

TAREA_CURSO_R

Visualización

Sthefanny Meliza Villa Toledo

2023-03-03

```
# Instalar librerías requeridas
```

```
library(readr)
library(RCurl)
library(ggplot2)
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
```

```
library(ggpubr)
library(datasets)
library(gridExtra)
```

```
##
```

```
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      combine
```

R Markdown

Sthefanny Villa - sthefanny.villa@est.ikiam.edu.ec - Universidad Regional Amazónica IKIAM, Tena (Ecuador)

Tarea Visualizacion

¿Qué es una buena visualización?

Es algo que cautiva a la audiencia y refuerza el mensaje que se desea comunicar. Para ello se debe identificar lo siguiente:

- El mensaje utilizando la visualización exploratoria.
- La mejor representación para el mensaje.
- La apariencia que mejor resalta los hallazgos.
- Mensajes adicionales apropiados.

¿Cómo se logra una buena visualización?

Empleando los elementos adecuados en el diseño de la visualización de datos. Estos elementos fueron compilados y establecidos por Leland Wilkinson en su libro “The grammar of graphics: statistics and computing”. Se tiene dos tipos de visualizaciones concatenadas: **Visualización de análisis - visualización de presentación**

- **Visualización de Análisis**, para tener una primera impresión de los patrones, tendencias que ayudan a describir e interpretar el tipo de data. Aquí aplica la regla de 20 - 80, donde se tiene un **20%** de tiempo en análisis para obtener el **80%** de resultados esperados.
- **Visualización de Presentación**, donde se realiza el diseño (tipos de geom(), colores, tamaños, ejes, leyendas, agrupaciones) adecuado, se resalta lo importante y se ilustra un mensaje. Aquí se aplica la regla 80 - 20, donde se dedica un **80%** de tiempo en la actividad para obtener el **80%** de resultados esperados. `ggplot(data =) + (mapping = aes(), stat = , position =) + + +`

Instrucciones

Use un documento tipo *.Rmd* o *.qmd* y genere un informe *html*. Estos dos documentos (Rmd y html) guárdelos con control de cambios en un repositorio de GitHub y **pegue solamente el enlace del repositorio** en esta entrega. La entrega debe contener los códigos para obtener las siguientes figuras:

Data

Son datos preinstalados en R.

```
## **ToothGrowth:** contiene la longitud de los odontoblastos (células responsables del crecimiento den  
data("ToothGrowth")  
head(ToothGrowth, n=4)
```

```
##      len supp dose  
## 1   4.2   VC  0.5  
## 2  11.5   VC  0.5  
## 3   7.3   VC  0.5  
## 4   5.8   VC  0.5
```

```
###mtcars:** contiene el consumo de combustible y 10 aspectos del diseño y rendimiento del automóvil p
```

```
data("mtcars")
head(mtcars, n=4)
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1   4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1   4    4
## Datsun 710     22.8   4  108  93 3.85 2.320 18.61  1  1   4    1
## Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0   3    1
```

```
##Diamonds: contiene las mediciones sobre 10 variables diferentes para 53.940 diamantes. Tiene 53.940 o
```

```
data("diamonds")
head(diamonds, n=4)
```

```
## # A tibble: 4 x 10
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal    E      SI2      61.5    55   326   3.95   3.98   2.43
## 2  0.21 Premium E      SI1      59.8    61   326   3.89   3.84   2.31
## 3  0.23 Good    E      VS1      56.9    65   327   4.05   4.07   2.31
## 4  0.29 Premium I      VS2      62.4    58   334   4.2    4.23   2.63
```

1. Réplica de la figura compuesta.

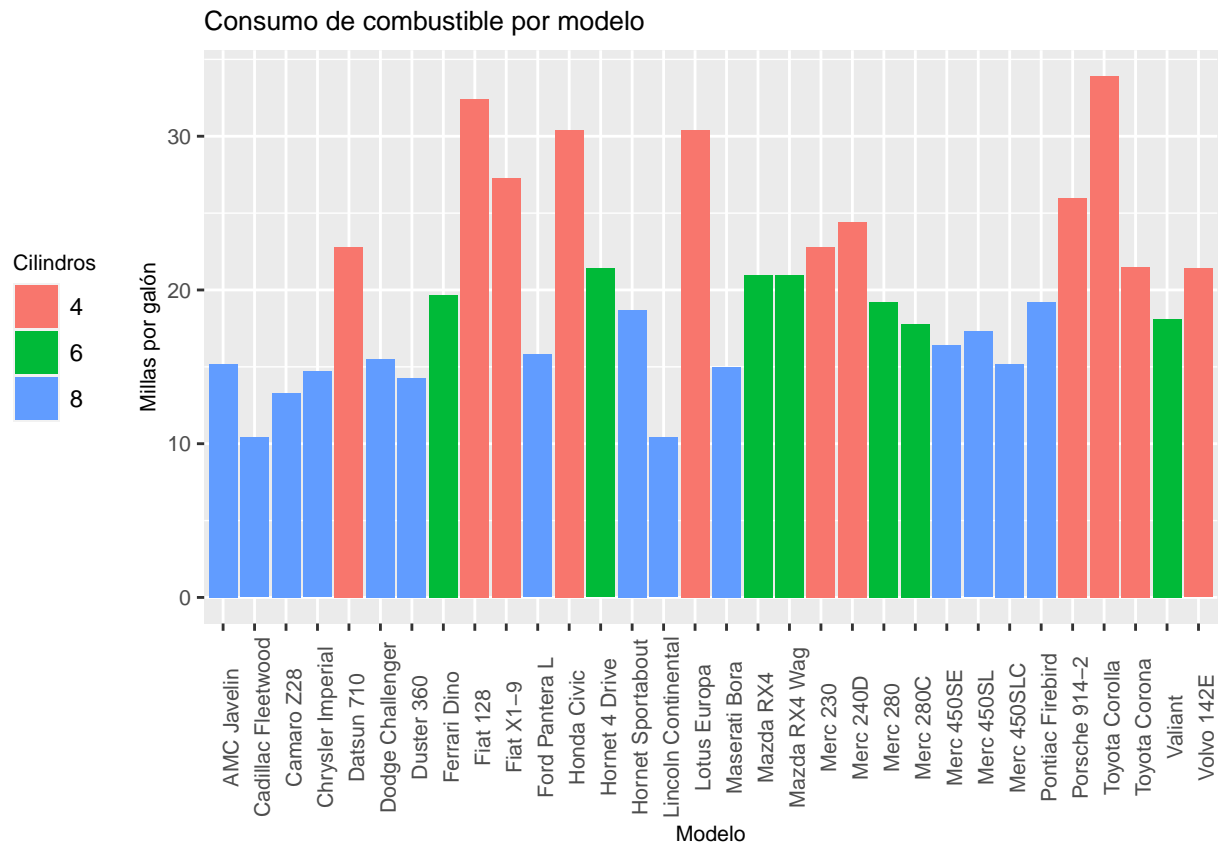
```
## Cambio en cyl a variable categorica y no numerica
```

```
data("mtcars")
mtcars$name = rownames(mtcars)
mtcars$cyl = as.factor(mtcars$cyl)

head(mtcars[, c("name", "wt", "mpg", "cyl")])
```

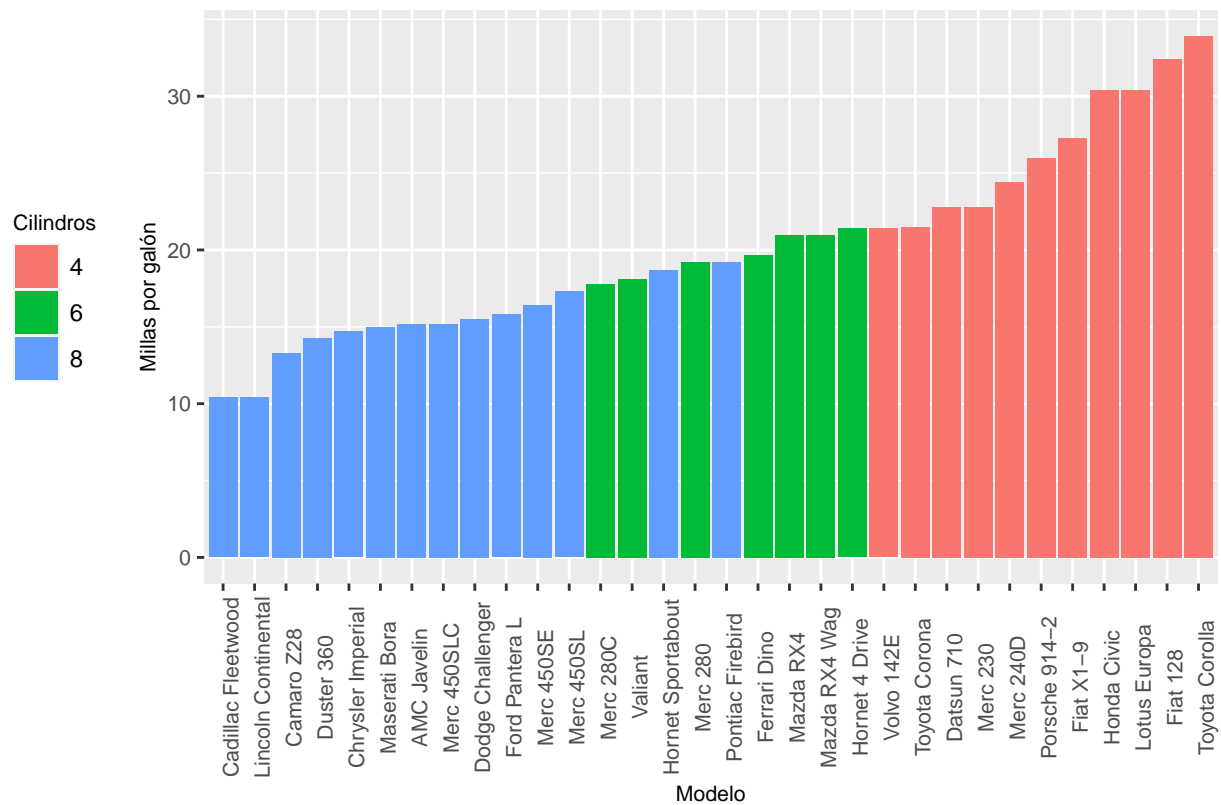
```
#Barplot por registro
```

```
## se definen los datos y las variables + se crea las barras + se mueve la leyenda a la izquierda + se
cars1<-ggplot(mtcars, aes(x = name, y = mpg, fill = cyl)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "left",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Consumo de combustible por modelo",
        x = "Modelo", # se agrega el nombre del eje x
        y = "Millas por galón", # se agrega el nombre del eje y
        fill = "Cilindros")+ # se agrega el nombre de la leyenda
  theme(axis.text = element_text(size = 8),
        axis.title = element_text(size = 8),
        title = element_text(size = 8))
cars1
```



```
cars2<-ggplot(mtcars, aes(x = reorder(name, mpg), y = mpg, fill = cyl)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "left",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Consumo de combustible por modelo y cilindro", x = "Modelo", y = "Millas por galón",
        fill = "Cilindros")+
  theme(axis.text = element_text(size = 8),
        axis.title = element_text(size = 8),
        title = element_text(size = 8))
cars2
```

Consumo de combustible por modelo y cilindro



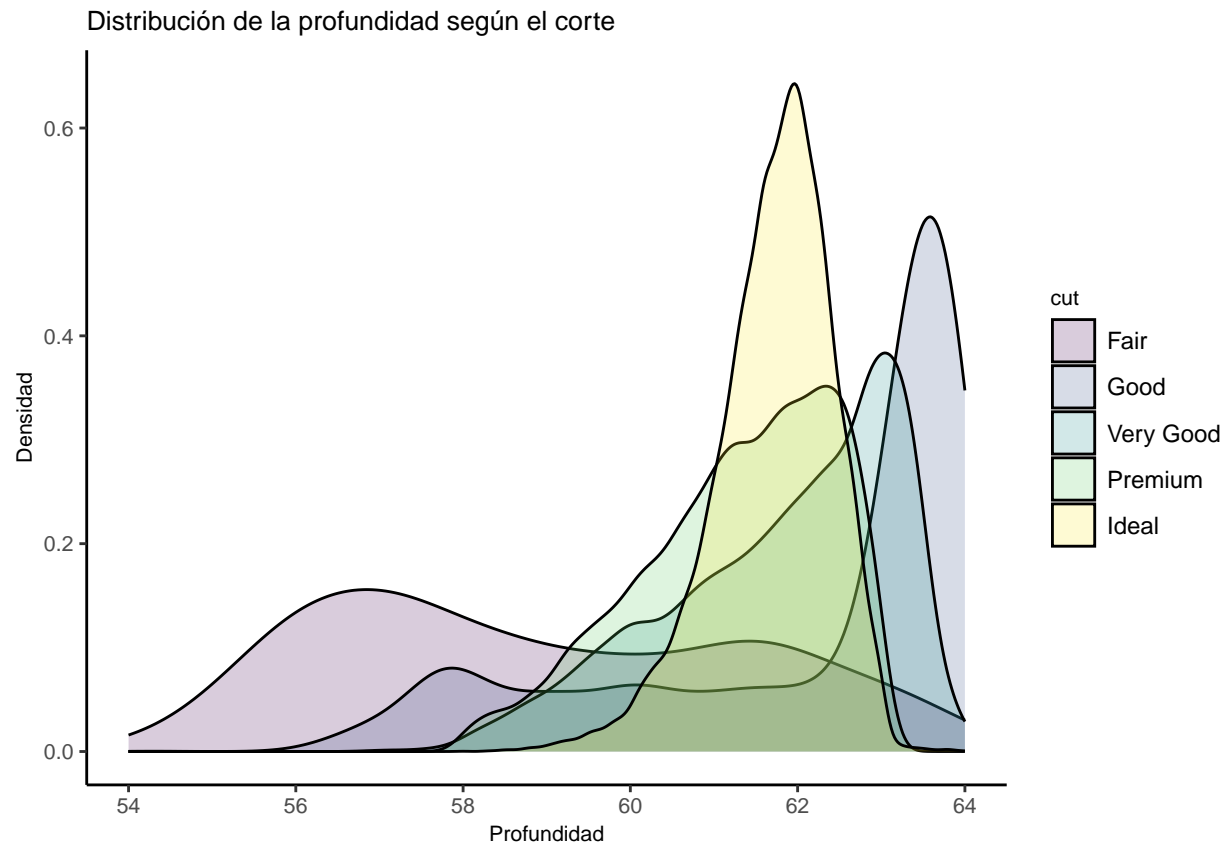
##Densidades

```
data("diamonds")
```

```
dens<- ggplot(diamonds, aes(x = depth, fill = cut)) + geom_density(alpha = 0.2) + theme(legend.position = "bottom",
axis.text.x = element_text(angle = 90)) +
labs(title = "Distribución de la profundidad según el corte", x = "Profundidad", y = "Densidad") +
theme_classic() +
xlim(54,64) +
theme(axis.text = element_text(size = 8),
axis.title = element_text(size = 8),
title = element_text(size = 8))
```

dens

Warning: Removed 1885 rows containing non-finite values ('stat_density()').



Gráfica de dispersión

```
dispercion <- ggscatter(mtcars, x = "wt", y = "mpg", add = "reg.line", conf.int = TRUE, color = "cyl",
  stat_cor(aes(color = cyl), label.x = 3) +
  labs(title= "Relacion peso y consumo de combustible",x = "Peso del coche", y = "Millas por galon") +
  theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8),
    title = element_text(size = 8))
dispercion
```

Relacion peso y consumo de combustible

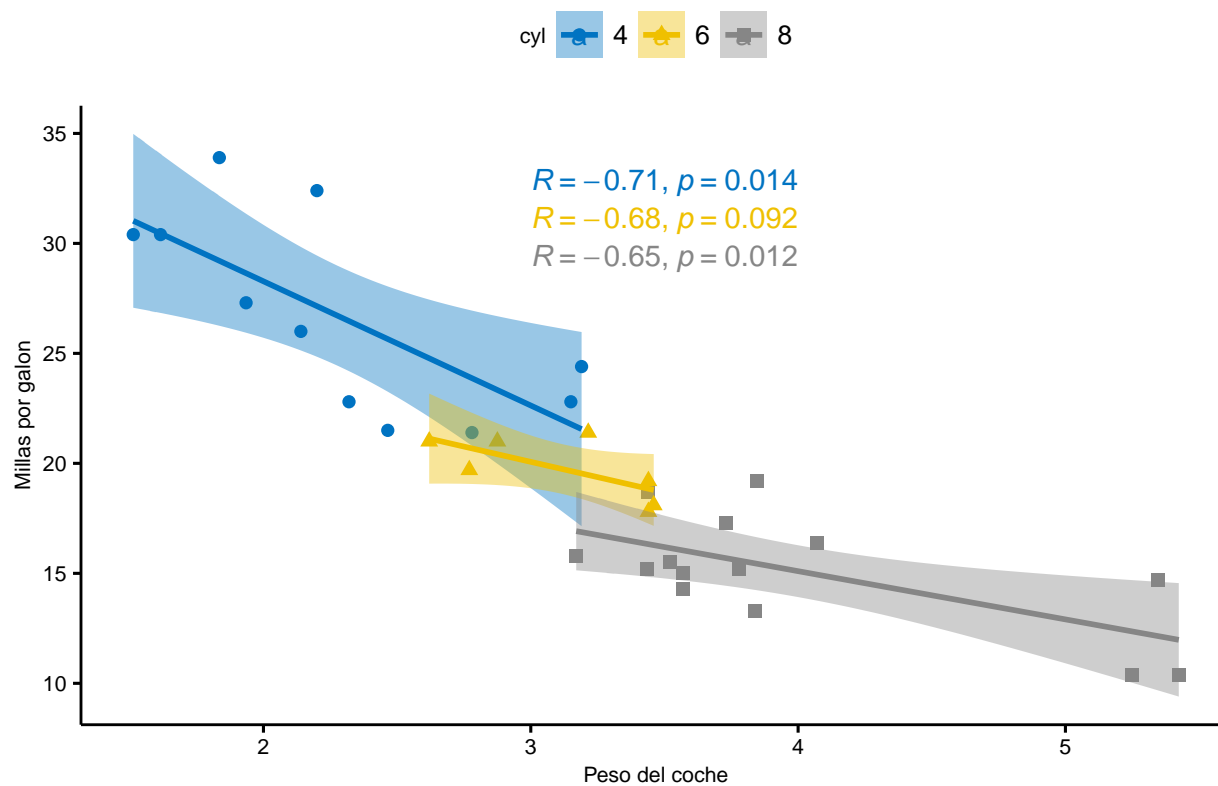
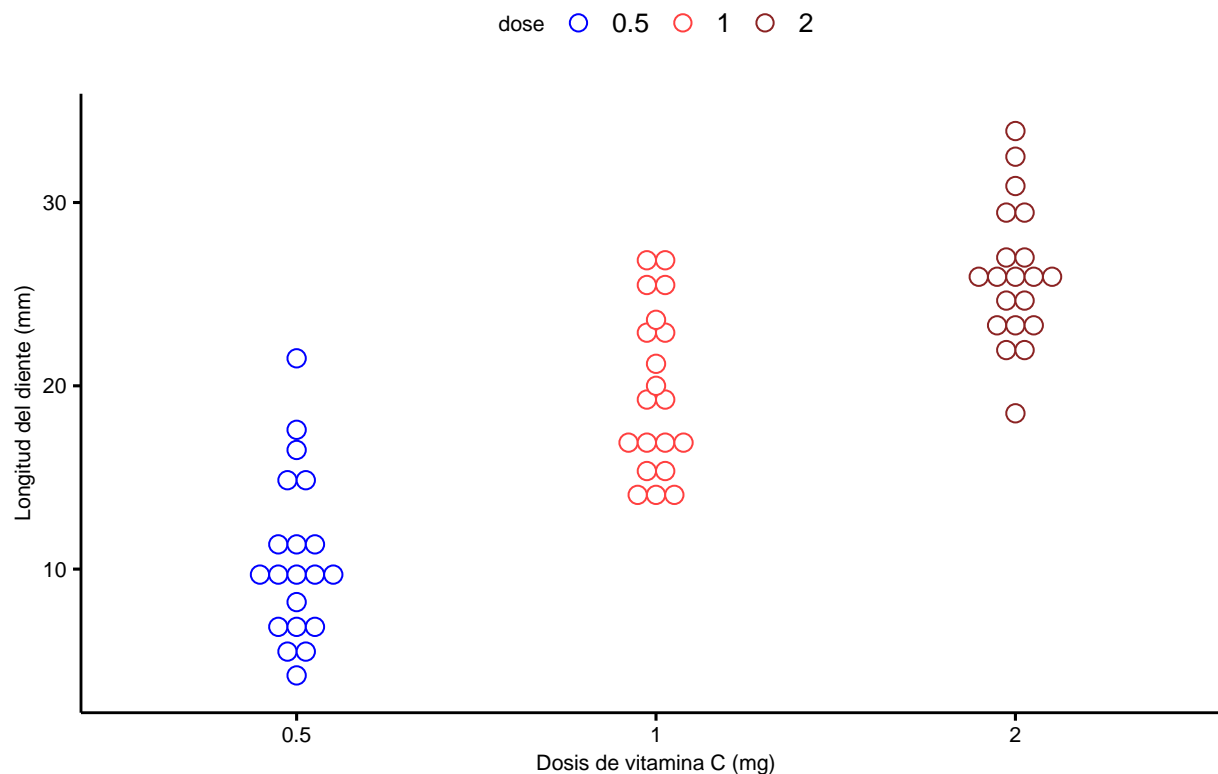


Gráfico de puntos

```
Puntos <- ggdotplot(ToothGrowth, x = "dose", y = "len",
  color = "dose",
  palette = "mpg",
  binwidth = 1) +
  labs(title = "Efecto de la vitamina C en el crecimiento del diente", x = "Dosis de vitamina C (mg)",
  scale_color_manual(values = c( "blue1", "brown1", "brown4")) +
  theme(axis.text = element_text(size = 8),
  axis.title = element_text(size = 8),
  title = element_text(size = 8))
```

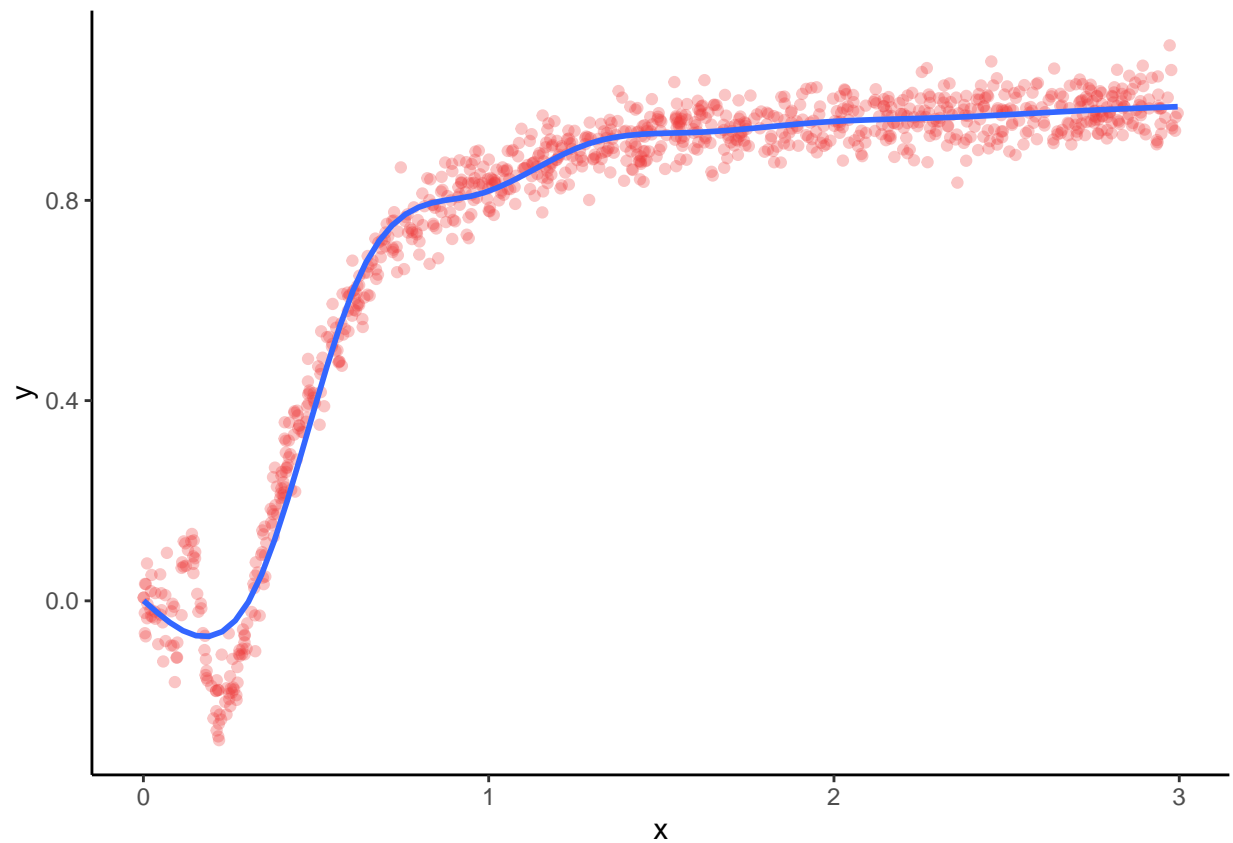
Puntos

Efecto de la vitamina C en el crecimiento del diente



```
## Gráfica zoom
set.seed(42) #semilla
n <- 1000    #tamaño
x <- runif(n) * 3 # Valores x, y
# Función no lineal de x, mas ruido normal
y <- x * sin(1/x) + rnorm(n) / 25
df <- data.frame(x = x, y = y)
zoom1 <- ggplot(df, aes(x, y)) + geom_point(alpha = 0.3, col = "brown2") +
  geom_smooth(se = FALSE) + theme_classic()
#se crea la segunda grafica con limites
zoom2 <- ggplot(df, aes(x, y)) + geom_point(alpha = 0.3, col = "lightblue2") + geom_smooth(se = FALSE) +
  scale_x_continuous(limits = c(0, 0.5)) +
  scale_y_continuous(limits = c(-0.5, 0.5)) +
  labs(x = NULL, y = NULL) + theme_classic()
zoom1
```

```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

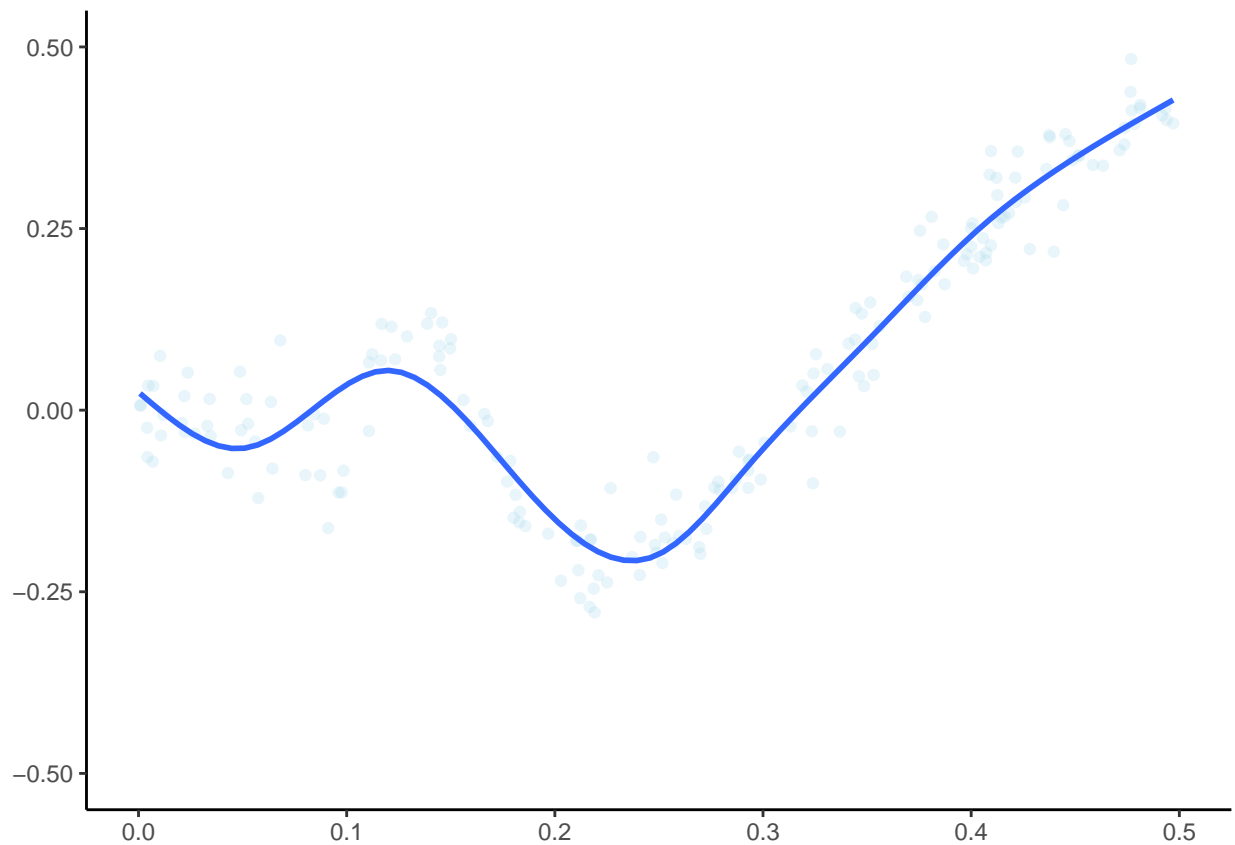



```
zoom2
```

```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 812 rows containing non-finite values ('stat_smooth()').
```

```
## Warning: Removed 812 rows containing missing values ('geom_point()').
```



```
# Se combinan las dos gráficas usando annotation_custom()
Z <- zoom1 + annotation_custom(ggplotGrob(zoom2), xmin = 1, xmax = 2, ymin = -0.5, ymax = 0.5) +
  labs(title = "Relación no lineal y oscilante entre x,y con zoom en la región baja x") + theme_dark()
  axis.title = element_text(size = 8),
  title = element_text(size = 5))
```

```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

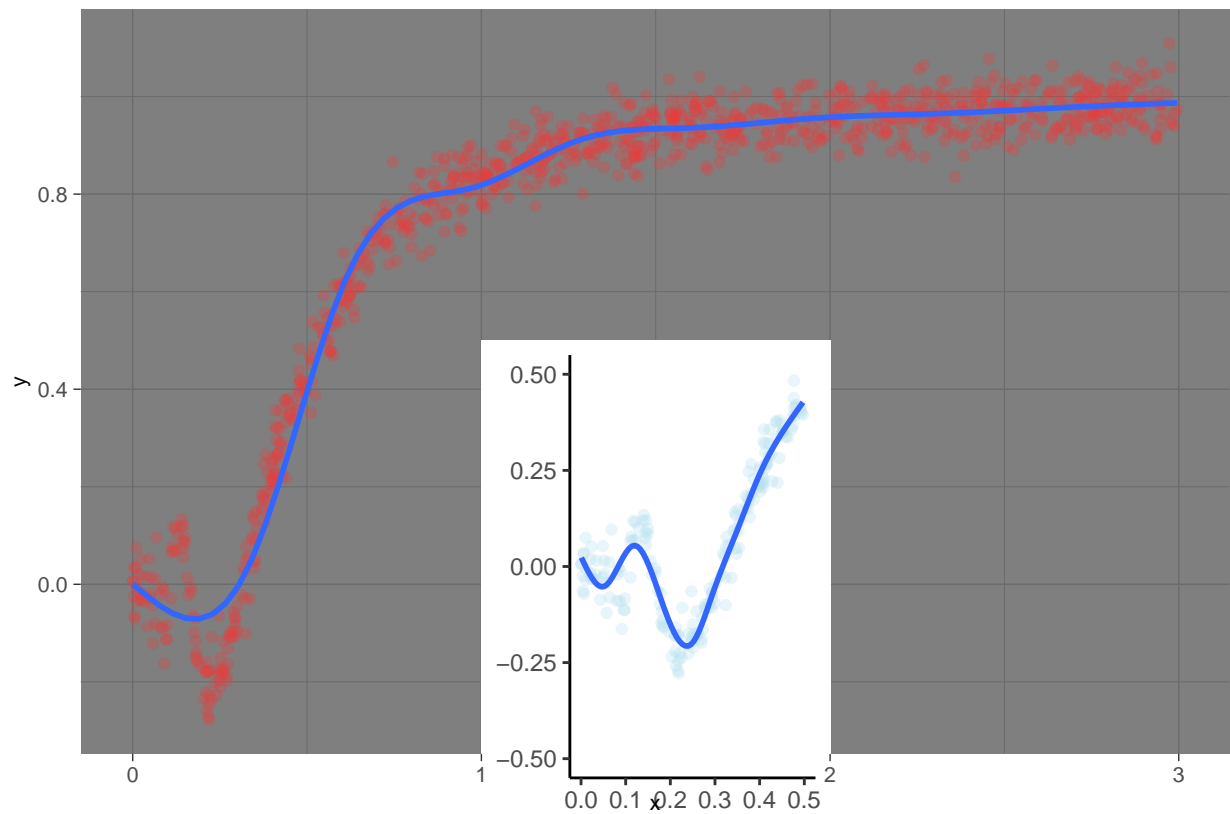
```
## Warning: Removed 812 rows containing non-finite values ('stat_smooth()').
```

```
## Warning: Removed 812 rows containing missing values ('geom_point()').
```

```
Z
```

```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

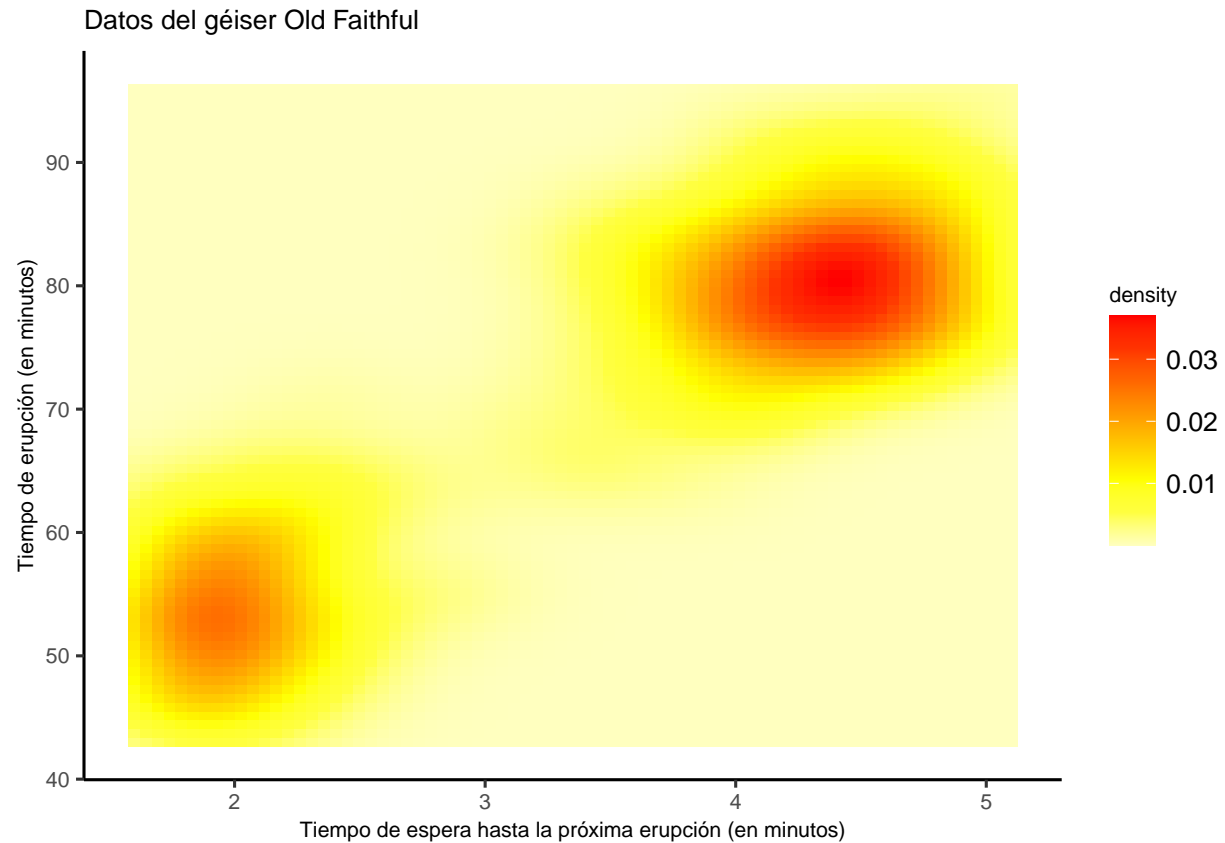
Relación no lineal y oscilante entre x,y con zoom en la región baja x



Grafica de calor

```
Calor <- ggplot(faithfuld, aes(eruptions, waiting)) +
  geom_raster(aes(fill = density))+
  theme_classic2()+
  scale_fill_gradientn(colours = heat.colors(8, rev = TRUE), na.value = "lightblue1") +
  labs(title = "Datos del géiser Old Faithful", x = "Tiempo de espera hasta la próxima erupción (en min",
        theme(axis.text = element_text(size = 8),
              axis.title = element_text(size = 8),
              title = element_text(size = 8))
```

Calor



```
##Graficas Unidas
```

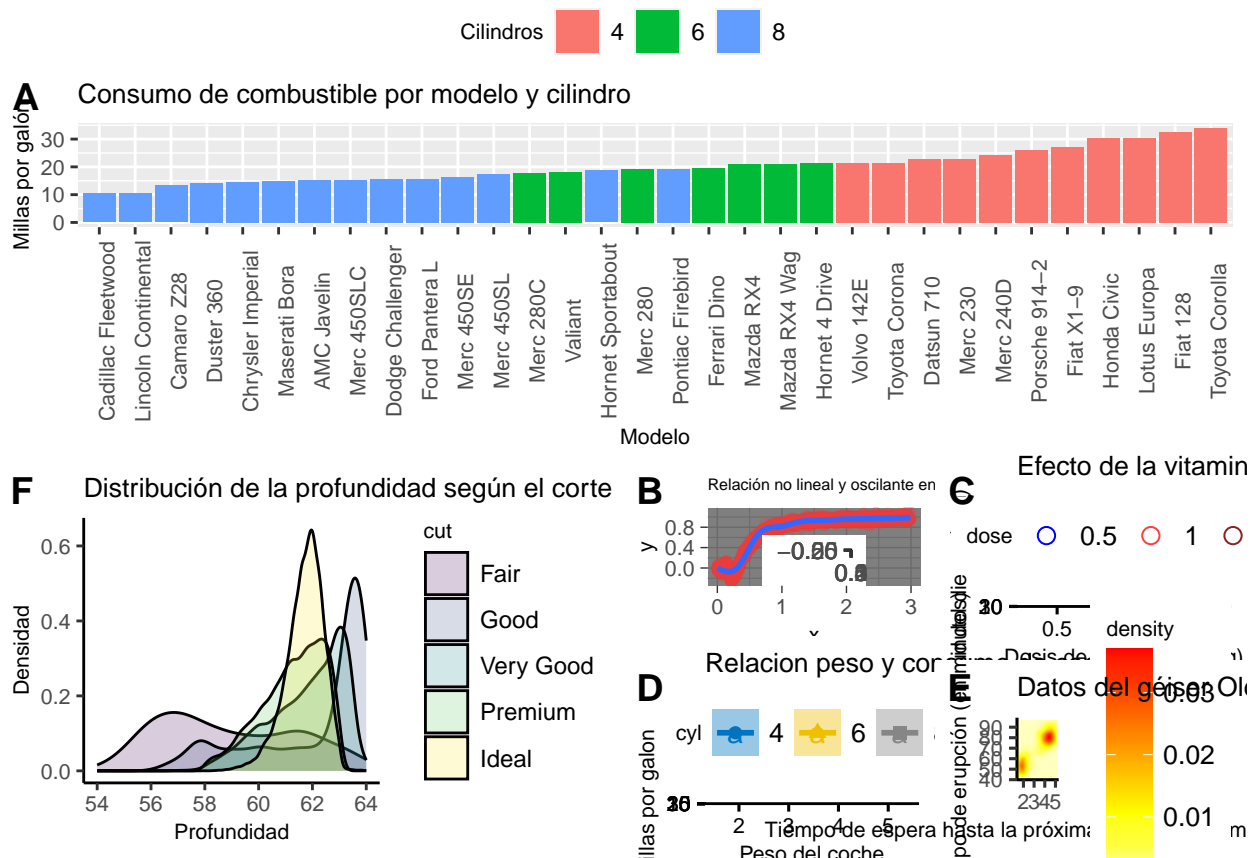
```
Completo =
```

```
  ggarrange(cars2, ggarrange(dens,ggarrange(Z, Puntos,dispercion, Calor, ncol = 2, nrow = 2,
      labels = c("B", "C", "D", "E")),ncol = 2,
      labels = "F"), labels = "A", nrow = 2, common.legend = TRUE) +
  theme(axis.text = element_text(size = 6),
        axis.title = element_text(size = 6),
        title = element_text(size = 6))
```

```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 1885 rows containing non-finite values ('stat_density()').
```

```
Completo
```



2. Réplica de la figura embebida

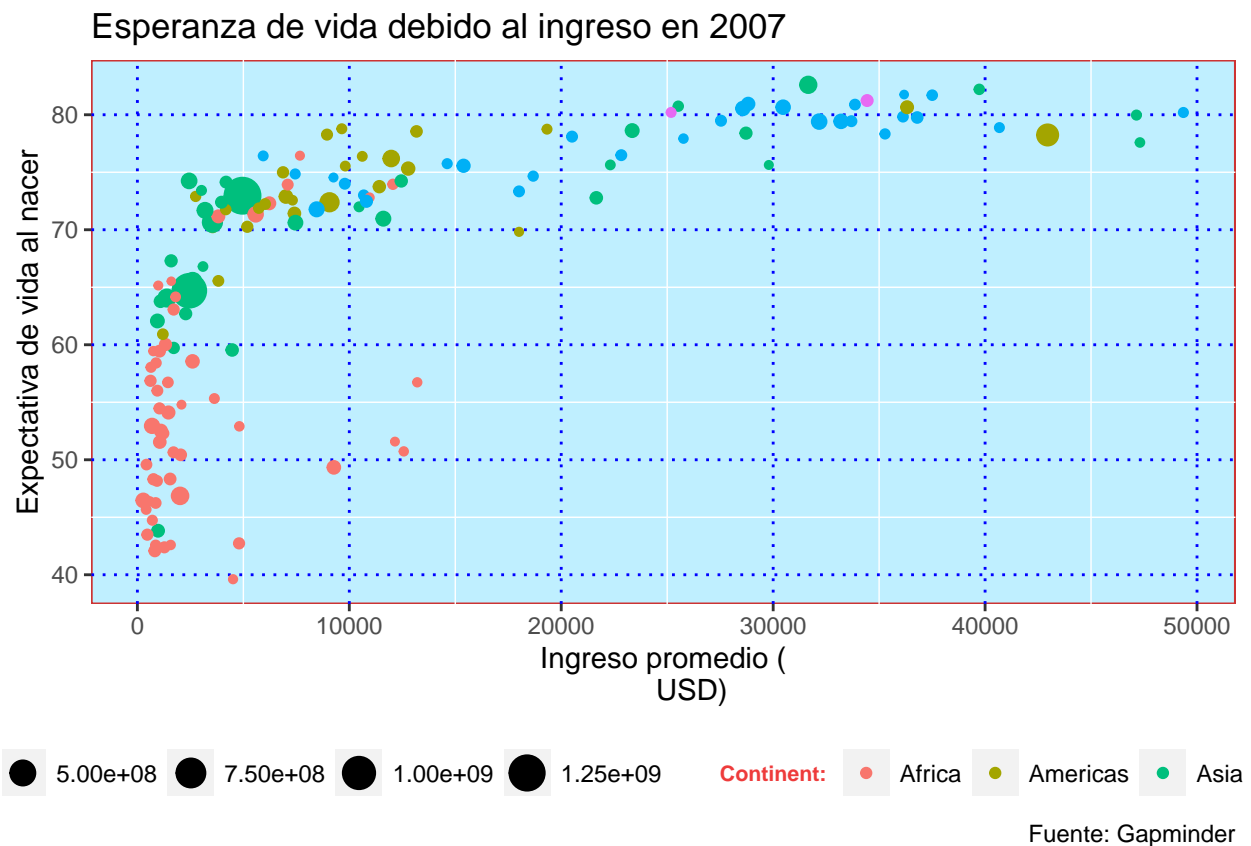
Datos “Gapminder”, que es un conjunto de datos ampliamente utilizado en la enseñanza y la investigación de la ciencia de datos, tiene información sobre la expectativa de vida, la población y el ingreso per cápita de diferentes países del mundo, entre 1952 y 2007. El conjunto de datos incluye información sobre 142 países diferentes. Gapminder. (2008). Gapminder World 2008 (v. 1.0) [Data set]. <https://www.gapminder.org/data/>

```
library(dplyr)
library(ggplot2)
library(grid)
library(gridExtra)
library(ggrepel)
```

```
# Para cargar la data, lo que deseamos visualizar
ruta <- "https://raw.githubusercontent.com/martintinch0/CienciaDeDatosParaCuriosos/master/data/gapminder"
df_gapminder <- read.table(file = ruta, sep=';', header = TRUE, stringsAsFactors = FALSE)
```

```
# Grafico con todos los paises
dfp1 <- df_gapminder %>%
  filter(year == 2007) %>%
  ggplot(aes(x = gdpPercap, y = lifeExp, size = pop, color = continent)) +
  geom_point() +
  labs(x = "Ingreso promedio (
```

```
USD)", y = "Expectativa de vida al nacer", title="Esperanza de vida debido al ingreso en 2007",
caption="Fuente: Gapminder",
color="Continent: ") +
theme(legend.position="bottom",
      legend.title = element_text(face = "bold", color = "brown2", size= 8),
      panel.background = element_rect(fill = "lightblue1", colour = "brown3"),
      panel.grid.major = element_line(colour = "blue", linetype = "dotted"))
dfp1
```



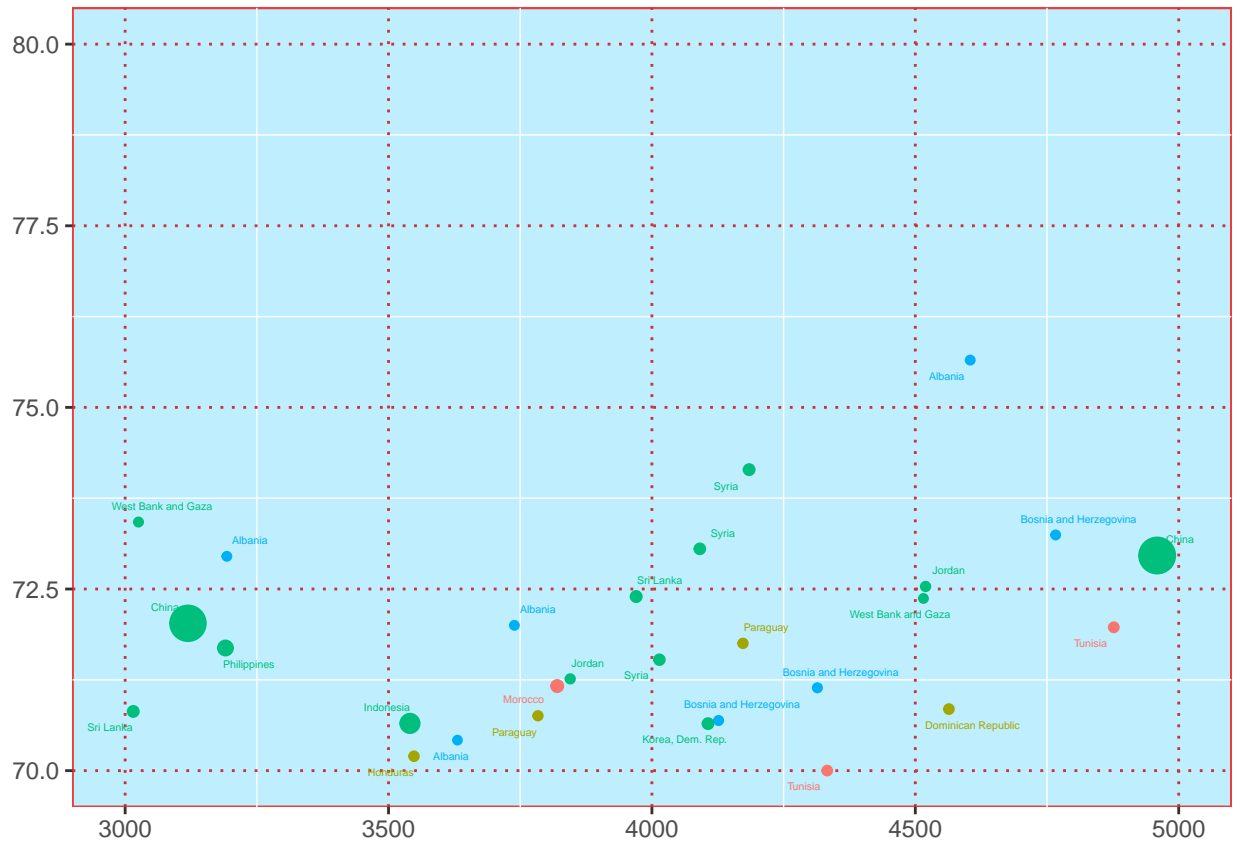
```
#grafico de algunos paises
dfp2 = ggplot(df_gapminder, aes(x = gdpPercap, y = lifeExp,
                                size=pop, color=continent)) +
  geom_point() +
  geom_text_repel(aes(label=country), size = 1.5) +
  scale_x_continuous(limits = c(3000, 5000)) +
  scale_y_continuous(limits = c(70, 80)) +
  guides(size=FALSE) +
  theme(legend.position = "null",
        axis.title = element_blank(),
        legend.title = element_text(face = "bold", color = "black", size= 8),
        panel.background = element_rect(fill = "lightblue1", colour = "brown2"),
        panel.grid.major = element_line(colour = "brown3", linetype = "dotted"))

## Warning: The '<scale>' argument of 'guides()' cannot be 'FALSE'. Use "none" instead as
## of ggplot2 3.3.4.
```

```
dfp2
```

```
## Warning: Removed 1676 rows containing missing values ('geom_point()').
```

```
## Warning: Removed 1676 rows containing missing values ('geom_text_repel()').
```



```
embeb = dfp1 + annotation_custom(ggplotGrob(dfp2),  
                                xmin = 30000, xmax = 50000,  
                                ymin = 40, ymax = 70)
```

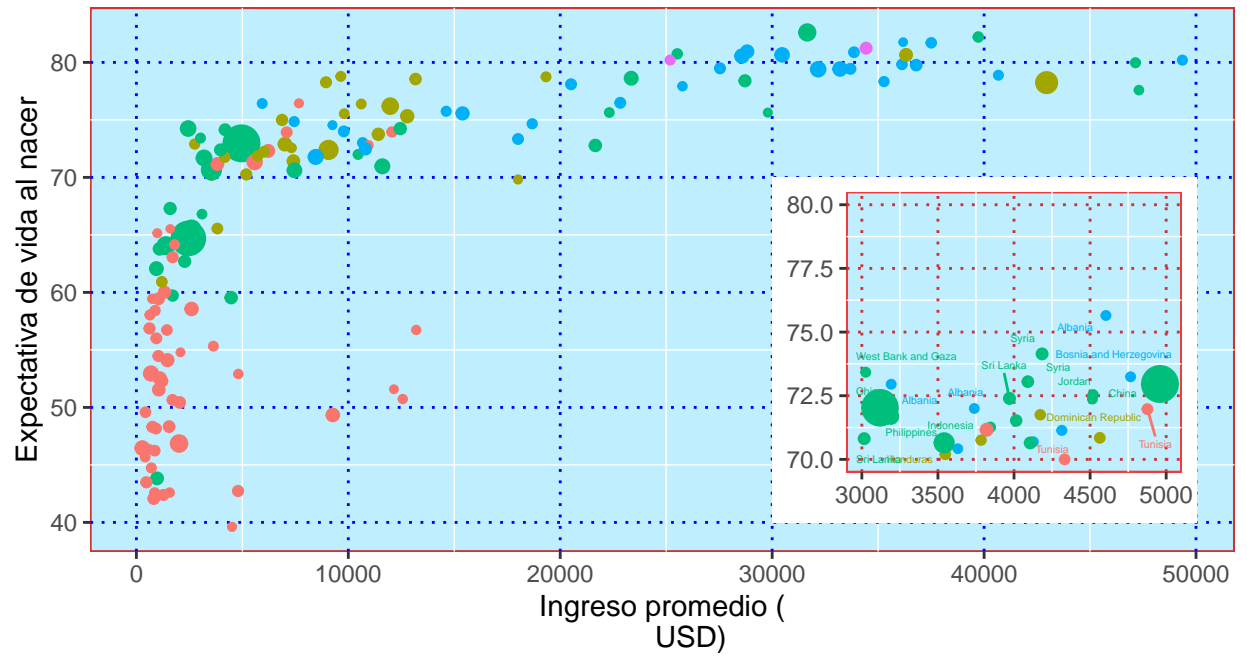
```
## Warning: Removed 1676 rows containing missing values ('geom_point()').
```

```
## Warning: Removed 1676 rows containing missing values ('geom_text_repel()').
```

```
embeb
```

```
## Warning: ggrepel: 10 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps
```

Esperanza de vida debido al ingreso en 2007



● 5.00e+08 ● 7.50e+08 ● 1.00e+09 ● 1.25e+09 **Continent:** ● Africa ● Americas ● Asia

Fuente: Gapminder