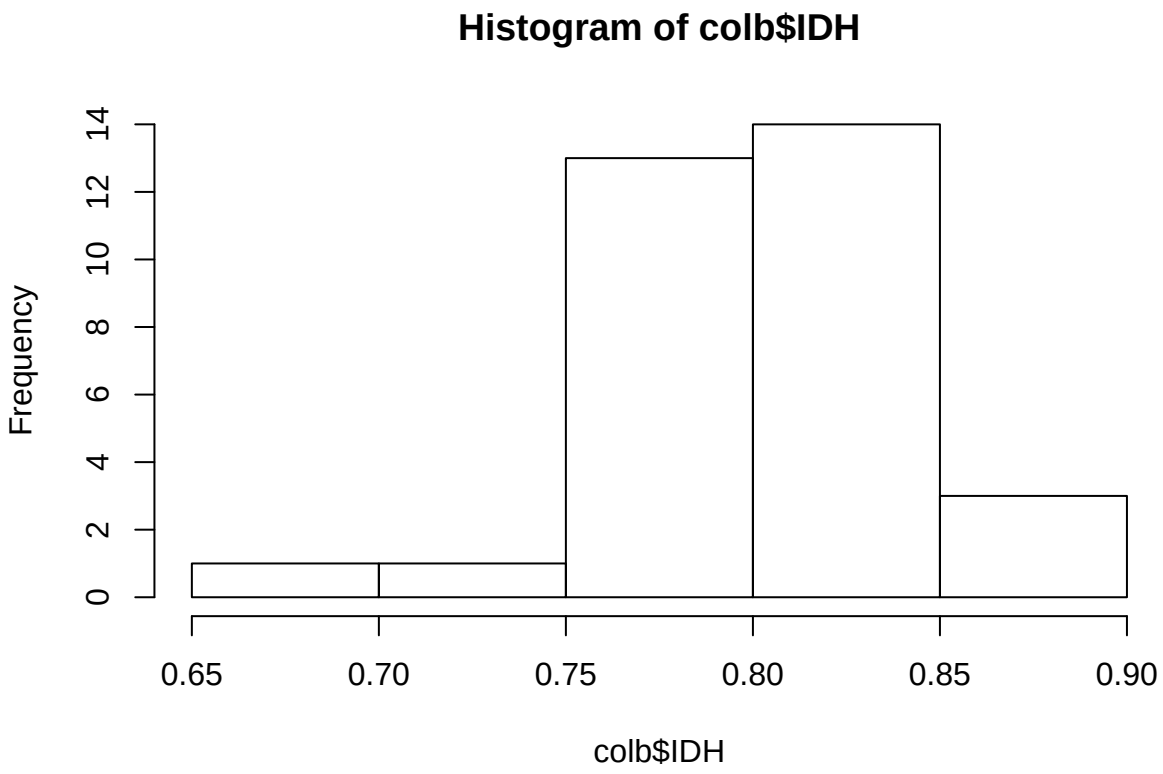
```
# Cargar archivo ----

# NO OLVIDAR set working directory
filename="colombia.csv"
colb=read.csv(filename, stringsAsFactors = FALSE)

# que variables y tipo
str(colb)
```

```
## 'data.frame':    32 obs. of  6 variables:
##  $ IDH               : num  0.879 0.867 0.865 0.849 0.842 0.839 0.837 0.835 0.834 0.832 ...
##  $ Departamento      : chr  "Santander" "Casanare" "Valle del Cauca" "Antioquia" ...
##  $ PoblaciÃ³n.Cabecera: int  1587972 281548 4169553 5262172 742812 761658 10070801 2438533 56487 506
##  $ PoblaciÃ³n.Resto   : int  502867 93701 586560 1428858 539251 206109 914484 107391 21926 68756 ...
##  $ PoblaciÃ³n.Total   : int  2090839 375249 4756113 6691030 1282063 967767 10985285 2545924 78413 575
##  $ DepartamentoNorm  : chr  "Santander" "Casanare" "Valle del Cauca" "Antioquia" ...
```

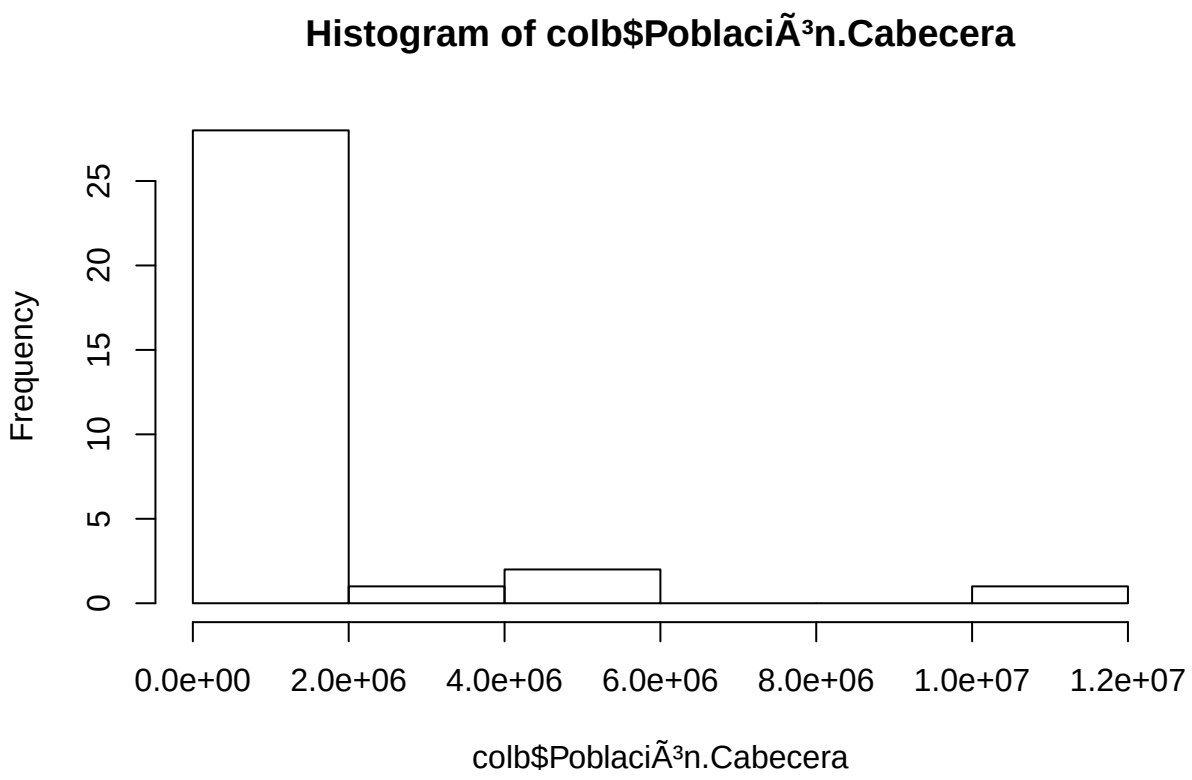
```
# Exploracion Univariada ------------------------------------------------

## estadisticos

# nos interesa IDH, y poblacion cabecera y poblacion resto
# no se puede secar tabla de frecuencia,
# solo estadisticos:
summary(colb)
```

```
##       IDH         Departamento       PoblaciÃ³n.Cabecera PoblaciÃ³n.Resto
##  Min.   :0.6910   Length:32          Min.   :   13090    Min.   :  21926
##  1st Qu.:0.7680   Class :character   1st Qu.:  234624    1st Qu.:  96969
##  Median :0.8040   Mode  :character   Median :  717197    Median : 268112
##  Mean   :0.8018                      Mean   : 1196730    Mean   : 360590
##  3rd Qu.:0.8343                      3rd Qu.:  970925    3rd Qu.: 487530
##  Max.   :0.8790                      Max.   :10070801    Max.   :1428858
##  PoblaciÃ³n.Total   DepartamentoNorm
##  Min.   :   43446   Length:32
##  1st Qu.:  371161   Class :character
##  Median : 1028429   Mode  :character
##  Mean   : 1557320
##  3rd Qu.: 1512087
##  Max.   :10985285
```

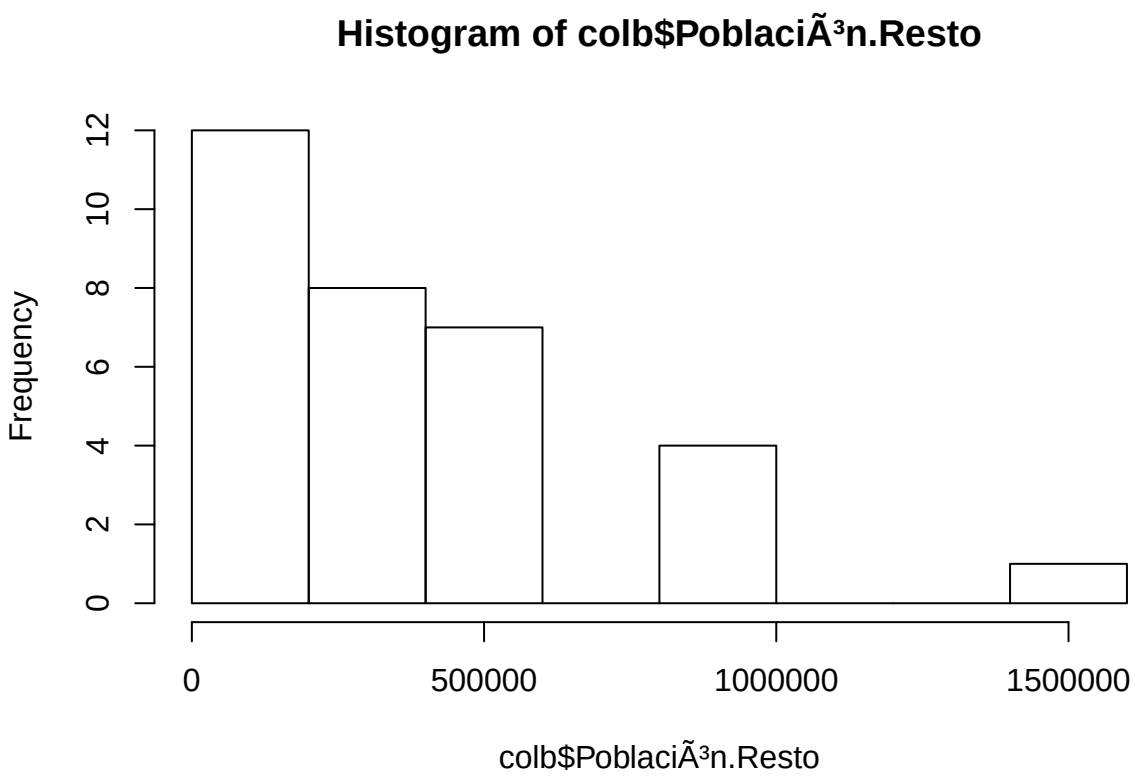
```
## graficos
# el plot de cada uno seria el histograma:

hist(colb$IDH)
```



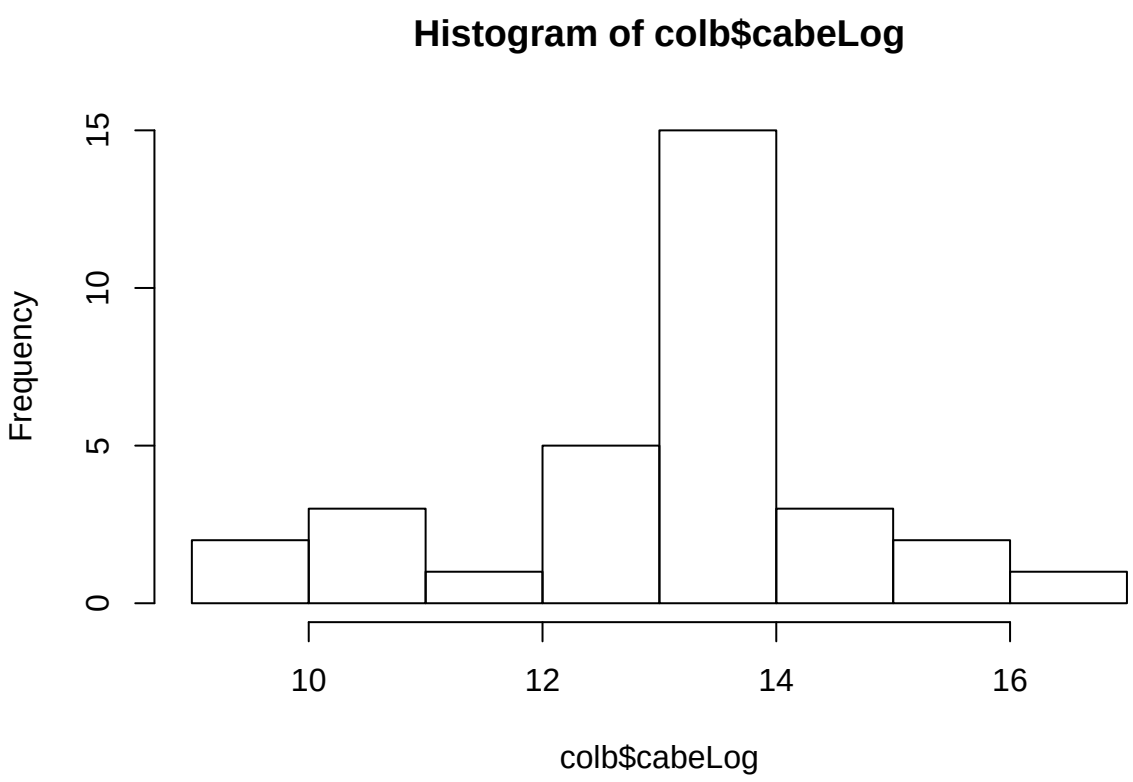

```
hist(colb$PoblaciÃ³n.Cabecera)
```



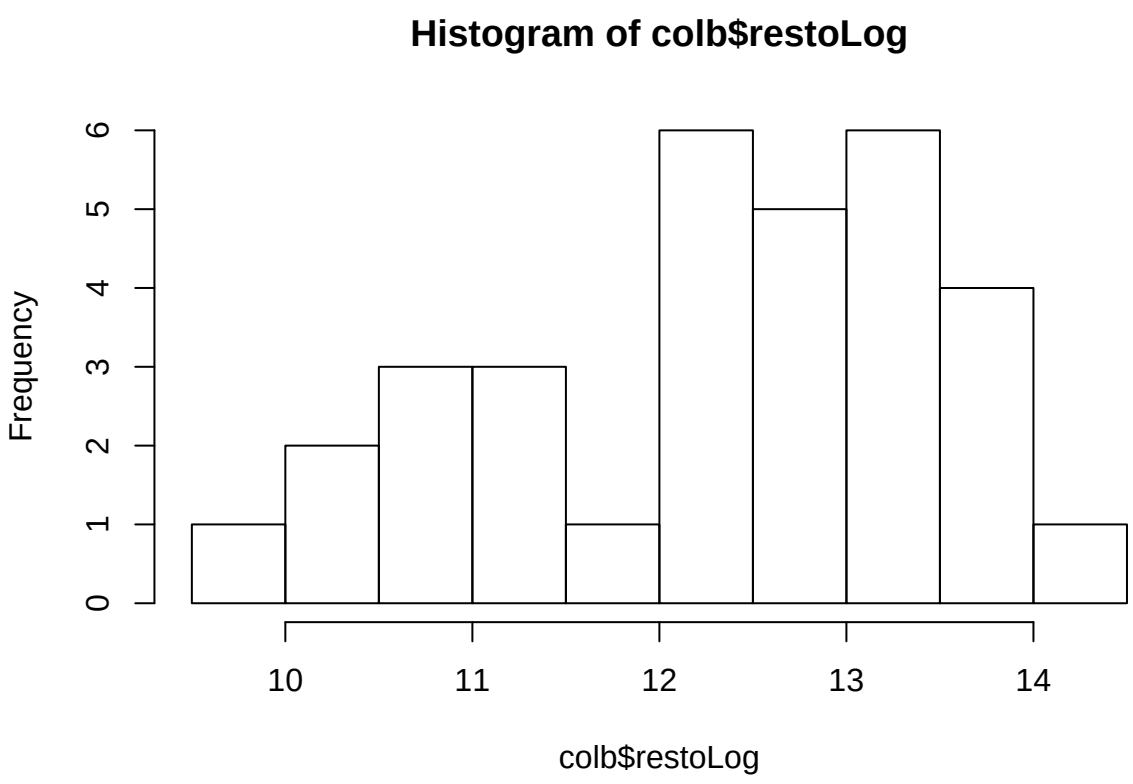

```
hist(colb$PoblaciÃ³n.Resto)
```




```
# dado el sesgo de las pobaciones,
# podriamos transformarla para que se acerque a la
# normalidad

colb$cabeLog=log(colb$PoblaciÃ³n.Cabecera)
colb$restoLog=log(colb$PoblaciÃ³n.Resto)

hist(colb$cabeLog)
```




```
hist(colb$restoLog)
```




```
# Exploracion Bivariada ----------------------------------------------------


# En este trabajo estamos interesados en el impacto de
# la poblacion en el el IDH, veamos IDH con cada uno:

explanans=names(colb)[c(7:8)] # usando las logs
corrDem=cor(colb$IDH,colb[,explanans],
            use = "na.or.complete")
corrDem
```

```
##         cabeLog  restoLog
## [1,] 0.4873974 0.1773112
```

```
# y la correlaciÃ³n entre las variables independientes:

corrTableX=round(cor(colb[,explanans],
                         use = "na.or.complete"),2)
corrTableX_copy=corrTableX
corrTableX[upper.tri(corrTableX)]<-""
#ver:
corrTableX
```

```
##           cabeLog restoLog
## cabeLog   "1"     ""
## restoLog "0.84"  "1"
```

```r
# visualmente:

plot(colb[,explanans])


# Modelos de RegresiÃ³n ----------------------------------------------------

# Veamos los modelos propuestos.
# Primero sin poblacion resto, luego con esa:

LinRegA = lm(IDH ~ ., data = colb[,c(1,7)])
LinRegB = lm(IDH ~ ., data = colb[,c(1,7:8)])

#resultados
summary(LinRegA)
```

```
##
## Call:
## lm(formula = IDH ~ ., data = colb[, c(1, 7)])
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.113668 -0.018473  0.001249  0.016927  0.071401
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.634408   0.055163  11.501  1.6e-12 ***
## cabeLog     0.012846   0.004202   3.057  0.00466 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03737 on 30 degrees of freedom
## Multiple R-squared:  0.2376, Adjusted R-squared:  0.2121
## F-statistic: 9.347 on 1 and 30 DF,  p-value: 0.004664
```

```r
summary(LinRegB)
```

```
##
## Call:
## lm(formula = IDH ~ ., data = colb[, c(1, 7:8)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09489 -0.02041  0.00433  0.01740  0.06372
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.765665   0.064813  11.813 1.32e-12 ***
## cabeLog      0.030664   0.006886   4.453 0.000116 ***
## restoLog    -0.029571   0.009626  -3.072 0.004592 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03302 on 29 degrees of freedom
```

```
## Multiple R-squared:  0.4247, Adjusted R-squared:  0.3851
## F-statistic: 10.71 on 2 and 29 DF,  p-value: 0.0003296
```

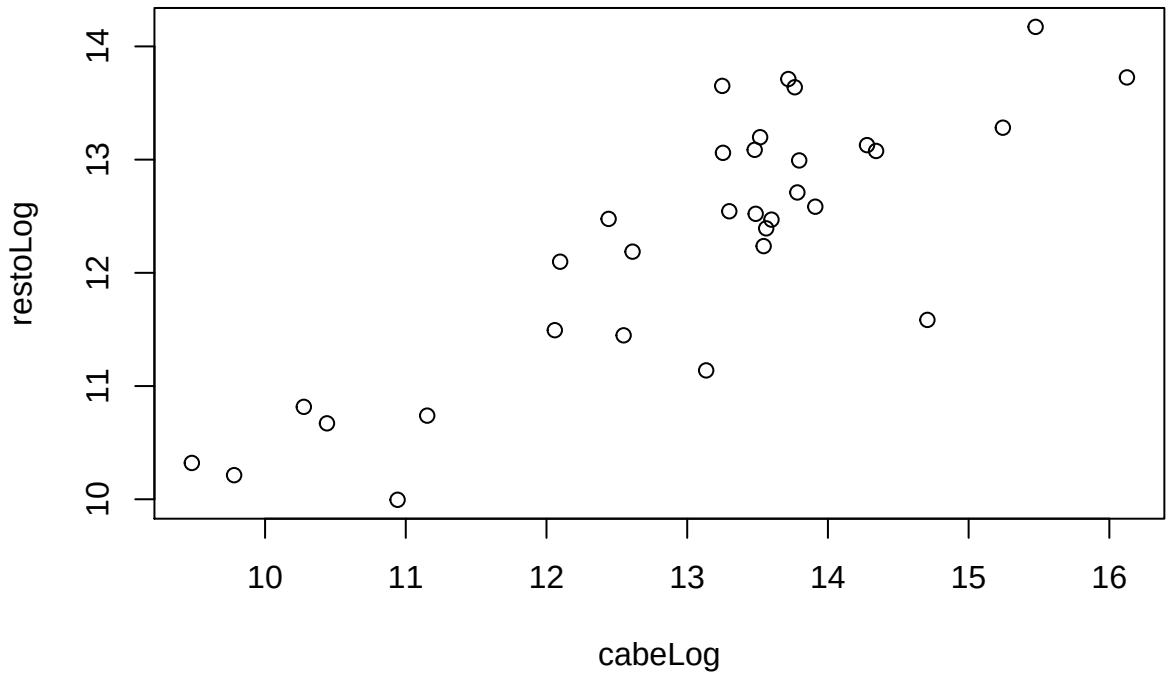
```
# ExploraciÃ³n Espacial ----------------------------------------------------

#Calculemos conglomerados de regiones,
#usando toda la informaciÃ³n de las tres variables.
# usaremos la tecnica de k-means propuesta por MacQueen.


library(rgdal)
```

```
## Warning: package 'rgdal' was built under R version 3.4.4
```

```
## Loading required package: sp
```

```
## Warning: package 'sp' was built under R version 3.4.4
```

```
## rgdal: version: 1.3-3, (SVN revision 759)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20
##  Path to GDAL shared files: C:/Users/nialv_000/Documents/R/win-library/3.4/rgdal/gdal
##  GDAL binary built with GEOS: TRUE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files: C:/Users/nialv_000/Documents/R/win-library/3.4/rgdal/proj
##  Linking to sp version: 1.3-1
```




```
folder='COL_maps'
file='COL_adm1.shp'
```

```
mapaFile=file.path(folder,file)
mapCol <- rgdal::readOGR(mapaFile,stringsAsFactors=F)
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\nialv_000\OneDrive\ANDES\Herramientas compu\Proyecto\ProyectoFinal\ProyectoFinal\C
## with 32 features
## It has 9 fields
## Integer64 fields read as strings:  ID_0 ID_1
```

```
# lo tenemos:
plot(mapCol)
```

```
# veamos que variables hay:

head(mapCol@data)
```

```
##   ID_0 ISO   NAME_0 ID_1     NAME_1       TYPE_1   ENGTYPE_1 NL_NAME_1
## 0   53 COL Colombia    1   Amazonas   ComisarÃa Commissiary      <NA>
## 1   53 COL Colombia    2  Antioquia Departamento  Department      <NA>
## 2   53 COL Colombia    3     Arauca  Intendencia  Intendancy      <NA>
## 3   53 COL Colombia    4 AtlÃ¡ntico Departamento  Department      <NA>
## 4   53 COL Colombia    5   BolÃvar Departamento  Department      <NA>
## 5   53 COL Colombia    6    BoyacÃ¡ Departamento  Department      <NA>
##   VARNAME_1
## 0      <NA>
## 1      <NA>
## 2      <NA>
```

```
## 3      <NA>
## 4      <NA>
## 5      <NA>
```

```
# con esto hagamos el merge:
sub_colb=colb[,c(1:2,7:8)]
mapCol_idh=merge(mapCol,sub_colb, by.x='NAME_1', by.y='Departamento',all.x=F)

# cuantas regiones me quedaron luego del merge?
nrow(mapCol_idh)  # todas!!...
```

```
## [1] 32
```

```
# preparacion para clusterizar:

# que tengo?:
names(mapCol_idh)
```

```
##  [1] "NAME_1"    "ID_0"      "ISO"       "NAME_0"    "ID_1"
##  [6] "TYPE_1"    "ENGTYPE_1" "NL_NAME_1" "VARNAME_1" "IDH"
## [11] "cabeLog"   "restoLog"
```

```
# nombre de la variables que usarÃ©:
dimensions=c("NAME_1","IDH","cabeLog","restoLog")

# creo un nuevo data frame con esas:
dataCluster=mapCol_idh@data[,c(dimensions)]

# como la data es numerica la normalizo (menos la column 1):
dataCluster[,-1]=scale(dataCluster[,-1])

## APLICANDO TECNICA KMEANS

# calculo 3 clusters

resultado=kmeans(dataCluster[,-1],3)

#creo data frame con los clusters:
clusters=as.data.frame(resultado$cluster)

# aÃ±ado columna con nombre de regiones
clusters$NAME_1=dataCluster$NAME_1
names(clusters)=c('cluster','NAME_1')
#hago el merge hacia el mapa:
mapCol_idh=merge(mapCol_idh,clusters, by='NAME_1',all.x=F)

# lo tengo?
names(mapCol_idh)
```

```
##  [1] "NAME_1"    "ID_0"      "ISO"       "NAME_0"    "ID_1"
##  [6] "TYPE_1"    "ENGTYPE_1" "NL_NAME_1" "VARNAME_1" "IDH"
## [11] "cabeLog"   "restoLog"  "cluster"
```

```
## a pintar:

library(RColorBrewer)
library(classInt)
```

```
## Warning: package 'classInt' was built under R version 3.4.4
## Loading required package: spData
## Warning: package 'spData' was built under R version 3.4.4
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source'))`
```

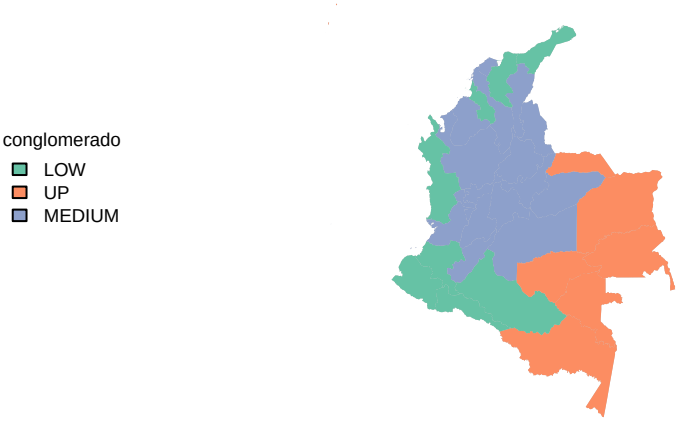
```
#variable a colorear
varToPLot=mapCol_idh$cluster

# decidir color:
unique(varToPLot)
```

```
## [1] 2 3 1
```

```
aggregate(mapCol_idh@data[,c(10,11,12)],
          by=list(mapCol_idh@data$cluster),FUN=mean)
```

```
##   Group.1       IDH  cabeLog restoLog
## 1       1 0.7560000 13.05663 12.80485
## 2       2 0.7825714 10.58974 10.60684
## 3       3 0.8313529 14.03019 12.74569
```

```
#preparo colores
numberOfClasses = length(unique(varToPLot))
colorForScale='Set2'
paleta = brewer.pal(numberOfClasses, colorForScale)

# grafico mapa basico
plot(mapCol,col='grey',border=0)

# grafico mapa cluster
plot(mapCol_idh, col = paleta[varToPLot],border=F,add=T)
legend('left', legend = c("LOW","UP","MEDIUM"),
       fill = paleta,
       cex = 0.6,
       bty = "n",
       title="conglomerado")
```


conglomerado
LOW
UP
MEDIUM