# Master en Big Data. Fundamentos matemáticos del análisis de datos.

## Tema 6: Regresión Lineal. BORRADOR

Fernando San Segundo

Curso 2019-20. Última actualización: 2019-09-27

1. Relación entre dos variables.
2. Regresión lineal simple.
3. Bondad del ajuste (goodness of fit).
4. Modelo de regresión lineal simple e inferencia.
5. Diagnósticos del modelo de regresión lineal simple.
6. Complementos de R.

Sección 1

## Relación entre dos variables.

# Introducción.

- Vamos a extender los métodos de inferencia que hemos aprendido al estudio de las **relaciones entre dos variables aleatorias**, relación que representamos con un símbolo que ya conocemos:

$$Y \sim X$$

donde $X$ es la **variable explicativa**, mientras que $Y$ es la **variable respuesta.**

- Dependiendo del tipo de variables $X$ e $Y$ se pueden dar cuatro situaciones:

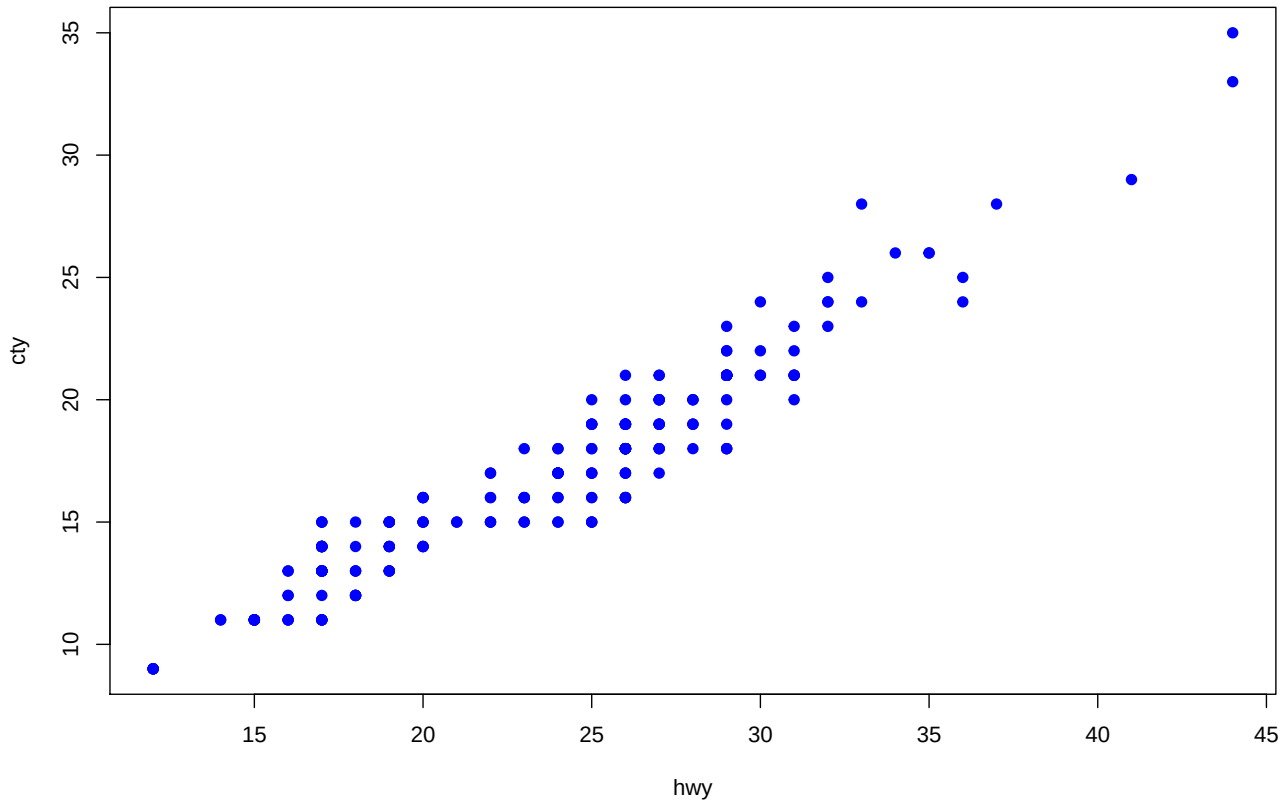
| | | Var. respuesta $Y$. | |
|---|---|---|---|
| | | **Cuantitativa (C)** | **Cualitativa (F)** |
| **Variable explicativa $X$** | **Cuantitativa (C)** | **C $\sim$ C**<br>**Regresión lineal.** | **F $\sim$ C**<br>**Regresión Logística.**<br>**o multinomial.** |
| | **Cualitativa (F)** | **C $\sim$ F**<br>**Anova.** | **F $\sim$ F**<br>**Contraste $\chi^2$.** |

Empezaremos por el caso $C \sim C$, la relación entre dos variables continuas. Pero primero vamos a hablar sobre la exploración gráfica de estos cuatro tipos de relaciones.

# Dos variables continuas.

- Recomendamos encarecidamente la lectura del Capítulo 7 de (Wickham and Grolemund 2016) (disponible online).
- Para representar gráficamente este tipo de situaciones usaremos un *diagrama de dispersión (sactterplot)*. Dibujamos pares $(x, y)$ donde $x$ es la variable explicativa e $y$ la respuesta. Con R clásico y las variables `cty` (respuesta) y `hwy` (explicativa) de `mpg` se obtiene este diagrama:

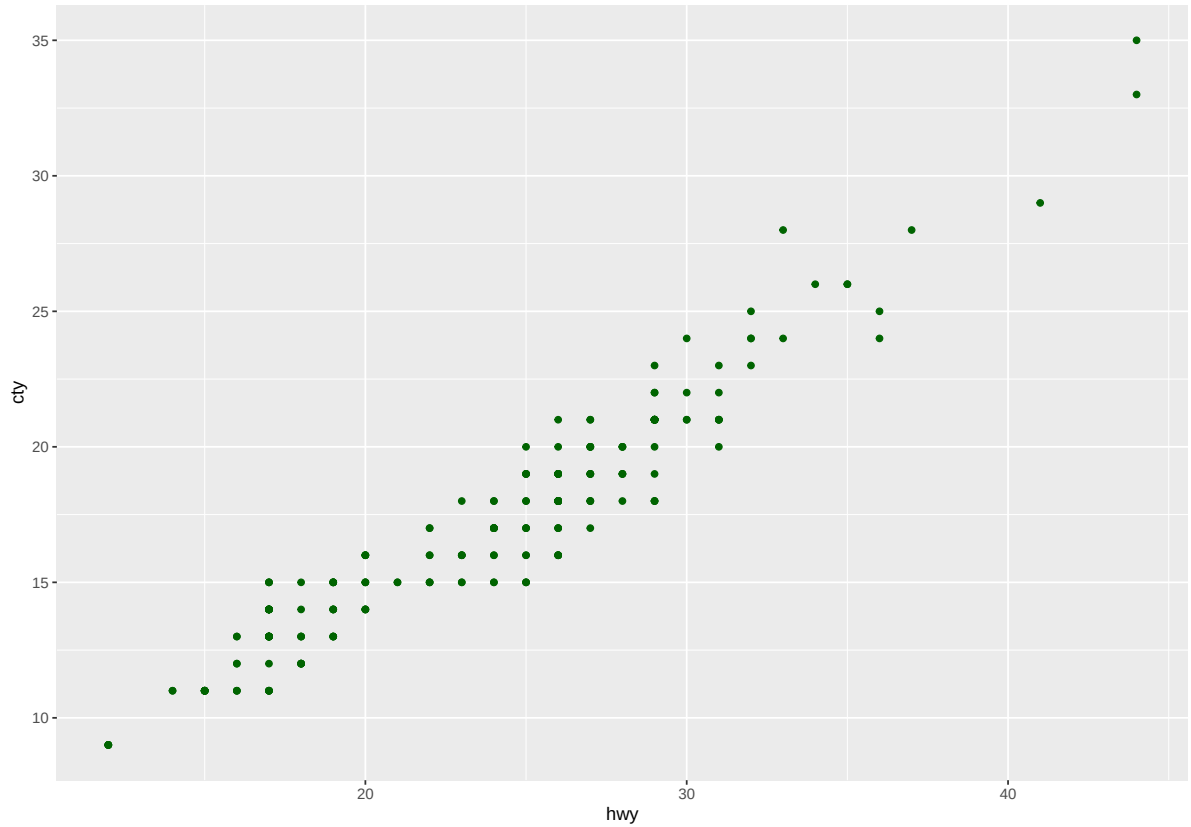
```
library(tidyverse)
plot(mpg$hwy, mpg$cty, pch = 19, col = "blue", xlab = "hwy", ylab = "cty")
```



# Diagrama de dispersión con ggplot.

- Con ggplot el código y el diagrama son:

```
library(tidyverse)
plt = ggplot(mpg) +
  geom_point(aes(hwy, cty), col = "darkgreen")
plt
```

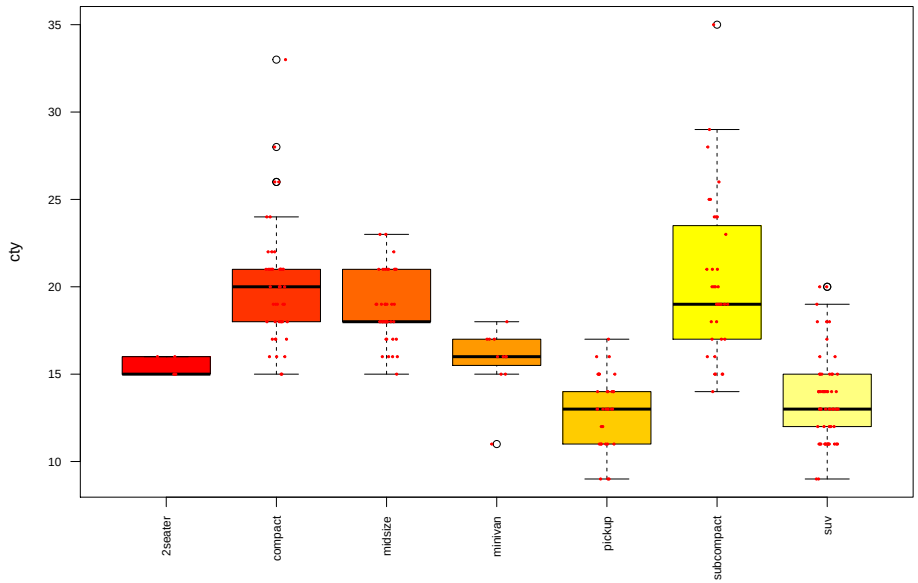



Pronto volveremos con más detalle sobre este tipo de gráficos. Hemos puesto nombre al gráfico porque lo reutilizaremos más adelante.

# Una variable continua $X$ y un factor $F$.

- Para este tipo de situaciones podemos emplear varios recursos gráficos. Puesto que la variable $X$ es continua sus valores se pueden representar mediantes boxplots, histogramas, curvas de dnesidad, etc. Para ilustrar la relación con $F$ mostramos esos diagramas para cada nivel del factor $F$.
- Por ejemplo, para ilustrar la relación entre $X =$ `cty` y el factor `class` de `mpg` dibujamos boxplots (o violinplots) paralelos por niveles y añadimos los puntos de las poblaciones.

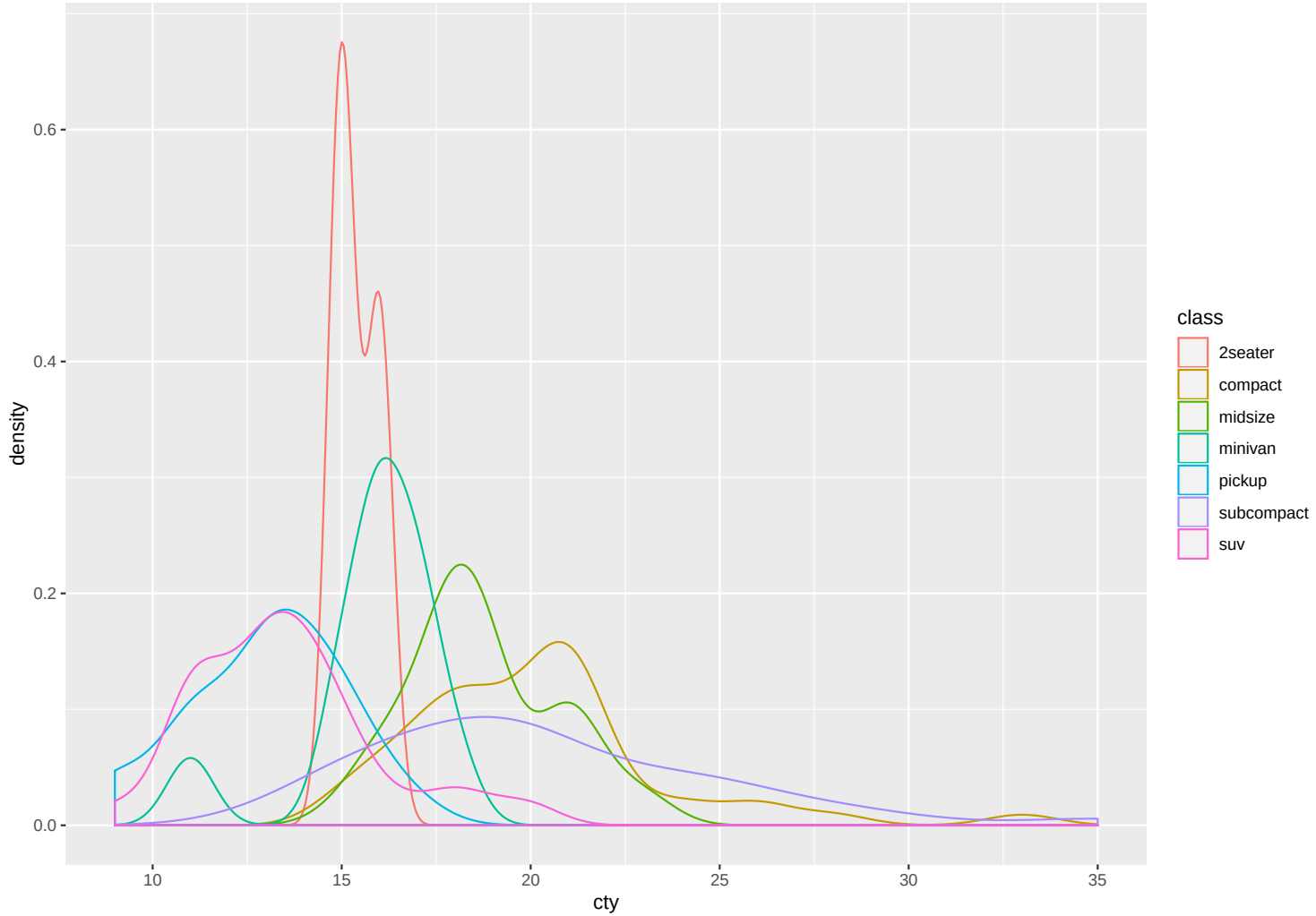
```
boxplot(cty ~ class, data = mpg, col= heat.colors(7),
        las=2, cex.axis=0.75, xlab = "")
stripchart(cty ~ class, data = mpg, method = "jitter",
           vertical = TRUE, pch = 19, col = "red", cex=0.3, add = TRUE)
```



# Otras opciones.

- Las curvas de densidad por grupos son otra opción común. Con ggplot (en R base es algo más complicado):

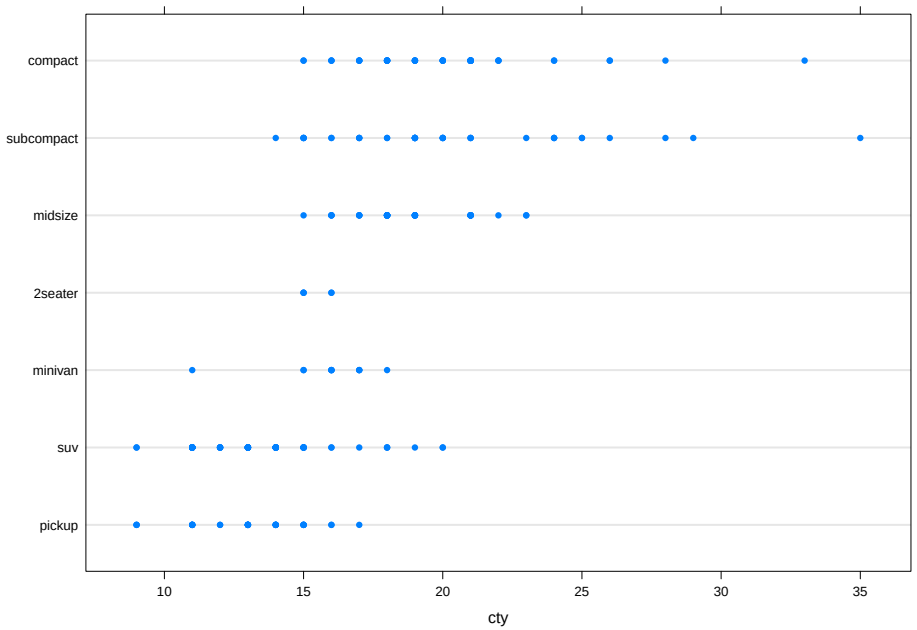
```
ggplot(mpg) +
  geom_density(aes(x = cty, color = class))
```



# Invirtiendo los papeles de $X$ y $F$

- Los dos gráficos anteriores invitan a pensar en $X$ como variable respuesta y el factor $F$ como variable explicativa, como en $X \sim F$. Pero a veces querremos cambiar los papeles. En casos así una opción es invertir el papel de los ejes y usar los mismos boxplots o bien diagramas de puntos con los valores de $X$ para cada nivel de $F$ como se ilustra aquí:

```
library(lattice)
mpg$class = reorder(mpg$class, mpg$hwy, FUN = mean)
dotplot(class ~ cty, data = mpg, lwd= 2)
```

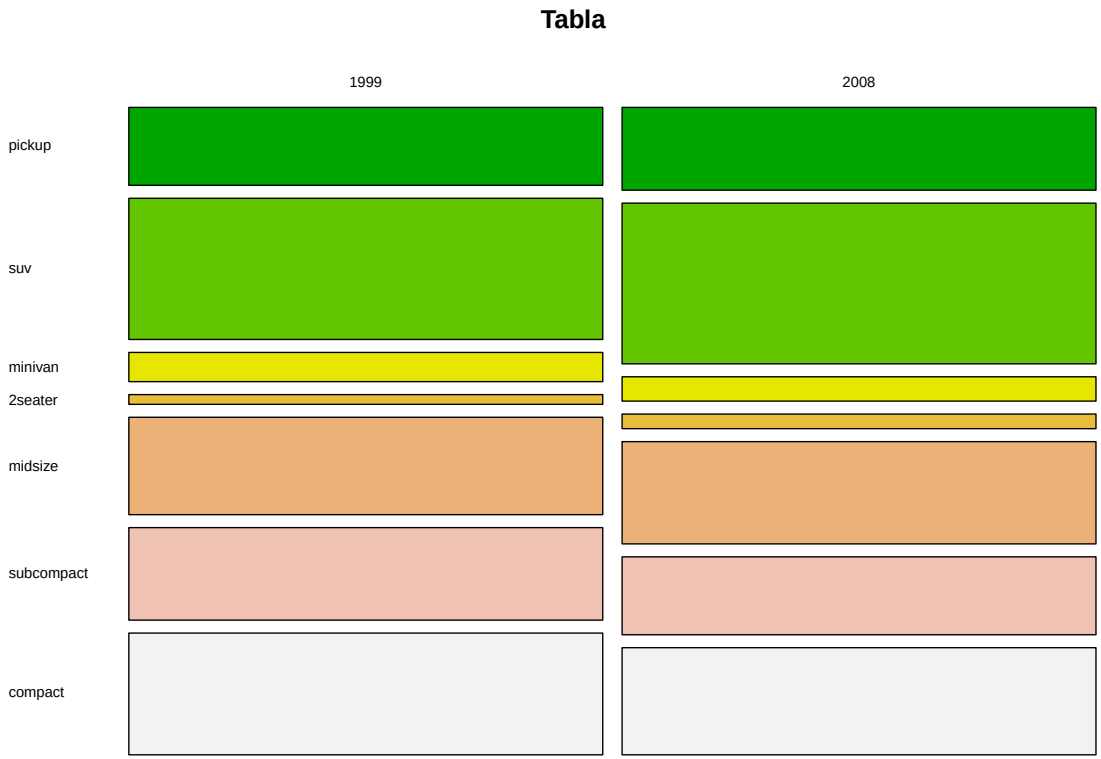



Hemos aprovechado para ordenar los niveles según el valor medio de $X$ para hacer más fácil la visualización.

# Dos factores.

- En casos con pocos niveles lo más sencillo es mostrar la información en una tabla. Pero si se desea una representación gráfica entonces se pueden usar gráficos de mosaico.

```
Tabla = table(mpg$year, mpg$class)
```

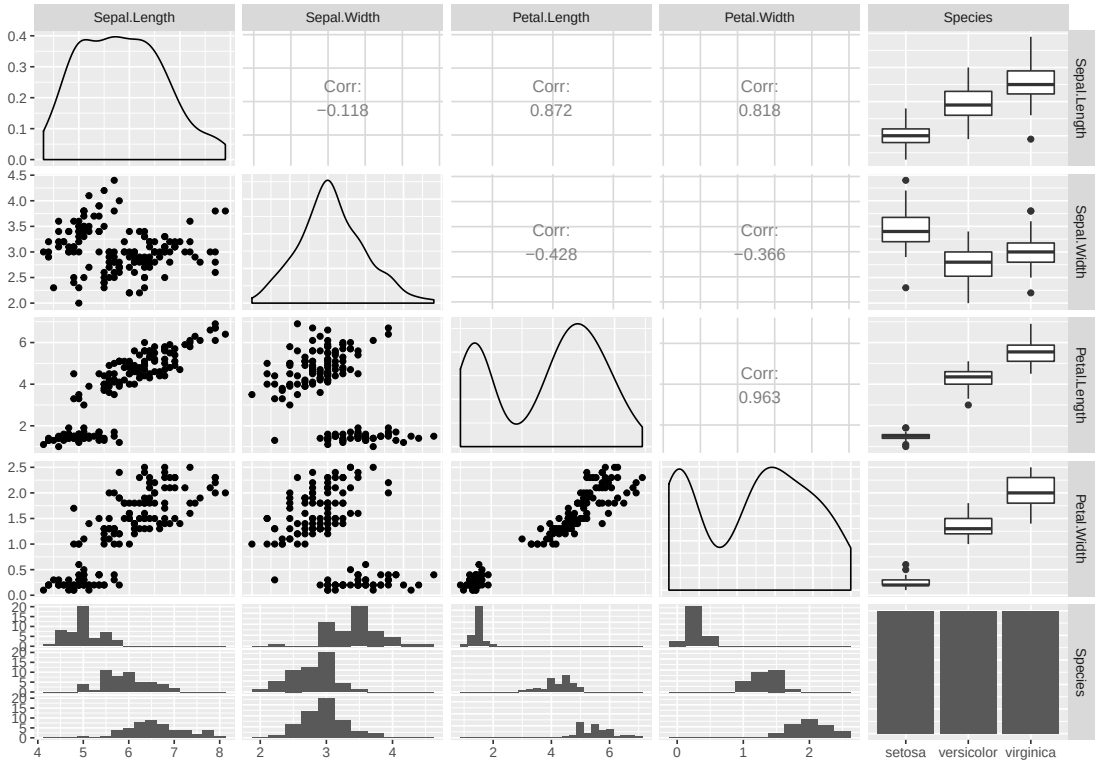



En este tipo de gráficos el *área de cada rectángulo* es proporcional al valor correspondiente en la tabla de contingencia.

# Matrices de gráficos de correlación.

- A veces para explorar las posibles relaciones entre variables de un conjunto de datos se utilizan este tipo de diagramas que comparan dos a dos las variables y disponen la información en forma de "matriz de gráficos".

```
library(GGally)
ggpairs(iris, progress = FALSE, lower = list(combo = wrap("facethist",
                                                          binwidth = 0.25)))
```




Aunque el contenido de la matriz puede ser distinto según la función que la crea, típicamente la información sobre cada par de variables se encuentra en los dos cruces de la tabla. La diagonal muestra información sobre la distribución de esa variable.

# Patrones y modelos.

- Ver la sección 7.6 de R for Data Science. Los gráficos, las tablas y las estimaciones que estamos aprendiendo a construir nos sirven para buscar *patrones* o *tendencias* en nuestros datos, que a su vez apuntan a la existencia de posibles relaciones entre las variables del problema.
- Y al explorar esos patrones, debemos tener presentes estas preguntas:
  – ¿el patrón que observamos puede ser fruto del azar?
  – ¿cómo describiríamos la relación que señala ese patrón?
  – ¿cómo de fuerte aparenta ser esa relación?
  – ¿puede haber otras variables implicadas?
  – y en particular ¿cambia la relación si se consideran subgrupos de los datos?
- Un *modelo* es una representación abstracta de las propiedades y relaciones que existen en un conjunto de variables. Al decir que una variable se distribuye como una normal ya estamos usando un modelo, De hecho, al decir que la media de una variable es $\mu$ ya estamos modelizando. Ahora queremos pensar en modelos de las *relaciones entre variables*. Vamos a empezar por uno de los modelos más sencillos, la regresión lineal simple.

Sección 2

# Regresión lineal simple.

# Ejemplo: consumo de oxígeno y temperatura en herrerillos comunes.

- En el artículo (Haftorn and Reinertsen 1985) los investigadores estudiaron la relación entre el consumo de oxígeno y la temperatura del aire en una hembra de *Herrerillo Común*, el ave que puedes ver en la Figura.

- ¿Qué crees que sucede con el consumo de oxígeno cuando sube la temperatura del aire?
- ¿Son igual de fáciles de medir ambas variables?

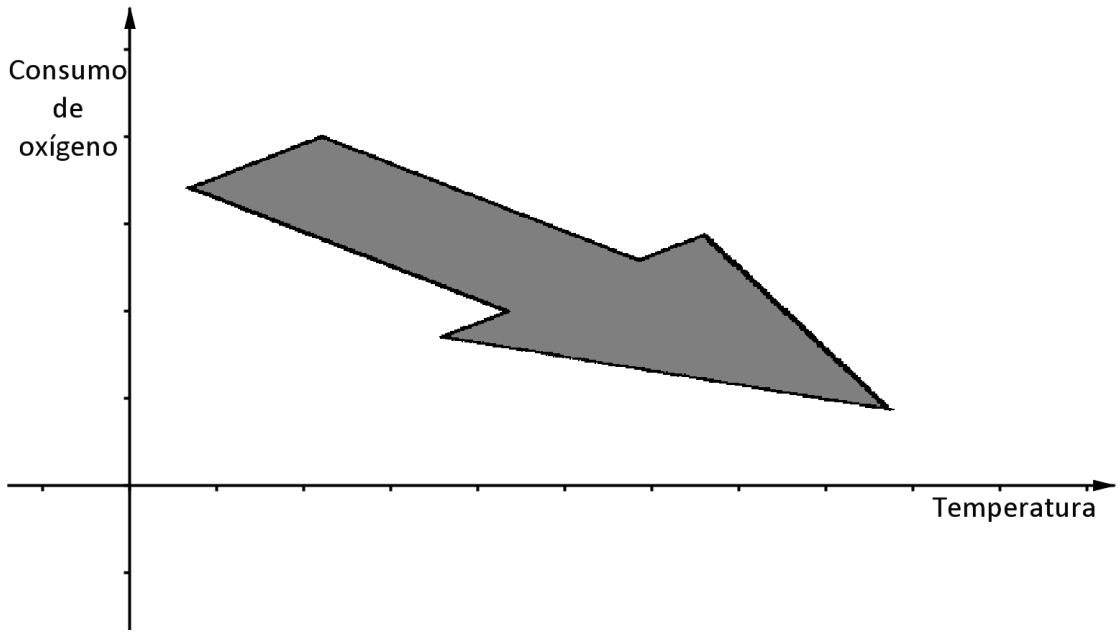
# Intuición.

- Al tratarse de dos variables continuas el resultado de las mediciones es un conjunto de **pares** de valores (por ejemplo $x$ = temp. del aire, $y$ = consumo de $O_2$)

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

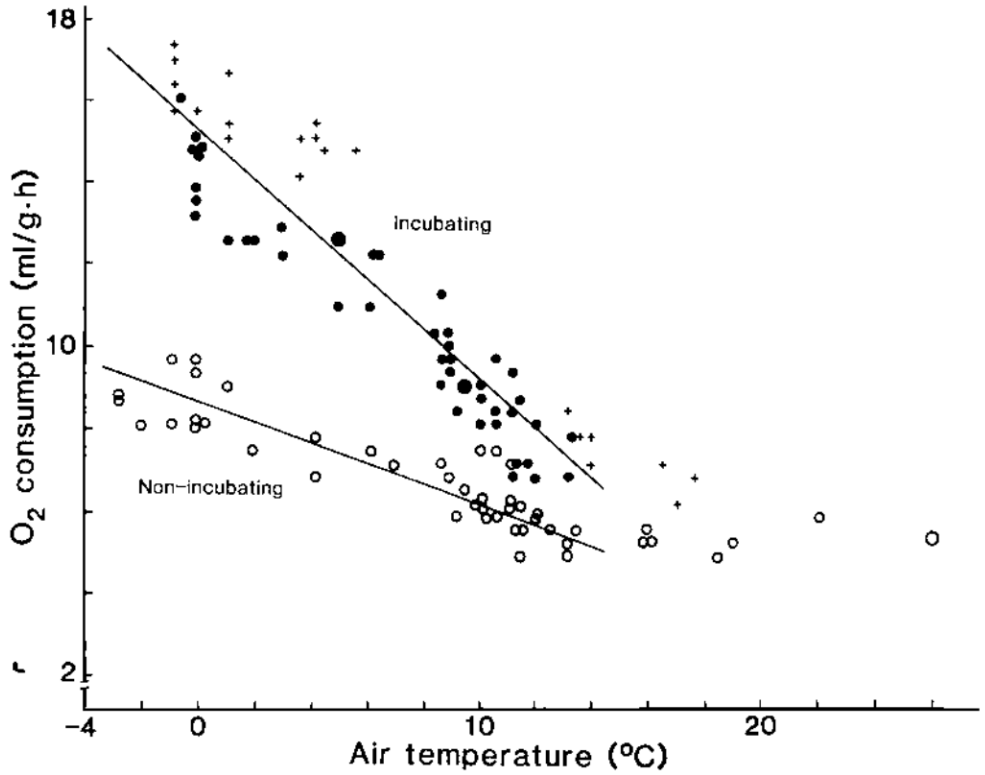
que podemos representar en unos ejes de coordenadas, con un *diagrama de dispersión*.

- En el caso de los herrerillos la conjetura natural es que si representamos esos valores la *tendencia* o *patrón* será esta:



# Patrones lineales en el diagrama de dispersión.

- En el caso de los herrerillos, los datos recogidos por los investigadores produjeron este gráfico:




que, como se ve, confirma nuestra intuición (hay dos muestras, una durante el periodo de incubación y otra fuera de ese periodo).

- Las rectas que aparecen representan el *patrón* que parecen indicar esos datos.¿Cómo podemos elegir la mejor representación, la *mejor recta*? ¿En qué sentido sería la mejor?

# Relaciones entre variables. Funciones deterministas.

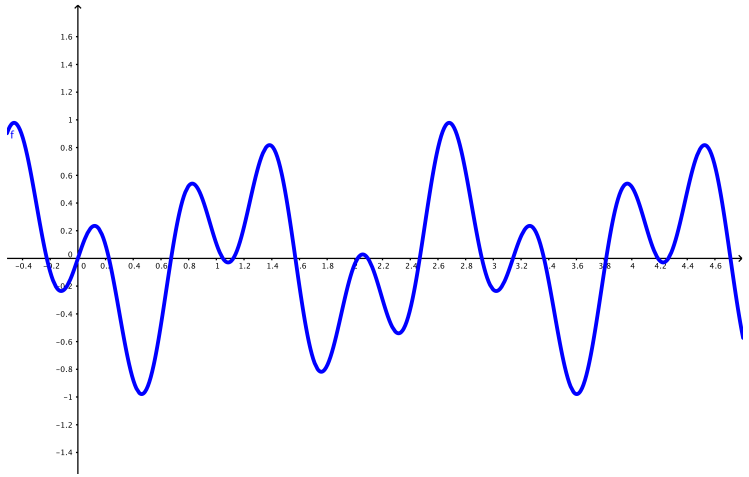
- Al estudiar Matemáticas nos hemos encontrado con la idea de **función**

$$y = f(x)$$

que describe la relación entre una **variable independiente** $x$ y una **variable dependiente** $y$, ligadas a menudo por una expresión, como por ejemplo:

$$y = \sin(3x)\cos(7x)$$

que produce una gráfica como esta:



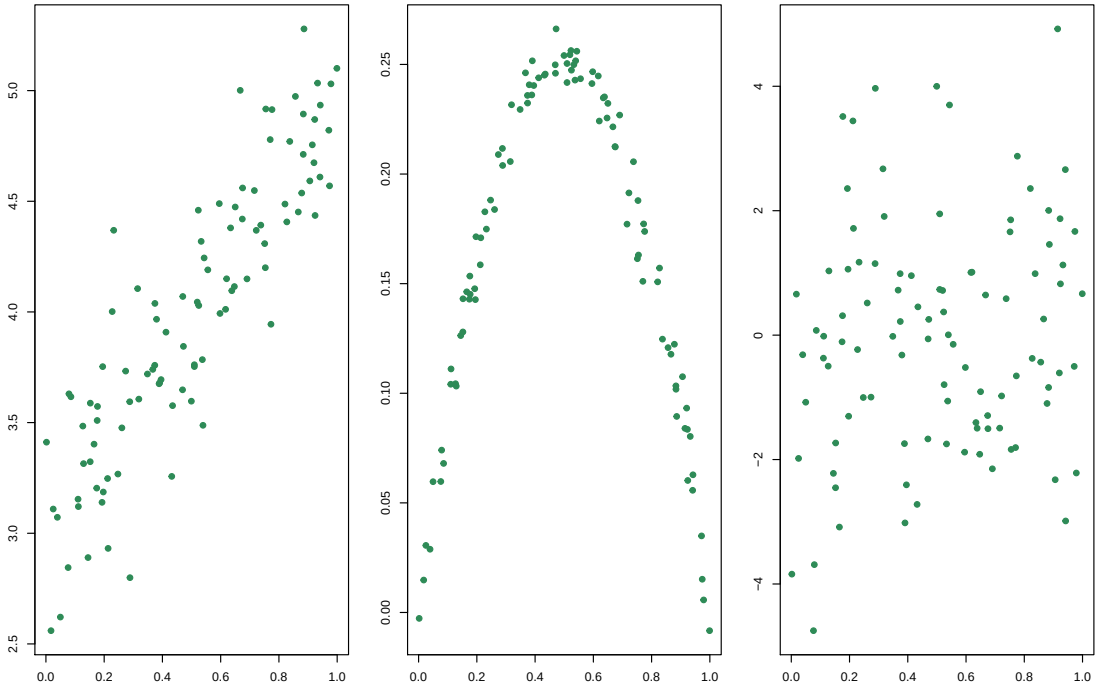
- Este tipo de relaciones pueden ser muy complicadas, pero son **deterministas**: dado el valor de $x$, calculamos el valor de $y$ obteniendo *siempre el mismo (único) resultado*. Se usan para describir relaciones teóricas entre variables, como las Leyes de la Física, o en operaciones formales como las conversiones de unidades, etc.

# Relaciones estadísticas entre variables. Notación y terminología.

- Las relaciones deterministas no bastan para describir muchas situaciones que involucran medidas, observaciones que llevan asociado algún tipo de *incertidumbre*. También hablaremos de *azar* o *ruido* para describir todos esos factores que hacen que la relación entre variables no sea determinista sino estadística. A menudo se usa también la terminología *señal y ruido*, procedente de las telecomunicaciones y popularizada por Nate Silver, para distinguir entre la relación que nos interesa (señal) y los factores aleatorios (ruido) que la enmascaran.
- En el caso de estas relaciones estadísticas a menudo seguirá siendo cierto que queremos utilizar los valores de una variable $x$ para *estimar o predecir* los valores de otra variable $y$. En este contexto diremos que $x$ es la **variable predictora (o explicativa)** mientras que $y$ es la **variable respuesta**.
- En lugar de la notación $y = f(x)$ de las relaciones deterministas, usamos $y \sim x$ para representar una de estas relaciones estadísticas. Por ejemplo, si $O_2$ es el consumo de oxígeno y $T$ la temperatura del aire, escribiremos $O_2 \sim T$. Esta ecuación indica que el valor de $T$, por si mismo, no permite calcular un único valor de $O_2$, porque existen elementos de incertidumbre (ruido) asociados con esa relación.

# Ejemplos de relaciones *ruidosas*.

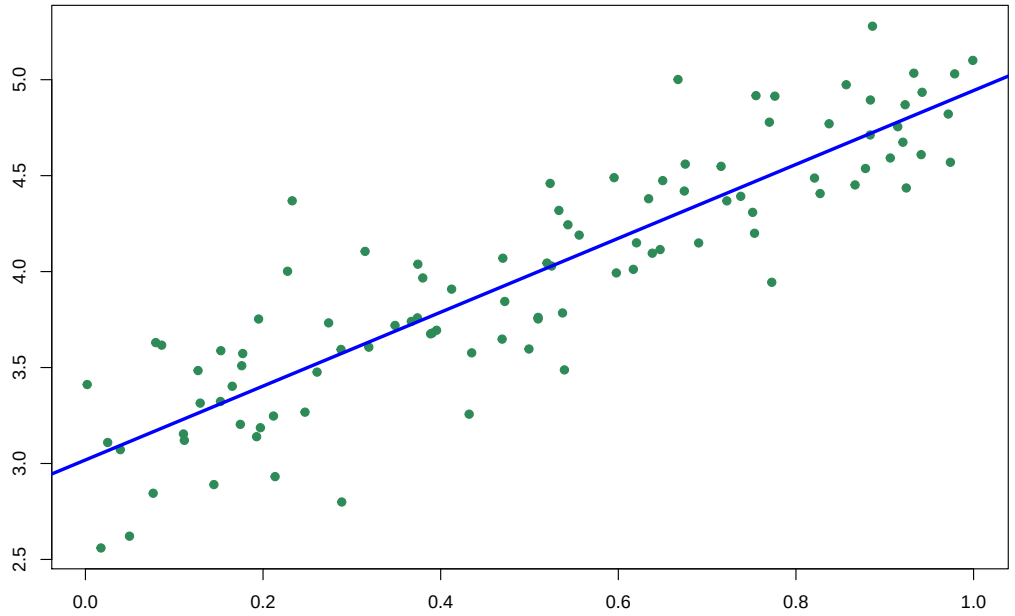
- Las tres gráficas ilustran tres ejemplos de relaciones con ruido que ilustran situaciones comunes en ese tipo de relaciones:




- La primera es una relación que se puede resumir bien mediante una recta. La segunda muestra una relación muy bien definida entre x e y (*mucha señal*, *poco ruido*), pero que no se puede resumir en una recta. La tercera no muestra relación aparente entre las variables (*poca señal*, *mucho ruido*). En este momento nos interesan especialmente situaciones como la primera.

# La recta de regresión.

- Nos centramos en el primer caso y tratamos de elegir *la mejor recta posible* para representar la relación estadística entre $x$ e $y$. Esa recta es la **recta de regresión lineal de $y$ frente a $x$**. En este ejemplo la *mejor recta* es esta:




- El plan de trabajo inmediato es este:
  ($a$) Entender en qué sentido la recta de regresión es la mejor recta posible.
  ($b$) Obtener su ecuación.
  ($c$) Entender que a veces incluso la mejor recta sigue siendo muy mala.
- Como lectura complementaria para este tema recomendamos el libro Regression Models de Brian Caffo y los vídeos que lo acompañan.

# Ecuación de la recta. Valores predichos y residuos.

- Vamos a fijar la notación que nos ayudará a avanzar. Escribimos la ecuación de la recta de regresión así:

$$y = b_0 + b_1\, x$$

donde $b_1$ es la **pendiente (slope)** de la recta, y refleja su inclinación. El signo de $b_1$ indica si la recta sube o baja. Su valor absoluto indica cuantas unidades cambia $y$ por unidad de cambio de $x$. El valor de $y$ cuando $x = 0$ es $b_0$, la **ordenada en el origen (intercept)**. A veces la recta se escribe $y = a + b\, x$ y de ahí la función `abline` de R.

- Supongamos conocidos $b_0$ y $b_1$. Dados los puntos de la muestra:

$$(x_1, y_1),\ (x_2, y_2), \ldots,\ (x_n, y_n)$$

al sustituir cada valor $x_i$ en la ecuación de la recta obtenemos *otro valor* de $y$, el **valor predicho** por la recta:

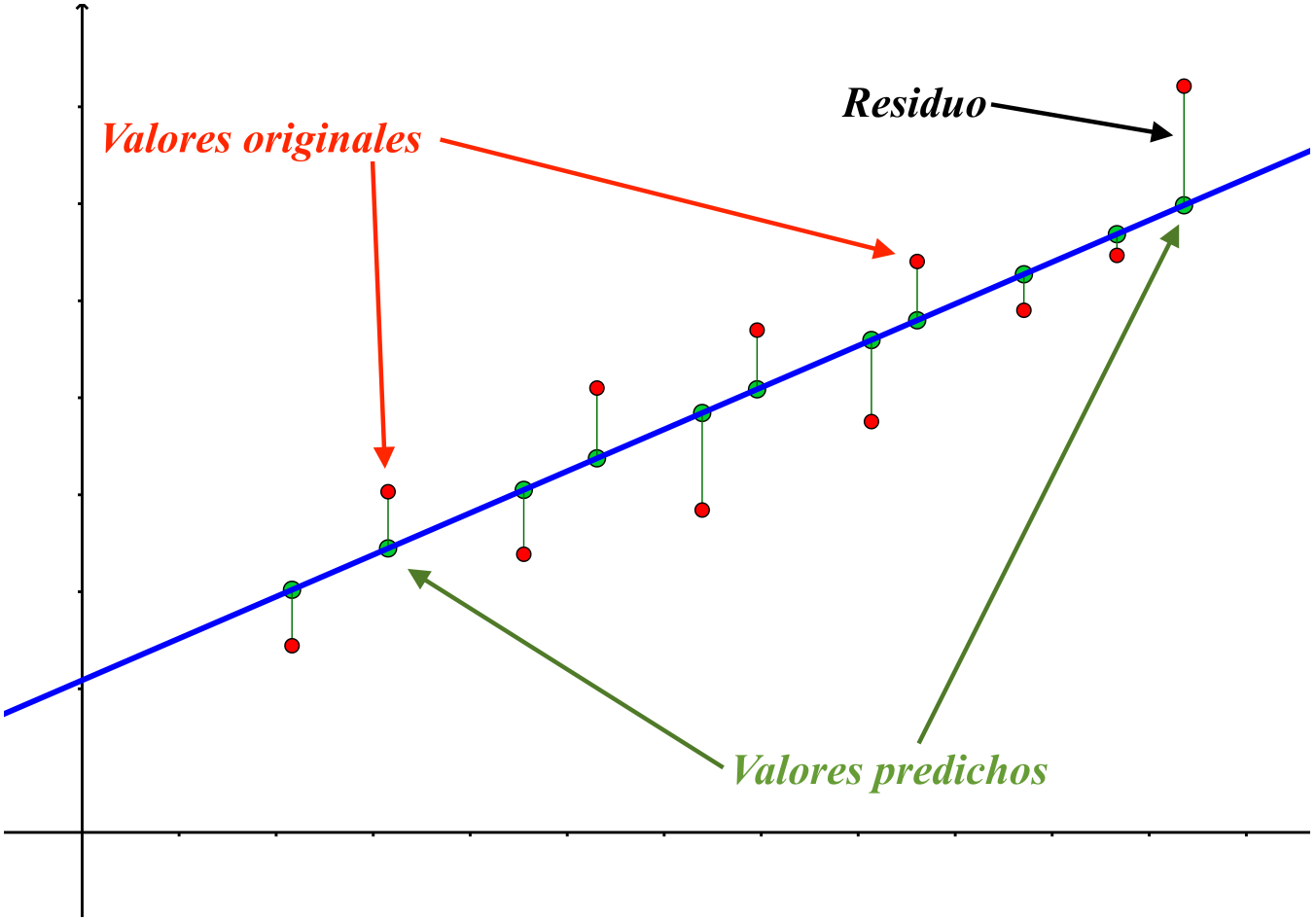
$$\hat{y}_i = b_0 + b_1\, x_i, \quad \text{para cada } i = 1, \ldots, n$$

- Los **residuos** son las diferencias:

$$e_1 = y_1 - \hat{y}_1,\ e_2 = y_2 - \hat{y}_2, \ldots,\ e_n = y_n - \hat{y}_n$$

# Representación gráfica de valores predichos y residuos.

- Los puntos rojos son los valores originales de la muestra $y_1, \ldots, y_n$, mientras que los verdes son los valores predichos $\hat{y}_1, \ldots, \hat{y}_n$. Los residuos miden la longitud de los segmentos verticales que los conectan.



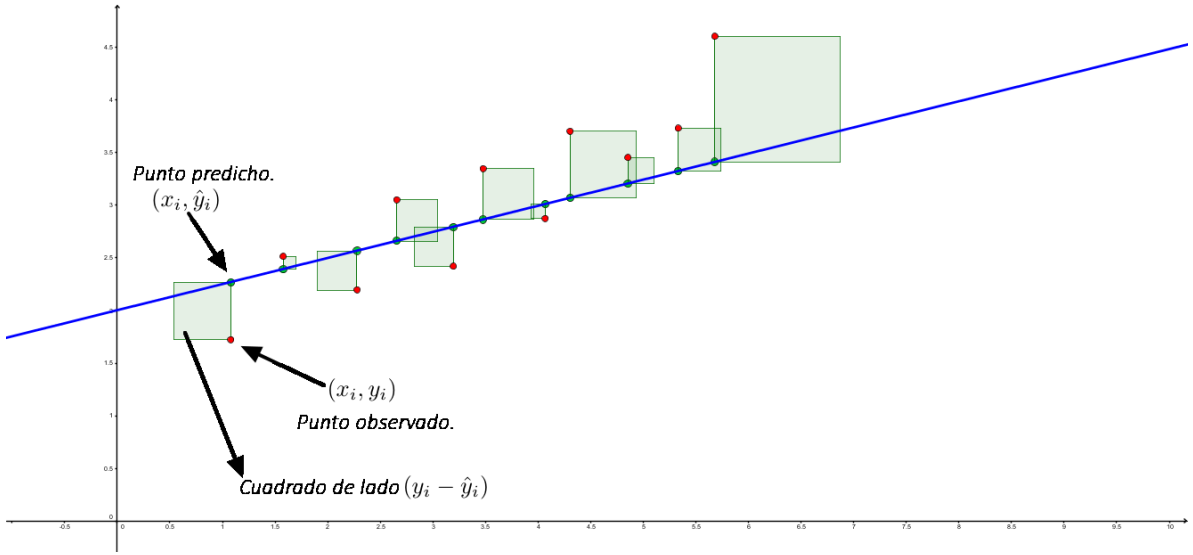
# Error cuadrático medio.

- Los residuos indican la distancia (vertical) entre la muestra y la recta. Una buena recta debería producir *residuos pequeños* en *"promedio"*.
- La primera tentación es usar la media aritmética de los residuos, pero los positivos y negativos se pueden cancelar y eso impide juzgar adecuadamente la calidad de la recta.
- El **error cuadrático (EC)** para una recta dada por $b_0$ y $b_1$ es

$$EC = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1 \cdot x_i)^2.$$

  y el error cuadrático medio muestral es $ECM = \dfrac{EC}{n-1}$.

- La siguiente figura y la construcción de este enlace ayudan a interpretar el ECM.



# Los coeficientes de la recta de regresión.

- La mejor recta es la que produce el ECM mínimo. Al resolver este problema de mínimos (usando métodos de Cálculo Diferencial) se obtiene:

**Recta de regresión.** La ecuación de la recta es

$$(y - \bar{y}) = \frac{\text{Cov}(x, y)}{s^2(x)} \cdot (x - \bar{x})$$

donde la **covarianza muestral** es:

$$\text{Cov}(x, y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n - 1}$$

Los coeficientes de la recta son $b_1 = \dfrac{\text{Cov}(x, y)}{s^2(x)}$, $\qquad b_0 = \bar{y} - \dfrac{\text{Cov}(x, y)}{s^2(x)} \cdot \bar{x}$.

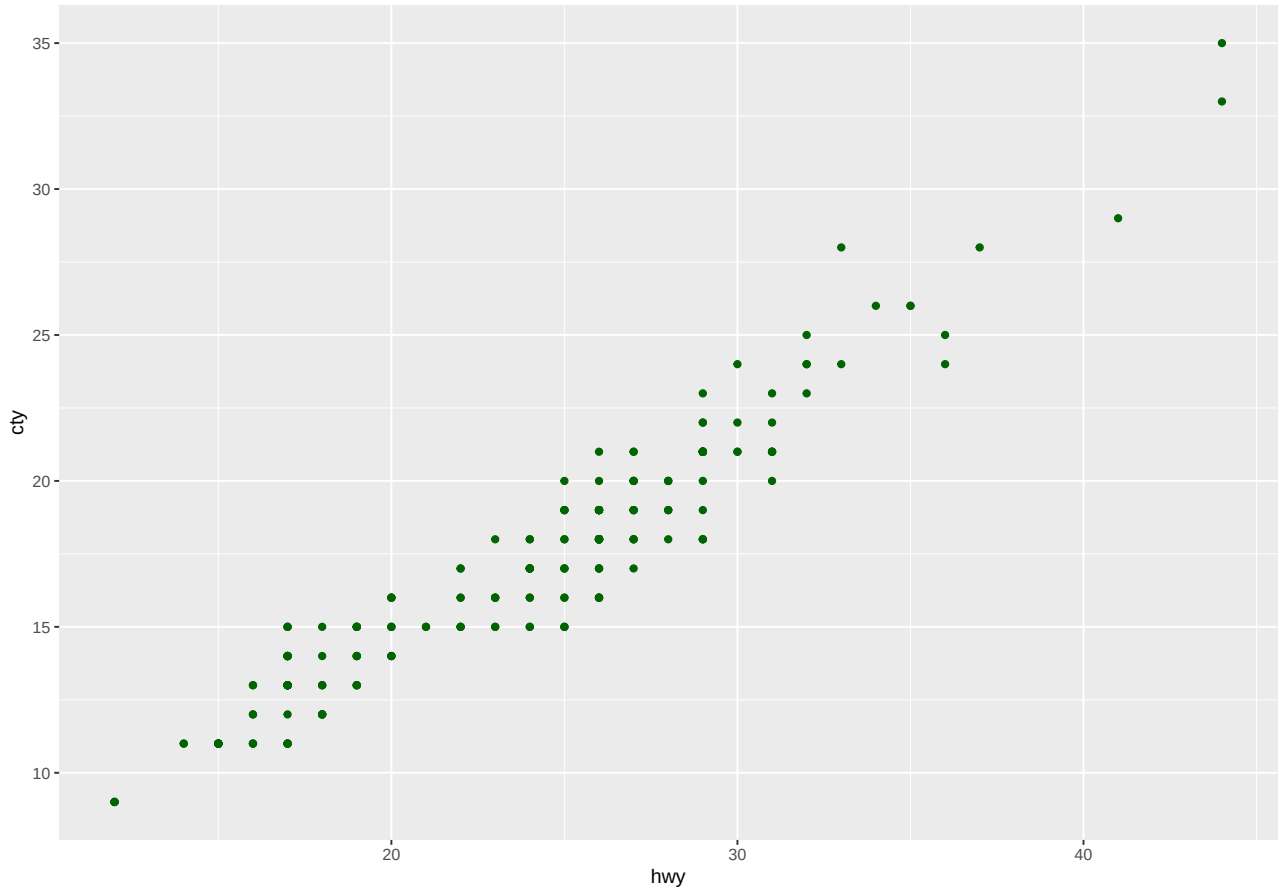
La covarianza se calcula en R con la función `cov`. Además, en el tema 4 hemos hablado de una *covarianza teórica*, pero esta es *muestral* (mira el $n - 1$).

- En particular:
  ($a$) La recta de regresión **siempre pasa por el centro de la muestra** $(\bar{x}, \bar{y})$.
  ($b$) **La suma de los residuos de la recta de regresión es siempre 0**.

# La recta de regresión con R.

- Vamos a pensar en la relación `cty ~ hwy` en los datos `mpg`. Antes vimos el diagrama de dispersión de los pares (hwy, cty). Recuerda que le pusimos de nombre `plt`, así que basta con invocarlo:

```
plt
```



# La función `lm`.

- Para obtener los coeficientes de la recta de regresión usamos `lm` (de *linear model*):

```
modelo = lm(cty ~ hwy, data = mpg)
modelo$coefficients
```

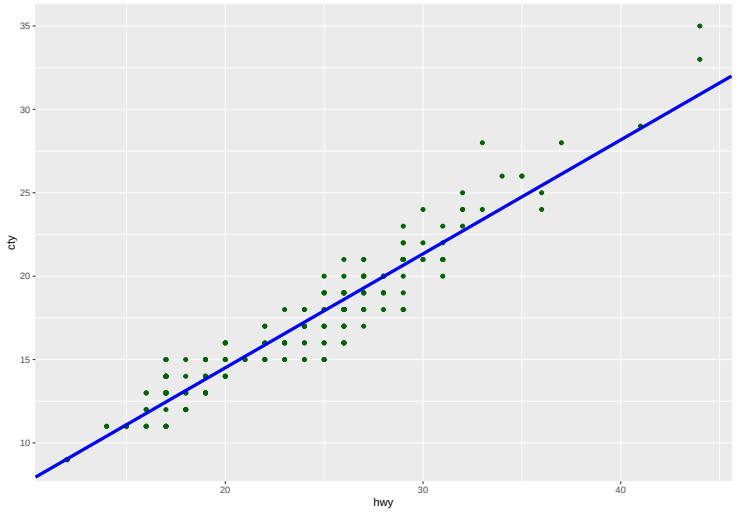
```
## (Intercept)         hwy
##   0.8442016   0.6832191
```

Para acceder a los coeficientes individuales les asignamos nombres:

```
b0 = modelo$coefficients[1]
b1 = modelo$coefficients[2]
```

Añadimos la recta al diagrama de dispersión (observa como reutilizamos 'plt):

```
plt +
  geom_abline(intercept = b0, slope = b1, color="blue", size = 1.5)
```




```
plt
```

# Predicción.

- Uno de los usos más comunes de la recta de regresión es para estimar/predecir el valor de $y$ correspondiente a un valor de $x$ determinado. Por ejemplo `hwy = 24.5` no está en la muestra. ¿Qué valor de `cty` predecimos en ese caso? Sustituyendo en la recta:

```
newHwy = 24.5
(ctyEstimado = b0 + b1 * newHwy)
```

```
## (Intercept)
##    17.58307
```

  El nombre `Intercept` se *hereda* de `b0`; se puede eliminar con `unname`):

- Cuando uses R para Análisis de Datos o Machine Learning construirás otros modelos mucho más complejos, en los que no será fácil sustituir. Para eso existe un mecanismo general con la función `predict`, en la forma:

  `prediccion = predict(`*modelo*, *datos_input*`)`

  Para la predicción anterior sería:

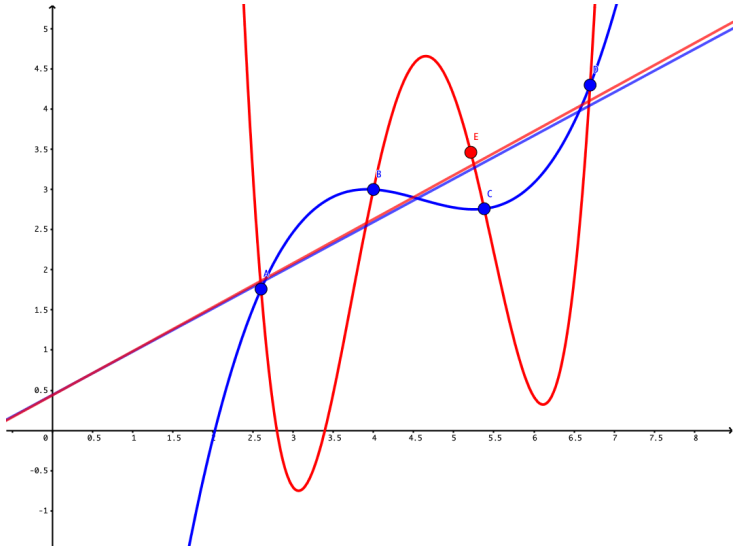
```
predict(modelo, newdata = data.frame(hwy = 24.5))
```

```
##        1
## 17.58307
```

- **Extrapolación: nunca se debe usar la recta con valores de $x$ fuera del recorrido de la muestra.**

# Sobreajuste (overfitting).

- Este puede ser un buen momento para introducir una reflexión sobre el modelo lineal ilustrada por la siguiente figura y la construcción de este enlace.




Fíjate en que las dos rectas de regresión se parecen mucho, pero que si nos empecinamos en hacer pasar una curva por todos los puntos de la muestra nuestro *modelo* se vuelve inestable y pierde sustancialmente capacidad de predecir.

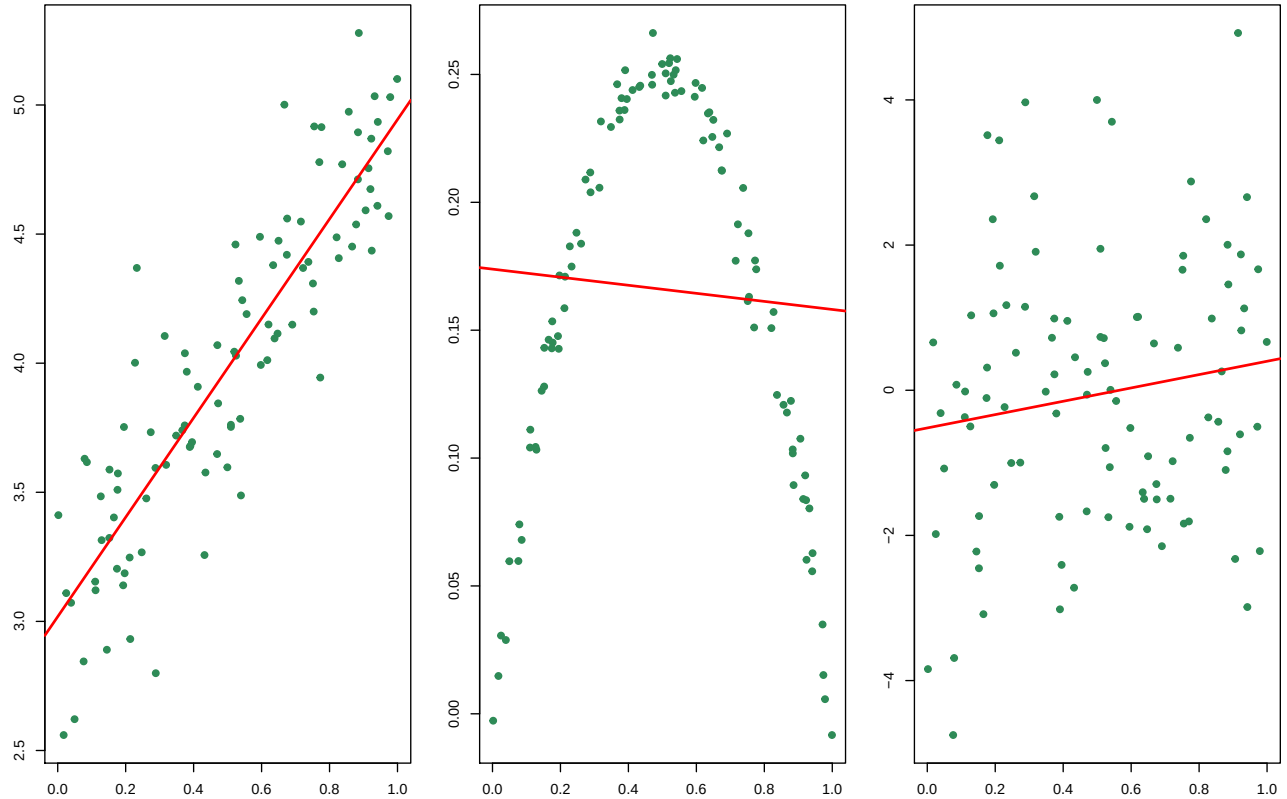
- Es muy importante entender el concepto de *señal y ruido*. Tratar de acallar a toda costa el ruido en los datos tiende a producir modelos muy desafinados. El problema que hemos encontrado aquí es del posible *sobreajuste (overfitting)* del modelo a la muestra. En Machine Learning aprenderás estrategias como la validación cruzada (cross validation) para paliar el problema.

Sección 3

# Bondad del ajuste (goodness of fit).

# La mejor recta puede ser muy mala.

- El método de mínimos cuadrados permite encontrar rectas de regresión *incluso en casos en los que es evidente que usar una recta es una mala idea*. Volviendo sobre algunos ejemplos que ya hemos visto:




En el gráfico de la izquierda la recta parece una buena representación o *modelo* de los datos. Pero en los otros dos gráficos el modelo no es adecuado, aunque por razones distintas en cada uno de ellos. ¿Ves la diferencia?

# Análisis de la varianza e identidad Anova en la regresión lineal simple.

- Recordemos que el error cuadrático EC es:

$$EC = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1 \cdot x_i)^2.$$

  El error cuadrático RC está asociado con los residuos de la recta y por tanto con la componente *ruido* en esa dualidad señal/ruido de la que hemos hablado.

- La segunda de esas expresiones recuerda al numerador de la varianza de $y$. Jugando con ese parecido se obtiene esta importantísima relación:

**Identidad Anova para la regresión lineal simple.**

$$\underbrace{\sum_{i=1}^{n} (y_i - \bar{y})^2}_{SStotal} = \underbrace{\sum_{i=1}^{n} e_i^2}_{SSresidual} + \underbrace{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}_{SSmodelo}$$

- Ya hemos dicho que el término SSresidual tiene que ver con la parte de *ruido* de los datos. En cambio el término SSmodelo se calcula usando los valores predichos por la recta (la parte *modelo* de los datos); es decir, incluso si no hubiera ruido y los puntos estuvieran perfectamente alineados seguirían teniendo cierto valor de dispersión (vertical), explicable completamente en tal caso por la presencia de la recta.

# Consecuencias de la identidad Anova.

- Dividiendo la identidad Anova $SS_{total} = SS_{residual} + SS_{modelo}$ por $SS_{total}$ obtenemos:

$$1 = \frac{SS_{residual}}{SS_{total}} + \frac{SS_{modelo}}{SS_{total}} = \frac{\text{EC}}{\sum_{i=1}^{n}(y_i - \bar{y})^2} + \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

  La división garantiza que los sumandos de la derecha son:
  ($a$) *adimensionales* y no dependen de la escala del problema.
  ($b$) Son cantidades *positivas y suman 1*. ($x$) El primer sumando se refiere a la parte *ruidosa* de los datos, mientras el segundo se refiere al *modelo* de regresión (la recta).

- En particular, parece ser que la recta será tanto mejor, cuanto más grande sea este segundo sumando y, por tanto, más pequeño sea el primero.

- Si sustituimos en $SS_{modelo}$ la ecuación $(\hat{y}_i - \bar{y}) = \frac{\text{Cov}(x,y)}{s^2(x)}(x - x_i)$ llegamos a:

$$1 = \frac{\text{EC}}{\sum_{i=1}^{n}(y_i - \bar{y})^2} + \left(\frac{\text{Cov}(x,y)}{s(x) \cdot s(y)}\right)^2$$

  El término entre paréntesis es por tanto una medida de la bondad del ajuste.

# Coeficiente de correlación.

- La definición es:

**Coeficiente de correlación $r$ (de Pearson)**

$$R = \text{Cor}(x, y) = \frac{\text{Cov}(x, y)}{s(x) \cdot s(y)}$$

Recuerda que aquí también hablamos de una *cantidad muestral*.

- En R se calcula con `cor`. Por ejemplo:

```
cor(mpg$hwy, mpg$cty)
```

```
## [1] 0.9559159
```

- Usando el coeficiente de correlación podemos rescribir algunos resultados:

**Identidad Anova y Recta de regresión con el coef. de correlación** $R$ La identidad Anova es:

$$1 = \frac{SS_{residual}}{SS_{total}} + R^2$$

y la recta de regresión es:

$$(y - \bar{y}) = \text{Cor}(x, y)\frac{s(y)}{s(x)}(x - \bar{x})$$

Observa la asimetría de esta última fórmula en $x$ e $y$.

# Propiedades e interpretación del coeficiente de correlación $R$.

- Es simétrico: Cor(X, Y) = Cor(Y, X).
- Es un número adimensional comprendido entre $-1$ y 1.
- El signo de $r$ es el mismo que el de la pendiente $b_1$ de la recta de regresión. Así, si $r > 0$ la recta es creciente y viceversa.
- Sólo vale 1 o $-1$ cuando **todos** los puntos de la muestra están situados exactamente sobre la recta de regresión (ajuste perfecto de la recta cuando los puntos están alineados).
- $R^2$ es el **coeficiente de determinación** y representa la proporción de variación total de $y$ que se explica con el modelo.
- Sean $\tilde{x}_i = \frac{x_i - \bar{x}}{s_x}$ los valores tipicados de los $x_i$ y análogamente sean $\tilde{y}_i$ los valores tipificados de los $y_i$. La recta de regresión se puede escribir:

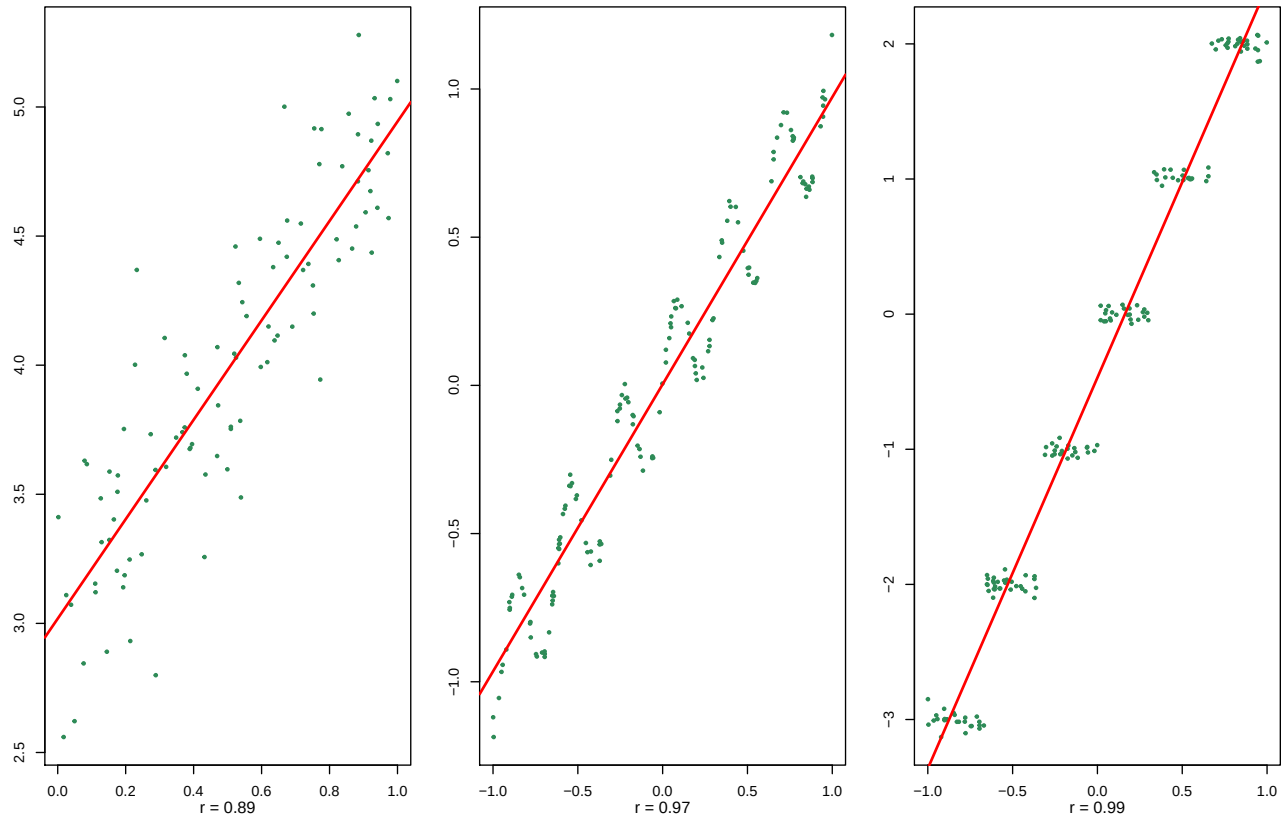
$$\tilde{y}_i = R \cdot \tilde{x}_i$$

  que puede verse como una recta de regresión para $\tilde{y}$ $\tilde{x}$. El hecho de que esta pendiente sea menor que 1 en valor absoluto es lo que explica el fenómeno de *regresión a la media*, que a su vez da nombre a todo el método.

- **Interpretación:**
  – Siempre que $r$ está cerca de 0, el ajuste de la recta a los datos es malo.
  – Siempre que el ajuste de la recta a los datos es bueno, $|r|$ está cerca de 1.
  **¡Cuidado, al revés no funciona!** Un valor de $|r|$ cercano a 1 **no garantiza** que el ajuste sea bueno. Siempre es necesario al menos examinar gráficamente el ajuste. Veamos ejemplos.

# Ejemplos de coeficientes de correlación.

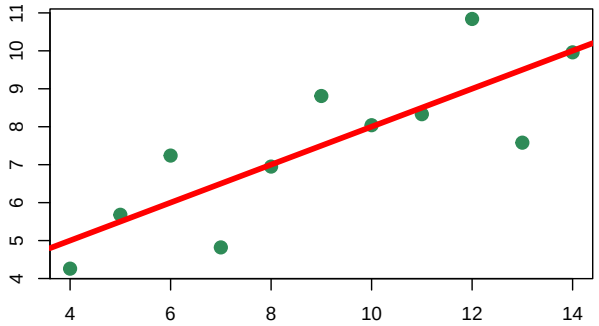
- Los tres gráficos muestran situaciones distintas con respecto al ajuste de la recta de regresión a los datos de la muestra (ver el código de esta sesión).

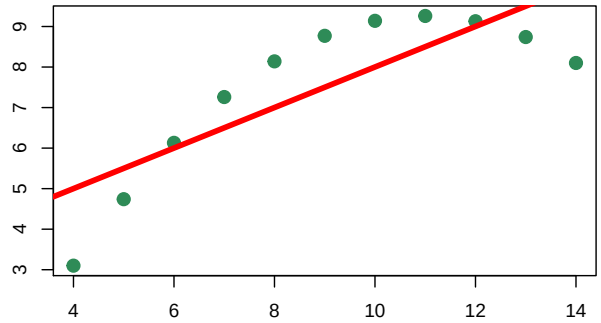


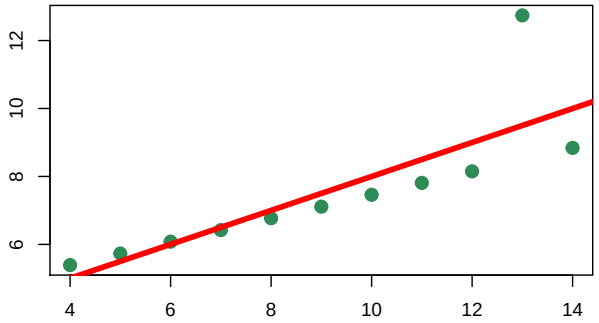

La observación más importante en este caso es que *el valor de r más bajo* de entre los tres es precisamente el que *corresponde al único modelo que es aceptable* como representación de los datos.

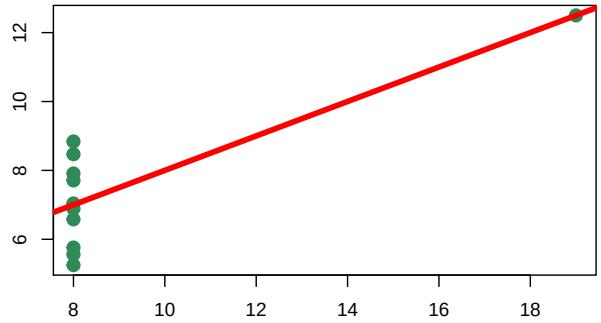
# El cuarteto de Anscombe.

- Es un ejemplo clásico disponible en el data.frame `anscombe` de R base, con cuatro muestras que ilustran el riesgo de juzgar la bondad del ajuste solo con $r$. Los cuatro diagramas de dispersión son estos.



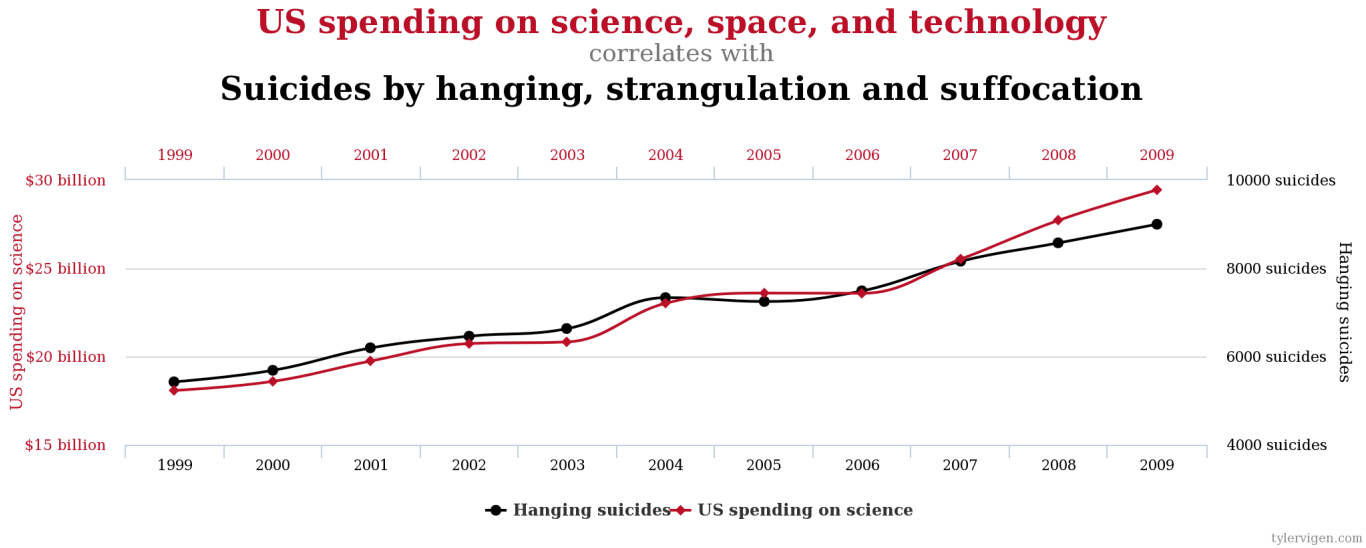




  Este ejemplo tiene la particularidad de que *las cuatro muestras* comparten los mismos valores de $b_0 = 3$, $b_1 = 0.5$ y, lo que es aun más sorprendente $r = 0.816$.

- **Nota:** el código que hemos usado es bastante más complicado que lo que hemos visto hasta ahora. Lo entenderás mejor cuando aprendas a usar `do` (de `dplyr`) y la familia `apply`.

# Correlación y causalidad.

- Otra observación importante sobre el concepto de correlación es que no debe confundirse con la idea de causalidad. En muchos casos leemos titulares que dicen cosas como *"el consumo de A vinculado con casos de B"*. Demasiado a menudo el titular se debe a que un estudio ha detectado una correlación entre A y B, sin que ello implique ni dependencia ni mucho menos causalidad (*cum hoc ergo propter hoc*).
- Sirva de ejemplo este diagrama que muestra una ¿asombrosa? correlación (procedente de la pagina de Tyler Vigen llamada spurious-correlations) entre dos series de datos:




En este caso se obtiene $r = 0.9979$ pero nadie en sus cabales sostendría que existe una relación de causa y efecto entre estas dos variables (ver también Investigación y Ciencia).

# Ejemplo

- *Con el conjunto de datos `mpg`, ¿qué porcentaje de la variabilidad total en `hwy` se explica con los valores de `cty`?*
  Empezamos construyendo un modelo lineal:

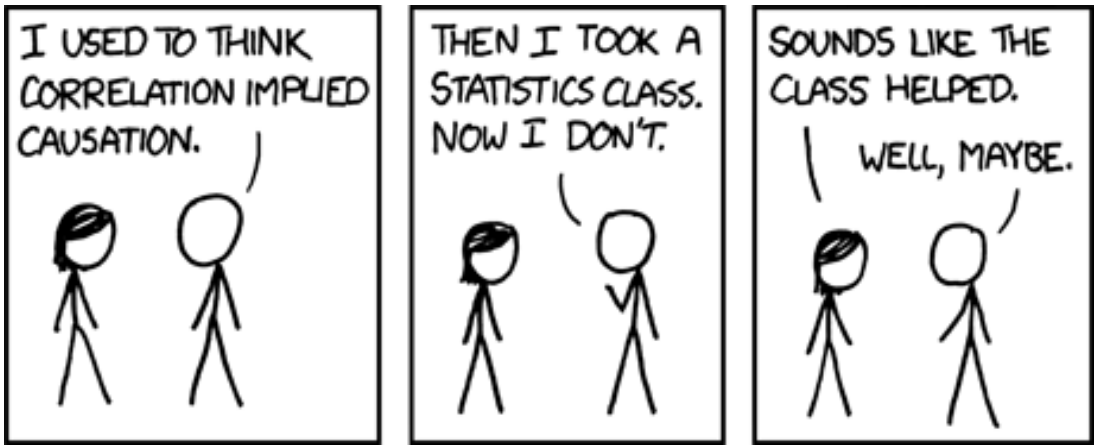
```
modelo = lm(hwy ~ cty, data = mpg)
```

  Para extraer esa información podemos usar

```
(R2 = cor(mpg$hwy, mpg$cty)^2)
```

```
## [1] 0.9137752
```

  que dice que el 91 % de la variación total en `hwy` se explica por la variación en `cty`.




XKCD

Sección 4

# Modelo de regresión lineal simple e inferencia.

# De nuevo, muestra y población. Ecuación del modelo.

- Es muy importante entender que todo lo que henos hecho en este tema hasta ahora (incluido el anásis de la bondad del ajuste) se refiere a una *muestra concreta*. Pero esto es Estadística y estamos interesados en hacer Inferencia.
- Vamos a suponer que el patrón lineal que hemos observado en la muestra es un reflejo de un *modelo lineal subyacente* en la población en la que están definidas $X$ e $Y$. Este modelo lineal es una abstracción teórica. Lo definimos as:

**Modelo de regresión lineal simple.** Viene dado por esta ecuación:

$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{modelo}} + \underbrace{\epsilon_i}_{\text{ruido}}$$

donde $\beta_0, \beta_1$ son los coeficientes del modelo, mientras que las *variables de error* $\epsilon_i$ se suponen independientes entre sí y todas con distribución normal $N(0, \sigma)$.

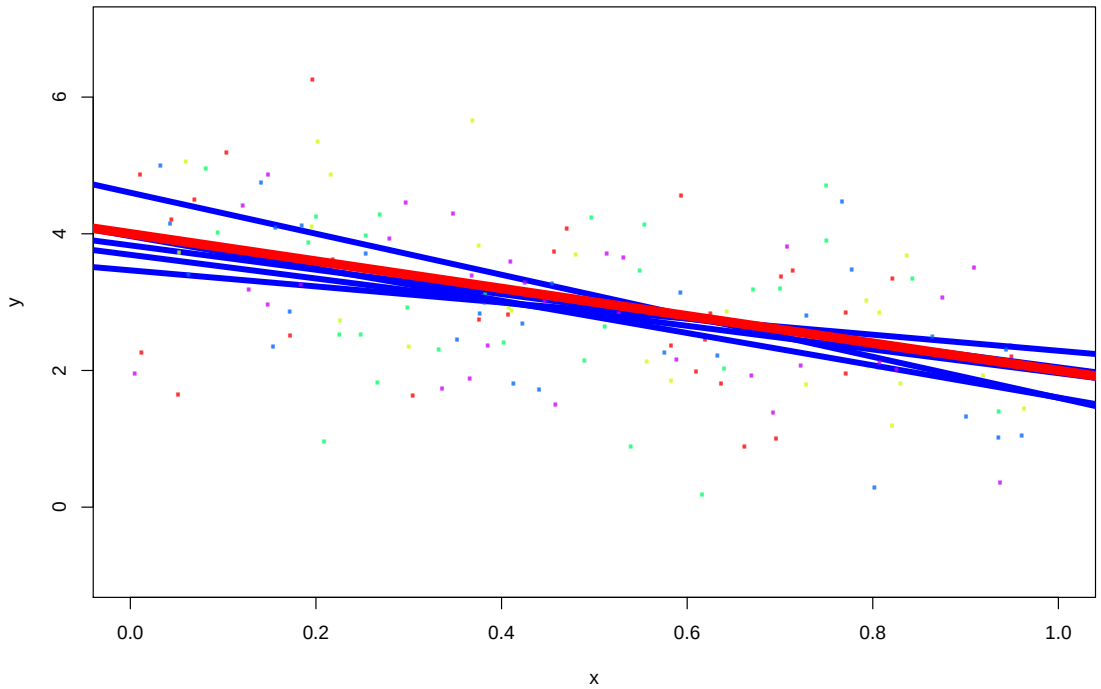
La *recta poblacional* que aparece aquí, con coeficientes $\beta_0, \beta_1$ es una *recta teórica, no observable*. Las muestras de las que venimos hablando desde el principio del tema nos permiten calcular rectas de regresión con *valores empíricos (observables)* de los coeficientes $b_0$ y $b_1$. Por supuesto, la idea es estimar

$$\beta_0 \approx b_0, \qquad \beta_1 \approx b_1$$

# Simulación de muestras, recta muestral y recta poblacional.

- Vamos a simular 5 muestras de tamaño 30 de una población en la que se tiene un modelo lineal. A partir de cada una de esas muestras calcularemos su recta de regresión como hemos aprendido a hacerlo. Puesto que es una simulación y conocemos la *recta poblacional (teórica)* compararemos esa recta (en rojo) con las que se obtienen de las muestras (en azul). En este ejemplo será $\beta_0 = 4$, $\beta_1 = -2$. Además la varianza común de los errores es $\sigma^2 = 1$.

```
set.seed(2019); colores = rainbow(5)
plot(x=c(0, 1), y=c(-1, 7), type = "n", xlab="x", ylab="y")
for(k in 1:5){
  x = runif(30)
  y = 4 - 2 * x + rnorm(30, mean = 0, sd = 1)
  points(x, y, col=alpha(colores[k], 0.8), pch="·", cex=2)
  abline(lm(y ~ x), col="blue", lwd=5)
}
abline(a = 4, b = -2, lwd=8, lty = 1, col="red")
```



# Varianza residual.

- Tenemos por tanto que ser capaces, entre otras cosas, de estimar $\beta_0$ y $\beta_1$, por ejemplo mediante intervalos de confianza calculados a partir de una muestra. Además también nos interesa el contraste de hipótesis nula $H_0 = \{\beta_1 = 0\}$, porque nos dirá si las variables están o no correlacionadas.
- Como veremos el ingrediente esencial para todo esto es la siguiente estimación de $\sigma^2$, la denominada **varianza residual**.

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum_i e_i^2$$

Observaciones:

· Usamos el símbolo $\hat{\sigma}$ en lugar de $\sigma$ para indicar que es una *estimación muestral*. Esta notación es la habitual en Estadística para estimadores.
· Dividimos por $n-2$ por la misma razón que en la varianza muestral, para tener un *estimador insesgado*. Además ese dos significa que tenemos *dos grados de libertad*, porque hay dos parámetros $\beta_0$ y $\beta_1$ en el modelo lineal.
· Si se piensa un poco sobre la ecuación del modelo y el papel de $\sigma$ es razonable que la estimación de $\sigma^2$ sea en términos de los cuadrados de los residuos (¡tienen media 0!).

# Inferencia sobre los valores de $\beta_0, \beta_1$.

- Las varianzas muestrales de los coeficientes son:

$$\sigma^2_{b_1} = \frac{\sigma^2}{\sum_{i=1}^n (X_i - \bar{X})^2}, \qquad \sigma^2_{b_0} = \left( \frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^n (X_i - \bar{X})^2} \right) \sigma^2$$

  Para usar esto en la estimación sustituiremos $\sigma^2$ por el estimador $\hat{\sigma}^2$ basado en la varianza residual que hemos visto.

- Si se cumplen las hipótesis del modelo entonces

$$\frac{b_i - \beta_i}{\sigma^2_{b_i}}$$

  (para $i = 0, 1$ y remplazando $\sigma^2$ por $\hat{\sigma}^2$) sigue una distribución $t$ de Student con $n - 2$ grados de libertad.

- A partir de estos resultados sobre distribución muestral podemos construir los intervalos de confianza y los contrastes de hipótesis necesarios. Por ejemplo, un intervalo de confianza al nivel $nc = 1 - \alpha$ para la pendiente $\beta_1$ es:

$$\beta_1 = b_1 \pm t_{n-2;\alpha/2} \frac{\hat{\sigma}^2}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

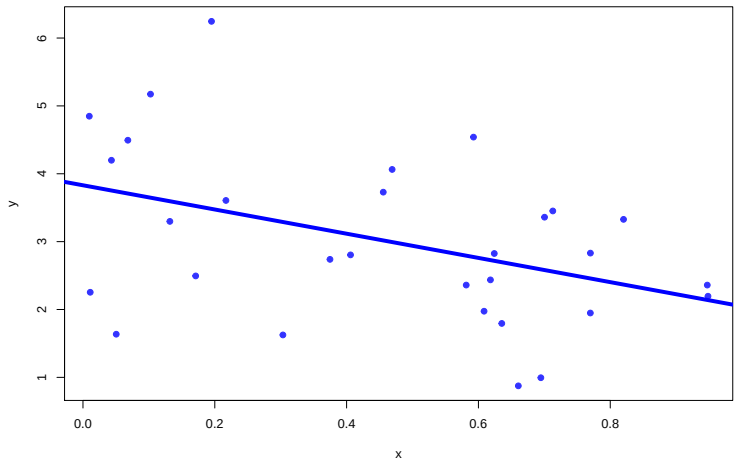
# Ejemplo extendido de cálculo con R.

- Vamos a volver sobre el modelo que hemos usado antes, con $\beta_0 = 4$, $\beta_1 = -2$. Empezamos por simular una muestra acorde con ese modelo:

```
set.seed(2019);
beta0 = 4; beta1 = -2; n = 30
x = runif(n)
y = beta0 + beta1 * x + rnorm(n, mean = 0, sd = 1)
```

Ahora vamos a usar `lm` para ajustar una recta de regresión. Y la dibujaremos en el diagrama de dispersión junto con la muestra:

```
modelo = lm(y ~ x)
plot(x, y, col=alpha("blue", 0.8), pch=19)
abline(modelo, col="blue", lwd=5)
```



# Continuación del ejemplo, 1. Estimación de la varianza residual.

- Al aplicar la función `summary` a un modelo de R se obtiene una gran cantidad de información (directa e indirectamente, como veremos).

```
(sumModelo = summary(modelo))
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10456 -0.70484  0.08237  0.78369  2.76346
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8304     0.3953   9.689 1.92e-10 ***
## x            -1.7840     0.7306  -2.442   0.0212 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.163 on 28 degrees of freedom
## Multiple R-squared:  0.1756,   Adjusted R-squared:  0.1461
## F-statistic: 5.962 on 1 and 28 DF,  p-value: 0.02119
```

Le hemos puesto nombre para poder acceder a las componentes. Por ejemplo, la estimación $\hat{\sigma}^2$ de la varianza residual (que R llama *Residual standard error*) es:

```
sumModelo$sigma
```

```
## [1] 1.162958
```

Podemos comprobarlo calculando directamente:

```
sqrt(sum(modelo$residuals^2)/(modelo$df))
```

```
## [1] 1.162958
```

# Continuación del ejemplo, 2. Intervalo de confianza para $\beta_i$

- Se pueden obtener fácilmente con

```
confint(modelo)
```

```
##                 2.5 %     97.5 %
## (Intercept)  3.020647  4.6402051
## x           -3.280511 -0.2874186
```

Vamos a comprobar a mano el de $\beta_1$:

```
tc = qt(1 - 0.025, df = n - 2) # valor crítico de la t de Student, df = n- 2
# Busca el siguiente valor en la salida de summary(lm)
(seB1 = sumModelo$sigma / sqrt(sum((x - mean(x))^2)))
```

```
## [1] 0.7305901
```

```
# Y ahora el intervalo
(intervalo = coefficients(modelo)[2] + c(-1, 1) * tc * seB1)
```

```
## [1] -3.2805105 -0.2874186
```

Dejamos el intervalo de $\beta_0$ como ejercicio.

- Si queremos contrastar $H_0 = \{\beta_1 = 0\}$ podemos ver el estadístico y el p-valor de ese contraste en las dos últimas columnas de la segunda fila de esta tabla:

```
sumModelo$coefficients
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)  3.830426  0.3953213  9.689401 1.922531e-10
## x           -1.783965  0.7305901 -2.441813 2.118716e-02
```

Y en el código de la sesión puedes ver como calcular estos valores a mano.

# Intervalos de confianza y predicción para valores de $Y$.

- Ya hemos visto antes como usar la función `predict` para lo que, en esencia, es simplemente sustituir valores de $x$ en la recta de regresión. Pero ahora hemos aprendido que esa recta de regresión es ella misma una estimación de la recta poblacional. Así que se plantean dos preguntas nuevas en relación con la predicción:

  · Por un lado, puede interesarnos calcular un *intervalo de confianza para la media de los valores de* $Y$, para un $x_0$ dado. La estimación de esa media es la que obtuvimos con `predict`, pero si la pendiente puede variar la media también, dentro de cierto intervalo.

  · Por otro lado, podemos obtener un *intervalo de predicción para los valores de* $Y$, igualmente para $x_0$.¿Qué valores mínimo y máximo de $Y$ esperamos encontrar para $x_0$ si además de la media tenemos en cuenta el ruido? Debería estar claro que este intervalo es más ancho que el anterior.

- En R es igualmente fácil usar `predict` para estos dos intervalos. En el último ejemplo el punto $x_0 = 1/2$ no está en la muestra (pero sí en su rango, *no extrapolamos*). Vamos a construir los correspondientes intervalos de confianza y predicción para ese $x_0$.

```
nuevoX = data.frame(x = 1/2)
predict(modelo, newdata = nuevoX, interval = "confidence")
```

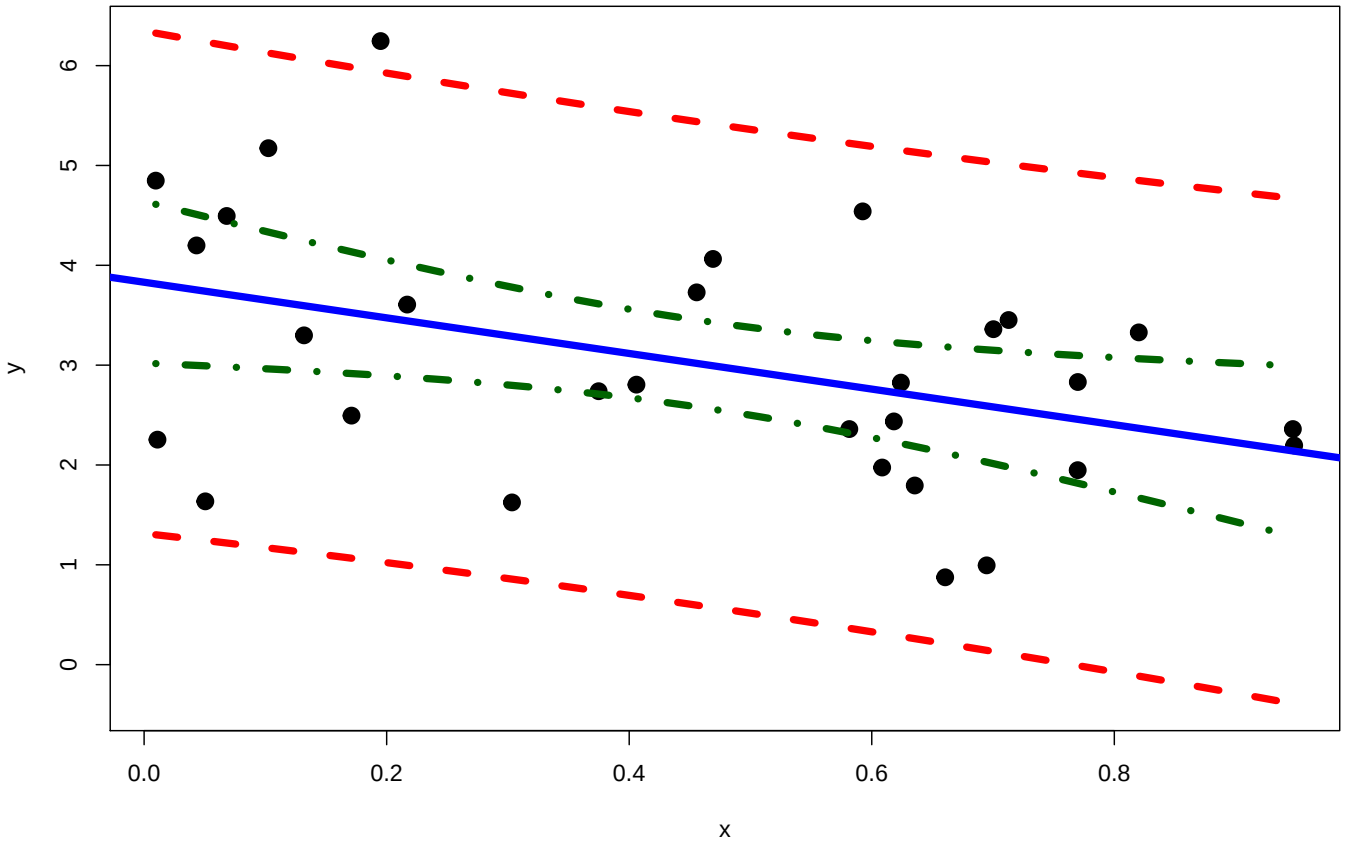
```
##        fit      lwr      upr
## 1 2.938444 2.498652 3.378235
```

```
predict(modelo, newdata = nuevoX, interval = "prediction")
```

```
##        fit       lwr      upr
## 1 2.938444 0.5159765 5.360911
```

# Bandas de confianza y predicción para valores de $Y$.

- Si repetimos esos intervalos de confianza y predicción para *todos los valores* $x_0$ dentro del recorrido de la muestra se obtienen unas bandas alrededor de la recta de regresión, más anchas en los extremos del rango y más estrechas en la zona central. En el ejemplo (ver el código que las dibuja). La banda de confianza se muestra en verde y la de predicción en rojo. Ambas se ensanchan hacia el borde pero el efecto es mucho más apreciable en la de confianza:



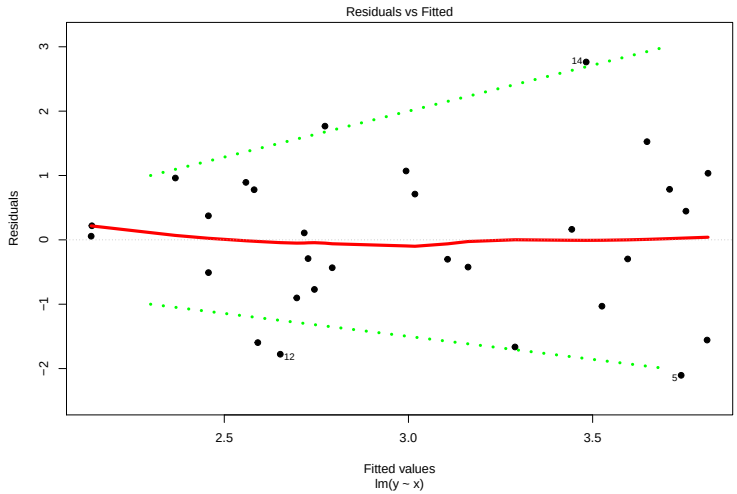
Sección 5

# Diagnósticos del modelo de regresión lineal simple.

# Gráficos de residuos con R para el diagnóstico del modelo de regresión.

- Ya hemos visto que la función `lm` permite acceder a los residuos del modelo. Además, si después de construir un modelo ejecutamos `plot(nombre_del_modelo, which = numero_de_1_a_5)` accederemos a cinco gráficos muy útiles para e diagnóstico del modelo, Para nuestro último ejemplo accedemos al primero de esos gráficos si hacemos (las líneas verdes de trazos las hemos añadido a posteriori con `segments`):

```
plotModelo = plot(modelo, which = 1, pch=19, lwd= 4)
segments(x0 = c(2.3, 2.3), y0 = c(1, -1), x1 = c(3.7, 3.7), y1 = c(3, -2),
         lty=3, lwd=4, col="green")
```

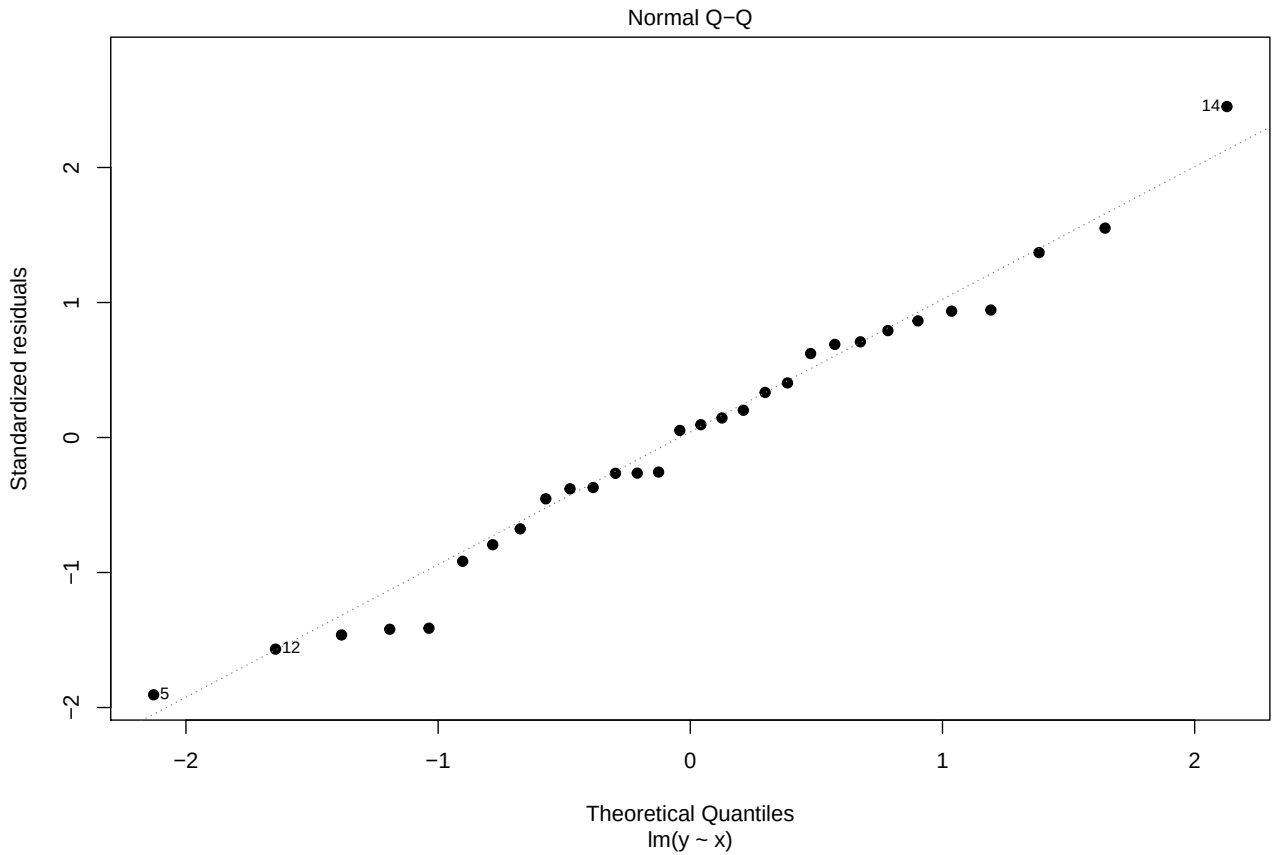



En este gráfico de *residuos frente a valores predichos* si las hipótesis se cumplen: (a) los puntos se distribuyen verticalmente de forma aleatoria y homogénea en todo el gráfico, formando una especie de banda horizontal de anchura similar y sin que ningún punto destaque frente al resto. En este ejemplo concreto hemos añadido las líneas de trazos verdes para destacar que parece haber una cierta forma de "cuña" en los datos, que podrían indicar falta de homogeneidad de las varianzas (*heterocedasticidad*). ¡Pero recuerda que la muestra es pequeña!

# Normalidad de los residuos, QQ-plot.

- La segunda de las gráficas sirve para analizar la normalidad de los residuos mediante un qq-plot, que ya vimos en el Tema 4.

```
plotModelo = plot(modelo, which = 2, pch=19)
```

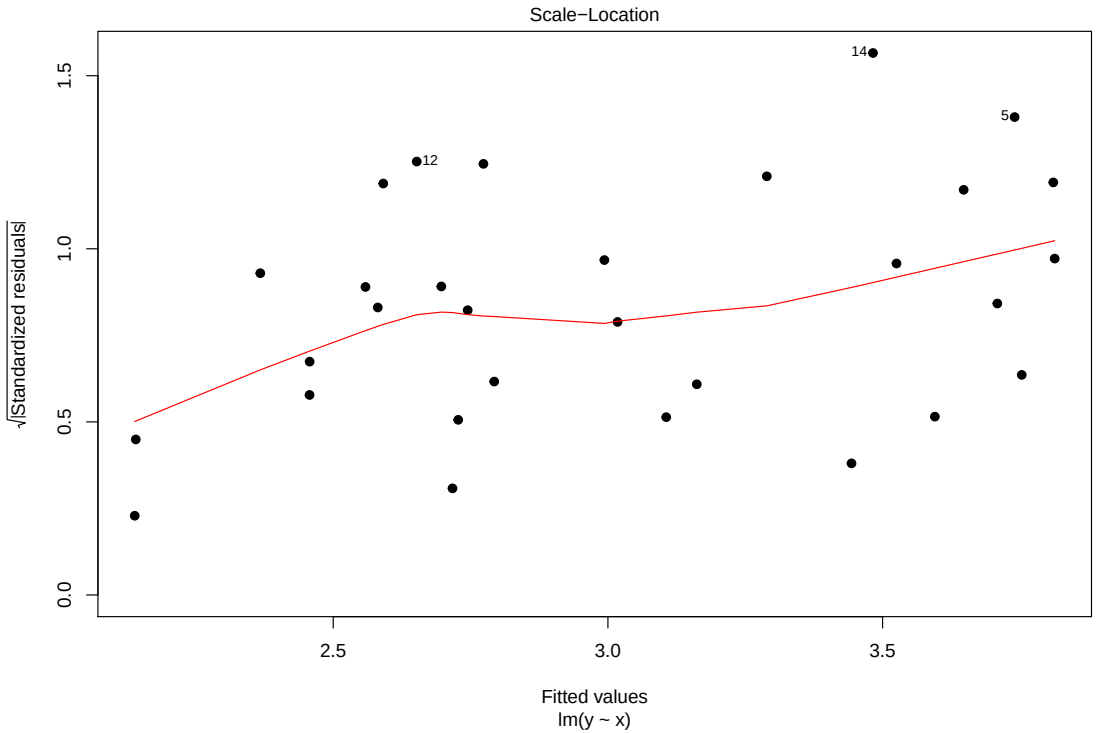



En este ejemplo concreto no parece haber problemas demasiado importantes con esa hipótesis.

# Gráfico scale-location.

- En el tercer tipo de gráfico diagnóstico lo que buscamos es: (−) que la línea roja sea aproximadamente horizontal. (−) que la anchura de la nube de puntos sea homogénea a lo ancho del gráfico. La información de este gráfico muchas veces complementa y refuerza la del primero. Aquí de nuevo vemos un patrón que nos hace sospechar de posible falta de homogeneidad de las varianzas.

```
plotModelo = plot(modelo, which = 3, pch=19)
```

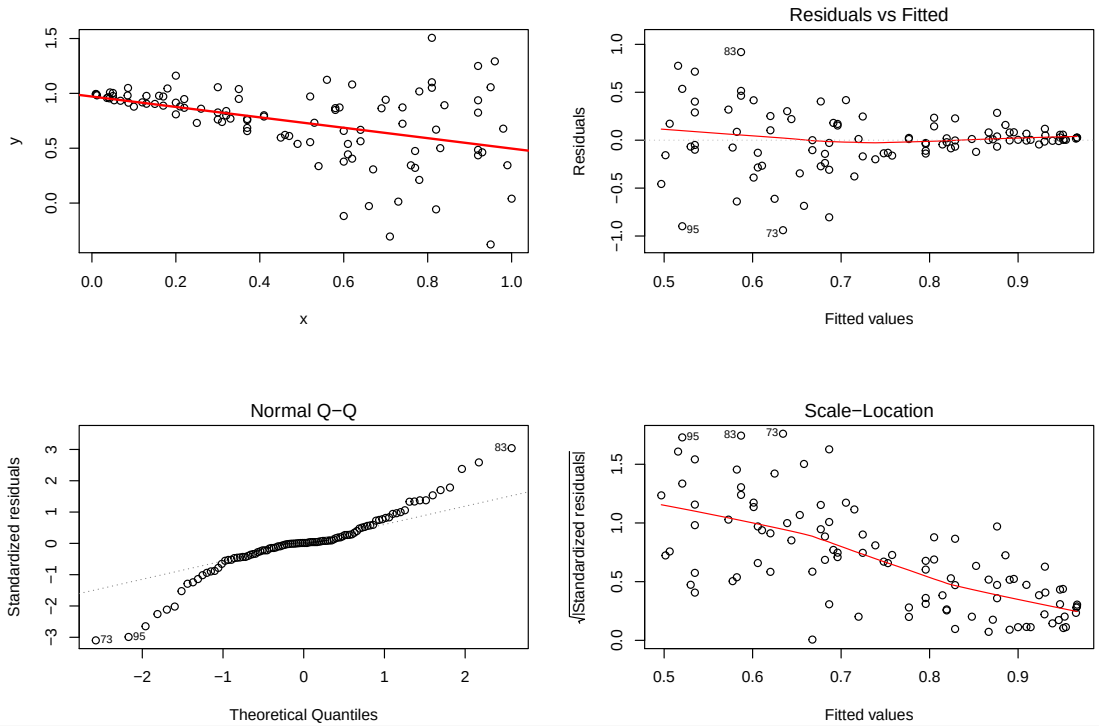



- Los gráficos cuarto y quinto se refieren a medidas de influencia y palanca para residuos atípicos. Volveremos sobre ellos tras discutir esas ideas.

# Ejemplos adicionales.

- En este ejemplo se han simulado unos datos que no cumplen la hipótesis de homogeneidad de las varinzas (homocedasticidad). Fíjate en que la vaarianza depende de $x$. Los gráficos diagnósticos de este caso reflejan de forma acusada ese problema. El primero de los cuatro gráficos es simplemente el diagrama de dispersión con la recta de regresión.

```
set.seed(2019)
n=100
x = sort(signif(runif(n, min = 0, max = 1), digits=2) )
y = 1 - (x/2) + rnorm(n, sd = 0.01*(1 + 50 * x))
par(mfrow=c(2, 2))
plot(x, y)
abline(lm(y ~x), col="red", lwd=2)
plot(lm(y ~x), which = 1:3)
```

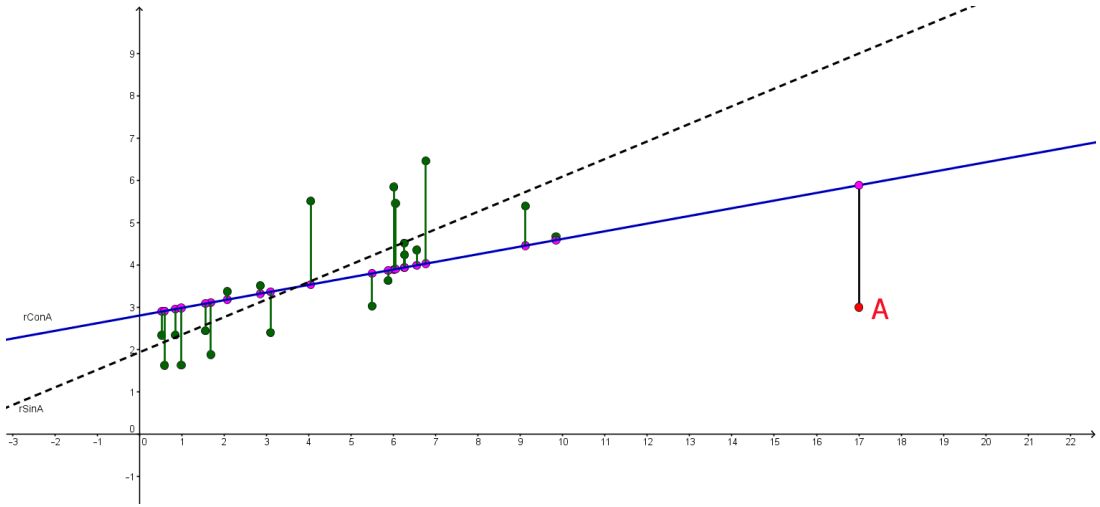



```
par(mfrow=c(1, 1))
```

# Valores atípicos y puntos influyentes en la regresión.

- A veces sucede que algún punto $(x_i, y_i)$ de la muestra afecta de manera exagerada al resultado del modelo. Y en ese caso diremos que $(x_i, y_i)$ es un *punto influyente* de la muestra. Es una situación similar a la de los puntos atípicos, pero aquí al existir dos coordenadas las cosas se complican un poco.
- Se puede pensar en la recta de regresión como un balancín apoyado en el punto $(\bar{x}, \bar{y})$ por el que siempre pasa. Hay dos mecanismos por los que un punto pueda llegar tener un efecto muy grande en la posición de la recta de regresión:

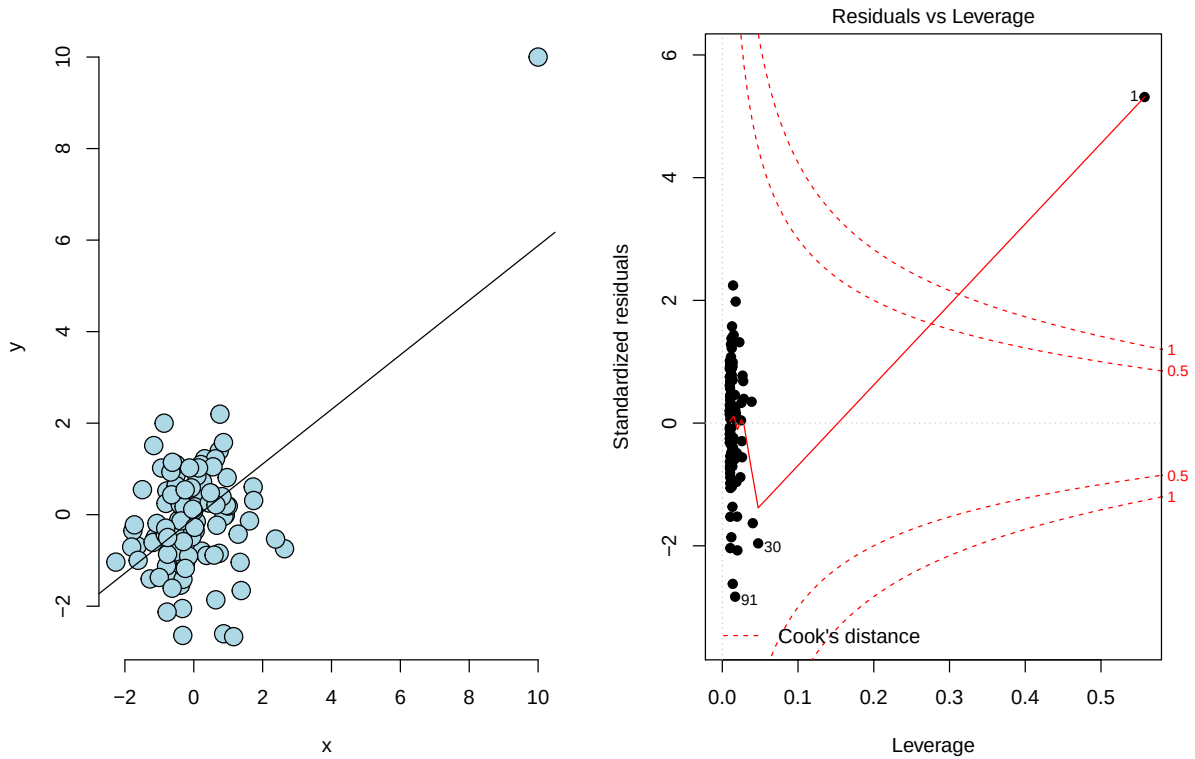
  (1) Puede tener una coordenada $x$ muy grande, con mucho brazo de *palanca (leverage)* muy largo. El punto $A$ de la figura tiene esa propiedad (se muestran las rectas de regresión con y sin A).




  (2) Aunque su coordenada $x$ no sea atípica puede tener un residuo excepcionalmente grande, como si una persona muy pesada se sentara en el balancín. Puedes explorar estas ideas en este enlace.

# Análisis del brazo de palanca (*leverage*) con R.

- La distancia de Cook es una medida que se usa a menudo para estiar el brazo de palanca (*leverage*) de los puntos. El último de los gráficos que se obtinene con `plot(modelo_con_lm)` muestra información sobre esa distancia. Si alguno de los puntos tiene mucha palanca, lo veremos situado fuera de las bandas de trazos que R dibuja. En este ejemplo de Brian Caffo (ver las Referencia y el código en la siguiente página) vemos como se refleja ese punto en el gráfico de diagnóstico:




En cualquier caso la palanca es *capacidad para la influencia* y un punto con mucha palanca puede ser o no influyente.

# Medidas de influencia. Hatvalues.

- Para medir la influencia se utiliza otro conjunto de valores, los llamados *hat values*. En R los podemos obtener con `hatvalues(modelo)`. Los valores del ejemplo anterior son:

```
set.seed(2019)
n <- 100
x <- c(10, rnorm(n))
y <- c(10, c(rnorm(n)))
modelo = lm(y ~x)
```

El punto "especial" se ha colocado al principio. Sus `hatvalues` (se muestran los primeros) son

```
head(hatvalues(modelo))
```

```
##          1          2          3          4          5
## 0.55756096 0.01269298 0.01151335 0.02519289 0.01425842
##          6
## 0.01911817
```

Y está claro que el primero es mucho mayor. En general los puntos con hatvalue mayor que $4/n$ se consideran puntos influyentes ($n$ es el tamaño muesral). Y como pasaba con los atípicos, al encontrar puntos influyentes tenemos que investigar específicamente qué ocurre con esos puntos, si se deben a errores o algún otra particularidad de los datos.

Sección 6

# Complementos de R.

# Datos limpios (tidy data).

- El Capítulo 12 de *R for Data Science* es una lectura casi obligada, ya que H. Wickham es el creador del concepto de *datos limpios*.
- Un conjunto de datos se considera *limpio* si cumple estas tres condiciones:

  1. Cada variable tiene su propia columna.
  2. Cada observación tiene su propia fila.
  3. Cada valor tiene su propia celda.

- Por ejemplo, los datos del conjunto de datos `anscombe` no son limpios, porque las filas no corresponden a observaciones:

```
head(anscombe, 3)
```

```
##   x1 x2 x3 x4   y1   y2    y3   y4
## 1 10 10 10  8 8.04 9.14  7.46 6.58
## 2  8  8  8  8 6.95 8.14  6.77 5.76
## 3 13 13 13  8 7.58 8.74 12.74 7.71
```

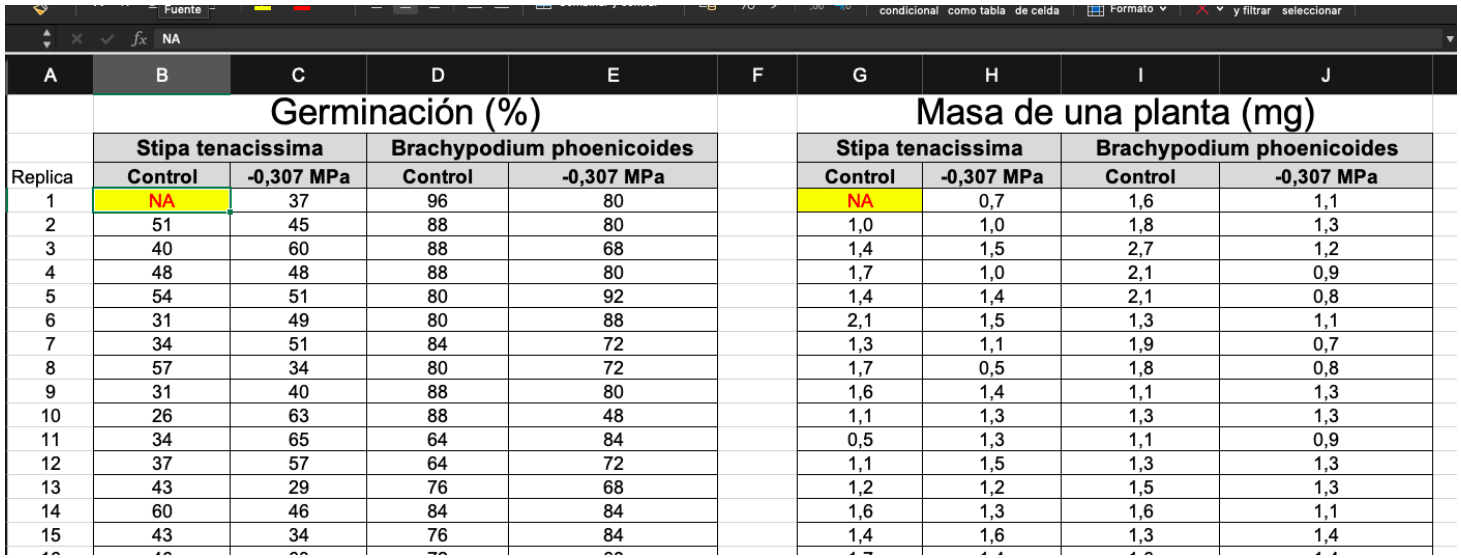
  Este otro ejemplo, contenido en el `tidyverse`, tampoco es un conjunto de datos limpio porque hay variables distintas almacenadas en una misma columna. ¿Cuáles son las unidades de la columan `count`?

```
library(tidyverse)
head(table2, 4)
```

```
## # A tibble: 4 x 4
##   country      year type          count
##   <chr>       <int> <chr>         <int>
## 1 Afghanistan  1999 cases           745
## 2 Afghanistan  1999 population 19987071
## 3 Afghanistan  2000 cases          2666
## 4 Afghanistan  2000 population 20595360
```

# Ejemplos varios.

- En otro ejemplo reciente, unos biólogos querían hacer unos análisis de unos datos de germinación de unas plantas obtenidos en un estudio de campo. Por comodidad a la hora de recoger los datos, estos estaban organizados en un *estadillo* que a su vez se reflejaba en una tabla Excel como la de la figura:



| | Germinación (%) | | | | Masa de una planta (mg) | | | |
|---|---|---|---|---|---|---|---|---|
| | Stipa tenacissima | | Brachypodium phoenicoides | | Stipa tenacissima | | Brachypodium phoenicoides | |
| Replica | Control | -0,307 MPa | Control | -0,307 MPa | Control | -0,307 MPa | Control | -0,307 MPa |
| 1 | NA | 37 | 96 | 80 | NA | 0,7 | 1,6 | 1,1 |
| 2 | 51 | 45 | 88 | 80 | 1,0 | 1,0 | 1,8 | 1,3 |
| 3 | 40 | 60 | 88 | 68 | 1,4 | 1,5 | 2,7 | 1,2 |
| 4 | 48 | 48 | 88 | 80 | 1,7 | 1,0 | 2,1 | 0,9 |
| 5 | 54 | 51 | 80 | 92 | 1,4 | 1,4 | 2,1 | 0,8 |
| 6 | 31 | 49 | 80 | 88 | 2,1 | 1,5 | 1,3 | 1,1 |
| 7 | 34 | 51 | 84 | 72 | 1,3 | 1,1 | 1,9 | 0,7 |
| 8 | 57 | 34 | 80 | 72 | 1,7 | 0,5 | 1,8 | 0,8 |
| 9 | 31 | 40 | 88 | 80 | 1,6 | 1,4 | 1,1 | 1,3 |
| 10 | 26 | 63 | 88 | 48 | 1,1 | 1,3 | 1,3 | 1,3 |
| 11 | 34 | 65 | 64 | 84 | 0,5 | 1,3 | 1,1 | 0,9 |
| 12 | 37 | 57 | 64 | 72 | 1,1 | 1,5 | 1,3 | 1,3 |
| 13 | 43 | 29 | 76 | 68 | 1,2 | 1,2 | 1,5 | 1,3 |
| 14 | 60 | 46 | 84 | 84 | 1,6 | 1,3 | 1,6 | 1,1 |
| 15 | 43 | 34 | 76 | 84 | 1,4 | 1,6 | 1,3 | 1,4 |

  en este conjunto de datos ni filas ni columnas corresponden a observaciones ni variables de una manera limpia.

- Descarga el fichero students3.csv, ábrelo con R y piensa qué problemas tiene esta tabla de datos.

# Herramientas para limpiar datos con tidyR.

- La librería `tidyR` (parte del `tidyverse`) contiene varias funciones que llevan a cabo operaciones que permiten limpiar muchos conjuntos de datos. Puedes consultar este resumen de comandos Las más importantes son:

  – `gather`: se aplica cuando los nombres de algunas columnas de la tabla no son realmente variables, sino valores de una variable (que no aparece por su nombre en la tabla). Por ejemplo, algunas columnas pueden ser nombres de países. Al aplicar `gather` obtenemos una tabla que es más estrecha y larga.
  – `spread`: lo usamos cuando una observación está repartida en varias filas de la tabla o cuando una columna contiene nombres de variables como en algunos ejemplos que hemos visto. Esta función produce tablas más anchas y cortas. – `spread`: Otras funciones auxiliares de `tidyR` como `separate` y `unite` son especialmente útiles para trabajar con columnas que contienen varias variables agrupadas con algún formato. Por ejemplo, puede ser conveniente separar una columna con fechas como 1988-02-15 en tres columnas año, mes, día.

- Es muy recomendable ver los esquemas gráficos que aparecen en el resumen de tidyR elaborado por RStudio (enlace al final del tema) para ver gráficamente el efecto de las operaciones `gather` y `spread`.

# Ejemplo de gather.

- La tabla de datos `USArrests` de la librería `datasets` comienza así:

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

  Fijate en que hay tres columnas que en realidad contienen la variable *tipo de delito* (tasa por 100000 habitantes). Vamos a usar `gather` para crear una variable llamada `felony` a partir de esas tres columnas:

```
USArrests %>%
  gather("Murder", "Assault", "Rape", key = "Felony", value = "ratePer100K") %>%
  head(5)
```

```
##   UrbanPop Felony ratePer100K
## 1       58 Murder        13.2
## 2       48 Murder        10.0
## 3       80 Murder         8.1
## 4       50 Murder         8.8
## 5       91 Murder         9.0
```

- **Ejercicio:** mira las dimensiones de las dos tablas, para ver que ha pasado.

# Ejemplo de spread

- Vamos a usar `tabla2` del `tidyverse` que hemos visto antes (se muestra el comienzo):

```
## # A tibble: 4 x 4
##   country      year type           count
##   <chr>       <int> <chr>          <int>
## 1 Afghanistan  1999 cases            745
## 2 Afghanistan  1999 population 19987071
## 3 Afghanistan  2000 cases           2666
## 4 Afghanistan  2000 population 20595360
```

Aplicamos

```
table2 %>%
  spread(key = "type", value = "count")
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

y ahora las variables de *cases* y population ya tienen columnas propias.

# Ejemplo de separate.

- Supongamos dada una tabla `datos` como esta en la que la segunda columna contiene códigos con una cierta estructura.

```
##    x codigo
## 1  9    1/B
## 2 10    5/A
## 3  1    2/A
## 4  8    5/B
## 5  3    5/A
## 6  7    1/B
```

A veces estaremos interesados en separar las dos partes del código. La función `separate` permite hacer esto y es suficientemente lista como para adivinar lo que queremos en casos sencillos:

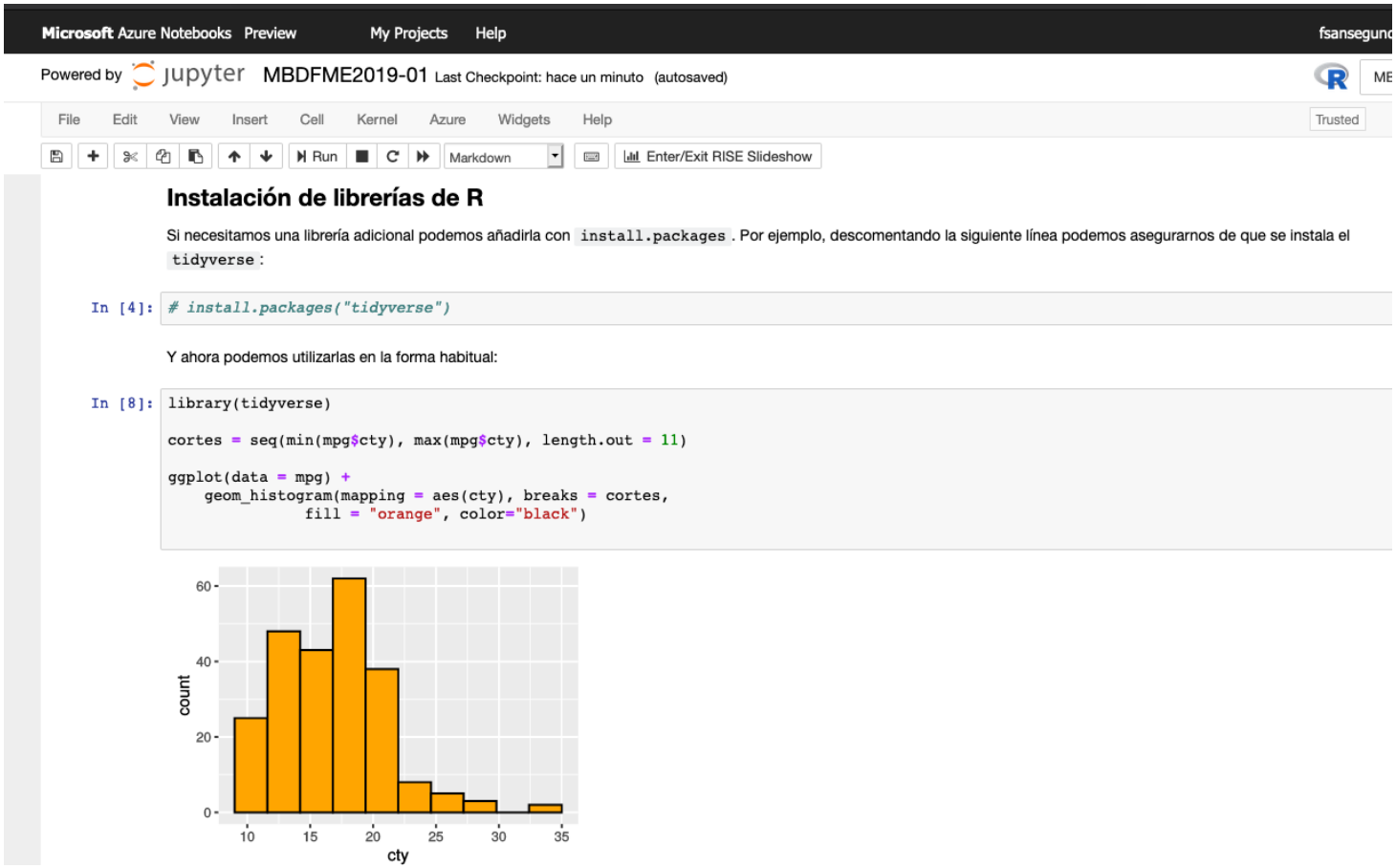
```
datos %>%
  separate("codigo", into = c("Numero", "Letra"))
```

```
##    x Numero Letra
## 1  9      1     B
## 2 10      5     A
## 3  1      2     A
## 4  8      5     B
## 5  3      5     A
## 6  7      1     B
```

La función `unite` sirve para hacer el proceso contrario. Cuando veamos fechas con R veremos más ejemplos de uso de estas funciones.

# Notebooks typo Jupyter con R en Azure.

- En clase veremos un ejemplo de este tipo del uso de R con notebooks Jupyter (gratuitos a día de hoy) en la plataforma Azure de Cloud Computing de Microsoft. Ver este enlace.



# Referencias para la sesión

**Enlaces**

- Resumen sobre tidyR (chuleta) elaborado por RStudio.
- Código de esta sesión
- Regression Models de Brian Caffo en Leanpub.

**Bibliografía**

Haftorn, Svein, and Randi Eidsmo Reinertsen. 1985. "The effect of temperature and clutch size on the energetic cost of incubation in a free-living blue tit (Parus caeruleus)." *The Auk*. JSTOR, 470–78.

Wickham, Hadley, and Garrett Grolemund. 2016. *R for data science: import, tidy, transform, visualize, and model data*. O'Reilly Media, Inc.