

JULIHO COLMENARES PH.D.

# ESTADÍSTICA MATEMÁTICA

WWW.ASIMOVIAN.ACADEMY  
MATH.VADER@YAHOO.COM

This work is licensed under the Creative Commons Reconocimiento 4.0 Internacional License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.



# Índice general

1	<i>Estadística descriptiva</i>	5
1.1	<i>Medidas de tendencia central</i>	5
1.2	<i>Medidas de dispersión</i>	11
1.3	<i>Correlación</i>	14
2	<i>Probabilidad básica</i>	21
2.1	<i>Fundamentos de probabilidad</i>	21
2.2	<i>Probabilidad condicional</i>	26
2.3	<i>Análisis combinatorio</i>	28
3	<i>Variables Aleatorias Discretas</i>	31
3.1	<i>Funciones de probabilidad discretas</i>	31
3.2	<i>Funciones de distribución para variables aleatorias discretas</i>	32
4	<i>Variable Aleatorias Continuas</i>	35
4.1	<i>Interpretación gráfica</i>	37
4.2	<i>Distribución conjunta de probabilidad</i>	37
4.3	<i>Variables Aleatorias Independientes</i>	40
4.4	<i>Distribución Condicional</i>	42
5	<i>Esperanza Matemática</i>	45
5.1	<i>Definición de Esperanza Matemática</i>	45
5.2	<i>Varianza y Desviación Estándar</i>	48
5.3	<i>Covarianza y correlación</i>	50

6	<i>Distribuciones especiales</i>	53
6.1	<i>La Distribución Binomial</i>	53
6.2	<i>Distribución Normal</i>	55
6.3	<i>Relación entre las distribuciones binomial y normal</i>	59
6.4	<i>La Distribución de Poisson</i>	60
6.5	<i>Relación entre las Distribuciones Binomiales y de Poisson</i>	61
6.6	<i>Distribución multinomial</i>	63
6.7	<i>Problemas Resueltos</i>	64
7	<i>Pruebas de Hipótesis</i>	69
7.1	<i>Muestreo aleatorio y teorema del límite central</i>	69
7.2	<i>Pruebas de hipótesis</i>	70
7.3	<i>Estadísticos <math>Z</math> y <math>t</math></i>	70
7.4	<i>Intervalos de confianza, niveles de significación y valores <math>p</math></i>	73
7.5	<i>Una guía paso a paso para realizar una prueba de hipótesis</i>	76
7.6	<i>Un ejemplo de la prueba de hipótesis</i>	76
7.7	<i>Prueba <math>\chi</math>—cuadrada</i>	78
8	<i>Regresiones lineales</i>	83
8.1	<i>Entendiendo las matemáticas detrás de la regresión lineal</i>	84
8.2	<i>Regresión lineal usando datos simulados</i>	85
8.3	<i>Encontrando el valor optimo de los coeficientes de una regresión lineal</i>	87
8.4	<i>Implementando regresiones lineales con <b>Python</b></i>	90
8.5	<i>Regresión lineal múltiple</i>	93
8.6	<i>Validación del modelo</i>	99
8.7	<i>Resumen de modelos</i>	102
8.8	<i>Regresión lineal con <b>scikit-learn</b></i>	102
8.9	<i>Manejando otros Problemas en lineales regresión</i>	104

# 1 Estadística descriptiva

## 1.1 Medidas de tendencia central

**Índice y subíndices** El símbolo  $X_j$  representa cualquiera de los valores  $X_1, X_2, X_3, \dots$  que puede tomar la variable discreta  $X$ .

El símbolo  $j$  denota cualquiera de los números naturales  $1, 2, 3, \dots$  y se le llama *índice* (o a veces *subíndice* o también *contador*).

**Definición 1.1** (Sumatoria).

$$\sum_{j=1}^N X_j = X_1 + \dots + X_N \quad (1.1)$$

**Ejemplo 1.1.** ■  $\sum_{k=1}^N X_k Y_k = X_1 Y_1 + \dots + X_N Y_N$

■  $\sum_{i=1}^N aX_i = aX_1 + \dots + aX_N = a \sum_{n=1}^N X_n.$

**Observación 1.1.** Cuando se *sobrentiende* que el contador  $j$  corre sobre los números  $1, 2, \dots, N$ , escribimos  $\sum X_j$  o simplemente  $\sum X$  en lugar de  $\sum_{j=1}^N$ .

*Linealidad*

**Problema 1.1.** Si  $a, b$  son constantes, demuestre que

$$\sum (aX + bY) = a \sum X + b \sum Y. \quad (1.2)$$

**Promedio** UN *promedio* es un valor representativo de un conjunto de datos que tiende a encontrarse en el centro de dicho conjunto.

Por esta razón, también se le conoce como *medidas de tendencia central*.

Se pueden definir varios tipo de promedios:

- Media aritmética;
- mediana;

- moda;
- media geométrica;
- media armónica.

*Observación 1.2.* Cada medida de tendencia central tiene ventajas y desventajas de acuerdo al tipo de datos y el propósito del uso.

**Definición 1.2** (Media aritmética).

$$\bar{X} = \frac{X_1 + \dots + X_N}{N} = \frac{\sum_{j=1}^N X_j}{N} = \frac{\sum X}{N} \quad (1.3)$$

**Ejemplo 1.2.** La media aritmética de 8, 3, 5, 12, 10 es...

Si los números  $X_1, X_2, \dots, X_k$  se presentan con *frecuencias*  $f_1, f_2, \dots, f_k$  respectivamente su media aritmética es

$$\bar{X} = \frac{f_1 X_1 + \dots + f_k X_k}{f_1 + \dots + f_k} = \frac{\sum fX}{\sum f} = \frac{\sum fX}{N}. \quad (1.4)$$

dónde  $N = \sum f$  es la *suma de frecuencias* o *total de casos*.

**Ejemplo 1.3.** Si 5, 8, 6, 2 se presentan con frecuencias 3, 2, 4, 1 respectivamente, su media aritmética es...

*Media aritmética ponderada* Algunas veces, a los números  $X_1, \dots, X_k$  se les asignan ciertos *factores de ponderación* o *pesos*  $w_1, \dots, w_k$ , tales que

$$\begin{cases} 0\% \leq w_i \leq 100\% \\ \sum w_i = 100\% \end{cases} \quad (1.5)$$

**Definición 1.3** (Media ponderada). Si  $w_1, \dots, w_k$  son *pesos* tales que  $0 \leq w_i \leq 1$  y  $\sum w_i = 1$ , entonces la correspondiente media (aritmética) ponderada de los números  $X_1, \dots, X_k$  es

$$\bar{X} = \frac{w_1 X_1 + \dots + w_k X_k}{w_1 + \dots + w_k} = \frac{\sum wX}{\sum w} = \sum wX. \quad (1.6)$$

**Ejemplo 1.4.** Si en una clase, al examen final se le da el triple del valor que a los exámenes parciales y un estudiante obtiene 85 en el final y 70 y 90 en los dos exámenes parciales, obtener su media ponderada.

1. Si  $w_i = \frac{1}{N}$ , obtenemos la media aritmética usual.
2. Si  $w_i = \frac{f_i}{N}$ , obtenemos la fórmula (1.4).

*Datos agrupados*

Cuando los números son muy grandes, se suele utilizar un pivote  $P$ :

$$\bar{X} = P + \frac{\sum f_i d_i}{N},$$

donde  $d_i = X_i - P$ .

En ocasiones, utilizaremos la notación

$$\bar{d} = \frac{\sum f_i d_i}{N}, \quad (1.7)$$

de manera que  $\bar{d}$  es la *desviación promedio* y  $\bar{X} = P + \bar{d}$ .

*Observación 1.3.* Para datos agrupados,  $X_i$  se escoge como la marca de la  $i$ -ésima clase.

*La mediana* La mediana  $\tilde{X}$  de un conjunto de números acomodados en un orden de magnitud (es decir, en una ordenación) es el valor central o la media de dos valores centrales.

**Ejemplo 1.5.** ■ La mediana de la lista de números 3, 4, 5, 6, 8, 8, 8, 10 es...

■ La mediana de la lista de números 5, 5, 7, 9, 11, 12, 15, 18 es..

**Definición 1.4** (Mediana para datos agrupados).

$$\text{Mediana} = L + \left( \frac{\frac{N}{2} - \sum_{C < C_M} f}{f_{C_M}} \right) \quad (1.8)$$

donde

- $L$  es la frontera inferior de la clase mediana, es decir, de la clase que contiene la mediana;
- $N$  es la frecuencia total;
- $\sum_{C < C_M} f$  suma de las frecuencias de todas las clases anteriores a la clase mediana;
- $f_{C_M}$  es la frecuencia de la clase mediana.

*Moda* La moda de una lista de números es un valor que se presenta con la mayor frecuencia  $f > 1$ . *La moda no es necesariamente existe ni es única.*

**Ejemplo 1.6.** ■ La moda de la lista 2, 2, 5, 7, 9, 9, 9, 10, 10, 11, 12, 18 es... En este caso, diremos que la lista es *unimodal*.

■ ¿Cuál es la moda de la lista 3, 5, 8, 0, 12, 15, 16?

- ¿Cuál es la moda de la lista 3, 8, 8, 8, 15, 15, 15? En este caso diremos que la lista es *bimodal*.

**Definición 1.5** (Moda para datos agrupados).

$$\text{Moda} = L + \left( \frac{\Delta_1}{\Delta_1 + \Delta_2} \right) c \quad (1.9)$$

donde

1.  $L$  : Frontera inferior de la clase modal, es decir, de la clase que contiene la moda.
2.  $\Delta_1$  : Exceso de frecuencia modal sobre la frecuencia en la clase inferior inmediata.
3.  $\Delta_2$  : Exceso de frecuencia modal sobre la frecuencia en la clase superior inmediata.
4.  $c$  : Amplitud del intervalo de la clase modal.

### Python

```
1 numpy.mean(a, axis=None, dtype=None, out=None,
2 keepdims=<class numpy._globals._NoValue>)
```

Calcula la media aritmética sobre los elementos de un arreglo.<sup>1</sup>

[,]Ejemplos

```
1 a = np.array([[1, 2], [3, 4]])
2 print np.mean(a)
3 #2.5
4 print np.mean(a, axis=0)
5 #array([ 2.,  3.])
6 print np.mean(a, axis=1)
7 #array([ 1.5,  3.5])
```

[,]numpy.median

```
1 numpy.median(a, axis=None, out=None,
2 overwrite_input=False, keepdims=False)
```

Calcula la mediana de los elementos de un arreglo de números.<sup>2</sup>

[,]Ejemplos

```
1 import numpy as np
2
3 a = np.array([[10, 7, 4], [3, 2, 1]])
4 print a
5 #array([[10,  7,  4], [ 3,  2,  1]])
6 print np.median(a)
7 #3.5
8 print np.median(a, axis=0)
9 #array([ 6.5,  4.5,  2.5])
10 print np.median(a, axis=1)
11 #array([ 7.,  2.])
12
13 m = np.median(a, axis=0)
```

<sup>1</sup> <https://github.com/numpy/numpy/blob/v1.13.0/numpy/L2909>

<sup>2</sup> <https://docs.scipy.org/doc/numpy/reference/generators>



```

14 out = np.zeros_like(m)
15 print np.median(a, axis=0, out=m)
16 #array([ 6.5,  4.5,  2.5])
17 print m
18 #array([ 6.5,  4.5,  2.5])
19 b = a.copy()
20 print np.median(b, axis=1, overwrite_input=True)
21 #array([ 7.,  2.])
22
23 assert not np.all(a==b)
24 b = a.copy()
25 print np.median(b, axis=None, overwrite_input=True)
26 #3.5
27 assert not np.all(a==b)

```

*SciPy* SciPy es una biblioteca open source de herramientas y algoritmos matemáticos para Python...

SciPy contiene módulos para optimización, álgebra lineal, integración, interpolación, funciones especiales, FFT, procesamiento de señales y de imagen, resolución de ODEs y otras tareas para la ciencia e ingeniería. Está dirigida al mismo tipo de usuarios que los de aplicaciones como MATLAB, GNU Octave, y Scilab.<sup>3</sup>

<sup>3</sup> <https://es.wikipedia.org/wiki/SciPy>

[, ]Moda

```

1 import numpy as np
2 from scipy import stats
3
4 a = np.array([3,5,6,5,6,5,6,6,3,1,5])
5 print stats.mode(a)
6 # ModeResult(mode=array([5]), count=array([4]))
7
8 b = np.array([[6, 8, 0, 0],
9               [3, 3, 0, 3],
10              [8, 1, 8, 5],
11              [5, 3, 0, 5],
12              [4, 7, 5, 3]])
13
14 print stats.mode(b)
15 # ModeResult(mode=array([[3, 3, 0, 3]]),
16 #             count=array([[1, 2, 3, 2]]))
17
18 print stats.mode(b, axis=1)
19 # ModeResult(mode=array([[0], [3], [8], [5], [3]]),
20 #             count=array([[2], [3], [2], [2], [1]]))
21
22 print stats.mode(b, axis=None)
23 # ModeResult(mode=array([3]), count=array([5]))

```

**Problema 1.2.** Escribir los términos de cada una de las siguientes sumas:

(a)  $\sum_{j=0}^6 X_j =$

(b)  $\sum_{k=1}^4 (Y_k - 3)^2 =$

$$(c) \sum_{k=1}^N a =$$

$$(d) \sum_{n=2}^5 f_n X_n =$$

$$(e) \sum_{m=0}^3 (X_m - a) =$$

**Problema 1.3.** De 100 números, 20 fueron 4, 40 fueron 5, 30 fueron 6 y los restantes fueron 7. Encuentre su media aritmética.

**Problema 1.4.** Los pesos medio de cuatro grupos de estudiantes que constan de 15, 20, 10 y 18 individuos son 162, 148, 153 y 140 libras, respectivamente. Encuentre el peso medio de todos los estudiantes.

**Problema 1.5.** Usando la distribución de frecuencias de las estaturas que se presenta en la siguiente tabla, hallar la estatura media de 100 estudiantes de cierta universidad.

Estatura (in)	Marcas de clase ( $X$ )	Frecuencias ( $f$ )	$fX$
60-62	61	5	305
63-65	64	18	1 152
66-68	67	42	2 814
69-71	70	27	1 890
72-74	73	8	584

**Problema 1.6.** Si las desviaciones de  $N$  números  $X_1, \dots, X_N$  respecto a un *pivote*  $P$  están dada por  $d_i = X_i - P$ ,  $i = 1, \dots, N$  respectivamente, demostrar que

$$\bar{X} = P + \frac{\sum d}{N}. \quad (1.10)$$

**Problema 1.7.** Demostrar que la suma de las desviaciones  $d_1, d_2, \dots, d_N$  de  $X_1, X_2, \dots, X_N$  usando como pivote su media  $\bar{X}$  es igual a cero.

**Problema 1.8.** Si  $Z_i = X_i + Y_i$ ,  $i = 1, 2, \dots, N$ , demostrar que  $\bar{Z} = \bar{X} + \bar{Y}$ .

**Problema 1.9.** Halle la media aritmética de los números 5, 8, 11, 9, 12, 6, 14 y 10 eligiendo como *pivote* a)  $P = 9$  y b)  $P = 20$ .

**Problema 1.10.** Utilice la marca de la clase media como pivote, para calcular la estatura de los estudiantes en la tabla 1.5.

**Problema 1.11.** Encontrar el peso mediano a partir de la siguiente tabla

Peso (lb)	Frecuencias
118-126	3
127-135	5
136-144	9
145-153	12
154-162	5
163-171	4
172-180	2

## 1.2 Medidas de dispersión

*Dispersión o variación* Si bien las medidas de tendencia central nos dicen alrededor de que valores se concentra un arreglo de datos, las *medidas de dispersión* nos dan una idea de que tan alejados están entre sí.

A continuación, veremos algunas medidas de dispersión comúnmente usadas en estadística.

*Rango* El *rango* de un conjunto de datos es la diferencia entre el mayor y el menor del conjunto.

**Ejemplo 1.7.** El rango del conjunto 2,3,3,5,5,5,8,10,12 es  $12 - 2 = 10$ .

*Desviación media* La *desviación media* o *desviación promedio* de un conjunto de  $N$  números  $X_1, \dots, X_N$  está definida como

$$DM = \frac{\sum |X_j - \bar{X}|}{N} \quad (1.11)$$

donde  $\bar{X}$  es la media aritmética de los números y  $|\cdot|$  denota el valor absoluto.

**Ejemplo 1.8.** Encuentre la desviación media de la lista 2, 3, 6, 8, 11.

*Desviación estándar* La *desviación estándar* de un conjunto de  $N$  números  $X_1, \dots, X_N$  se denota como  $s$  y está definida por

$$s = \sqrt{\frac{\sum (X_j - \bar{X})^2}{N}} = \sqrt{\sum \frac{x_j^2}{N}} \quad (1.12)$$

donde  $x_j := X_j - \bar{X}$ .

Si  $X_1, \dots, X_N$  se presentan con frecuencias  $f_1, \dots, f_N$  respectivamente, la desviación estándar se puede expresar como

$$s = \sqrt{\frac{\sum f_j (X_j - \bar{X})^2}{N}} = \sqrt{\frac{\sum f_j x_j^2}{N}} \quad (1.13)$$

*Observación 1.4.* En ocasiones,  $N$  se reemplaza por  $N - 1$  en las fórmulas anteriores, debido a que esta definición aproxima mejor a la población de la que se ha obtenido la muestra. Pero para muestras muy grandes  $N > 30$  prácticamente no hay diferencia.

*Varianza* La *varianza* de un conjunto de números se define como el cuadrado  $s^2$  de la desviación estándar  $s$ .

*Observación 1.5.* En estadística, es importante distinguir entre la desviación estándar de una *población* y una *muestra*. Para distinguirla, en el primer caso utilizaremos  $\sigma$  y en el segundo, continuaremos usando  $s$ .

*Método abreviado*

$$s^2 = \overline{X^2} - \bar{X}^2 \quad (1.14)$$

$$= E(X^2) - (E(X))^2 \quad (1.15)$$

En las distribuciones normales se tiene que

- (a) 68.27 % de los datos está comprendido entre  $\bar{X} \pm s$ .
- (b) 95.45 % de los datos está comprendido entre  $\bar{X} \pm 2s$ .
- (c) 99.73 % de los datos está comprendido entre  $\bar{X} \pm 3s$ .

Si 2 conjuntos de  $N_1$  y  $N_2$  datos respectivamente tienen correspondientes  $s_1^2$  y  $s_2^2$  varianzas pero una misma media aritmética  $\bar{X}$ , entonces la varianza de la unión de ambos conjuntos es

$$s^2 = \frac{N_1 s_1^2 + N_2 s_2^2}{N_1 + N_2}. \quad (1.16)$$

*Teorema de Chebyshev* Para  $k > 1$ , por lo menos  $1 - \frac{1}{k^2}$  de la distribución de probabilidad de cualquier variable aleatoria está a no más de  $k$  desviaciones estándar de la media.

## Python

`numpy.std`

```
1 numpy.std(a, axis=None, dtype=None, out=None, ddof=0,
2 keepdims=<class numpy._globals._NoValue>)
```

Calcule la desviación estándar a lo largo del eje especificado.

Devuelve la desviación estándar, una medida de la propagación de una distribución, de los elementos de la matriz. La desviación estándar se calcula para la matriz aplanada de forma predeterminada, de lo contrario sobre el eje especificado.<sup>4</sup>

□

<sup>4</sup> <https://docs.scipy.org/doc/numpy/reference/generated/>

```

1 import numpy as np
2
3 a = np.array([[1, 2], [3, 4]])
4 print np.std(a)
5 #1.1180339887498949
6 print np.std(a, axis=0)
7 #array([ 1.,  1.])
8 print np.std(a, axis=1)
9 #array([ 0.5,  0.5])

```

□

```

1 #In single precision, std() can be inaccurate:
2 a = np.zeros((2, 512*512), dtype=np.float32)
3 a[0, :] = 1.0
4 a[1, :] = 0.1
5 print np.std(a)
6 #0.45000005
7
8 #Computing the standard deviation in float64
9 #is more accurate:
10 print np.std(a, dtype=np.float64)
11 #0.44999999925494177

```

**Problema 1.12.** Encontrar el rango y las desviaciones media y estándar de los arreglos

(a) 12, 6, 7, 3, 15, 10, 18, 5

(b) 9, 3, 8, 8, 9, 8, 9, 18.

Compruebe sus resultados con Python.

**Problema 1.13.** Encontrar las desviaciones media y estándar de las estaturas de 100 estudiantes de la siguiente tabla:

Marcas de clase ( $X$ )	Desviación $d = X - A$	Frecuencias ( $f$ )
61	-6	5
64	-3	18
$A \rightarrow 67$	0	42
70	3	27
73	6	8

**Problema 1.14.** Encontrar las desviaciones media y estándar de las estaturas de 100 estudiantes de la siguiente tabla:

**Problema 1.15.** Demostrar que

$$s = \sqrt{\frac{\sum X^2}{N} - \left(\frac{\sum X}{N}\right)^2} = \sqrt{\overline{X^2} - \bar{X}^2} \quad (1.17)$$

$$s = \sqrt{\frac{\sum fX^2}{N} - \left(\frac{\sum fX}{N}\right)^2} = \sqrt{\overline{X^2} - \bar{X}^2} \quad (1.18)$$

Estatura (in)	Cantidad de estudiantes
60-62	5
63-65	18
66-68	42
69-71	27
72-74	8

**Problema 1.16.** Utilizando las fórmulas anteriores, encuentre la desviación estándar de los datos en la tabla 1.14:

Estatura (in)	Cantidad de estudiantes
60-62	5
63-65	18
66-68	42
69-71	27
72-74	8

**Problema 1.17.** Si  $d = X - P$  son desviaciones de  $X$  respecto a un pivote  $P$ , demostrar que

$$s = \sqrt{\frac{\sum fd^2}{N} - \left(\frac{\sum fd}{N}\right)^2}. \quad (1.19)$$

**Problema 1.18.** Utilizando las fórmulas anteriores, encuentre la desviación estándar de los datos en la tabla 1.14:

**Problema 1.19.** Encuentre la media aritmética y la desviación estándar de los siguientes datos:

### 1.3 *Correlación*

La correlación es la premisa de la modelación predictiva, en el sentido de que es un factor con base en el cuál podemos predecir resultados.

Una buena correlación entre dos variables sugiere que existe una suerte de dependencia entre ambas: Si una cambia, la otra también lo hará.

Estatura (in)	Cantidad de estudiantes
60-62	5
63-65	18
66-68	42
69-71	27
72-74	8

Salarios	Número de empleados
\$250.00-\$259.99	8
\$260.00-\$269.99	10
\$270.00-\$279.99	16
\$280.00-\$289.99	14
\$290.00-\$299.99	10
\$300.00-\$309.99	5
\$310.00-\$319.99	2

Podemos decir que una buena correlación asegura una relación matemática entre dos variables y debido a esto, seremos capaces de predecir su comportamiento.

La relación puede ser de cualquier tipo. Si  $x$  y  $y$  son dos variables correlacionadas, entonces podemos escribir:

$$Y = f(X) \quad (1.20)$$

#### *Ejemplos de correlación*

##### ■ Correlación lineal

$$y = ax + b \quad (1.21)$$

##### ■ Correlación exponencial

$$y = be^{ax} \quad (1.22)$$

*Definición de correlación* Si  $X, Y$  son dos variables aleatorias discretas, que pueden tomar valores  $\{x_0, x_1, \dots\}$  y  $\{y_0, y_1, \dots\}$ , con promedios  $\bar{x}, \bar{y}$  respectivamente, su correlación se define como

$$\rho(X, Y) = \frac{\sum_{x_i, y_j} ((x_i - \bar{x})(y_j - \bar{y}))}{\sqrt{\sum_{x_i, y_j} (x_i - \bar{x})^2 \sum_{x_i, y_j} (y_i - \bar{y})^2}} \quad (1.23)$$

*Propiedades de la correlación*

- El valor del coeficiente de correlación está entre  $-1$  y  $1$ , es decir  $-1 \leq \rho(X, Y) \leq 1$ .
- Una correlación positiva significa una relación directa entre las dos variables.
- Una correlación negativa significa una relación inversa entre las dos variables.
- Entre mayor sea la magnitud del coeficiente, más fuerte será la relación entre variables.

*¡Advertencia!* Aunque una correlación fuerte sugiere algún tipo de relación que puede ser utilizada para la predicción del comportamiento de una variable respecto de otra, *esto no implica que dicha relación sea el único factor que explique dicho comportamiento.*

*Observación 1.6. ¡Correlación no implica causalidad!*

Por ejemplo, existe una correlación positiva muy fuerte entre el número de palabras que conoce un ser humano y el número de calzado que utiliza... Pero esto se explica porque a medida que el ser humano crece, necesita zapatos más grandes y aprende palabras nuevas.

*¡No quiere decir que si utilizas un número más grande serás más culto!*

Tratemos de entender mejor este concepto mirando una base de datos y tratando de encontrar una correlación entre sus variables. La base de datos que estaremos observando es una muy popular sobre los costos varios incurridos en *publicidad por diferentes medios y ventas de un producto en particular.*

Posteriormente, utilizaremos el método conocido como *regresión lineal* para explorar estos mismo datos.

Por ahora, importemos esta base de datos y calculemos los coeficientes de correlación:

[,]advertising.py

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import pandas as pd
5
6  advert = pd.read_csv("./dataBases/Advertising.csv")
7  print(advert.head())
8
9  import numpy as np
10
11  advert["dX*dY"] = ( advert["TV"] -
12    np.mean(advert["TV"])) * ( advert["Sales"]
13    np.mean(advert["Sales"]))
14  advert["dX**2"] = ( advert["TV"] -

```



```

15 np.mean(advert["TV"])) ** 2
16 advert["dY**2"] = ( advert["Sales"] -
17 np.mean(advert["Sales"])) ** 2
18
19 sxy = advert.sum()["dX*dY"]
20 sxx = advert.sum()["dX**2"]
21 syy = advert.sum()["dY**2"]
22
23 r = sxy/np.sqrt(sxx*syy)

```

[,]Salida de la pantalla

```

1      TV  Radio  Newspaper  Sales
2 0  230.1   37.8         69.2   22.1
3 1   44.5   39.3         45.1   10.4
4 2   17.2   45.9         69.3    9.3
5 3  151.5   41.3         58.5   18.5
6 4  180.8   10.8         58.4   12.9

```

[,]advertising2.py

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import pandas as pd
5
6 advert = pd.read_csv("./dataBases/Advertising.csv")
7 print(advert.head())
8
9 import numpy as np
10
11 def rCoef(df, var1, var2):
12     df["dX*dY"] = (df[var1]-np.mean(df[var1]))*(df[var2]-np.
13     mean(df[var2]))
14     df["dX**2"] = (df[var1]-np.mean(df[var1]))**2
15     df["dY**2"] = (df[var2]-np.mean(df[var2]))**2
16     sxy = df.sum()["dX*dY"]
17     sxx = df.sum()["dX**2"]
18     syy = df.sum()["dY**2"]
19     r = sxy/np.sqrt(sxx*syy)
20     return r
21
22 tvVsSales = rCoef(advert, "TV", "Sales")
23 ## 0.782224424862
24 radioVsSales = rCoef(advert, "Radio", "Sales")
25 ## 0.576222574571

```

	TV	Radio	Newspaper	Sales
TV	1	0.05	0.06	0.78
Radio	0.05	1	0.35	0.57
Newspaper	0.06	0.35	1	0.23
Sales	0.78	0.57	0.23	1

Figura 1.1: Matriz de correlación

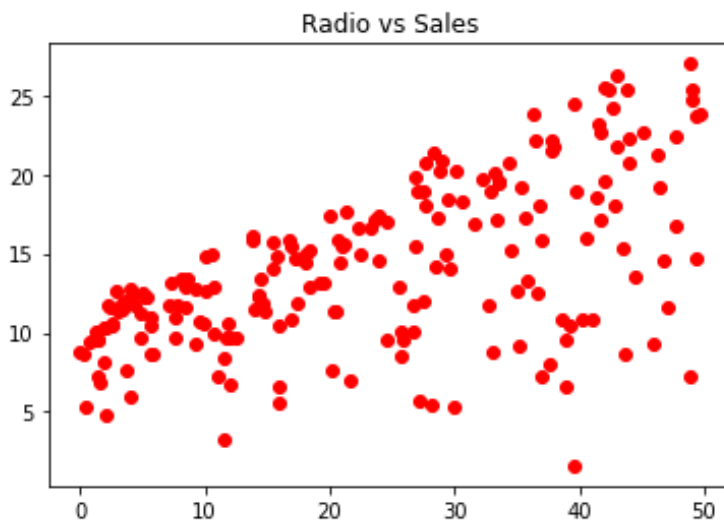
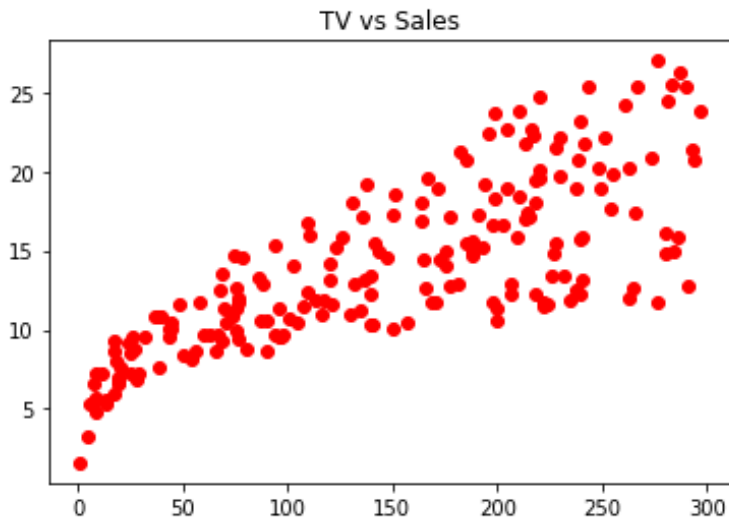
Veamos la naturaleza de esta correlación graficando las variables TV, Radio y Newspaper vs Sales del *data frame* advert.

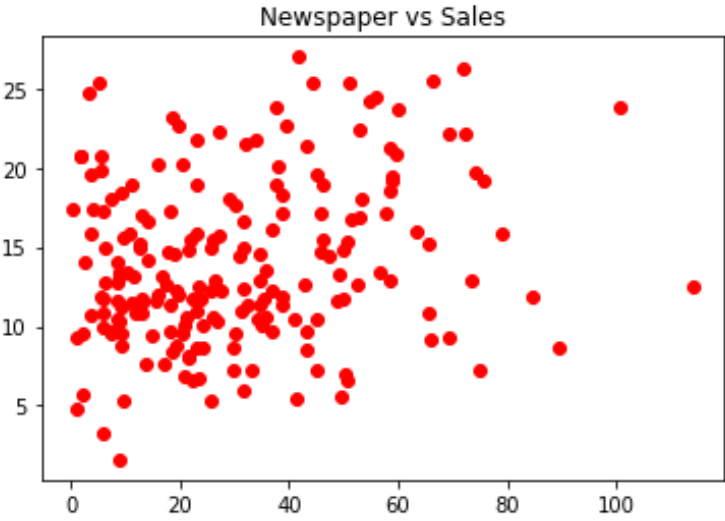
[,]tvVsSales.py

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import pandas as pd
5 advert = pd.read_csv("./dataBases/Advertising.csv")
6
7 import matplotlib.pyplot as plt
8
9 plt.plot(advert['TV'],advert['Sales'],'ro')
10 plt.title('TV vs Sales')
11 plt.show()
12
13 plt.plot(advert['Radio'],advert['Sales'],'ro')
14 plt.title('Radio vs Sales')
15 plt.show()
16
17 plt.plot(advert['Newspaper'],advert['Sales'],'ro')
18 plt.title('Newspaper vs Sales')
19 plt.show()

```







## 2 Probabilidad básica

### 2.1 Fundamentos de probabilidad

#### *Experimentos aleatorios*

**Ejemplo 2.1.** Si lanzamos una moneda, el resultado del experimento será “águila” (*tail* en inglés) que simbolizaremos por  $T$  o 0 o “sol” (*head* en inglés) simbolizado por  $H$  o 1, es decir, uno de los elementos del conjunto  $\{T, H\}$  (o bien  $\{0, 1\}$ .)

**Ejemplo 2.2.** Si lanzamos un dado, el resultado del experimento resultará en uno de los números del conjunto  $\{1, 2, 3, 4, 5, 6\}$ .

**Ejemplo 2.3.** Si lanzamos una moneda dos veces, existen cuatro posibles resultados:

$$\{HH, HT, TH, TT\}.$$

**Ejemplo 2.4.** Si estamos haciendo tornillos con una máquina, el resultado del experimento es que un tornillo puede salir defectuoso. Entonces cuando el tornillo este fabricado pertenecerá al conjunto

$$\{\text{defectuoso}, \text{no defectuoso}\}$$

**Ejemplo 2.5.** Si un experimento consiste en medir la *vida útil* de una bombilla eléctrica producida por una compañía, entonces el resultado del experimento es tiempo  $t$  medido en horas en algún intervalo

$$0 \leq t \leq T,$$

donde  $T$  es el tiempo de vida máximo de una bombilla.

#### *El espacio muestral*

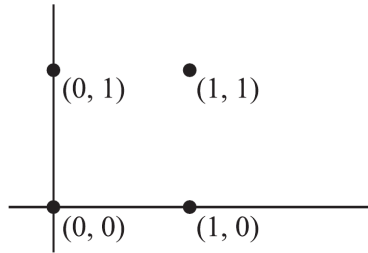
Un conjunto  $S$  que consiste de todos los posibles resultados de un experimento aleatorio es llamado *espacio muestral*, y cada posible resultado es llamado un *punto muestral*.

Usualmente existirá más de un espacio muestral que describe un experimento, pero usualmente, sólo uno provee la mayor información.

**Ejemplo 2.6.** Si lanzamos un dado, un posible espacio muestral está dado por  $\{1, 2, 3, 4, 5, 6\}$ , mientras que otro está dado por  $\{\text{par}, \text{impar}\}$ .

**Ejemplo 2.7.** Si lanzamos una moneda dos veces seguidas un posible espacio muestral esta dado en el ejemplo 2.3, mientras que otro esta dado por

$$\{(0, 0), (0, 1), (1, 0), (1, 1)\}.$$



Tipos de espacio muestral

- *Finito*: tiene un número finito de puntos.
- *Infinito numerable*: Tiene tantos puntos como los números naturales  $\mathbb{N}$  (es decir, podemos numerar el espacio).
- *Infinito no numerable*: Tiene tantos puntos como la recta real  $\mathbb{R}$ .  
Por ejemplo, el intervalo  $0 < x < 1$ .

Si el espacio muestral es finito o infinito numerable, diremos que es *discreto*. Si es infinito no numerable, diremos que es *continuo*.

### Eventos

Un *evento* es un subconjunto  $A$  de un espacio muestral  $S$ , es decir, un subconjunto de todos los posibles resultados de un experimento.

Si el resultado es un elemento de  $A$ , diremos que  $A$  *ha ocurrido*.

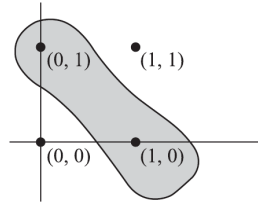
Un evento que consiste de un único punto de  $S$  es llamado a veces *evento elemental o simple*.

**Ejemplo 2.8.** Si lanzamos una moneda dos veces, el evento de que obtengamos *exactamente* un águila es un subconjunto del espacio muestral:

Como eventos particulares, podemos considerar todo el espacio muestral  $S$  como el *evento cierto o seguro* y el conjunto vacío  $\emptyset$  como el *evento imposible*.

Operaciones entre eventos Supongamos que  $A, B$  son dos eventos.

- $A \cup B$  es el evento "*ocurre  $A$  o  $B$  o ambos*", y también es llamado *unión de  $A$  con  $B$* .



- $A \cap B$  es el evento "ocurre  $A$  y  $B$ ", y también es llamado *intersección de  $A$  con  $B$* .
- $A'$  es el evento "no ocurre  $A$ ", y también es llamado *negación de  $A$* .
- $A - B = A \cap B'$  es el evento "ocurre  $A$  pero no  $B$ ", y también es llamado *diferencia de  $A$  menos  $B$* . Observe que  $A' = S - A$ .

Si  $A \cap B = \emptyset$ , entonces diremos que  $A$  y  $B$  son *disjuntos* o *mutuamente excluyentes*.

**Definición 2.1.** Si  $A_1, A_2, \dots$  es una colección de eventos tales que  $A_i \cap A_j = \emptyset$  siempre que  $i \neq j$ , entonces diremos que son *eventos mutuamente excluyentes*

**Definición 2.2.** Si  $A_1, A_2, \dots$  son eventos mutuamente excluyentes diremos que  $A_1 \cup A_2 \cup \dots$  es la *unión disjunta* de tales eventos y en ese caso escribiremos

$$A_1 \sqcup A_2 \sqcup \dots$$

Si

$$S = A_1 \sqcup A_2 \sqcup \dots$$

diremos que  $A_1, A_2, \dots$  es una *partición de  $S$* .

**Ejemplo 2.9.** Respecto al experimentos de lanzar una moneda dos veces, consideremos el evento  $A$  que consiste en *obtener al menos un sol*, mientras que el evento  $B$  consiste en que *el segundo lanzamiento sea un águila*.

Entonces  $A = \{TH, HT, HH\}$ ,  $B = \{HT, TT\}$  y por tanto

- (a)  $A \cup B = \{HT, TH, HH, TT\} = S$
- (b)  $A \cap B = \{HT\}$
- (c)  $A' = \{TT\}$
- (d)  $A - B = \{TH, HH\}$

### *El Concepto de Probabilidad*

A cualquier evento en un espacio muestral se le puede asignar un número entre  $0 = 0\%$  y  $1 = 100\%$  que representa su *probabilidad* de ocurrir.

Enfoque clásico Si un evento puede ocurrir en  $h$  diferentes maneras de un total de  $n$  posibles resultados, todos igualmente plausibles, entonces la probabilidad del evento es  $h/n$ .

**Ejemplo 2.10.** Supongamos que queremos conocer la probabilidad de que un sol aparezca en un solo volado. Desde que hay dos maneras diferentes *igualmente probables* en que la moneda caiga, y de esas dos maneras un sol sólo puede hacerlo de una manera, razonamos que su probabilidad es  $1/2$ .

*Observación 2.1.* Aquí suponemos que la moneda no está cargada.

Enfoque frecuencial Si después de  $n$  repeticiones de un experimento, donde  $n$  es suficientemente grande, se observa que un evento ocurre en  $h$  ocasiones, entonces diremos que la probabilidad del evento es  $h/n$ . Esta es también llamada *probabilidad empírica* del evento.

**Ejemplo 2.11.** Si lanzamos una moneda 1000 veces y obtenemos sol 532 veces, estimamos que la probabilidad resultantes es  $532/1000 = 0.532$ .

*Observación 2.2.* Ambos enfoque tienen sus inconvenientes:

- (a) En el caso clásico, la expresión “*igualmente probable*” es vaga;
- (b) mientras que en el enfoque frecuencial, “*un número muy grande*” no es preciso.

Por estas razones, los matemáticos han desarrollado un *enfoque axiomático* de la probabilidad.

### *Los Axiomas de la probabilidad*

Supongamos que tenemos un espacio muestral  $S$ . Supongamos que  $C$  es la colección de todos los eventos en  $S$ . Diremos que  $P : C \rightarrow \mathbb{R}$  es una función de probabilidad si satisface las siguientes propiedades:

**Axioma 2.1.** Para cada evento  $A$ , se tiene que

$$P(A) \geq 0. \quad (2.1)$$

**Axioma 2.2.** La probabilidad del evento cierto  $S$  es

$$P(S) = 1. \quad (2.2)$$



**Axioma 2.3.** Para cualquier cantidad numerable de eventos mutuamente excluyentes  $A_1, A_2, \dots$  tenemos que

$$P(A_1 \sqcup A_2 \sqcup \dots) = P(A_1) + P(A_2) + \dots \quad (2.3)$$

En particular, para dos eventos mutuamente excluyentes  $A_1, A_2$ ,

$$P(A_1 \sqcup A_2) = P(A_1) + P(A_2) \quad (2.4)$$

### *Algunos teoremas importantes en probabilidad*

**Teorema 2.1.** Si  $A_1 \subset A_2$ , entonces  $P(A_1) \leq P(A_2)$  y

$$P(A_2 - A_1) = P(A_2) - P(A_1).$$

**Teorema 2.2.** Para cada evento  $A$ ,

$$0 \leq P(A) \leq 1, \quad (2.5)$$

es decir, la probabilidad siempre se encuentra entre 0% y 100%.

**Teorema 2.3.** El evento imposible tiene probabilidad cero, es decir,

$$P(\emptyset) = 0. \quad (2.6)$$

**Teorema 2.4.** La probabilidad de un evento complementarios está dada por

$$P(A') = 1 - P(A) \quad (2.7)$$

**Teorema 2.5.** Si  $A = A_1 \sqcup \dots \sqcup A_N$  es la unión disjunta de eventos mutuamente excluyentes entonces

$$P(A) = P(A_1) + \dots + P(A_N). \quad (2.8)$$

En particular, si  $S = A_1 \sqcup \dots \sqcup A_N$  entonces

$$P(A_1) + \dots + P(A_N) = 1. \quad (2.9)$$

**Teorema 2.6.** Si  $A, B, C$  son dos eventos no necesariamente excluyentes, entonces

$$P(A \cup B) = P(A) + P(B) - P(A \cap B). \quad (2.10)$$

$$P(A \cup B \cup C) = P(A) + P(B) + P(C) \quad (2.11)$$

$$- P(A \cap B) - P(B \cap C) - P(C \cap A) \quad (2.12)$$

$$+ P(A \cup B \cup C). \quad (2.13)$$

**Teorema 2.7.** Para cualesquiera eventos  $A, B$ ,

$$P(A) = P(A \cap B) + P(A \cap B'). \quad (2.14)$$

**Teorema 2.8.** Si  $A_1, A_2, \dots, A_N$  es una partición del espacio muestral  $S$ , es decir,  $S = A_1 \sqcup A_2 \sqcup \dots \sqcup A_N$  entonces para cualquier evento  $A$

$$P(A) = P(A \cap A_1) + P(A \cap A_2) + \dots + P(A \cap A_N). \quad (2.15)$$

### Asignación de probabilidades

Si un espacio muestral consiste en una cantidad *finita* de posibles resultados  $a_1, \dots, a_N$ , entonces por el teorema 2.5,

$$P(A_1) + \dots + P(A_n) = 1 \quad (2.16)$$

donde  $A_1, \dots, A_n$  son *conjuntos elementales* o *eventos simples* dados por  $A_i = \{a_i\}$ .

Se sigue que uno puede escoger de manera arbitraria cualesquiera números no negativos como probabilidades de estos eventos simples, siempre que se satisfaga (2.16).

En particular, si suponemos *probabilidades iguales* para todos los eventos, entonces

$$P(A_k) = \frac{1}{n}, \quad k = 1, 2, \dots, n, \quad (2.17)$$

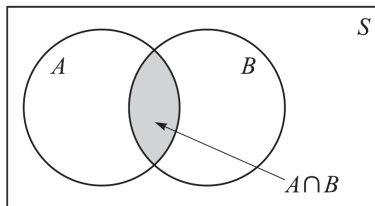
y si  $A$  es un evento formado por la unión disjunta de  $h$  eventos simples, entonces

$$P(A) = \frac{h}{n}. \quad (2.18)$$

*Observación 2.3.* Esto es equivalente al *enfoque clásico*. Pero podemos usar el *enfoque frecuencial* para asignar dichas probabilidades.

**Ejemplo 2.12.** Un solo dado se lanza. Encuentre la probabilidad de que obtengamos un 2 o un 5.

Sean  $A, B$  dos eventos tales que  $P(A) > 0$ .



Denotaremos por  $P(B|A)$  la probabilidad de  $B$  dado que  $A$  haya ocurrido y diremos que es la *probabilidad condicional* de  $B$  dado  $A$ .

## 2.2 Probabilidad condicional

### Definición

**Definición 2.3** (Probabilidad condicional).

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad (2.19)$$

$$P(A \cap B) = P(A)P(B|A) \quad (2.20)$$

*Observación 2.4.* La probabilidad condicional satisface todos los axiomas de una función de probabilidad. Podemos pensar  $P(\cdot|A)$  como la función de probabilidad que se obtiene al reemplazar el espacio muestral  $S$  por  $A$ .

[t]

**Ejemplo 2.13.** Encontrar la probabilidad de que una solo lanzamiento de un dado resulte en un número menor que 4 si

- (a) no hay más información;
- (b) se sabe que el lanzamiento resultó en un número impar.

### *Teoremas sobre Probabilidad Condicional*

**Teorema 2.9.** Para cualesquiera tres eventos  $A_1, A_2, A_3$ , tenemos que

$$P(A_1 \cap A_2 \cap A_3) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2) \quad (2.21)$$

**Teorema 2.10.** Si  $S = A_1 \sqcup \dots \sqcup A_N$ , entonces

$$P(A) = P(A_1)P(A|A_1) + \dots + P(A_N)P(A|A_N) \quad (2.22)$$

### *Eventos independientes*

Si  $P(B|A) = P(B)$ , i.e., la probabilidad de que  $B$  ocurra no está afectada por la ocurrencia de  $A$ , entonces diremos que  $A$  y  $B$  son independientes.

**Definición 2.4.**  $A$  y  $B$  son eventos independientes si y solo si

$$P(A \cap B) = P(A)P(B). \quad (2.23)$$

La definición se puede generalizar a más de dos eventos. Por ejemplo, diremos que  $A_1, A_2, A_3$  son eventos independientes si

$$k \neq j \rightarrow P(A_j \cap A_k) = P(A_j)P(A_k), \quad j, k = 1, 2, 3 \quad (2.24)$$

$$P(A_1 \cap A_2 \cap A_3) = P(A_1)P(A_2)P(A_3). \quad (2.25)$$

**Teorema 2.11** (Teorema de Bayes). Si  $S = A_1 \sqcup A_2 \sqcup \dots \sqcup A_N$ , entonces

$$P(A_k|A) = \frac{P(A_k)P(A|A_k)}{\sum_j P(A_j)P(A|A_j)} \quad (2.26)$$

[t]

**Ejemplo 2.14.** Demuestre el teorema de Bayes.

[t]

**Ejemplo 2.15.** La caja *I* contiene 3 canicas rojas y 2 azules, mientras que la caja *II* contiene 8 canicas rojas y 8 azules. Una moneda se lanza: Si cae un sol, se escoge una moneda de la caja *I* y si cae águila, de la caja *III*. Encuentre la probabilidad de obtener una canica roja.

[t]

**Ejemplo 2.16.** Supongamos que en el problema anterior, quien lanza la moneda no revela si ha caído águila o sol (de manera que la caja de la que se obtiene la canica no se revela) pero revela que una canica roja se ha obtenido. ¿Cual es la probabilidad de haber obtenido un sol?

### 2.3 *Análisis combinatorio*

El *análisis combinatorio* es una manera sofisticada de contar.

#### *Principio fundamental del conteo y diagramas de árbol*

Si una tarea se puede realizar en  $n$  formas diferentes y otra en  $m$  formas diferentes, entonces las dos tareas se pueden realizar en  $n \times m$  formas diferentes.

[t]

**Ejemplo 2.17.**

1. Si una persona tiene 2 camisas y 4 corbatas, ¿de cuantas formas puede combinarlas?
2. Construya un diagrama de árbol para representar todas estas opciones.

#### *Permutaciones*

Si tenemos  $n$  objetos distintos y queremos ordenarlos tendremos

$$n \times (n - 1) \times \dots \times 2 \times 1$$

formas diferentes de hacerlo.

**Definición 2.5** ( $n$  factorial).

$$n! = \begin{cases} 1 & n = 0 \\ n \times (n - 1)! & n > 0 \end{cases} \quad (2.27)$$

Si tenemos  $n$  objetos distintos y queremos arreglar  $r$  de estos en una linea, entonces tendremos una *permutación* de  $n$  en  $r$  dada por

$$P_r^n = n \times (n - 1) \times \dots \times (n - r + 1) \quad (2.28)$$

o de manera equivalente

$$P_r^n = \frac{n!}{(n-r)!} \quad (2.29)$$

[t]

**Ejemplo 2.18.** ¿Cuántas permutaciones de longitud 3 se pueden formar con las letras  $A, B, C, D, E, F, G$ ?

[t]

**Ejemplo 2.19.** Encuentre el número de permutaciones diferentes de las 11 letras de la palabra *MISSISSIPPI*.

### Combinaciones

En una *permutación*, uno está interesado en el orden de los objetos. Así  $abc$  y  $bca$  son permutaciones diferentes. Pero en algunos problemas, uno está interesado sólo en elegir objetos sin importar su orden. Tales selecciones se llaman *combinaciones*. Por ejemplo,  $abc$  y  $bca$  representan la misma combinación.

El número de combinación  $C_r^n$  al elegir  $r$  objetos de una colección de  $n$  diferentes está dada por el *número combinatorio*

$$C_r^n = \binom{n}{r} = \frac{n!}{r!(n-r)!} \quad (2.30)$$

Algunas fórmulas combinatorias

$$\binom{n}{r} = \frac{P(n, r)}{r!} \quad (2.31)$$

$$\binom{n}{r} = \binom{n}{n-r} \quad (2.32)$$

$$\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r} \quad (2.33)$$

[t]

**Ejemplo 2.20.** En una baraja inglesa, ¿cuántas formas hay de escoger dos cartas del mismo palo?

**Teorema 2.12** (Teorema del binomio).

$$(x+y)^n = \sum_{r=0}^n \binom{n}{r} x^r y^{n-r} \quad (2.34)$$

Aproximación de Stirling

$$n! \approx \sqrt{2\pi n} (n^n e^{-n}) \quad (2.35)$$



## 3 Variables Aleatorias Discretas

Supongamos que a cada punto del espacio muestral se le asigna un número. Entonces hemos definido una *función* en el espacio muestral. Esta función es llamada *variable aleatoria* (o *variable estocástica*) o de manera más precisa *función aleatoria*.

Usualmente, las variables aleatorias se denotan por letras mayúsculas como  $X$  o  $Y$ . En general, una variable aleatoria tiene algún significado físico, geométrico, económico, financieros, etc.

[t]

**Ejemplo 3.1.** Supongamos que una moneda se lanza dos veces de manera que el espacio muestral es  $\{HH, HT, TH, TT\}$ . Digamos que  $X$  representa el número de soles ( $H$ ) que obtenemos.

### 3.1 Funciones de probabilidad discretas

Sea  $X$  una variable aleatoria discreta. Supongamos que los valores que puede tomar son  $x_1, \dots, x_k$ , arreglados en algún orden dado. Supongamos también que esos valores tienen alguna probabilidad dada por

$$P(X = x_k) = f(x_k). \quad (3.1)$$

Función de probabilidad

$$P(X = x) = \begin{cases} f(x) & x = x_k \\ 0 & \text{en otro caso} \end{cases} \quad (3.2)$$

En general,  $f(x)$  será una función de probabilidad si

$$\begin{cases} f(x) \geq 0 \\ \sum_x f(x) = 1. \end{cases}$$

**Ejemplo 3.2.** Encuentre la función de probabilidad correspondiente a la variable aleatoria  $X$  del ejemplo 3.1.

**Ejemplo 3.3.** Suponga que un par de dados se lanzan. Sea  $X$  la variable aleatoria dada por la suma de los puntos. Encuentre la distribución de probabilidad de  $X$ .

**Ejemplo 3.4.** Encuentre la distribución de probabilidad de niños y niñas en familias con 3 hijos, suponiendo la misma probabilidad para niños y niñas.

### 3.2 Funciones de distribución para variables aleatorias discretas

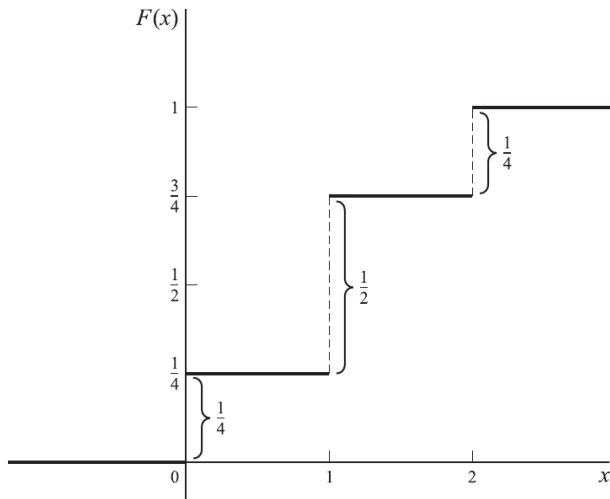
La función de distribución de una variable discreta  $X$  se obtiene de la función de probabilidad a través de la siguiente fórmula

$$F(x) = P(X \leq x) = \sum_{u \leq x} f(u). \quad (3.3)$$

Si  $X$  toma sólo un número finito de valores  $x_1, \dots, x_n$  entonces la función de distribución está dada por

$$F(x) = \begin{cases} 0 & -\infty < x < 0 \\ f(x_1) & x_1 \leq x < x_2 \\ f(x_1) + f(x_2) & x_2 \leq x < x_3 \\ \vdots & \vdots \\ f(x_1) + \dots + f(x_n) & x_n \leq x < \infty \end{cases} \quad (3.4)$$

**Ejemplo 3.5.** Encuentre la función de distribución para la variable aleatoria  $X$  del ejemplo 3.2 y obtenga su gráfica.



**Observación 3.1.** ■ Los saltos en la función de distribución están determinados por el valor de la función de probabilidad.

- Este tipo de funciones se conoce como *función escalonada*. Debe observarse que son *continuas por la derecha*.



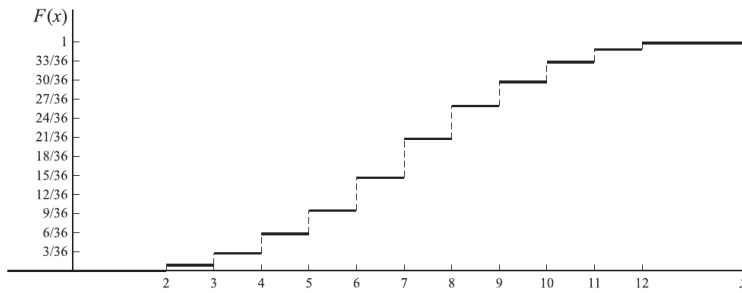
- La función de distribución es *monótonamente creciente*.

La función de probabilidad se puede obtener a partir de la función de distribución con la siguiente fórmula

$$f(x) = F(x) - \lim_{u \rightarrow x^-} F(u). \quad (3.5)$$

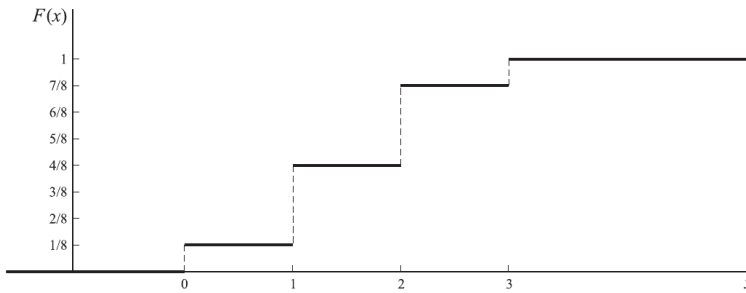
**Ejemplo 3.6.** (a) Encuentre la función de distribución  $F(x)$  para la variable aleatoria del problema resuelto 3.3;

- (b) grafique esta función de distribución.



**Ejemplo 3.7.** (a) Encuentre la función de distribución  $F(x)$  para la variable aleatoria del problema resuelto 3.4;

- (b) grafique esta función de distribución.





## 4 Variable Aleatorias Continuas

Una variable aleatoria no discreta  $X$  se llama *absolutamente continua* (o simplemente *continua*) si su función de distribución puede ser representada como

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(u)du, \quad -\infty < x < \infty. \quad (4.1)$$

La función  $f$  usualmente se llama *densidad de probabilidad* y debe satisfacer las siguientes propiedades:

1.  $f(x) \geq 0$
2.  $\int_{-\infty}^{\infty} f(x)dx = 1.$

La probabilidad de que  $X$  se encuentre entre dos valores  $a$  y  $b$  está dada por

$$P(a < x < b) = \int_a^b f(x)dx. \quad (4.2)$$

[t]

$$P(X = a) = 0. \quad (4.3)$$

Por tanto, en (4.2) podemos reemplazar cualquier signo  $<$  por  $\leq$ .

[t]

**Ejemplo 4.1.** (a) Encuentre la constante  $c$  tal que la función

$$f(x) = \begin{cases} cx^2 & 0 < x < 3 \\ 0 & \text{en otro caso} \end{cases} \quad (4.4)$$

sea una función de probabilidad.

(b) Calcule  $P(1 < X < 2)$ .

[t]

**Ejemplo 4.2.** Encuentre la distribución de probabilidad para la variable aleatoria del ejemplo 4.1 y utilícela para calcular  $P(1 < x \leq 2)$ .

La probabilidad de que  $X$  se encuentre entre  $x$  y  $x + \Delta x$  esta dada por

$$P(x \leq X \leq x + \Delta x) = \int_x^{x+\Delta x} f(u)du, \quad (4.5)$$

de manera que si  $\Delta x \approx 0$ , tendremos que

$$P(x \leq X \leq x + \Delta x) \approx f(x)\Delta x. \quad (4.6)$$

También podemos deducir de (4.1), al diferenciar de ambos lados, que

$$\frac{dF(x)}{dx} = f(x) \quad (4.7)$$

en todos aquellos puntos en que  $f(x)$  sea continua. Es decir, la derivada de la función de distribución es la función de densidad.

*Observación 4.1.* Existen variables aleatorias que no son discretas ni continuas. Por ejemplo

$$F(x) = \begin{cases} 0 & x < 1 \\ \frac{x}{2} & 1 \leq x < 2 \\ 1 & x \leq 2. \end{cases} \quad (4.8)$$

**Ejemplo 4.3.** Una variable aleatoria  $X$  tiene función de densidad

$$f(x) = \frac{c}{x^2 + 1}, \quad -\infty < x < \infty. \quad (4.9)$$

- (a) Encuentre el valor de  $c$ ;
- (b) encuentre la probabilidad de que

$$\frac{1}{3} < X^2 < 1.$$

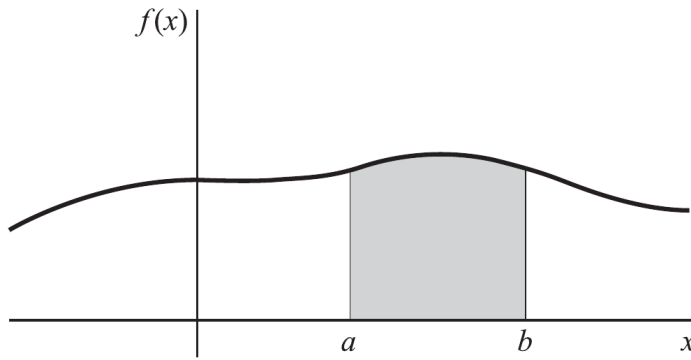
**Ejemplo 4.4.** Encuentre la función de distribución correspondiente a la función de densidad del problema resuelto 4.3

**Ejemplo 4.5.** La función de distribución para una variable aleatoria  $X$  es

$$F(x) = \begin{cases} 1 - e^{-2x} & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (4.10)$$

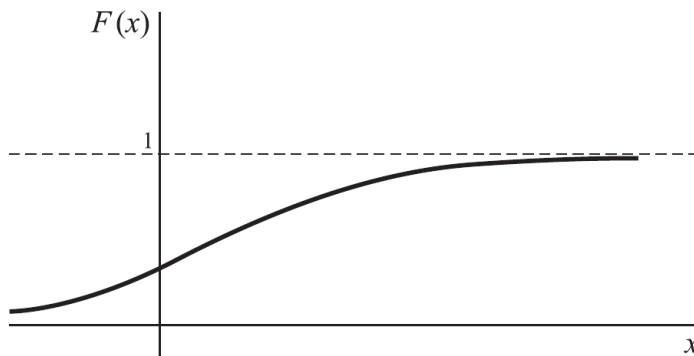
Encuentre

- (a) la función de densidad;
- (b) la probabilidad de que  $X > 2$ ;
- (c) la probabilidad que  $-3 < X \leq 4$ .



#### 4.1 Interpretación gráfica

La distribución  $F(x) = P(X \leq x)$  es monótonamente creciente de 0 a 1...  
...y el área bajo dicha curva es igual a 1.



#### 4.2 Distribución conjunta de probabilidad

Las ideas anteriores se generalizan fácilmente a dos o más variables.

Caso discreto Si  $X$  y  $Y$  son ambas variables aleatorias discretas, definimos la *función de probabilidad conjunta* de  $X$  y  $Y$  por

$$P(X = x, Y = y) = f(x, y) \quad (4.11)$$

donde

1.  $f(x, y) \geq 0$ ;
2.  $\sum_k \sum_y f(x, y) = 1$ .

Supongamos que  $X$  sólo toma uno de los  $m$  valores  $x_1, \dots, x_m$ , mientras que  $Y$  sólo toma uno de los  $n$  valores  $y_1, \dots, y_n$ .

Entonces la probabilidad del evento  $X = x_j, Y = y_k$  está dada por

$$P(X = x_j, Y = y_k) = f(x_j, y_k) \quad (4.12)$$

Una función de probabilidad conjunta para  $X$  y  $Y$  puede ser representada por una *tabla de probabilidad conjunta* como la siguiente:

$\begin{smallmatrix} Y \\ X \end{smallmatrix}$	$y_1$	$y_2$	$\dots$	$y_n$	Totals ↓
$x_1$	$f(x_1, y_1)$	$f(x_1, y_2)$	$\dots$	$f(x_1, y_n)$	$f_1(x_1)$
$x_2$	$f(x_2, y_1)$	$f(x_2, y_2)$	$\dots$	$f(x_2, y_n)$	$f_1(x_2)$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$x_m$	$f(x_m, y_1)$	$f(x_m, y_2)$	$\dots$	$f(x_m, y_n)$	$f_1(x_m)$
Totals →	$f_2(y_1)$	$f_2(y_2)$	$\dots$	$f_2(y_n)$	1 ← Grand Total

La probabilidad de  $X = x_j$  se obtiene de la siguiente manera

$$P(X = x_j) = f_X(x_j) = \sum_{k=1}^n f(x_j, y_k). \quad (4.13)$$

De manera similar, la probabilidad de  $Y = y_k$  se obtiene de la siguiente manera

$$P(Y = y_k) = f_Y(y_k) = \sum_{j=1}^m f(x_j, y_k). \quad (4.14)$$

Nos referiremos a  $f_X(x)$  y  $f_Y(y)$  como *funciones de probabilidad marginal* de  $X$  y  $Y$  respectivamente.

Observe que

$$\sum_{j=1}^m f_X(x_j) = 1, \sum_{k=1}^n f_Y(y_k) = 1, \quad (4.15)$$

lo cual se puede reescribir como

$$\sum_{j=1}^m \sum_{k=1}^n f(x_j, y_k) = 1. \quad (4.16)$$

La *función de distribución conjunta* está definida por

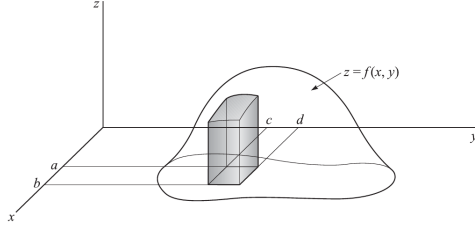
$$F(x, y) = P(X \leq x, Y \leq y) = \sum_{u \leq x} \sum_{v \leq y} f(u, v) \quad (4.17)$$

**Caso continuo** El caso en el que ambas variables son continuas es obtenido de manera análoga reemplazando las sumas por integrales.

La *función de probabilidad conjunta* (o de manera más común *función de densidad conjunta*) de  $X$  y  $Y$  está definida por

1.  $f(x, y) \geq 0$ ;
2.  $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy = 1$ .

Gráficamente  $z = f(x, y)$  representa una *superficie de probabilidad*



tal que el volumen bajo la superficie es igual a 1.

$$P(a < X < b, c < Y < d) = \int_a^b \int_c^d f(x, y) dy dx. \quad (4.18)$$

A cada evento  $A$  corresponde una región  $\mathcal{R}_A$  del plano  $xy$  tal que

$$P(A) = \iint_{\mathcal{R}_A} f(x, y) dx dy. \quad (4.19)$$

La *función de distribución conjunta* de  $X$  y  $Y$  en este caso está definida por

$$F(x, y) = P(X \leq x, Y \leq y) = \int_{-\infty}^x \int_{-\infty}^y f(u, v) dv du. \quad (4.20)$$

Se sigue que

$$\frac{\partial^2 F}{\partial x \partial y} = f(x, y) \quad (4.21)$$

$$P(X \leq x) = F_X(x) = \int_{-\infty}^x \int_{-\infty}^{\infty} f(u, v) dv du \quad (4.22)$$

$$P(Y \leq y) = F_Y(y) = \int_{-\infty}^{\infty} \int_{-\infty}^y f(u, v) dv du \quad (4.23)$$

Diremos que  $F_X(x), F_Y(y)$  son las *funciones de distribución marginal*, o simplemente *funciones distribuciones*, de  $X$  y  $Y$ , respectivamente.

Las derivadas de (4.22) y (4.23) con respecto a  $x$  y  $y$  son llamadas *funciones de densidad marginal*, o simplemente las *funciones de densidad*, de  $X$  y  $Y$  están dados por

$$f_X(x) = \int_{-\infty}^{\infty} f(x, v) dv, \quad f_Y(y) = \int_{-\infty}^{\infty} f(u, y) du. \quad (4.24)$$

### 4.3 Variables Aleatorias Independientes

Supongamos que  $X$  y  $Y$  son variables aleatorias discretas. Si los eventos  $X = x$  y  $Y = y$  son eventos independientes para todo  $x, y$ , entonces diremos que  $X, Y$  son v.a's independientes.

En tal caso,

$$P(X = x, Y = y) = P(X = x)P(Y = y), \quad (4.25)$$

o de manera equivalente

$$f(x, y) = f_X(x)f_Y(y). \quad (4.26)$$

De manera inversa, si para todo  $x, y$  la función de probabilidad conjunta  $f(x, y)$  pueden ser expresada como el producto de funciones de probabilidad marginal  $f_X(x)f_Y(y)$ , entonces  $X, Y$  son independientes.

Si no pueden expresarse de dicha manera, entonces  $X, Y$  son dependientes.

Si  $X, Y$  son v.a's continuas, diremos que son *independientes* si los eventos  $X \leq x$  y  $Y \leq y$  son independientes para todo  $x, y$ .

En tal caso, escribiremos

$$P(X \leq x, Y \leq y) = P(X \leq x)P(Y \leq y) \quad (4.27)$$

o de manera equivalente

$$F(x, y) = F_X(x)F_Y(y) \quad (4.28)$$

donde  $F_X(x)$  y  $F_Y(y)$  son las funciones de distribución marginal de  $X, Y$  respectivamente.

De manera inversa, si para todo  $x, y$  la función de probabilidad conjunta  $f(x, y)$  pueden ser expresada como el producto de funciones de probabilidad marginal  $F_X(x)F_Y(y)$ , entonces  $X, Y$  son independientes.

Si no pueden expresarse de dicha manera, entonces  $X, Y$  son dependientes.

Para v.a's independientes continuas, también es cierto que la función de densidad conjunta  $f(x, y)$  es el producto de funciones  $f_X(x)f_Y(y)$  y estas son las funciones de densidad marginal de  $X, Y$  respectivamente.

**Ejemplo 4.6.** La función de probabilidad conjunta de dos variables discretas  $X, Y$  está dada por

$$f(x, y) = \begin{cases} c(2x + y) & 0 \leq x \leq 2, 0 \leq y \leq 3 \\ 0 & \text{en otro caso.} \end{cases} \quad (4.29)$$

(a) Encuentre el valor de la constante  $c$ ;



(b) encuentre  $P(X = 2, Y = 1)$ ;

(c) encuentre  $P(X \geq 1, Y \leq 2)$ .

$X \backslash Y$	0	1	2	3	Totals ↓
0	0	$c$	$2c$	$3c$	$6c$
1	$2c$	$3c$	$4c$	$5c$	$14c$
2	$4c$	$5c$	$6c$	$7c$	$22c$
Totals →	$6c$	$9c$	$12c$	$15c$	$42c$

**Ejemplo 4.7.** Encuentre las funciones de probabilidad marginal para  $X$  y  $Y$  en el problema resuelto 4.6.

**Ejemplo 4.8.** Muestre que las variables aleatorias del problema resuelto 4.6 son dependientes.

**Ejemplo 4.9.** La función de densidad conjunta de dos variables aleatorias continuas  $X$  y  $Y$  es

$$f(x, y) = \begin{cases} cxy & 0 < x < 4, 1 < y < 5 \\ 0 & \text{en otro caso.} \end{cases} \quad (4.30)$$

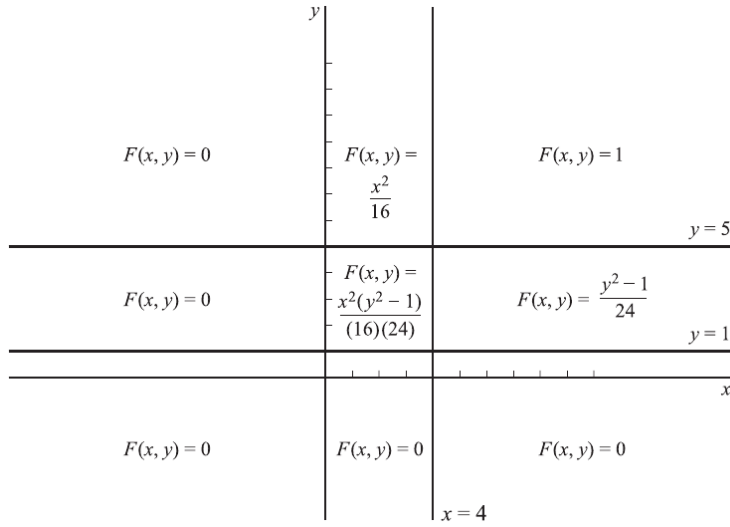
(a) Encuentre el valor de  $c$ ;

(b) encuentre  $P(1 < X < 2, 2 < Y < 3)$ ;

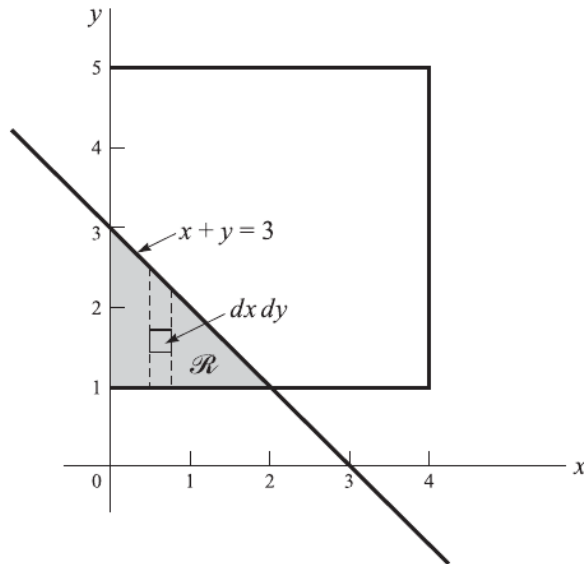
(c) encuentre  $P(X \geq 3, Y \leq 2)$ .

**Ejemplo 4.10.** Encuentre las funciones de probabilidad marginal de las v.a's  $X, Y$  del problema resuelto 4.9.

**Ejemplo 4.11.** Encuentre la función de distribución conjunta para las v.a's del problema resuelto 4.9.



**Ejemplo 4.12.** En el problema resuelto 4.9, encuentre  $P(X + Y < 3)$ .



#### 4.4 Distribución Condicional

Nosotros ya sabemos que si  $P(A) > 0$ ,

$$P(B|A) = \frac{P(A \cap B)}{P(A)}. \quad (4.31)$$

Si  $X, Y$  son v.a's discretas y tenemos los eventos  $A = \{X = x\}$ ,  $B =$

$\{Y = y\}$ , entonces (4.31) se convierte

$$P(Y = y|X = x) = \begin{cases} \frac{f(x, y)}{f_X(x)} & 0 < f_X(x) \\ 0 & \text{en otro caso} \end{cases} \quad (4.32)$$

$f(x, y) = P(X = x, Y = y)$  es la función de probabilidad conjunta, mientras que  $f_X(x)$  es la función de probabilidad marginal para  $X$ .

Definimos la *función de probabilidad condicional de  $Y$  dado  $X$*  como

$$f(y|x) = \begin{cases} \frac{f(x, y)}{f_X(x)} & 0 < f_X(x) \\ 0 & \text{en otro caso} \end{cases} \quad (4.33)$$

De manera similar, definimos la *función de probabilidad condicional de  $X$  dado  $Y$*  como

$$f(x|y) = \begin{cases} \frac{f(x, y)}{f_Y(y)} & 0 < f_Y(y) \\ 0 & \text{en otro caso} \end{cases} \quad (4.34)$$

Estas ideas son fácilmente extensibles al caso donde  $X, Y$  son v.a's continuas.

Por ejemplo, la *función de densidad condicional de  $Y$  dado  $X$*  es

$$f(y|x) = \begin{cases} \frac{f(x, y)}{f_X(x)} & 0 < f_X(x) \\ 0 & \text{en otro caso} \end{cases} \quad (4.35)$$

donde  $f(x, y)$  es la función de densidad conjunta de  $X$  y  $Y$  y  $f_X(x)$  es la función de densidad marginal de  $X$ .

Usando (4.35) podemos por ejemplo encontrar que la probabilidad que  $Y$  se encuentre entre  $c$  y  $d$  dado que  $X = x$  es

$$P(c < Y < d|X = x) = \int_c^d f(y|x) dy. \quad (4.36)$$

**Ejemplo 4.13.** Para la distribución del problema resuelto 4.6, encuentre

(a)  $f(y|2)$ ; y

(b)  $P(Y = 1|X = 2)$

**Ejemplo 4.14.** Si  $X$  y  $Y$  tienen función de densidad conjunta

$$f(x, y) = \begin{cases} \frac{3}{4} + xy & 0 < x < 1, 0 < y < 1 \\ 0 & \text{en otro caso,} \end{cases} \quad (4.37)$$

encuentre

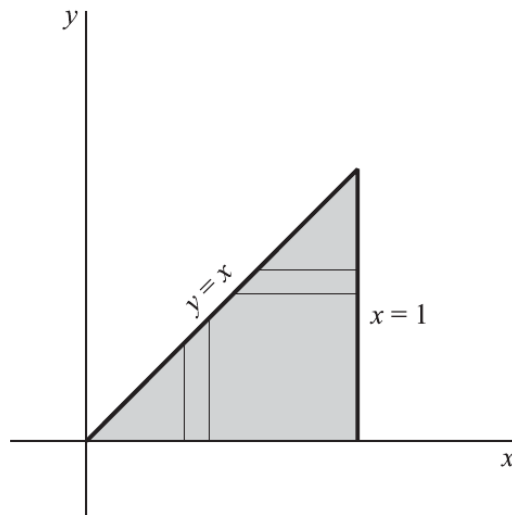
- (a)  $f(y|x)$ ;
- (b)  $P(Y > \frac{1}{2} | X = \frac{1}{2})$ .

**Ejemplo 4.15.** La función de densidad conjunta de las variables aleatorias  $X$  y  $Y$  está dada por

$$f(x, y) = \begin{cases} 8xy & 0 \leq x \leq 1, 0 \leq y \leq x \\ 0 & \text{en otro caso.} \end{cases} \quad (4.38)$$

Encuentre

- (a) la densidad marginal de  $X$ ;
- (b) la densidad marginal de  $Y$ ;
- (c) la densidad condicional de  $X$ ;
- (d) la densidad condicional de  $Y$ .



**Ejemplo 4.16.** Determine si las v.a's del problema resuelto 4.15 son independientes.

## 5 Esperanza Matemática

### 5.1 Definición de Esperanza Matemática

Para una variable aleatoria discreta  $X$  que toma valores  $x_1, \dots, x_n$ , la *esperanza matemática* se define como

$$E(X) = \sum_{j=1}^n x_j P(X = x_j) =: \sum x P(X = x), \quad (5.1)$$

o de manera equivalente

$$E(X) = \sum_{j=1}^n x_j f(x_j) =: \sum x f(x), \quad (5.2)$$

donde  $f(x) = P(X = x)$ .

Como un caso especial, cuando  $f(x) \equiv \frac{1}{n}$ , obtenemos la *media aritmética*:

$$E(X) = \frac{\sum_{i=1}^n x_i}{n}. \quad (5.3)$$

**Ejemplo 5.1.** Sea  $X$  el número que se obtiene al lanzar un dado. Entonces, cada cara  $x$  tiene la misma probabilidad

$$f(x) = \frac{1}{6}$$

de caer.

$$\text{Por tanto, } E(X) = (1) \left(\frac{1}{6}\right) + \dots + (6) \left(\frac{1}{6}\right) = \frac{1 + \dots + 6}{6} = 3.5$$

**Caso Discreto Numerable** En el caso en que  $X$  tome un cantidad (infinita) numerable de valores  $x_1, x_2, \dots$ , definimos

$$E(X) = \sum_{i=1}^{\infty} x_i f(x_i),$$

siempre y cuando dicha *serie* converja.

**Caso Continuo** Para una variable aleatoria continua  $X$  que tenga función de densidad  $f(x)$ , la esperanza de  $X$  se define como

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx \quad (5.4)$$

siempre y cuando dicha *integral* converja.

La esperanza de  $X$  es llamada a menudo *media* de  $X$  y es denotada por  $\mu_x$ , o simplemente  $\mu$ , cuando la variable aleatoria subyacente se sobreentiende.

La media o esperanza de  $X$  da un único valor que representa el promedio de los valores de  $X$ , y por esta razón decimos que es una *medida de tendencia central*.

**Ejemplo 5.2.** Supongamos que un juego se juega con un dado único que se suponen justos. En este juego, un jugador gana \$20 si un sale un 2; \$40 con un 4; \$30 con un 6; y no gana ni pierde con cualquier otra cara. Encuentre la suma esperada de dinero que ganaría.

$$\begin{aligned}\mu &= \$20 \left(\frac{1}{6}\right) + \$40 \left(\frac{1}{6}\right) + \$60 \left(\frac{1}{6}\right) \\ &= \frac{\$20 + \$40 + \$60 + 3 \times \$0}{6} \\ &= \$15\end{aligned}$$

**Ejemplo 5.3.** La función de densidad de una variable aleatoria  $X$  está dada por

$$f(x) = \begin{cases} \frac{1}{2}x & 0 < x < 2 \\ 0 & \text{en otro caso} \end{cases}$$

Encuentre el valor esperado de  $X$ .

$$\begin{aligned}\mu &= E(X) \\ &= \int_{-\infty}^{\infty} x f(x) dx \\ &= \int_0^2 x \left(\frac{1}{2}x\right) dx \\ &= \frac{1}{6} x^3 \Big|_0^2 \\ &= \frac{1}{6} (2)^3 - \frac{1}{6} (0)^3 \\ &= \frac{4}{3}\end{aligned}$$

### *Funciones de Variables Aleatorias*

Sea  $X$  una variable aleatoria discreta con función de probabilidad  $f(x)$ . Entonces  $Y = g(X)$  es una variable aleatoria discreta con función de probabilidad

$$h(y) = P(g(X) = y) = \sum_{\{x|g(x)=y\}} g(x)f(x) \quad (5.5)$$

Entonces, en el caso discreto.

$$E(g(X)) = \sum_x g(x)f(x) \quad (5.6)$$

De manera similar, en el caso continuo

$$E(g(X)) = \int_{-\infty}^{\infty} g(x)f(x)dx. \quad (5.7)$$

**Ejemplo 5.4.** Si  $X$  es la variable aleatoria del ejemplo 5.3, encuentre  $E(3X^2 - 2X)$ .

En este caso,  $g(x) = 3x^2 - 2x$ .

Recordemos que

$$f(x) = \begin{cases} \frac{1}{2}x & 0 < x < 2 \\ 0 & \text{en otro caso} \end{cases}$$

$$\begin{aligned} E(3X^2 - 2X) &= E(g(X)) \\ &= \int_{-\infty}^{\infty} (3x^2 - 2x) f(x) dx \\ &= \int_0^2 (3x^2 - 2x) \left(\frac{1}{2}x\right) dx \\ &= \int_0^2 \frac{3}{2}x^3 - x^2 dx \\ &= \left. \frac{3}{8}x^4 - \frac{1}{3}x^3 \right|_0^2 \\ &= \frac{10}{3} \end{aligned}$$

### Algunos temas sobre esperanza matemática

Linealidad

**Teorema 5.1.** Si  $c, d$  son constantes y  $X, Y$  son variables aleatorias, entonces

$$E(cX + dY) = cE(X) + dE(Y) \quad (5.8)$$

Esperanza e independencia

**Teorema 5.2.** Si  $X, Y$  son variables aleatorias independientes, entonces

$$E(XY) = E(X)E(Y) \quad (5.9)$$

## 5.2 Varianza y Desviación Estándar

Ya vimos que la espereza matemática de una variable aleatoria  $X$  es una medida de tendencia central y que generaliza a la *media aritmética*  $\mu$ .

*Observación 5.1.* Por esta razón, de aquí en adelante definiremos

$$\mu = \mu_X = E(X).$$

Otra cantidad de gran importancia es la *varianza* que se define como

$$\sigma_X^2 = \text{Var}(X) = E\left((X - \mu_X)^2\right) \quad (5.10)$$

La *desviación estándar* se definirá como

$$\sigma_X = \sqrt{\text{Var } X} \quad (5.11)$$

*Observación 5.2.* Si la variable aleatoria  $X$  se sobreentiende del contexto, omitiremos el subíndice correspondiente, es decir,

$$\mu = \mu_X, \sigma = \sigma_X, \sigma^2 = \sigma_X^2.$$

Si  $X$  es una variable aleatoria discreta, la varianza está dada por

$$\sigma^2 = E\left((X - \mu)^2\right) = \sum (x - \mu)^2 f(x), \quad (5.12)$$

siempre y cuando esta suma converja.

En el caso de que todas las probabilidades sean iguales y la variable aleatoria  $X$  sea finita tenemos

$$\sigma^2 = \frac{(x_1 - \mu)^2 + \dots + (x_n - \mu)^2}{n} \quad (5.13)$$

**Ejemplo 5.5.** Como vimos anteriormente, si  $X$  es la cara obtenida al lanzar un dado, entonces  $\mu_X = 3.5$ .

La varianza de  $X$  es

$$\begin{aligned} \sigma^2 &= \frac{(1 - 3.5)^2 + \dots + (6 - 3.5)^2}{6} \\ &= \frac{17.5}{6} \\ &\approx 2.916 \end{aligned}$$

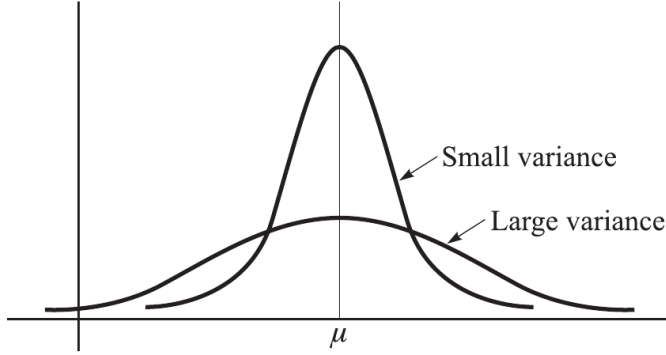
Si  $X$  es una variable aleatoria continua con función de densidad  $f(x)$ , entonces la varianza está dada por

$$\sigma^2 = E\left((X - \mu)^2\right) = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx \quad (5.14)$$

siempre y cuando la integral converja.

Tanto la varianza como la desviación estándar es una *medida de dispersión*.





**Ejemplo 5.6.** Encuentre la varianza y la desviación estándar de la variable aleatoria del ejemplo 5.3.

Recordemos que esta variable aleatoria  $X$  tiene densidad de probabilidad

$$f(x) = \begin{cases} \frac{1}{2}x & 0 < x < 2 \\ 0 & \text{en otro caso} \end{cases}$$

$$\begin{aligned} \sigma^2 = \text{Var}(X) &= \int_0^2 \left(x - \frac{4}{3}\right)^2 f(x) dx \\ &= \int_0^2 \frac{1}{2} \left(x - \frac{4}{3}\right)^2 x dx \\ &= \left. \frac{1}{8} x^4 - \frac{4}{9} x^3 + \frac{4}{9} x^2 \right|_0^2 \\ &= \frac{2}{9} \\ &\approx 0.2 \end{aligned}$$

Algunos teoremas sobre Varianza

$$\sigma^2 = E(X^2) - \mu^2 \quad (5.15)$$

$$\text{Var}(cX) = c^2 \text{Var}(X) \quad (5.16)$$

$$\sigma^2 = \min_a \left\{ E\left((X - a)^2\right) \right\} \quad (5.17)$$

Si  $X, Y$  son independientes

$$\text{Var}(X \pm Y) = \text{Var}(X) + \text{Var}(Y) \quad (5.18)$$

**Variables Aleatorias Estandarizadas** Sea  $X$  una variable aleatoria con media  $\mu$  y desviación estándar  $\sigma > 0$ . Diremos que la *variable aleatoria estandarizada* asociada está dada por

$$X^* = \frac{X - \mu}{\sigma}. \quad (5.19)$$

$$E(X^*) = 0, \text{Var}(X^*) = 1. \quad (5.20)$$

### 5.3 Covarianza y correlación

Los resultados dados anteriormente para una variable aleatoria pueden extenderse a dos variables.

$$\begin{aligned} \mu_X &= E(X) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f(x, y) dx dy \\ \mu_Y &= E(Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y f(x, y) dx dy \end{aligned} \quad (5.21)$$

$$\begin{aligned} \sigma_X^2 &= E((X - \mu_X)^2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \mu_X)^2 f(x, y) dx dy \\ \sigma_Y^2 &= E((Y - \mu_Y)^2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (y - \mu_Y)^2 f(x, y) dx dy \end{aligned} \quad (5.22)$$

Covarianza

$$\sigma_{XY} = \text{Cov}(X, Y) = E((X - \mu_X)(Y - \mu_Y)) \quad (5.23)$$

$$\sigma_{XY} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \mu_X)(y - \mu_Y) f(x, y) dx dy \quad (5.24)$$

Caso Discreto

$$\begin{aligned} \mu_X &= \sum_x \sum_y x f(x, y) \\ \mu_Y &= \sum_x \sum_y y f(x, y) \end{aligned} \quad (5.25)$$

$$\sigma_{XY} = \sum_x \sum_y (x - \mu_X)(y - \mu_Y) f(x, y) \quad (5.26)$$

$$\sigma_{XY} = E(XY) - E(X)E(Y) = E(XY) - \mu_X \mu_Y \quad (5.27)$$

Si  $X, Y$  son independientes, entonces

$$\sigma_{XY} = \text{Cov}(X, Y) = 0 \quad (5.28)$$

$$\text{Var}(X \pm Y) = \text{Var}(X) \pm 2 \text{Cov}(X, Y) + \text{Var}(Y). \quad (5.29)$$

De manera equivalente,

$$\sigma_{X \pm Y}^2 = \sigma_X^2 \pm 2\sigma_{XY} + \sigma_Y^2 \quad (5.30)$$

Coeficiente de correlación de Pearson

$$\rho = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} \quad (5.31)$$

**Teorema 5.3.**

$$|\sigma_{XY}| \leq \sigma_X \sigma_Y \quad (5.32)$$

$$|\rho| \leq 1. \quad (5.33)$$

Propiedades de la correlación

1.  $-1 \leq \rho \leq 1$
2.  $\rho \approx 0$ : *Correlación débil*, prácticamente no existe una correlación lineal.
3.  $|\rho| \approx 1$ : *Correlación fuerte*, la correlación está dada prácticamente por una función afín  $y = mx + b$ .
4.  $\rho > 0$ : *Correlación positiva*, en la medida que una crece, la otra también crece.
5.  $\rho < 0$ : *Correlación negativa*, en la medida que una crece, la otra decrece.

**Ejemplo 5.7.** Sean  $X, Y$  variables aleatorias discretas con densidad de probabilidad conjunta

$$f(x, y) = \begin{cases} \frac{2x+y}{42} & 0 \leq x \leq 2, 0 \leq y \leq 3 \\ 0 & \text{en otro caso.} \end{cases} \quad (5.34)$$

Encuentre los siguientes estadísticos:

- |                    |                                  |                     |
|--------------------|----------------------------------|---------------------|
| (a) $\mu_X = E(X)$ | (e) $E(Y^2)$                     | (i) $\sigma_Y$      |
| (b) $\mu_Y = E(Y)$ | (f) $\sigma_X^2 = \text{Var}(X)$ | (j) $\sigma_{XY} =$ |
| (c) $E(XY)$        | (g) $\sigma_Y$                   | $\text{Cov}(X, Y)$  |
| (d) $E(X^2)$       | (h) $\sigma_Y^2 = \text{Var}(Y)$ | (k) $\rho$          |

**Ejemplo 5.8.** Sean  $X, Y$  variables aleatorias continuas con densidad de probabilidad conjunta

$$f(x, y) = \begin{cases} \frac{1}{210}(2x+y) & 2 < x < 6, 0 < y < 5 \\ 0 & \text{en otro caso.} \end{cases} \quad (5.35)$$

Encuentre los siguientes estadísticos:

- |                    |                                  |                     |
|--------------------|----------------------------------|---------------------|
| (a) $\mu_X = E(X)$ | (e) $E(Y^2)$                     | (i) $\sigma_Y$      |
| (b) $\mu_Y = E(Y)$ | (f) $\sigma_X^2 = \text{Var}(X)$ | (j) $\sigma_{XY} =$ |
| (c) $E(XY)$        | (g) $\sigma_Y$                   | $\text{Cov}(X, Y)$  |
| (d) $E(X^2)$       | (h) $\sigma_Y^2 = \text{Var}(Y)$ | (k) $\rho$          |

## 6 Distribuciones especiales

### 6.1 La Distribución Binomial

Si  $p$  es la probabilidad de que en un solo ensayo ocurra un evento (llamada la probabilidad de éxito) y  $q = 1 - p$  es la probabilidad de que este evento no ocurra en un solo ensayo (llamada probabilidad de fracaso), entonces la probabilidad de que el evento ocurra exactamente  $x$  veces en  $N$  ensayos (es decir, que ocurran  $x$  éxitos y  $N - x$  fracasos) está dada por

$$f(x) = P(X = x) = \binom{N}{x} p^x q^{N-x} \quad (6.1)$$

donde  $x = 0, 1, \dots, N$ .

**Ejemplo 6.1.** La probabilidad de obtener exactamente dos caras en seis lanzamientos de una moneda es

$$\binom{6}{2} \left(\frac{1}{2}\right)^2 \left(\frac{1}{2}\right)^{6-2} = \frac{15}{64}$$

empleando (6.1) con  $N = 6, x = 2, p = q = \frac{1}{2}$ .

**Ejemplo 6.2.** Calcule la probabilidad de obtener al menos 4 caras en 6 lanzamientos de una moneda.

En lo subsecuente, daremos por hecho que hemos importado los siguientes paquetes:

- `scipy.stats`
- `numpy` como `np`

```
[fragile, allowframebreaks]statsBinom.py
```

```
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt
```

```
#Consideremos 6 experimentos con p de éxito 1/2
```

```

p=0.5
N=6
binDist = stats.binom(N,p)
#probabilidad de obtener dos éxitos
print binDist.pmf(2)
##0.234375
#probabilidad de obtener al menos 4 éxitos
print sum(binDist.pmf(np.arange(4,6+1)))
##0.34375

```

**Ejemplo 6.3.** Desarrolle  $(p+q)^4$ .

[fragile, allowframebreaks]coefBinom.py

```

from scipy import stats
import numpy as np

#coeficientes de (p+q)^4
p=.5
N=4
binomDist = stats.binom(N,p)
binDistExmp = binomDist.pmf(np.arange(5))
print binDistExmp*2**N
##[ 1.  4.  6.  4.  1.]

```

Propiedades de la distribución binomial Supongamos que realizamos  $N$  experimentos con probabilidad éxito  $p$  y de fracaso  $q = 1 - p$ .

$$\mu = Np \quad (6.2)$$

$$\sigma^2 = Npq \quad (6.3)$$

[fragile,allowframebreaks]histBinom.py

```

import numpy as np
import matplotlib.pyplot as plt

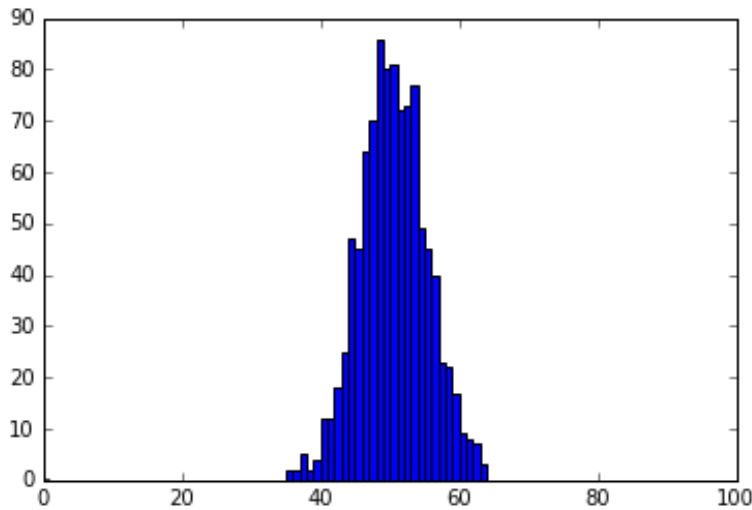
#Ejemplo de distribución binomial
N,p=100, 0.5
s = np.random.binomial(N,p,1000)

miHist = np.histogram(s, bins = np.arange(100+1))
print miHist[0]
print miHist[1]
print np.mean(s)
print N*p

```

```
print np.var(s)
print N*p*(1-p)

plt.hist(s, bins = np.arange(100+1))
plt.show()
```



## 6.2 Distribución Normal

Una de las distribuciones de probabilidad continua más importantes es la *distribución normal*, también llamada *distribución gaussiana*, que se define mediante la función de densidad

$$f_{a,b}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} \frac{(x-a)^2}{b^2}} \quad (6.4)$$

donde  $a, b$  son parámetros específicos para cada v.a.  $X$ .

Propiedades de la distribución normal Si la v.a.  $X$  tiene la función de densidad dada por (6.7), con parámetros  $a, b$  entonces

$$a = \mu_X \quad (6.5)$$

$$b = \sigma_X \quad (6.6)$$

Si una variable aleatoria normal  $X$  tiene función de densidad

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}}, \quad (6.7)$$

escribiremos  $X \sim N(\mu, \sigma^2)$ .

Variable aleatoria normalizada

$$Z = \frac{X - \mu}{\sigma} \quad (6.8)$$

$$\mu_Z = 0 \quad (6.9)$$

$$\sigma_Z = 1 \quad (6.10)$$

Forma Estándar

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \quad (6.11)$$

En este caso, diremos que  $Z$  está *normalmente distribuida*.

[fragile, allowframebreaks]distribucionNormal.py

```
import scipy.integrate as integrate
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon

def fn(x,m=0,s=1):
    return np.exp(-(x-m)**2/(2*s**2))/(s*np.sqrt(2*np.pi))
x1 = np.arange(-4,4,0.1)
plt.plot(x1, fn(x1))
plt.show()

for s in np.arange(1,4+1):
    result = integrate.quad(lambda x:fn(x),-s,s)
    print result

for s in np.arange(1,4+1):
    result = integrate.quad(lambda x:fn(x),-s,s)

    a, b = -s, s # integral limits
    x = np.arange(-4,4,0.01)
    y = fn(x)

    fig, ax = plt.subplots()
    plt.plot(x, y, 'r', linewidth=2)
    plt.ylim(ymin=0)

    # Make the shaded region
    ix = np.linspace(a, b)
    iy = fn(ix)
    verts = [(a, 0)] + list(zip(ix, iy)) + [(b, 0)]
    poly = Polygon(verts, facecolor='0.9', edgecolor='0.5')
    ax.add_patch(poly)
```



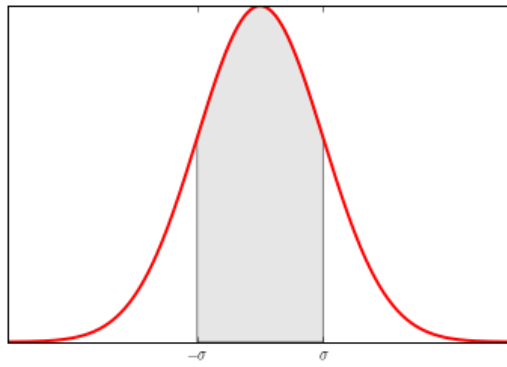
```

ax.set_xticks((a, b))
ax.set_xticklabels(('$_{-\sigma}$', '$_{\sigma}$'))
ax.set_yticks([])

plt.show()
print result

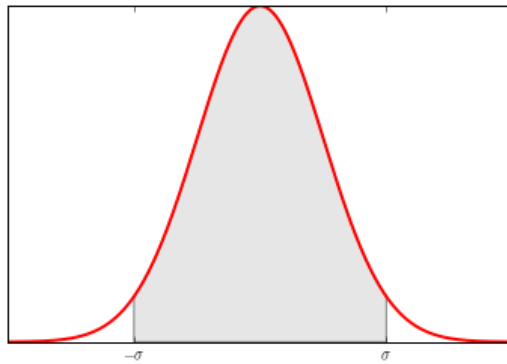
```

[fragile]



```
#(0.682689492137086, 7.579375928402476e-15)
```

[fragile]



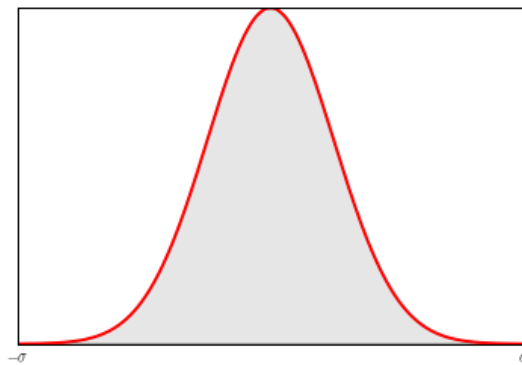
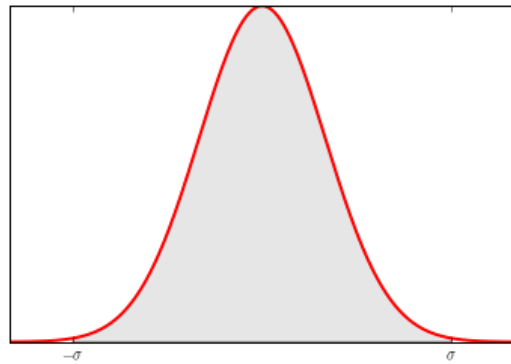
```
#(0.9544997361036417, 1.8403548653972355e-11)
```

[fragile]

```
#(0.9973002039367399, 1.1072256503105314e-14)
```

[fragile]

```
#(0.9999366575163339, 4.838904125482879e-12)
```



[fragile, allowframebreaks]normalCDF.py

```
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt

mu = 3.5
sigma = 0.76
nd = stats.norm(mu, sigma)

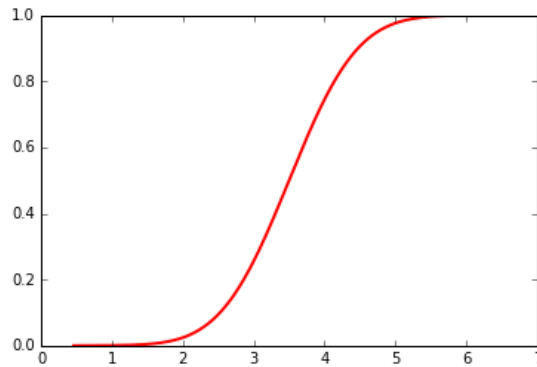
x = np.arange(mu - 4*sigma, mu + 4*sigma, 0.01)
y = nd.cdf(x)

fig, ax = plt.subplots()
plt.plot(x, y, 'r', linewidth=2)
plt.ylim(ymin=0)

for k in range(1,5):
    print nd.cdf(mu+k*sigma)-nd.cdf(mu-k*sigma)

#0.682689492137
```

```
#0.954499736104
#0.997300203937
#0.999936657516
```



### 6.3 Relación entre las distribuciones binomial y normal

Si  $N \sim \infty, p, q \gg 0$ , y  $X$  es una distribución binomial con parámetros  $N, p$  entonces

$$\frac{X - Np}{\sqrt{Npq}} \sim N(0, 1). \quad (6.12)$$

**Ejemplo 6.4.** Consideremos el experimento de lanzar 16 veces una moneda. Repitamos 1,000,000 dicho experimento. Compruebe que dicho experimento se puede modelar por una variable aleatoria con distribución  $N(\mu = 8, \sigma^2 = 4)$

[fragile, allowframebreaks]relBinomNormal.py

```
import numpy as np
import matplotlib.pyplot as plt

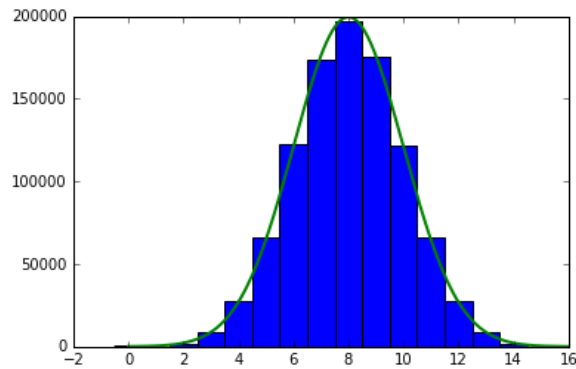
def fn(x,m=0,s=1):
    C = 1/(s*np.sqrt(2*np.pi))
    return C*np.exp(-(x-m)**2/(2*s**2))

N,p=30, 0.5
R = 1000000
q=1-p
mB = N*p
sB = np.sqrt(N*p*q)
X = np.random.binomial(N,p,R)
myBins = np.arange(-0.5,N+0.5,1)
plt.hist(X, bins = myBins)
x = np.arange(mB-4*sB,mB+4*sB+0.1,0.1)
```

```

y = R*fn(x, m=mB, s=sB)
plt.plot(x,y,lw=2)
plt.ylim(ymin=0)
plt.show()

```



## 6.4 La Distribución de Poisson

Distribución de Poisson Diremos que una variable aleatoria *discreta*  $X$  tiene distribución de Poisson si su función de probabilidad está dada por:

$$f(n) = \frac{\lambda^n e^{-\lambda}}{n!}, \quad n = 0, 1, 2, \dots \quad (6.13)$$

En este caso,  $\mu_X = \sigma^2 = \lambda$ .

En teoría de probabilidad y estadística, la distribución de Poisson es una distribución de probabilidad discreta que expresa, a partir de una frecuencia de ocurrencia media, la probabilidad de que ocurra un determinado número de eventos durante cierto período de tiempo. Concretamente, se especializa en la probabilidad de ocurrencia de sucesos con probabilidades muy pequeñas, o sucesos raros.

[Wikipedia: Distribución de Poisson](#)

**Ejemplo 6.5.** El número de personas por día que llegan a una sala de urgencias tiene una distribución de Poisson con media 5. Hallar la probabilidad de que cuando mucho lleguen tres por día y la probabilidad de que por lo menos lleguen 8 personas por día.

[fragile, allowframebreaks]distPoisson.py

```

from scipy import stats
import numpy as np
import matplotlib.pyplot as plt

```

```

def f(x, mu=1):
    return stats.poisson.pmf(x, mu)

def F(x, mu=1):
    return stats.poisson.cdf(x, mu)

x1 = np.arange(0,100+1)
plt.plot(x1, f(x1, mu=5), 'bo')
plt.show()

s = np.random.poisson(5,365)
M = np.max(s)
myBins = np.arange(0,M+1)
plt.hist(s, bins = myBins)
plt.show()

print F(3, mu=5)
print 1 - F(7, mu=5)

for k in range(12+1):
    print k, F(k, 5)
"""
0 0.00673794699909
1 0.0404276819945
2 0.124652019483
3 0.265025915297
4 0.440493285065
5 0.615960654833
6 0.762183462973
7 0.86662832593
8 0.931906365278
9 0.968171942694
10 0.986304731402
11 0.994546908087
12 0.997981148373
"""

```

## 6.5 Relación entre las Distribuciones Binomiales y de Poisson

Si en la función de probabilidad binomial,  $N$  es muy grande pero  $p \approx 0$ , esto modela un *evento raro*. En la práctica esto significa  $N \gg 50$ ,  $Np \ll 5$ .

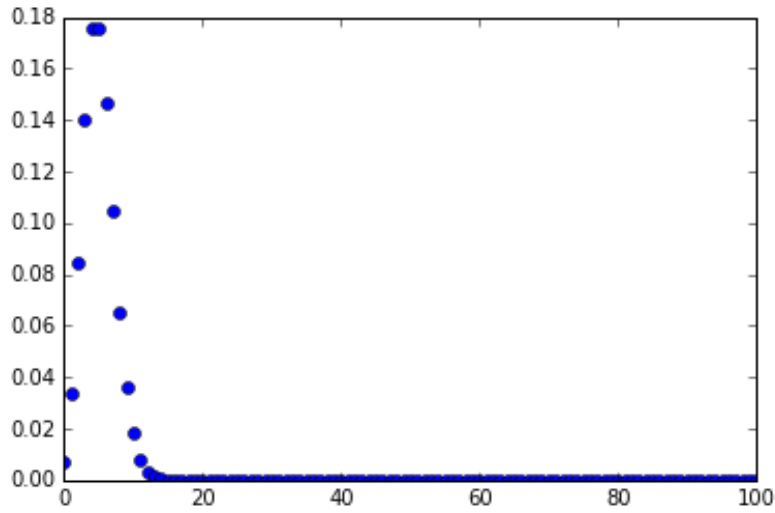
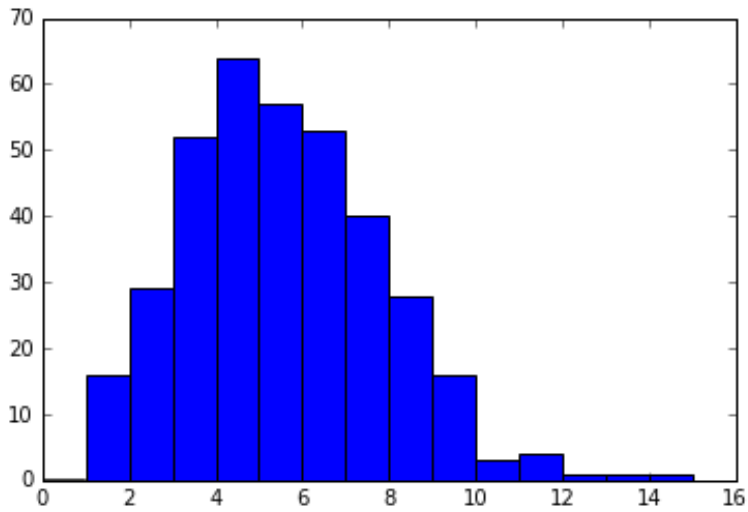


Figura 6.1: Distribución de Poisson

Figura 6.2: Histograma de pacientes en sala de urgencias durante un año con media  $\lambda = 5$ 

En este caso, la distribución Binomial con parámetros  $N, p$  se aproxima a una Poisson con parámetro  $\lambda = Np$ .

[fragile, allowframebreaks]relBinomPoisson.py

```
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl

mpl.style.use("ggplot")

fig, ax = plt.subplots(1, 1)
```

```

def fP(x, mu=1):
    return stats.poisson.pmf(x, mu)

def fB(x, N=30, p=0.5):
    return stats.binom(N,p).pmf(x)

N_=50
p_=5./N_
mu_ = N_*p_
x1 = np.arange(0,20+1)
ax.plot(x1, fP(x1, mu=mu_), 'bo', label="Poisson")
ax.plot(x1, fB(x1, N=N_, p=p_), 'ro', label="Binomial")
legend = ax.legend(loc='upper center', shadow=True)
plt.show()

```

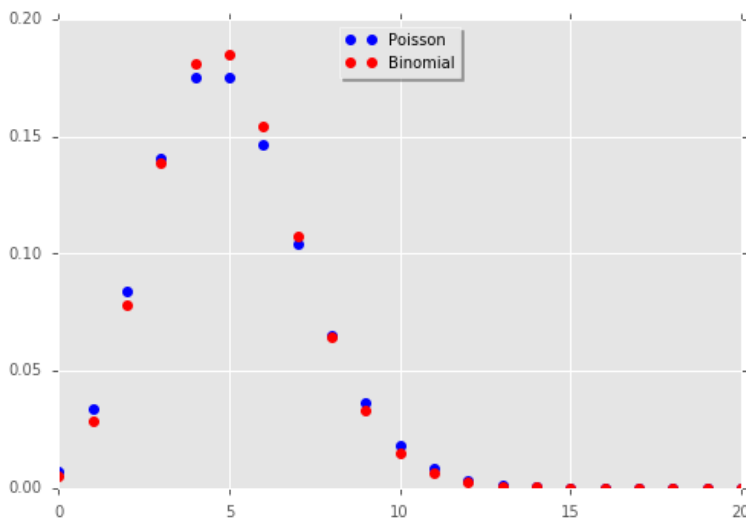


Figura 6.3: Comparación entre distribuciones Binomial y Poisson para eventos raros.

## 6.6 Distribución multinomial

Si los eventos  $E_1, \dots, E_k$  pueden ocurrir con probabilidades  $p_1, \dots, p_k$  respectivamente, entonces la probabilidad de que ocurran  $X_1, \dots, x_k$  veces respectivamente esta dado por la *distribución multinomial*

$$f(x_1, \dots, x_k) = \frac{x_1 + \dots + x_k}{x_1! \dots x_k!} p_1^{x_1} \dots p_k^{x_k}. \quad (6.14)$$

**Ejemplo 6.6.** Si un dado se lanza 12 veces, encontrar la probabilidad de obtener cada uno de los números 1, 2, 3, 4, 5, 6 exactamente dos veces.

## 6.7 Problemas Resueltos

Percentil Diremos que  $x = P_q$  es el percentil  $q$ ,  $0 \leq q \leq 100$  de la distribución  $F(x)$  si  $F(P_q) = q\%$ . En el caso de que  $q$  sea un valor realizable de  $F(x)$ , podemos “despejar”

$$P_q = F^{-1} \left( \frac{q}{100} \right). \quad (6.15)$$

A tal función se le llama *distribución inversa*.

Cuartiles En la literatura se definen conceptos similares. Por ejemplo, el primer *cuartil* corresponde al percentil 25; el segundo cuartil al percentil 50; y así sucesivamente.

Combinaciones

**Ejemplo 6.7.** Encuentre

(a)  $5!$

(b)  $\binom{8}{3}$

utilizando Python.

[fragile, allowframebreaks]combinaciones.py

```
import math
import scipy.special

print math.factorial(5)
print scipy.special.binom(8,3)
```

Distribución Binomial

**Ejemplo 6.8.** Supóngase que 15 % de la población es zurda. Encontrar la probabilidad de que en un grupo de 50 individuos haya:

- (a) cuando mucho 10 zurdos;
- (b) por lo menos 5 zurdos;
- (c) entre 3 y 6 zurdos;
- (d) exactamente 5 zurdos.

[fragile, allowframebreaks]solvedBinom.py

```
from scipy import stats
```

```
#7.2 N=50, p=15%
def f(x):
```



```

    return stats.binom(50,.15).pmf(x)
def F(x):
    return stats.binom(50,.15).cdf(x)
#(a) P(X<=10)
print sum([f(x) for x in range(0,10+1)])
##0.8800826828
print F(10)
##0.8800826828
#(b) P(X>=5)
print 1-sum([f(x) for x in range(0,4+1)])
##0.887894791945
print 1-F(4)
##0.887894791945
#(c) P(3<=X<=6)
print sum([f(x) for x in range(3,6+1)])
##0.3471108697
print F(6)-F(2)
##0.3471108697
#(d) P(X=5)
print f(5)
##0.3471108697

```

#### Distribución Normal

**Ejemplo 6.9.** En un examen final de matemáticas, la media fue 72 y la desviación estándar fue 15. Determinar las puntuaciones estándar de los estudiantes que obtuvieron:

- (a) 60;
- (b) 93;
- (c) 72.

**Ejemplo 6.10.** Con los datos del problema 6.9, encontrar las calificaciones que corresponden a las siguientes puntuaciones estándar:

- (a)  $-1$ ;
- (b) 1.6.

**Ejemplo 6.11.** Supóngase que la cantidad de juegos en que participan los beisbolistas de la liga mayor durante su carrera se distribuye normalmente con media de 1500 juegos y desviación estándar 350 juegos. Emplear Python para responder las siguientes preguntas:

- (a) ¿Qué porcentaje participa en menos de 750 juegos?;

- (b) ¿qué porcentaje participa en más de 2000 juegos?;
- (c) encontrar el *percentil* 90 de la cantidad de juegos en los que participan en su carrera.

[fragile, allowframebreaks]solvedNorm.py

```
from scipy import stats
```

```
mu = 1500
```

```
sigma = 350
```

```
nd = stats.norm(mu, sigma)
```

```
def F(x):
```

```
    return nd.cdf(x)
```

```
#a
```

```
print F(750)
```

```
##0.3471108697
```

```
#b
```

```
print 1-F(2000)
```

```
##0.0765637255098
```

```
def inverseF(x):
```

```
    return nd.ppf(x)
```

```
#c
```

```
print inverseF(.90)
```

```
##1948.54304794
```

Eventos raros

**Ejemplo 6.12.** Si la probabilidad de que un individuo tenga una reacción adversa por la inyección de determinado suero es 0.001, determinar la probabilidad de que de 2000 individuos:

- (a) exactamente 3;
- (b) más de 2

sufran una reacción adversa.

[fragile, allowframebreaks]eventosRaros.py

```
from scipy import stats
```

```
#7.28
```

```
#a
```

```
N = 2000
```

```
p = 0.001
print stats.binom(N,p).pmf(3)
##0.180537328032
print stats.poisson(N*p).pmf(3)
##0.180447044315
#(b)
print 1-stats.binom(N,p).cdf(2)
##0.32332356124
print 1-stats.poisson(N*p).cdf(2)
##0.323323583817
```



## 7 Pruebas de Hipótesis

### *Conceptos Estadísticos Importantes*

1. Pruebas de hipótesis.
2.  $p$ -valores.
3. Distribución normal.

*Muestreo aleatorio y teorema del límite central* Entender el concepto de *muestreo aleatorio* a través de ejemplos e ilustrar las aplicaciones del *teorema del límite central*.

*Pruebas de hipótesis* Entender el significado de los términos tales como *hipótesis nula*, *hipótesis alternativa*, *intervalos de confianza*,  $p$ -valores, *nivel de significación*, etc.

*Pruebas  $\chi$ -cuadrada* Calcularemos el estadístico  $\chi$ -cuadrada y describiremos el uso de pruebas  $\chi$ -cuadrada con un par de ejemplos.

### 7.1 *Muestreo aleatorio y teorema del límite central*

*Ejemplo* Supongamos que tratamos de encontrar la edad promedio en una ciudad, digamos Oaxaca. Una manera de hacerlo sería por *fuerza bruta*, es decir, recolectando esta información persona por persona. Pero este método sería muy costoso en términos de infraestructura y tiempo.

En estadística, este es un problema común, cuya solución está en el *muestreo aleatorio*: Tomemos un grupo de 1000 individuos (o 10,000 dependiendo de tu capacidad, obviamente entre más, es mejor) y calculemos la edad promedio en este grupo, a la que denotaremos por  $A_1$ .

Repitamos este procedimiento, digamos 100 veces, y denotaremos por  $A_1, A_2, \dots, A_{100}$  el promedio de edades obtenido en cada respectivo intento.

De acuerdo a la *ley de los grandes números*, la cantidad

$$\bar{A}_{100} = \frac{A_1 + \dots + A_{100}}{100} \quad (7.1)$$

es una aproximación muy cercana al promedio real de la edad de los pobladores de la ciudad.

De acuerdo al *teorema del límite central*, si el número de tales muestras es suficientemente grande,  $A_1, A_2, \dots, A_{100}$  estarán distribuidos de manera normal.

*Observación 7.1.* No estamos más interesados en obtener el valor exacto de la edad promedio, si no establecer un *estimador* para la misma.

En tal caso, tenemos que conformarnos con la definición de un *rango de valores* en el que el valor real podría estar.

## 7.2 Pruebas de hipótesis

En la prueba de hipótesis, asumimos una premisa inicial (generalmente relacionada con el valor del estimador) denominada *hipótesis nula* y trataremos de ver si es cierta o no aplicando.

Tenemos otra premisa llamada *hipótesis alternativa*, la cuál es la negación de la hipótesis nula.

*Hipótesis nula vs. alternativa* Cuando alguien está haciendo una *investigación* cuantitativa para calibrar el valor de un estimador, el *valor conocido* del parámetro se toma como *hipótesis nula*, mientras que el *nuevo valor* encontrado (de la investigación) se toma como la *hipótesis alternativa*.

En nuestro caso (encontrar la edad media de nuestra ciudad), un investigador puede afirmar que la edad *menor que 35*. Esto puede servir como la *hipótesis nula*.

Si una nueva agencia afirma que es *mayor que 35*, entonces se puede denominar como la *hipótesis alternativa*.

## 7.3 Estadísticos Z y t

1. Suponga que el valor del parámetro asumido en la hipótesis nula es  $A_0$ .
2. Tomemos una muestra aleatoria de 100 o 1000 personas o eventos del evento.
3. Calculemos la media del parámetro, por ejemplo la edad promedio de una ciudad, el tiempo medio de suministro de la pizza, la media ingresos, etc.

#### 4. Podemos llamarlo $A$ .

El estadístico  $Z$  se calcula para convertir una variable normalmente distribuida (por ejemplo, la distribución de la media poblacional de edad) a una distribución normal estándar.

El estadístico  $Z$  se da por la siguiente fórmula:

$$Z = \frac{A - A_0}{\sigma/\sqrt{n}} \quad (7.2)$$

donde  $\sigma$  es la desviación estándar de la población y  $n$  es el número de personas en la muestra

Ahora, debemos considerar dos casos

*Prueba  $Z$  (distribución normal)* El investigador conoce la desviación estándar del parámetro de su experiencia pasada.

Un buen ejemplo de esto es el caso del tiempo de entrega de una pizza. En este caso (7.2) seguirá una distribución normal y los valores normalizados se conocerán como *valores  $Z$* .

*Prueba  $t$  (distribución  $t$  de Student)* En este caso, el investigador no conoce la desviación estándar de la población.

Esto puede pasar porque:

- No existen tales datos en algún registro histórico;
- o el número de eventos o personas es demasiado pequeño para suponer una distribución normal.

En este caso, la media y la desviación estándar son desconocidas, y la expresión asume una distribución diferente a la normal llamada *distribución  $t$  de Student*.

El valor estandarizado en este caso es llamado  *$t$ -valor* y la prueba es llamada *prueba- $t$* .

#### *Distribución $t$ de Student*

La distribución de Student fue descrita en 1908 por William Sealy Gosset. Gosset trabajaba en una fábrica de cerveza, Guinness, que prohibía a sus empleados la publicación de artículos científicos debido a una difusión previa de secretos industriales. De ahí que Gosset publicase sus resultados bajo el seudónimo de Student.<sup>1</sup>

<sup>1</sup> Wikipedia: Distribución  $t$  de Student

```
1 from scipy import stats
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def ft(x, nu):
6     return stats.t.pdf(x, df=nu)
7 def Ft(x, nu):
```

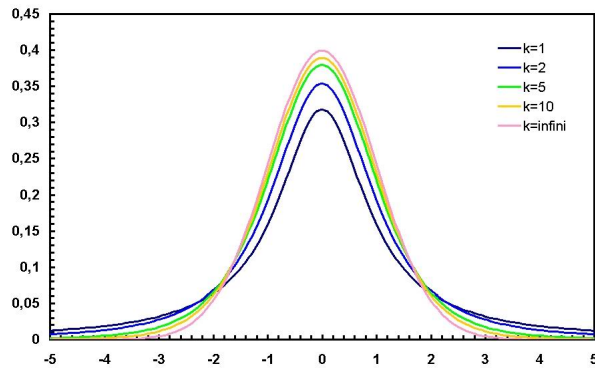


Figura 7.1: De The original uploader was Thorin de Wikipedia en francés - Transferido desde fr.wikipedia a Commons., CC BY-SA 1.0, <https://commons.wikimedia.org/w/index.php?curid=18>

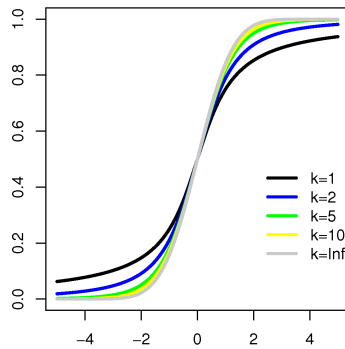


Figura 7.2: De Desconocido, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=78>

```

8     return stats.t.cdf(x, df=nu)
9 x = np.arange(-4,4,0.01)
10 yd = ft(x,30)
11 yc = Ft(x,30)
12
13 fig, ax = plt.subplots()
14 plt.plot(x, yd, 'r', linewidth=2)
15 plt.plot(x, yc, 'b', linewidth=2)
16 plt.ylim(ymin=0)
17 plt.show()

```

Listing 7.1: Distribución  $t$  en Python

El parámetro  $df$  se le conoce como *grados de libertad* y generalmente se denota como  $\nu$  (la letra  $\nu$  griega).

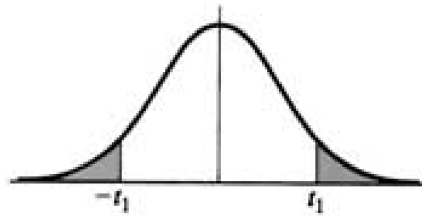
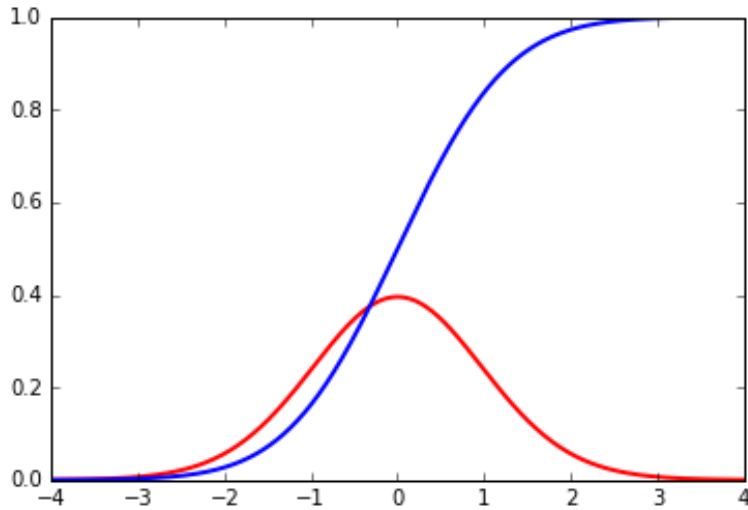
Si una variable aleatoria  $X$  tiene distribución  $t$  con  $\nu$  grados de libertad, entonces

$$\mu_X = 0, \sigma_X^2 = \frac{\nu}{\nu - 2} \quad (7.3)$$

**Ejemplo 7.1.** Consideremos una variable con distribución  $t$  y  $\nu = 9$  grados de libertad. Encuentre el valor de  $t$  para el cuál el área a la derecha sea 0.05 pero el total del área sin sombrear sea 0.90.

□tExample.py





```

1 from scipy import stats
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def tp(x, nu):
6     return stats.t.ppf(x, df=nu)
7
8 print tp(0.05, 9)
9 ##-1.83311293265
10 print tp(1-0.05, 9)
11 ##1.83311293265

```

*Varianza muestral*

$$S^2 = \sum \frac{(A_i - A_0)^2}{n-1} \quad (7.4)$$

*Estadístico t*

$$t = \frac{(A - A_0)}{S/\sqrt{n}} \quad (7.5)$$

## 7.4 Intervalos de confianza, niveles de significación y valores p

Supongamos que  $Z_1$  y  $Z_2$  son dos  $Z$ -estadísticos correspondientes a dos valores de una variable aleatoria y  $p_1$  y  $p_2$  son áreas encerradas

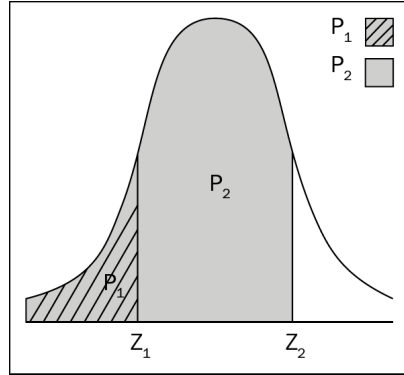


Figura 7.3: Una distribución típica normal con valores  $p$ .

por la curva de densidad a la derecha de esos valores.

En otras palabras

$$P(X > Z_1) = p_1 \quad (7.6)$$

$$P(X > Z_2) = p_2 \quad (7.7)$$

Entonces, podemos definir un intervalo en el cual encontrar el valor de una variable aleatoria, al cual llamaremos *intervalo de confianza*.

Por ejemplo, para una distribución normal con media  $\mu$  y desviación estándar  $\sigma$ , el valor de la variable aleatoria estará en el *intervalo*  $[\mu - 3\sigma, \mu + 3\sigma]$  con una *confianza* (problemaabilidad) del 99%.

Para cualquier *estimador* (variable aleatoria) que tenga una distribución normal, uno puede definir un intervalo de confianza si decidimos el nivel de confianza o problemaabilidad.

Podemos pensar en los *intervalos de confianza* como el umbral de los valores aceptados para sostener que la *hipótesis nula* es cierta.

Si el valor del estimador vive en este rango, será estadísticamente correcto decir que la hipótesis nula es correcta.

Para definir un intervalo de confianza, se necesita definir antes un *nivel (o problemaabilidad) de confianza*. Esta problemaabilidad necesita ser definida por el investigador dependiendo del contexto.

Digamos que esta problemaabilidad es  $p$ . En general, utilizaremos el *nivel de significación*

$$\beta = 1 - p, \quad (7.8)$$

que representa la problemaabilidad de que la hipótesis nula no sea correcta.

$\beta$  es definida por el investigador y usualmente esta en el orden de 0.01 a 0.1.

Un concepto importante que aprender aquí es el *valor de problemaabilidad* o simplemente *valor-p* de un estadístico: Es la problemaabilidad de que una variable aleatoria asuma un valor mayor al *valor-Z*

(o al *valor-t*)

$$p - \text{valor} = P(X > Z) \tag{7.9}$$

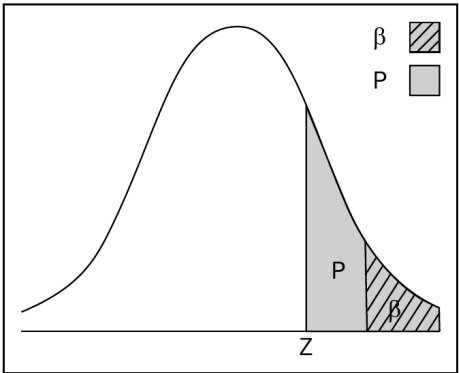


Figura 7.4: Una distribución normal típica con  $p$ -valores y nivel de significación.

*Criterio*

- Aceptar la hipótesis nula y rechazar la alternativa si  $p - \text{valor} > \beta$
- Aceptar la hipótesis alternativa y rechazar la nula si  $p - \text{valor} < \beta$

Debido a la simetría de la distribución normal, existen tres tipos de pruebas de hipótesis:

1. Cola izquierda;
2. cola derecha;
3. ambas colas.

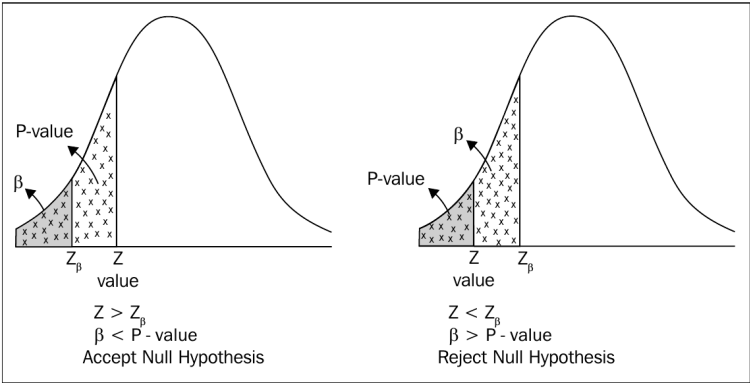


Figura 7.5: Prueba de hipótesis: Cola izquierda

*Cola izquierda*

*Cola derecha*

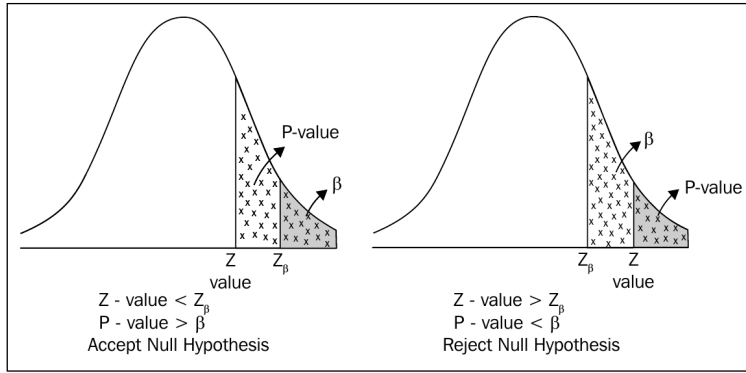


Figura 7.6: Prueba de hipótesis: Cola derecha

*Dos colas* Analizamos ambas colas y si en alguna de las dos colas, falla la respectiva prueba de hipótesis, la prueba de hipótesis se rechaza.

### 7.5 Una guía paso a paso para realizar una prueba de hipótesis

*Paso #1* Defina sus hipótesis nula y alternativa. Las hipótesis nula es algo que ya se establece y se acepta como cierto, al cuál denotamos como  $H_0$ . También, suponemos que el valor del parámetro en la hipótesis nula es  $A_0$ .

*Paso #2* Tome una muestra, digamos de unas 100 o 1000 personas o ocurrencias de eventos, y calcule el valor del estimador (por ejemplo, el promedio del parámetro como es la edad promedio, el tiempo promedio de entrega de la pizza, ingreso promedio, etc.) Digamos que este valor fue  $A_m$ .

*Paso #3* Calcule el valor normal estándar del valor  $Z$

$$Z = \frac{A_m - A_0}{\sigma/\sqrt{n}} \quad (7.10)$$

En la fórmula anterior,  $\sigma$  es la desviación estándar de la población o ocurrencias de eventos y  $n$  es el número de personas en la muestra.

La probabilidad asociada con el valor  $Z$  calculada en el paso 3 se debe comparar con el nivel de significación de la prueba a determinar si la hipótesis nula será aceptada o rechazada.

### 7.6 Un ejemplo de la prueba de hipótesis

*Planteamiento* Un famoso restaurante de pizza afirma que su tiempo de entrega es de 20 minutos, con una desviación estándar de 3 minutos.

Un investigador de mercados independiente afirma que ellos están desinflando los números para ganar clientes y el tiempo de entrega promedio es de hecho *21.2 minutos*.

¿Es su afirmación justificada o está el negocio de pizza correcto en su afirmación? *Suponga un nivel de significación del 5 %*

Definamos las hipótesis nula y alternativa:

- Lo que el negocio de pizzas afirma:  $H_0 : A_0 = 20$ .
- Lo que el investigador reclama:  $H_a : A_0 > 20$ .
- Desviación estándar (conocida):  $\sigma = 3$ .
- Tamaño de la muestra:  $n = 64$ .
- Nivel de significación:  $\beta = 0.05$ .

Calculemos el valor  $Z$ :

$$Z = \frac{21.2 - 20}{3/\sqrt{64}} = 3.2 \quad (7.11)$$

Denotemos por  $F$  la función de distribución acumulativa de una variable normal  $N(0, 1)$ .

Calculamos el valor  $p$  correspondiente al valor  $Z = 3.2$ :

$$\text{valor-p} = 1 - F(3.2) \quad (7.12)$$

$$= 1 - 0.999312862062 \quad (7.13)$$

$$= 0.000687137937916 \quad (7.14)$$

Esto significa que si la media fuera  $A_0 = 20$ , la probabilidad de que la media muestral fuera  $A = 21.2$  es  $\approx 0.069 \%$ .

Como elegimos un nivel de significación  $\beta = 5 \%$ , y  $0.069 \% < 5 \%$ , podemos rechazar la hipótesis nula  $H_0 : A_0 = 20$ .

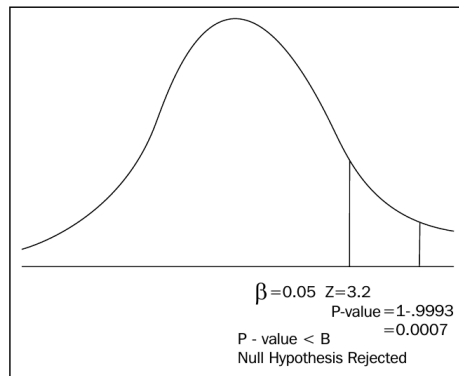


Figura 7.7: La hipótesis nula se rechaza porque el  $p$ -valor es menor al nivel de significación  $\beta$ .

## 7.7 Prueba $\chi$ -cuadrada

La prueba  $\chi^2$  se usa comúnmente para comparar *datos observados* vs *datos esperados* suponiendo que los datos siguen ciertas hipótesis.

Debemos suponer cierta hipótesis, la cuál nuestros datos seguirán y calculamos los datos esperados de acuerdo a esa hipótesis.

Debemos ya tener los datos observados, y calcular la desviación entre estos y los esperados usando el estadístico definido en la siguiente fórmula:

$$\text{valor } \chi^2 : g = \sum \frac{(O - E)^2}{E}, \quad (7.15)$$

donde  $O$  es el valor observado y  $E$  el esperado, con la suma sobre todos los posibles datos.

*Aplicaciones del estadístico  $\chi$ -cuadrada* La prueba de ji cuadrado se puede usar para hacer lo siguiente:

- Mostrar una relación causal o independencia entre una variable de entrada y otra de salida.
- Verificar si los datos observados provienen de una fuente justa / imparcial.
- Comprobar si los datos son demasiado buenos para ser verdad.

*Ejemplo* Realicemos un experimento hipotético en el que una moneda se lanza 10 veces. ¿Cuántas veces espera obtener ya sea un águila o un sol? La respuesta adecuada sería 5. Ahora bien, ¿qué pasaría si realizamos este experimento 1000 veces y registramos los números de águilas y soles.

Supongamos que observamos soles 553 veces y águilas el resto de ocasiones:

$H_0$  : La proporción de soles y águilas es 0.5

$H_a$  : La proporción no es 0.5

	Soles	Águilas
Observado	553	447
Esperado	500	500

Calculemos el valor  $\chi^2$  :

$$g = \frac{((553 - 500)^2 + (447 - 500)^2)}{500} \approx 11.236 \quad (7.16)$$

Este valor  $\chi^2$  se compara al valor en una *distribución  $\chi^2$*  para un número dado de *grados de libertad* y un nivel de significación.

La Distribución  $\chi^2$  Sean  $X_1, X_2, \dots, X_\nu$  variables aleatorias independientes  $N(0, 1)$ . Consideremos la variable aleatoria

$$\chi^2 = X_1^2 + \dots + X_\nu^2 \quad (7.17)$$

a la que llamaremos *chi cuadrada*. Su correspondiente distribución de probabilidad recibe el mismo nombre.

Propiedades de  $\chi^2$

$$\mu = \nu, \sigma = 2\nu \quad (7.18)$$

[]`stats.chi2`

```
1 from scipy.stats import chi2
2 import numpy as np
3 import matplotlib.pyplot as plt
4 fig, ax = plt.subplots(1, 1)
5
6 df = 55
7
8 x = np.linspace(chi2.ppf(0.01, df),
9                 chi2.ppf(0.99, df), 100)
10 ax.plot(x, chi2.pdf(x, df), 'r-',
11         lw=5, alpha=0.6, label='chi2 pdf')
```

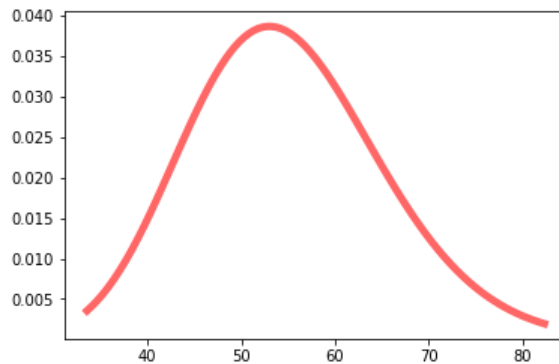
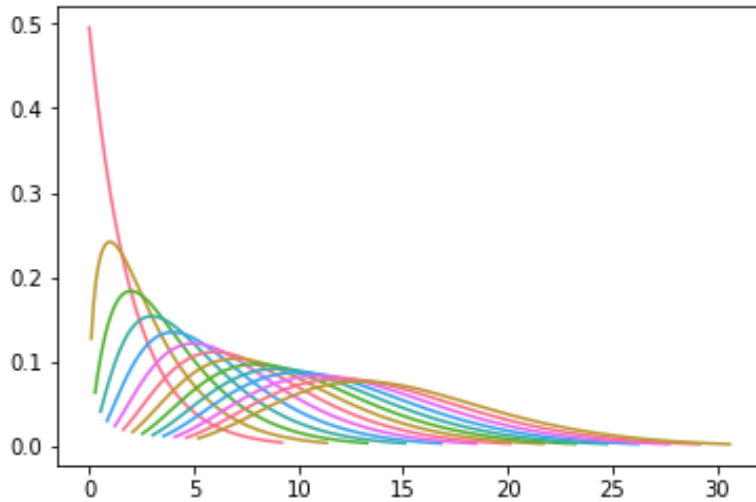


Figura 7.8: Función de densidad de distribución  $\chi^2$  con  $\nu = 55$

[]`statsChi2.py`

```
1 from scipy.stats import chi2
2 import numpy as np
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5
6 sns.set_palette("husl")
7 fig, ax = plt.subplots(1, 1)
8
9 for df in range(2, 15+1):
10     x = np.linspace(chi2.ppf(0.01, df),
11                     chi2.ppf(0.99, df), 100)
12     ax.plot(x, chi2.pdf(x, df), label='chi2 pdf')
```



*Regresando a nuestro ejemplo...* El número de grados de libertad es el número de categorías menos uno. En nuestro ejemplo  $\nu = 2 - 1 = 1$ . Supongamos un nivel de significación  $\beta = 0.05$ .

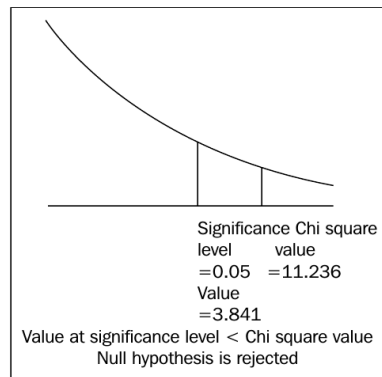


Figura 7.9: La hipótesis nula se rechaza porque el valor del estadístico  $\chi^2$  al nivel de significación es menor que el valor del estadístico.

¶ Otro ejemplo Examinemos otros ejemplo donde queremos demostrar que el género de un estudiante y las materias que escoge son independientes.

Supongamos que un grupo de estudiantes, la siguiente tabla representa el número de hombres y mujeres que toman matemáticas, arte y comercio como sus materias principales.

	Matemáticas	Artes	Comercio	Total
Hombres	68	52	90	210
Mujeres	28	37	35	100
Total	106	89	125	310

Si en la elección de las materias, no fuera relevante el género, entonces el número esperado de hombres y mujeres tomando diferentes materias sería



	Matemáticas	Arte	Comercio	Total
Niños				
Niñas				
Total				

Las desviaciones se calculan usando la fórmula  $(O - E)^2 / E$ :

	Matemáticas	Arte	Comercio	Total
Hombres				
Mujeres				
Total				

El estadístico  $\chi^2$  se obtiene al sumar todos estos valores.

*Conclusiones (del profesor)* Como  $\chi^2 = 4.99$  y el valor del estadístico  $\chi^2$  a un nivel de significación es 11.07, la hipótesis nula se acepta.

De manera equivalente

$$\text{valor-}p = 1 - F_{\chi^2}(4.99) = 0.416991040312 > \beta = 0.05, \quad (7.19)$$

obtenemos la misma conclusión:

*La elección de materias es independiente del género.*



## 8 Regresiones lineales

En esta unidad, trataremos con una técnica básica de modelación predictiva llamada *regresión lineal*, la cuál permite crear un modelo a partir de una base de datos histórica.

Nuestro propósito es entender las matemáticas detrás de la regresión lineal e ilustrar sus resultado a través de su implementación en varias bases de datos.

### *Mapa de ruta*

- Las matemáticas detrás de la regresión lineal.
- Implementación de la regresión lineal con Python.
- Interpretación de los parámetros resultantes.
- Validación del modelo.
- Manejo de Problemas relacionados con regresión lineal.

*Modelos matemáticos* Un *modelo matemático/estadístico/predictivo* es una ecuación matemática que consiste en *entradas* que producen *salidas* cuando el valor de las variables entrantes se introduce en el modelo.

*Ejemplo* Por ejemplo, supongamos que el precio  $P$  de una casa es *linealmente dependiente* en su tamaño  $S$ , comodidades  $A$  y disponibilidad de transporte  $T$ .

La ecuación correspondiente sería

$$P = a_1 \times S + a_2 \times A + a_3 \times T \quad (8.1)$$

Esta ecuación es llamado el *modelo* y los coeficientes  $a_1, a_2, a_3$  son sus parámetros.

La variable  $P$  es resultado predicho, mientras que  $S, A, T$  con las variables de entrada, que son datos conocidos.

Sin embargo, los parámetros  $a_i$  deben ser estimados a partir de los datos históricos.

Una vez que estos parámetros son determinados, el modelo está listo para ser problemaado.

### 8.1 Entendiendo las matemáticas detrás de la regresión lineal

Supongamos que tenemos una base de datos hipotética que contiene la información acerca del costo (en unidades de \$10000) de varias casas y sus respectivos tamaños (en pies cuadrados  $ft^2$ ).

Tamaño	Costo
1500	45
1200	38
1700	48
800	27

En este caso, el costo es la variable de salida, mientras que el tamaño es la variable de entrada.

La entrada y la salida generalmente se denotan por  $X$  y  $Y$ , respectivamente.

En el caso de la regresión lineal, supondremos que el costo  $Y$  es una función lineal de tamaño  $X$  y para estimar  $Y$ , proponemos el modelo

$$Y_e = \alpha + \beta X, \quad (8.2)$$

donde  $Y_e$  es el *valor estimado* de  $Y$  con base en nuestra ecuación lineal.

*Observación 8.1.* El propósito de la regresión lineal es encontrar valores  $\alpha, \beta$  estadísticamente significativos, que *minimicen* la diferencia entre  $Y$  y  $Y_e$ .

En el caso de nuestro ejemplo, si encontramos los valores de  $\alpha = 2$  y  $\beta = 0.03$ , entonces la ecuación será

$$Y_e = 2 + 0.03X. \quad (8.3)$$

Usando esta ecuación, podemos estimar el costo una casa de cualquier tamaño. Por ejemplo, para una casa de  $900ft^2$ , el costo será

$$Y_e = 2 + 0.03(900) = 29. \quad (8.4)$$

La siguiente pregunta que nos haremos es como estimar  $\alpha$  y  $\beta$ . Para esto usaremos un método llamado suma de *mínimos cuadrados* para la diferencia entre  $Y$  y  $Y_e$ , que representaremos como

$$\epsilon = Y - Y_e. \quad (8.5)$$

Nuestro objetivo es minimizar

$$\sum \epsilon^2 = \sum (Y - Y_e)^2 \quad (8.6)$$

$$= \sum (Y - (\alpha + \beta X))^2 \quad (8.7)$$

respecto de los parámetros  $\alpha, \beta$ .

Utilizando un poco de cálculo, se puede demostrar que los valores de los parámetros que minimizan la suma anterior son

$$\beta = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2} \quad (8.8)$$

$$\alpha = \bar{y} - \beta \times \bar{x} \quad (8.9)$$

## 8.2 Regresión lineal usando datos simulados

Para propósitos de regresión lineal, escribimos  $Y_e = \alpha + \beta X$ , aunque  $Y$  rara vez será lineal y podría tener un componente de error o residual, y en ese caso escribimos

$$Y = \alpha + \beta X + K. \quad (8.10)$$

En la ecuación anterior,  $K$  es el error, el cuál es una variable aleatoria que supondremos está normalmente distribuida.

Simulemos los datos para  $X$  y  $Y$  y tratemos de observar como es que los valores estimados ( $Y_e$ ) difieren del valor real ( $Y$ ).

Para  $X$ , generamos 100 números aleatorios normalmente distribuidos con media 1.5 y desviación estándar 2.5 (pero usted puede tomar otro par de números y experimentar.)

### Consideraciones

1. Para el valor ( $Y_e$ ), supondremos una ordenada al origen  $\alpha = 2$  y una pendiente  $\beta = 0.3$ .
2. Posteriormente, calcularemos los valores óptimos de  $\alpha$  y  $\beta$ , usando los datos simulados y veremos como cambia la eficacia del modelo.
3. Para el valor actual  $Y$ , adicionamos un término residual, que no es otra cosa que una variable normalmente distribuida con media  $\mu = 0$  y desviación estándar de  $\sigma = 0.5$ .

[,]fittingLinearRegression.py

```
1 import pandas as pd
2 import numpy as np
3
4 np.random.seed(1234)
5
6 x=2.5*np.random.randn(100)+1.5
7 res=.5*np.random.randn(100)+0
8 ypred=2+.3*x
9 yact=2+.3*x+res
10 xlist=x.tolist()
11 ypredlist=ypred.tolist()
12 yactlist=yact.tolist()
13 df=pd.DataFrame({'Input_Variable(X)':xlist,'Predicted_Output(ypred)':ypredlist,'Actual_Output(yact)':yactlist})
```

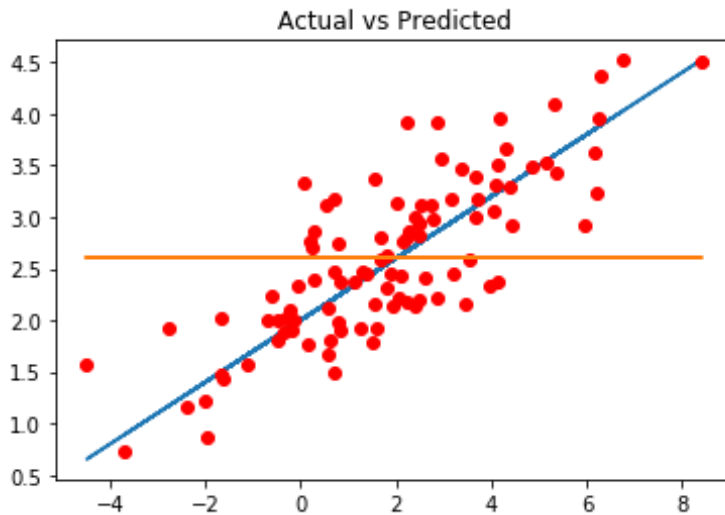
```

14 print(df.head())
15
16
17 import matplotlib.pyplot as plt
18
19 % x=2.5*np.random.randn(100)+1.5
20 % res=.5*np.random.randn(100)+0
21 % ypred=2+.3*x
22 % yact=2+.3*x+res
23
24 ymean=np.mean(yact)
25 yavg=[ymean for i in range(1,len(xlist)+1)]
26
27 plt.plot(x,ypred)
28 plt.plot(x,yact,'ro')
29 plt.plot(x,yavg)
30 plt.title('Actual vs Predicted')

```

```
[,]
```

	Actual_Output(yact)	Input_Variable(X)	Predicted_Output(ypred)
0	2.949179	2.678588	2.803576
1	1.840035	-1.477439	1.556768
2	3.776326	5.081767	3.524530
3	2.358159	0.718370	2.215511
4	2.151703	-0.301472	1.909558



En la gráfica anterior, la línea horizontal representa la *media* de los datos. En caso de que no tuviéramos algún otro modelo predictivo, nuestra mejor elección sería la *media aritmética*.

Otro punto para pensar es en como juzgar la eficiencia de nuestros modelos. Si usted pasa cualquier dato conteniendo dos variables, una de entrada y otra de salida, el programa de estadística generará algunos valores  $\alpha$ ,  $\beta$ . ¿Pero cómo entender que esos valores que se nos están dando son un buen modelo?

*Suma de Cuadrados Total*

$$SST = \sum (Y_i - \bar{Y})^2 \quad (8.11)$$

donde  $\bar{Y}$  es el valor promedio de  $Y_1, Y_2, \dots$ , los valores reales de  $Y$ .

*Suma de Cuadrados de Regresión*

$$SSR = \sum (Y_{e,i} - \bar{Y})^2 \quad (8.12)$$

donde  $\bar{Y}$  es el valor promedio de  $Y_1, Y_2, \dots$ , los valores reales de  $Y$ , mientras que  $Y_{e,i}$  son los valores predichos por el modelo para cada  $Y_i$ .

*Suma de Cuadrados de Diferencia*

$$SSD = \sum (Y_i - Y_{e,i})^2 \quad (8.13)$$

Recordemos que  $Y_e = \alpha + \beta X$ , con  $\beta$  definida por (8.8) y  $\alpha$  por (8.9). Utilizando estas identidades se puede demostrar que

$$SST = SSR + SSD. \quad (8.14)$$

- $SSR$ : diferencia explicada por el modelo;
- $SSD$ : diferencia no explicada por el modelo;
- $SST$ : error total.

*Observación 8.2.* Entre mayor sea la proporción de  $SSR : SST$ , mejor será el modelo.

*Coefficiente de determinación* R-cuadrado:

$$R^2 = \frac{SSR}{SST} \quad (8.15)$$

Como  $SSR \leq SST$ , entonces  $0 \leq R^2 \leq 1$ , y entre más cercano sea a 1 mejor será el modelo.

$R^2$  es un buen indicador de que una regresión lineal será efectiva.

En el script `fittingLinearRegression.py`, podemos agregar el siguiente pedazo de código para calcular el valor  $R^2$ .

[.]rCuadrada.py

```
1 df['SSR']=(df['Predicted_Output(ypred)']-ymean)**2
2 df['SST']=(df['Actual_Output(yact)']-ymean)**2
3 SSR=df.sum()['SSR']
4 SST=df.sum()['SST']
5 SSR/SST
```

El valor obtenido es  $a \approx 0.65$ , que es algo bueno. Sin embargo, los valores  $\alpha = 2, \beta = 0.3$  para  $Y_e$  pueden que no sean los mejores.

### 8.3 Encontrando el valor optimo de los coeficientes de una regresión lineal

Regresemos a nuestro marco de datos `df`. La columna `Input_Variable(X)` es la variable predictora. La variable `Actual_Output(yact)`, como su nombre lo sugiere, es la variable de salida real.

Utilizando estas dos variables, podemos calcular los valores de  $\alpha$  y  $\beta$  de acuerdo a las fórmulas (8.9) y (8.8). En el siguiente script, implementaremos estas fórmulas para obtener un modelo optimo de regresión lineal.

[.]optimalValue.py

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Mon Oct 23 00:37:36 2017
5
6 @author: jdk2py
7 """
8
9 import pandas as pd
10 import numpy as np
11
12 np.random.seed(1234)
13
14 x=2.5*np.random.randn(100)+1.5
```

```

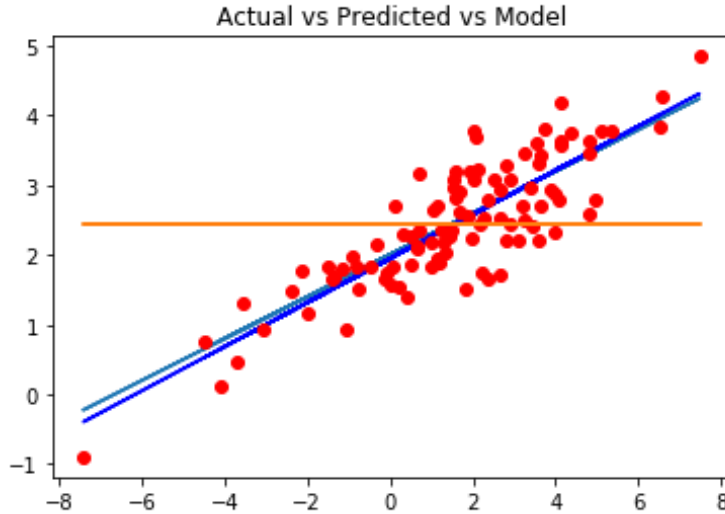
15 res=.5*np.random.randn(100)+0
16 ypred=2+.3*x
17 yact=2+.3*x+res
18 xlist=x.tolist()
19 ypredlist=ypred.tolist()
20 yactlist=yact.tolist()
21 df=pd.DataFrame({'Input_Variable(X)':xlist,'Predicted_Output(ypred)':ypredlist,'Actual_Output(yact)':yactlist})
22
23 ymean=np.mean(yact)
24 yavg=[ymean for i in range(1,len(xlist)+1)]
25
26 import matplotlib.pyplot as plt
27
28 xmean=np.mean(df['Input_Variable(X)'])
29 ymean=np.mean(df['Actual_Output(yact)'])
30 df['betan']=(df['Input_Variable(X)']-xmean)*(df['Actual_Output(yact)']-ymean)
31 df['xvar']=(df['Input_Variable(X)']-xmean)**2
32 betan=df.sum()['betan']
33 betad=df.sum()['xvar']
34 beta=betan/betad
35 alpha=ymean-beta*xmean
36 print(beta,alpha)
37
38 df['ymodel']=beta*df['Input_Variable(X)']+alpha
39
40 df['SSR']=(df['ymodel']-ymean)**2
41 df['SST']=(df['Actual_Output(yact)']-ymean)**2
42 SSR=df.sum()['SSR']
43 SST=df.sum()['SST']
44 R2 = SSR/SST
45
46 print(df.head())
47
48 plt.plot(x,ypred)
49 plt.plot(x,df['ymodel'], "b")
50 plt.plot(x,yact,'ro')
51 plt.plot(x,yavg)
52 plt.title('Actual vs Predicted vs Model')

```

[.]

	Actual_Output(yact) ypred	betan \	Input_Variable(X)	Predicted_Output(ypred)
0	2.803576	2.949179	2.678588	
1	1.556768	1.840035	-1.477439	
2	3.524530	1.873530	5.081767	
3	2.215511	3.776326	0.718370	
4	1.909558	4.629773	-0.301472	
		2.358159		
		0.080941		
		2.151703		
		0.565934		
	xvar	ymodel	SSR	SST
0	1.189860	2.796261	0.119028	0.247926
1	9.395573	1.481781	0.939885	0.373592
2	12.207943	3.556345	1.221220	1.755808
3	0.755875	2.176277	0.075614	0.008667
4	3.569275	1.853719	0.357052	0.089733





Además del estadístico  $R^2$ , hay otros estadísticos y parámetros que uno necesita mirar para hacer lo siguiente:

1. Seleccionar algunas variables y desechar otras para el modelo.
2. Evaluar la relación entre el predictor y la variable de salida y verificar si una variable de predicción es significativa en el modelo o no.
3. Calcular el error en los valores predichos por el modelo seleccionado.

Veamos ahora algunas de las estadísticas que ayudan a abordar los Problemas discutidos anteriormente.

[Valor- $p$ ] Es importante notar que al calcular los valores de  $\alpha$  y  $\beta$ , obtenemos estimados y no son exactos. Necesitamos demostrar su significación estadística usando una prueba de hipótesis.

La prueba de hipótesis es acerca de si el valor  $\beta$  es diferente de cero o no. En otras palabras, si existe la correlación necesaria entre  $X$  y  $Y$ . De haberla,  $\beta \neq 0$ .

En la ecuación  $y = \alpha + \beta x$ , si hacemos  $\beta = 0$ , no existirá correlación entre  $x$  y  $y$ . Entonces la prueba de hipótesis se define como

$$\text{Hipótesis Nula } H_0 : \beta = 0 \quad (8.16)$$

$$\text{Hipótesis alternativa } H_a : \beta \neq 0 \quad (8.17)$$

En general, si se realiza una regresión lineal y  $\beta$  es calculado, este proceso estará acompañado por un estadístico- $t$  y el valor- $p$  correspondiente.

Como veremos más adelante, Python tiene implementado un método para calcular este valor- $p$ .

Nuestra tarea entonces consistirá en comparar este valor- $p$  con un nivel dado de significación.

Como la desigualdad  $\beta \neq 0$  se puede descomponer en dos desigualdades

$$\beta > 0 \text{ o } \beta < 0, \quad (8.18)$$

entonces será una prueba de dos colas, y si el valor- $p$  es menor que el nivel de significación, entonces la hipótesis nula  $H_0 : \beta = 0$  se rechaza y diremos que  $\beta$  es significativo estadísticamente.

En caso contrario, nos permitirán no rechazar la hipótesis nula, de manera que  $\beta$  sería, muy poco significativo estadísticamente.

Como veremos en el caso de una regresión múltiple, este hecho nos ayudará a omitir columnas innecesarias de nuestro modelo: Entre mayor sea el valor- $p$ , menos significativas serán para el modelo y viceversa.

[Estadístico- $F$ ] Cuando uno se mueve de una regresión lineal simple a una regresión múltiple, existirán múltiples coeficientes  $\beta$  y cada uno de estos indicará una estimación.

En tal caso, aparte de problematizar la significación de cada variable en particular en el modelo (revisando los valores- $p$  asociados con su estimador), también será necesario revisar si, como un grupo, todos los estimadores son significativos o no.

Esto se puede hacer de la siguiente manera:

$$\text{Hipótesis nula } H_0 : \beta_1 = \beta_2 = \dots = \beta_n = 0 \quad (8.19)$$

$$\text{Hipótesis alternativa } H_a : \exists \beta_i \neq 0 \quad (8.20)$$

El estadístico que se usa en esta prueba de hipótesis se llama *estadístico- $F$*  y se define de la siguiente manera:

$$F = \frac{(SST - SSD) / p}{SSD / (n - p - 1)} \quad (8.21)$$

Dicho estadístico sigue la distribución  $F$ . Existirá un valor- $p$  asociado con este estadístico, tal que si dicho valor es suficientemente pequeño (es decir, menor al nivel de significación), la hipótesis nula puede ser rechazada.

[

La significación del estadístico- $F$  es como sigue:

- Los valores  $-p$  son acerca de relaciones individuales entre un predictor y un resultado. En caso de más de un predictor, dicha relación puede cambiar debido a la presencia de otras variables.
- El estadístico  $-F$  provee un manera de observar el cambio parcial en el valor  $-p$  asociado debido a la adición de la nueva variable.
- Cuando el número de los predictores en el modelos es muy grande y todas las  $\beta_i \approx 0$ , los valores  $-p$  individuales asociados con los predictores pueden ser muy pequeños.
- En tal caso, si sólo confiamos en valores  $-p$  individuales, podríamos concluir incorrectamente que existe una relación entre los predictores y el resultado, cuando no es así en realidad, y debemos fijarnos en el valor  $-p$  asociado con el estadístico  $-F$ .

[, ]Error residual estándar

Otro concepto para aprender es el concepto de *error residual estándar*.

Para un modelo de regresión lineal simple, se define de la siguiente manera:

$$RSE = \sqrt{\frac{1}{n-2} SSD} \quad (8.22)$$

donde  $n$  es el número de datos puntuales.

En general,

$$RSE = \sqrt{\frac{SSD}{n-p-1}} \quad (8.23)$$

donde  $p$  es el número de predictores en el modelo.

RSE es un estimado de la desviación estándar del término de error (res).

Este es el error que es inevitables aún si los coeficientes son conocidos correctamente.

Esto puede ser el caso porque el modelo carece de algo más, o quizá puede existir alguna variable en el modelo.

Nosotros sólo hemos mirado a una variable hasta ahora, pero en la mayoría de los escenarios tenemos que lidiar con regresiones múltiples, donde puede haber más de una variable de entrada.

En regresiones múltiples, los valores RSE tienden a disminuir, a medida que adicionamos más variables que son predictores más significativos de las variables de salida.

El valor RSE para un modelo puede ser calculado usando el siguiente pedazo de código. Aquí, estamos calculando RSE para el marco de datos que hemos usado en nuestro modelo, df:

[]

```
1 n = len(df['Input_Variable(X)'])
2 df['SSD']=(df['Actual_Output(yact)']-df['ymodel'])**2
3 SSD=df.sum()['SSD']
4 RSE=np.sqrt(SSD/(n-2))
```

El valor RSE resultante en este caso es  $\approx 0.4925$ . Como se puede intuir, *entre más pequeño sea RSE, mejor es el modelo*. Nuevamente, el punto de referencia para comparar este error es la media de los datos reales yact. Así que observaremos un error de 0.4925 sobre 2.4512, que es  $\approx 20.09\%$  de error.

## 8.4 Implementando regresiones lineales con Python

Avancemos y tratemos de hacer un modelo de regresión lineal simple y veamos cuales son los Problemas que encaramos, y como pueden ser resueltos para hacer el modelo más robusto.

Usaremos los datos de publicidad que usamos anteriormente.

Los siguientes dos métodos implementan regresiones lineales en Python:

- El método ols ("ordinary least squares") y la librería statsmodel.formula.api
- El paquete scikit-learn

Implementemos una regresión lineal simple usando el primer método y después construyamos sobre un modelo de regresión lineal múltiple. Después también nos fijaremos como es que el segundo método es usado para hacer lo mismo.

### Regresiones lineales usando statsmodel

[]statsModelExample.py Primero importemos los datos desde Advertising.csv:

```
1 import pandas as pd
2 advert=pd.read_csv('./dataBases/Advertising.csv')
3 print(advert.head())
```

Recordemos que esta base de satos contiene información de presupuestos de publicidad gastados en TV, radio y periódicos, para ciertos productos en particular y sus ventas resultantes.

Esperamos una correlación positiva entre tales costos de publicidad y las ventas. Ya hemos visto que existe una buena correlación entre costos de publicidad en TV y ventas.

[] Ahora averigüemos como es esta relación con el siguiente código

```
1 import statsmodels.formula.api as smf
2 model1=smf.ols(formula='Sales~TV',data=advert).fit()
3 print(model1.params)
```

del cual obtenemos la siguiente información

```

1 Intercept    7.032594
2 TV          0.047537
3 dtype: float64

```

Aquí hemos supuesto que existe una regresión lineal entre costos de publicidad en TV y ventas, y hemos creado el mejor ajuste usando el método de mínimos cuadrados. Entonces, con nuestra notación, esto quiere decir que los parámetros de la regresión lineal tenemos que

$$\alpha = 7.032594, \beta = 0.047537 \quad (8.24)$$

y la ecuación de nuestro modelo será

$$\text{Ventas} = 7.032 + 0.047 * \text{TV} \quad (8.25)$$

Si recuerdas, hemos aprendido que los valores de estos parámetros son estimados y existirán valores  $-p$  asociados a estos. Si los valores  $-p$  son muy pequeños, podemos aceptar que tales parámetros tienen un valor diferente de cero y son estadísticamente significativos en el modelo.

[] Miremos estos valores  $p$  para dichos parámetros

```

1 print(model1.pvalues)
2
3 Intercept    1.406300e-35
4 TV          1.467390e-42
5 dtype: float64

```

Como puede apreciarse, los valores  $-p$  son muy pequeños; por tanto, los parámetros son significativos.

[] Revisemos ahora otro indicador importante de la eficacia del modelo,  $R^2$ . Aunque nosotros lo implementamos manualmente, podemos obtenerlos con la siguiente línea de código:

```

1 print(model1.rsquared)
2
3 0.61187505085

```

[] Si requerimos todos los parámetros del modelo en un sólo paso, podemos ocupar la siguiente línea de código

```
1 print(model1.summary())
```

De lo cuál obtenemos  
[]

```

1 OLS Regression Results
2 =====
3 Dep. Variable:          Sales    R-squared:
4 Model:                0.612      OLS    Adj. R-squared:
5 Method:                0.610      Least Squares    F-statistic:
6 Date:                312.1
7 Time:                Sun, 29 Oct 2017    problema (F-statistic)
8 No. Observations:    1.47e-42
9 Df Residuals:        04:11:12    Log-Likelihood:
10 Df Model:            -519.05
11 Covariance Type:    200    AIC:
12                    1042.
13                    198    BIC:
14                    1049.
15                    1
16                    nonrobust
17 =====
18
19                    coef    std err          t      P>|t|
20 [0.025    0.975]
21 -----
22 Intercept    7.0326    0.458    15.360    0.000
23             6.130    7.935
24 TV          0.0475    0.003    17.668    0.000
25             0.042    0.053

```

```

17 =====
18 Omnibus:                0.531    Durbin-Watson:
19      1.935
19 problema(Omnibus):      0.767    Jarque-Bera (JB):
20      0.669
20 Skew:                  -0.089    problema(JB):
21      0.716
21 Kurtosis:              2.779    Cond. No.
      338.

```

Como podemos ver, el estadístico- $F$  para este modelo es muy alto y el respectivo valor- $p$  es despreciable, lo cual sugiere que los estimados del parámetro para este modelo son todos significativos y no nulos.

[] Ahora predigamos el valor de las ventas basados en la ecuación que acabamos de encontrar. Esto podemos hacer de la siguiente manera

```

1 sales_pred=model1.predict(pd.DataFrame(advert['TV']))
2 print(sales_pred.head())
3
4 0    17.970775
5 1    9.147974
6 2    7.850224
7 3   14.234395
8 4   15.627218
9 dtype: float64

```

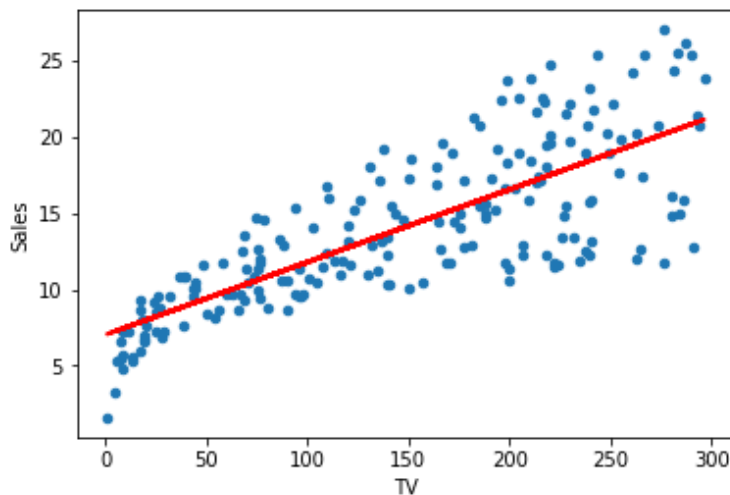
Esta ecuación básicamente calcula el valor de las ventas predichas para cada fila basada en la ecuación del modelo usando los costos de TV.

[] Podemos trazar sales\_pred contra el costo de publicidad en TV para encontrar la línea que mejor ajusta:

```

1 import matplotlib.pyplot as plt
2 advert.plot(kind='scatter', x='TV', y='Sales')
3 plt.plot(pd.DataFrame(advert['TV']),sales_pred,c='red',
4          linewidth=2)
4 plt.show()

```



[] Ahora calculemos el valor RSE

```

1 advert['sales_pred']=0.047537*advert['TV']+7.03
2 advert['RSE']=(advert['Sales']-advert['sales_pred'])**2
3 SSD=advert.sum()['RSE']
4 n = len(advert["Sales"])
5 RSE=np.sqrt(SSD/(n-2))
6 salesmean=np.mean(advert['Sales'])

```

```

7 error=RSE/salesmean
8 print(RSE,salesmean,error)

```

La salida consta de tres números, el primero de los cuales es  $RSE = 3.25$ , el segundo es  $salesmean$  (media de ventas reales) = 14.02 y  $error$  es su proporción, que es igual a 0.23.

Por lo tanto, en promedio, este modelo tendrá un 23 %, incluso si los coeficientes son correctamente predichos.

Esta es una cantidad significativa de errores y nos gustaría bajarla de alguna manera. Además, se puede mejorar el valor de  $R^2 = 0.61$ .

Algo que podemos intentar es agregar más columnas en el modelo, como predictores y ver si mejora el resultado o no.

## 8.5 Regresión lineal múltiple

Cuando la regresión lineal involucra más de un predictor, entonces es llamada *regresión lineal múltiple*.

La naturaleza del modelo permanece igual, lineal, excepto que puede haber múltiples pendientes  $\beta_i$  asociadas con cada predictor.

El modelo se representaría como sigue:

$$Y = \alpha + \beta_1 X_1 + \dots + \beta_n X_n \quad (8.26)$$

Cada  $\beta_i$  se estimará usando el mismo método, *mínimos cuadrados*; por tanto, tendríamos un valor  $-p$  asociado con las estimación:

1. Entre más pequeño sea este, más significativo será la variable para el modelo.
2. En cambio, las variables con valores  $-p$  muy grandes deberán ser eliminadas del mismo.

*Pros y contras*

1. Como la regresión lineal múltiple nos da la posibilidad de incluir más variables como predictores, entonces se incrementa la eficiencia del modelo.
2. Sin embargo, también incrementa la complejidad del proceso de construcción del modelo, ya que la selección de las variables significativas puede ser tedioso.

[] Con una base de datos simples de tres predictores, como en el ejemplo de la publicidad, puede haber varios modelos. Estos son:

Modelo 1: Ventas~TV

Modelo 2: Ventas~periódico

Modelo 3: Ventas~radio

Modelo 4: Ventas~TV+radio

Modelo 5: Ventas~TV+periódico

Modelo 6: Ventas~Periódico+radio

Modelo 7: Ventas~TV+radio+periódico

*Observación 8.3.* Un modelos con  $n$  posibles predictores, tendrá  $2^n - 1$  posibles modelos. Por tanto, a medida que los predictores se incrementan, la selección se volverá laboriosa.

Afortunadamente, tenemos algunos lineamientos para filtrar predictores y escoger los más eficientes:

- Mantenga las variables con valores  $-p$  más bajos y elimine aquellas con valores  $-p$  más altos.
- La inclusión de una variable al modelos idealmente debería incrementar el valor  $R^2$ . Sin embargo, más adelante *ajustaremos* dicho valor para que sea un indicador más confiable.

[] Con base en lo anterior, hay dos enfoques para seleccionar los predictores que quedarán en el modelo final:

*Selección progresiva*

1. En este enfoque, empezamos con modelo vacío (sin predictores) y entonces, comenzamos adicionando variables predictoras una por una.
2. La variable cuya adición resulte en el modelo con la menos de la suma residual de cuadrados será adicionada primero al modelo.
3. Si el valor  $-p$  para la variable es suficientemente pequeña y el valor  $R^2$  (ajustado) crece, el predictor se incluye en el modelo.
4. En otro caso, no.

*Selección regresiva*

1. En este enfoque, empezamos con un modelo que tiene todas las variables predictoras en el modelo y descartamos algunas de ellas.
2. Si el valor  $-p$  de una variable predictora es grande y el valor  $R^2$  (ajustado) decrece, el predictor se descarta del modelo.
3. En otro caso, permanece en el.

Muchos programas estadísticos, incluyendo Python, dan opciones para seleccionar entre los dos enfoques anteriores cuando se implementa una regresión lineal.

Por ahora, agreguemos algunas variables y veamos como cambia el modelo y la eficiencia, de manera que podamos tener un mejor panorama de que está tras el telón cuando estos enfoque se implementan en un programa estadístico.

*Modelo 2: 'Sales~TV+Newspaper'*

1. Nosotros ya hemos visto un modelo suponiendo una relación lineal entre ventas y costos de publicidad en TV,
2. Podemos ignorar los otros modelos que consisten de una sola variable.
3. Ahora tratemos de agregar más variables al modelo que ya tenemos y veamos como los parámetros cambian su eficiencia.

```
[ ]advertisingModel2.py
```

```
1 import pandas as pd
2 import statsmodels.formula.api as smf
3
4 advert = pd.read_csv("./dataBases/Advertising.csv")
5 model2=smf.ols(formula='Sales~TV+Newspaper',
6                 data=advert).fit()
7 print(model2.params)
8 print(model2.pvalues)
9 print(model2.rsquared)
```

```
[.]
```

```
1 Intercept      5.774948
2 TV             0.046901
3 Newspaper      0.044219
4 dtype: float64
5 Intercept      3.145860e-22
6 TV             5.507584e-44
7 Newspaper      2.217084e-05
8 dtype: float64
9 0.645835493829
```

Los valores  $-p$  para los coeficientes son muy pequeños, lo que sugiere que todos los estimados son significantes. La ecuación para este modelo será

$$\text{Ventas} = 5.77 + 0.046 * \text{TV} + 0.04 * \text{Periódico} \quad (8.27)$$

El valor  $R^2$  es 0.6458, el cuál resulta en una mejora muy pequeña del valor obtenido en el modelo anterior.

[ ] Los valores pueden ser predichos usando el siguiente retazo de código:

```
1 sales_pred=model2.predict(advert[['TV','Newspaper']])
2 print(sales_pred.head())
```

```
[.]
```

```
1 0      19.626901
2 1       9.856348
3 2       9.646055
4 3      15.467318
5 4      16.837102
6 dtype: float64
```

[.] Para calcular el valor RSE, utilizamos las siguientes líneas

```

1 #RSE
2 import numpy as np
3 advert['sales_pred']= 5.77 + 0.046901*advert['TV'] + \
4                       0.044219*advert['Newspaper']
5 advert['SSD'] = (advert['Sales']- \
6                 advert['sales_pred'])**2
7 SSD=advert.sum()['SSD']
8 n = len(advert["Sales"])
9 print("n",n)
10 p = 2
11 RSE=np.sqrt(SSD/(n-p-1))
12 print("RSE", RSE)
13 salesmean=np.mean(advert['Sales'])
14 print("salesmean", salesmean)
15 error=RSE/salesmean
16 print("error", error)

```

[.]

```

1 n 200
2 RSE 3.12072391442
3 salesmean 14.022500000000003
4 error 0.222551179492

```

El valor *RSE* resulta ser 3.12(22%), no muy diferente del modelo sólo con la *TV*. En la fórmula,  $p = 2$  porque es el número de predictores que estamos utilizando en el modelo.  
[] Utilizando la línea de código

```
1 print(model2.summary())
```

obtenemos la siguiente tabla que es el resumen del modelo:  
[.]

```

1
2                               OLS Regression Results
3 =====
4 Dep. Variable:                Sales    R-squared:
5 Model:                        0.646
6 Method:                      OLS      Adj. R-squared:
7 Date:                        0.642
8 Time:                        Least Squares    F-statistic:
9 No. Observations:            179.6
10 Df Residuals:                Fri, 03 Nov 2017    problema (F-statistic)
11 Df Model:                    3.95e-45
12 Covariance Type:             21:57:12    Log-Likelihood:
13                               nonrobust
14                               -509.89
15                               200    AIC:
16                               1026.
17                               197    BIC:
18                               1036.
19                               2
20                               nonrobust
21 =====
22
23      coef    std err          t      P>|t|
24 [0.025    0.975]
25 -----
26 Intercept    5.7749    0.525    10.993    0.000
27      4.739    6.811
28 TV           0.0469    0.003    18.173    0.000
29      0.042    0.052
30 Newspaper    0.0442    0.010    4.346    0.000
31      0.024    0.064

```

```

19 =====
20 Omnibus:                0.658    Durbin-Watson:
21      1.969
22 problema(Omnibus):      0.720    Jarque-Bera (JB):
23      0.415
24 Skew:                   -0.093    problema(JB):
25      0.813
26 Kurtosis:              3.122    Cond. No.
27      410.
28 =====
29
30 Warnings:
31 [1] Standard Errors assume that the covariance matrix of the
32     errors is correctly specified.

```

Aunque el estadístico  $-F$  decrece, el valor  $-p$  asociado también lo hace. Pero es solo una mejora marginal al modelo, como podemos ver en el valor  $R^2$ . De manera que agregar el periódico no mejora sustantivamente el modelo.

### Modelo 3: 'Ventas~TV+Radio'

Ahora tratemos de agregar la radio al modelo, en lugar del periódico. El radio tiene la segunda mejor correlación con la variable Ventas en la matriz de correlación que hemos creado anteriormente. Entonces se espera que existe alguna mejora significativa en el modelo debido a su adición al modelo. Veamos si esto ocurre o no:

[advertisingModel3.py]

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import pandas as pd
4 import statsmodels.formula.api as smf
5
6 advert = pd.read_csv("./dataBases/Advertising.csv")
7 model3=smf.ols(formula='Sales~TV+Radio',data=advert).fit()
8 print(model3.params)
9 print(model3.pvalues)
10
11 a = model3.params[0]
12 bTV = model3.params[1]
13 bRadio = model3.params[2]
14
15 advert["sales_pred"] = a + bTV * advert["TV"] + \
16     bRadio * advert["Radio"]
17
18 sales_pred=model3.predict(advert[['TV','Radio']])
19 print(sales_pred.head())
20
21 print(model3.summary())

```

[.]

```

1 #print(model3.params)
2 Intercept    2.921100
3 TV           0.045755
4 Radio        0.187994
5 dtype: float64
6
7 #print(model3.pvalues)
8 Intercept    4.565557e-19
9 TV           5.436980e-82
10 Radio        9.776972e-59
11 dtype: float64
12
13 #print(sales_pred.head())

```



```

14 0      20.555465
15 1      12.345362
16 2      12.337018
17 3      17.617116
18 4      13.223908
19 dtype: float64

```

```
[.]
```

```

1 #print(model3.summary())
2
3 =====
4 Dep. Variable:          Sales    R-squared:
5 Model:              0.897          OLS    Adj. R-squared:
6 Method:              0.896          Least Squares    F-statistic:
7 Date:              859.6          Sun, 05 Nov 2017    problema (F-statistic
8 ):              4.83e-98
9 Time:              20:09:44    Log-Likelihood:
10 No. Observations:      200    AIC:
11 Df Residuals:          197    BIC:
12 Df Model:              2
13 Covariance Type:      nonrobust
14 =====
15
16      [0.025      coef      std err      t      P>|t|
17      -----
18 Intercept      2.9211      0.294      9.919      0.000
19      2.340      3.502
20 TV      0.0458      0.001      32.909      0.000
21      0.043      0.048
22 Radio      0.1880      0.008      23.382      0.000
23      0.172      0.204
24 =====
25
26 Omnibus:          60.022    Durbin-Watson:
27      2.081
28 problema(Omnibus):          0.000    Jarque-Bera (JB):
29      148.679
30 Skew:          -1.323    problema(JB):
31      5.19e-33
32 Kurtosis:          6.292    Cond. No.
33      425.
34 =====
35
36 Warnings:
37 [1] Standard Errors assume that the covariance matrix of the
38      errors is correctly specified.
39
40 Observemos que el valor  $R^2$  se ha incrementado considerablemente debido a la adición de la
41 radio al modelo. De la misma manera, el estadístico  $-F$  se incrementado considerablemente del último
42 modelo indicando un modelo altamente eficiente.
43 El valor RSE puede ser calculado usando el mismo método descrito anteriormente:
44 [.]advertisingModel3.py (continuación)
45
46 #RSE

```

```

2 import numpy as np
3
4 advert['SSD'] = (advert['Sales'] - \
5                 advert['sales_pred'])**2
6 SSD=advert.sum()['SSD']
7 n = len(advert["Sales"])
8 print("n",n)
9 p = 2
10 RSE=np.sqrt(SSD/(n-p-1))
11 print("RSE", RSE)
12 salesmean=np.mean(advert['Sales'])
13 print("salesmean", salesmean)
14 error=RSE/salesmean
15 print("error", error)

```

[.]

```

1 n 200
2 RSE 1.68136091251
3 salesmean 14.022500000000003
4 error 0.119904504369

```

El valor para este modelos es  $\approx 1.68(12\%)$ , el cual es mucho mejor que el  $22 \sim 23\%$  de los modelos anteriores.

### Modelo 3: 'Ventas~TV+Radio+Newspaper'

**Problema 8.1.** Desarrolle un modelo para

"Ventas"~"TV+Radio+Periódico";

haga un análisis de los estadísticos asociados. Con base en estas observaciones, trata de responder porque el modelo resulta poco beneficiado de la incorporación del predictor "Periódico".

*Conclusiones sobre el modelo*

- Existe un coeficiente negativo pequeño para el periódico. Cuando consideramos sólo TV y periódico, el coeficiente de periódico fue significativamente positivo.
- Para este modelo, el estadístico  $-F$  ha decrecido considerablemente de 859.6 a 570.3.
- Sin embargo, el valor RSE se incremento aunque de manera modesta.

Con todas estas consideraciones, concluimos que la incorporación del periódico al modelo es poco eficiente (¿porqué?).

### Multicolinealidad

La *multicolinealidad* es la razón para el desempeño subóptimo del modelo cuando el predictor periódico es añadido al modelo final.

La multicolinealidad alude a la correlación entre los propios predictores del modelo.

Estas son algunas de las señales de este problema común encontrado durante la regresión lineal: Algunas páginas atrás, cuando creamos la *matriz de correlación* para este conjunto de datos, encontramos que existe una correlación importante de 0.35 entre el radio y el periódico.

Esto significa que el gasto en periódico esta relacionado con el gasto en radio. La relación entre predictores incrementa la variabilidad de los estimados de los coeficientes de las variables predictoras relacionadas.

El estadístico  $-t$  para este coeficiente es calculado al dividir el valor promedio por el la variabilidad. A medida que la variabilidad se incrementa, el valor del estadístico decrementa y entonces el valor  $-p$  crece.

Por lo cual la problemaabilidad de que la hipótesis nula asociada con el estadístico  $-F$  sea aceptado se incrementan. Esto reduce la significación del predictor en el modelo.

Por tanto, la colinealidad es un problema que debe ser tomado en cuenta. Para predictores altamente correlacionados, necesitamos hacer un análisis más a fondo con estas variables y ver cuales inclusiones en el modelo lo hacen más eficaz.

Es una buena práctica identificar parejas de predictores con alta correlación, usando la matriz de correlación y verificar el efecto de la multicolinealidad en el modelo. Las variables responsables deben de ser removidas del modelo: El *factor de inflación de varianza* (VIF, por sus siglas en inglés) es un método para abordar este problema.

**VIF (Variance Inflation Factor)** Es un método para cuantificar el aumento de la variabilidad del estimado del coeficiente de una variable particular debido a la alta correlación entre dos o más predictores.

El cuantificador VIF necesita ser calculado para cada una de las variables y si el valor es muy alto para una en particular, esta debe ser eliminada del modelo.

[] El siguiente es el proceso subyacente para calcular el valor VIF:

1. Calcule  $X_i$  como una función lineal de otras variables predictoras:

$$X_i = \sum_{j \neq i} a_j X_j \quad (8.28)$$

2. Calcule el valor  $R^2$  para este modelo y denótelos por  $R_i^2$ . El valor VIF para  $X_i$  está dado por

$$VIF = \frac{1}{1 - R_i^2} \quad (8.29)$$

3.
  - $VIF = 1$ : Los predictores no están correlacionados.
  - $1 < VIF < 5$ : Los predictores están moderadamente correlacionados con otros predictores y pueden seguir siendo parte del modelo.
  - $5 < VIF$ : Los predictores están altamente correlacionados y necesitan ser eliminados del modelo.

Podemos calcular los valores VIF asociados a cada predictor con el siguiente retazo de código:  
[.]VIF.py

```
1 modelN = smf.ols(formula = "Newspaper~TV+Radio", data =
  advert).fit()
2 r2N = modelN.rsquared
3 VIFN = 1/(1-r2N)
4 print("VIF(Newspaper):", VIFN)
5
6 modelT = smf.ols(formula = "TV~Newspaper+Radio", data =
  advert).fit()
7 r2T = modelT.rsquared
8 VIFT = 1/(1-r2T)
9 print("VIF(TV):", VIFT)
10
11 modelR = smf.ols(formula = "Radio~TV+Newspaper", data =
  advert).fit()
12 r2R = modelR.rsquared
13 VIFR = 1/(1-r2R)
14 print("VIF(Radio):", VIFR)
```

[.] Del cual obtenemos los siguientes resultados

```
1 VIF(Newspaper): 1.14518737872
2 VIF(TV): 1.00461078494
3 VIF(Radio): 1.14495191711
```

Los predictores Newspaper y Radio tienen prácticamente los mismos valores VIF, indicando que están correlacionados uno con el otro y no así con el predictor TV. En este caso, la radio y el periódico están fuertemente correlacionados. Sin embargo, el modelo con TV y Radio como predictores es mucho mejor que aquel con TV y Newspaper como tales. El modelo con las tres variables como predictores no mejora mucho el modelo. De hecho, incrementa la variabilidad y el estadístico  $-F$ .

Parece adecuado abandonar el predictor Newspaper del modelo y escoger el modelo 3 como el mejor candidato para el modelo final:

$$\text{Ventas} = 2.92 + 0.45 * \text{TV} + 0.18 * \text{Radio} \quad (8.30)$$

## 8.6 Validación del modelo

Cualquier modelo predictivo necesita ser validado para observar como es su rendimiento en diferentes conjuntos de datos y determinar si la precisión del modelo es constante todas fuentes de datos similares o no.

Esto nos ayuda a detectar algún problema de exceso de ajuste (over-fitting), en el que el modelo se ajusta muy bien en un conjunto de datos, pero no encaja bien en otro conjunto de datos. Un método común es crear un modelo dividiendo los datos en categorías de entrenamiento/prueba. Otro método es validación cruzada de  $k$  iteraciones, sobre la cual aprenderemos más en el capítulo posterior.

### División entre datos de entrenamiento y prueba

Idealmente, este paso debería hacerse justo al inicio del proceso de modelado para que no haya sesgos de muestreo en el modelo; en otras palabras, el modelo debería funcionar bien incluso para un conjunto de datos que tiene las mismas variables de predicción, pero sus medias y las varianzas son muy diferentes sobre de las que se ha construido el modelo.

Esto puede suceder porque el conjunto de datos en el que se basa el modelo (entrenamiento o capacitación) y el de que se aplica (prueba) puede provenir de diferentes fuentes. Una forma más robusta de hacer esto es un proceso llamado la validación cruzada de  $k$  iteraciones, sobre la cual hablaremos en detalle en un momento.

Veamos cómo podemos dividir el conjunto de datos disponible en el conjunto de datos de entrenamiento y prueba y aplicar el modelo al conjunto de datos de prueba para obtener otros resultados:  
[.]crossValidation.py

```
1 import pandas as pd
2 import statsmodels.formula.api as smf
3
```

```

4 advert = pd.read_csv("./dataBases/Advertising.csv")
5 model3=smf.ols(formula='Sales~TV+Radio',data=advert).fit()
6 sales_pred=model3.predict(advert[['TV','Radio']])
7 print(sales_pred.head())
8
9 print(model3.summary())
10
11 import numpy as np
12
13 N = len(advert)
14 arr = np.arange(N)
15 np.random.shuffle(arr)
16 check = arr < 0.8*N
17 training=advert[check].copy()
18 testing=advert[~check].copy()

```

Vamos a crear un modelo para entrenar los datos y problemaar el rendimiento del modelo en datos de prueba. Creemos el único modelo que funciona mejor (lo hemos encontrado ya), el que tiene variables de TV y radio, como variables de predicción:

```

1 import statsmodels.formula.api as smf
2 model5=smf.ols(formula='Sales~TV+Radio',data=training).fit()
3 print(model5.summary())

```

```
[.]print(model3.summary())
```

```

1                      OLS Regression Results
2  =====
3  Dep. Variable:          Sales    R-squared:
4                      0.897
5  Model:                  OLS      Adj. R-squared:
6                      0.896
7  Method:                 Least Squares    F-statistic:
8                      859.6
9  Date:                   Sun, 12 Nov 2017    problema (F-statistic
10 ):                      4.83e-98
11 Time:                   22:40:26    Log-Likelihood:
12                      -386.20
13 No. Observations:        200    AIC:
14                      778.4
15 Df Residuals:            197    BIC:
16                      788.3
17 Df Model:                 2
18 Covariance Type:         nonrobust
19 =====
20
21      coef    std err          t      P>|t|
22  [0.025    0.975]
23 -----
24 Intercept    2.9211    0.294     9.919    0.000
25      2.340     3.502
26 TV           0.0458    0.001    32.909    0.000
27      0.043     0.048
28 Radio        0.1880    0.008    23.382    0.000
29      0.172     0.204
30 =====
31
32 Omnibus:                60.022    Durbin-Watson:
33      2.081
34 problema(Omnibus):      0.000    Jarque-Bera (JB):
35      148.679

```

```

21 Skew:                    -1.323    problema(JB):
22 Kurtosis:                5.19e-33
23 =====
24
25 Warnings:
26 [1] Standard Errors assume that the covariance matrix of the
    errors is correctly specified.

    [,]

1      OLS Regression Results
2 =====
3 Dep. Variable:            Sales    R-squared:
4 Model:                    OLS      Adj. R-squared:
5 Method:                    Least Squares    F-statistic:
6 Date:                      Sun, 12 Nov 2017    problema (F-statistic
7 Time:                      22:40:26    Log-Likelihood:
8 No. Observations:          160    AIC:
9 Df Residuals:              157    BIC:
10 Df Model:                  2
11 Covariance Type:          nonrobust
12 =====
13      coef    std err          t      P>|t|
14 [0.025    0.975]
15 Intercept    2.8708    0.318    9.035    0.000
16 TV           0.0448    0.001   30.797    0.000
17 Radio        0.1959    0.009   22.803    0.000
18 =====
19 Omnibus:            11.888    Durbin-Watson:
20 problema(Omnibus):    0.003    Jarque-Bera (JB):
21 Skew:              0.00138
22 Kurtosis:          2.807    Cond. No.
23 =====
24
25 Warnings:
26 [1] Standard Errors assume that the covariance matrix of the
    errors is correctly specified.

```

La mayoría de los parámetros del modelo, como la intercepción, las estimaciones de coeficientes y  $R^2$  son muy similares.

La diferencia en el estadístico— $F$  se puede atribuir a un conjunto de datos más pequeño. Cuanto menor sea el conjunto de datos, mayor será el valor de SSD y menor será el valor de el término  $(n - p - 1)$  en la fórmula del estadístico— $F$ ; ambos contribuyen a la disminución en el valor estadístico— $F$ .

El modelo puede reescribirse como sigue:

$$\text{Ventas} \sim 2.86 + 0.04 * \text{TV} + 0.17 * \text{Radio} \quad (8.31)$$

[.] Ahora, predigamos los valores de las ventas para los valores de prueba:

```
1 sales_pred=model5.predict(training[['TV','Radio']])
2 sales_pred

El valor RSE para esta predicción en el conjunto de datos de prueba puede ser calculadas usando el
siguiente pedazo de código:
[.]

1 testing['sales_pred']=2.86 + 0.04*testing['TV'] + 0.17*
  testing['Radio']
2 n = len(testing)
3 p = 2
4 testing['SSD']=(testing['Sales']-testing['sales_pred'])**2
5 SSD=testing.sum()['SSD']
6 RSE=np.sqrt(SSD/(n-p-1))
7 salesmean=np.mean(testing['Sales'])
8 error=RSE/salesmean
9 print(RSE,salesmean,error)
10 ##2.33080393856 14.527500000000003 0.160440814907
```

El valor RSE resulta ser 2.54 sobre una venta promedio (en el conjunto de datos) de 14.80, lo cual es un error del 17%.

Podemos ver que el modelo no se generaliza muy bien en el conjunto de datos de prueba, ya que el valor RSE para el mismo modelo es diferente en los dos casos.

Implica un cierto grado de exceso de ajuste cuando tratamos de construir el modelo basado en todo el conjunto de datos.

El valor RSE con la división de pruebas de entrenamiento, aunque un poco mayor, es más confiable y replicable.

## 8.7 Resumen de modelos

Name	Definition	R2/Adj-R2	F-statistic	F-statistic (p-value)	RSE
Model 1	Sales ~ TV	0.612/0.610	312.1	1.47e <sup>-42</sup>	3.25 (23%)
Model 2	Sales ~ TV+Newspaper	0.646/0.642	179.6	3.95e <sup>-45</sup>	3.12(22%)
Model 3	Sales ~ TV+Radio	0.897/0.896	859.6	4.83e <sup>-98</sup>	1.71(12%)
Model 4	Sales ~ TV+Radio+Newspaper	0.897/0.896	570.3	1.58e <sup>-96</sup>	1.80(13%)

Figura 8.1: Guía para la selección de variables.

[.] Finalmente, para resumir, para un buen modelo lineal, los predictores deberían escogerse con base en los siguientes criterios:

- $R^2$ : Este valor siempre aumentará cuando se agregue una nueva variable de predicción al modelo. Sin embargo, no es una verificación muy confiable de la mayor eficiencia de el modelo. Más bien, para un modelo eficiente, debemos verificar el  $R^2$  ajustado. Esto debería aumentar al agregar una nueva variable de predicción.
- valor— $p$ : Cuanto menor sea el valor— $p$  para la estimación de la variable de predicción, mejor es agregar la variable de predicción al modelo.
- Estadístico— $F$ : El valor del estadístico— $F$  para el modelo debería aumentar después de la adición de una nueva variable de predicción para considerarse una adición eficiente al modelo. El aumento en el estadístico— $F$  es un buen indicador para la mejora en el modelo debida únicamente por la adición de ese variable particular. Alternativamente, el valor— $p$  asociado con el estadístico  $F$  debería disminuir al agregar una nueva variable de predicción.
- RSE: Este valor para el nuevo modelo debería disminuir al agregar la nueva variable de predicción.
- VIF: Para ocuparse de los Problemas que surgen debido a la colinealidad múltiple, se necesita eliminar las variables con grandes valores VIF.

## 8.8 Regresión lineal con *scikit-learn*

Vamos a implementar el modelo de regresión lineal utilizando el paquete *scikit-learn*. Este método es más elegante ya que tiene más métodos incorporados para realizar los procesos regulares asociados con la regresión.

Por ejemplo, podrías recordar del último capítulo que hay un método separado para dividir el conjunto de datos en entrenamiento y prueba de conjuntos de datos:

[.]*scikitExample.py*

```

1 advert = pd.read_csv("./dataBases/Advertising.csv")
2 feature_cols = ['TV', 'Radio']
3 X = advert[feature_cols]
4 Y = advert['Sales']
5 trainX, testX, trainY, testY = train_test_split(X, Y, test_size =
    0.2)
6 lm = LinearRegression()
7 lm.fit(trainX, trainY)

```

[.] El siguiente código nos devuelve los parámetros

```

1 print(lm.intercept_)
2 for _ in zip(feature_cols, lm.coef_):
3     print(_)

```

[.] El valor  $R^2$  se obtiene de la siguiente manera

```

1 print("R2 =", lm.score(trainX, trainY))

```

[.] Un resultado típico de este código sería

```

1 2.73465274245
2 ('TV', 0.046830472078714387)
3 ('Radio', 0.18642021416992249)
4 R2 = 0.893905959809

```

Los valores de  $R^2$  resultan alrededor del 89 %, muy cercanos al valor obtenido por el método usado anteriormente.

[.] El modelo puede ser usado para predecir el valor de las ventas usando los predictores TV y Radio del conjunto de datos de prueba, como sigue:

```

1 lm.predict(testX)

```

### Selección de características con *scikit-learn*

Muchas de las herramientas y paquetes estadísticos tienen métodos incorporados para llevar a cabo un proceso de selección de variables (selección hacia adelante y selección hacia atrás).

Si se hace manualmente, consumirá mucho tiempo y seleccionar los más importantes variables serán una tarea tediosa que compromete la eficiencia del modelo.

Una ventaja de usar el paquete *scikit-learn* para la regresión en Python es que tiene este método particular para la selección de características. Esto funciona más o menos como selección hacia atrás (no exactamente) y se llama eliminación de características recursivas, (RFE, por sus siglas en inglés). Se puede especificar el número de variables que desean en el modelo final.

El modelo se ejecuta primero con todas las variables y se asignan ciertos pesos a todos las variables. En las iteraciones siguientes, las variables con los pesos más pequeños se borran de la lista de variables hasta que se deja el número deseado de variables.

Vamos cómo se puede hacer una selección de funciones en *scikit-learn*:

[.]RFE.py

```

1 import pandas as pd
2 advert = pd.read_csv("./dataBases/Advertising.csv")
3
4 from sklearn.feature_selection import RFE
5 from sklearn.svm import SVR
6 feature_cols = ['TV', 'Radio', 'Newspaper']
7 X = advert[feature_cols]
8 Y = advert['Sales']
9 estimator = SVR(kernel="linear")
10 selector = RFE(estimator, 2, step=1)
11 selector = selector.fit(X, Y)

```

Usamos los métodos denominados RFE y SVR integrados en *scikit-learn*. Indicamos que queremos estimar un modelo lineal y que el número de variables deseadas en el modelo son dos.

[.] Para obtener la lista de variables seleccionadas, uno puede escribir el siguiente fragmento de código:

```

1 print(selector.support_)
2 ##[ True True False]

```

En nuestro caso, X consta de tres variables: TV, Radio y Newspaper. La matriz anterior sugiere que la TV y la radio se han seleccionado para el modelo, mientras que el periódico no ha sido seleccionado. Esto concuerda con la selección de variables que tuvimos hecho manualmente

[.] Este método también devuelve una clasificación, como se describe en el siguiente ejemplo:

```

1 print(selector.ranking_)
2 ##[1 1 2]

```

*Observación 8.4.* Todas las variables seleccionadas tendrán una clasificación de 1 mientras que las siguientes serán clasificadas en orden descendente respecto de su importancia. Una variable con rango 2 será más significativa para el modelo que la que tiene un rango de 3 y así sucesivamente.

## 8.9 Manejando otros Problemas en lineales regresión

Hasta ahora en este capítulo, hemos aprendido:

- Cómo implementar un modelo de regresión lineal usando dos métodos
- Cómo medir la eficiencia del modelo usando los parámetros del modelo

Sin embargo, hay otros Problemas que deben tenerse en cuenta al tratar con fuentes de datos de diferentes tipos. Repasemos uno por uno. Usaremos un diferente conjunto de datos simulado para ilustrar estos Problemas. Vamos a importarlo y echarle un vistazo en eso:

[.]handlingIssues.py

```
1 import pandas as pd
2 df=pd.read_csv('./dataBases/EcomExpense.csv')
3 print(df.head())
```

[.] Deberíamos obtener los siguientes resultados:

	Transaction ID	Age	Items	Monthly Income	Transaction
0	TXN001	42	10	7313	
1	TXN002	24	8	17747	
2	TXN003	47	11	22845	
3	TXN004	50	11	18552	
4	TXN005	60	2	14439	
	Gender	City	Tier	Total Spend	
0	Female		Tier 1	4198.385084	
1	Female		Tier 2	4134.976648	
2	Male		Tier 2	5166.614455	
3	Female		Tier 1	7784.447676	
4	Female		Tier 2	3254.160485	

[] La captura de pantalla anterior es un conjunto de datos simulados del sitio web de cualquier comercio. Esto captura la información sobre varias transacciones realizadas en el sitio web.

Una breve La descripción de los nombres de columna del conjunto de datos es la siguiente:

- Identificación de transacción: ID de transacción para la transacción
- Edad: Edad del cliente
- Artículos: número de artículos en el carrito de compras (comprado)
- Ingreso mensual: Ingreso disponible mensual del cliente
- Tiempo de transacción: tiempo total pasado en el sitio web durante la transacción
- Registro: cuántas veces el cliente ha comprado con el sitio web en el pasado
- Género: Género del cliente
- Nivel de la ciudad:
- Gasto total: monto total gastado en la transacción

La variable de salida es la variable `Total Spend` (Gasto total). Los otros son predictores potenciales variables y sospechamos que el gasto total está relacionado linealmente con todos estos variables predictoras.

## Manejando variables categóricas

Hasta ahora, hemos supuesto que las variables de predicción solo pueden ser cuantitativas o numéricas, pero sabemos por experiencias de la vida real que la mayoría de las veces *el conjunto de datos contiene una variable categórica o cualitativa* y muchas de las veces estas variables tendrá un impacto significativo en el valor de la salida. Sin embargo, la pregunta es *¿cómo procesar estas variables, para usarlas en el modelo?*

No podemos asignarles valores, como 0, 1, 2, etc., y luego usarlos en el modelo, ya que dará un peso excesivo a las categorías debido a los números asignado a ellos.

La mayoría de las veces puede dar un resultado incorrecto y cambiará, como cambia el número asignado a una categoría en particular.

En el marco de datos que acabamos de importar, Género y Nivel de ciudad son los categóricos variables.

Para manejar variables categóricas, usaremos variables “tontas” (dummies, en inglés) o ficticias.



Una regresión lineal es de la forma

$$Y_{\text{model}} = \alpha + \sum \beta_i X_i \quad (8.32)$$

en la cual algunas  $X_i$  pueden ser categóricas. Digamos que  $X_g$  es tal variable. En nuestro ejemplo,

$$X_g = \begin{cases} 1 & \text{cliente masculino} \\ 0 & \text{cliente femenino} \end{cases} \quad (8.33)$$

Si hay tres niveles en la variable categórica, entonces uno necesita definir dos variables en comparación con 1 cuando había dos niveles en la variable categórica.

Por ejemplo, la variable City Tier tiene tres niveles en nuestro conjunto de datos. Para esto, podemos definir dos variables, tales que:

$$X_{t1} = \begin{cases} 1 & \text{"City Tier" = 1} \\ 0 & \text{"City Tier" \neq 1} \end{cases} \quad (8.34)$$

$$X_{t2} = \begin{cases} 1 & \text{"City Tier" = 2} \\ 0 & \text{"City Tier" \neq 2} \end{cases} \quad (8.35)$$

Entonces, el modelo puede ser alguno de los siguientes

$$Y = \begin{cases} \alpha + \beta_1 X_1 + \dots + \beta_{t1} X_{t1} + \dots + b_n X_n & \text{"City Tier"=1} \\ \alpha + \beta_1 X_1 + \dots + \beta_{t1} X_{t1} + \dots + b_n X_n & \text{"City Tier"=2} \\ \alpha + \beta_1 X_1 + \dots + b_n X_n & \text{"City Tier"=3} \end{cases} \quad (8.36)$$

Tengan en cuenta que uno no tiene que crear la tercera variable. Esto es debido a la naturaleza en que se definen estas variables.

Si un el cliente no pertenece a la ciudad de nivel 1 o nivel 2, entonces ciertamente lo hará pertenece a una ciudad de nivel 3. Por lo tanto, no se requiere una variable para uno de los niveles.

*Observación 8.5.* En general, para variables categóricas que tienen  $n$  niveles, uno debería crear  $(n - 1)$  variables ficticias.

Sin embargo, por simplicidad, utilizaremos cada uno de los niveles.

Creemos ahora las variables ficticias para nuestras variables categóricas y luego agreguémoslas a nuestro marco de datos, como se muestra:

```
1 dummy_gender=pd.get_dummies(df['Gender'],prefix='Sex')
2 dummy_city_tier=pd.get_dummies(df['City Tier'],prefix='City')
```

Veamos cómo se ven y si satisfacen las condiciones que hemos definido antes o no. Así es como se ve dummy\_city\_tier:

```
1 City_Tier 1 City_Tier 2 City_Tier 3
2 0          1          0          0
3 1          0          1          0
4 2          0          1          0
5 3          1          0          0
6 4          0          1          0
```

[.] dummy\_gender es similar a la siguiente tabla:

```
1 Sex_Female Sex_Male
2 0          1          0
3 1          1          0
4 2          0          1
5 3          1          0
6 4          1          0
```

Ahora, tenemos estas variables ficticias creadas pero no son parte del marco principal de datos todavía. Vamos a adjuntar estas nuevas variables al marco de datos principal para que se puede usar en el modelo:

```
1 Transaction ID Age Items Monthly Income Transaction
   Time Record \
2 0 TXN001 42 10 7313
   627.668127 5
3 1 TXN002 24 8 17747
   126.904567 3
```

```

4 2      TXN003      47      11      22845
      873.469701      2
5 3      TXN004      50      11      18552
      380.219428      7
6 4      TXN005      60      2      14439
      403.374223      2
7
8  Gender City Tier  Total Spend  Sex_Female  Sex_Male
   City_Tier 1  \
9 0  Female   Tier 1  4198.385084          1          0
      1
10 1  Female   Tier 2  4134.976648          1          0
      0
11 2   Male   Tier 2  5166.614455          0          1
      0
12 3  Female   Tier 1  7784.447676          1          0
      1
13 4  Female   Tier 2  3254.160485          1          0
      0
14
15  City_Tier 2  City_Tier 3
16 0           0           0
17 1           1           0
18 2           1           0
19 3           0           0
20 4           1           0

```

Hay cinco nuevas columnas en el marco de datos, dos de las variables ficticia de **Gender** y tres de las variables ficticias de **City Level**.

Si lo compara con el conjunto de datos completo, **City\_Tier\_1** tiene el valor 1 si **City\_Tier** tiene valor de Tier 1, **City\_Tier\_2** tiene valor 1 si **City\_Tier** tiene valor de Tier 2 y **City\_Tier\_3** tiene valor 1 si **City\_Tier** tiene valor Tier 3. Todas las demás variables ficticias en esa fila particular tendrán valores 0. Esto es lo que queríamos.

Vamos cómo incluir estas variables ficticias en el modelo y cómo evaluarlas sus coeficientes.

Para el conjunto de datos anterior, supongamos una relación lineal entre la salida variable de Gasto Total y las variables predictoras: Ingreso Mensual y Tiempo de transacción, y ambos conjuntos de variables ficticias:

[.]

```

1 from sklearn.linear_model import LinearRegression
2 feature_cols = ['Monthly Income', 'Transaction Time', '
      City_Tier 1', \
3               'City_Tier 2', 'City_Tier 3', 'Sex_Female', '
      Sex_Male']
4 X = df2[feature_cols]
5 Y = df2['Total Spend']
6 lm = LinearRegression()
7 lm.fit(X,Y)

```

[.] Los parámetros del modelo se pueden encontrar de la siguiente manera:

```

1 print(lm.intercept_)
2 for _ in zip(feature_cols, lm.coef_):
3     print(_)
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

[.] El valor  $R^2$  para este modelo se puede encontrar escribiendo lo siguiente:

```

1 R2 = lm.score(X,Y)
2 print(R2)

```

```
3 ## 0.194789205529
```

El valor resulta ser 0.19, lo que podría deberse a que no hemos usado las otras variables y el resultado también podrían estar relacionados con ellos.  
 Necesitamos ajustar el modelo al transformar adecuadamente algunas de las variables y agregarlas al modelo.  
 Por ejemplo, si agregamos la variable Record al modelo,  $R^2$  salta a 0.91 (intente eso por su cuenta).  
 Es un buen conjunto de datos para jugar.  
 El modelo se puede escribir de la siguiente manera:

$$\text{Total\_Spend} = 3655.72 + 0.12 * \text{Transaction Time} \quad (8.37)$$

$$+ 0.15 * \text{Monthly Income} + 119 * \text{City\_Tier 1} \quad (8.38)$$

$$- 16 * \text{City\_Tier 2} - 102 * \text{City\_Tier 3} \quad (8.39)$$

$$- 94 * \text{Sex\_Female} + 94 * \text{Sex\_Male} \quad (8.40)$$

[.] El RSE se puede calcular de la siguiente manera:

```
1 import numpy as np
2 df2['total_spend_pred']=3720.72940769 + 0.12*df2['Transaction
   Time']+ \
3     0.15*df2['Monthly Income']+119*df2['City_Tier 1']-16*df2[
   'City_Tier 2']
4 -102*df2['City_Tier 3']-94*df2['Sex_Female']+94*df2['Sex_Male
   ']
5 df2['RSE']=(df2['Total Spend']-df2['total_spend_pred'])**2
6 RSEd=df2.sum()['RSE']
7 RSE=np.sqrt(RSEd/2354)
8 salesmean=np.mean(df2['Total Spend'])
9 error=RSE/salesmean
10 print(RSE,salesmean,error)
11 ##2518.85203887 6163.176415976714 0.408693808008
```

Para diferentes niveles de género y ciudad, el modelo se reducirá a seguir para diferentes casos:

Gender	City Tier	Model
Male	1	$\text{Total\_Spend} = 3655.72 + 0.12 * \text{Transaction Time} + 0.15 * \text{Monthly Income} + 119 * \text{City\_Tier 1} + 94 * \text{Sex\_Male}$
Male	2	$\text{Total\_Spend} = 3655.72 + 0.12 * \text{Transaction Time} + 0.15 * \text{Monthly Income} - 16 * \text{City\_Tier 2} + 94 * \text{Sex\_Male}$
Male	3	$\text{Total\_Spend} = 3655.72 + 0.12 * \text{Transaction Time} + 0.15 * \text{Monthly Income} - 102 * \text{City\_Tier 3} + 94 * \text{Sex\_Male}$
Female	1	$\text{Total\_Spend} = 3655.72 + 0.12 * \text{Transaction Time} + 0.15 * \text{Monthly Income} + 119 * \text{City\_Tier 1} - 94 * \text{Sex\_Female}$
Female	2	$\text{Total\_Spend} = 3655.72 + 0.12 * \text{Transaction Time} + 0.15 * \text{Monthly Income} - 16 * \text{City\_Tier 2} - 94 * \text{Sex\_Female}$

## Transformando una variable para ajustarla a una relación no lineal

A veces, la variable de salida no tiene una relación lineal directa con el variable de predicción, es decir, tienen una relación no lineal.

Estas relaciones podrían funciones simples como cuadrática, exponencial, logaritmo o complejas como polinomios. En tales casos, la transformación de la variable es muy útil.

La siguiente es una guía aproximada sobre cómo hacerlo:

1. Trace un diagrama de dispersión de la variable de salida con cada uno de los predictores variables. Este se puede pensar como una matriz de diagrama de dispersión similar a la matriz de correlación
2. Si la gráfica de dispersión asume más o menos una forma lineal para una variable de predicción entonces está relacionado linealmente con la variable de salida.
3. Si el diagrama de dispersión asume una forma característica de diferente de la lineal para una variable de predicción, entonces transformaremos esa variable en particular aplicando esa función.

Vamos a ilustrar esto con un ejemplo. Usaremos el conjunto de datos Auto.csv para esto. Este conjunto de datos contiene información sobre millas por galón (mpg) y caballos de fuerza para una serie de modelos de automóviles y mucho más. El mpg es la variable de predicción y es considerado altamente dependiente de la potencia de un modelo de automóvil.

[.]nonLinear.py

```
1 import pandas as pd
2 data = pd.read_csv('./dataBases/Auto.csv')
3 print(data.head())
```

[.]

```

1      mpg  cylinders  displacement  horsepower  weight
   acceleration \
2 0  18.0          8          307.0        130.0    3504
   12.0
3 1  15.0          8          350.0        165.0    3693
   11.5
4 2  18.0          8          318.0        150.0    3436
   11.0
5 3  16.0          8          304.0        150.0    3433
   12.0
6 4  17.0          8          302.0        140.0    3449
   10.5
7
8      model year  origin          car name
9 0          70      1  chevrolet chevelle malibu
10 1          70      1      buick skylark 320
11 2          70      1  plymouth satellite
12 3          70      1      amc rebel sst
13 4          70      1      ford torino

```

Tiene 406 filas y 9 columnas. Algunas de las variables tienen valores de NA (not available) y tiene sentido eliminar los valores NA antes de usarlos.

[.] Ahora, tracemos un diagrama de dispersión entre las variables de potencia y mpg para ver ya sea que exhiban una forma lineal o alguna forma no lineal:

```

1 import matplotlib.pyplot as plt
2 %matplotlib inline
3 data['mpg']=data['mpg'].dropna()
4 data['horsepower']=data['horsepower'].dropna()
5 plt.plot(data['horsepower'],data['mpg'],'ro')
6 plt.xlabel('Horsepower')
7 plt.ylabel('MPG (Miles Per Gallon)')

```

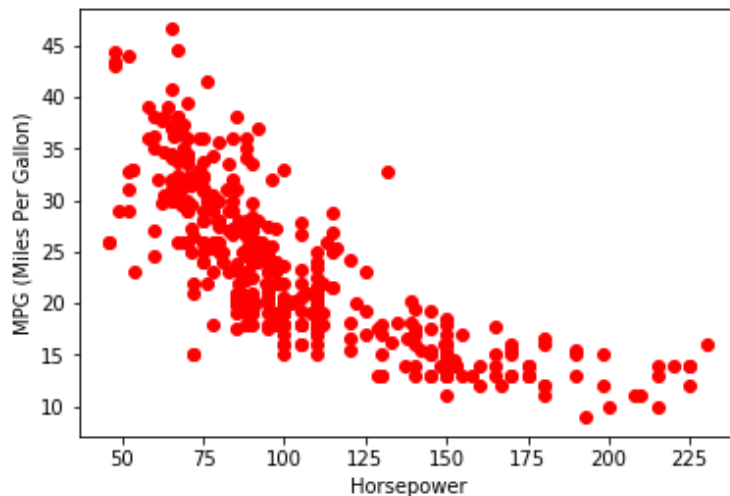


Figura 8.2: HP vs MPG

Aunque hemos supuesto que el modelo es lineal

$$mpg = c_0 + c_1 * hp \quad (8.41)$$

en realidad parece más un *modelo cuadrático*

$$mpg = c_0 + c_1 * hp + c_2 hp^2 \quad (8.42)$$

El siguiente fragmento de código se ajustará a un modelo lineal entre potencia y mpg variables. Los valores de NA deben eliminarse de las variables antes de que puedan ser utilizados en el modelo. También simultáneamente, creemos un modelo asumiendo una lineal relación entre mpg y cuadrado de potencia:

[.]

```

1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3 X=data['horsepower'].fillna(data['horsepower'].mean())
4 Y=data['mpg'].fillna(data['mpg'].mean())
5 lm=LinearRegression()
6 lm.fit(X[:,np.newaxis],Y)

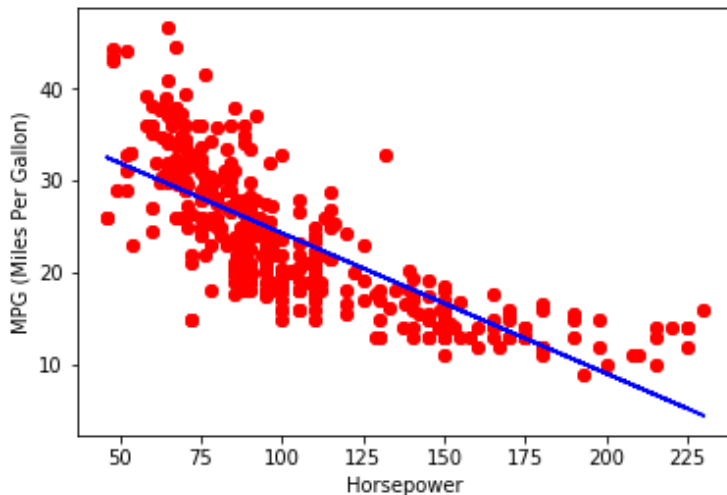
```

El método de regresión lineal por defecto requiere que X sea una matriz de dos dimensiones. Usando np.newaxis, estamos creando una nueva dimensión para que funcione correctamente. La línea de mejor ajuste se puede trazar con el siguiente fragmento:

```

1 import matplotlib.pyplot as plt
2 %matplotlib inline
3 plt.plot(data['horsepower'],data['mpg'],'ro')
4 plt.plot(X,lm.predict(X[:,np.newaxis]),color='blue')

```



[.]

```

1 R2 = lm.score(X[:,np.newaxis],Y)
2 print(R2)
3 ## 0.574653340645
4
5 RSEd=(Y-lm.predict(X[:,np.newaxis]))**2
6 RSE=np.sqrt(np.sum(RSEd)/389)
7 ymean=np.mean(Y)
8 error=RSE/ymean
9 print(RSE,error)
10 ## 5.1496254787 0.21899719414

```

Aquí, estamos usando el método predict para calcular el valor predicho de la modelo en lugar de escribirlos explícitamente. El valor de RSE para este modelo resulta ser 5.14, que sobre un valor medio de 23.51 da un error del 21 %. Si el modelo es cuadrático, esto se puede ajustar utilizando el método PolynomialFeatures en la biblioteca scikit-learn. En este modelo, asumimos una relación polinómica entre mpg y caballos de fuerza:

[.]nonLinear.py

```

1 from sklearn.preprocessing import PolynomialFeatures
2 from sklearn import linear_model
3 X=data['horsepower'].fillna(data['horsepower'].mean())
4 Y=data['mpg'].fillna(data['mpg'].mean())
5 poly = PolynomialFeatures(degree=2)
6 X_ = poly.fit_transform(X[:,np.newaxis])
7 clf = linear_model.LinearRegression()
8 clf.fit(X_, Y)

```

```

9
10 print(clf.intercept_)
11 ##55.0261924471
12 print(clf.coef_)
13 ##[ 0.          -0.43404318  0.00112615]

```

El modelo se puede escribir entonces como

$$\text{mpg} = 55.02 - 0.43 * \text{hp} + 0.001 * \text{hp}^2 \quad (8.43)$$

[.] Con el siguiente fragmento de código, podemos visualizar el modelo cuadrático:

```

1 plt.plot(data['horsepower'], data['mpg'], 'ro')
2 plt.plot(X, clf.predict(X_), "bo")
3 plt.show()

```

