

# Tarea3

*Jesus Pacheco*

*July 16, 2017*

## I

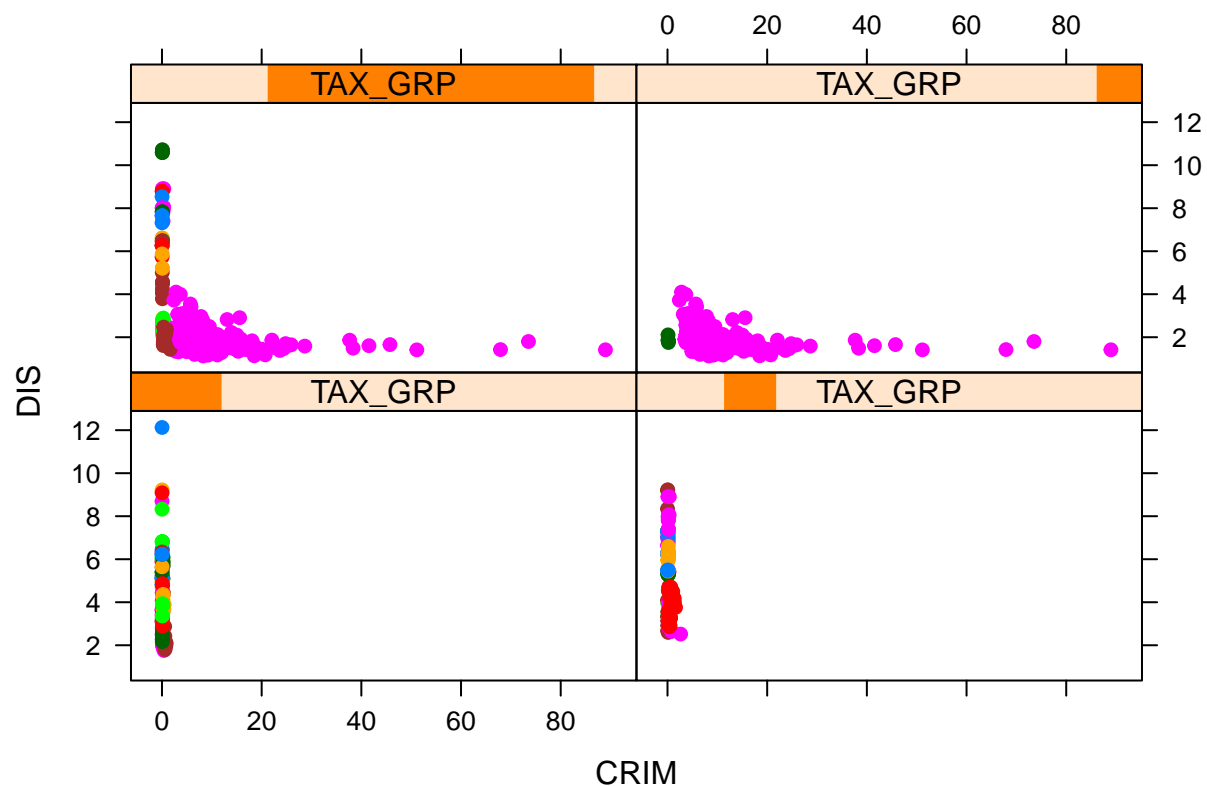
```
# carga del paquete {lattice}
suppressWarnings(suppressMessages(library(lattice)))
# carga de los datos
housing <- read.csv("data/housing.csv", head = FALSE)
names(housing) <- c("CRIM", "ZN", "INDUS", "NOX", "RM",
                    "AGE", "DIS", "RAD", "TAX", "PTRATIO")
```

### I.1

Se explora la relación entre las tasa de criminalidad CRIM, las tasa de impuestos a la propiedad TAX y la distancia a los centros de empleo DIS.

Se utiliza la variable TAX para crear la segmentación en grupos y crear el `xyplot()`:

```
housing$TAX_GRP <- equal.count(housing$TAX, number = 4, overlap = 0)
xyplot(DIS ~ CRIM | TAX_GRP, data = housing, pch = 19,
        groups = TAX_GRP)
```

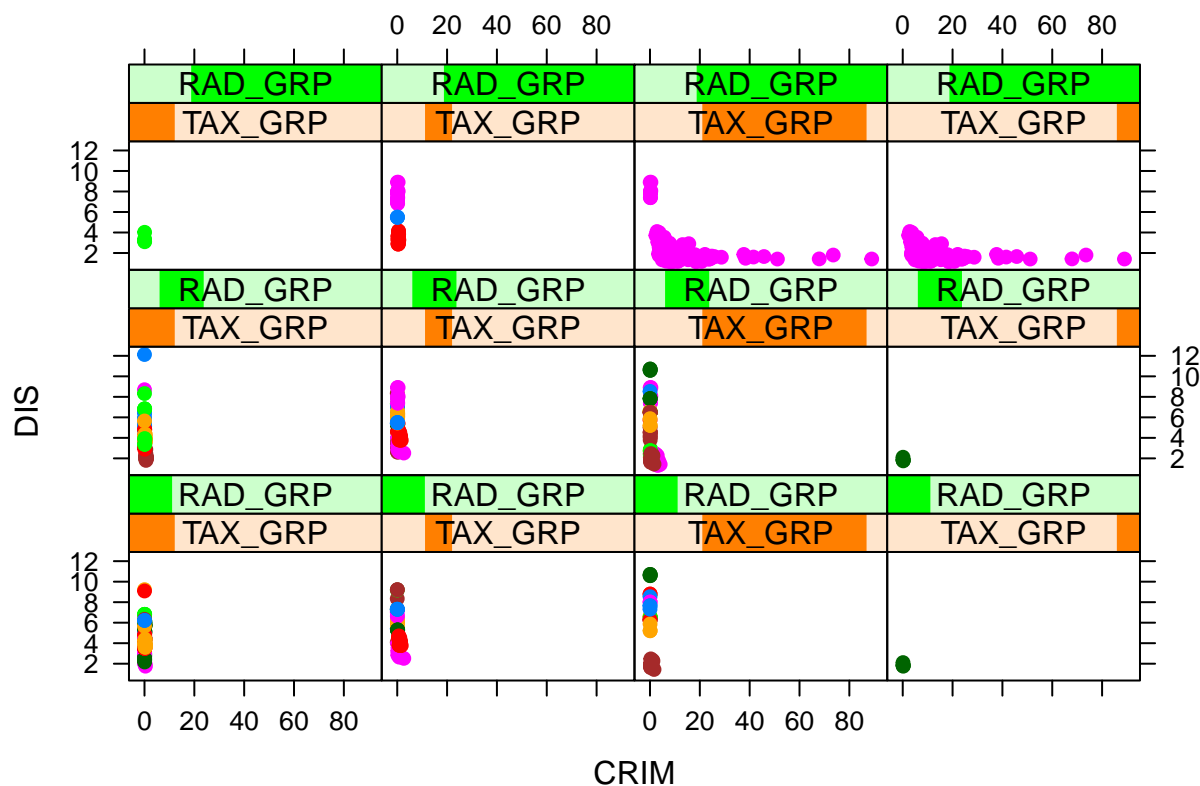


Se puede apreciar que en los grupos con menores tasas de impuesto la relación entre criminalidad y distancia es muy baja. En las zonas con mayores tasas impositivas hay una relación inversa debil entre la tasas de criminalidad y la distancia a los centros de empleo.

## II.2

Se agrega una segunda variable cualitativa descrita por la accesibilidad a autopistas RAD:

```
housing$RAD_GRP <- equal.count(housing$RAD, number = 3, overlap = 0)
xyplot(DIS ~ CRIM | TAX_GRP * RAD_GRP,
       data = housing, pch = 19,
       groups = TAX_GRP)
```



En general no se aprecia ningún aporte significativo por la adición de la nueva dimensión al gráfico; se puede apreciar solamente que los mismos comentarios realizados en el inciso anterior para las zonas con alta tasa de impuestos se puede realizar para las zonas con alto índice de accesibilidad a pistas.

Se puede corroborar que efectivamente mucha de la información contenida por la variable **TAX** se puede extraer de la variable **RAD** a partir de la alta correlación entre ambas variables:

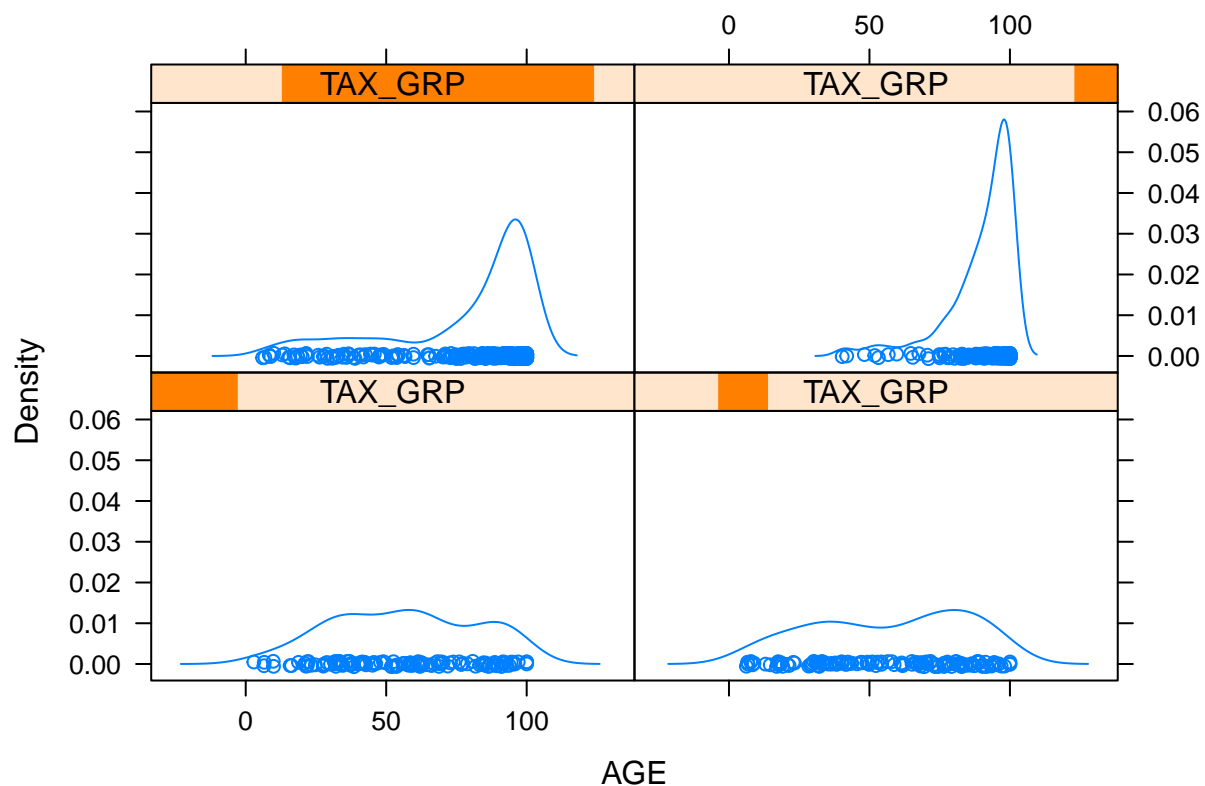
```
cor(housing[,c("RAD", "TAX")])
```

```
##          RAD          TAX
## RAD 1.0000000 0.9102282
## TAX 0.9102282 1.0000000
```

### I.3

Ahora se muestra la distribución de edad **AGE** segmentando por la tasa de impuestos.

```
densityplot( ~ AGE | TAX_GRP, data = housing)
```



Note que la distribución de antigüedad de los inmuebles es bastante pareja para las zonas con tasas impositivas menores, y en zonas con tasas impositivas mayores hay mucho peso para antigüedades mayores.

## II.

### III. 1

```
# carga de la libreria {maps}
library(maps)
# lectura de los datos
WDS2014 <- data.frame(data.table::fread("data/WDS2014v3.csv"))
```

Se inicializa el objeto necesario para dibujar el mapa y se procede a la asignacion de los valores de interés a cada país:

```
# set up del mapa
x <- map(plot = FALSE)
x$measure <- array(NA,dim=length(x$names))
x$idx <- array(NA,dim=length(x$names))

# agregar datos para paises incluidos en el dataset
for(i in 1:length(x$names)) {
  idx <- grepl(x$names[i], WDS2014$Country.Name,
              ignore.case = T)
  if (any(idx)) {
```

```

    x$idx[i] <- which(idx)[1]
    x$measure[i] <- WDS2014$ABS[x$idx[i]]
  }
}

# asignacion manual de valores para uk, usa, japon, y otros
`%like%` <- function(x, pattern, ...) grepl(pattern, x, ignore.case = T, ...)
x$measure[x$names %like% "^USA"] <- WDS2014$ABS[WDS2014$Country.Name == "United States"]
x$measure[x$names %like% "^UK[~r]"] <- WDS2014$ABS[WDS2014$Country.Name == "United Kingdom"]
x$measure[x$names %like% "japan"] <- WDS2014$ABS[WDS2014$Country.Name == "Japan"]
x$measure[x$names %like% "south korea"] <- WDS2014$ABS[WDS2014$Country.Name == "Korea, Rep."]
x$measure[x$names %like% "north korea"] <- WDS2014$ABS[WDS2014$Country.Name %like% "Korea.*Dem.*People"]
x$names[x$names %like% "new zealand"] <- WDS2014$ABS[WDS2014$Country.Name == "New Zealand"]

```

Finalmente se especifican las opciones de color, partición de los grupos y graficación del mapa:

```

#Definición de color
library(RColorBrewer)
colors = brewer.pal(5,"GnBu")

#Vector de colores en función al valor de EVT. Se crean 5 grupos
sd <- data.frame(col=colors, values <- seq(min(x$measure[!is.na(x$measure)]), max(x$measure[!is.na(x$measure)]))
#Vector de valores en blanco para los países que no se encuentren
sc<-array("#FFFFFF",dim=length(x$names))

#Asignar el color según el intervalo en el cual se encuentre el país
for (i in 1:length(x$measure))
  if(!is.na(x$measure[i]))
    sc[i]=as.character(sd$col[findInterval(x$measure[i],sd$values)])

#2-Layout para poner la leyenda a la derecha del mapa
layout(matrix(data=c(2,1), nrow=1, ncol=2), widths=c(8,1), heights=c(8,1))

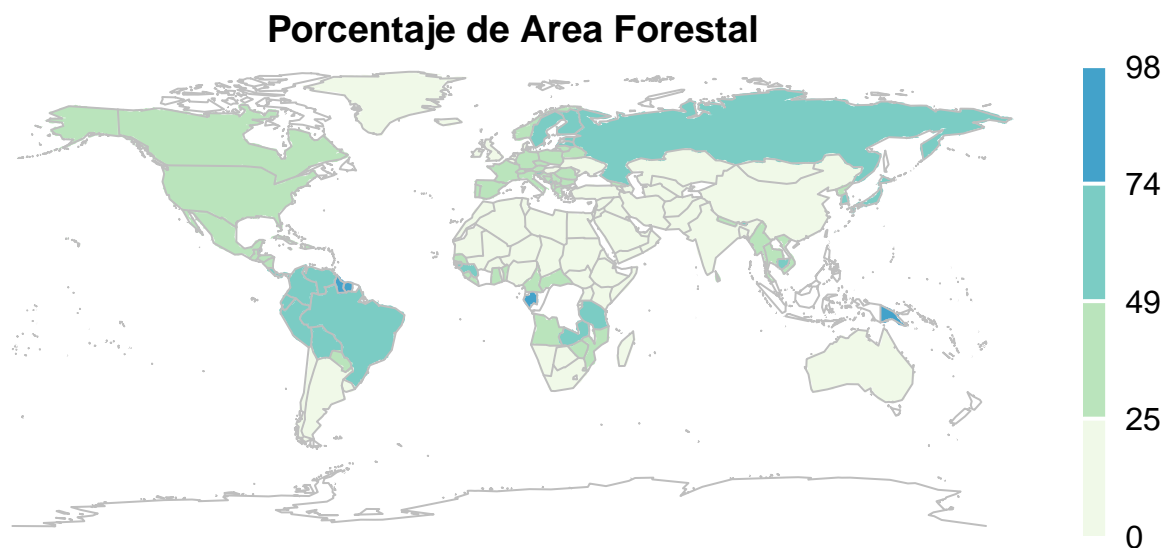
# Escala de colores (leyenda)
breaks<-sd$values
par(mar = c(10,1,4,5),oma=c(0.2,0.2,0.2,0.2),mex=0.5)

#leyenda como imagen
image(x=1, y=0:length(breaks),z=t(matrix(breaks))*1.001,col=colors[1:length(breaks)-1],axes=FALSE, break=FALSE)

axis(side=4,at=0:(length(breaks)-1), labels = round(breaks), col="white", las=1)
#simular líneas separadoras
abline(h=c(1:length(breaks)),col="white",lwd=2,xpd=F)

#Mapa solo colores
map(col=sc,fill=TRUE,lty="blank")
#silueta
map(add=TRUE,col="gray",fill=FALSE)
title("Porcentaje de Area Forestal")

```



Se puede observar valores más altos en Sudamérica.

## III.2

Se copia el código:

```
#Instalar y cargar el mapa
# install.packages("RgoogleMaps", dependencies = TRUE)
library(RgoogleMaps)
#Datos de capitales
datos <- read.csv(file="data/world_cities.csv", head=TRUE, sep=",", dec = ".")
capital <- datos[datos$city=="Ottawa",]
lat <- c(capital$lat -20, capital$lat+20) #Rango en y
lon <- c(capital$lng-20, capital$lng+20) #Rango en x
center = c(capital$lat, capital$lng) #Centro del gráfico
zoom <- 5 #zoom: 1 = Todo el globo,
#Mapa
terrmap <- GetMap(center=center,
                  zoom=zoom,
                  maptype= "roadmap" ,
                  destfile = "CA.png") #graficar mapa
PlotOnStaticMap(terrmap)
text(x=1, y= capital$lat, labels = "EVT:=82", cex = 0.8)
```



A partir de los datos cargados, se escogen las capitales de Norte América

```
# carga de dplyr
suppressMessages(suppressWarnings(library(dplyr)))

# filtrado de ciudades de interés
datos2 <- dplyr::filter(datos,
  grepl("mexico city|washington, d\\.c\\.|ottawa",
    city,
    ignore.case = T))

# cálculo de latitudes/longitudes promedio para insumo de GetMap()
range1 <- datos2 %>% filter(iso3 %in% c("USA", "MEX", "CAN")) %>%
  summarise(avg_lat = mean(lat), avg_lng = mean(lng))
center <- with(range1, c(avg_lat, avg_lng))

# extraccion de etiquetas
datos2 <- mutate(datos2, life_exp = c(WDS2014$EVT[WDS2014$Country.Name == "Canada"],
  WDS2014$EVT[WDS2014$Country.Name == "Mexico"],
  WDS2014$EVT[WDS2014$Country.Name == "United States"]))

# creacion del mapa
termap <- GetMap(center=center,
  zoom=4,
  maptype= "roadmap" ,
  destfile = "CA.png") #graficar mapa

tmp <- PlotOnStaticMap(termap, destfile = "MyTile1.png",
```

```
lat = datos2$lat,
lon = datos2$lng,
pch = 15, col = "red", add = FALSE)
```

```
PlotOnStaticMap(terrmap,
  lat = datos2$lat + 1,
  lon = datos2$lng + 5,
  cex = 0.7,
  FUN = pryr::partial(text,
    labels = paste("EVT=", format(datos2$life_exp))),
  add = TRUE)
```

