

# Curso sobre el software estadístico R: La librería googleVis

Ponente: Carlos Pérez Glez.

Entidades participantes en el curso:



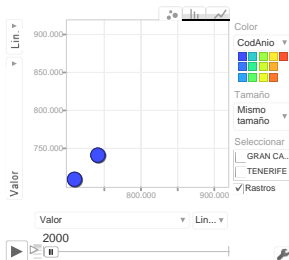
## La librería googleVis

- 1 googleVis es una librería que proporciona una interfaz entre R y las herramientas Google Chart.
- 2 Las funciones de la librería permiten representar datos con Google Chart Tools sin necesidad de hacer un “upload” de los datos a Google. Lo que sí se hace es referenciar a funciones javascript alojadas en Google.
- 3 La salida de las funciones googleVis es un código html que contiene los datos y referencias a las funciones javascript. Por tanto, para ver la salida se necesita un navegador con Flash y conexión a internet para renderizar el gráfico.  
Website : <http://code.google.com/p/google-motion-charts-with-r/> Website (blog): <http://www.r-bloggers.com/search/googlevis>

# Gráfico gvisMotionChart()

Para representar un gráfico dinámico que responda a los eventos de usuario:

```
library(googleVis)
data.pob.capitales <- data.pob.municipios[data.pob.municipios$Indicador == "Cifras absolutas" &
  data.pob.municipios$CodMunicipio %in% c("ES705", "ES709"), ]
motion.capitales <- gvisMotionChart(data.pob.capitales, idvar = "Municipio",
  timevar = "Anio", options = list(height = 350, width = 400))
plot(motion.capitales)
# print(motion.capitales, tag='chart')
```



## Gráfico gvisMotionChart(): código XML

Podemos ver el código XML que contiene este objeto y lo podemos insertar como un widget

```
# create gadget
cat(createGoogleGadget(motion.capitales), file = "motionchart.xml")
# se sube el gadget a algún repositorio y se vincula en alguna página
# https://sites.google.com/site/cpgonzal/
```

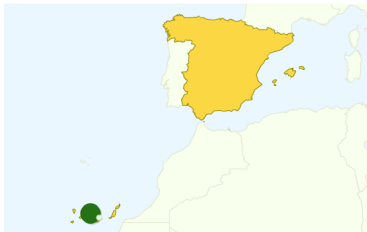
```
## <?xml version="1.0" encoding="UTF-8" ?>
## <Module>
##   <ModulePrefs title="MotionChartIDfec48b13dbd" />
##   <Content type="html">
##     <![CDATA[
##       <!-- MotionChart generated in R 3.0.0 by googleVis 0.4.2 pa
```

## Gráfico gvisGeoMap()

Podemos representar datos geo-espaciales utilizando las funciones de mapas:

```
data.coords <- read.table("datos_coordenadas_canarias.txt", header = TRUE, sep = ",")  
  
idx <- match(data.pob.capitales$CodMunicipio, data.coords$cod)  
  
data.pob.capitales$LatLong <- paste(data.coords[idx, "latitude"], data.coords[idx,  
  "longitude"], sep = ":")  
  
map.capitales <- gvisGeoMap(data.pob.capitales[data.pob.capitales$Anio == "2012",  
  ], locationvar = "LatLong", numvar = "Valor", options = list(region = "ES",  
  dataMode = "markers", showLegend = FALSE))  
  
# Display chart  
plot(map.capitales)
```

Esta función no es muy adecuada para regiones que no estén codificadas como ISO 3166-1 ([http://www.iso.org/iso/iso-3166-1\\_decoding\\_table.html](http://www.iso.org/iso/iso-3166-1_decoding_table.html))

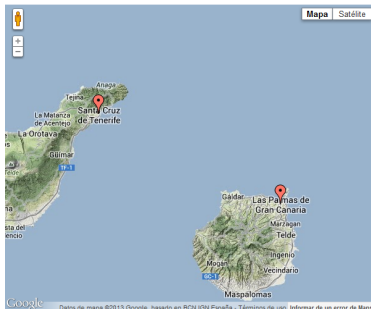


# Gráfico gvisMap()

Podemos mejorar la representación utilizando gvisMap:

```
data.pob.capitales$Tip <- paste("Poblacion:", data.pob.capitales$Valor)

map.capitales <- gvisMap(data.pob.capitales[data.pob.capitales$Anio == "2012",
], "LatLong", "Tip", options = list(enableScrollWheel = TRUE, mapType = "terrain",
useMapTypeControl = TRUE))
# Display chart
plot(map.capitales)
```



# Tabla gvisTable()

También se puede mostrar una representación tabular de datos

```
table.capitales <- gvisTable(data.pob.capitales, options = list(width = 800,
  height = 470))
# Display chart
plot(table.capitales)

table.capitales <- gvisTable(data.pob.capitales, options = list(width = 800,
  height = 470, page = "enable"))
# Display chart
plot(table.capitales)
```

CodMunicipio	Municipio	CodAño	Año	CodIndicador	Indicador	Valor	LatLong	Tip
ES705	GRAN CANARIA	2012	2012	000000	Cifras absolutas	852225	28.1157,-15.4408	Poblacion 852225
ES705	GRAN CANARIA	2011	2011	000000	Cifras absolutas	850391	28.1157,-15.4408	Poblacion 850391
ES705	GRAN CANARIA	2010	2010	000000	Cifras absolutas	845676	28.1157,-15.4408	Poblacion 845676
ES705	GRAN CANARIA	2009	2009	000000	Cifras absolutas	838397	28.1157,-15.4408	Poblacion 838397
ES705	GRAN CANARIA	2008	2008	000000	Cifras absolutas	829597	28.1157,-15.4408	Poblacion 829597
ES705	GRAN CANARIA	2007	2007	000000	Cifras absolutas	815379	28.1157,-15.4408	Poblacion 815379
ES705	GRAN CANARIA	2006	2006	000000	Cifras absolutas	807049	28.1157,-15.4408	Poblacion 807049
ES705	GRAN CANARIA	2005	2005	000000	Cifras absolutas	802247	28.1157,-15.4408	Poblacion 802247
ES705	GRAN CANARIA	2004	2004	000000	Cifras absolutas	790360	28.1157,-15.4408	Poblacion 790360
ES705	GRAN CANARIA	2003	2003	000000	Cifras absolutas	789908	28.1157,-15.4408	Poblacion 789908
ES705	GRAN CANARIA	2002	2002	000000	Cifras absolutas	771333	28.1157,-15.4408	Poblacion 771333

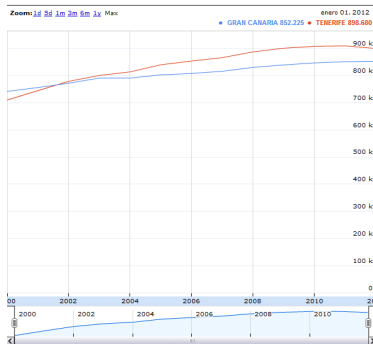
## Gráfico gvisAnnotatedTimeLine()

Vemos un ejemplo de gráficos con serie temporal o líneas:

```
data.pob.capitales$Date <- as.Date(paste(data.pob.capitales$Anio, "-01-01",
  sep = ""))

line.capitales <- gvisAnnotatedTimeLine(data.pob.capitales, datevar = "Date",
  numvar = "Valor", idvar = "Municipio", date.format = "%d/%m/%Y", options = list(legendPosition = "newRow",
  displayExactValues = "TRUE", width = 600, height = 550))

# Display chart
plot(line.capitales)
```





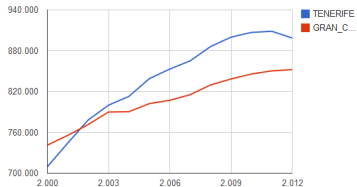
## Gráfico gvisLineChart()

Vemos un ejemplo de gráficos de líneas:

```
## Line chart
data.pob.capitales2<-data.frame(Anio=data.pob.capitales[data.pob.capitales$Municipio==" TENERIFE","Anio"],
                                TENERIFE=data.pob.capitales[data.pob.capitales$Municipio==" TENERIFE","Valor"],
                                GRAN_CANARIA=data.pob.capitales[data.pob.capitales$Municipio==" GRAN CANARIA","Valor"])

line.capitales <- gvisLineChart(data.pob.capitales2,xvar="Anio",yvar=c("TENERIFE","GRAN_CANARIA"),
                                options=list(legend='right', width=600, height=400))

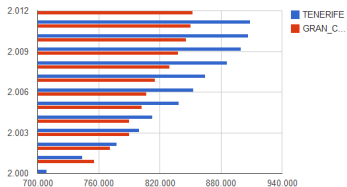
plot(line.capitales)
```



## Gráfico gvisBarChart()

Vemos un ejemplo de gráficos de barras:

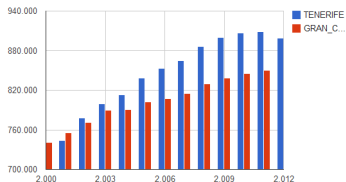
```
## Bar chart
bar.capitales <- gvisBarChart(data.pob.capitales2, xvar = "Anio", yvar = c("TENERIFE",
  "GRAN_CANARIA"), options = list(legend = "right", width = 600, height = 400))
plot(bar.capitales)
```



## Gráfico gvisColumnChart()

Podemos representar gráficos columnas:

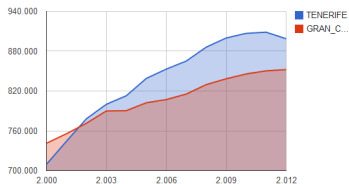
```
## Column chart
col.capitales <- gvisColumnChart(data.pob.capitales2, xvar = "Anio", yvar = c("TENERIFE",
  "GRAN_CANARIA"), options = list(legend = "right", width = 600, height = 400))
plot(col.capitales)
```



## Gráfico gvisAreaChart()

Y gráficos de áreas:

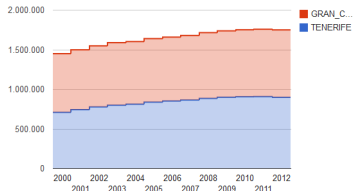
```
## Area chart
area.capitales <- gvisAreaChart(data.pob.capitales2, xvar = "Anio", yvar = c("TENERIFE",
  "GRAN_CANARIA"), options = list(legend = "right", width = 600, height = 400))
plot(area.capitales)
```



## Gráfico gvisSteppedAreaChart()

Veamos los gráficos de escalera:

```
## Stepped Area Chart
step.capitales <- gvisSteppedAreaChart(data.pob.capitales2, xvar = "Anio", yvar = c("TENERIFE",
  "GRAN_CANARIA"), options = list(isStacked = TRUE, width = 600, height = 400))
plot(step.capitales)
```



# Gráfico gvisScatterChart()

También se pueden representar gráficos de dos variables:

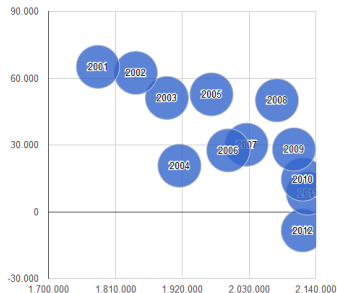
```
data.pob.canarias <- data.pob.municipios[data.pob.municipios$Indicador %in%  
  c("Cifras absolutas", "Variación interanual") & data.pob.municipios$CodMunicipio %in%  
  c("ES70"), ]  
  
data.pob.canarias2 <- data.frame(CIFRAS_ABSOLUTAS = data.pob.canarias[data.pob.canarias$Indicador ==  
  "Cifras absolutas", "Valor"], VAR_INTERANUAL = data.pob.canarias[data.pob.canarias$Indicador ==  
  "Variación interanual", "Valor"])  
  
## Scatter chart  
scatter.canarias <- gvisScatterChart(data.pob.canarias2, options = list(legend = "none",  
  pointSize = 2, vAxis = "{title:'Variacion interanual'}", hAxis = "{title:'Poblacion absoluta'}",  
  title = "Comparacion de poblacion absoluta y var. inter.", width = 600,  
  height = 600))  
plot(scatter.canarias)
```



# Gráfico gvisBubbleChart()

## Un gráfico de burbujas:

```
data.pob.canarias2 <- data.frame(CIFRAS_ABSOLUTAS = data.pob.canarias[data.pob.canarias$Indicador ==  
  "Cifras absolutas", "Valor"], VAR_INTERANUAL = data.pob.canarias[data.pob.canarias$Indicador ==  
  "Variación interanual", "Valor"], ANIO = data.pob.canarias[data.pob.canarias$Indicador ==  
  "Variación interanual", "Anio"])  
  
## Bubble chart  
bubble.canarias <- gvisBubbleChart(data.pob.canarias2, idvar = "ANIO", xvar = "CIFRAS_ABSOLUTAS",  
  yvar = "VAR_INTERANUAL", options = list(width = 600, height = 600))  
plot(bubble.canarias)
```



## Gráfico gvisBubbleChart()

Otro ejemplo del gráfico de burbujas:

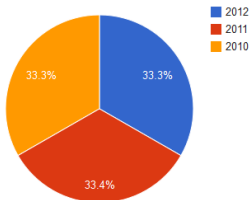
```
data.pob.capitales <- data.pob.municipios[data.pob.municipios$Indicador %in%  
  c("Cifras absolutas", "Variación interanual") & data.pob.municipios$CodMunicipio %in%  
  c("ES705", "ES709") & data.pob.municipios$Anio %in% c("2010", "2011", "2012"),  
  ]  
  
data.pob.capitales2 <- data.frame(CIFRAS_ABSOLUTAS = data.pob.capitales[data.pob.capitales$Indicador ==  
  "Cifras absolutas", "Valor"], VAR_INTERANUAL = data.pob.capitales[data.pob.capitales$Indicador ==  
  "Variación interanual", "Valor"], ANIO = data.pob.capitales[data.pob.capitales$Indicador ==  
  "Variación interanual", "Anio"], ISLA = data.pob.capitales[data.pob.capitales$Indicador ==  
  "Variación interanual", "Municipio"])  
  
bubble.canarias <- gvisBubbleChart(data.pob.capitales2, idvar = "ISLA", xvar = "CIFRAS_ABSOLUTAS",  
  yvar = "VAR_INTERANUAL", colorvar = "ANIO", sizevar = "VAR_INTERANUAL",  
  options = list(width = 300, height = 300))  
  
plot(bubble.canarias)
```



## Gráfico gvisPieChart()

Un gráfico de sectores (o de tarta):

```
data.pob.canarias <- data.pob.municipios[data.pob.municipios$Indicador %in%  
  c("Cifras absolutas") & data.pob.municipios$CodMunicipio %in% c("ES70") &  
  data.pob.municipios$Anio %in% c("2010", "2011", "2012"), ]  
  
data.pob.canarias2 <- data.frame(ANIO = as.character(data.pob.canarias[, "Anio"]),  
  CIFRAS_ABSOLUTAS = data.pob.canarias[, "Valor"])  
  
## Pie chart  
pie.canarias <- gvisPieChart(data.pob.canarias2, labelvar = "ANIO", numvar = "CIFRAS_ABSOLUTAS",  
  options = list(width = 600, height = 400))  
plot(pie.canarias)
```



## Gráfico gvisGauge()

Y los gráficos de calibración:

```
## Gauge
gauge.canarias <- gvisGauge(data.pob.canarias2, options = list(fontSize = 9,
  min = 2e+06, max = 2200000, redFrom = 2e+06, redTo = 2100000, yellowFrom = 2100000,
  yellowTo = 2150000, greenFrom = 2150000, greenTo = 2200000, width = 800,
  height = 320))
plot(gauge.canarias)
```



# Gráfico gvisOrgChart()

También se pueden representar esquemas organizativos:

```
data.org.canarias <- data.frame(ISLA = c("CANARIAS", "LANZAROTE", "FUERTEVENTURA",  
    "GRAN CANARIA", "TENERIFE", "LA GOMERA", "LA PALMA", "EL HIERRO"), PARENT = c(NA,  
    "GRAN CANARIA", "GRAN CANARIA", "CANARIAS", "CANARIAS", "TENERIFE", "TENERIFE",  
    "TENERIFE"), CODE = c("ES70", "ES708", "ES704", "ES705", "ES709", "ES706",  
    "ES707", "ES703"))  
  
## Org chart  
org.canarias <- gvisOrgChart(data.org.canarias, idvar = "ISLA", parentvar = "PARENT",  
    tipvar = "CODE", options = list(width = 600, height = 210, size = "large",  
    allowCollapse = TRUE))  
  
plot(org.canarias)
```

