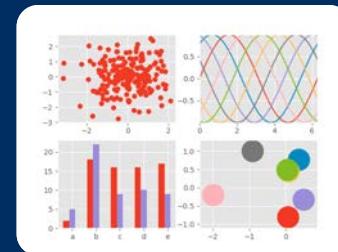
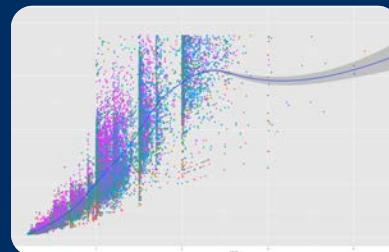


# Curso-taller: Introducción a R – Día 4

## Visualización de datos utilizando ggplot2



Presentan:

Dr. Ulises Olivares Pinto

Dr. Alberto Prado Farías

Escuela Nacional de Estudios Superiores Unidad Juriquilla

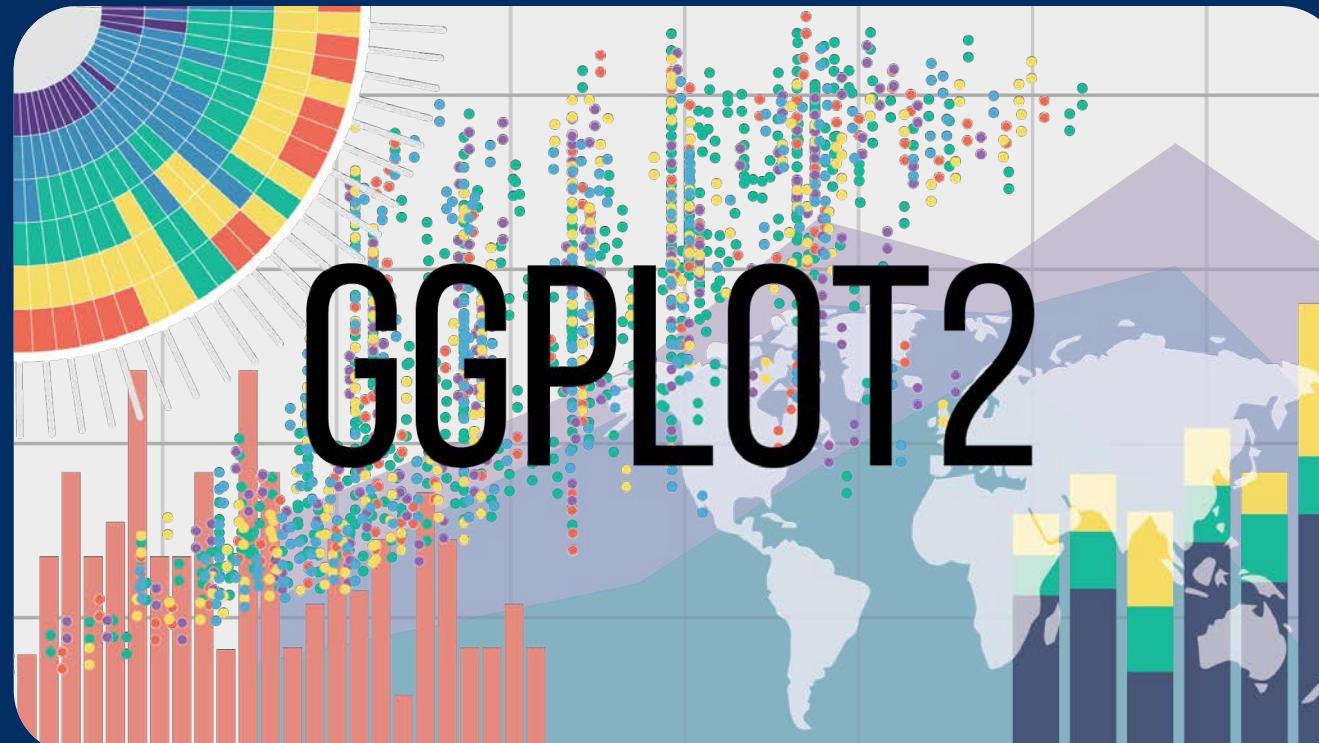




# Introducción a R – Visualización de datos – ggplot

1. Introducción
2. Instalación de librería ggplot2
3. Ejemplos de uso de ggplot2
  - ✓ Gráficos de dispersión
    - Estéticas
    - Facetas
      - Grids
  - ✓ Gráficos de línea
    - Regresiones
  - ✓ Boxplots
  - ✓ Transformaciones estadísticas
    - Histogramas
  - ✓ Más geometrías
4. Exportar gráficos con calidad de publicación

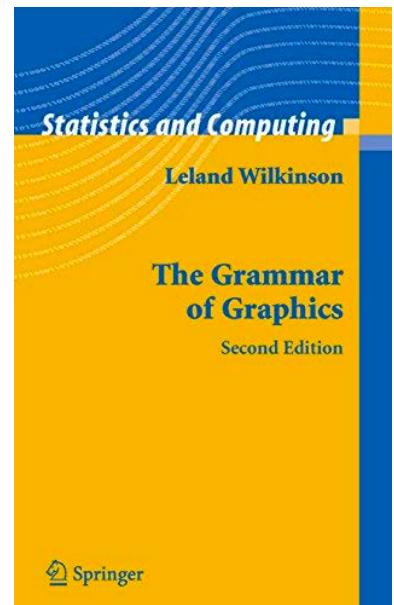
# 1. Introducción





# 1. Introducción – ggplot2

- **Ggplot2** es una librería para la obtención de gráficos de alta calidad, esta basada en el libro “**The grammar of graphics**”.
- **Gramática** tiene una acepción técnica en lingüística. Chomsky define a una gramática como un **conjunto de reglas para construir enunciados válidos en un lenguaje** (Chomsky, 1956).
- Una gramática hace que un lenguaje sea **expresivo**:  
Ejemplo:
  - Lenguaje sin gramática => (*palabra = enunciado*) => **ambiguo e inexpresivo**.
  - Lenguaje con gramática => (*palabras se combinan = enunciado*) => **expande el lenguaje (contexto)**



# 1. Introducción – ggplot2 – Geometrías

## GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemploy))  
b <- ggplot(seals, aes(x = long, y = lat))
```

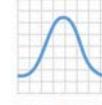
-  **a + geom\_blank()**  
(Useful for expanding limits)
-  **b + geom\_curve(aes(yend = lat + 1,  
xend=long+1),curvature=1) - x, xend, y, yend,  
alpha, angle, color, curvature, linetype, size**
-  **a + geom\_path(lineend="butt", linejoin="round",  
linemitre=1)  
x, y, alpha, color, group, linetype, size**
-  **a + geom\_polygon(aes(group = group))  
x, y, alpha, color, fill, group, linetype, size**
-  **b + geom\_rect(aes(xmin = long, ymin=lat, xmax=  
long + 1, ymax = lat + 1)) - xmax, xmin, ymax,  
ymin, alpha, color, fill, linetype, size**
-  **a + geom\_ribbon(aes(ymin=unemploy - 900,  
ymax=unemploy + 900)) - x, ymax, ymin,  
alpha, color, fill, group, linetype, size**

## ONE VARIABLE continuous

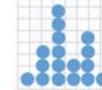
```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)
```



**c + geom\_area(stat = "bin")**  
x, y, alpha, color, fill, linetype, size



**c + geom\_density(kernel = "gaussian")**  
x, y, alpha, color, fill, group, linetype, size, weight



**c + geom\_dotplot()**  
x, y, alpha, color, fill



**c + geom\_freqpoly()** x, y, alpha, color, group,  
linetype, size



**c + geom\_histogram(binwidth = 5)** x, y, alpha,  
color, fill, linetype, size, weight



**c2 + geom\_qq(aes(sample = hwy))** x, y, alpha,  
color, fill, linetype, size, weight

.....

## TWO VARIABLES

### continuous x , continuous y

```
e <- ggplot(mpg, aes(cty, hwy))
```



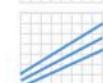
**e + geom\_label(aes(label = cty), nudge\_x = 1,  
nudge\_y = 1, check\_overlap = TRUE) x, y, label,  
alpha, angle, color, family, fontface, hjust,  
lineheight, size, vjust**



**e + geom\_jitter(height = 2, width = 2)**  
x, y, alpha, color, fill, shape, size



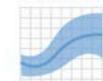
**e + geom\_point(), x, y, alpha, color, fill, shape,  
size, stroke**



**e + geom\_quantile(), x, y, alpha, color, group,  
linetype, size, weight**



**e + geom\_rug(sides = "bl"), x, y, alpha, color,  
linetype, size**



**e + geom\_smooth(method = lm), x, y, alpha,  
color, fill, group, linetype, size, weight**

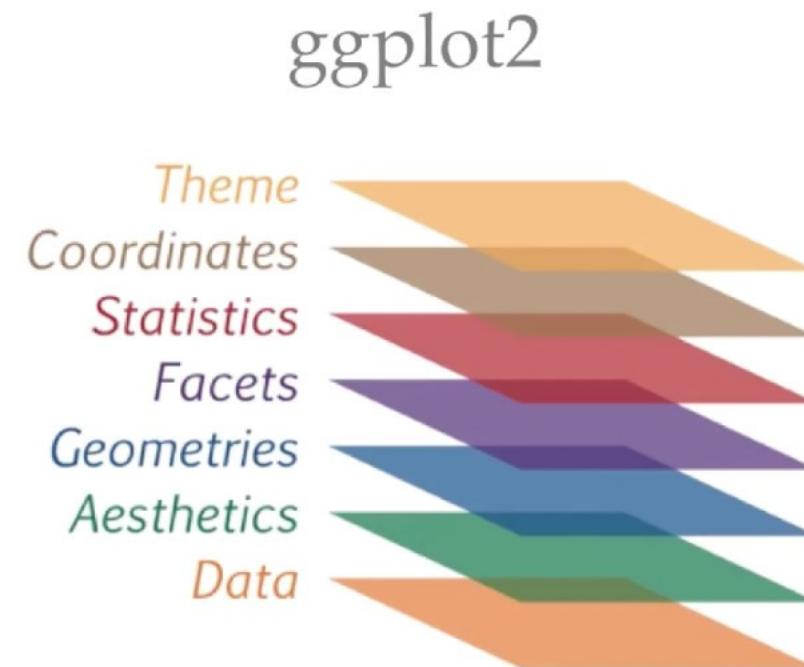


**e + geom\_text(aes(label = cty), nudge\_x = 1,  
nudge\_y = 1, check\_overlap = TRUE), x, y, label,  
alpha, angle, color, family, fontface, hjust,  
lineheight, size, vjust**

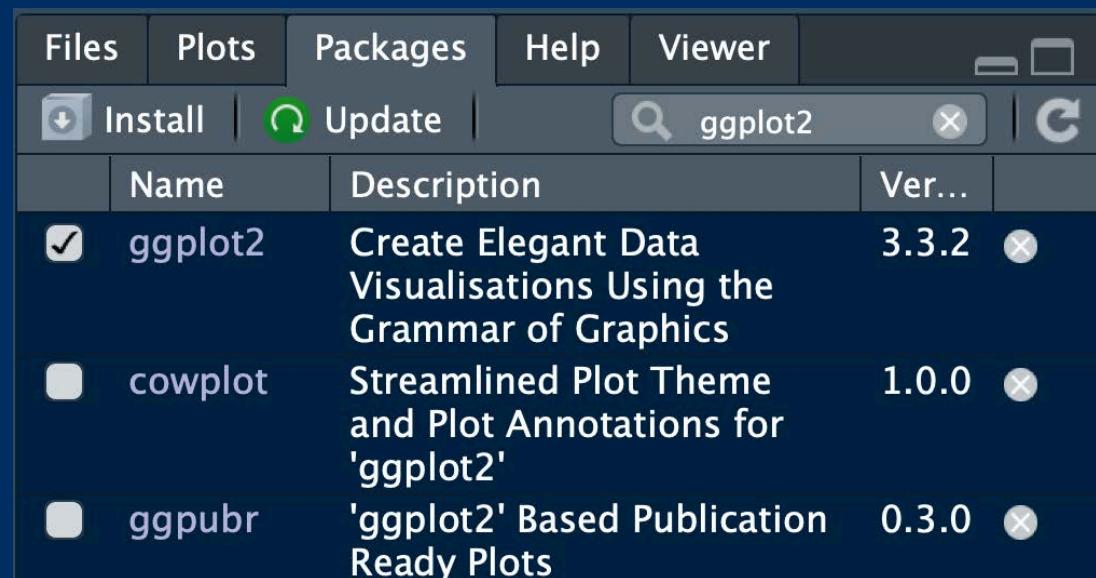
Fuente: <https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf>



# 1. Introducción – ggplot2



## 2. Instalación de Ggplot2



A screenshot of the RStudio interface showing the 'Packages' tab selected in the top menu bar. The search bar contains 'ggplot2'. A table lists three packages: ggplot2, cowplot, and ggpubr. The 'ggplot2' row has a checked checkbox in the first column, indicating it is selected for installation.

	Name	Description	Ver...	
<input checked="" type="checkbox"/>	ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics	3.3.2	
<input type="checkbox"/>	cowplot	Streamlined Plot Theme and Plot Annotations for 'ggplot2'	1.0.0	
<input type="checkbox"/>	ggpubr	'ggplot2' Based Publication Ready Plots	0.3.0	

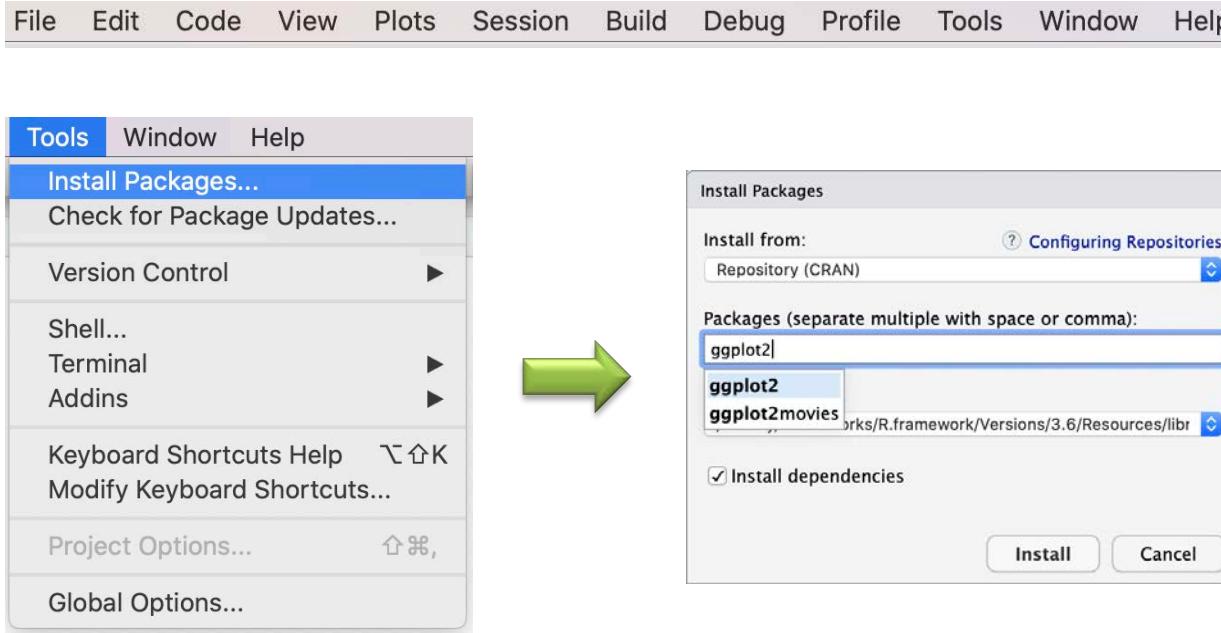
```
> install.packages("ggplot2")
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/ggplot2_3.3.2.tgz'
Content type 'application/x-gzip' length 4068619 bytes (3.9 MB)
=====
downloaded 3.9 MB
```



## 2. Instalación de ggplot2

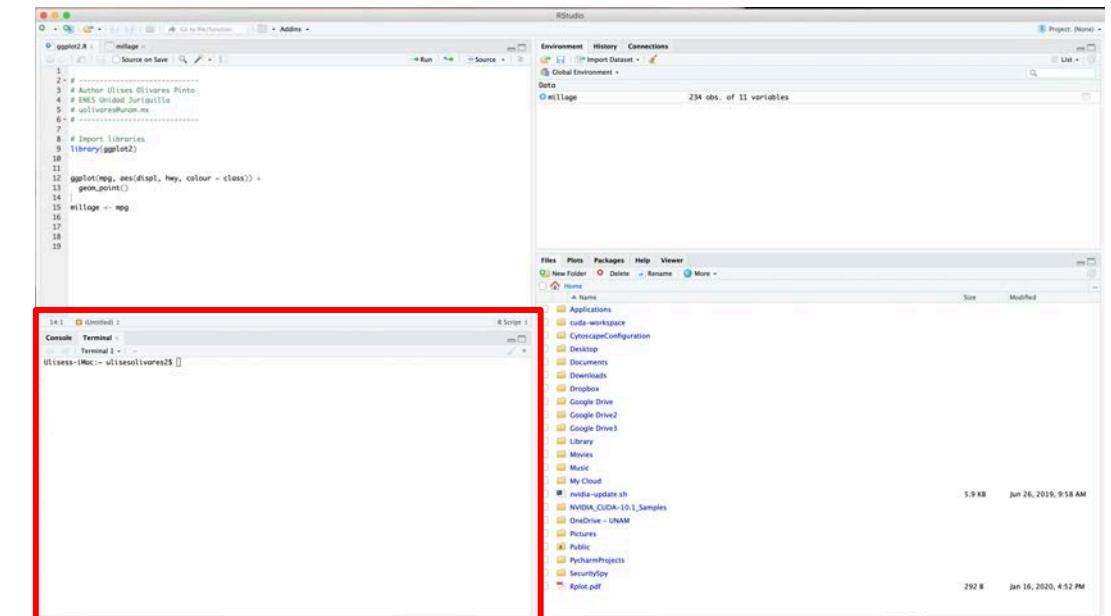
### Opción 1: R-studio

Rstudio cuenta con un gestor que permite administrar **repositorios** e instalar **librerías** de forma gráfica.



### Opción 2: Línea de comandos (terminal)

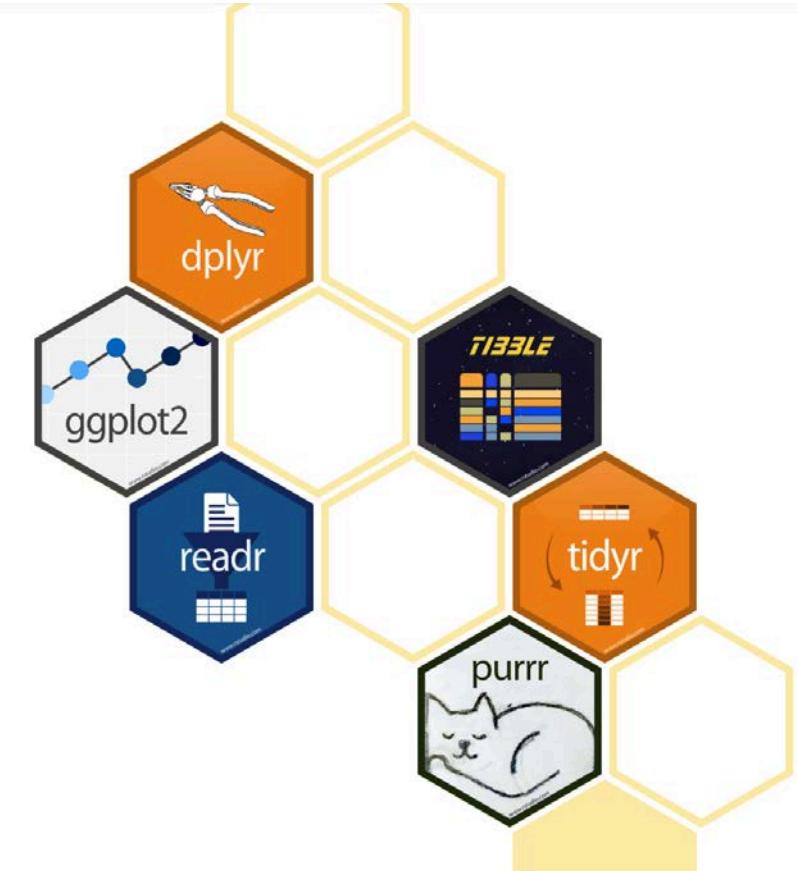
Rstudio cuenta con un **gestor** que permite administrar librerías.



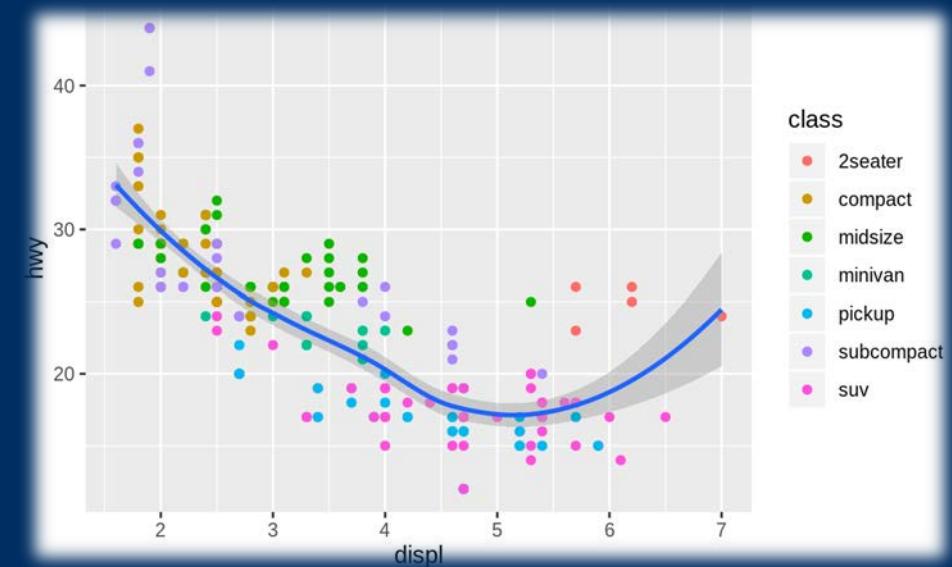
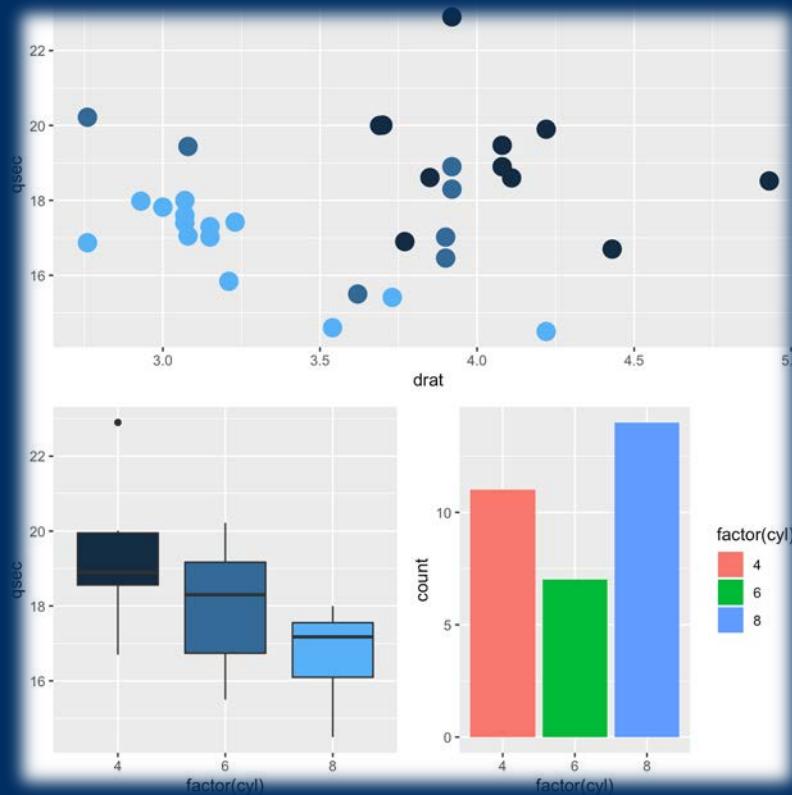
[install.packages\("ggplot2"\)](#)

## 2. Instalación de ggplot2 – Tidyverse (Opcional)

- Opcionalmente puede instalarse **Tidyverse** es una colección de paquetes de R que incluye ggplot2.
- Se utiliza ampliamente para realizar análisis en el área de ciencia de datos



# 3. Uso de Ggplot2

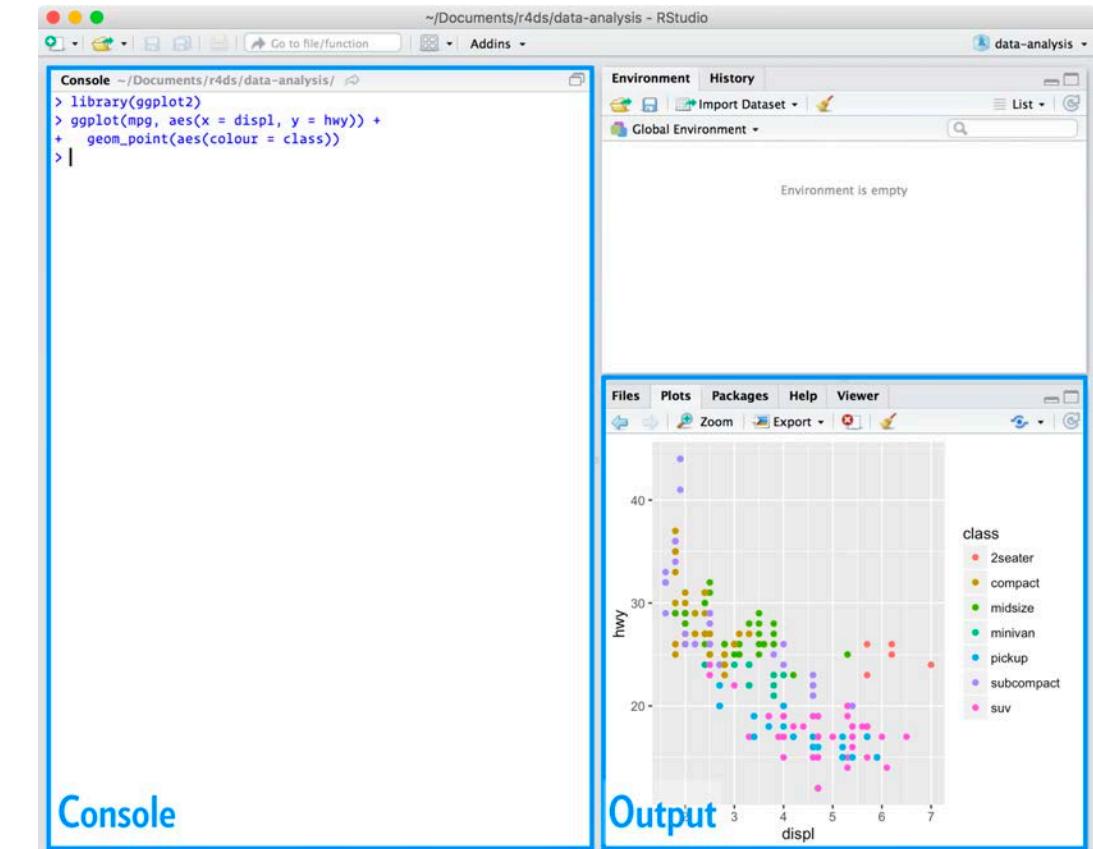




### 3. Uso de ggplot2

- Las funciones de la librería ggplot2 se pueden instanciar, una vez que se importa la librería con la siguiente instrucción.

```
# Import libraries  
library(ggplot2)
```



## 3.1 Uso de ggplot2 – data frame – Consumo de combustible

El data frame cuenta con 234 filas, 38 modelos de automóviles (1999 - 2008)  
 11 columnas (variables).

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact
11	audi	a4 quattro	2.0	2008	4	auto(s6)	4	19	27	p	compact
12	audi	a4 quattro	2.8	1999	6	auto(l5)	4	15	25	p	compact
13	audi	a4 quattro	2.8	1999	6	manual(m5)	4	17	25	p	compact
14	audi	a4 quattro	3.1	2008	6	auto(s6)	4	17	25	p	compact
15	audi	a4 quattro	3.1	2008	6	manual(m6)	4	15	25	p	compact





## 3.1 Uso de ggplot2 – data frame – ayuda

El data frame cuenta con diversas variables, dos de las más representativas son:

- ✓ **Displ (Desplazamiento):** Tamaño de motor de automóvil
- ✓ **Hwy:** Eficiencia del automóvil en autopista medida en millas/galón

- Es posible obtener más detalles sobre el data frame utilizando el siguiente comando

```
# Detalles sobre el dataframe  
?mpg  
  
# Estadísticos sencillos del data frame  
summary(mpg)
```

mpg {ggplot2}

R Documentation

Fuel economy data from 1999 and 2008 for 38 popular models of car

### Description

This dataset contains a subset of the fuel economy data that the EPA makes available on <http://fueleconomy.gov>. It contains only models which had a new release every year between 1999 and 2008 - this was used as a proxy for the popularity of the car.

### Usage

mpg

### Format

A data frame with 234 rows and 11 variables

manufacturer  
model

model name

displ

engine displacement, in litres

year

year of manufacture

cyl

number of cylinders

trans

type of transmission

drv

f = front-wheel drive, r = rear wheel drive, 4 = 4wd

cty

city miles per gallon

hwy

highway miles per gallon

fl

fuel type

class

"type" of car

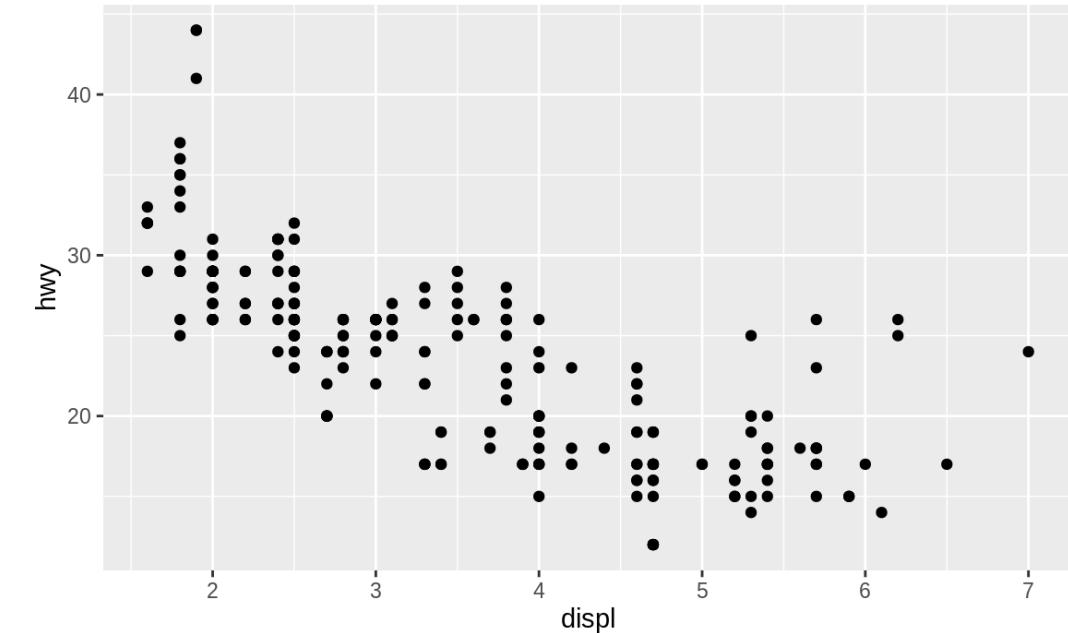
## 3.2 Gráficos de dispersión

- Plantilla básica
- ✓ Notación: <variables que el usuario debe especificar>

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

- Actividades básicas:
  1. Correr el siguiente comando:  
`ggplot (data = consumo)`
  2. Hacer una gráfica de dispersión sencilla de displ vs hwy.

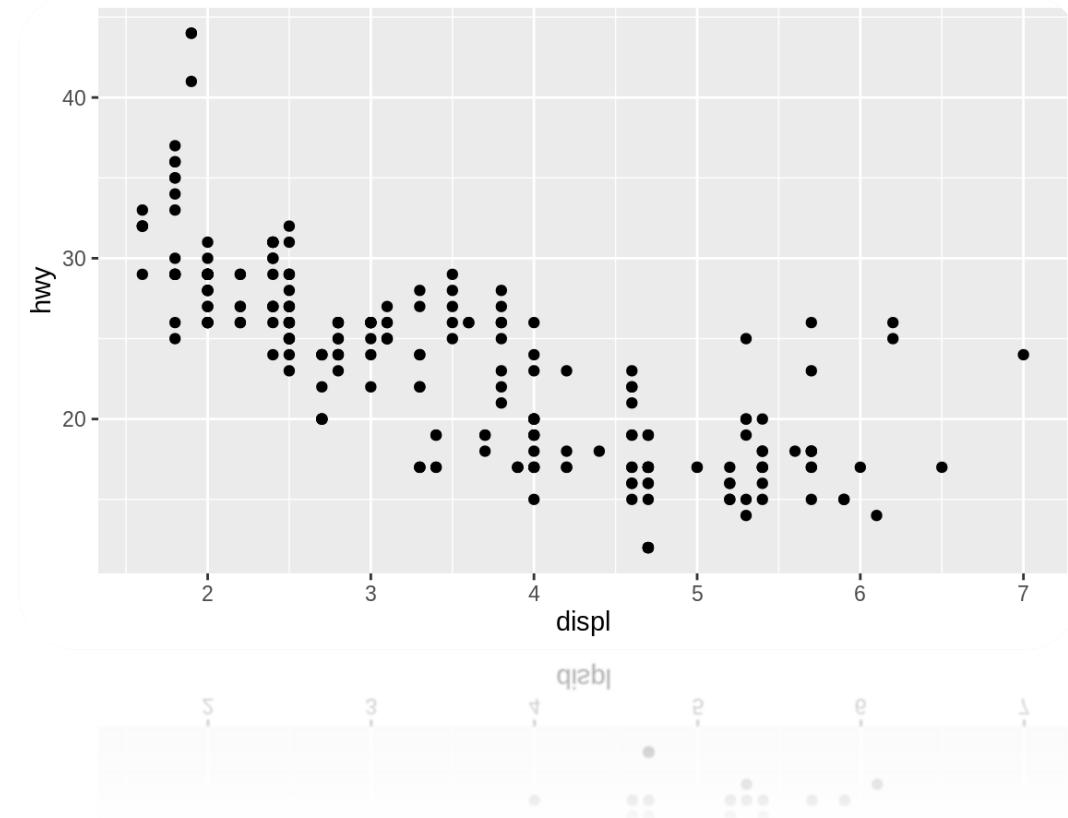
```
ggplot(data = consumo, aes(x = desp, y = autop )) +  
  geom_point()
```





## 3.2 Gráficos de dispersión

1. ¿Qué significado tiene la gráfica disp vs hwy?
  
2. ¿Qué relación existe entre el tamaño de motor y rendimiento?
  
3. ¿Es esta gráfica realmente expresiva, se podría brindar información adicional?





## 3.2 Gráficos de dispersión – Ejercicios 1

### Ejercicios (15 Minutos)

#### 1. Traducir títulos del data-frame

(fabricante, modelo, desplazamiento, año, cilindros, transmisión, tracción, ciudad, autopista, combustible, clase)

manufacturer	▼	model	▼	displ	▼	year	▼	cyl	▼	trans	▼	drv	▼	cty	▼	hwy	▼	fl	▼	class	▼
--------------	---	-------	---	-------	---	------	---	-----	---	-------	---	-----	---	-----	---	-----	---	----	---	-------	---

#### 2. Crear una gráfica de dispersión de **despl vs ciudad**

- ✓ ¿Se mantiene la misma relación en ciudad que en autopista?

#### 3. Crear una nueva gráfica de dispersión de **cil vs autop**

- ✓ ¿Qué significado tiene esta gráfica?

#### 4. Crear una nueva gráfica de dispersión de **fabricante vs autop**

- ✓ ¿Qué significado tiene esta gráfica?

#### 5. Crear una gráfica **clase vs trans**

- ✓ ¿Es esta gráfica útil?

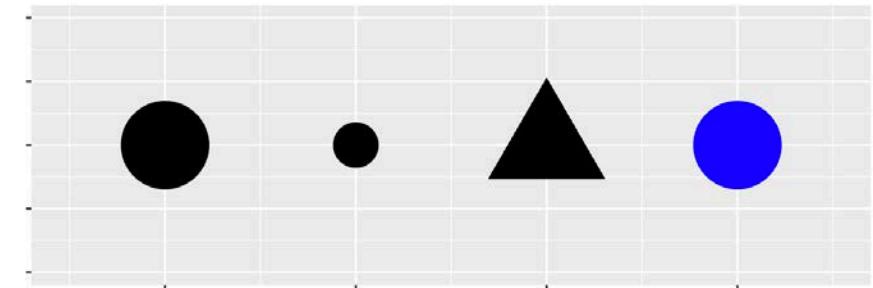
## 3.2 Uso de ggplot – Gráficos de dispersión – Estéticas

Es posible agregar una **propiedad** para clasificar los valores de una gráfica y hacerla más descriptiva.

A esta propiedad se le conoce como **estética “aesthetic”**. Este concepto es una propiedad visual que se le asigna a los objetos en un gráfico.

La estética incluye aspectos tales como:

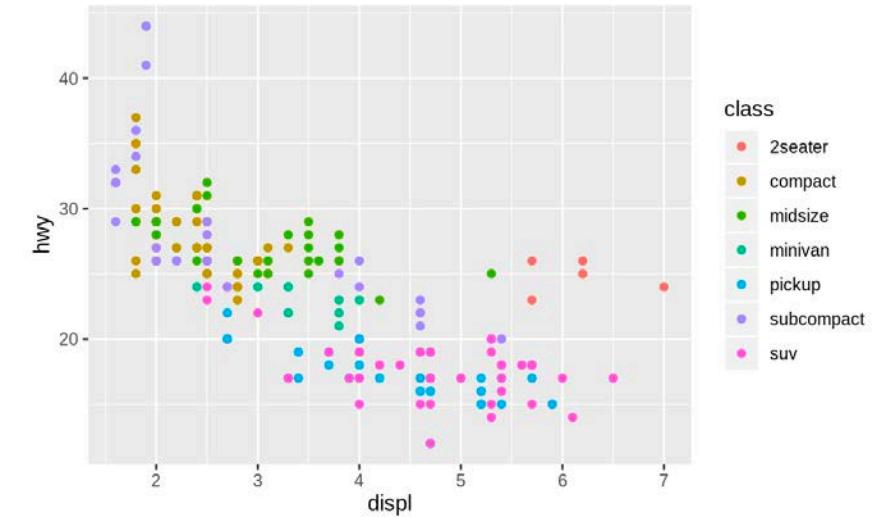
- ✓ Tamaño
- ✓ Color
- ✓ Forma



## 3.2 Uso de ggplot – Gráficos de dispersión – Estéticas

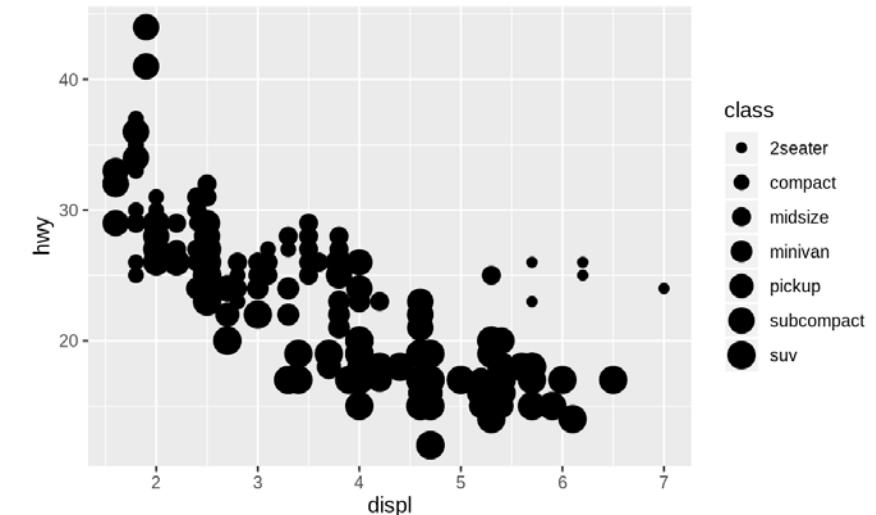
Para el ejemplo anterior, es posible clasificar los puntos del diagrama de dispersión por **colores** de acuerdo a la clase de automóvil:

```
ggplot(data = consumo, aes(x = desp, y = autop, color = clase)) +  
  geom_point()
```



De la misma forma, se puede realizar la clasificación por **tamaño**:

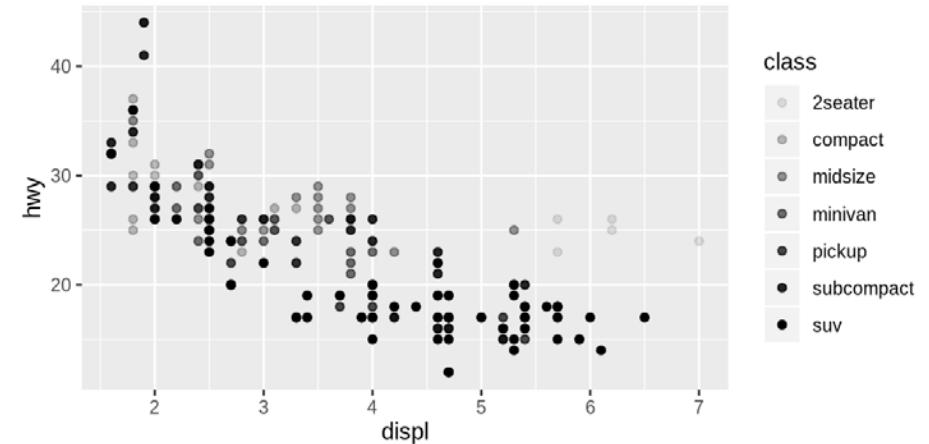
```
ggplot(data = consumo, aes(x = desp, y = autop, size = clase)) +  
  geom_point()
```



## 3.2 Uso de ggplot – Gráficos de dispersión – Estéticas

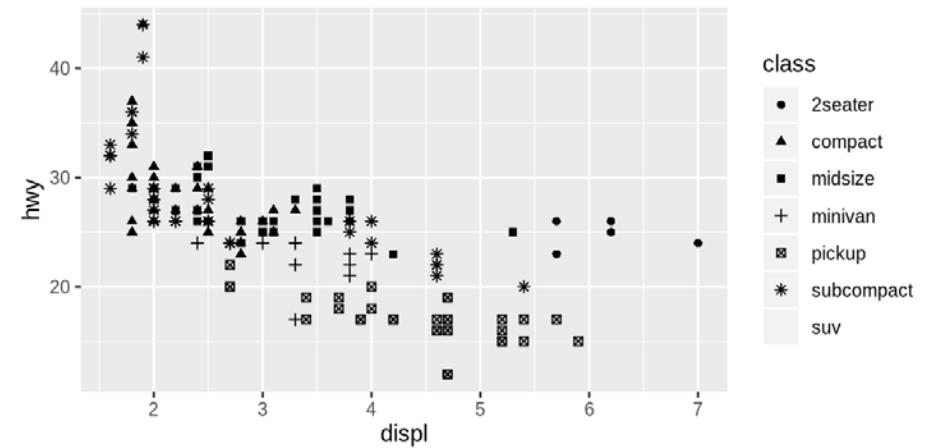
Adicionalmente, se puede realizar la clasificación utilizando **transparencia**.

```
ggplot(data = consumo, aes(x = desp, y = autop, alpha = clase)) +  
  geom_point()
```



O clasificar por **formas**

```
ggplot(data = consumo, aes(x = desp, y = autop, shape = clase)) +  
  geom_point()
```

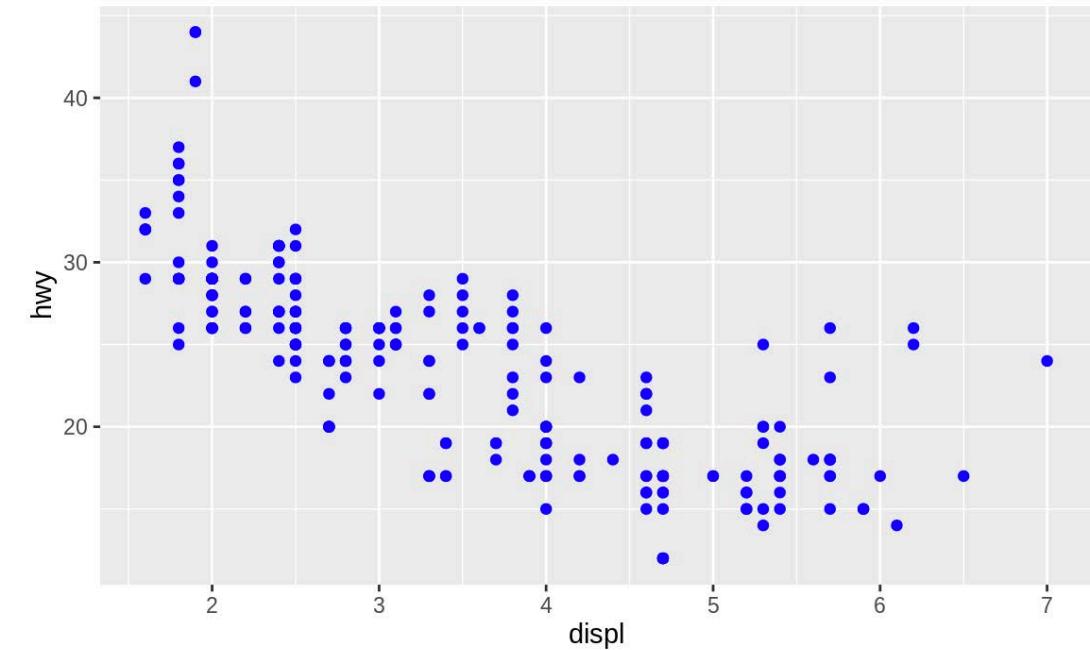


## 3.2 Uso de ggplot – Gráficos de dispersión – Estéticas

Es posible definir **manualmente** la estética de una gráfica.

En esta caso, el color no brindará información adicional de una variable. No obstante, si modificará la apariencia de esta.

```
ggplot(data = consumo, aes(x = desp, y = autop)) +  
  geom_point(color = "blue")
```



## 3.2 Gráficos de dispersión – Ejercicios 2

1. ¿Qué variables de `mpg` son continuas y cuales son categóricas?

✓ Hint: Utilizar la ayuda de `mpg`

- `?mpg`
- `str(mpg)`

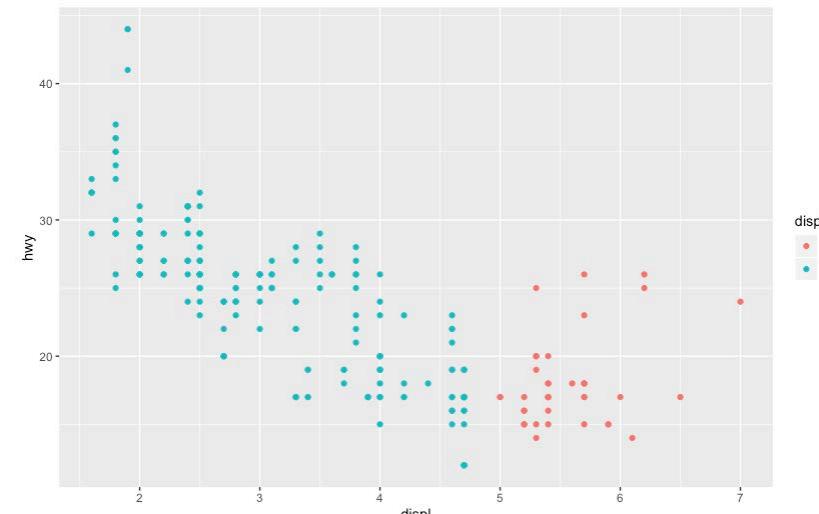
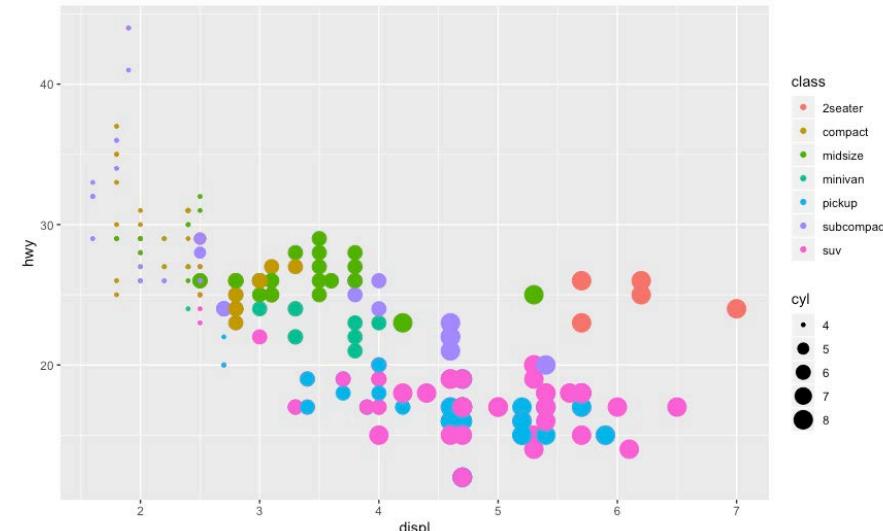
2. Identificar un par de variables continuas del data frame `mpg`. Posteriormente, se deberán generar tres gráficos usando las estéticas siguientes:

- ✓ Color
- ✓ Tamaño
- ✓ Forma

3. ¿Qué pasa si se mapean para la misma variable distintas estéticas?

4. ¿Qué pasa se se mapea una estética con una condición?

✓ Ej: `color = displ < 5`





## 3.2 Gráficos de dispersión – Ejercicio 3

### Ejercicio (10 minutos)

Generar dos nuevas columnas para el dataframe `consumo` con nombres `ciudadkm` y `autopkm` en estas nuevas columnas se deberá hacer la conversión de millas/galón a km/lt utilizando la siguiente fórmula:

$$\text{rendimiento} = \frac{\text{rendimiento (millas, galón)}}{2.352}$$

Longitud / Volumen

1 = 0.425144

Milla / Galón estadounidense Kilometro / Litro

Fórmula para obtener un resultado aproximado, divide el valor de longitud / volumen entre 2.352

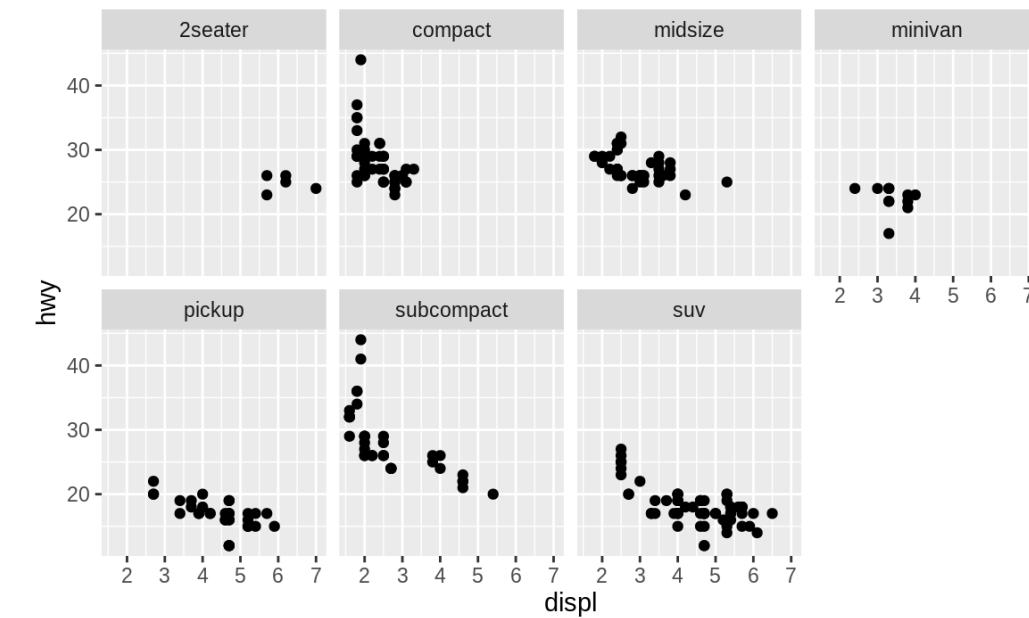
Graficar ambas columnas con respecto al desplazamiento del motor. Se deberá clasificar por clase de automóvil

- ✓ ¿Se mantiene la misma distribución de datos?

## 3.2.1 Gráficos de dispersión – Facetas

- Un método para agregar variables adicionales a un gráfico consiste en el uso de **estéticas**.
- Otra forma, que es muy útil para variables categóricas es el uso de **facetas**. Gráficos que de forma independiente muestran un subconjunto de datos

```
ggplot(data = consumo) +  
  geom_point(aes(x= desp, y= autop))+  
  facet_wrap(~ clase, nrow = 2)
```

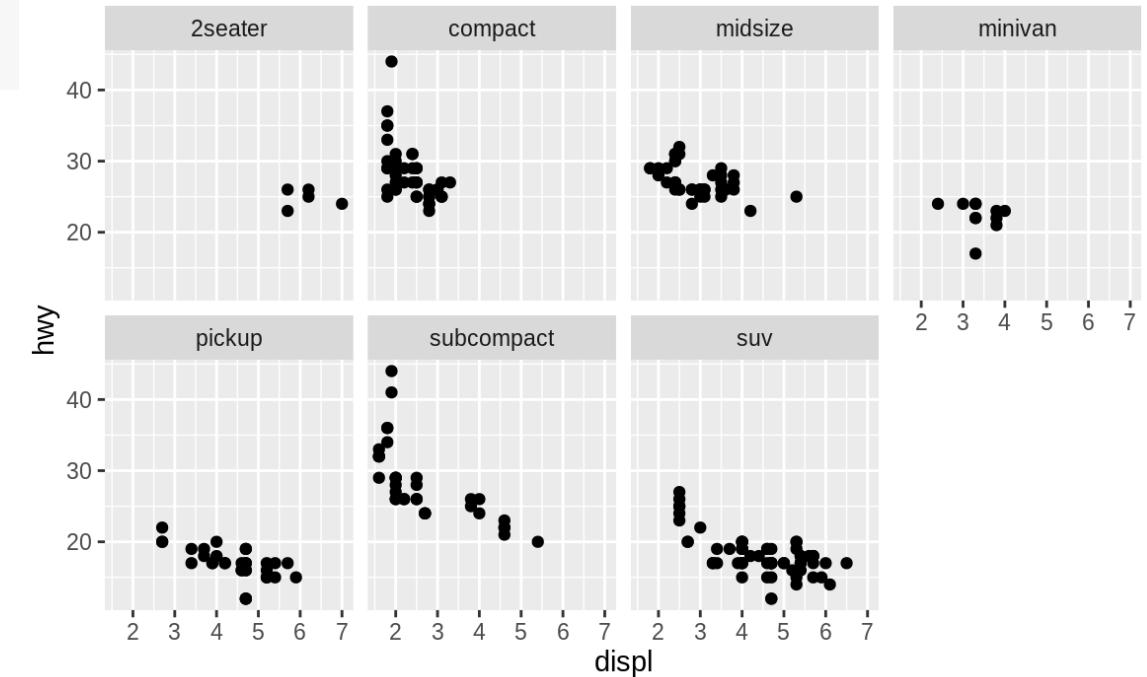


- ¿Qué ventajas tiene esta división de datos?

## 3.2.2 Gráficos de dispersión – Facetas

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_wrap(~ class, nrow = 2)
```

- Utilizar la ayuda para saber como funciona el parámetro `nrow` de `?facet_wrap`
1. ¿Cuales son la ventajas de utilizar facetas en lugar de agrupar los datos por formas o colores?
  2. ¿Cuáles son las desventajas de utilizar facetas?



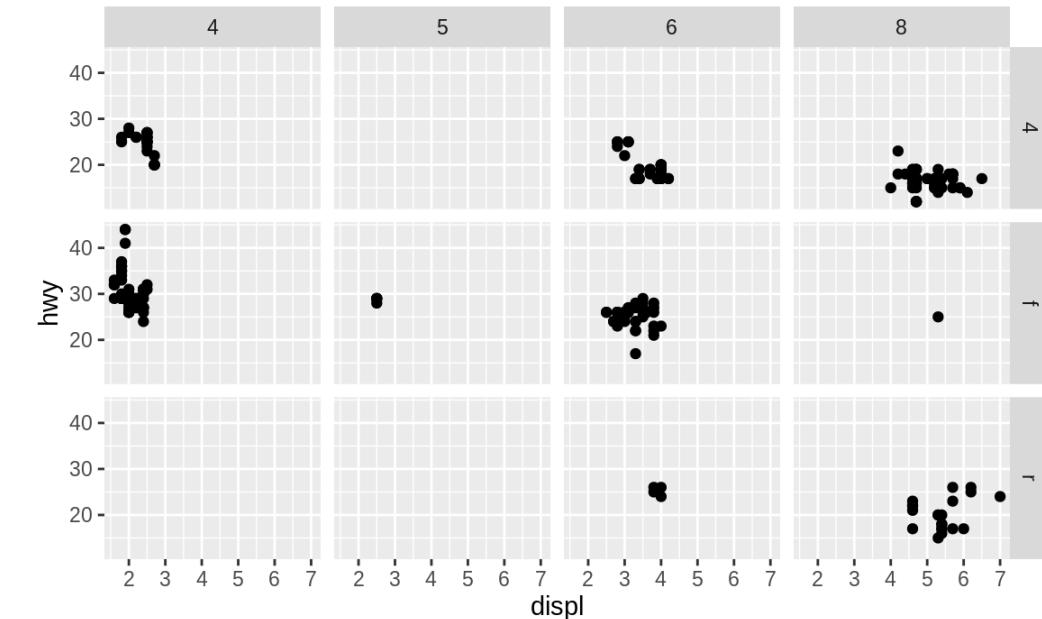
## 3.2.2 Gráficos de dispersión – Grids

- Para realizar la combinación de dos variables se agrega la instrucción `facet_grid()`
- El primer argumento, también es una formula.

```
ggplot(data = consumo) +  
  geom_point(aes(x= desp, y= autop))+  
  facet_grid(cil ~ trac)
```

- Si se desea omitir una columna o una fila se puede utilizar el carácter .
- ✓ Ejemplo:

```
ggplot(data = consumo) +  
  geom_point(aes(x= desp, y= autop))+  
  facet_grid(. ~ cil)
```





## 3.2 Gráficos de dispersión - Facetas – Grids - Ejercicios 4

### Ejercicio (10 – 15 minutos)

Utilizar el set de datos [iris.csv](#)

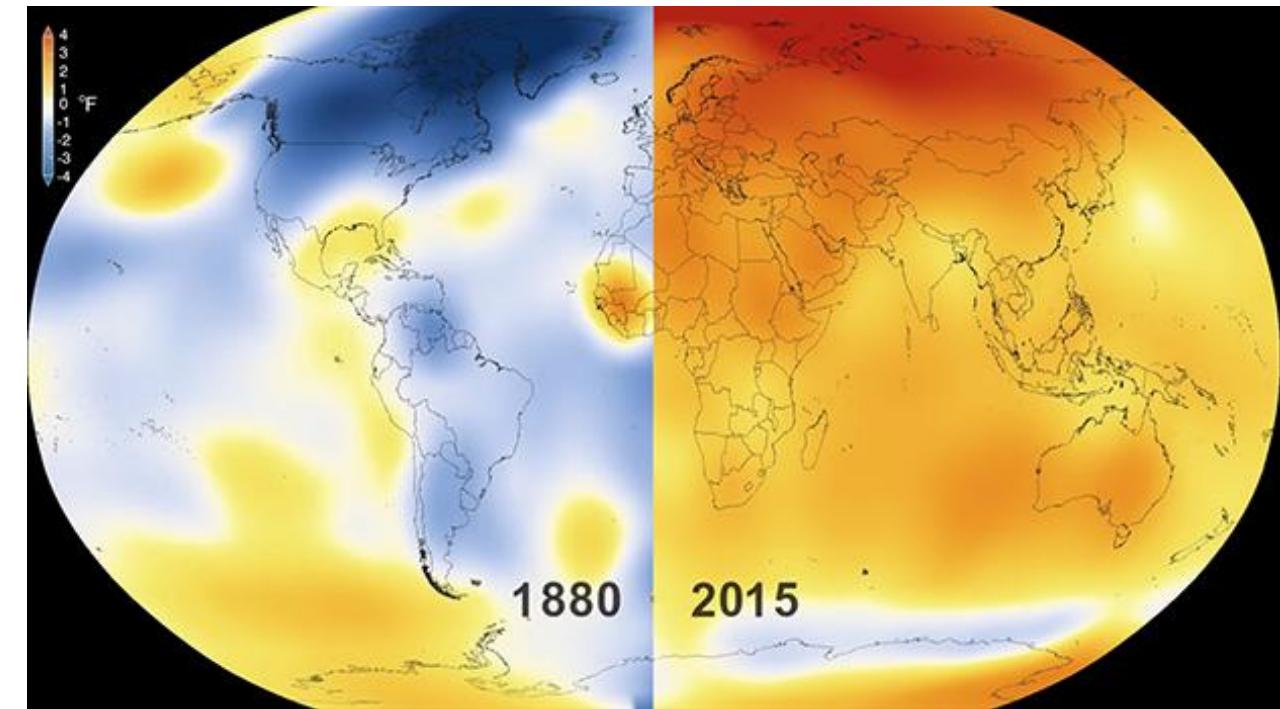
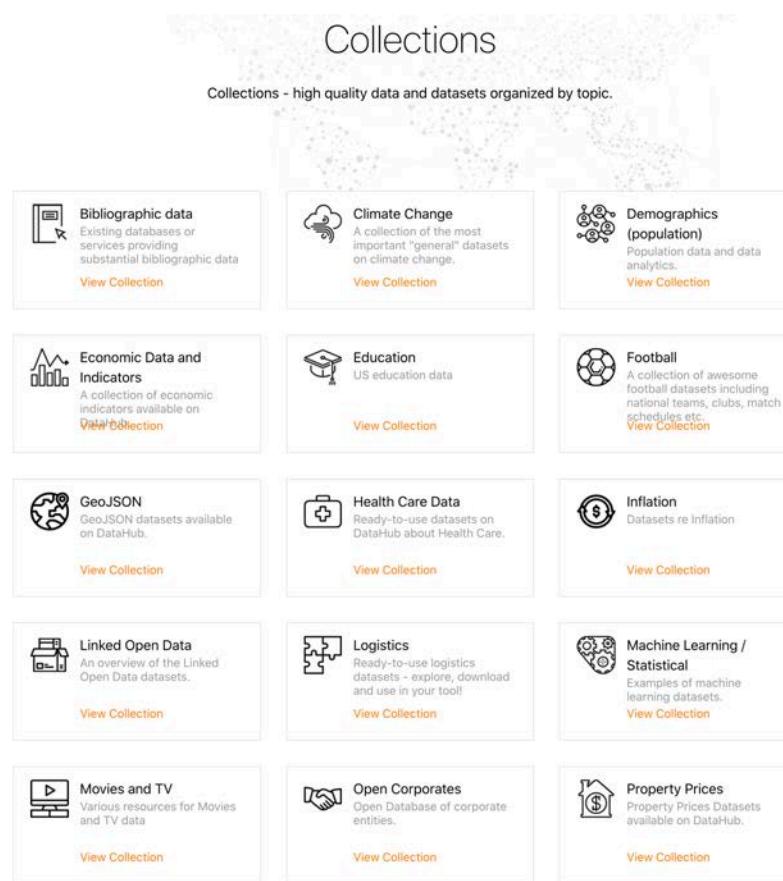
1. Se deberá generar un **gráfico de dispersión** utilizando ggplot2: **X** = longitud de sépalo, **Y** = longitud de pétalo.
  - a) Se deberán generar **dos gráficos**: Una estética distinta para cada uno.
  - b) Se deberán combinar 2 estéticas en una **tercer gráfica**.
2. Se deberán utilizar facetas para separar los datos por **especie**
  - ✓ Utilizar al menos una **estética**

# 3.3 Gráficos de línea

## Set de datos

DataHub: <https://datahub.io/core/global-temp#data>

Temperatura media de la tierra 1980 - 2016

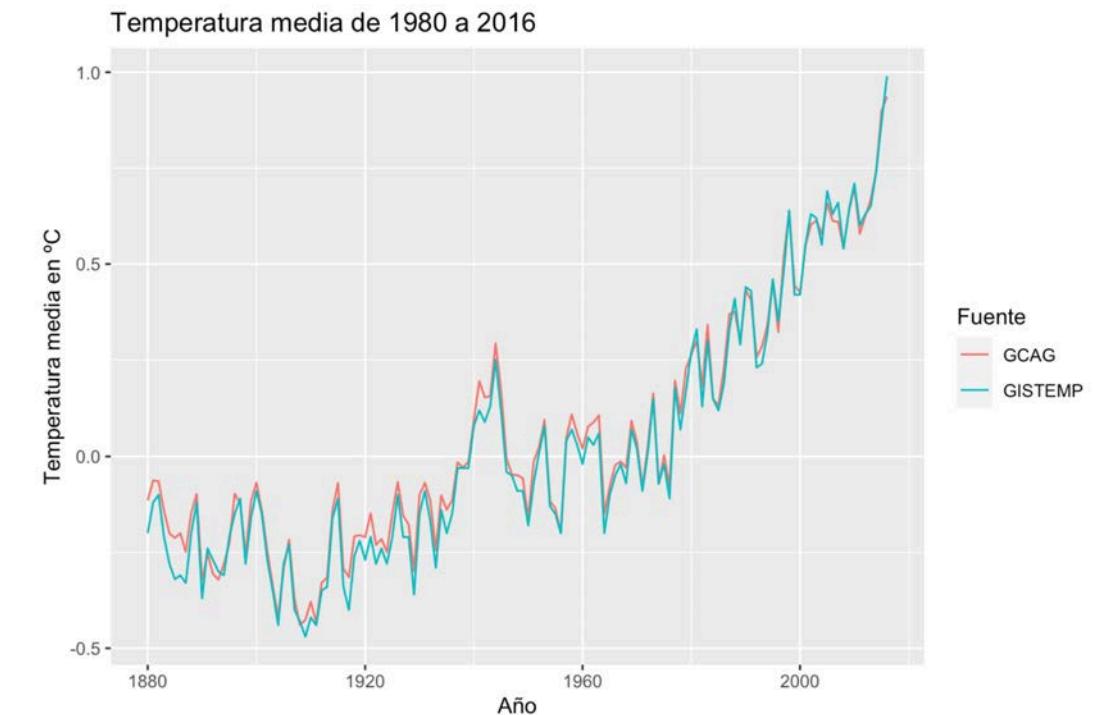
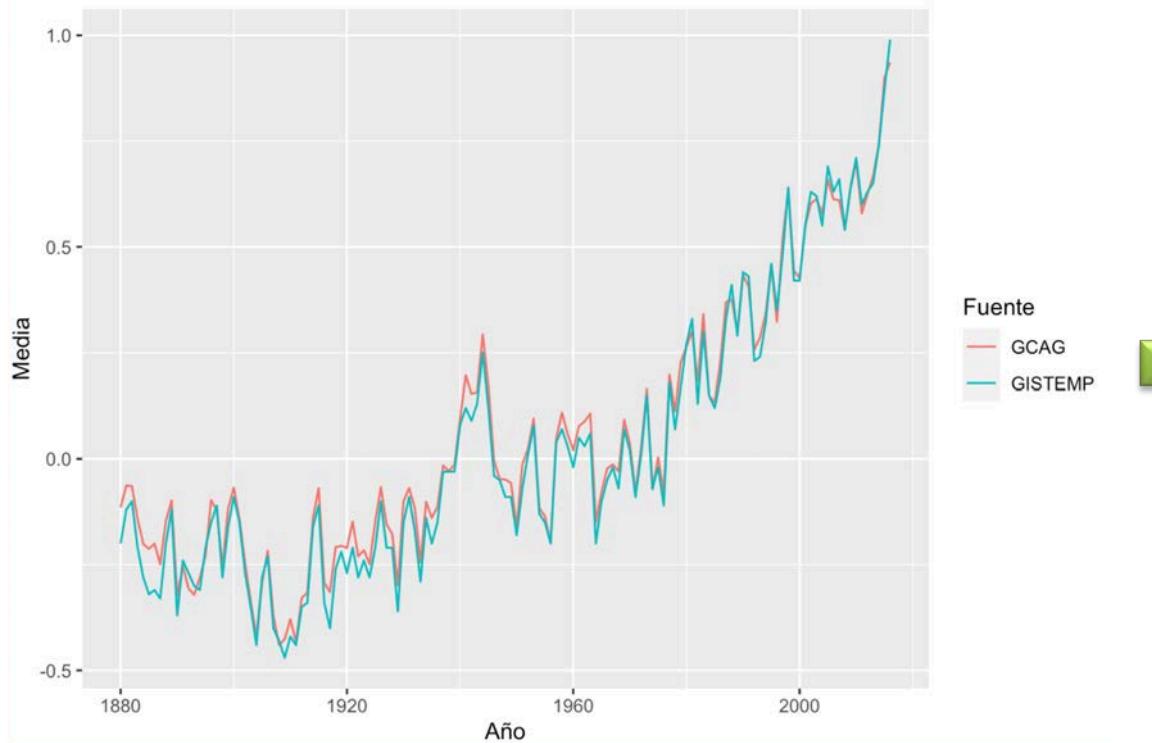




## 3.3 Gráficos de línea – Títulos

```
ggplot(data = temp) +  
  geom_line(aes(x = Año, y=Media, color = Fuente))
```

```
ggplot(data = temp) +  
  geom_line(aes(x = Año, y=Media, color = Fuente))+  
  labs(title = "Temperatura media de 1980 a 2016", x = "Año", y = ("Temperatura media en °C"))
```

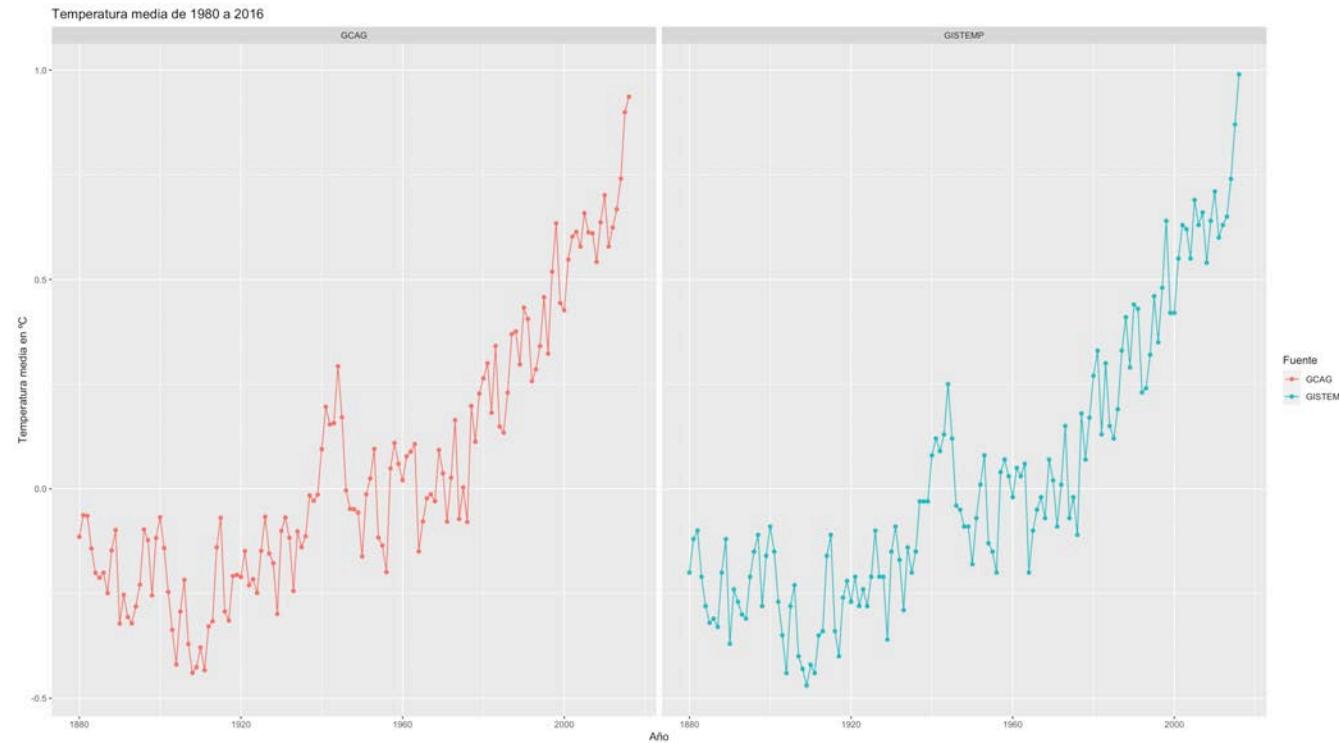




## 3.3 Gráficos de línea – Ejercicio 5

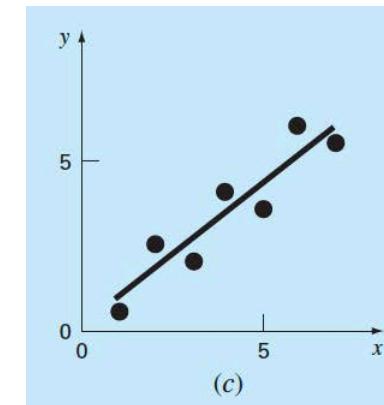
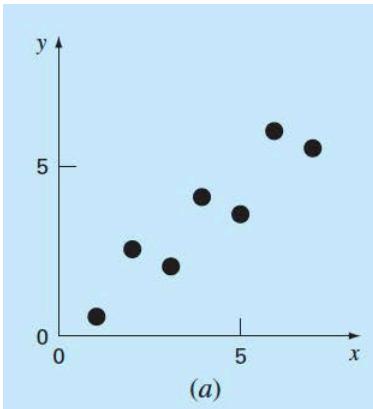
### Ejercicio (5 minutos)

Se deberán separar los gráficos por **Fuente**, empleando como referencia el gráfico anterior a través del uso de **facetas**.



## 3.3 Gráficos de línea – regresiones

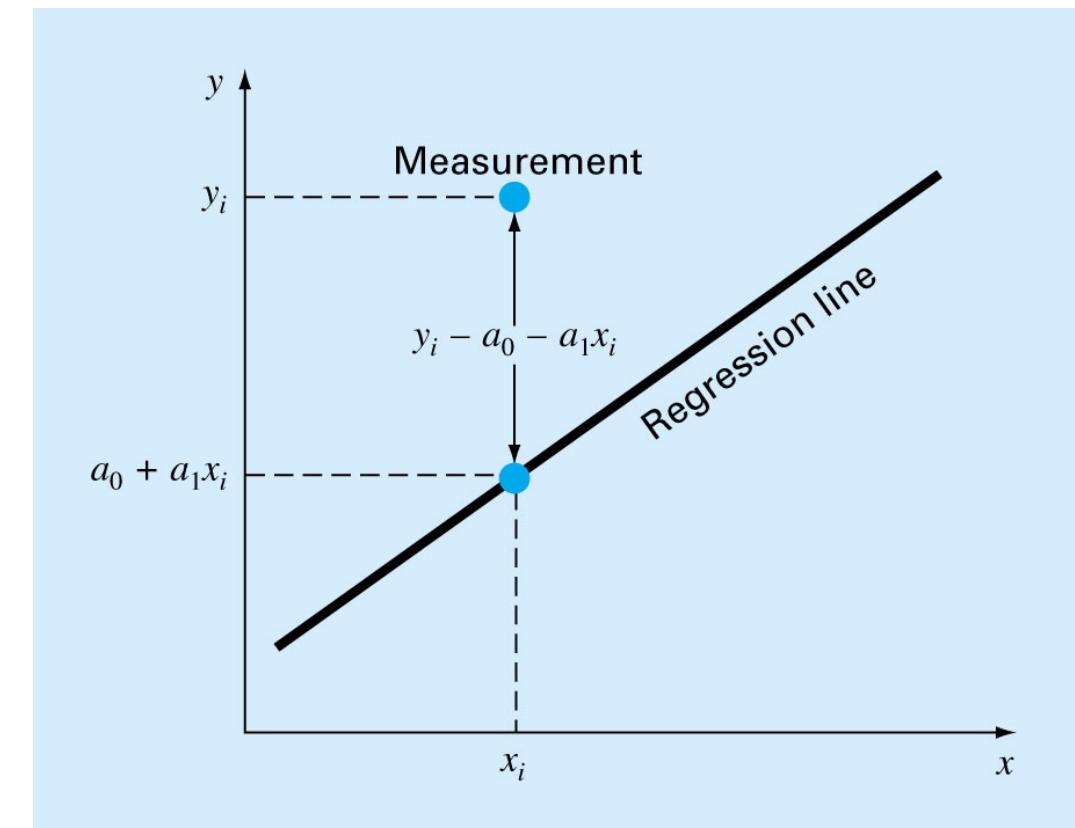
- El ejemplo más simple de una regresión es una **línea recta**, a través de la cual se pueden aproximar  $n$  puntos:  $\{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$ .



$$y = mx + b$$

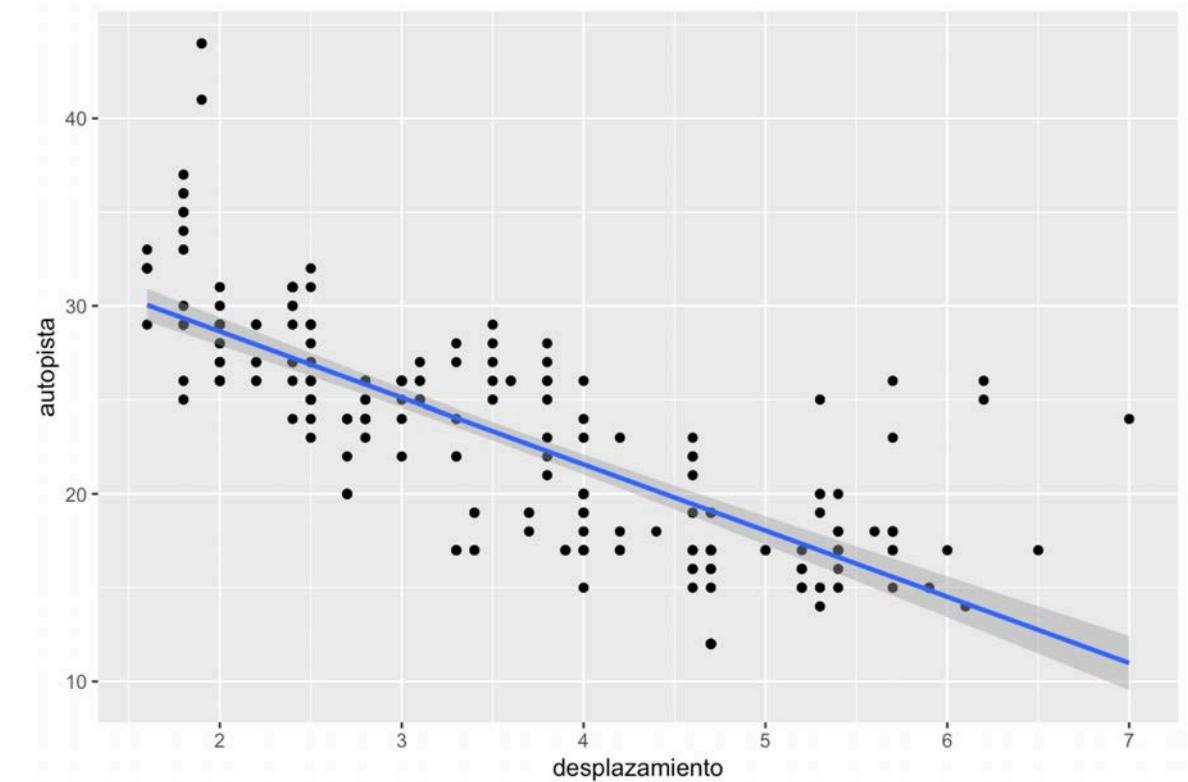
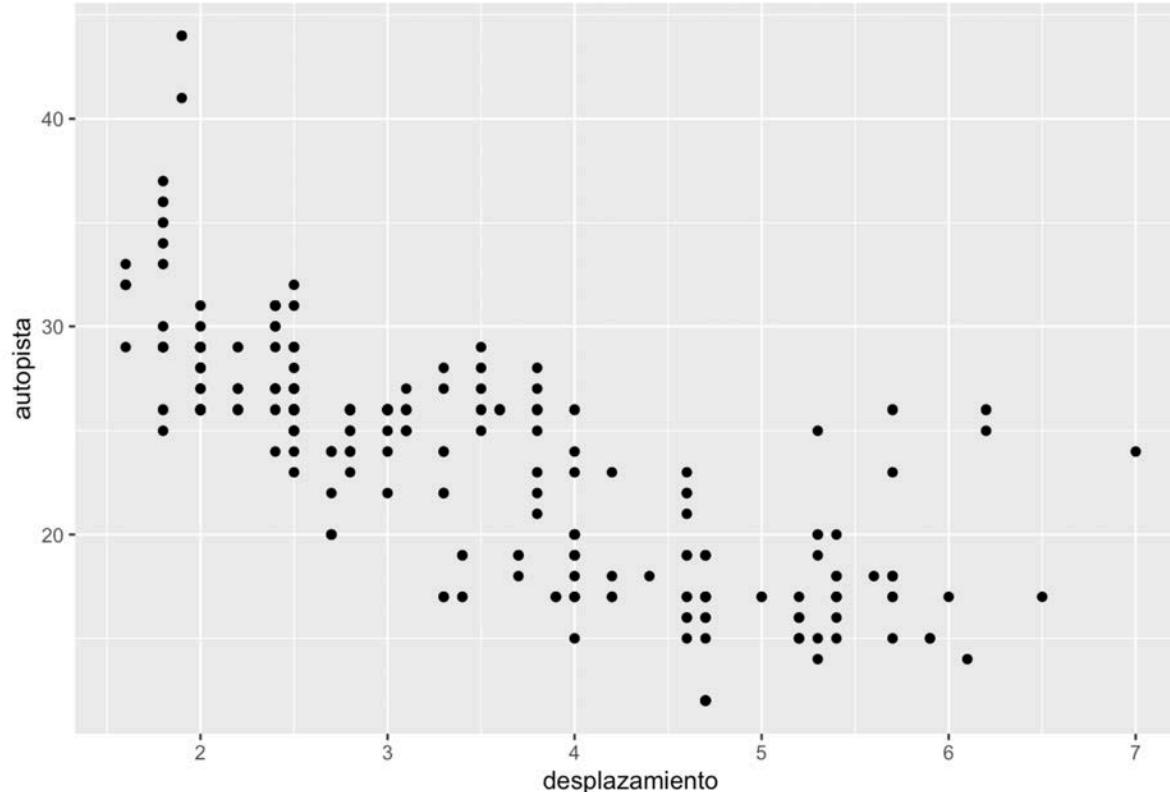
Ecuación de la recta

$m$  = pendiente,  $b$  = intercepto



## 3.3 Gráficos de línea – regresiones

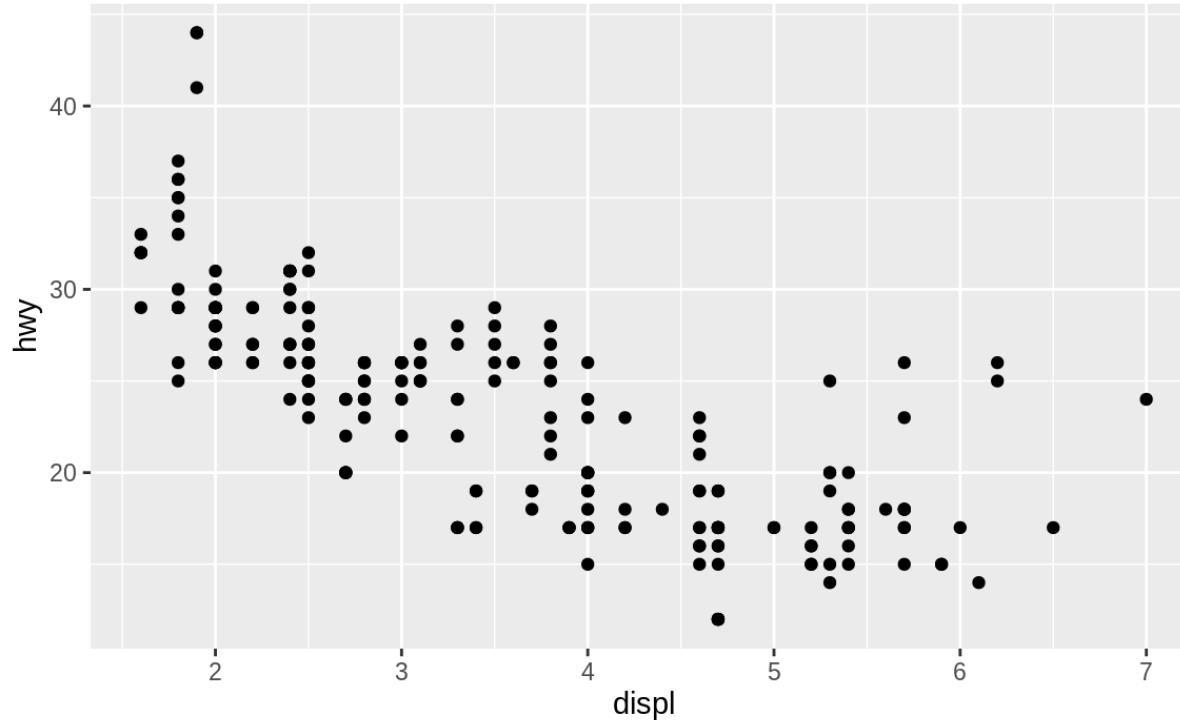
```
ggplot (data=consumo, aes(x = desplazamiento, y = autopista))+  
  geom_point()  
  geom_smooth(method="lm")
```



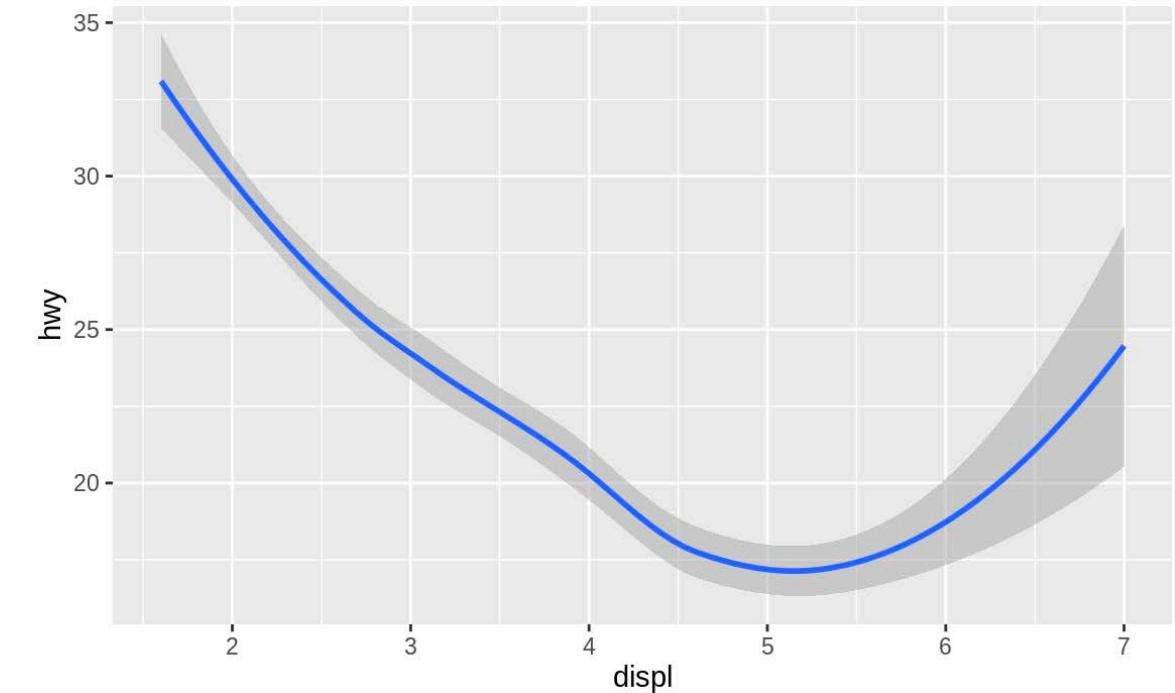


## 3.3 Gráficos de línea – regresiones

¿Qué tan distintas son estas gráficas?



```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```



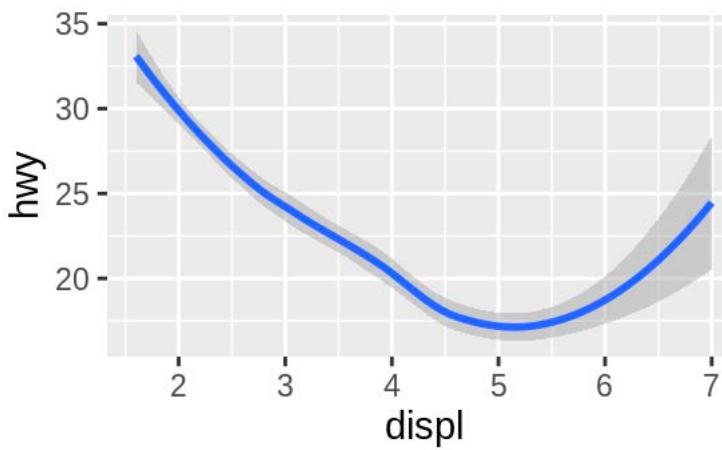
```
ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```



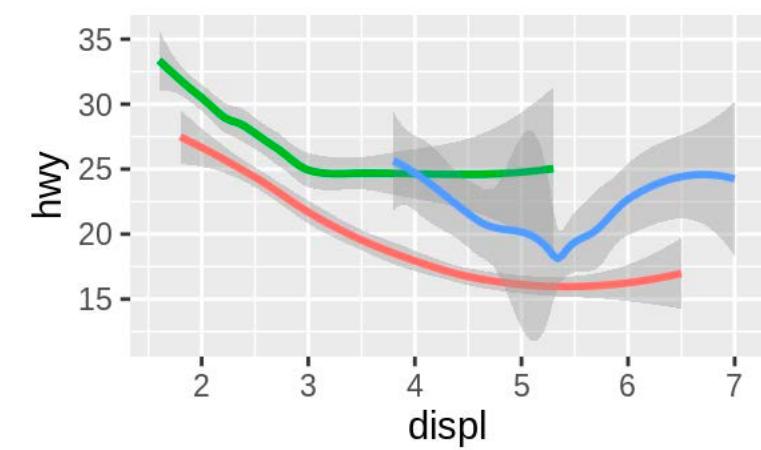
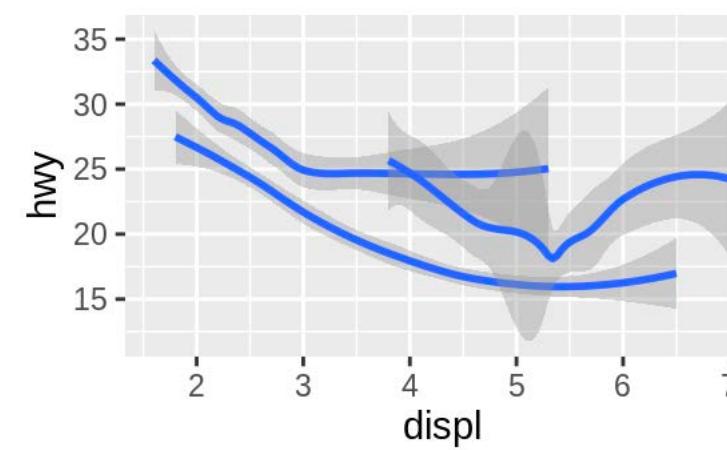
## 3.3 Gráficos de línea – regresiones

```
ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y = hwy, group = drv))
```

```
ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```



```
ggplot(data = mpg) +  
  geom_smooth(  
    mapping = aes(x = displ, y = hwy, color = drv),  
    show.legend = FALSE  
  )
```



## 3.3 Gráficos de línea – regresiones

- Despliegue de geometrías multiples

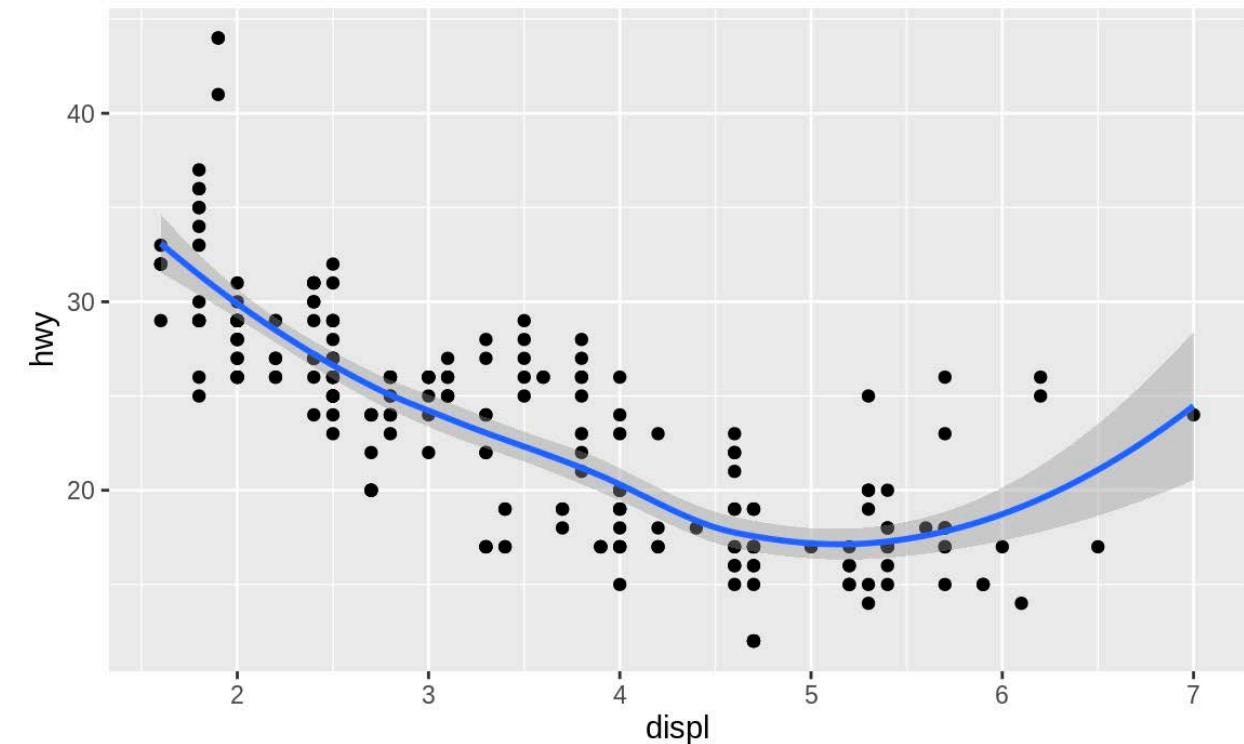
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```



```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point() +  
  geom_smooth()
```

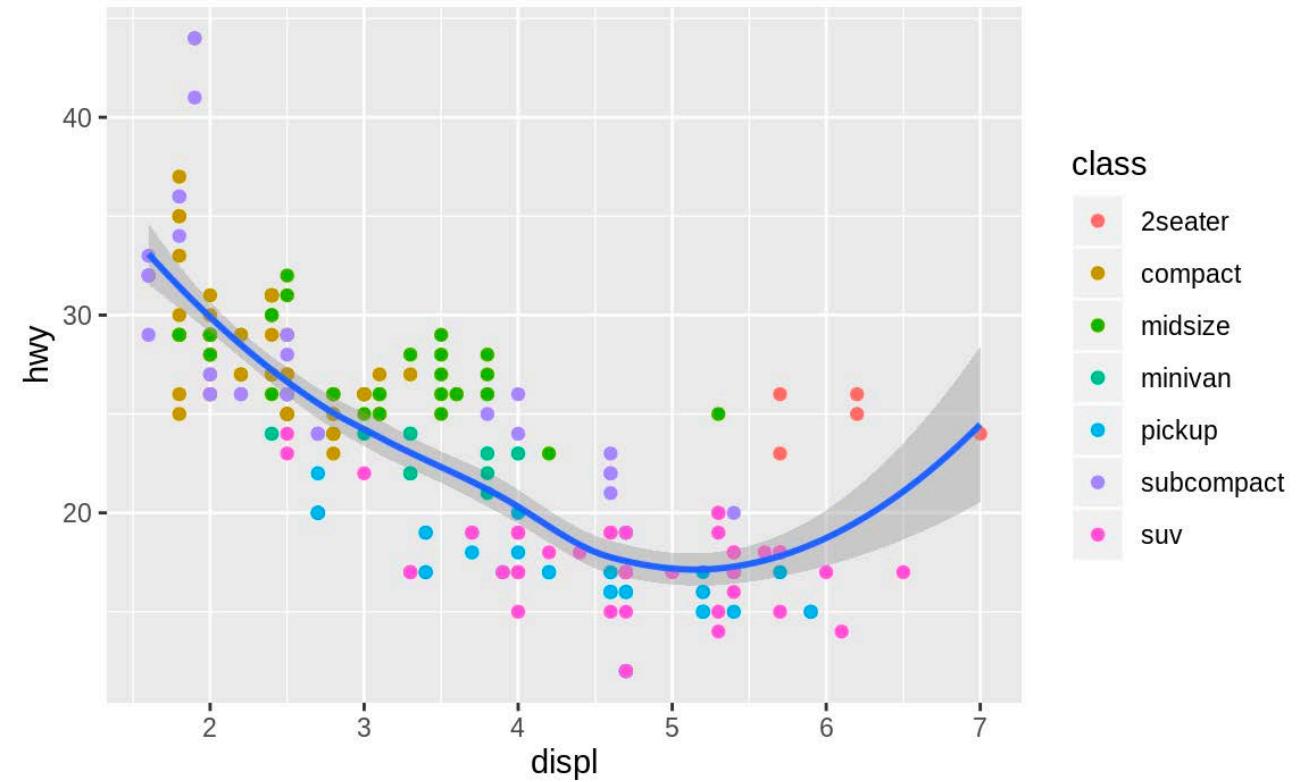
- ¿Para que sirve el parametro span?

✓ ?span



### Ejercicio (5-7) Minutos:

Modifique los parametros de ggplot para obtener la siguiente figura:



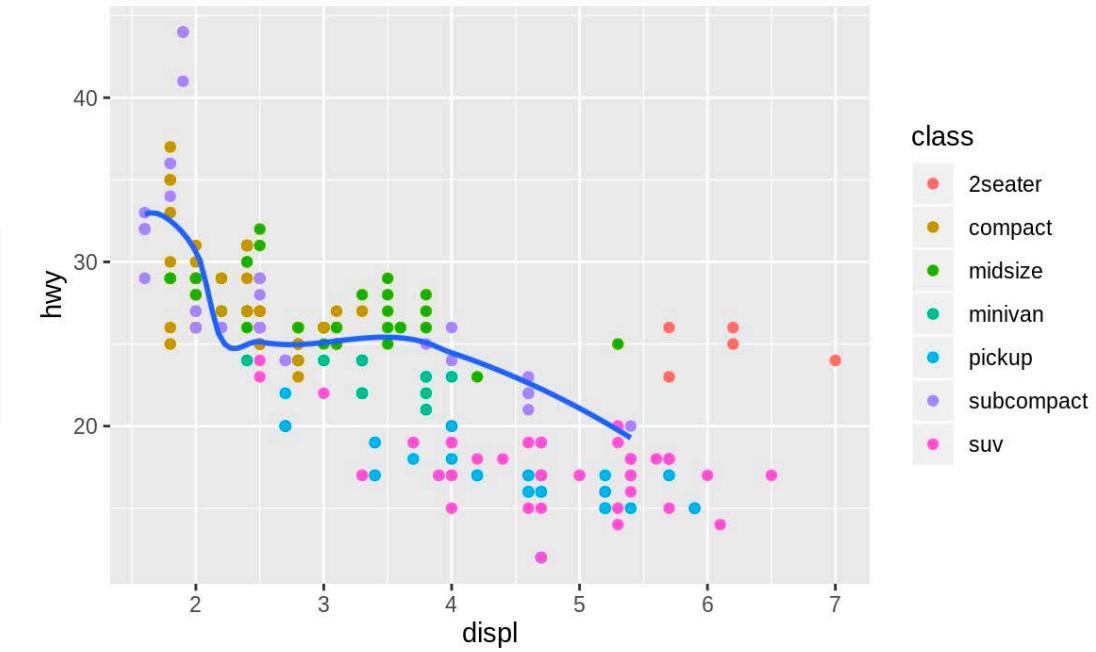


## 3.3 Gráficos de línea – regresiones – se

¿Para qué sirve el parametron **se**?

?geom\_smooth

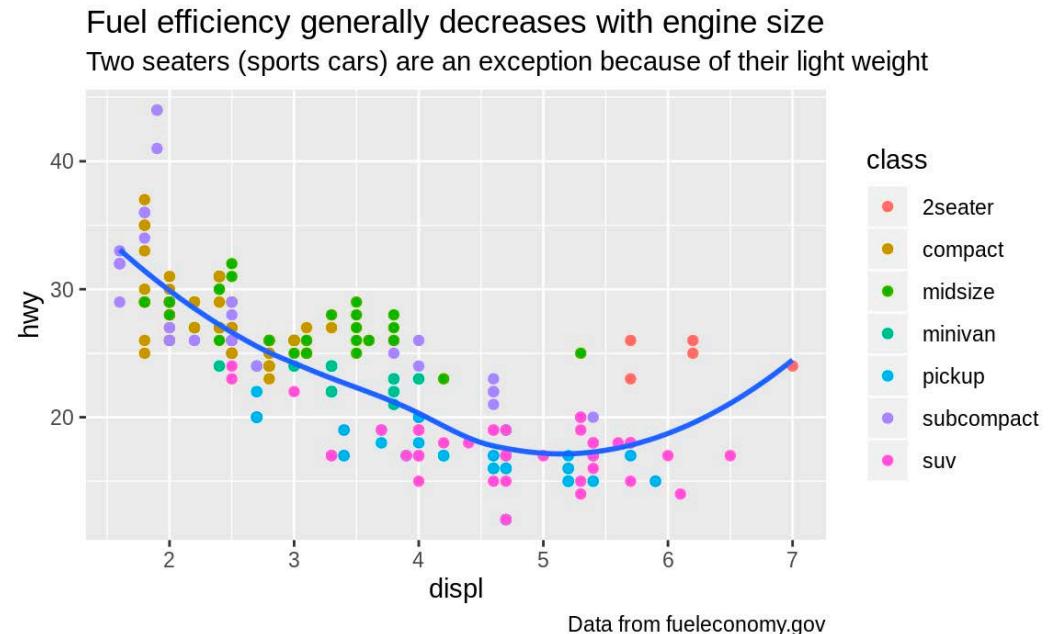
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point(mapping = aes(color = class)) +  
  geom_smooth(data = filter(mpg, class == "subcompact"), se = FALSE)
```



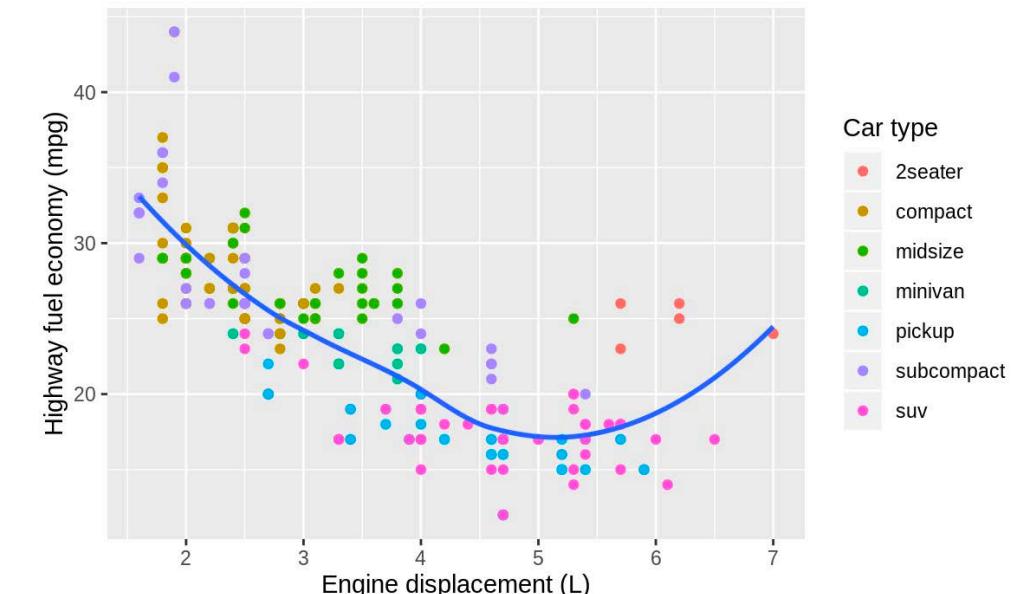
### 3.3 Gráficos de línea – regresiones – títulos

A través de etiquetas es posible establecer títulos del gráfico, de los ejes subtítulos leyendas, etc

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  labs(
    title = "Fuel efficiency generally decreases with engine size",
    subtitle = "Two seaters (sports cars) are an exception because of their light weight",
    caption = "Data from fueleconomy.gov"
  )
```

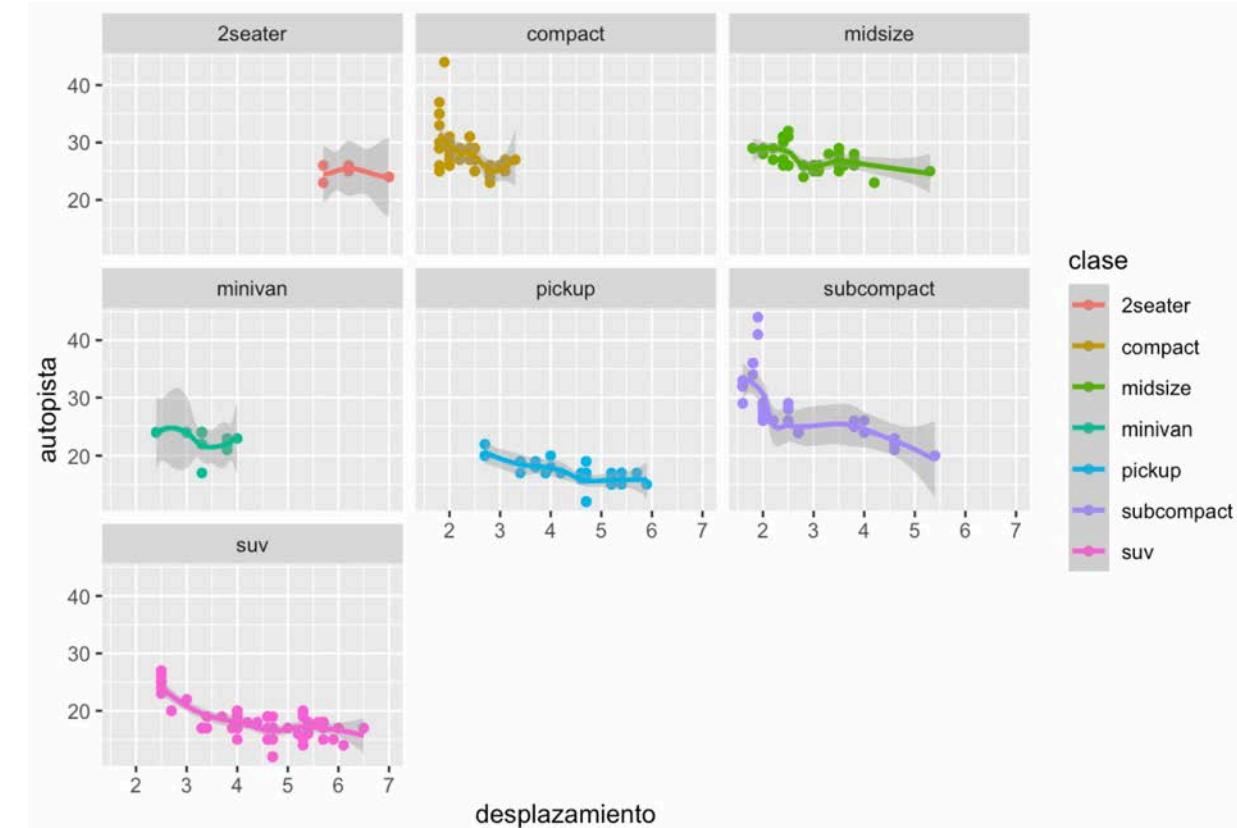


```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(colour = class)) +
  geom_smooth(se = FALSE) +
  labs(
    x = "Engine displacement (L)",
    y = "Highway fuel economy (mpg)",
    colour = "Car type"
  )
```



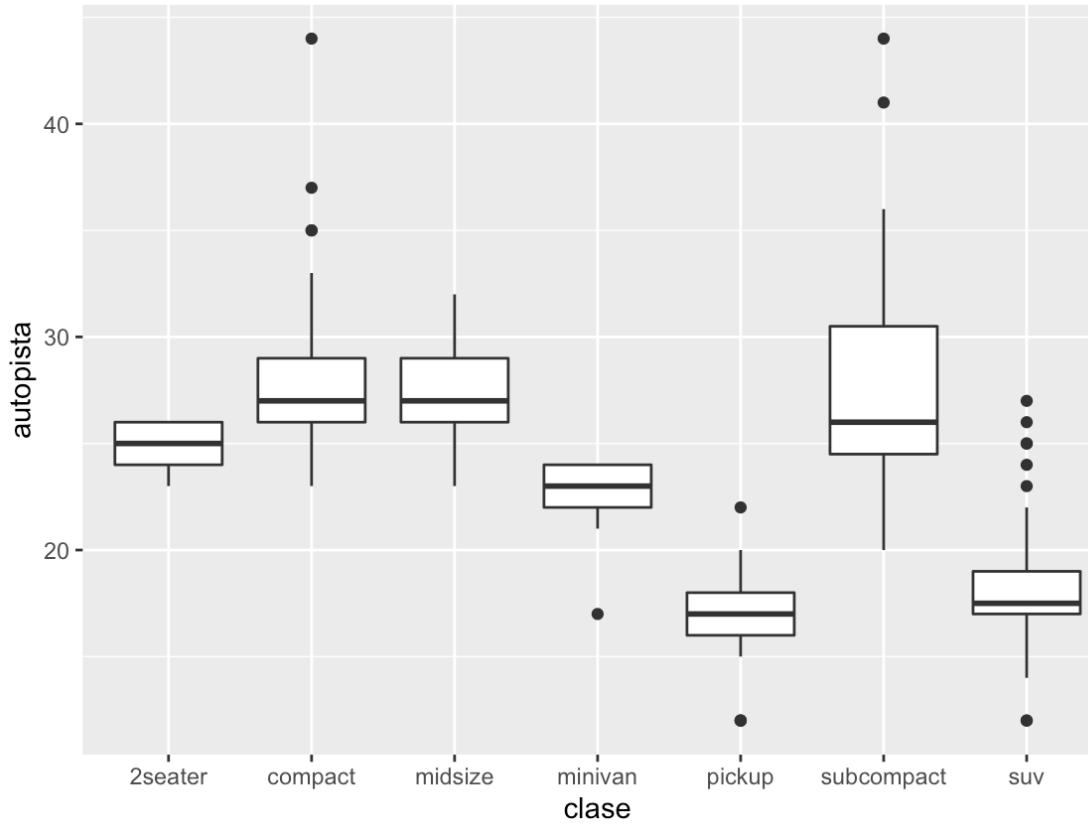
## Ejercicio (5-7 minutos)

- Utilice **facetas** para agrupar los gráficos de dispersión y su respectiva regresión polinomial por clase de automóvil.

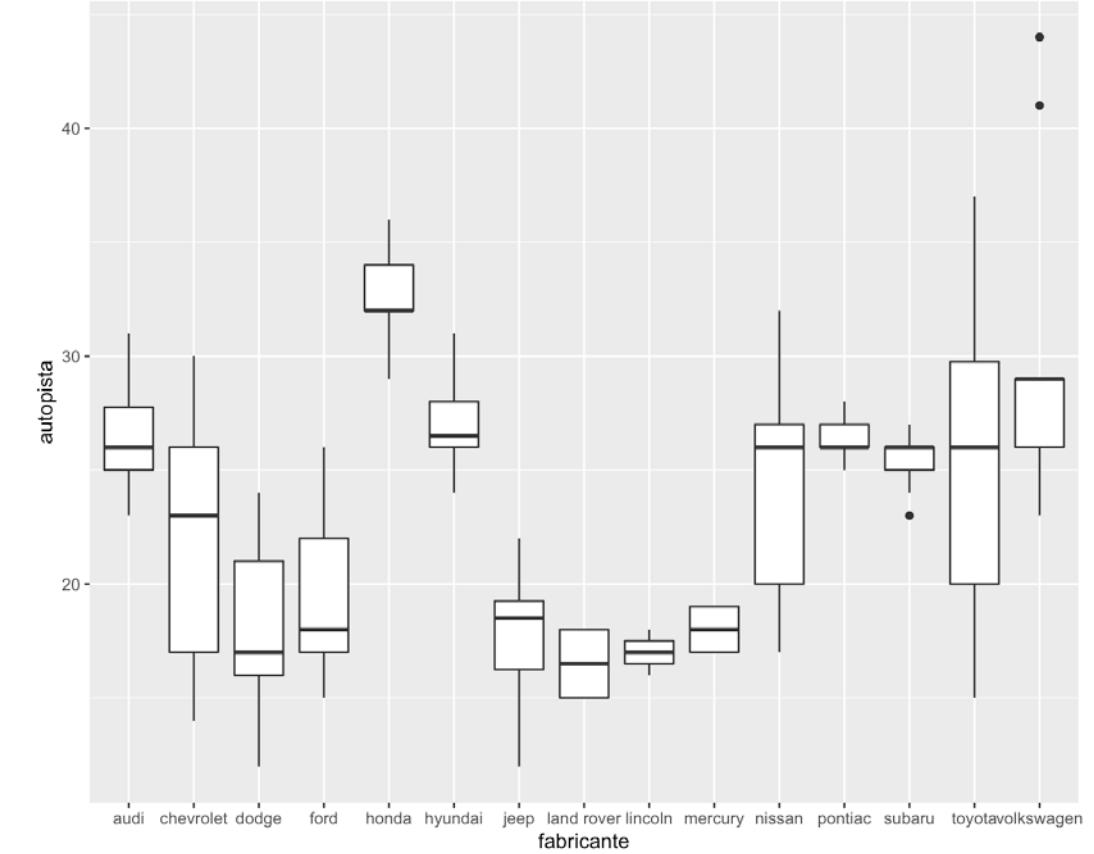


### 3.3 Boxplots

```
ggplot(data=consumo, aes(x =clase, y=autopista))+  
  geom_boxplot()
```



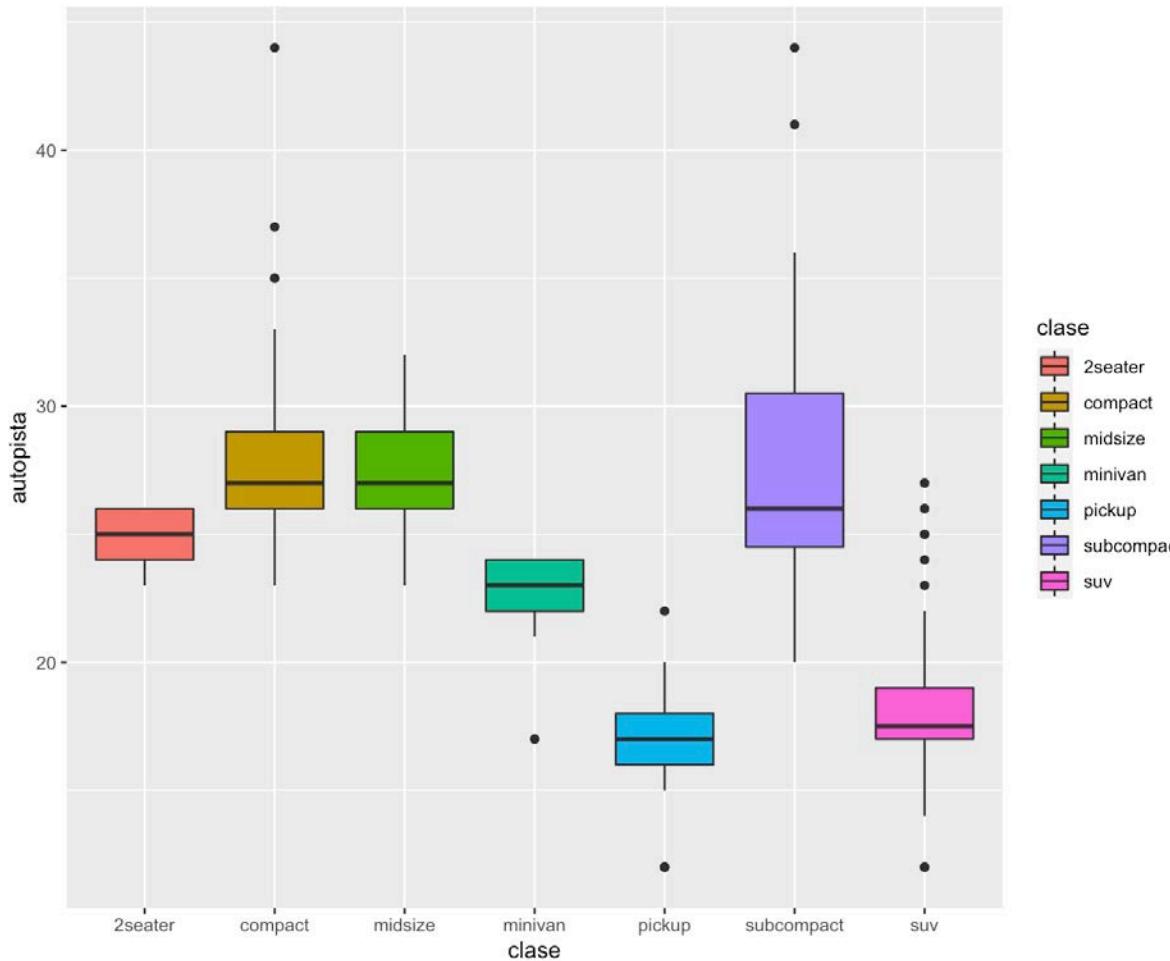
```
ggplot(data=consumo, aes(x =fabricante, y=autopista))+  
  geom_boxplot()
```



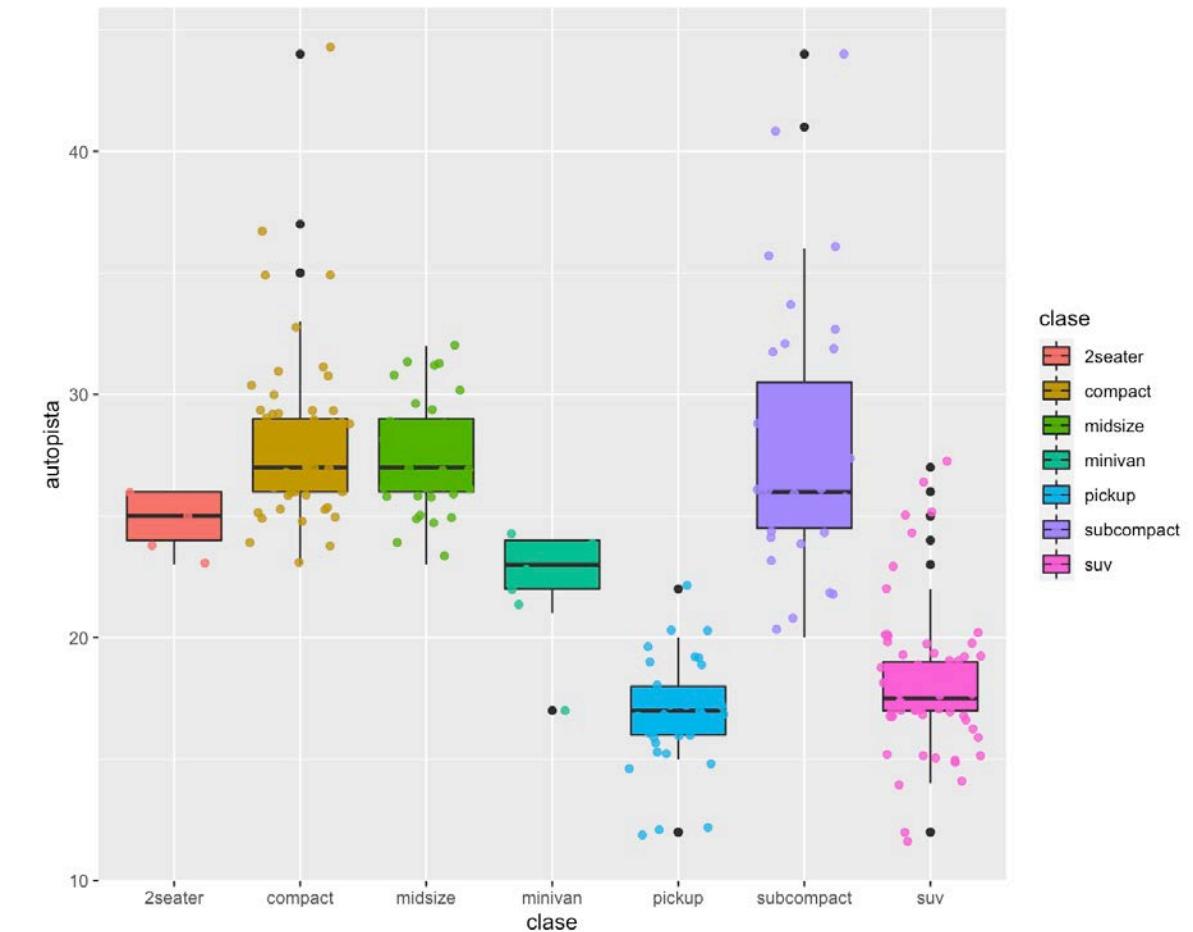


### 3.3 Boxplots – Color – Jitter

```
ggplot(data=consumo, aes(x =clase, y=autopista, , fill= clase))+  
  geom_boxplot()
```

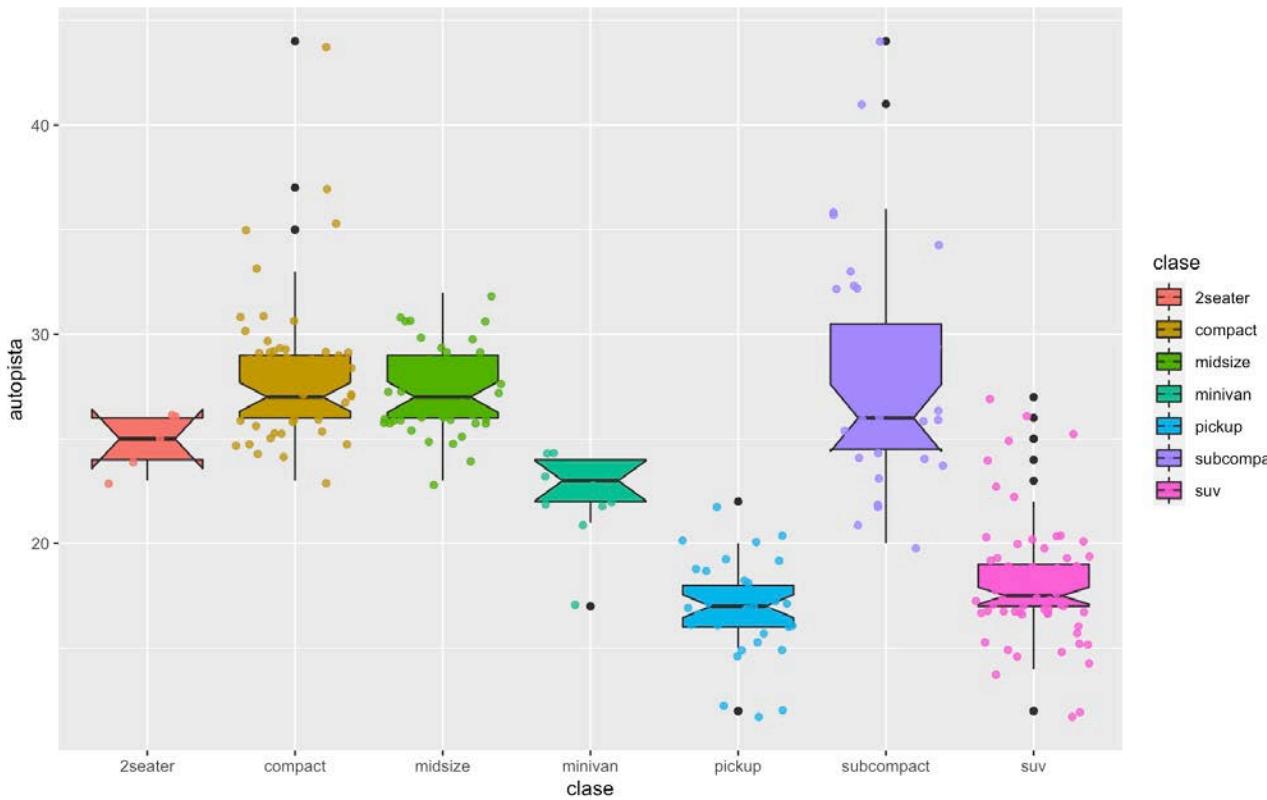


```
ggplot(data=consumo, aes(x =clase, y=autopista, , fill= clase))+  
  geom_boxplot() +  
  geom_jitter(aes(color = clase), alpha= 0.8)
```

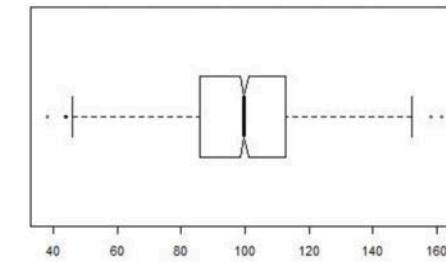


### 3.3 Boxplots – Notch (muesca)

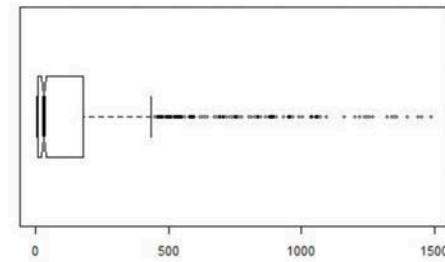
```
ggplot(data=consumo, aes(x = clase, y=autopista, , fill= clase))+  
  geom_boxplot(notch = TRUE)+  
  geom_jitter(aes(color = clase), alpha= 0.8)
```



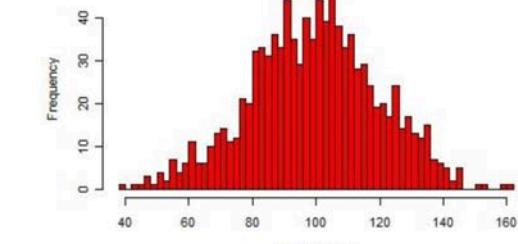
Notched Box Plot Normal Data



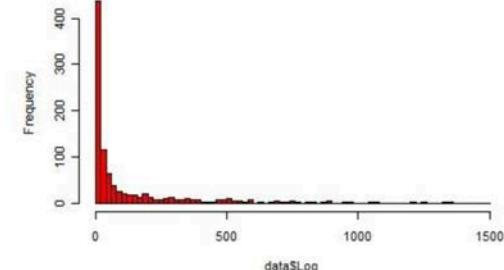
Notched Box Plot of Skewed Data



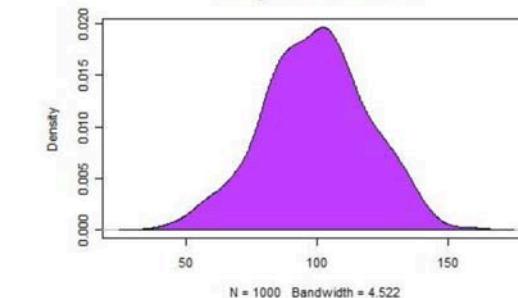
Histogram of Normal Data



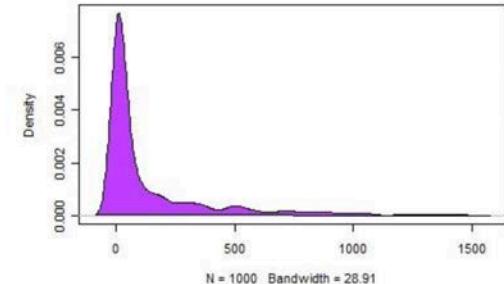
Histogram of Skewed Data



Density Plot of Normal Data



Density Plot of Skewed Data



### 3.3 Boxplots – Subconjuntos – Tibble

```
library(tidyverse)
```

```
data <- as_tibble(consumo)

pickup <- data %>% filter(consumo$clase == "pickup")

plot(density(pickup$autopista))
```

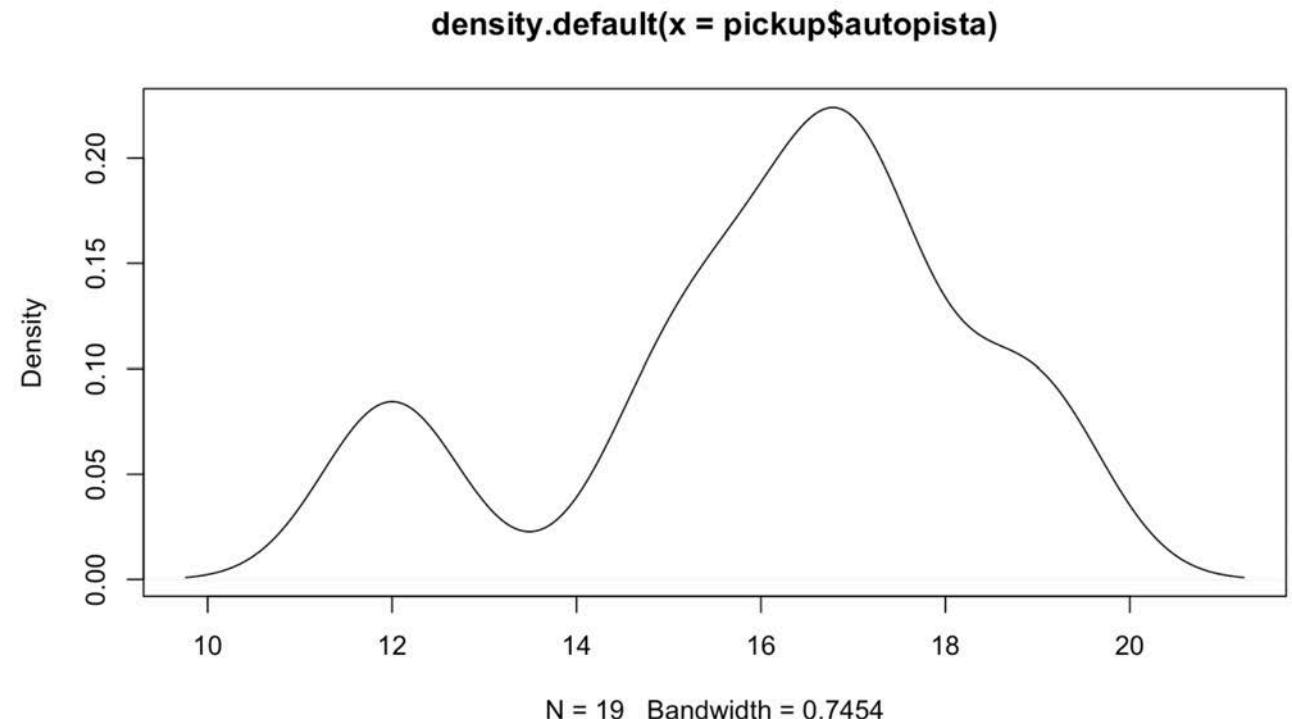
**filter()**

storms

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Alex	45	1009	1998-07-30
Allison	65	1005	1995-06-04
Ana	40	1013	1997-07-01
Arlene	50	1010	1999-06-13
Arthur	45	1010	1996-06-21

→

storm	wind	pressure	date
Alberto	110	1007	2000-08-12
Allison	65	1005	1995-06-04

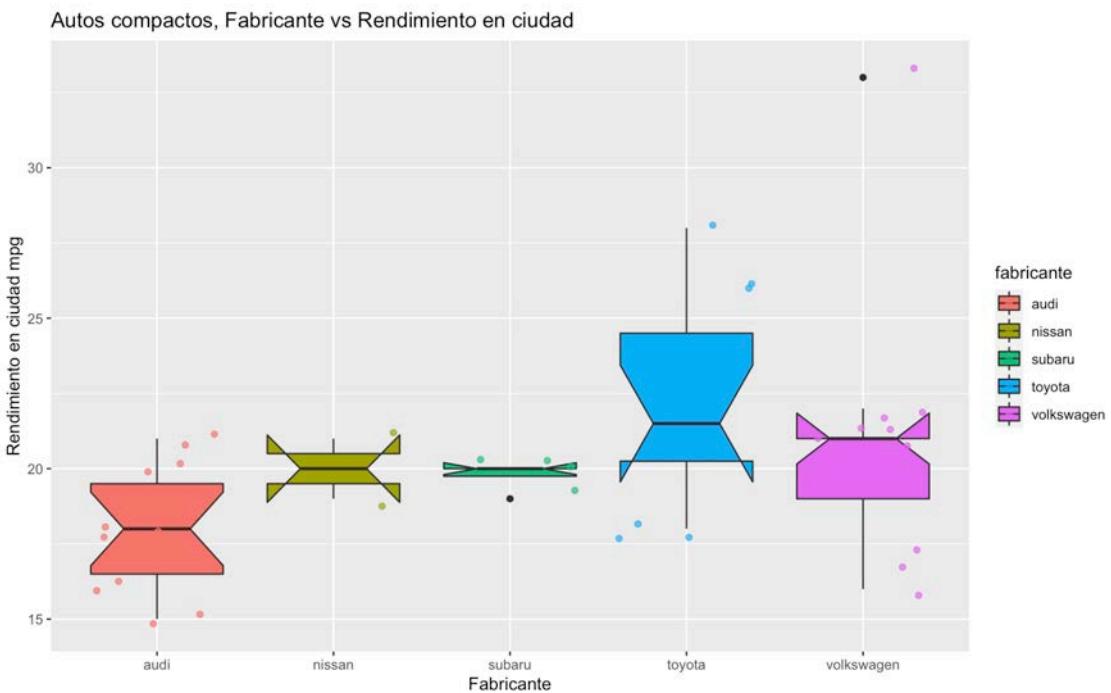


## 3.3 Boxplots – Ejercicio 8

### Ejercicio (10 – 12 minutos)

Se deberá generar un boxplot de los automóviles compactos comparando la marca del auto (fabricante) vs el consumo de combustible en ciudad.

- ✓ Utilice la geometría de jitter para observar la varianza
- ✓ Utilice muestras para conocer la distribución de los datos.





## 3.4 Ggplot – Transformaciones estadísticas

- Hasta ahora, se han utilizado gráficos de dispersión y de línea para conocer el comportamiento de los datos.
- Es posible utilizar de gráficos de barra para generar de conteo tales como histogramas.
- Se utilizará un nuevo set de datos denominado **diamonds**.
  - ✓ ~54,000 elementos
- Utilizar ayuda para ver los detalles de data frame **?diamonds**.

### Format

A data frame with 53940 rows and 10 variables:

**price** price in US dollars (‐\$326--‐\$18,823)

**carat** weight of the diamond (0.2--5.01)

**cut** quality of the cut (Fair, Good, Very Good, Premium, Ideal)

**color** diamond colour, from D (best) to J (worst)

**clarity** a measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))

**x** length in mm (0--10.74)

**y** width in mm (0--58.9)

**z** depth in mm (0--31.8)

**depth** total depth percentage =  $z / \text{mean}(x, y) = 2 * z / (x + y)$  (43--79)

**table** width of top of diamond relative to widest point (43--95)

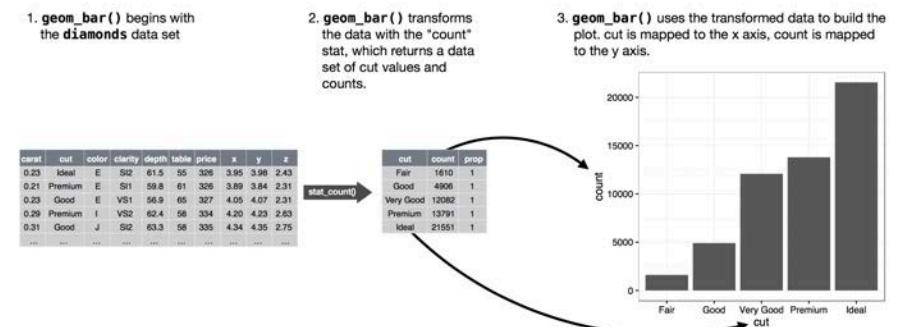
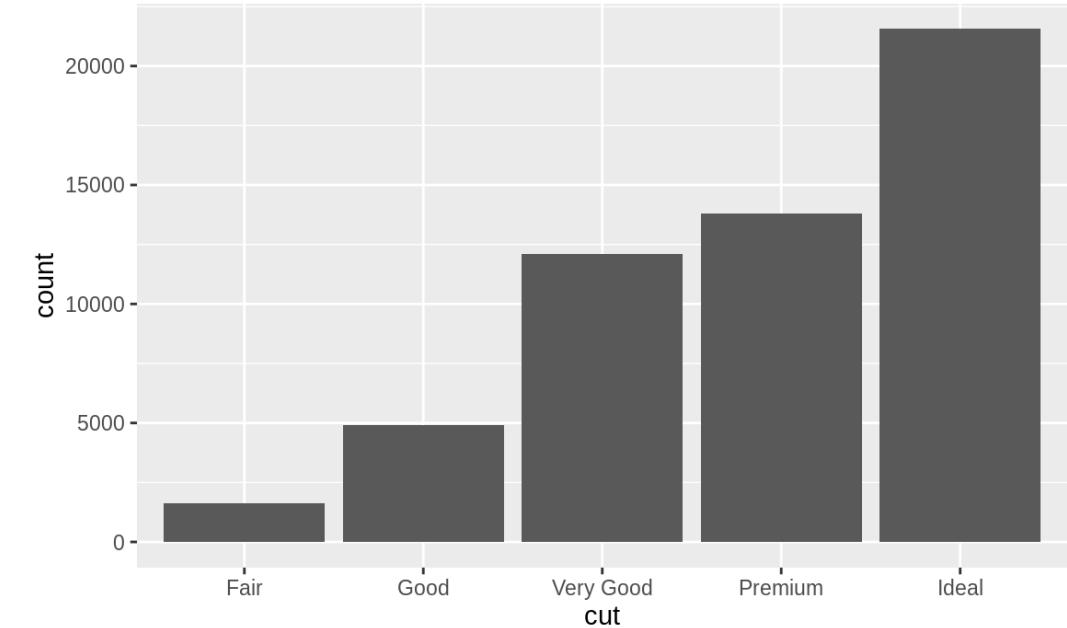
## 3.4 Ggplot – Transformaciones estadísticas – Gráfico de Barras

- Considere el código siguiente:

```
ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut))
```

- Dicho código genera el siguiente histograma. En código solo se especifica el eje de las X.
- ¿De donde viene el número de ocurrencias?

- ✓ Las geometrías similares a gráficos de barras (bar plots) calculan nuevos valores a graficar .

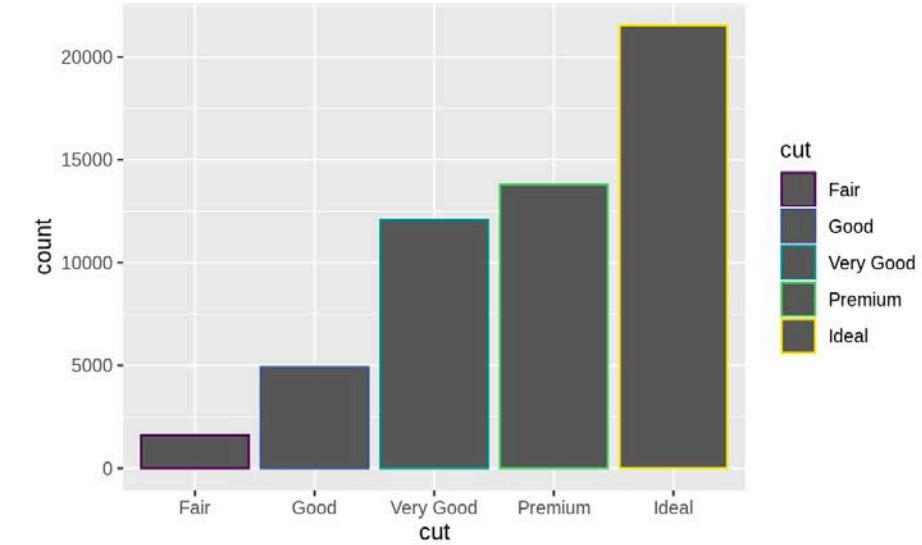


## 3.4 Ggplot – Transformaciones estadísticas – Gráfico de Barras

- También es posible utilizar **estéticas** en gráficos de barras, existen dos muy útiles

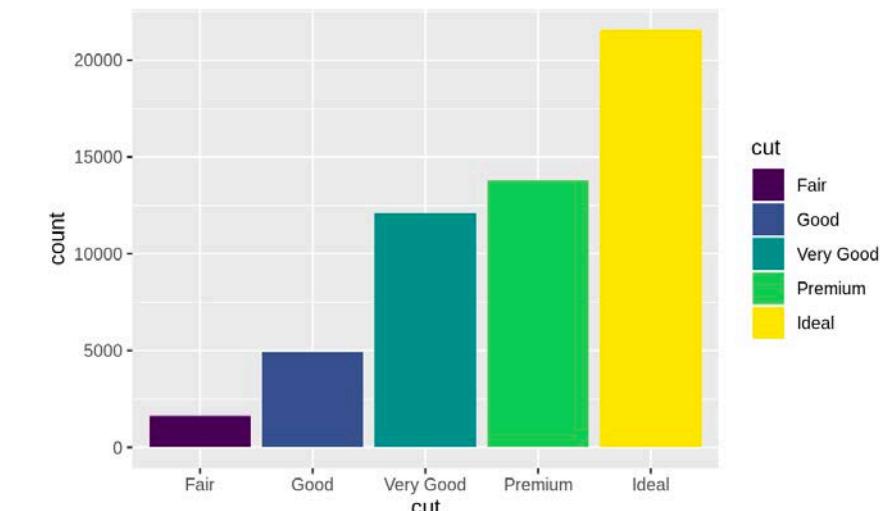
### ✓ Color o color

```
ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, colour = cut))
```



### ✓ Fill

```
ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = cut))
```



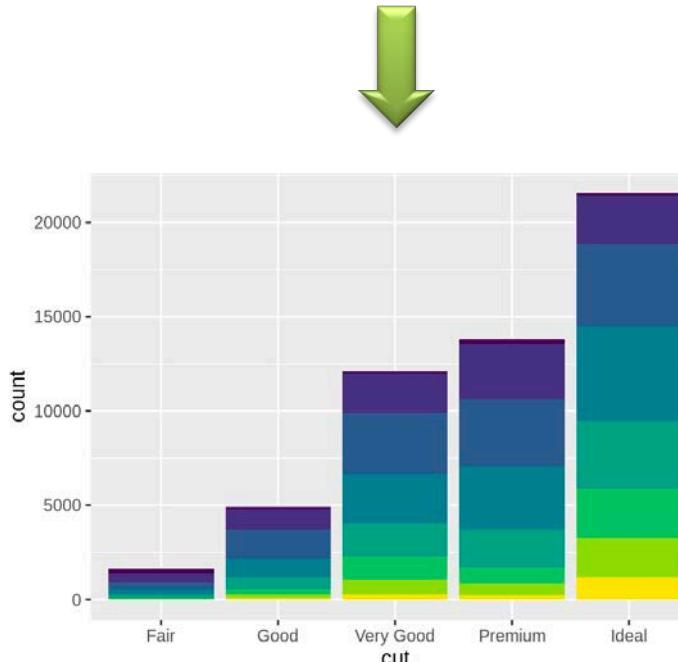
**Ejercicio:** Utilizar **clarity** en la estética de fill y verificar su comportamiento



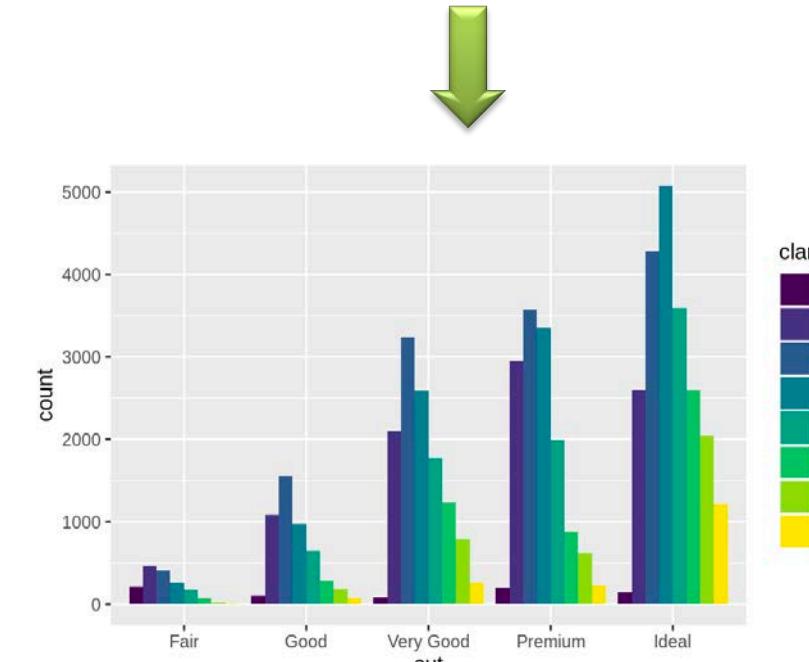
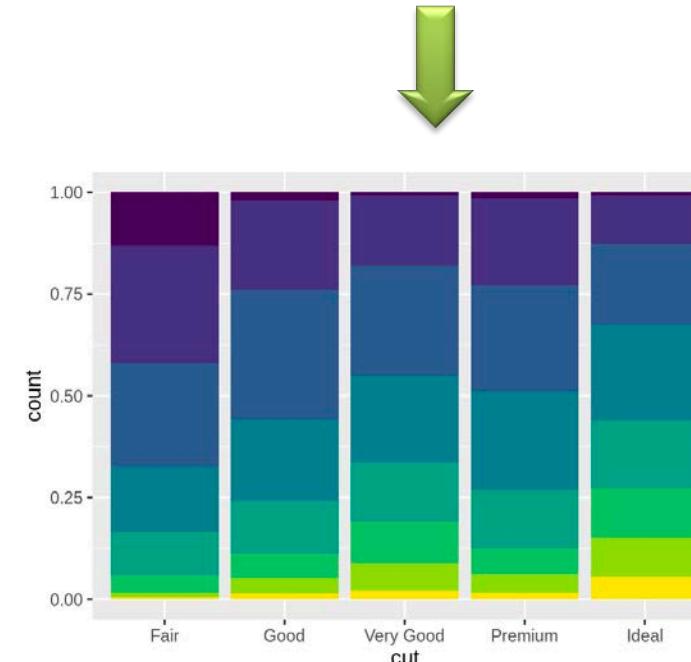
## 3.4 Ggplot – Transformaciones estadísticas – Gráfico de Barras

```
ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = clarity), position = "fill")
```

```
ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = clarity))
```



```
ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = clarity), position = "dodge")
```

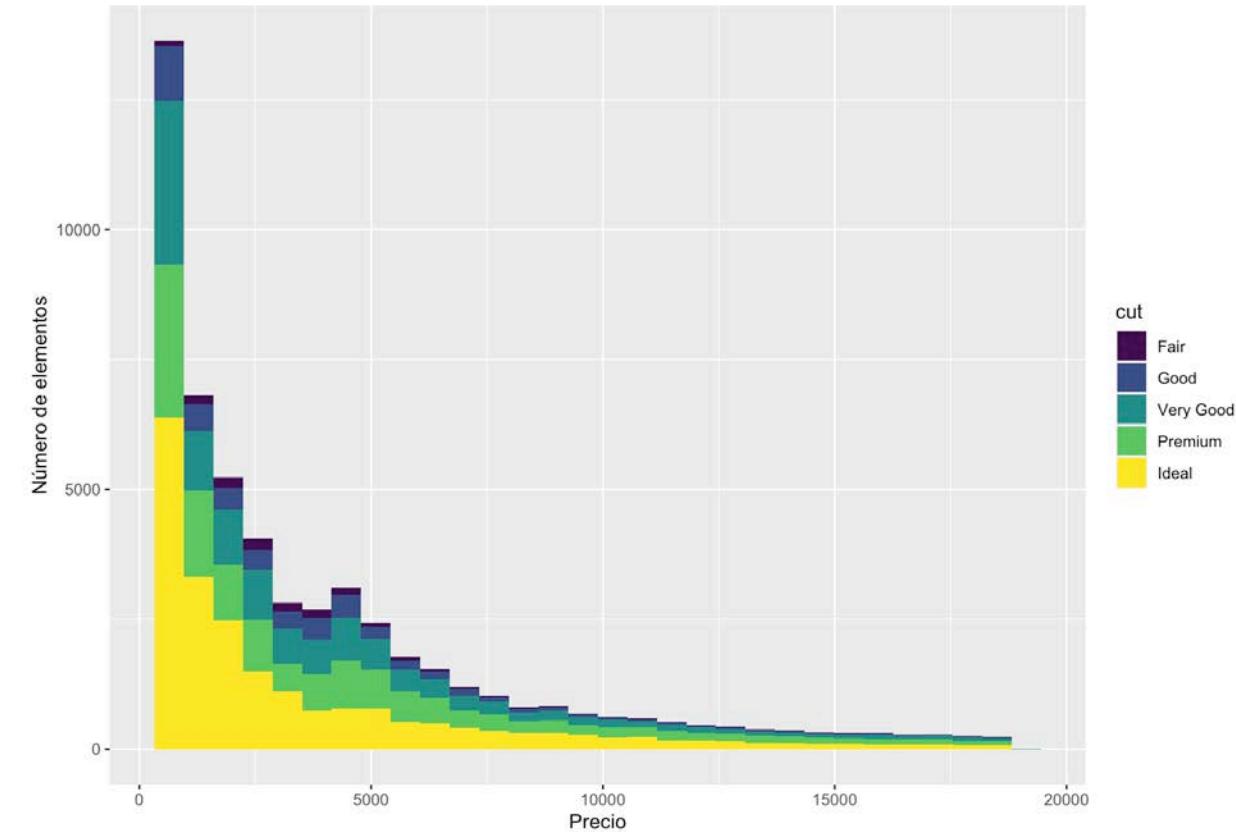


## 3.4 Ggplot – Transformaciones estadísticas – Histogramas

La principal diferencia entre las geometrías de:

- ✓ `geom_bar`: Valores discretos
- ✓ `geom_histogram`: Valores Continuos

```
ggplot(data = diamonds) +  
  geom_histogram(aes(x = price , fill = cut )) +  
  labs(x = "Precio", y = "Número de elementos")
```



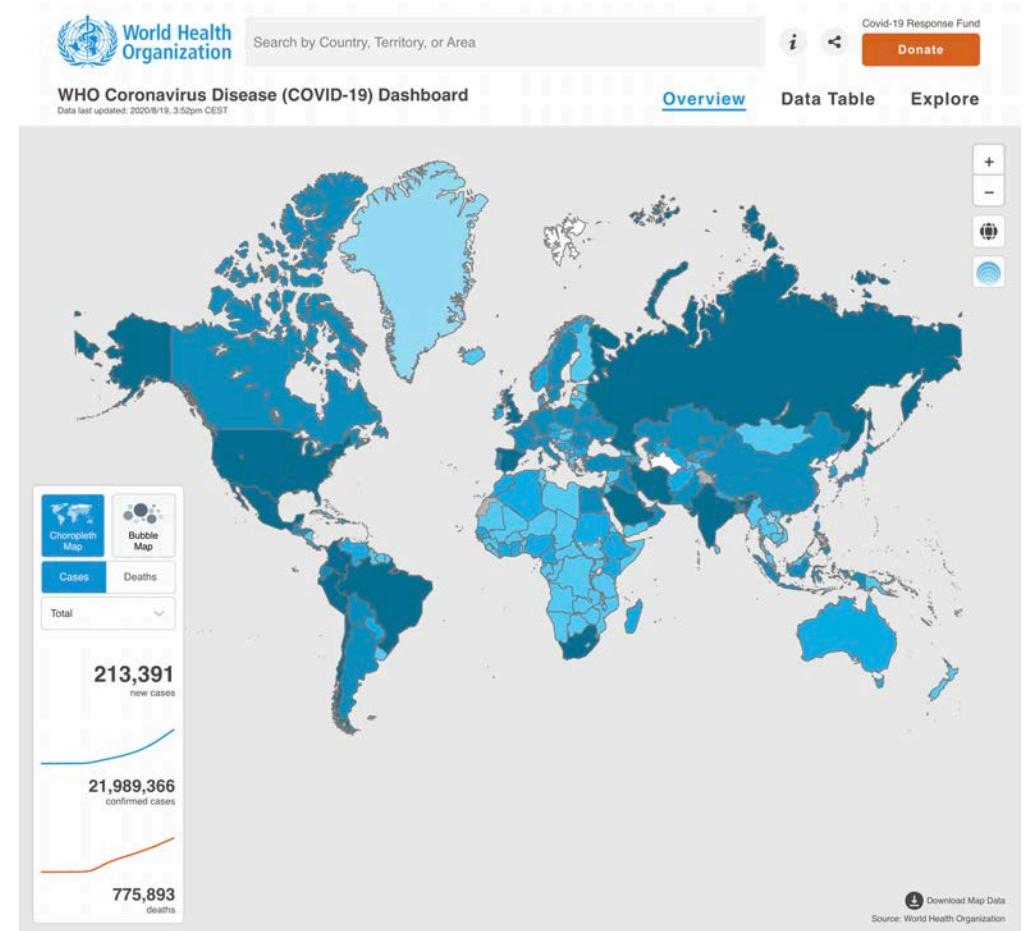


## 3.4 Ggplot – Ejercicio 9

### Ejercicio 9

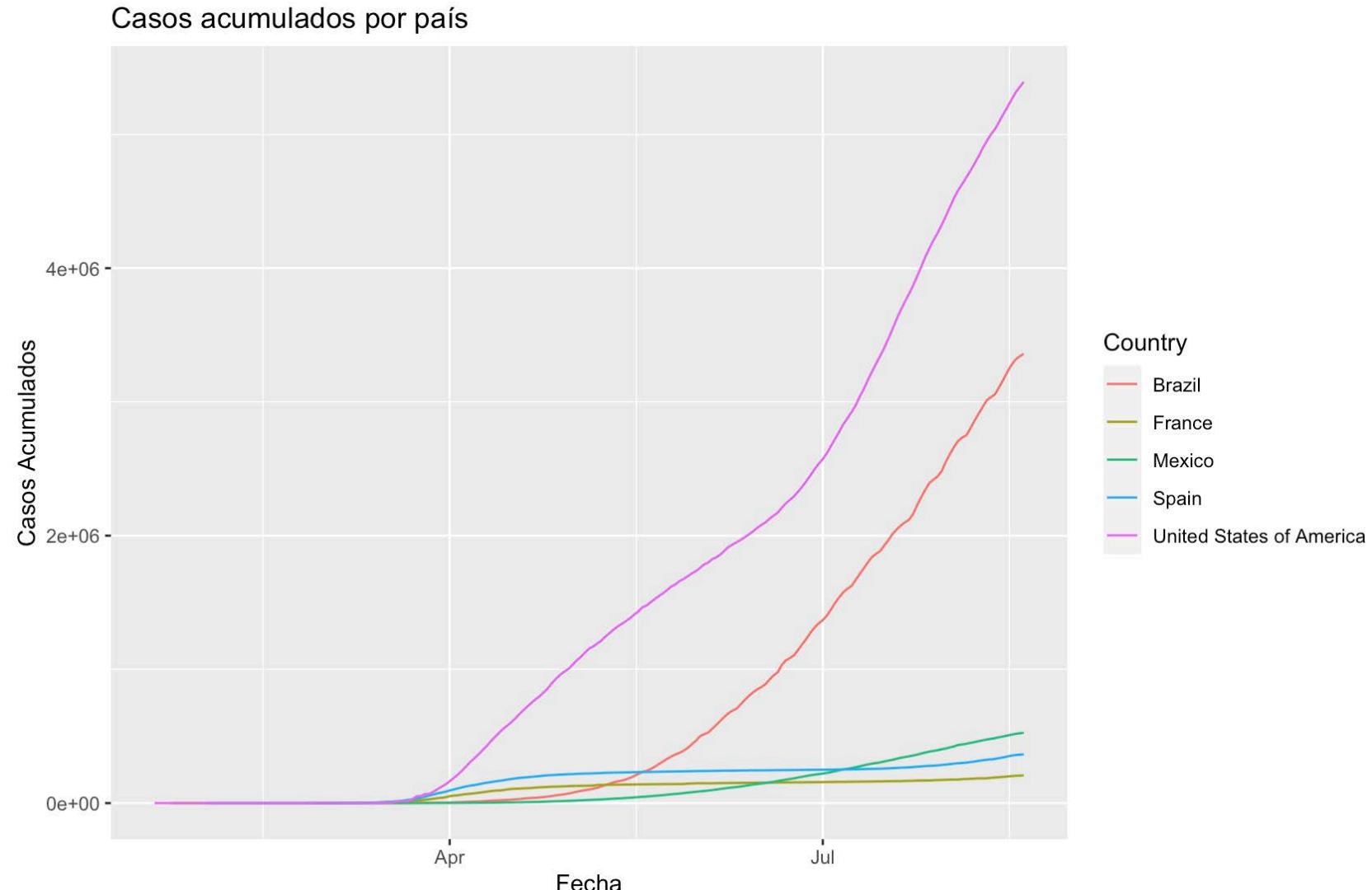
Dados los datos de la WHO <https://covid19.who.int/> sobre COVID-19 por país, se deberá generar un diagrama de línea comparando el número de casos acumulados para los países de:

- ✓ México
- ✓ España
- ✓ Brazil
- ✓ Estados Unidos
- ✓ Francia.





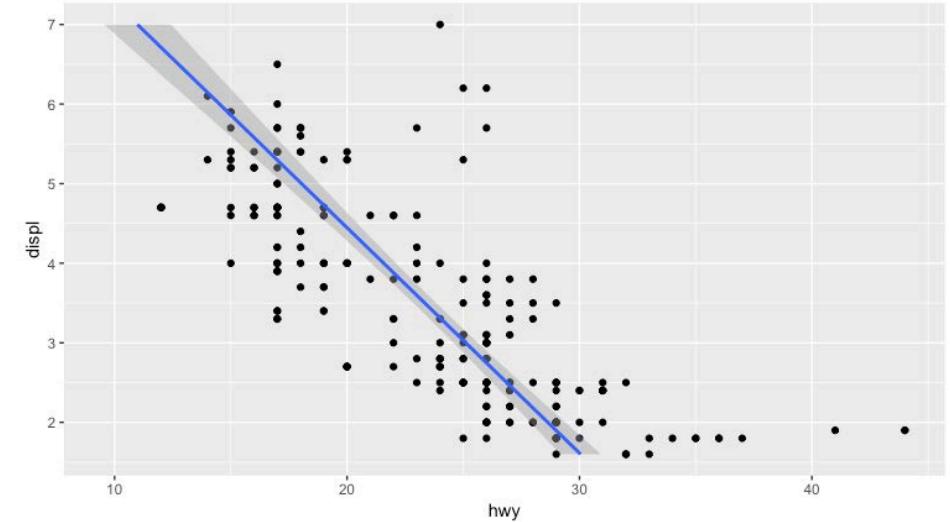
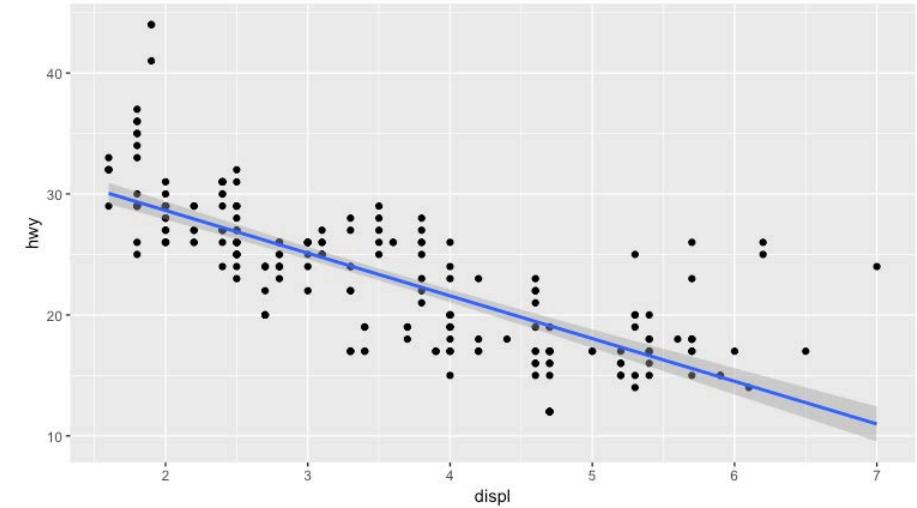
## 3.4 Ggplot – Ejercicio 9





## 3.5 Ggplot – Sistemas de Coordenadas

- Es posible invertir el Sistema de coordenadas utilizando la función **coord\_flip()**
- Esta función invierte el eje X por el eje Y y visceversa



# 3.6 Ggplot – Más Geometrías

- Una geometría en ggplot hace referencia a una forma de representar los datos en una gráfica. Ggplot define más de 40 geometrías. Además, existen algunas extensiones extras.

## ✓ Barcharts

- geom\_bar

## ✓ Boxplots

- geom\_boxplot

## ✓ Abline

- geom\_abline

## ✓ Smooth

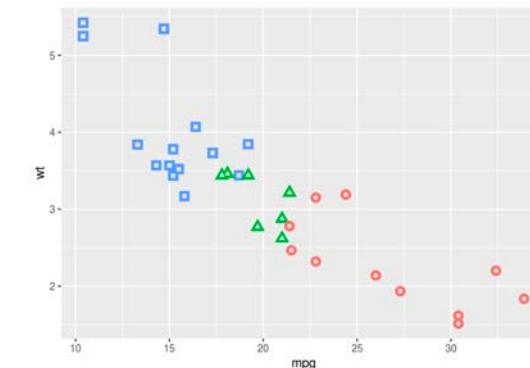
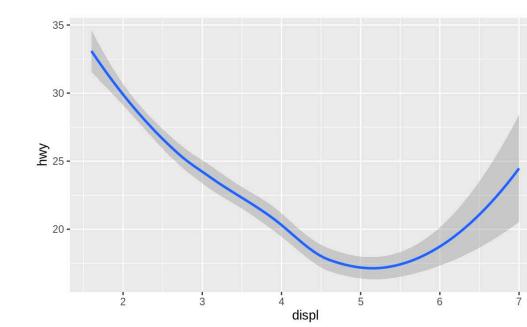
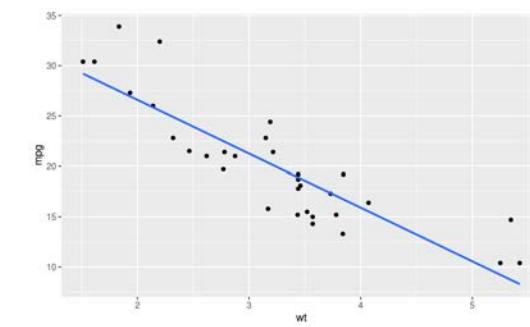
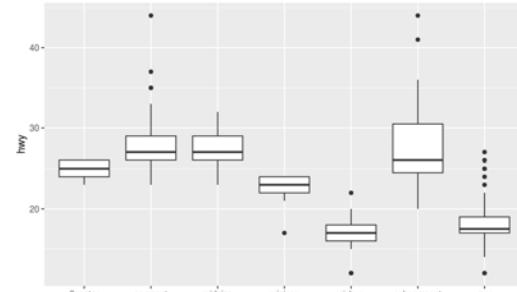
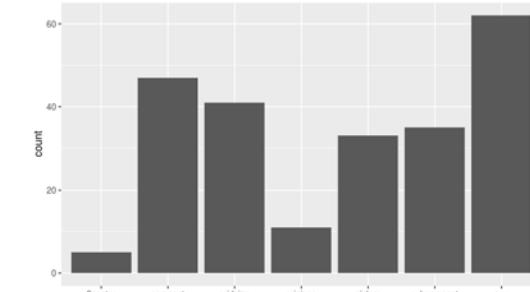
- geom\_smooth

## ✓ Puntos

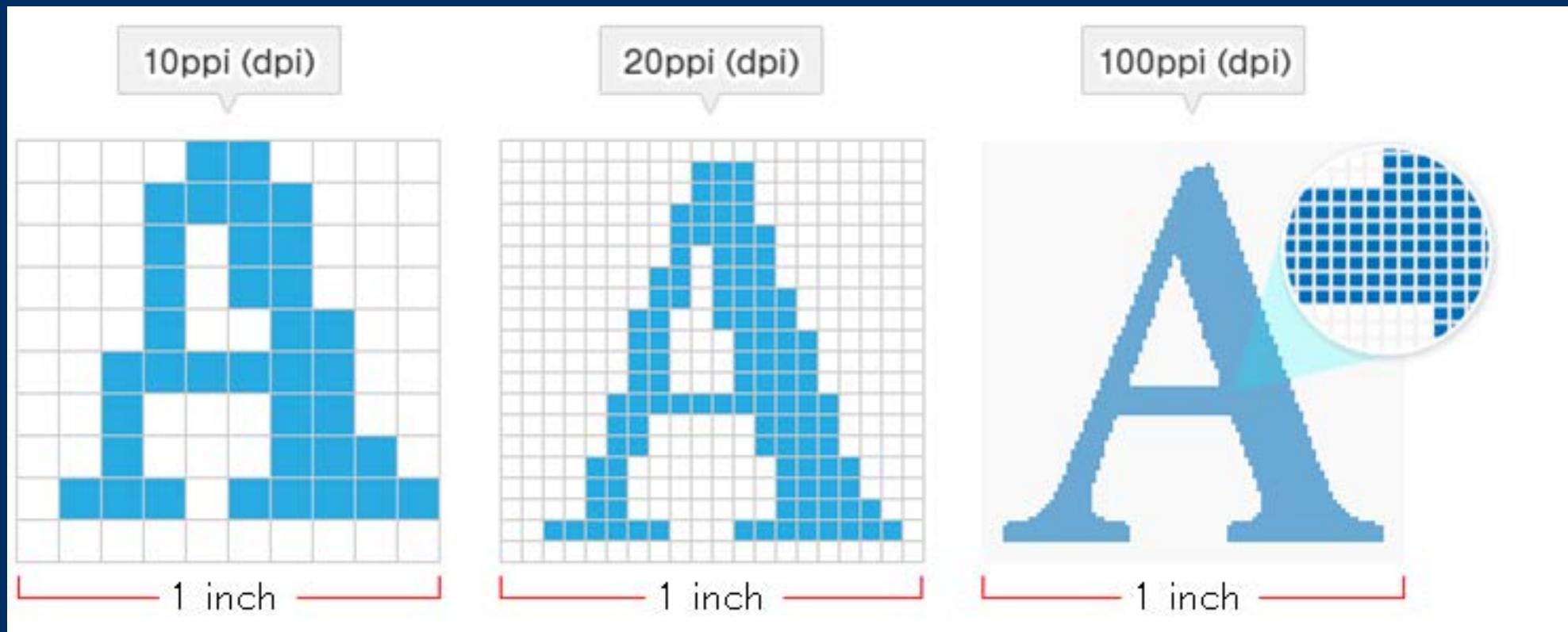
- geom\_point

## ✓ Mapas

- geom\_map



## 4. Exportando un gráfico con calidad para publicación

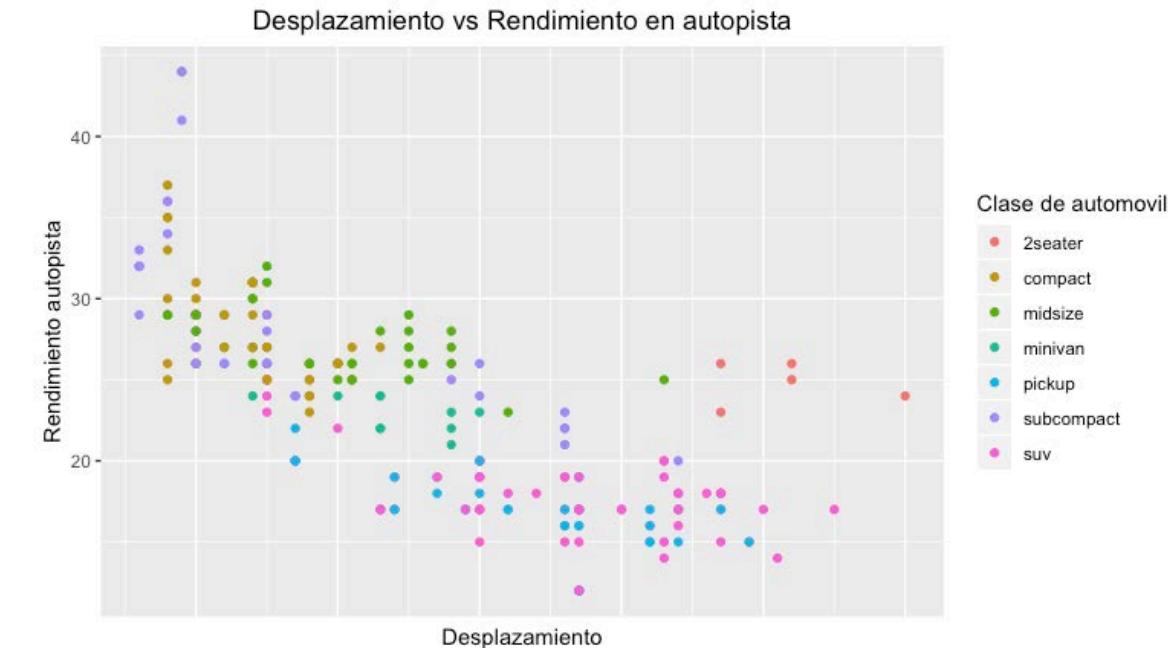


## 4.1 Exportando una gráfica

Finalmente, con `ggplot` es posible exportar **imágenes de alta calidad**. Para esto es necesario definir elementos tales como:

- ✓ Título de gráfico
- ✓ Títulos de los ejes
- ✓ Título de la leyenda

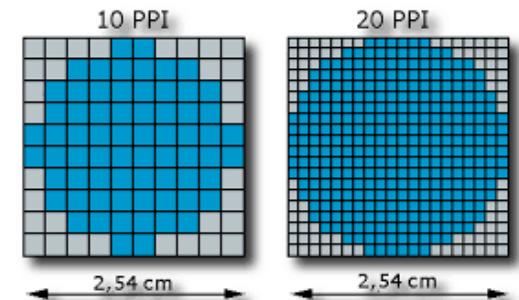
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color=class))+  
  labs(  
    x = "Desplazamiento",           # título del eje x  
    y = "Rendimiento autopista",   # título del eje y  
    title = "Desplazamiento vs Rendimiento en autopista", # título principal de la figura  
    color = "Clase de automovil"   # título de la leyenda  
) +  
  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank(), plot.title = element_text(hjust = 0.5))
```



## 4.1 Exportando una gráfica

- Es posible almacenar una gráfica como un objeto de R, para tal fin se utiliza el operador `<-` y se define una variable (objeto).

```
plot1 <- ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color=class))+  
  labs(  
    x = "Desplazamiento",          # título del eje x  
    y = "Rendimiento autopista",   # título del eje y  
    title = "Desplazamiento vs Rendimiento en autopista",  # título principal de la figura  
    color = "Clase de automovil"    # título de la leyenda  
) +  
  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank(), plot.title = element_text(hjust = 0.5))
```



- Una vez que se ha almacenado dicho objeto, se utiliza la función **ggsave()** para escribir la gráfica en un archivo con una extensión definida. Revisar la ayuda **?ggsave()** para verificar parámetros

```
ggsave(filename = "mpg.png", plot = plot1, width = 16, height = 9, dpi = 300, units = "cm")
```

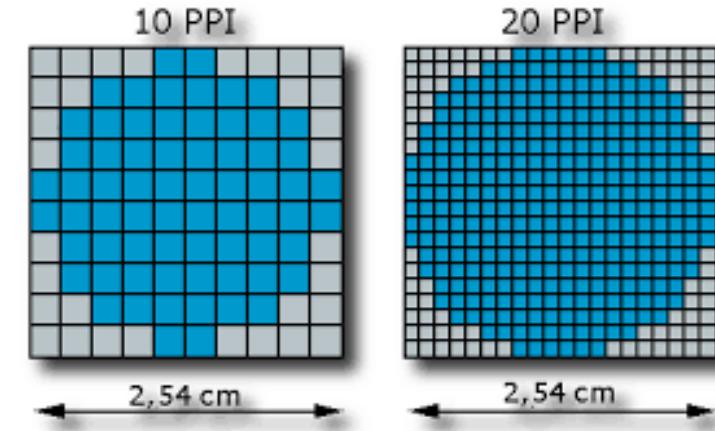


### Ejercicio 10

Se deberá exportar dos gráficas utilizando la función `ggsave` la primera deberá tener una resolución para publicación web **72 dpi**, la segunda deberá tener una resolución para publicación de **300 dpi**.

Se deberá verificar para cada figura:

1. Calidad
2. Tamaño



## 5. Ejercicio Final

```
dens <- density(data, n = npts)
dx <- dens$x
dy <- dens$y
if(add == TRUE)
  plot(0., 0,
       main
       ylab
       if(orientation == "vertical")
         dx2 <- (dx - min(dx))/max(dx)
         x[1.]
         dy2 <- (dx - min(dy))/max(dy)
         y[1.]
         seqbelow <- rep(y[1.], length(dx))
         if(Fill == T)
           confshade(dx2, seqbelow, dy2
```



# Ejercicio Final

- En equipos de 3 - 4 personas , deberán seleccionar una **categoría** y **set de datos** en la plataforma datahub

[https://datahub.io/.](https://datahub.io/)

- Deberán presentar rápidamente su tema y al menos 3 tipos de gráficos representativos sobre los datos y preparar algunas observaciones.

- ✓ **Dispersión**
- ✓ **Línea**
- ✓ **Boxplot**
- ✓ **Histograma**



## Collections

Collections - high quality data and datasets organized by topic.

 <b>Bibliographic data</b> Existing databases or services providing substantial bibliographic data. <a href="#">View Collection</a>	 <b>Climate Change</b> A collection of the most important "general" datasets on climate change. <a href="#">View Collection</a>	 <b>Demographics (population)</b> Population data and data analytics. <a href="#">View Collection</a>
 <b>Economic Data and Indicators</b> A collection of economic indicators available on DataHub. <a href="#">View Collection</a>	 <b>Education</b> US education data <a href="#">View Collection</a>	 <b>Football</b> A collection of awesome football datasets including national teams, clubs, match schedules etc. <a href="#">View Collection</a>
 <b>GeoJSON</b> GeoJSON datasets available on DataHub. <a href="#">View Collection</a>	 <b>Health Care Data</b> Ready-to-use datasets on DataHub about Health Care. <a href="#">View Collection</a>	 <b>Inflation</b> Datasets re Inflation <a href="#">View Collection</a>
 <b>Linked Open Data</b> An overview of the Linked Open Data datasets. <a href="#">View Collection</a>	 <b>Logistics</b> Ready-to-use logistics datasets - explore, download and use in your tool! <a href="#">View Collection</a>	 <b>Machine Learning / Statistical</b> Examples of machine learning datasets. <a href="#">View Collection</a>
 <b>Movies and TV</b> Various resources for Movies and TV data <a href="#">View Collection</a>	 <b>Open Corporates</b> Open Database of corporate entities. <a href="#">View Collection</a>	 <b>Property Prices</b> Property Prices Datasets available on DataHub. <a href="#">View Collection</a>