**Regresión Lineal Simple**

La **regresión lineal** es un **modelo muy simple** de **aprendizaje supervisado**. Es una herramienta útil para predecir una variable cuantitativa. IE: Tiempo de demora de un vuelo; precio de una propiedad.

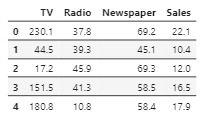
Ejemplo Práctico: Somos consultores estadísticos. Nos contratan para aumentar las ventas de un producto determinado. Tenemos el Dataset Advertising con las ventas del producto en 200 mercados (en miles de unidades) y el presupuesto dedicado en publicidad en 3 medios: TV, radio y diario (en miles de [USD). Buscamos identificar una relación entre la inversión en publicidad y las ventas para recomendarle a nuestro cliente hacia dónde debe dirigir su inversión en publicidad.

**En Python:**

advertising\_file\_path = ‘../Data/advertising.csv’

advertising = pd.read\_csv(advertising\_file\_path)

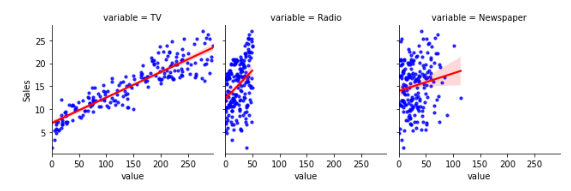
advertising.head(4)



advertising\_grid = pd.melt(advertising, id\_vars = “Sales”, value\_vars = [‘TV’, ‘Radio’, ‘Newspaper’])

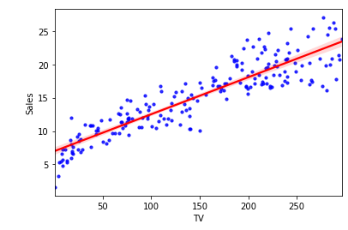
g = sns.FacetGrid(advertising\_grid, col = ‘variable’)

g.map(sns.regplot, ‘value’, ‘Sales’, ci = 95, scatter\_kws = {“color”: “blue”, ‘s’:10}, line\_kws = {‘color’:’red’})

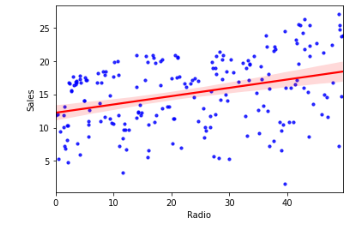


Mirando en Detalle:

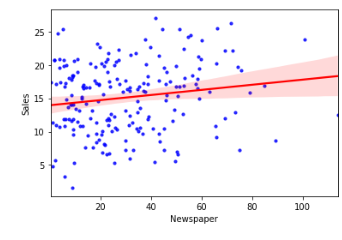
sns.regplot(data = advertising, x = “TV”, y = “Sales”, ci = 95, scatter\_kws = {“color”: “blue”, “s”: 10}, line\_kws = {“color”, “red”})



sns.regplot(data = advertising, x = “Radio”, y = “Sales”, ci = 95, scatter\_kws = {“color”: “blue”, “s”: 10}, line\_kws = {“color”, “red”})



sns.regplot(data = advertising, x = “Newspaper”, y = “Sales”, ci = 95, scatter\_kws = {“color”: “blue”, “s”: 10}, line\_kws = {“color”, “red”})



Ante este dataset nos podemos plantear preguntas como cuáles de los medios mencionados contribuyen a las ventas? Qué tan fuerte es esta relación? Con qué precisión podemos predecir ventas futuras? Esta relación es lineal?

**Regresión Lineal Simple:** Buscapredecir una respuesta cuantitativa **Y** a partir de una variable predictora **X**, asumiendo que hay *aproximadamente* una relación lineal entre **X** e **Y**.



β 0 y β1 son dos constantes que representan el intercepto y la pendiente en el modelo lineal. También llamadas **parámetros del modelo**. Una vez que entrenamos al modelo con un conjunto de entrenamiento, podremos, por ejemplo, predecir futuras ventas en función de la inversión en TV. Al entrenar el modelo lo hacemos calcular sus parámetros β 0 y β1.



El ˆ se usa para indicar que tenemos un valor estimado para un parámetro o coeficiente desconocido; o bien para identificar el valor predicho de la respuesta.

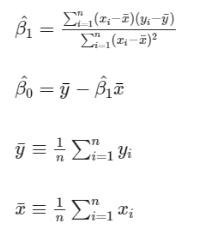
**β 0 Ordenada al origen**

**β1 Pendiente**

**Estimación de los Coeficientes:**

**Residuo o error de predicción:** 

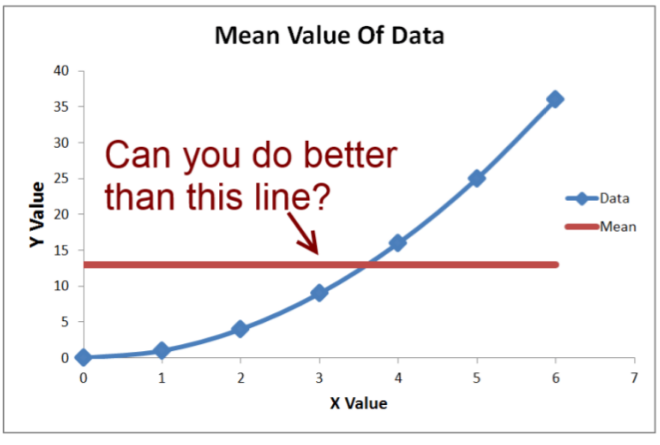
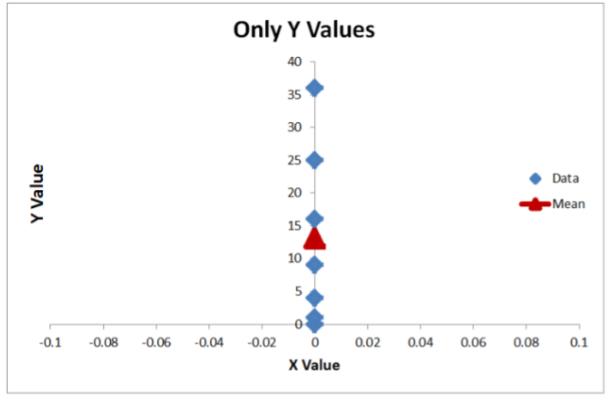
**Suma de cuadrados (Residual o Sum Squares): **

****

Volviendo al ejemplo, supongamos que el ajuste de una regresión lineal simple al dataset Advertising da **β 0 = 6.975 y β1 = 0.055**. Esto indica que por cada 1000$ gastados en publicidad em TV se venderían aproximadamente 55 unidades adicionales de producto.

**Evaluación del Modelo**

Vamos a suponer que nuestros datos son los puntos azules de este gráfico:

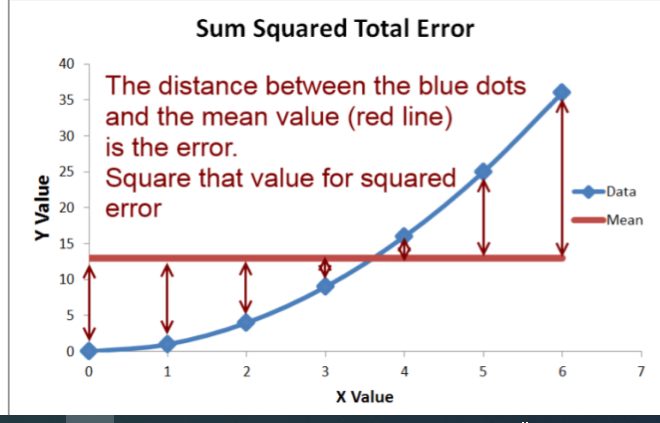
El segundo gráfico lo que indica es que si bien la media es representativa de los datos, en cierta manera al usar la media, se pierde la riqueza que puede aportar cada dato al modelo.

La media de los puntos azules (el triángulo rojo) es el valor que minimiza el error cuadrático.

Si tenemos la posibilidad de usar los valores de **X**, podemos hacer una mejor aproximación que la media de los puntos.

**R2: Total Sum of Squares (TSS)** es la **variabilidad total de los datos**.



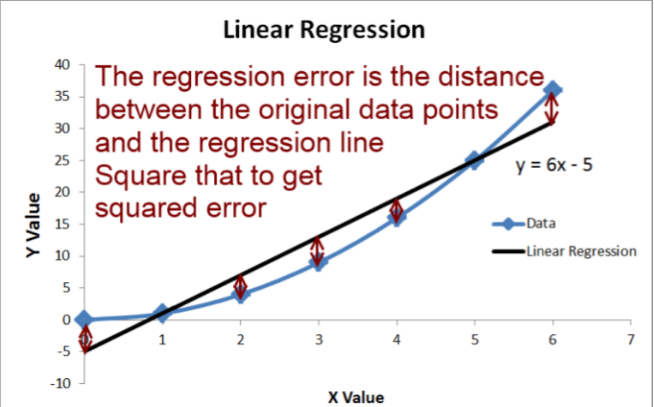


Esta métrica sólo está comparando las diferencias en el eje Y de los valores reales con respecto a la media Y, elevada al cuadrado.

Ahora en lugar de usar la media vamos a usar la regresión. Vamos a calcular el **error en los valores de la regresión vs los valores reales**. De esta forma obtenemos el **error de la regresión.** Idealmente será muy bajo, cercano a cero.

**Residual Sum of Squares (RSS):** Es la variabilidad no explicada por el modelo.

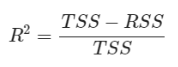




La división entre el error de la regresión y el error total (con respecto de la media) RSS/TSS nos indica qué **proporción del error total** **se mantiene en el modelo de regresión**.

Restar este cociente de 1 da la **proporción de error que se elimina** al usar una regresión.

**R2** es la **proporción de la variabilidad explicada por el modelo:**



**Si R2 tiende a cero** entonces el **modelo explica muy poco** sobre la variabilidad de los datos. La **proporción de error que elimina el modelo es muy baja**. Esto indica o bien que el modelo lineal está mal; o bien que el error inherente σ2 es muy alto; o las dos cosas.

Si **R2 tiende a uno** entonces el **modelo explica mucho sobre la variabilidad** de los datos. La **proporción de error que elimina el modelo es muy alta**.

R2 puede incluso ser negativo, en caso donde la regresión lineal performee peor que la media.

En definitiva, lo que está haciendo este ratio es una resta de errores: Al error de comparar los datos contra la media le resta el error de comparar los datos contra la regresión. Y a lo que da esta diferencia lo divide por el error de comparar los datos contra la media. Por eso es que si R2 está cerca de 1, lo que está pasando es que RSS está cerca de 0 y nuestra regresión está aproximando a los valores reales mucho mejor que la media.

Volviendo al ejemplo de Advertising, suponiendo que tenemos calculados los parámetros **β 0 = 6.975 y β1 = 0.055** en una recta que relaciona inversión en TV vs ventas.

**En Python:**

beta\_0 = 6.975

beta\_1 = 0.055

mean\_y = advertising.Sales.mean()

tss\_i = advertising.Sales.apply(lambda yi: (yi – mean\_y) \*\* 2)

tss = tss\_i.sum()

y\_hat\_i = beta\_o + beta\_1 \* advertising.TV

i\_count = advertising.shape[0]

rss\_i = [(advertising.Sales[i] – y\_hat\_i[i]) \*\* 2 for i in range(i\_count)]

rss = sum(rss\_i)

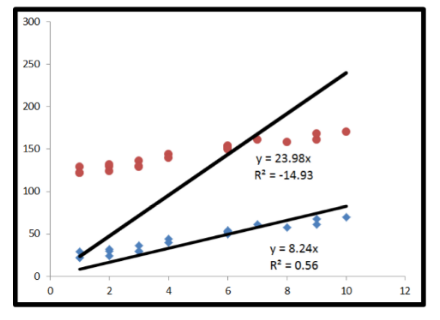
r2 = (tss - rss) / tss

print(“R2: “, np.round(r2, 3))

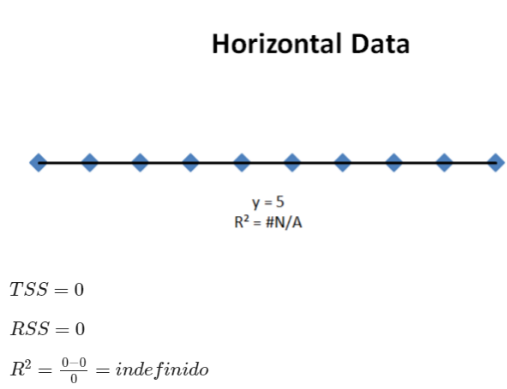


Más del 80% de la variabilidad en las ventas la está explicando una regresión lineal sobre TV.

**R2 puede dar negativo** en aquellos casos en los que la regresión sea peor que usar la media:



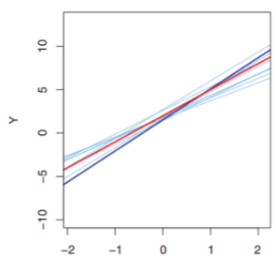
**R2 puede dar indefinido**



**Precisión de los Coeficientes Estimados**

Función de Regresión Poblacional y Muestral. “Essentially all models are wrong, but some of them are useful”.

Tenemos datos generados a partir de una función conocida y ruido aleatorio con media cero. Entonces podemos graficar una **verdadera relación**, llamada también **función de regresión poblacional**; y una o varias **estimaciones por mínimos cuadrados** estimadas a partir de submuestras de los datos. Cada línea es diferente, pero en promedio, todas están cerca de la función de regresión poblacional.

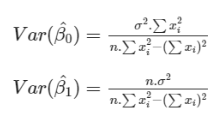


**Error Estándar de los Coeficientes:**

Dada la fórmula de regresión lineal agregándole un término que representa el error:

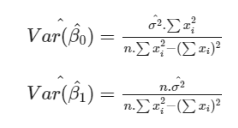


Queremos calcular las variabilidades de los coeficientes β:

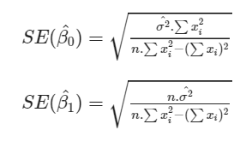


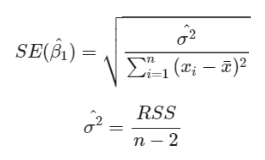
σ2 es la **varianza del error** y ε es desconocido. La estimamos así:





Si tomamos la raíz cuadrada obtendremos los **errores estándar de los coeficientes** de la estimación:





**Intervalo de Confianza para β1**

Tomando un intervalo de confianza del 95%, la expresión para β1 sería:



**Test de Hipótesis sobre los coeficientes estimados**: Hay evidencias para afirmar que hay relación entre **X**  e **Y**?

Hipótesis Nula: No hay relación entre X e Y



Hipótesis Alternativa: Hay alguna relación entre X e Y

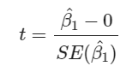


Si **β1** = 0; entonces Y = **β0 + ε**

Entonces **X** no estaría asociado a **Y**.

Por ende, para que haya relación entre **X** e **Y** necesitamos que **β1** esté lo suficientemente lejos de 0 como para que tengamos certeza de que no puede ser 0. Qué tan lejos de cero? Dependerá de la precisión de **β1**, **del error estándar de nuestro estimador del coeficiente**, el cual a su vez, dependerá, entre otros, del tamaño muestral **n**.

Se computa el **estadístico t** con **n-2 grados de libertad:**

****

Cuando tengamos más variables, los grados de libertad serán n – p -1. La cantidad de observaciones, menos la cantidad de parámetros que estemos estimando.

Vamos a calcular la probabilidad de observar cualquier valor igual a |t| o mayor asumiendo que β1 = 0 (***p-value***). Bajo H0, el estadístico de prueba es una variable con distribución T de Student, con n-p-1 grados de libertad. Un **p-value** pequeño indica que es poco probable observar un valor del estadístico como el observado o más pequeño asumiendo que H0 es verdadera. Con un **p-value chico**, podremos **rechazar H0 con baja probabilidad de equivocarnos**.

Entonces, cuando ajustamos una regresión lineal, es común reportar el error estándar de cada estimador .

Esto es útil para construir **intervalos de confianza** de los estimadores de los coeficientes; **evaluar la significatividad de cada estimador** con un test estadístico:

* Con p-value bajo, p < 0.05 o p < 0.01 es improbable observar al azar una observación semejante entre **X** e  **Y**.
* Con p-value alto, es probable que la asociación observada sea sólo consecuencia del azar.

Ejemplo: **en Python**

**import** statsmodel.api **as** sm

X\_t = np.array(advertising.TV, ndmin=2) # Convierte a X en una matriz de 2 dimensiones para que pueda funcionar con statsmodel, pero toma el shape de una fila con muchas columnas.

X = np.transpose (X\_t) # lo reacomoda a muchas filas con una columna.

y = advertising.Sales

# Tenemos que agregar explícitamente una constante

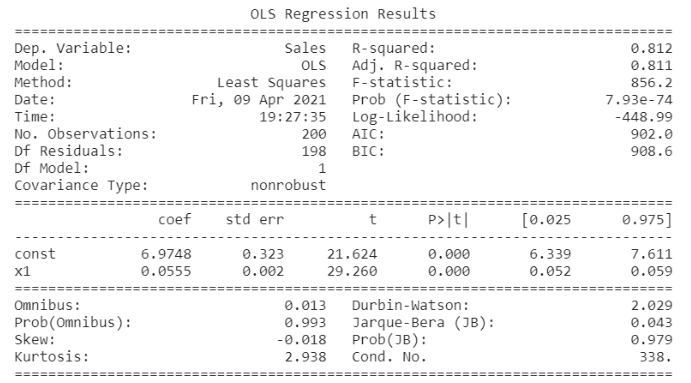
X = sm.add\_constant(X)

# Ahora ajusta el modelo a los datos.

model = sm. OLS(y, X).fit()

# Y nos da un resultado

print(model.summary())



# El pvalue chico nos indica que podemos rechazar la hipótesis nula y que X e y por lo tanto sí están relacionadas; o por lo menos, no tengo pruebas suficientes que me indiquen que no lo están.

Con el método **.summary()** podemos obtener, entre otros:

* El nombre de la variable dependiente y (**Dep. Variable**)
* Los grados de libertad (**Df Residuals**
* El valor de R2 (**R-squared**)
* Los valores de los coeficientes (β0: **const** y β1:**x1**)
* Los valores estadísticos de t y su respectivo p-value. Con ellos podemos aprobar o rechazar la hipótesis nula (en este caso, es 0 para ambos coeficientes; esto es estadísticamente significativo y por lo tanto, rechazamos las hipótesis nulas H0 : β0 = 0 y H0 : β1 = 0)
* El intervalo de confianza para los coeficientes, con un α de 95%.

**Supuestos de Gauss-Markov**

Son los supuestos en los que se basa el método de Regresión Lineal:

1 – El modelo es lineal en los parámetros



2 – Los estimadores de los parámetros poblacionales se estiman a partir de una muestra aleatoria.

3 – No hay colinealidad perfecta entre variables explicativas.

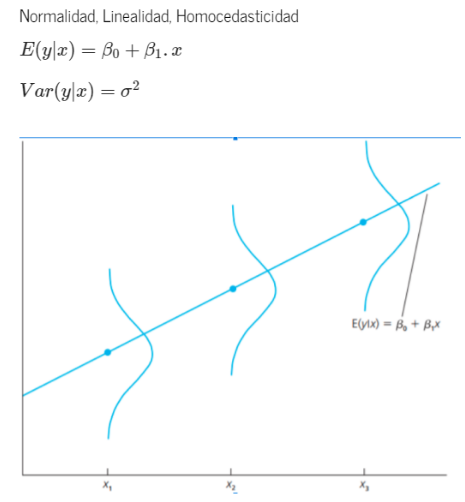
4 – El valor esperado del error es 0 para cualquier valor de la variable explicativa.

5 – Homocedasticidad. Para cualquier valor de la variable explicativa, el error tiene la misma varianza.

6 – Normalidad de los errores. Para cada valor x, de la variable predictora X, la variable respuesta Y debe tener distribución Normal. 

7 - Independencia de los errores. Cuando dos variables son independientes, su correlación es cero. En general la recíproca no es cierta, pero bajo el supuesto de normalidad, el supuesto de independencia de los errores se reduce a que no estén correlacionados.



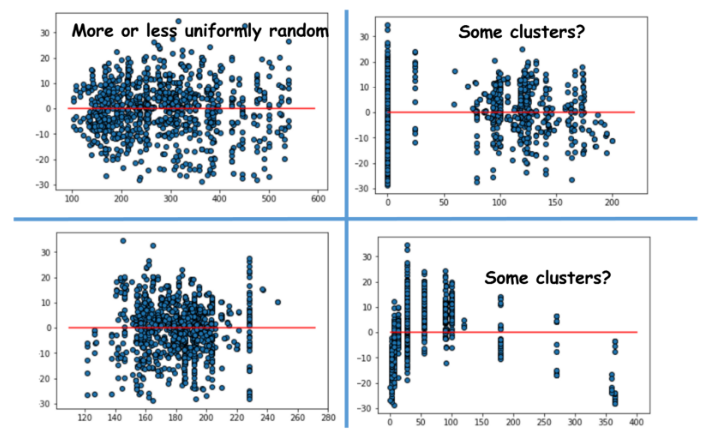


**Análisis de Residuos**

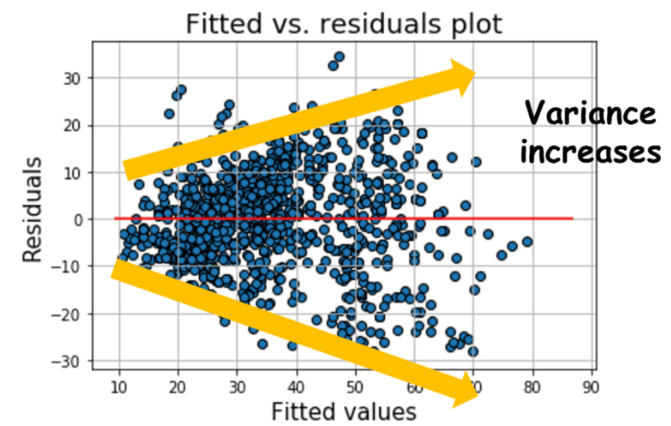
Usando los datos y los residuos de los datos con respecto a una recta ajustada pueden validarse los supuestos de Gauss-Markov:

Podemos conseguir una impresión sobre los supuestos de linealidead y homocedasticidad con un diagrama de dispersión. Con el análisis de residuos podemos confirmar dicha impresión inicial y también validar los supuestos de normalidad e independencia.

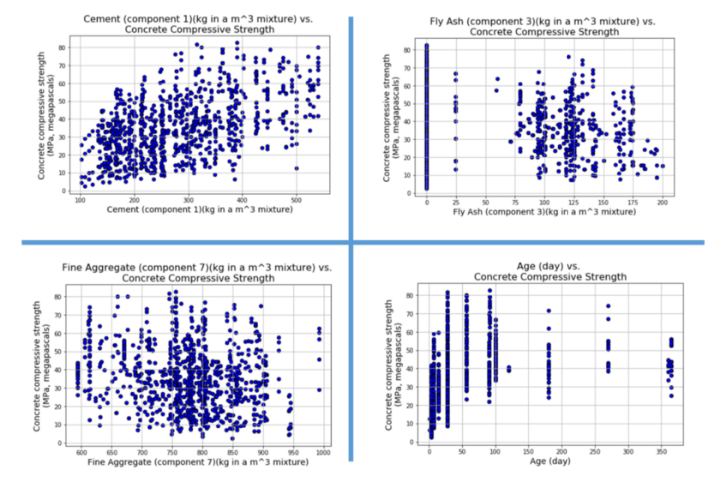
Si graficamos los residuos contra cada una de las variables predictoras, los puntos deberían estar distribuidos en forma de nube alrededor del valor 0 del residuo y no formar clusters. Si logramos esto, podemos considerar como válido el supuesto de independencia de los errores. En este ejemplo se ven algunos clusters:



Acá tenemos una figura donde no se satisface el supuesto de homocedasticidad: Los residuos tienen variabilidad creciente a medida que crece el valor de los predichos:



Con un Scatterplot podemos verificar si se cumple o no con el supuesto de linealidad:

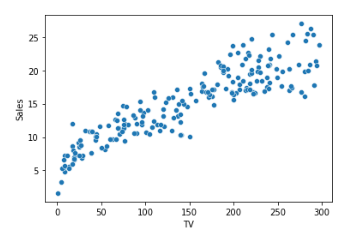


Si se satisfacen los supuestos de **normalidad de los errores**, **linealidad**, **homocedasticidad** e **independencia de los errores**, entonces los errores no están relacionados y tienen una distribución normal con media cero y varianza constante.

Ejemplo práctico **en Python**

1. **El modelo es lineal en los parámetros **

sns.scatterplot(data = advertising, x = “TV”, y = “Sales”)



1. **Los estimadores de los parámetros poblacionales se estiman a partir de una muestra aleatoria**: Vamos a suponer que las observaciones del dataset provienen de una muestra aleatoria de la población.
2. **No hay colinealidad perfecta entre las variables explicativas**:Comola regresión lineal simple cuenta sólo con una variable explicativa, no hay posibilidad de colinealidad con ninguna otra variable.
3. **El valor esperado del error es 0 para cualquier valor de la variable explicativa**:

X\_t = np.array(advertising.TV, ndmin = 2)

X = np.transpose(X\_t)

y = advertising.Sales

#Tenemos que agregar explícitamente una constante:

X = sm.add\_constant(X)

model = sm.OLS(y, X).fit()

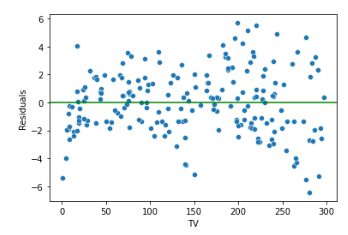
residuals = model.resid

p = sns.scatterplot(data = advertising, x = “TV”, y = residuals)

p.axhline(0, color = “green”)

p.set(ylabel = “Residuals”)

#otra opción: sns.residplot(data = advertising, x = “TV”, y = “Sales”)

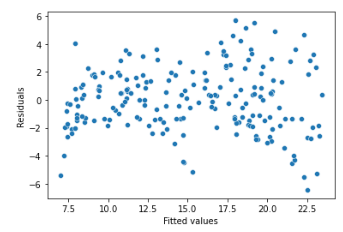


1. **Homocedasticidad:** El error tiene la misma varianza para cada valor de la variable explicativa:

y\_hat = model.predict(X)

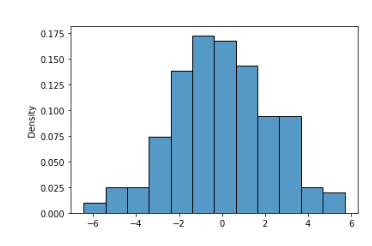
p = sns.scatterplot(x = y\_hat, y = residuals)

p.set(xlabel = “Fitted values”, ylabel = “Residuals”)

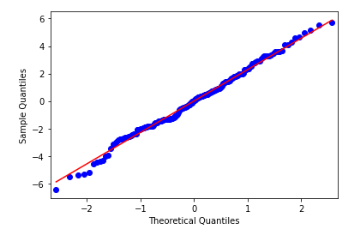


1. **Normalidad de los Errores:** para cada valor x de la variable predictora X, la variable respuesta Y debe tener distribución Normal: 

Con el método **ProbPlot** podemos graficar un QQplot para testear la normalidad de los residuos ProbPlot:

sns.histplot(residuals, kde = False, stat = “density”, line\_kws = {‘linewidth’:5}) 

sm.ProbPlot(model.resid).qqplot(line=’s’)



1. **Independencia de los errores:** Cuando la correlación de dos variables independientes es cero, en general la recíproca no es cierta. Bajo el supuesto de normalidad, el supuesto de independencia de los errores se reduce a que no estén correlacionados:

****

**Conclusiones**

El modelo de Regresión Lineal:

* Nos permite **relacionar linealmente** una **variable explicativa** con una **variable objetivo cuantitativa**.
* Podemos **determinar la significancia estadística** de cada **coeficiente encontrado.**
* **R2** es una **medida de la variabilidad** explicada por el **modelo.**
* Al haber estimado el modelo, verificamos si se cumplen o no los supuestos de Gauss-Markov. Si los cumplen, entonces β1 es el mejor estimador lineal insesgado.