



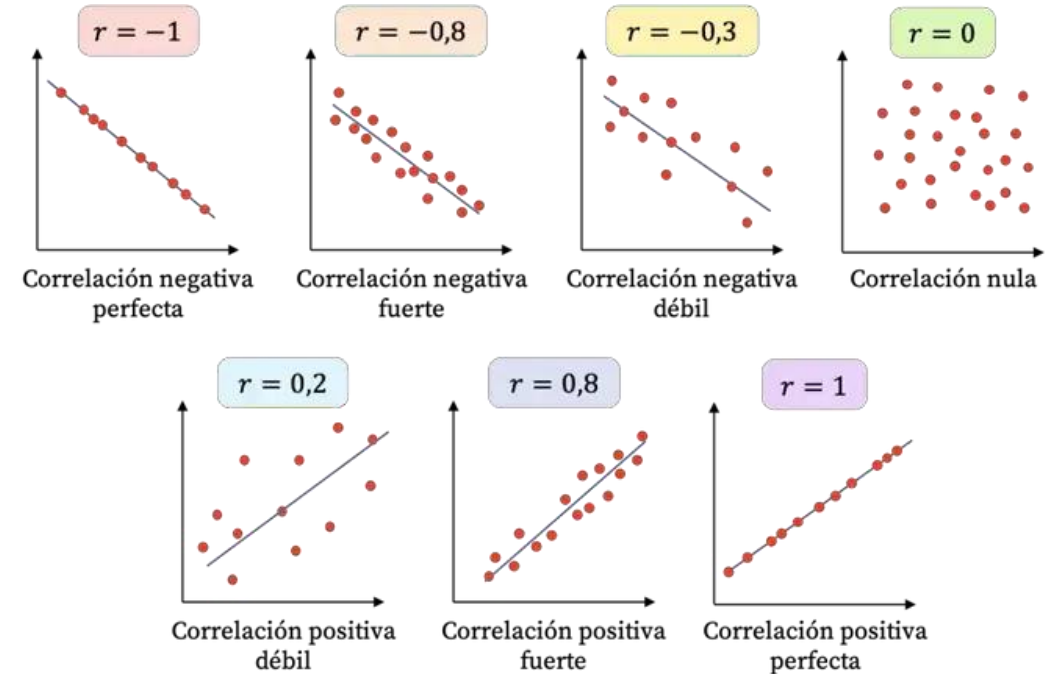
# ANÁLISIS DE DATOS

HERRAMIENTAS COMPUTACIONALES: EL ARTE DE LA ANALÍTICA

SEMANA TEC

# CORRELACIÓN

- Medida estadística
- Expresa el cambio linear conjunto de una variable en relación con otra
- Ayuda a predecir relaciones de causa y efecto
- Toma valores entre  $-1$  y  $+1$ 
  - Valores positivos indican un crecimiento conjunto de ambas variables
  - Valores negativos indican que una variable crece, mientras que la otra decrece
  - Valor de cero indica que no existe correlacion



# CORRELACIÓN EN R

- Dataset

<https://verso.mat.uam.es/~joser.berrendero/datos/EdadPesoGrasas.txt>

- Lectura en R


```
dataGrasas <-  
read.table("https://verso.mat.uam.es/~jose  
r.berrendero/datos/EdadPesoGrasas.txt",  
header = TRUE)
```

	peso	edad	grasas
1	84	46	354
2	73	20	190
3	65	52	405
4	70	30	263
5	76	57	451
6	69	25	302
7	63	28	288
8	72	36	385
9	79	57	402
10	75	44	365

Showing 1 to 11 of 25 entries, 3 total columns

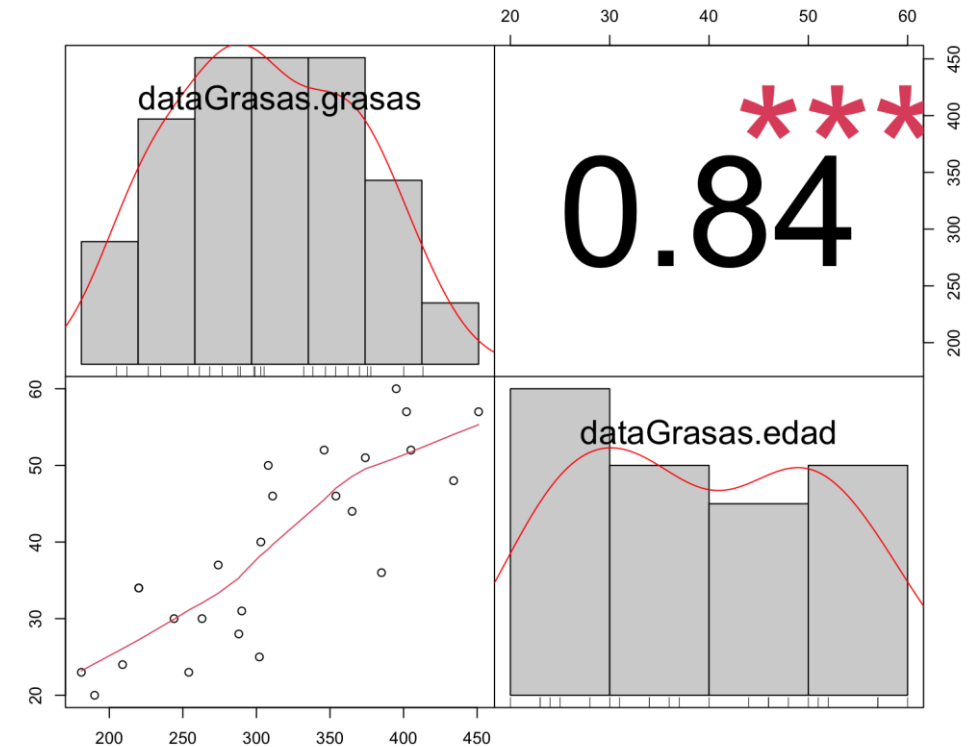
# CORRELACIÓN

MATRIZ DE  
COEFICIENTES DE  
CORRELACIÓN

```
Console Terminal × Background Jobs ×  
R 4.3.3 · ~/   
> cor(dataGrasas)  
           peso      edad      grasas  
peso  1.0000000 0.2400133 0.2652935  
edad  0.2400133 1.0000000 0.8373534  
grasas 0.2652935 0.8373534 1.0000000  
> |
```

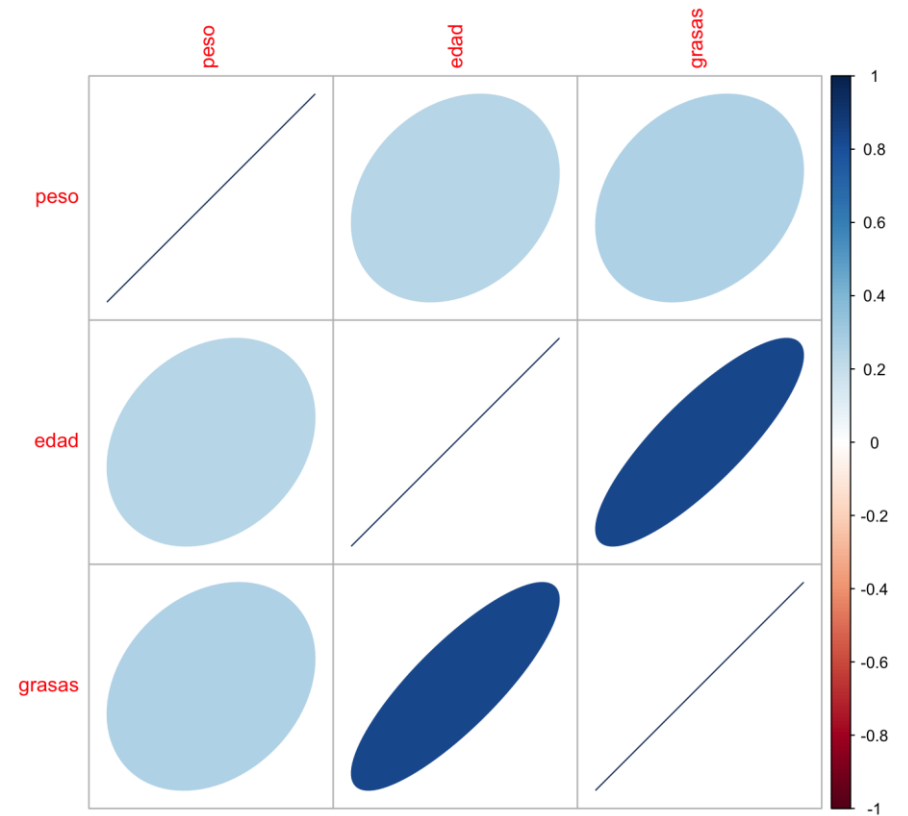
# CORRELACIÓN ENTRE GRASAS Y EDAD

```
9 #usando libreria externa
10 library(PerformanceAnalytics)
11 #correlación entre 2 variables
12 cor(dataGrasas$grasas,dataGrasas$edad)
13 corGE <- data.frame(dataGrasas$grasas,dataGrasas$edad)
14 chart.Correlation(corGE)
15
```



# GRÁFICA MATRIZ CORRELACIÓN

```
#Grafica matriz de correlación  
library(corrplot)  
corrGrasas <- cor(dataGrasas)  
corrplot(corrGrasas,method = "ellipse")
```



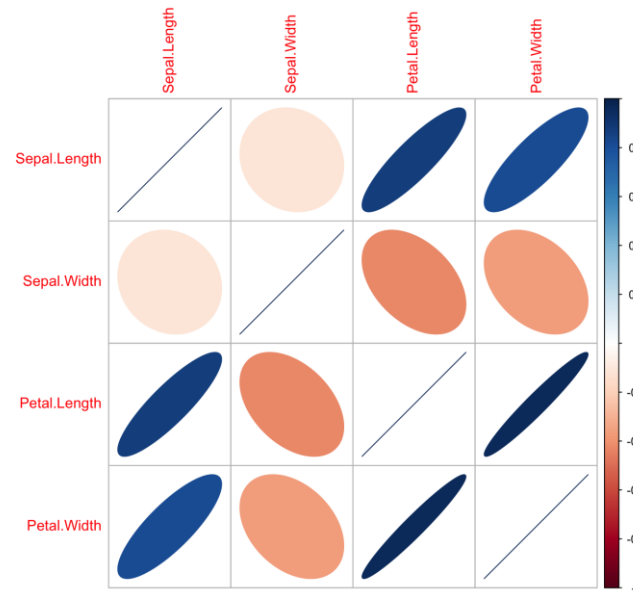
2024FJ juresti@tec.mx	Filter				
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.5	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa

# ELIMINAR VARIABLES

- Iris dataset tiene atributos no numéricos
- Para la correlación todos los atributos tienen que ser numéricos



# ELIMINAR VARIABLES ...



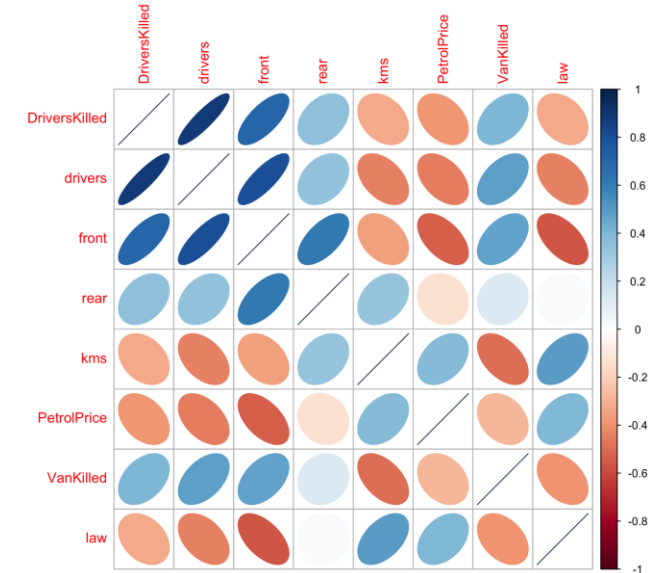
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1
11	5.4	3.7	1.5	0.2
12	4.8	3.4	1.6	0.2
13	4.8	3.0	1.4	0.1
14	4.3	3.0	1.1	0.1
15	5.8	4.0	1.2	0.2
16	5.7	4.4	1.5	0.4
17	5.4	3.9	1.3	0.4

```
#Ejemplo con dataset Iris
data("iris")
View(iris)
#Eliminar variable Species
iris$Species = NULL
#Ya era un dataframe
corrIris <- cor(iris)
corrplot(corrIris,method = "ellipse")
```



# OTRO EJEMPLO ELIMINAR VARIABLES

	DriversKilled	drivers	front	rear	kms	PetrolPrice	VanKilled	law
1	107	1687	867	269	9059	0.10297181	12	0
2	97	1508	825	265	7685	0.10236300	6	0
3	102	1507	806	319	9963	0.10206249	12	0
4	87	1385	814	407	10955	0.10087330	8	0
5	119	1632	991	454	11823	0.10101967	10	0
6	106	1511	945	427	12391	0.10058119	13	0
7	110	1559	1004	522	13460	0.10377398	11	0
8	106	1630	1091	536	14055	0.10407640	6	0
9	107	1579	958	405	12106	0.10377398	10	0
10	134	1653	850	437	11372	0.10302640	16	0
11	147	2152	1109	434	9834	0.10273011	13	0
12	180	2148	1113	437	9267	0.10199719	14	0
13	125	1752	925	316	9130	0.10127456	14	0
14	134	1765	903	311	8933	0.10070398	6	0
15	110	1717	1006	351	11000	0.10013961	8	0
16	102	1558	892	362	10733	0.09862110	11	0
17	103	1575	990	486	12912	0.09834929	7	0

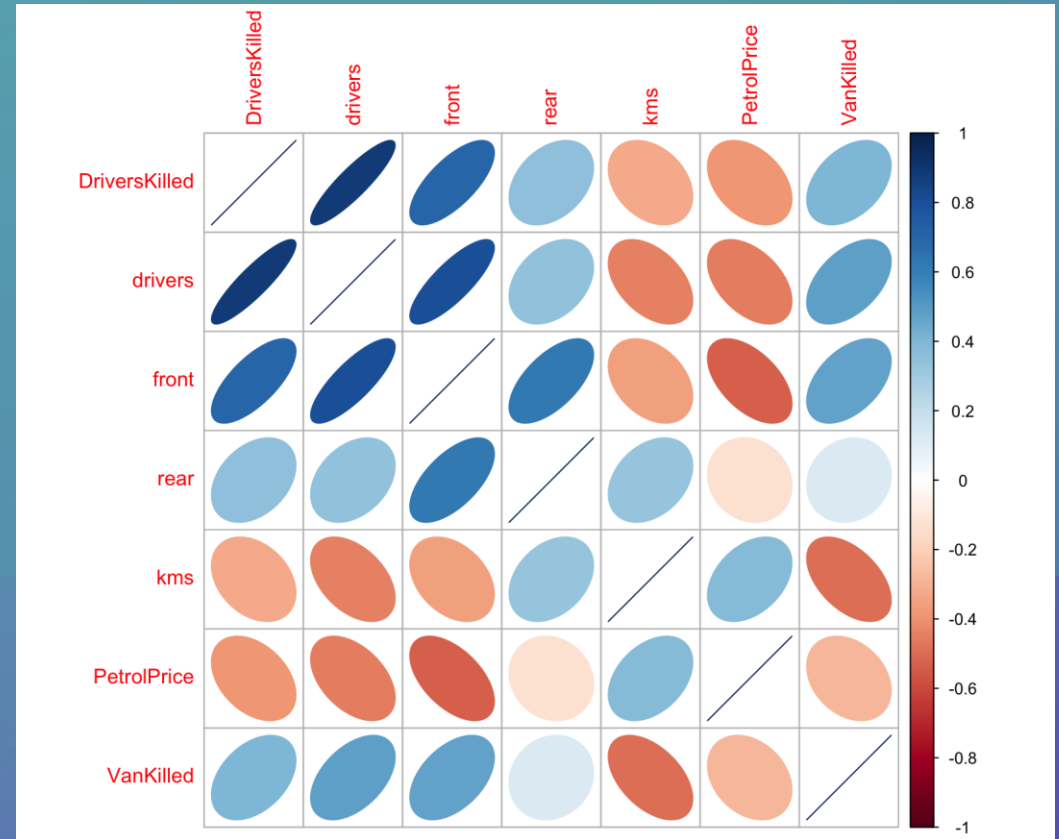


```
#Dataset con variables no utiles
data("Seatbelts")
View(Seatbelts)
#Atributo law es binario, posiblemente no sirve
corrSeat <- cor(Seatbelts)
corrplot(corrSeat,method = "ellipse")
```

# OTRO EJEMPLO ELIMINAR VARIABLES...

```
#Convertir dataset a frame para poder manipular
dataSeat <- data.frame(Seatbelts)
View(dataSeat)
#Borrar columna law
dataSeat$law = NULL
View(dataSeat)

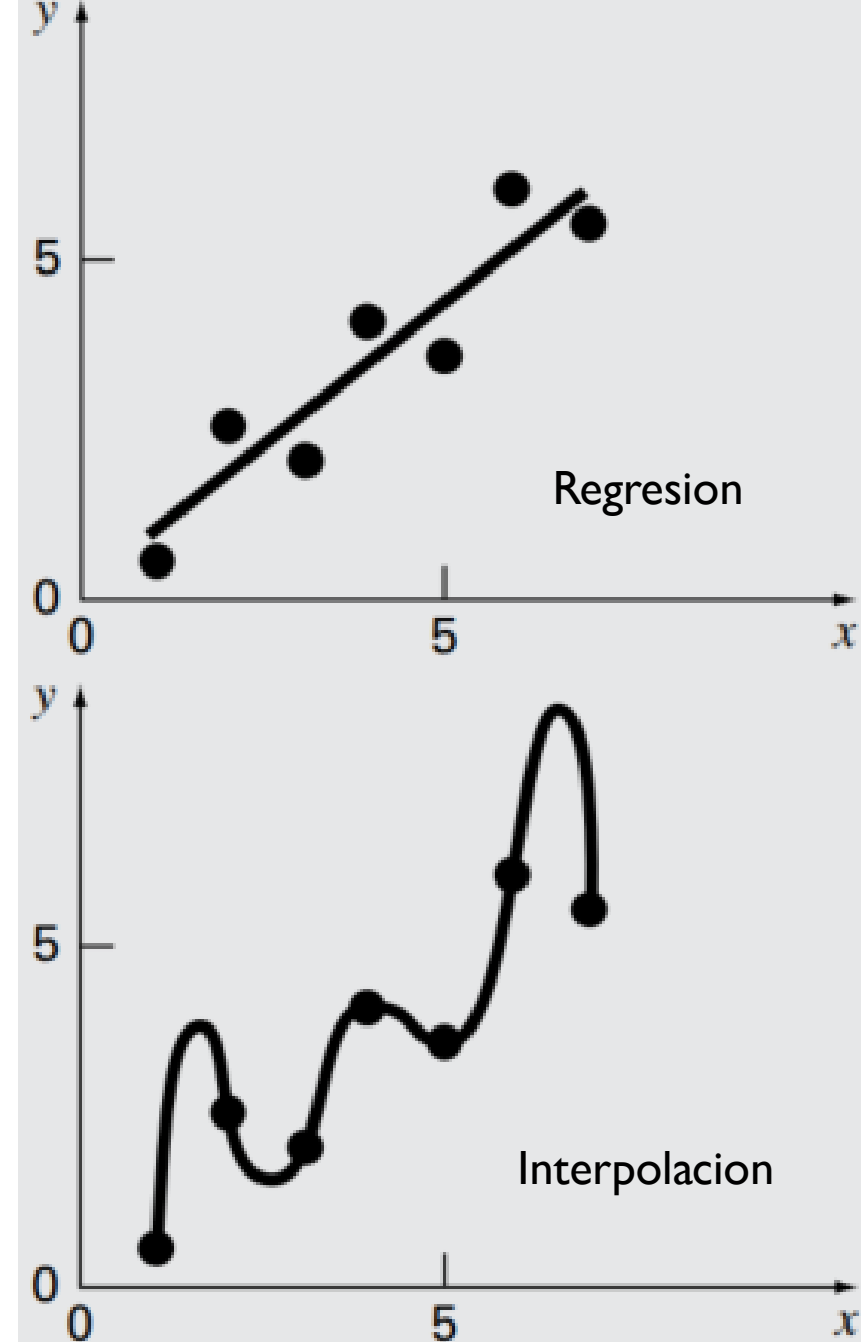
corrSeat <- cor(dataSeat)
corrplot(corrSeat,method = "ellipse")
```



# REGRESIÓN LINEAL

# REGRESIÓN LINEAL

- Permite predecir un punto estimado (algo que no ha sido observado)
- Genera un modelo que describe el comportamiento de los datos
- Puede ser utilizado cuando los datos tienen un poco de errores
- No es lo mismo que una interpolación

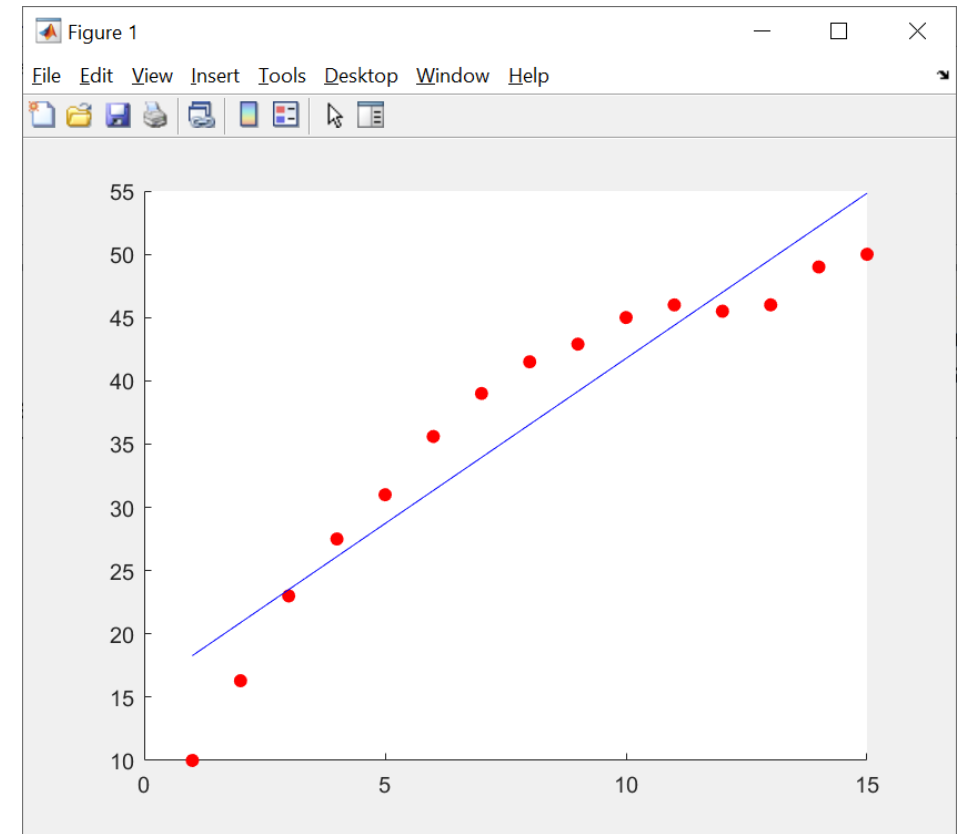


# REGRESIÓN LINEAL...

- Ajusta una linea a un conjunto de puntos observados
- Trata de encontrar la mayor relación entre la entrada y la salida
- Ecuación lineal del tipo

$$y = mx + b$$

donde  $m$  es la pendiente y  $b$  es la intersección



# REGRESIÓN LINEAL EN R

- **Primer argumento** fórmula  $y \sim x$ 
  - Y es la variable dependiente o de respuesta
  - X es la variable independiente o regresora
- El **segundo argumento**, llamado data especifica cuál es el archivo en el que se encuentran las variables.
- Ecuación obtenida es  $y = 102.575 + 5.321 x$

```
# Regresión lineal -----

regGrasas <- lm(dataGrasas$grasas ~ dataGrasas$edad)
summary(regGrasas)
```

Console   Terminal x   Background Jobs x

R 4.3.3 · ~/Desktop/ ↗

```
> summary(regGrasas)
```

Call:  
lm(formula = dataGrasas\$grasas ~ dataGrasas\$edad)

Residuals:

Min	1Q	Median	3Q	Max
-63.478	-26.816	-3.854	28.315	90.881

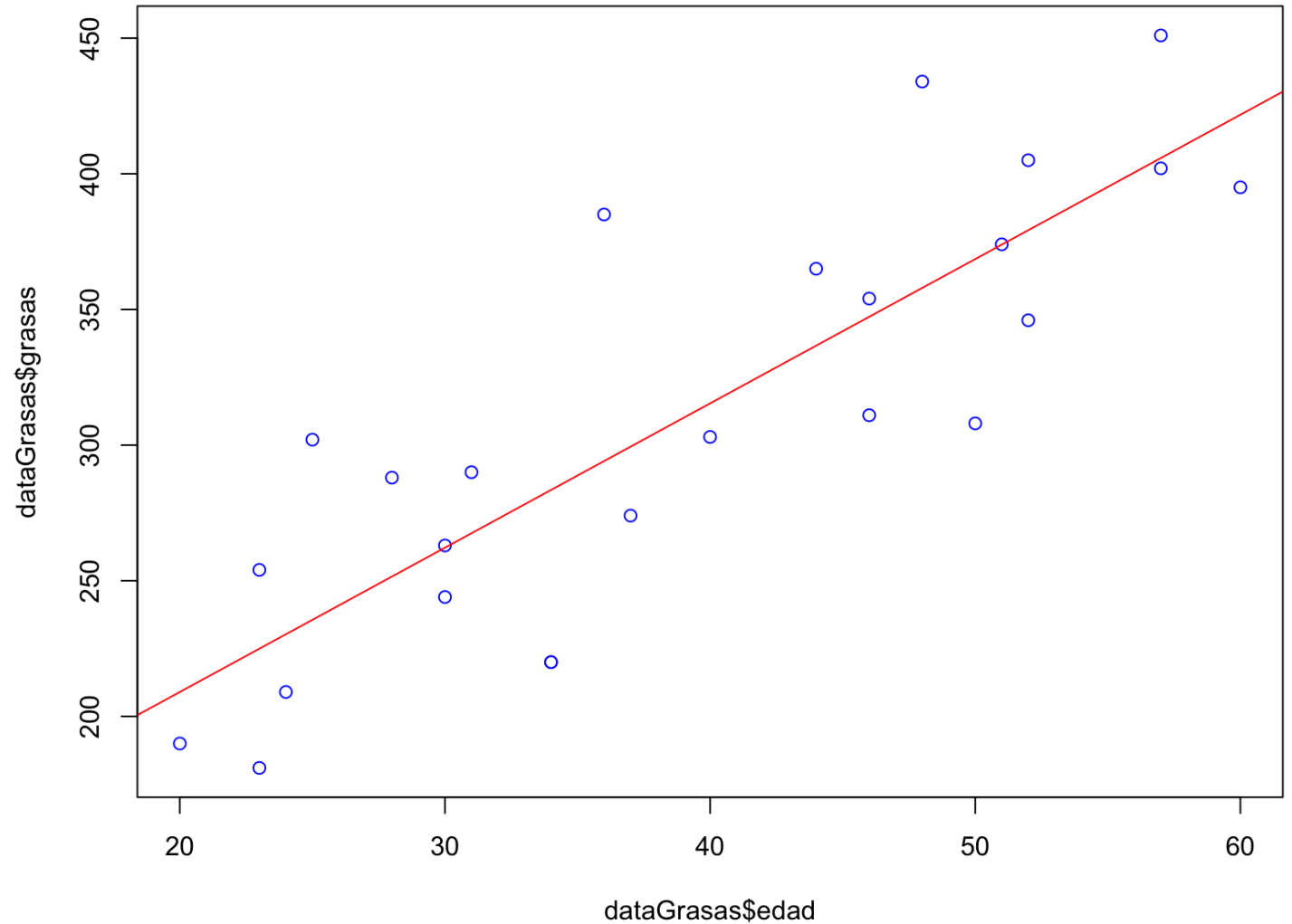
Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	102.5751	29.6376	3.461	0.00212 **
dataGrasas\$edad	5.3207	0.7243	7.346	1.79e-07 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 43.46 on 23 degrees of freedom  
Multiple R-squared: 0.7012,      Adjusted R-squared: 0.6882  
F-statistic: 53.96 on 1 and 23 DF, p-value: 1.794e-07

# GRÁFICA DE REGRESIÓN LINEAL



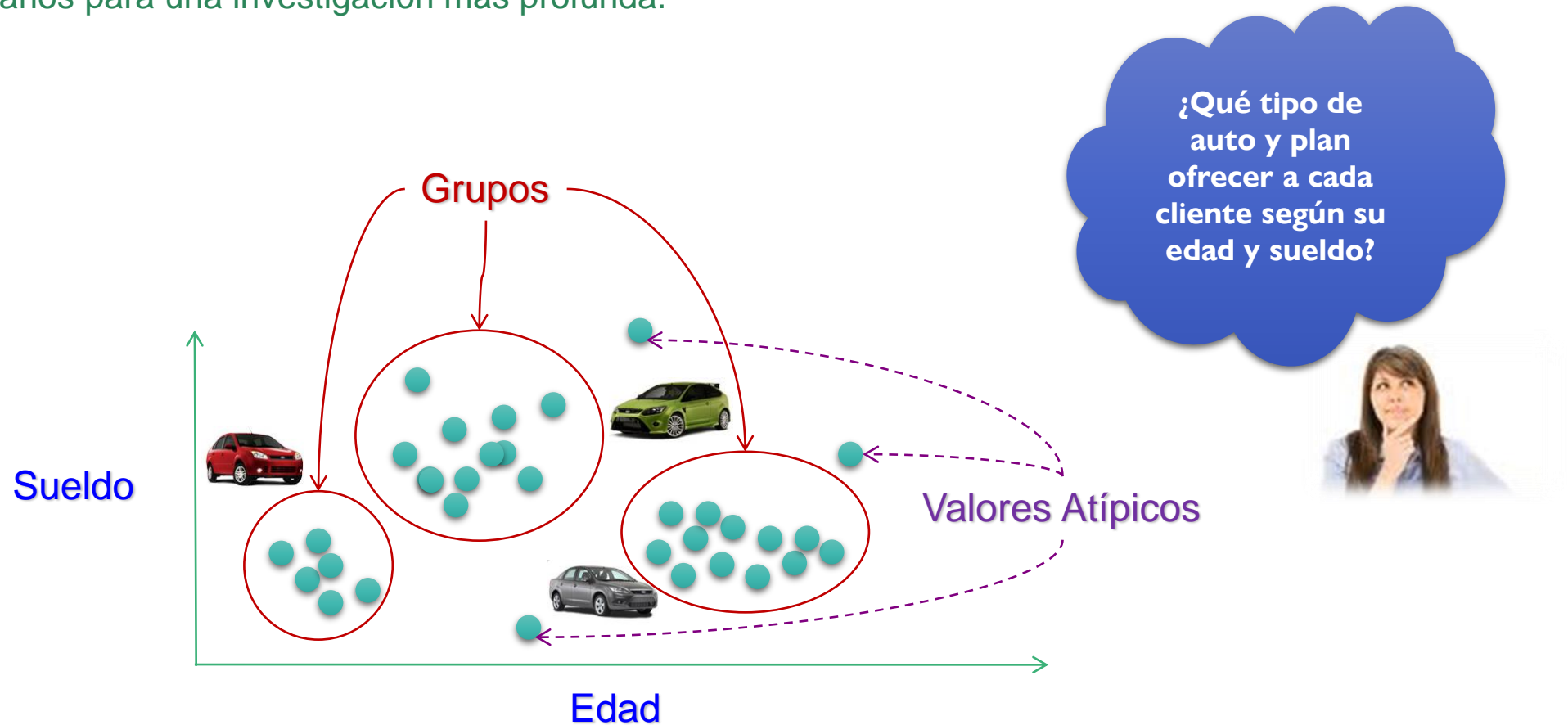
```
plot(dataGrasas$edad,dataGrasas$grasas,col="blue")  
abline(regGrasas,col="red")
```



# K - MEANS

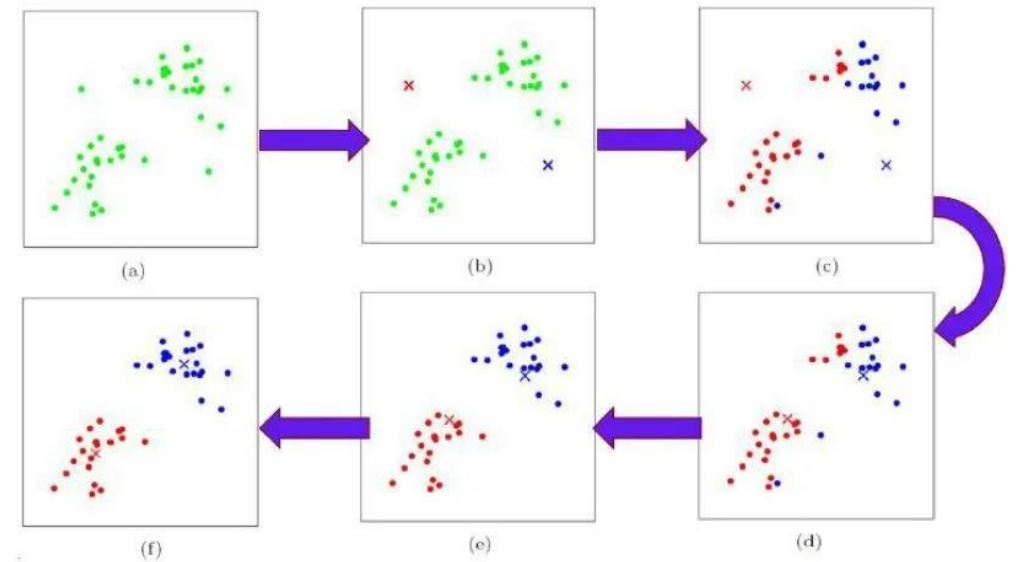
## Agrupación (Cluster)

Ofrece la capacidad de agrupar elementos de acuerdo a características que tengan en común para análisis así como detectar casos extraordinarios para una investigación más profunda.



# ALGORITMO K-MEANS

Select	Seleccionar el número de clusters (K)
Select	Seleccionar K puntos aleatorios del dataset • Serán utilizados como centroides de nuestros clusters
Assign	Asignar el resto de los puntos al cluster más cercano
Calculate	Calcular nuevos centroides para los clusters actuales
Repeat	Repetir los pasos Assign y Calculate hasta que se cumpla una condición de terminación

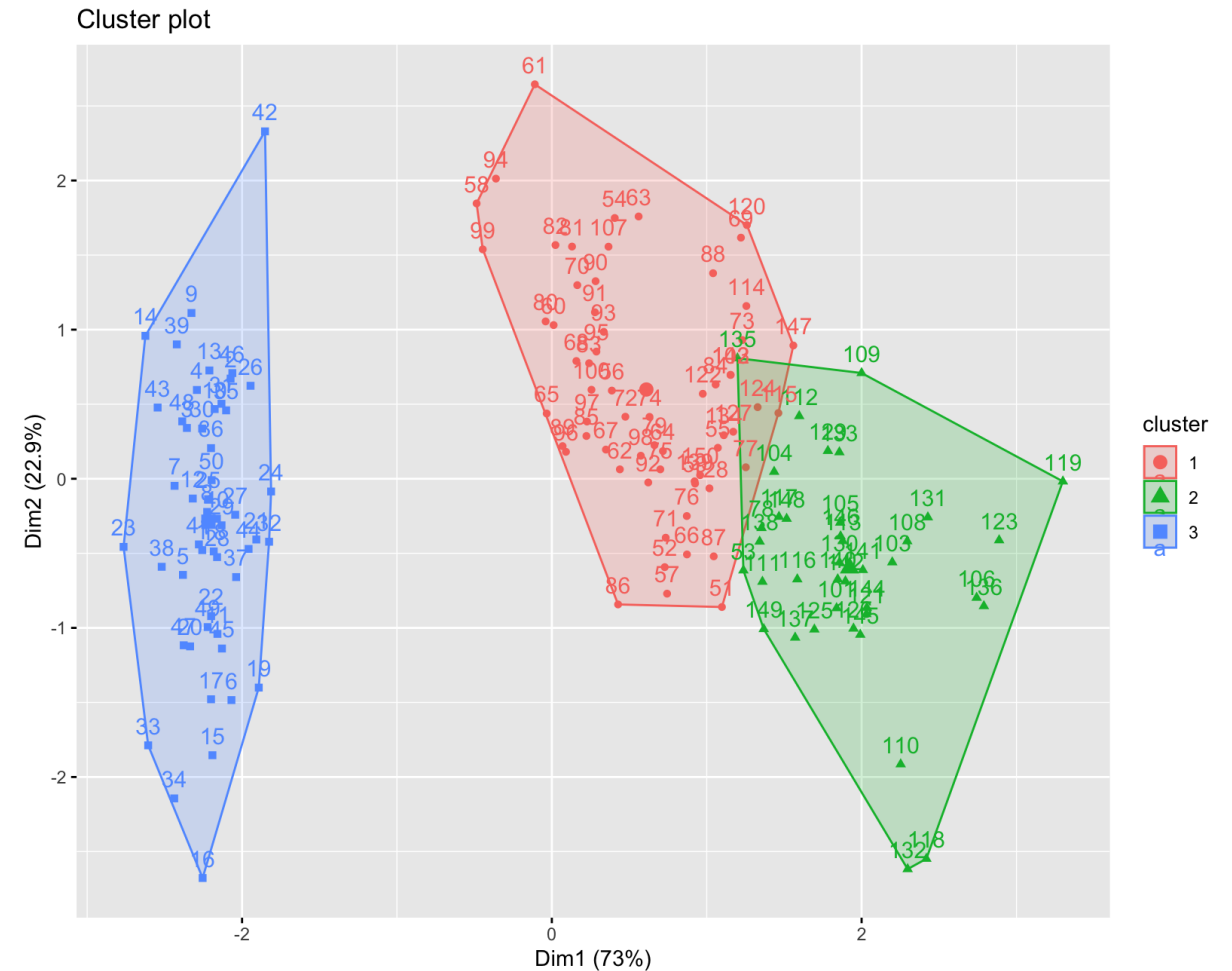


```
data("iris")
View(iris)
iris$Species = NULL
kM <- kmeans(iris,3)
kM
```

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

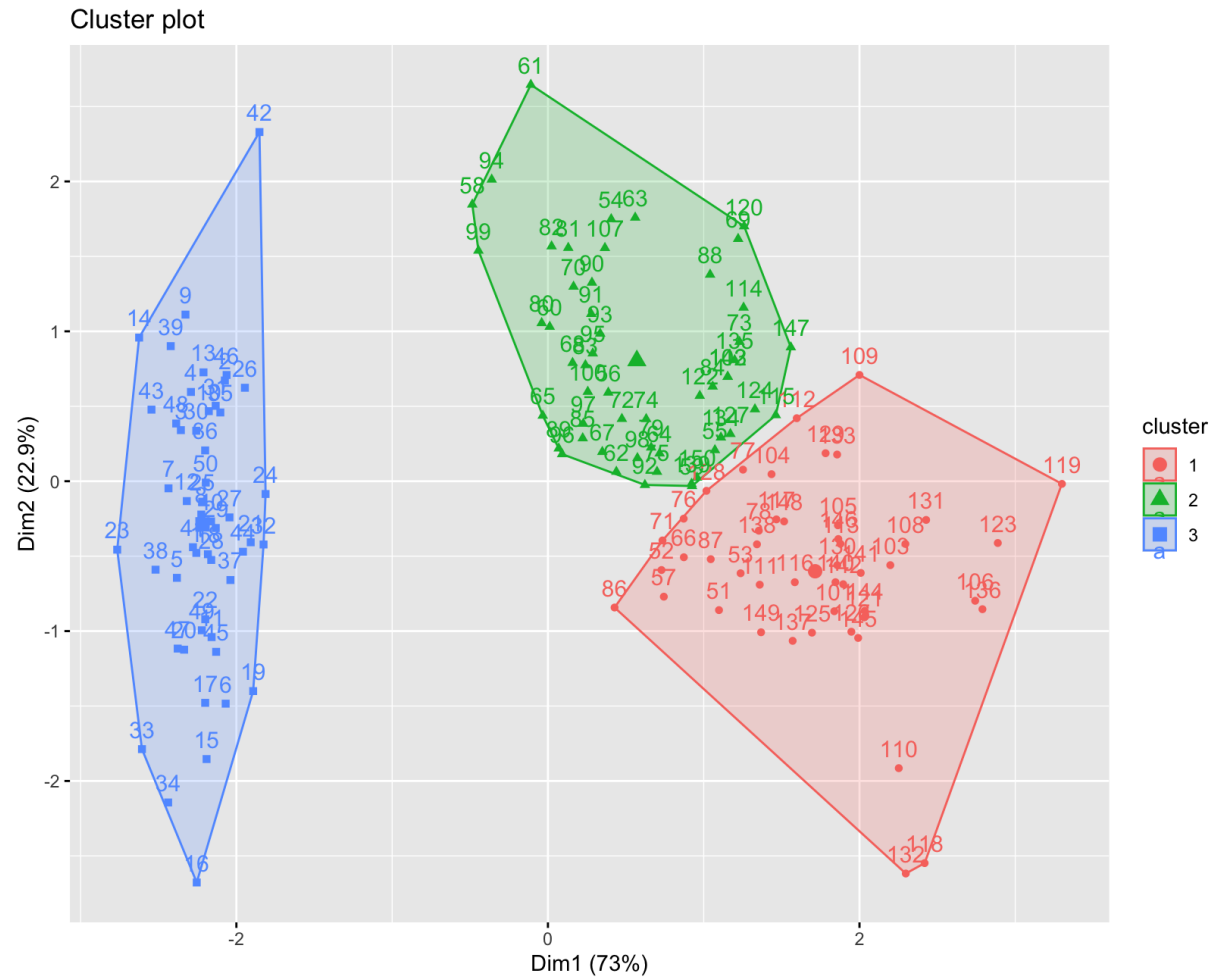
# VISUALIZACIÓN DE CLUSTERS

```
#Graficación de los clusters
install.packages("factoextra")
library(factoextra)
library(ggplot2) #lo usa fviz_cluster
fviz_cluster(kM,iris) #en factoextra
```



# ESCALADO DE LOS DATOS

```
#Escalaro datos para mejorar calculos
irisEscalaro <- scale(as.matrix(iris[, 1:4]))
kM <- kmeans(irisEscalaro,3)
fviz_cluster(kM,irisEscalaro)
```



# EJERCICIO INTEGRADOR

- Utiliza el dataset “mtcars” ya cargado en R
- Realiza:
  - Análisis de correlación
  - Análisis utilizando K-means
- Entrega
  - Código Fuente
  - Documento PDF que contenga:
    - Correlación
      - ¿Cuáles variables están correlacionadas?
      - ¿Qué implicaciones tiene?
    - K-means
      - ¿Cuántos grupos son mejores para representar la información: 2, 3 o 4?
  - Conclusión
    - Toma en cuenta los resultados de la correlación y de K-means