JULIHO CASTILLO COLMENARES

MODELACIÓN ESTADÍS-TICA

WWW.STEMPUNK.XYZ

Índice general

1 Estadística descriptiva

1.1 Medidas de tendencia central

Índice y subíndices

El símbolo X_j representa cualquiera de los valores $X_1, X_2, X_3, ...$ que puede tomar la variable discreta X.

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

Definición (Sumatoria)**.**

$$\sum_{j=1}^{N} X_j = X_1 + ... + X_N \tag{1.1}$$

Ejemplo 1.1.1. ▪ $\sum_{k=1}^{N} X_k Y_k = X_1 Y_1 + ... + X_N Y_N$

- $\sum_{i=1}^{N} aX_i = aX_1 + ... + aX_N = a\sum_{n=1}^{N} X_n.$
- Si a, b son constantes, demuestre que

$$\sum (aX + bY) = a\sum X + b\sum Y. \tag{1.2}$$

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

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

Media aritmética

Definición (Media aritmética)**.**

$$\bar{X} = \frac{X_1 + ... + X_N}{N} = \frac{\sum_{j=1}^{N} X_j}{N} = \frac{\sum X}{N} \tag{1.3}$$

Ejemplo 1.1.2. Calcula la media de $8, 3, 5, 12, 10$.

```
"""
La media aritm'etica de 8,3,5,12,10 es...
"""
data = [8,3,5,12,19]
n = len(data)
media_aritmetica = sum(data)/n
print(media_aritmetica) ## 9.4
```

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

$$\bar{X} = \frac{f_1X_1 + ... + f_kX_k}{f_1 + ... + f_k} = \frac{\sum fX}{\sum f} = \frac{\sum fX}{N}. \tag{1.4}$$

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

Ejemplo 1.1.3. Si $5, 8, 6, 2$ se presentan con frecuencias $3, 2, 4, 1$ respectivamente, su media aritmética es...

```
"""
Si 5,8,6,2 se presentan con frecuencias 3,2,4,1
    respectivamente, su media aritm'etica es...
"""
import numpy as np
data = np.array([5,8,6,2])
freq = np.array([3,2,4,1])
n = np.sum(freq)
media = np.sum(freq*data)/n
print(media) # 5.7
```

Algunas veces, a los números $X_1, ..., X_k$ se les asignan ciertos *factores de ponderación* o *pesos* $w_1, ..., w_k$, tales que

$$\begin{cases} 0\,\% \leq w_i \leq 100\,\% \\ \sum w_i = 100\,\% \end{cases} \tag{1.5}$$

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

$$\bar{X} = \frac{w_1 X_1 + ... + w_k X_k}{w_1 + ... + w_k} = \frac{\sum wX}{\sum w} = \sum wX. \tag{1.6}$$

Ejemplo 1.1.4. 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).

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

$$\bar{d} = \frac{\sum f_i d_i}{N}, \tag{1.7}$$

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

Observación. 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 1.1.5.
- La mediana de la lista de números $5, 4, 3, 8, 6, 2, 5, 2$ es...
- La mediana de la lista de números $3, 9, 1, 1, 4, 1, 3, 2, 4$ es..

```
"""
La mediana de la lista de n'umeros 5, 4, 3, 8, 6, 2, 5, 2 es
    ..
"""
data = [5, 4, 3, 8, 6, 2, 5, 2]
#ordenamos los datos
data = sorted(data)
print(data) # [2, 2, 3, 4, 4, 5, 6, 8]
n = len(data)
pos_mediana = (n-1)/2
print(pos_mediana) # 3.5
m = (n-1)//2 # 3
```

```
mediana = (data[m]+data[m+1])/2
print(mediana) # 4.5

"""
La mediana de la lista de n'umeros 3, 9, 1, 1, 4, 1, 3, 2, 4
    es..
"""
data = [3, 9, 1, 1, 4, 1, 3, 2, 4]
#ordenamos los datos
data = sorted(data)
print(data) # [1, 1, 1, 2, 3, 3, 4, 4, 9]
n = len(data)
pos_mediana = (n-1)/2
print(pos_mediana) # 4
m = (n-1)//2
mediana = data[m]
print(mediana) # 3
```

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

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

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 1.1.6.
- 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 (Moda para datos agrupados)**.**

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

donde

1. L : Frontera inferior de la clase modal, es decir, de la clase que contiene la moda.
2. Δ_1 : Exceso de frecuencia modal sobre la frecuencia en la clase inferior inmediata.
3. Δ_2 : Exceso de frecuencia modal sobre la frecuencia en la clase superior inmediata.
4. c : Amplitud del intervalo de la clase modal.

Paquetes especializados

`Numpy` es el módulo numérico de Python. Nos permite hacer cálculos numéricos con gran velocidad. Con `Numpy`, es sencillo calcular la media aritmética sobre los elementos de un arreglo.

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

También podemos calcular la mediana.

```
import numpy as np

a = np.array([[10, 7, 4], [3, 2, 1]])
print a
#array([[10,  7,  4],6[ 3,  2,  1]])
print np.median(a)
#3.5
print np.median(a, axis=0)
#array([ 6.5,  4.5,  2.5])
print np.median(a, axis=1)
#array([ 7.,  2.])

m = np.median(a, axis=0)
out = np.zeros_like(m)
print np.median(a, axis=0, out=m)
#array([ 6.5,  4.5,  2.5])
print m
#array([ 6.5,  4.5,  2.5])
b = a.copy()
print np.median(b, axis=1, overwrite_input=True)
#array([ 7.,  2.])

assert not np.all(a==b)
b = a.copy()
print np.median(b, axis=None, overwrite_input=True)
#3.5
assert not np.all(a==b)
```

Sin embargo, no hay una forma sencilla de calcular la moda con `Numpy`. Por lo que usaremos otro módulo llamado `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.[1]

[1] https://es.wikipedia.org/wiki/SciPy

```
import numpy as np
from scipy import stats

a = np.array([3,5,6,5,6,5,6,6,3,1,5])
print stats.mode(a)
# ModeResult(mode=array([5]), count=array([4]))
```

Problemas

Problema 1.1.1. Escribir los términos de cada una de las siguientes sumas:

1. $\sum_{j=0}^{6} X_j =$
2. $\sum_{k=1}^{4} (Y_k - 3)^2 =$
3. $\sum_{k=1}^{N} a =$
4. $\sum_{n=2}^{5} f_n X_n =$
5. $\sum_{m=0}^{3} (X_m - a) =$

Problema 1.1.2. De 100 números, 20 fueron 4, 40 fueron 5, 30 fueron 6 y los restantes fueron 7. Encuentre su media aritmética.

Problema 1.1.3. 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 1.1.4. 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.

Problema 1.1.5. Si las desviaciones de N números $X_1, .., X_N$ respecto a un *pivote* P están dada por $d_i = X_i - P,\ i = 1, ..., N$ respectivamente, demostrar que

$$\bar{X} = P + \frac{\sum d}{N}. \tag{1.10}$$

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

Problema 1.1.7. Si $Z_i = X_i + Y_i,\ i = 1, 2, ..., N$, demostrar que $\bar{Z} = \bar{X} + \bar{Y}$.

Problema 1.1.8. 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 1.1.9. Utilice la marca de la clase media como pivote, para calcular la estatura de los estudiantes en la tabla 1.1.4.

Problema 1.1.10. 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 1.2.1. 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 $X_1, ..., X_N$ está definida como

$$DM = \frac{\sum |X_j - \bar{X}|}{N} \tag{1.11}$$

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

Ejemplo 1.2.2. Encuentre la desviación media de la lista $2, 3, 6, 8, 11$.

Desviación estándar

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

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

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

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

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

Observación. En ocasiones, N se reemplaza por $N - 1$ en las fórmulas anteriores, debido a que está 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. En estadística, es importante distinguir entre la desviación estándar de una *población* y una *muestra*. Para distinguirla, en el primer caso utilizaremos σ y en el segundo, continuaremos usando s.

Método abreviado

$$s^2 = \overline{X^2} - \overline{X}^2 \tag{1.14}$$
$$= E(X^2) - (E(X))^2 \tag{1.15}$$

En las distribuciones normales se tiene que

1. 68.27 % de los datos está comprendido entre $\bar{X} \pm s$.
2. 95.45 % de los datos está comprendido entre $\bar{X} \pm 2s$.
3. 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

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

Teorema de Chebyshev

Para $k > 1$, por lo menos $1 - \frac{1}{k^2}$ de la distribución de problemaabilidad de cualquier variable aleatoria está a nomas de k desviaciones estándar de la media.

Python

[]`numpy.std`

```
numpy.std(a, axis=None, dtype=None, out=None, ddof=0,
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.[2]

[2] https://docs.scipy.org/doc/numpy/reference/generate

[]

```
import numpy as np

a = np.array([[1, 2], [3, 4]])
print np.std(a)
#1.1180339887498949
print np.std(a, axis=0)
#array([ 1.,  1.])
print np.std(a, axis=1)
#array([ 0.5,  0.5])
```

[]

```
#In single precision, std() can be inaccurate:
a = np.zeros((2, 512*512), dtype=np.float32)
a[0, :] = 1.0
a[1, :] = 0.1
print np.std(a)
#0.45000005

#Computing the standard deviation in float64
#is more accurate:
print np.std(a, dtype=np.float64)
#0.44999999925494177
```

Problema 1.2.1. Encontrar el rango y las desviaciones media y estándar de los arreglos

1. $12, 6, 7, 3, 15, 10, 18, 5$
2. $9, 3, 8, 8, 9, 8, 9, 18$.

Compruebe sus resultados con `Python`.

Problema 1.2.2. 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 1.2.3. Encontrar las desviaciones media y estándar de las estaturas de 100 estudiantes de la siguiente tabla:

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

Problema 1.2.4. Demostrar que

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

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

Problema 1.2.5. Utilizando las fórmulas anteriores, encuentre la desviación estándar de los datos en la tabla 1.2.3:

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

Problema 1.2.6. Si $d = X - P$ son desviaciones de X respecto a un pivote P, demostrar que

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

Problema 1.2.7. Utilizando las fórmulas anteriores, encuentre la desviación estándar de los datos en la tabla 1.2.3:

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

Problema 1.2.8. Encuentre la media aritmética y la desviación estándar de los siguientes datos:

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

2 Probabilidad

2.1 Notación de conjuntos

Introducción

En esta actividad resolveremos problemas similares al de la figura 2.1.

Abordaremos el problema con el siguiente plan:

1. Definir una partición en un conjunto.
2. Fijar una partición estándar para describir las operaciones entre dos conjuntos.
3. Describir las operaciones entre conjuntos utilizando esta partición.
4. Plantear y resolver el problema en términos de dicha partición.

Como prerrequisitos, se necesitarán conocer los siguientes conceptos:

1. Conjuntos y subconjuntos.
2. Operaciones entre conjuntos.
3. Solución de sistemas de ecuaciones lineales.

Para mayor claridad, recordaremos algunas definiciones de subconjuntos $A, B \subset S$:

- Unión
$$A \cup B = \{x \in S | \, x \in A \ \texttt{o} \ x \in B\}$$
- Intersección
$$A \cap B = \{x \in S | \, x \in A \ \texttt{y} \ x \in B\}$$
- Resta
$$A \backslash B = \{x \in S | \, x \in A \ \texttt{y} \ x \notin B\}$$

En ocasiones también denotamos la resta por $A - B$.

- Complemento
$$A' = \{x \in S | \, x \notin A\}$$
A veces el complemento también se denota por A^c o $\bar{A}$. De manera equivalente, se puede escribir como $S \backslash A$.

- Diferencia simétrica
$$A \triangle B = \{x \in S | \, x \in A \ \texttt{o} \ x \in B \ \texttt{pero} \ x \notin A \cap B\}$$
.

Particiones

Consideremos un conjunto S. Una partición (finita) es una colección $\{P_i \subset S\}_{i=0}^N, N \in \mathbb{N}$ de subconjuntos de S que satisface

1. $P_i \cap P_j = \emptyset$ siempre que $i \neq j$;
2. $\bigcup_{i=0}^N P_i = S$.

En otras palabras, son subconjuntos disjuntos entre sí que, al unirse todos, forman de nuevo el conjunto S. El lector puede pensarlos como piezas de un rompecabezas.

Consideremos dos conjuntos $A, B \subset S$. El diagrama de Ven correspondiente esta dado por la figura 2.3.

Entonces podemos definir una partición $\mathfrak{P}(A, B)$ para S con los siguientes elementos

- $A \cap B$
- $A \backslash B = A \cap B'$
- $B \backslash A = B \cap A'$
- $(A \cup B)' = A' \cap B'$

Para hacer la notación más concisa, definimos las siguientes aplicaciones:

$$\delta_C(x) = \begin{cases} 1 & x \in C \\ 0 & x \notin C \end{cases} \tag{2.1}$$

donde 1 denota `verdadero`, mientras que 0 denota `falso`, y

$$E_{(i,j)} = \{x \in S | \, \delta_A(x) = i \wedge \delta_B(x) = j\} \tag{2.2}$$

De manera que

- $A' \cap B' = E_{(0,0)}$
- $A' \cap B = E_{(0,1)}$

- $A \cap B' = E_{(1,0)}$
- $A \cap B = E_{(1,1)}$

Para hacer aún más sencilla la notación, identificaremos la pareja (i, j) con el correspondiente número binario $[ij]_2$, convertido a base 10. De manera que

- $A' \cap B' = E_0$
- $A' \cap B = E_1$
- $A \cap B' = E_2$
- $A \cap B = E_3$

Planteamiento del problema

Como los elementos de un partición son disjuntos, entonces sabemos que

$$\#\left(E_i \cup E_j\right) = \#E_i + \#E_j \tag{2.3}$$

siempre que $i \neq j$.

Para simplificar la notación, definimos $x_i = \#E_i$.

La primera ecuación que se nos plantea es

$$\left(A \triangle B\right)' = 140, \tag{2.4}$$

donde

$$A \triangle B = \left(A \backslash B\right) \cup \left(B \backslash A\right) \tag{2.5}$$

es la diferencia simétrica de A con B. En otras palabras $x \in A \triangle B$ si y solo si $x \in A$ o $x \in B$ pero no en ambos.

Observa que entonces

$$\left(A \triangle B\right)' = E_0 \cup E_3 \Rightarrow x_0 + x_3 = 140. \tag{2.6}$$

De manera similar, obtenemos las siguientes conclusiones:

$$A \cup B = E_1 \cup E_2 \cup E_3 \Rightarrow x_1 + x_2 + x_3 = 177 \tag{2.7}$$

$$A = E_2 \cup E_3 \Rightarrow x_2 + x_3 = 144 \tag{2.8}$$

$$A' = E_0 \cup E_1 \Rightarrow x_0 + x_1 = 99 \tag{2.9}$$

Al resolver el sistema de ecuaciones dado por 2.6-2.9, obtenemos la solución:

$$x_0 = 66 \tag{2.10}$$

$$x_1 = 33 \tag{2.11}$$

$$x_2 = 70 \tag{2.12}$$

$$x_3 = 74 \tag{2.13}$$

Solución del problema

A continuación presentamos el desarrollo y conclusión de cada una de las preguntas en nuestro problema. Por ejemplo, podemos describir el complemento de B en términos de nuestra partición y utilizar las soluciones anteriores.

$$\begin{aligned} \#B' &= \#\left(E_1 \cup E_3\right)' && (2.14)\\ &= \#\left(E_0 \cup E_2\right) && (2.15)\\ &= x_0 + x_2 = 153 && (2.16) \end{aligned}$$

El resto de las soluciones se encuentras de la siguiente manera:

$$\begin{aligned} \#\left(A \triangle B\right) &= x_1 + x_2 = 103 && (2.17)\\ \#\left(A \backslash B\right) &= x_2 = 70 && (2.18)\\ \#\left(B\right) &= x_1 + x_3 = 107 && (2.19)\\ \#\left(A \cap B\right) &= x_3 = 74 && (2.20)\\ \#\left((B \backslash A)'\right) &= x_0 + x_2 + x_3 = 210 && (2.21)\\ \#\left((A \backslash B)'\right) &= x_0 + x_1 + x3 = 173 && (2.22)\\ \#\left(\emptyset\right) &= 0 && (2.23)\\ \#\left(S\right) &= x_0 + x_1 + x_2 + x_3 = 243 && (2.24)\\ \#\left(A' \cap B'\right) &= x_0 = 66 && (2.25)\\ \#\left(B \backslash A\right) &= x_1 = 33 && (2.26) \end{aligned}$$

Con esto concluimos nuestro ejercicio.

Problemas

Problema 2.1.1. Extiende la construcción de una partición al caso de tres subconjuntos.

2.2 *Fundamentos de probabilidad*

Experimentos aleatorios

Ejemplo 2.2.1. Si lanzamos una moneda, el resultado del experimento será reverso (*tail* en inglés) que simbolizaremos con la letra "T.[el] número 0 o anverso (*head* en inglés) simbolizado por H o 1, es decir, uno de los elementos del conjunto $\{T, H\}$ (o bien $\{0, 1\}$.)

Ejemplo 2.2.2. Si lanzamos un dado, el resultado del experimento resultará en uno de los números del conjunto $\{1, 2, 3, 4, 5, 6\}$.

Ejemplo 2.2.3. Si lanzamos una moneda dos veces, existen cuatro posibles resultados:

$$\{HH, HT, TH, TT\}. \tag{2.27}$$

Ejemplo 2.2.4. 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

$$\{\texttt{defectuoso, no defectuoso}\} \tag{2.28}$$

Ejemplo 2.2.5. 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

$$0 \leq t \leq T, \tag{2.29}$$

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 elegiremos el que nos provee la mayor información.

Ejemplo 2.2.6. 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 2.2.7. Si lanzamos una moneda dos veces seguidas un posible espacio muestral esta dado en el ejemplo 2.2.3, mientras que otro esta dado por

$$\{(0,0), (0,1), (1,0), (1,1)\}. \tag{2.30}$$

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 2.2.8. Si lanzamos una moneda dos veces, el evento de que obtengamos *exactamente* un reverso es un subconjunto del espacio muestral mostrado en la figura 2.4.

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 $A, B \subset S$ 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. Si $A_1, A_2, ... \subset S$ 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. Si $A_1, A_2, ... \subset S$ son eventos mutuamente excluyentes diremos que $A_1 \cup A_2 \cup ...$ es la *unión disjunta* de tales eventos y en ese caso escribiremos

$$A_1 \sqcup A_2 \sqcup ... \tag{2.31}$$

Definición. Si

$$S = A_1 \sqcup A_2 \sqcup ... \tag{2.32}$$

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

Ejemplo 2.2.9. 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 reverso.*

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

1. $A \cup B = \{HT, TH, HH, TT\} = S$
2. $A \cap B = \{HT\}$
3. $A' = \{TT\}$
4. $A - B = \{TH, HH\}$

Enfoques de la 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.

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 2.2.10. 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. Aquí suponemos que la moneda no está cargada.

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

Observación. Ambos enfoque tienen sus inconvenientes:

1. En el caso clásico, la expresión *"igualmente probable"* es vaga;
2. 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 $P : C \to \mathbb{R}$ es una función de probabilidad si satisface las siguientes propiedades:

Axioma. Para cada evento A, se tiene que

$$P(A) \geq 0. \tag{2.33}$$

Axioma. La probabilidad del evento cierto S es

$$P(S) = 1. \tag{2.34}$$

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

$$P(A_1 \sqcup A_2 \sqcup ...) = P(A_1) + P(A_2) + ... \tag{2.35}$$

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

$$P(A_1 \sqcup A_2) = P(A_1) + P(A_2) \tag{2.36}$$

Algunos teoremas importantes en probabilidad

Teorema 2.2.1. *Si $A_1 \subset A_2$, entonces $P(A_1) \leq P(A_2)$ y*

$$P(A_2 - A_1) = P(A_2) - P(A_1). \tag{2.37}$$

Teorema 2.2.2. *Para cada evento A,*

$$0 \leq P(A) \leq 1, \tag{2.38}$$

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

Teorema 2.2.3. *El evento imposible tiene probabilidad cero, es decir,*

$$P(\emptyset) = 0. \tag{2.39}$$

Teorema 2.2.4. *La probabilidad de un evento complementarios está dada por*

$$P(A') = 1 - P(A) \tag{2.40}$$

Teorema 2.2.5. *Si $A = A_1 \sqcup ... \sqcup A_N$ es la unión disjunta de eventos mutuamente excluyentes entonces*

$$P(A) = P(A_1) + ... + P(A_N). \tag{2.41}$$

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

$$P(A_1) + ... + P(A_N) = 1. \tag{2.42}$$

Teorema 2.2.6. *Si A, B, C son dos eventos no necesariamente excluyentes, entonces*

$$P(A \cup B) = P(A) + P(B) - P(A \cap B). \tag{2.43}$$

$$\begin{aligned} P(A \cup B \cup C) = {} & P(A) + P(B) + P(C) && (2.44)\\ & - P(A \cap B) - P(B \cap C) - P(C \cap A) && (2.45)\\ & + P(A \cap B \cap C). && (2.46) \end{aligned}$$

Teorema 2.2.7. *Para cualesquiera eventos A, B,*

$$P(A) = P(A \cap B) + P(A \cap B'). \tag{2.47}$$

Teorema 2.2.8. *Si $A_1, A_2, ..., A_N$ es una partición del espacio muestral S, es decir, $S = A_1 \sqcup A_2 \sqcup ... \sqcup A_N$ entonces para cualquier evento A*

$$P(A) = P(A \cap A_1) + P(A \cap A_2) + ... + P(A \cap A_N). \tag{2.48}$$

Asignación de probabilidades

Si un espacio muestral consiste en una cantidad *finita* de posibles resultados $a_1, ..., a_N$, entonces por el teorema 2.2.5,

$$P(A_1) + ... + P(A_n) = 1 \tag{2.49}$$

donde $A_1, ..., A_n$ 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 (2.49).

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

$$P(A_k) = \frac{1}{n},\ k = 1, 2, ..., n, \tag{2.50}$$

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

$$P(A) = \frac{h}{n}. \tag{2.51}$$

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

Ejemplo 2.2.12. Si solo dado se lanza, la probabilidad de que obtengamos un 2 o un 5 es

$$P(\{2, 5\}) = P(\{2\}) + P(\{5\}) = 1/6 + 1/6 = 1/3.$$

Problemas

La baraja está dividida en cuatro palos (en inglés: suit), dos de color rojo y dos de color negro:

- Espadas (conocidas como picas) ♠,
- Corazones ♡,
- Rombos (conocidos como diamantes, oros o cocos) ♢,
- Tréboles (conocidos como flores) ♣

Cada palo está formado por 13 cartas, de las cuales 9 cartas son numerales y 4 literales. Se ordenan de menor a mayor rango"de la siguiente forma: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q y K.

Problema 2.2.1. [1] Una carta se obtiene al azar de una baraja inglesa. Describe el espacio muestral si se consideran los palos.

[1] Consulta también la solución en línea.

Solución.
La solución está dada por el producto cartesiano del conjunto

$$A = \{1, 2, 3, 4\},$$

donde cada número representa alguno de los palos, y el conjunto

$$B = \{A, 2, ..., 10, J, Q, K\}.$$

De manera que el espacio muestral contiene $4 \times 13 = 52$ puntos muestrales.

Problema 2.2.2. [2] Supongamos que A es el evento `"se obtiene un rey"` o simplemente $\{K\}$, mientras que B es `"se obtiene un trébol` o simplemente $\{\clubsuit\}$. Describe los siguiente eventos:

[2] Consulta también la solución en línea.

1. $A \cup B$
2. $A \cap B$
3. $A \cup B'$
4. $A' \cup B'$
5. $A - B$
6. $A' - B'$
7. $(A \cap B) \cup (A \cap B')$

Solución.

1. Se obtiene un rey o un trébol.
2. Se obtiene un rey y un trébol.
3. Se obtiene un rey o no se obtiene un trébol. De manera equivalente: Si se obtiene un trébol, entonces se obtiene un rey.
4. No se obtiene un rey o no se obtiene un trébol.
5. Se obtiene un rey, pero no un trébol.
6. No se obtiene un rey ni un trébol.
7. O bien se obtiene un rey y un trébol, o bien se obtiene un rey pero no un trébol. De manera equivalente: Se obtiene un rey.

Problema 2.2.3. De una baraja inglesa se extraen 2 cartas. Encuentre la probabilidad de que las dos sean ases si la primera carta...

1. ...se devuelve a la baraja.
2. ...no se devuelve a la baraja.

Solución.

1. $$\frac{4}{52} \times \frac{4}{52} = \frac{1}{13} \times \frac{1}{13} = \frac{1}{169}$$
2. $$\frac{4}{52} \times \frac{3}{51} = \frac{1}{221}$$

Problema 2.2.4. En un contenedor hay 6 pelotas rojas, 4 blancas y 5 azules. Se extraen sucesivamente 3 pelotas. Encuéntrese la probabilidad de que se extraigan en el orden roja, blanca y azul si...

1. ...cada pelota se devuelve a la caja.
2. ...no se devuelve.

Solución. En total, hay 15 pelotas en el contenedor.

1. $$\frac{6}{15} \times \frac{4}{15} \times \frac{5}{15} = \frac{8}{225}.$$
2. $$\frac{6}{15} \times \frac{4}{14} \times \frac{5}{13} = \frac{4}{91}.$$

Problema 2.2.5. Encuéntrese la probabilidad de que en dos lanzamientos de un dado se obtengan por lo menos un 4 en alguno de los dos lanzamientos.

Solución. El espacio solución está dado por

$$E = \{4,5,6\} \times \{1,..,6\} \cup \{1,...,6\} \times \{4,...,6\}$$

cuya cardinalidad es 9.

Ahora bien, el espacio muestral está dado por

$$S = \{1,...,6\} \times \{1,...,6\},$$

cuya cardinalidad es 36.

De manera que la probabilidad es

$$\frac{9}{36} = \frac{1}{4}.$$

Problema 2.2.6. Encuentre la probabilidad de no obtener una suma de 7 o de 11 puntos al lanzar ambos dados.

Solución. Consideremos nuevamente el espacio muestral

$$S = \{1,...,6\} \times \{1,...,6\}.$$

Podemos obtener una suma igual con 7 con los siguientes puntos:

$$\{(1,6),(2,5),(3,4),(4,3),(5,2),(6,1)\},$$

mientras que la de 11, con los siguientes:

$$\{(5,6),(6,5)\}$$

.

Estos eventos son mutuamente excluyentes (es decir, como conjuntos son disjuntos). Por lo que existen 8 posibles eventos simples con los que obtendríamos o bien una suma de 7 o bien una suma de 11.

Por tanto, la probabilidad es

$$1 - \frac{8}{36} = \frac{28}{36}.$$

2.3 *Probabilidad condicional*

En esta sección, revisaremos el teorema de Bayes, el cuál describe la probabilidad de que ocurra un evento, basado en el conocimiento previo de las condiciones a las que está sujeto el problema.

Para entender este teorema, necesitamos definir la probabilidad condicional, y como es que podemos usar los elementos de una partición para calcular la probabilidad de un evento.

Sean $A, B \subset S$ 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. De manera formal:

Definición (Probabilidad condicional)**.**

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \tag{2.52}$$

$$P(A \cap B) = P(A)P(B|A) \tag{2.53}$$

Observación. 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.

Ejemplo 2.3.1. En un bote se colocan cinco canicas: tres blancas y dos negras. Inmediatamente después se agita el bote, con el fin de sortear su contenido. Determina la probabilidad de que en el segundo turno extraigamos una canica negra, dado que en el primero obtuvimos una blanca.

Solución. Digamos que el evento E_i consiste en obtener una canica blanca en el í-esimo intento, de manera que E_i' es obtener una canica negra.

Entonces, la probabilidad de sacar primero una canica blanca y después una canica negra es

$$P(E_1 \cap E_2') = \frac{3}{5} \times \frac{2}{4} = \frac{6}{20}. \tag{2.54}$$

Esto porque en el primer turno hay tres canicas blancas de cinco, pero si sacamos en una canica blanca (sin reemplazarla), entonces quedaran cuatro canicas, de las cuales dos serán blancas.

Ahora bien, la probabilidad $P(E_1)$ de sacar una canica blanca al primer intento es $\frac{3}{5}$, de manera que

$$P(E_2'|E_1) = \frac{1}{2}. \tag{2.55}$$

Teorema 2.3.1. *Para cualesquiera tres eventos A_1, A_2, A_3, tenemos que*

$$P(A_1 \cap A_2 \cap A_3) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2) \tag{2.56}$$

Teorema 2.3.2. *Si* $S = A_1 \sqcup ... \sqcup A_N$, *entonces*

$$P(A) = P(A_1)P(A|A_1) + ... + P(A_N)P(A|A_N) \tag{2.57}$$

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. A y B son eventos independientes si y solo si

$$P(A \cap B) = P(A)P(B). \tag{2.58}$$

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

$$k \neq j \to P(A_j \cap A_k) = P(A_j)P(A_k),\ j,k = 1,2,3 \tag{2.59}$$

$$P(A_1 \cap A_2 \cap A_3) = P(A_1)P(A_2)P(A_3). \tag{2.60}$$

2.4 Teorema de Bayes

Si tanto $P(A)$ como $P(B)$ son diferentes de cero, entonces, a partir de la definición de probabilidad condicional, podemos deducir que

$$P(A \cap B) = P(A) \cdot P(B|A) \tag{2.61}$$

$$P(B \cap A) = P(B) \cdot P(A|B), \tag{2.62}$$

y como $P(A \cap B) = P(B \cap A)$, concluimos

$$P(A) \cdot P(B|A) = P(B) \cdot P(A|B) \tag{2.63}$$

y por tanto

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}. \tag{2.64}$$

El resultado es conocido como *teorema de Bayes.*

Generalizaciones

El resultado anterior se puede extender de la siguiente manera: Como

$$\begin{cases} (B \cap A) \cup (B \cap A') &= B \\ (B \cap A) \cap (B \cap A') &= \emptyset \end{cases}, \tag{2.65}$$

entonces

$$P(B) = P\left((B \cap A) \cup (B \cap A')\right) \tag{2.66}$$

$$= P(B \cap A) + P(B \cap A') \tag{2.67}$$

$$= P(B|A)P(A) + P(B|A')P(A'), \tag{2.68}$$

de manera que

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B|A)P(A) + P(B|A')P(A')}. \tag{2.69}$$

Podemos generalizar este concepto usando una partición $\{E_i\}_{i=0}^{N}$ del espacio muestral $S \supset A, B$. En ese caso

$$A = \bigcup_{i=0}^{N} (A \cap E_i) \tag{2.70}$$

y por tanto

$$P(B) = \sum_{i=0}^{N} P(B \cap E_i) \tag{2.71}$$

$$= \sum_{i=0}^{N} P(B|E_i)P(E_i). \tag{2.72}$$

A este resultado lo llamaremos *regla de la cadena.*

Ahora bien, de manera similar al primer caso del teorema de Bayes, concluímos que

$$P(B|E_j)P(E_j) = P(E_j|B)P(B), \tag{2.73}$$

de forma que utilizando la regla de la cadena, obtenemos

$$P(E_j|B) = \frac{P(B|E_j)P(E_j)}{P(B)} \tag{2.74}$$

$$= \frac{P(B|E_j)P(E_j)}{\sum_{i=0}^{N} P(B|E_i)P(E_i)} \tag{2.75}$$

Ejemplo 2.4.1. Una planta productora de gelatinas cuenta con tres máquinas empacadoras. Así la distribución de volumen de empaque se realiza de la siguiente manera:

- Máquina 1: 38 %
- Máquina 2: 32 %
- Máquina 3: 30 %

Ahora bien, la probabilidad de que el empaque salga defectuoso es de 11 %, 15 % y 14 %, respectivamente por cada máquina.

La gerencia de producción de la planta está interesada en conocer cuál es la probabilidad de que si se selecciona una unidad al azar y es defectuoso, esta se haya empacado en la máquina 2.

Denotemos por M_i el evento de que una unidad de gelatina se haya empacado en la i-ésima máquina, mientras que D es el evento de que la unidad sea defectuosa.

Calcula la probabilidad de que una unidad de gelatina venga de la segunda máquina dado que es defectuosa.

Solución.
De acuerdo al problema

$$P(M_1) = 0.38 \tag{2.76}$$
$$P(M_2) = 0.32 \tag{2.77}$$
$$P(M_3) = 0.30 \tag{2.78}$$
$$P(D|M_1) = 0.11 \tag{2.79}$$
$$P(D|M_2) = 0.15 \tag{2.80}$$
$$P(D|M_3) = 0.14 \tag{2.81}$$

Entonces

$$P(M_2|D) = \frac{P(D|M_2)P(M_2)}{P(D|M_1)P(M_1) + P(D|M_2)P(M_2) + P(D|M_3)P(M_3)} \tag{2.82}$$
$$= \frac{(0.15)(0.32)}{(0.11)(0.38) + (0.15)(0.32) + (0.14)(0.30)} \tag{2.83}$$
$$= 0.3642 = 36.42\,\% \tag{2.84}$$

Problemas

Problema 2.4.1. [3] Encontrar la probabilidad de que una solo lanzamiento de un dado resulte en un número menor que 4 si

[3] Solución al problema 2.4.1

1. no hay más información;
2. se sabe que el lanzamiento resultó en un número impar.

Problema 2.4.2. 1. La caja 1 contiene 3 canicas rojas y 2 azules, mientras que la caja 2 contiene 2 canicas rojas y 8 azules.

   Se lanzan dos dados y se calcula la suma: Si se obtiene una suma de a lo más seis puntos, se elige una canica de la caja I; en otro caso, se elige una canica de la caja 2.

   Calcula la probabilidad de que se elija una canica roja.

2. Supongamos que quien lanza la moneda no revela si que número se obtuvo del dado, pero sí revela que se eligió una canica roja. ¿Cuál es la probabilidad de que se eligiera la caja 1?

2.5 *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.

Ejemplo 2.5.1. 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.

Funciones de probabilidad discretas

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

$$P(X = x_k) = f(x_k). \tag{2.85}$$

Función de probabilidad

$$P(X = x) = \begin{cases} f(x) & x = x_k \\ 0 & \text{en otro caso} \end{cases} \tag{2.86}$$

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

$$\begin{cases} f(x) \geq 0 \\ \sum_x f(x) = 1. \end{cases} \tag{2.87}$$

Ejemplo 2.5.2. Encuentre la función de probabilidad correspondiente a la variable aleatoria X del ejemplo 2.5.1.

Ejemplo 2.5.3. 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 2.5.4. Encuentre la distribución de probabilidad de niños y niñas en familias con 3 hijos, suponiendo la misma probabilidad para niños y niñas.

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

$$F(x) = P(X \leq x) = \sum_{u \leq x} f(u). \tag{2.88}$$

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

$$F(x) = \begin{cases} 0 & -\infty < x < 0 \\ f(x_1) & x_1 \leq x < x_2 \\ f(x_1) + f(x_2) & x_2 \leq x < x_3 \\ \vdots & \vdots \\ f(x_1) + \cdots + f(x_n) & x_n \leq x < \infty \end{cases} \tag{2.89}$$

Ejemplo 2.5.5. Encuentre la función de distribución para la variable aleatoria X del ejemplo 2.5.2 y obtenga su gráfica.

Observación.
- 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

$$f(x) = F(x) - \lim_{u \to x^-} F(u). \tag{2.90}$$

Ejemplo 2.5.6. 1. Encuentre la función de distribución $F(x)$ para la variable aleatoria del problema resuelto 2.5.3;

2. grafique esta función de distribución.

Ejemplo 2.5.7. 1. Encuentre la función de distribución $F(x)$ para la variable aleatoria del problema resuelto 2.5.4;

2. grafique esta función de distribución.

2.6 *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

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)du, \ -\infty < x < \infty. \tag{2.91}$$

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

$$P(a < x < b) = \int_a^b f(x)dx. \tag{2.92}$$

$$P(X = a) = 0. \tag{2.93}$$

Por tanto, en (2.92) podemos reemplazar cualquier signo $<$ por $\leq$.

Ejemplo 2.6.1. 1. Encuentre la constante c tal que la función

$$f(x) = \begin{cases} cx^2 & 0 < x < 3 \\ 0 & \texttt{en otro caso} \end{cases} \tag{2.94}$$

sea una función de probabilidad.

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

Ejemplo 2.6.2. Encuentre la distribución de probabilidad para la variable aleatoria del ejemplo 2.6.1 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

$$P(x \leq X \leq x + \Delta x) = \int_x^{x+\Delta x} f(u)du, \tag{2.95}$$

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

$$P(x \leq X \leq x + \Delta x) \approx f(x)\Delta x. \tag{2.96}$$

También podemos deducir de (2.91), al diferenciar de ambos lados, que

$$\frac{dF(x)}{dx} = f(x) \tag{2.97}$$

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

$$F(x) = \begin{cases} 0 & x < 1 \\ \frac{x}{2} & 1 \leq x < 2 \\ 1 & x \leq 2. \end{cases} \tag{2.98}$$

Ejemplo 2.6.3. Una variable aleatoria X tiene función de densidad

$$f(x) = \frac{c}{x^2+1},\ -\infty < x < \infty. \tag{2.99}$$

1. Encuentre el valor de c;
2. encuentre la probabilidad de que

$$\frac{1}{3} < X^2 < 1. \tag{2.100}$$

Ejemplo 2.6.4. Encuentre la función de distribución correspondiente a la función de densidad del problema resuelto 2.6.3

Ejemplo 2.6.5. La función de distribución para una variable aleatoria X es

$$F(x) = \begin{cases} 1 - e^{-2x} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.101}$$

Encuentre

1. la función de densidad;
2. la probabilidad de que $X > 2$;
3. la probabilidad que $-3 < X \leq 4$.

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.

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

$$P(X = x, Y = y) = f(x, y) \tag{2.102}$$

donde

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

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

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

$$P(X = x_j, Y = y_k) = f(x_j, y_k) \tag{2.103}$$

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

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

$$P(X = x_j) = f_X(x_j) = \sum_{k=1}^{n} f(x_j, y_k). \tag{2.104}$$

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

$$P(Y = y_k) = f_Y(y_k) = \sum_{j=1}^{m} f(x_j, y_k). \tag{2.105}$$

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

Observe que

$$\sum_{j=1}^{m} f_X(x_j) = 1, \sum_{k=1}^{n} f_Y(y_k) = 1, \tag{2.106}$$

lo cual se puede reescribir como

$$\sum_{j=1}^{m} \sum_{k=1}^{n} f(x_j, y_k) = 1. \tag{2.107}$$

La *función de distribución conjunta* está definida por

$$F(x, y) = P(X \leq x, Y \leq y) = \sum_{u \leq x} \sum_{v \leq y} f(u, v) \tag{2.108}$$

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 Xy 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.

$$P(a < X < b, c < Y < d) = \int_a^b \int_c^d f(x, y) dy dx. \tag{2.109}$$

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

$$P(A) = \iint_{\mathcal{R}_A} f(x,y)dxdy. \tag{2.110}$$

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

$$F(x,y) = P(X \leq x, Y \leq y) = \int_{-\infty}^{x}\int_{-\infty}^{y} f(u,v)dvdu. \tag{2.111}$$

Se sigue que

$$\frac{\partial^2 F}{\partial x \partial y} = f(x,y) \tag{2.112}$$

$$P(X \leq x) = F_X(x) = \int_{-\infty}^{x}\int_{-\infty}^{\infty} f(u,v)dvdu \tag{2.113}$$

$$P(Y \leq y) = F_Y(y) = \int_{-\infty}^{\infty}\int_{-\infty}^{y} f(u,v)dvdu \tag{2.114}$$

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 (2.113) y (2.114) 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

$$f_X(x) = \int_{-\infty}^{\infty} f(x,v)dv,\ f_Y(x) = \int_{-\infty}^{\infty} f(u,y)du. \tag{2.115}$$

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,

$$P(X = x, Y = y) = P(X = x)P(Y = y), \tag{2.116}$$

o de manera equivalente

$$f(x,y) = f_X(x)f_Y(y). \tag{2.117}$$

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

$$P(X \leq x, Y \leq y) = P(X \leq x)P(Y \leq y) \tag{2.118}$$

o de manera equivalente

$$F(x, y) = F_X(x)F_Y(y) \tag{2.119}$$

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 2.6.6. La función de probabilidad conjunta de dos variables discretas X, Y está dada por

$$f(x, y) = \begin{cases} c(2x + y) & 0 \leq x \leq 2,\ 0 \leq y \leq 3 \\ 0 & \texttt{en otro caso.} \end{cases} \tag{2.120}$$

1. Encuentre el valor de la constante c;
2. encuentre $P(X = 2, Y = 1)$;
3. encuentre $P(X \geq 1, Y \leq 2)$.

Ejemplo 2.6.7. Encuentre las funciones de probabilidad marginal para X y Y en el problema resuelto 2.6.6.

Ejemplo 2.6.8. Muestre que las variables aleatorias del problema resuelto 2.6.6 son dependientes.

Ejemplo 2.6.9. La función de densidad conjunta de dos variables aleatorias continuas X y Y es

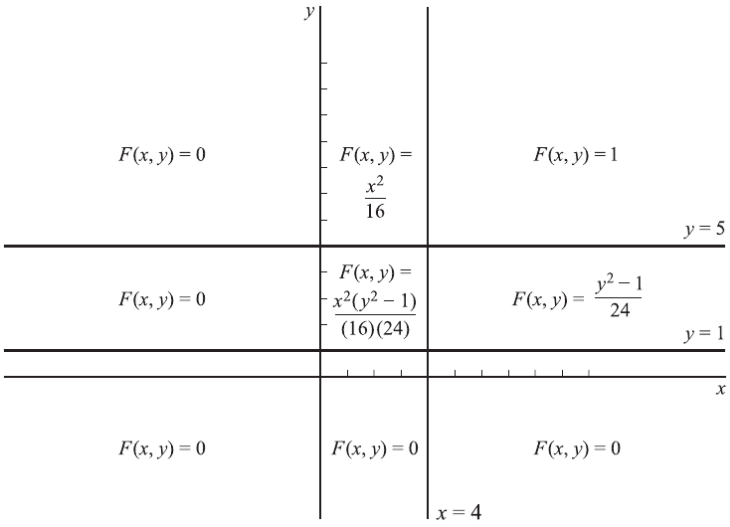
$$f(x, y) = \begin{cases} cxy & 0 < x < 4,\ 1 < y < 5 \\ 0 & \texttt{en otro caso.} \end{cases} \tag{2.121}$$

1. Encuentre el valor de c;
2. encuentre $P(1 < X < 2, 2 < Y < 3)$;

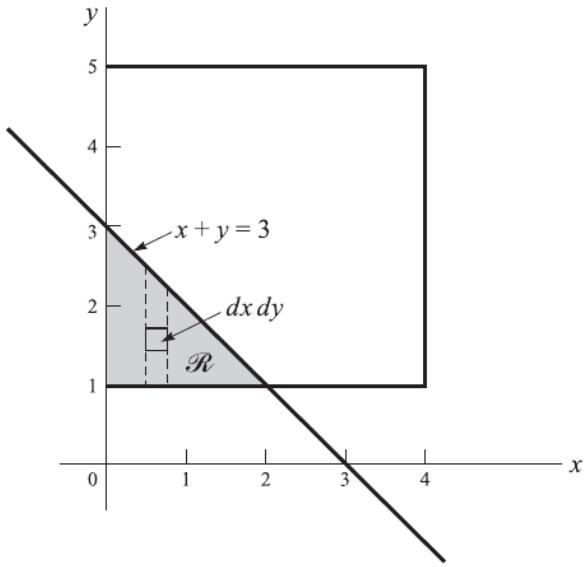
3. encuentre $P(X \geq 3, Y \leq 2)$.

Ejemplo 2.6.10. Encuentre las funciones de probabilidad marginal de las v.a's X, Y del problema resuelto 2.6.9.

Ejemplo 2.6.11. Encuentre la función de distribución conjunta para las v.a's del problema resuelto 2.6.9.



Ejemplo 2.6.12. En el problema resuelto 2.6.9, encuentre $P(X + Y < 3)$.



Distribución Condicional

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

$$P(B|A) = \frac{P(A \cap B)}{P(A)}. \tag{2.122}$$

Si X, Y son v.a's discretas y tenemos los eventos $A = \{X = x\}$, $B = \{Y = y\}$, entonces (2.122) se convierte

$$P(Y = y|X = x) = \begin{cases} \dfrac{f(x,y)}{f_X(x)} & 0 < f_X(x) \\ 0 & \texttt{en otro caso} \end{cases} \tag{2.123}$$

$f(x, y) = P(X = x, Y = y)$ 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

$$f(y|x) = \begin{cases} \dfrac{f(x,y)}{f_X(x)} & 0 < f_X(x) \\ 0 & \texttt{en otro caso} \end{cases} \tag{2.124}$$

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

$$f(x|y) = \begin{cases} \dfrac{f(x,y)}{f_Y(y)} & 0 < f_Y(y) \\ 0 & \texttt{en otro caso} \end{cases} \tag{2.125}$$

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

$$f(y|x) = \begin{cases} \dfrac{f(x,y)}{f_X(x)} & 0 < f_X(x) \\ 0 & \texttt{en otro caso} \end{cases} \tag{2.126}$$

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

$$P(c < Y < d|X = x) = \int_c^d f(y|x)dy. \tag{2.127}$$

Ejemplo 2.6.13. Para la distribución del problema resuelto 2.6.6, encuentre

1. $f(y|2)$; y
2. $P(Y = 1|X = 2)$

Ejemplo 2.6.14. Si X y Y tienen función de densidad conjunta

$$f(x,y) = \begin{cases} \frac{3}{4} + xy & 0 < x < 1,\ 0 < y < 1 \\ 0 & \texttt{en otro caso}, \end{cases} \tag{2.128}$$

encuentre

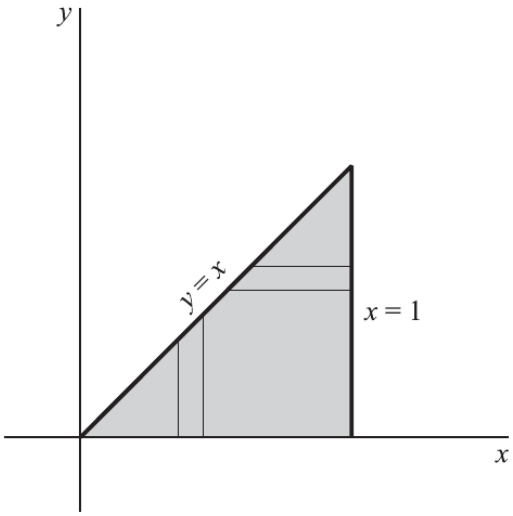
1. $f(y|x)$;
2. $P(Y > \frac{1}{2}|X = \frac{1}{2})$.

Ejemplo 2.6.15. La función de densidad conjunta de las variables aleatorias X y Y está dada por

$$f(x,y) = \begin{cases} 8xy & 0 \leq x \leq 1, 0 \leq y \leq x \\ 0 & \texttt{en otro caso}. \end{cases} \tag{2.129}$$

Encuentre

1. la densidad marginal de X;
2. la densidad marginal de Y;
3. la densidad condicional de X;
4. la densidad condicional de Y.



Ejemplo 2.6.16. Determine si las v.a's del problema resuelto 2.6.15 son independientes.

2.7 *Esperanza Matemática*

Definición de Esperanza Matemática

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

$$E(X) = \sum_{j=1}^{n} x_j P(X = x_j) =: \sum xP(X = x), \tag{2.130}$$

o de manera equivalente

$$E(X) = \sum_{j=1}^{n} x_j f(x_j) =: \sum xf(x), \tag{2.131}$$

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

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

$$E(X) = \frac{\sum_{i=1}^{n} x_i}{n}. \tag{2.132}$$

Ejemplo 2.7.1. Sea X el número que se obtiene al lanzar un dado. Entonces, cada cara x tiene la misma probabilidad

$$f(x) = \frac{1}{6} \tag{2.133}$$

de caer.

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

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

$$E(X) = \sum_{i=1}^{\infty} x_i f(x_i), \tag{2.134}$$

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

$$E(X) = \int_{-\infty}^{\infty} xf(x)dx \tag{2.135}$$

siempre y cuando dicha *integral* converja.

La esperanza de X es llamada a menudo *media* de X y es denotada por μ_x, o simplemente μ, 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 2.7.2. 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.

$$\mu = \$20\left(\frac{1}{6}\right) + \$40\left(\frac{1}{6}\right) + \$60\left(\frac{1}{6}\right) \tag{2.136}$$

$$= \frac{\$20 + \$40 + \$60 + 3 \times \$0}{6} \tag{2.137}$$

$$= \$15 \tag{2.138}$$

Ejemplo 2.7.3. La función de densidad de una variable aleatoria X está dada por

$$f(x) = \begin{cases} \frac{1}{2}x & 0 < x < 2 \\ 0 & \texttt{en otro caso} \end{cases} \tag{2.139}$$

Encuentre el valor esperado de X.

$$\mu = E(X) \tag{2.140}$$

$$= \int_{-\infty}^{\infty} xf(x)dx \tag{2.141}$$

$$= \int_{0}^{2} x\left(\frac{1}{2}x\right)dx \tag{2.142}$$

$$= \frac{1}{6}x^3\Big|_0^2 \tag{2.143}$$

$$= \frac{1}{6}(2)^3 - \frac{1}{6}(0)^3 \tag{2.144}$$

$$= \frac{4}{3} \tag{2.145}$$

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

$$h(y) = P(g(X) = y) = \sum_{\{x|g(x)=y\}} g(x)f(x) \tag{2.146}$$

Entonces, en el caso discreto.

$$E\left(g(X)\right) = \sum_{x} g(x)f(x) \tag{2.147}$$

De manera similar, en el caso continuo

$$E\left(g(X)\right) = \int_{-\infty}^{\infty} g(x)f(x)dx. \tag{2.148}$$

Ejemplo 2.7.4. Si X es la variable aleatoria del ejemplo 2.7.3, encuentre $E\left(3X^2 - 2X\right)$.

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

$$f(x) = \begin{cases} \frac{1}{2}x & 0 < x < 2 \\ 0 & \texttt{en otro caso} \end{cases} \tag{2.149}$$

$$\begin{aligned} E(3X^2 - 2X) &= E(g(X)) && (2.150)\\ &= \int_{-\infty}^{\infty} \left(3x^2 - 2x\right) f(x)dx && (2.151)\\ &= \int_0^2 \left(3x^2 - 2x\right) \left(\frac{1}{2}x\right) dx && (2.152)\\ &= \int_0^2 \frac{3}{2}\, x^3 - x^2 dx && (2.153)\\ &= \frac{3}{8}\, x^4 - \frac{1}{3}\, x^3 \Big|_0^2 && (2.154)\\ &= \frac{10}{3} && (2.155) \end{aligned}$$

Algunos temas sobre esperanza matemática

Linealidad

Teorema 2.7.1. *Si c, d son constantes y X, Y son variables aleatorias, entonces*

$$E\left(cX + dY\right) = cE\left(X\right) + dE\left(Y\right) \tag{2.156}$$

Esperanza e independencia

Teorema 2.7.2. *Si X, Y son* variables aleatorias independientes, *entonces*

$$E\left(XY\right) = E(X)E(Y) \tag{2.157}$$

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* μ.

Observación. Por esta razón, de aquí en adelante definiremos

$$\mu = \mu_X = E(X). \tag{2.158}$$

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

$$\sigma_X^2 = \mathrm{Var}(X) = E\left(\left(X - \mu_X\right)^2\right) \tag{2.159}$$

La *desviación estándar* se definirá como

$$\sigma_X = \sqrt{\mathrm{Var}\, X} \tag{2.160}$$

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

$$\mu = \mu_X,\ \sigma = \sigma_X,\ \sigma^2 = \sigma_X^2. \tag{2.161}$$

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

$$\sigma^2 = E\left((X-\mu)^2\right) = \sum (x-\mu)^2 f(x), \tag{2.162}$$

siempre y cuando esta suma converja.

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

$$\sigma^2 = \frac{(x_1-\mu)^2 + ... + (x_n-\mu)^2}{n} \tag{2.163}$$

Ejemplo 2.7.5. Como vimos anteriormente, si X es la cara obtenida al lanzar un dado, entonces $\mu_X = 3.5$.

La varianza de X es

$$\sigma^2 = \frac{(1-3.5)^2 + ... + (6-3.5)^2}{6} \tag{2.164}$$

$$= \frac{17.5}{6} \tag{2.165}$$

$$\approx 2.916 \tag{2.166}$$

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

$$\sigma^2 = E\left((X-\mu)^2\right) = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)dx \tag{2.167}$$

siempre y cuando la integral converja.

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

Ejemplo 2.7.6. Encuentre la varianza y la desviación estándar de la variable aleatoria del ejemplo 2.7.3.

Recordemos que esta variable aleatoria X tiene densidad de probabilidad

$$f(x) = \begin{cases} \frac{1}{2}x & 0 < x < 2 \\ 0 & \texttt{en otro caso} \end{cases} \tag{2.168}$$

$$\sigma^2 = Var(X) = \int_0^2 (x-\frac{4}{3})^2 f(x)dx \tag{2.169}$$

$$= \int_0^2 \frac{1}{2}\left(x-\frac{4}{3}\right)^2 xdx \tag{2.170}$$

$$= \frac{1}{8}x^4 - \frac{4}{9}x^3 + \frac{4}{9}x^2\Big|_0^2 \tag{2.171}$$

$$= \frac{2}{9} \tag{2.172}$$

$$\approx 0.\bar{2} \tag{2.173}$$

Algunos teoremas sobre Varianza

$$\sigma^2 = E\left(X^2\right) - \mu^2 \tag{2.174}$$

$$\text{Var}\left(cX\right) = c^2\,\text{Var}\left(X\right) \tag{2.175}$$

$$\sigma^2 = \min_a \left\{ E\left((X-a)^2 \right) \right\} \tag{2.176}$$

Si X, Y son independientes

$$\text{Var}(X \pm Y) = \text{Var}(X) + \text{Var}(Y) \tag{2.177}$$

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

$$X^* = \frac{X-\mu}{\sigma}. \tag{2.178}$$

$$E\left(X^*\right) = 0,\ \text{Var}(X^*) = 1. \tag{2.179}$$

Covarianza y correlación

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

$$\mu_X = E(X) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x f(x,y)dxdy$$

$$\mu_Y = E(Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} y f(x,y)dxdy$$

$$\sigma_X^2 = E\left((X-\mu_X)^2\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\mu_X)^2 f(x,y)dxdy$$

$$\sigma_Y^2 = E\left((Y-\mu_Y)^2\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (y-\mu_Y)^2 f(x,y)dxdy$$

Covarianza

$$\sigma_{XY} = \text{Cov}(X,Y) = E\left((X-\mu_X)(Y-\mu_Y)\right) \tag{2.180}$$

$$\sigma_{XY} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\mu_X)(y-\mu_Y) f(x,y)dxdy \tag{2.181}$$

Caso Discreto

$$\mu_X = \sum_x \sum_y x f(x,y)$$

$$\mu_Y = \sum_x \sum_y y f(x,y)$$

$$\sigma_{XY} = \sum_x \sum_y (x - \mu_X)(y - \mu_Y) f(x, y) \tag{2.182}$$

$$\sigma_{XY} = E(XY) - E(X)E(Y) = E(XY) - \mu_X \mu_Y \tag{2.183}$$

Si X, Y son independientes, entonces

$$\sigma_{XY} = \operatorname{Cov}(X, Y) = 0 \tag{2.184}$$

$$\operatorname{Var}(X \pm Y) = \operatorname{Var}(X) \pm 2 \operatorname{Cov}(X, Y) + \operatorname{Var}(Y). \tag{2.185}$$

De manera equivalente,

$$\sigma^2_{X \pm Y} = \sigma^2_X \pm 2\sigma_{XY} + \sigma^2_Y \tag{2.186}$$

Coeficiente de correlación de Pearson

$$\rho = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} \tag{2.187}$$

Teorema 2.7.3.

$$|\sigma_{XY}| \leq \sigma_X \sigma_Y \tag{2.188}$$

$$|\rho| \leq 1. \tag{2.189}$$

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 2.7.7. *Sean X, Y variables aleatorias discretas* con densidad de probabilidad conjunta

$$f(x, y) = \begin{cases} \dfrac{2x + y}{42} & 0 \leq x \leq 2,\ 0 \leq y \leq 3 \\ 0 & \texttt{en otro caso}. \end{cases} \tag{2.190}$$

Encuentre los siguientes estadísticos:

1. $\mu_X = E(X)$
2. $\mu_Y = E(Y)$
3. $E(XY)$
4. $E(X^2)$
5. $E(Y^2)$
6. $\sigma_X^2 = \text{Var}(X)$
7. σ_Y
8. $\sigma_Y^2 = \text{Var}(Y)$
9. σ_Y
10. $\sigma_{XY} = \text{Cov}(X, Y)$
11. ρ

Ejemplo 2.7.8. *Sean X, Y variables aleatorias continuas* con densidad de probabilidad conjunta

$$f(x,y) = \begin{cases} \frac{1}{210}(2x+y) & 2 < x < 6,\ 0 < y < 5 \\ 0 & \texttt{en otro caso.} \end{cases} \tag{2.191}$$

Encuentre los siguientes estadísticos:

1. $\mu_X = E(X)$
2. $\mu_Y = E(Y)$
3. $E(XY)$
4. $E(X^2)$
5. $E(Y^2)$
6. $\sigma_X^2 = \text{Var}(X)$
7. σ_Y
8. $\sigma_Y^2 = \text{Var}(Y)$
9. σ_Y
10. $\sigma_{XY} = \text{Cov}(X, Y)$
11. ρ

2.8 *Distribuciones especiales*

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

$$f(x) = P(X = x) = \binom{N}{x} p^x q^{N-x} \tag{2.192}$$

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

Ejemplo 2.8.1. La probabilidad de obtener exactamente dos caras en seis lanzamientos de una moneda es

$$\binom{6}{2}\left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right)^{6-2} = \frac{15}{64} \tag{2.193}$$

empleando (2.192) con $N = 6, x = 2, p = q = \frac{1}{2}$.

Ejemplo 2.8.2. 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 2.8.3. 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 $q = 1 - p$.

$$\mu = Np \tag{2.194}$$

$$\sigma^2 = Npq \tag{2.195}$$

[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()
```

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

$$f_{a,b}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x-a)^2}{b^2}} \tag{2.196}$$

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 (2.199), con parámetros a, b entonces

$$a = \mu_X \tag{2.197}$$
$$b = \sigma_X \tag{2.198}$$

Si una variable aleatoria normal X tiene función de densidad

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}, \tag{2.199}$$

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

Variable aleatoria normalizada

$$Z = \frac{X - \mu}{\sigma} \tag{2.200}$$
$$\mu_Z = 0 \tag{2.201}$$
$$\sigma_Z = 1 \tag{2.202}$$

Forma Estándar

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \tag{2.203}$$

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]

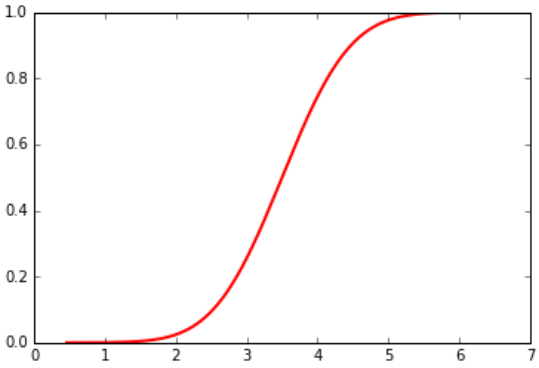
```
#(0.9973002039367399, 1.1072256503105314e-14)
```

[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
```



Relación entre las distribuciones binomial y normal

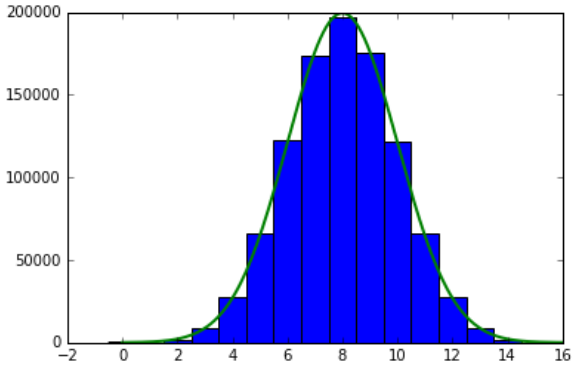
Si $N \sim \infty, p, q >> 0$, y X es un distribución binomial con parámetros N, p entonces

$$\frac{X - Np}{\sqrt{Npq}} \sim N(0, 1). \tag{2.204}$$

Ejemplo 2.8.4. 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()
```



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:

$$f(n) = \frac{\lambda^n e^{-\lambda}}{n!},\ n = 0, 1, 2, ... \tag{2.205}$$

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 2.8.5. 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
"""
```

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 $N >> 50, Np << 5$.

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()
```

Distribución multinomial

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

$$f(x_1, ..., x_2) = \frac{x_1 + ... + x_k}{x_1!...x_k!} p_1^{x_1} \cdots p_k^{x_k}. \tag{2.206}$$

Ejemplo 2.8.6. 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.

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"*

$$P_q = F^{-1}\left(\frac{q}{100}\right). \tag{2.207}$$

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 2.8.7. Encuentre

1. 5!
2. $\binom{8}{3}$

utilizando `Python`.

[fragile, allowframebreaks]combinaciones.py

```
import math
import scipy.special

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

Distribución Binomial

Ejemplo 2.8.8. Supóngase que 15 % de la población es zurda. Encontrar la probabilidad de que en un grupo de 50 individuos haya:

1. cuando mucho 10 zurdos;
2. por lo menos 5 zurdos;
3. entre 3 y 6 zurdos;
4. 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 2.8.9. 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:

1. 60;
2. 93;
3. 72.

Ejemplo 2.8.10. Con los datos del problema 2.8.9, encontrar las calificaciones que corresponden a las siguientes puntuaciones estándar:

1. -1;
2. 1.6.

Ejemplo 2.8.11. 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:

1. ¿Qué porcentaje participa en menos de 750 juegos?;
2. ¿qué porcentaje participa en más de 2000 juegos?;
3. 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 2.8.12. 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:

1. exactamente 3;
2. 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
```

Figura 2.1: Problema de conteo

Dada la siguiente información,

----Evento----	Cardinalidad
$(A \triangle B)'$	140
$A \cup B$	177
A	144
A'	99

calcule los elementos en cada conjunto

----Evento----	Cardinalidad
B'	
$A \triangle B$	
$A - B$	
B	
$A \cap B$	
$(B - A)'$	
$(A - B)'$	
$\emptyset$	
S	
$A' \cup B'$	
$A' \cap B'$	
$B - A$	

Figura 2.2: Los rompecabezas son ejemplos de particiones.

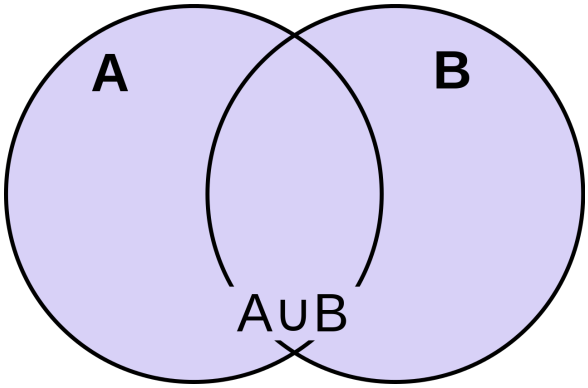
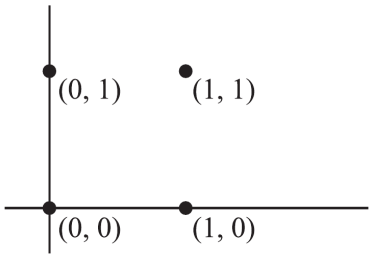


Figura 2.3: Diagram de Ven para dos conjuntos



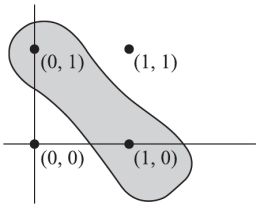


Figura 2.4: Espacio muestral para el lanzamiento de una moneda dos veces seguidas.

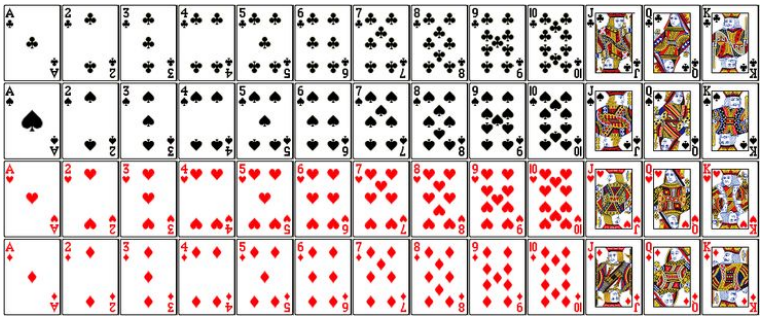


Figura 2.5: Baraja inglesa

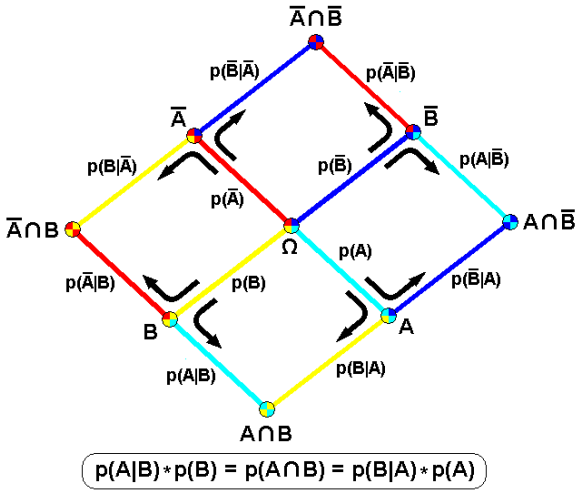


Figura 2.6: Diagrama para el teorema de Bayes.

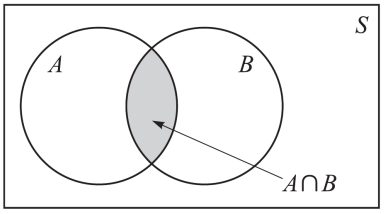
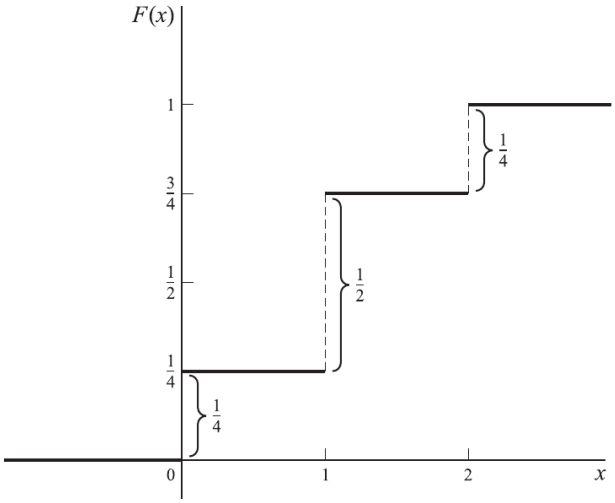
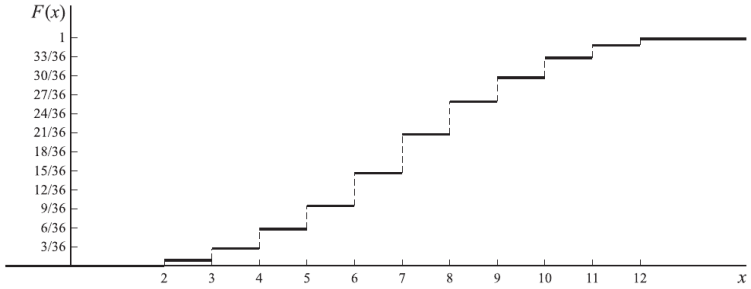
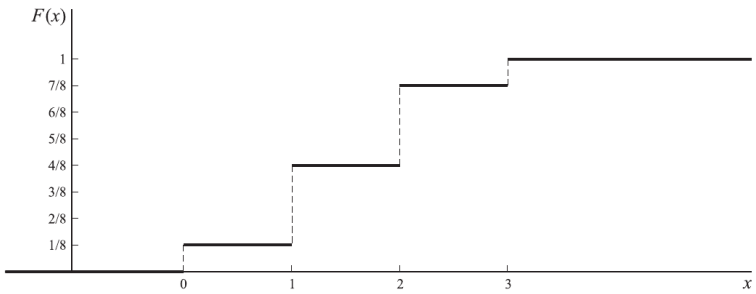
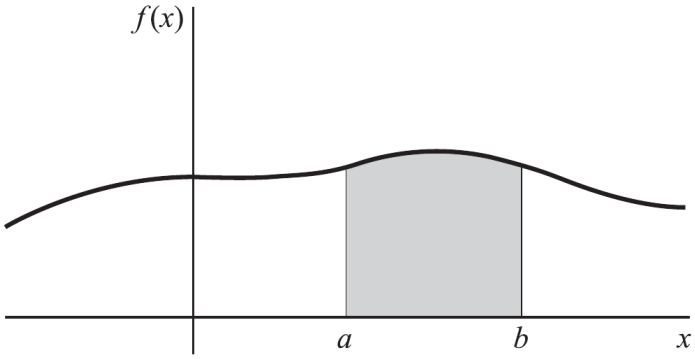


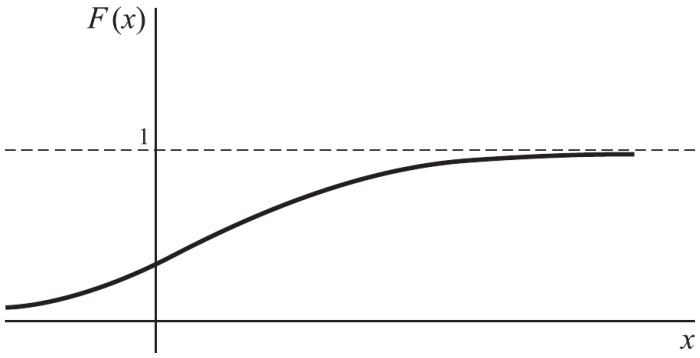
Figura 2.7: Unión de dos conjuntos


F(x)
1
3/4
1/2
1/4
1/4
1/2
1/4
0
1
2
x

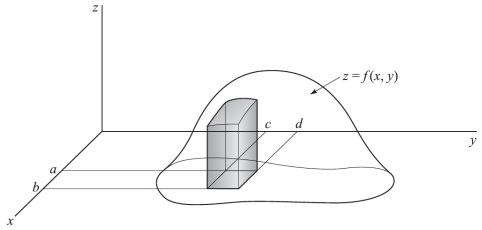

F(x)
1
33/36
30/36
27/36
24/36
21/36
18/36
15/36
12/36
9/36
6/36
3/36
2
3
4
5
6
7
8
9
10
11
12
x


F(x)
1
7/8
6/8
5/8
4/8
3/8
2/8
1/8
0
1
2
3
x

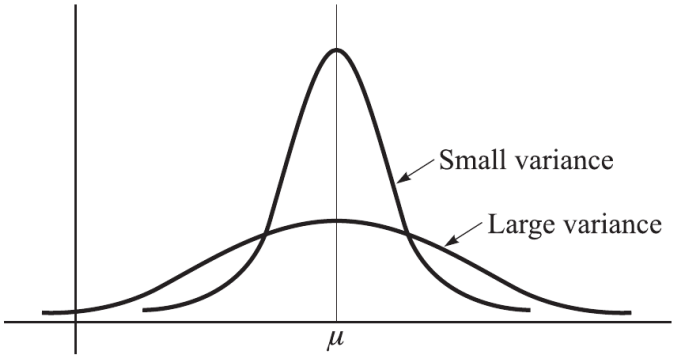

f(x)
a
b
x

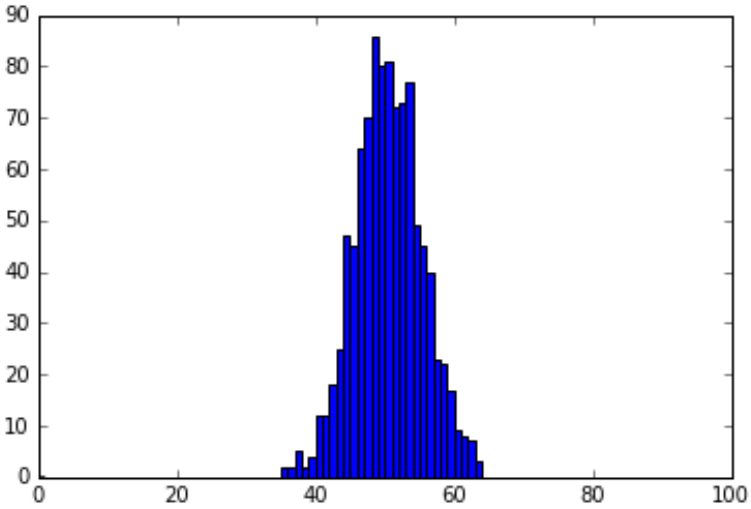


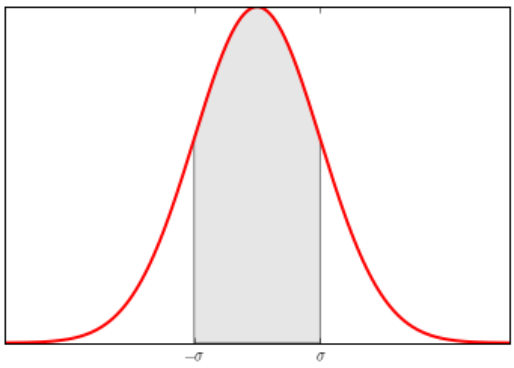
X \ Y	y_1	y_2	...	y_n	Totals ↓	
x_1	$f(x_1, y_1)$	$f(x_1, y_2)$	...	$f(x_1, y_n)$	$f_1(x_1)$	
x_2	$f(x_2, y_1)$	$f(x_2, y_2)$	...	$f(x_2, y_n)$	$f_1(x_2)$	
⋮	⋮	⋮		⋮	⋮	
x_m	$f(x_m, y_1)$	$f(x_m, y_2)$	...	$f(x_m, y_n)$	$f_1(x_m)$	
Totals →	$f_2(y_1)$	$f_2(y_2)$	...	$f_2(y_n)$	1	← Grand Total

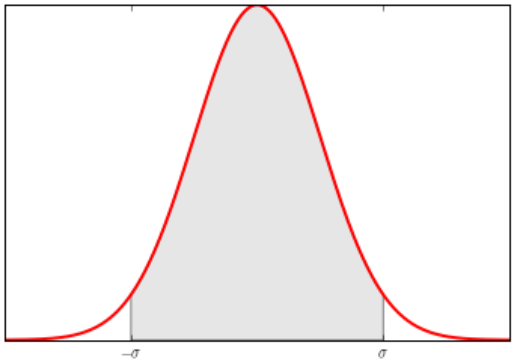


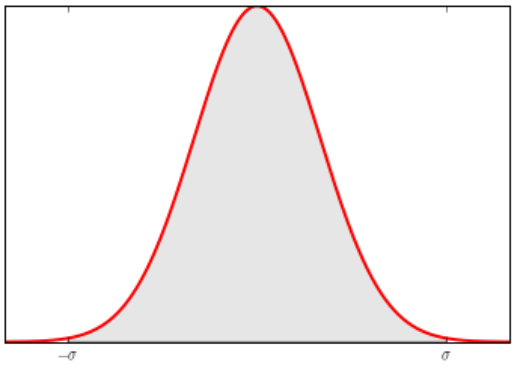
X \ Y	0	1	2	3	Totals ↓
0	0	c	$2c$	$3c$	$6c$
1	$2c$	$3c$	$4c$	$5c$	$14c$
2	$4c$	$5c$	$6c$	$7c$	$22c$
Totals →	$6c$	$9c$	$12c$	$15c$	$42c$

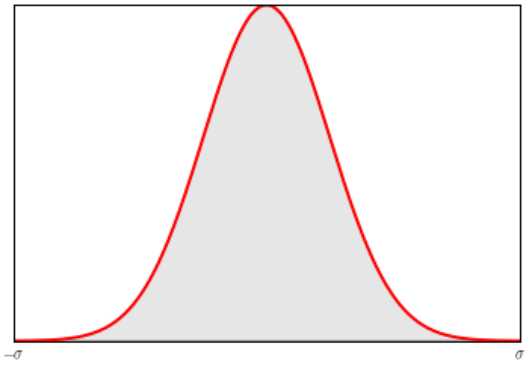

Small variance
Large variance
μ


90
80
70
60
50
40
30
20
10
0
0
20
40
60
80
100


$-\sigma$
σ


−σ
σ


−σ
σ


−σ
σ

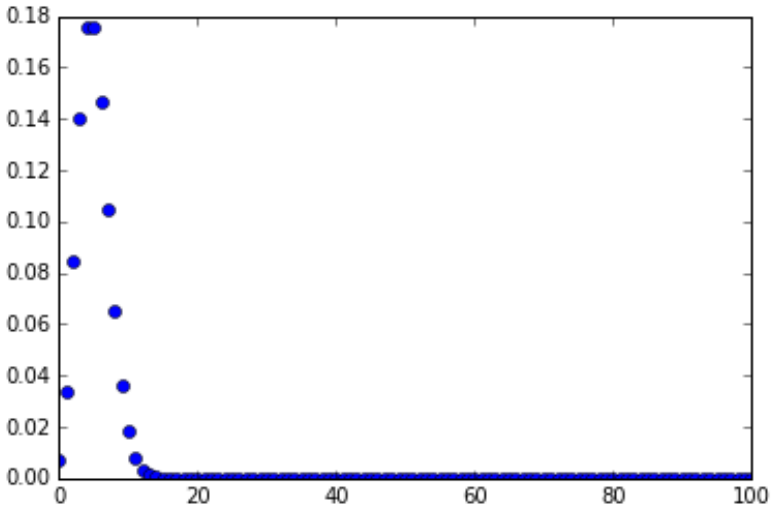


Figura 2.8: Distribución de Poisson

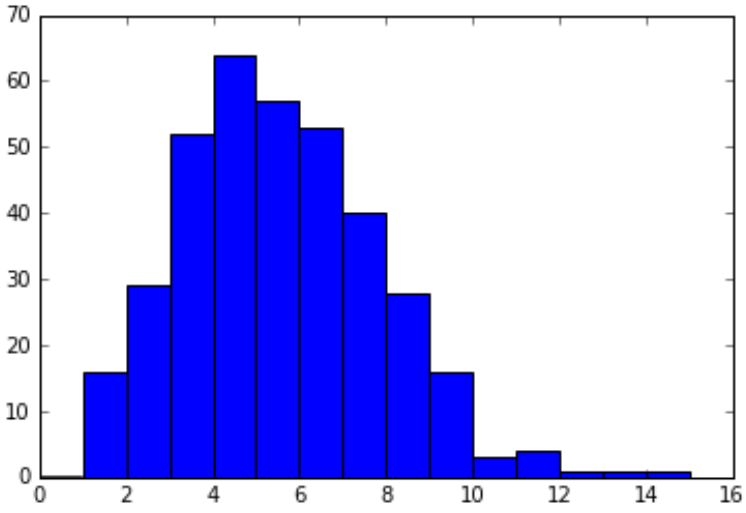


Figura 2.9: Histograma de pacientes en sala de urgencias durante un año con media $\lambda = 5$

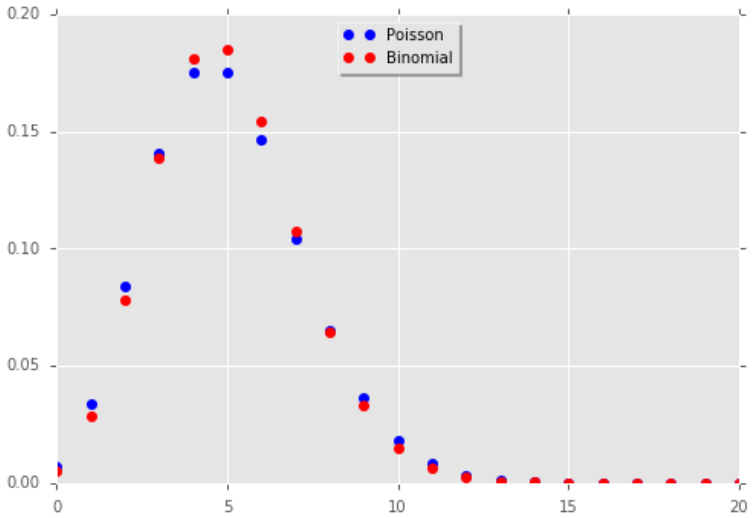


Figura 2.10: Comparación entre distribuciones Binomial y Poisson para eventos raros.

3 Estadística Inferencial

3.1 Conceptos más importantes

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

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*.

Estos dos conceptos son la columna vertebral de las pruebas de hipótesis.

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 χ-cuadrada y describiremos el uso de pruebas χ-cuadrada con un par de ejemplos.

Correlación

Entenderemos el significado y la significación de la correlación entre dos variables, de los coeficientes de correlación y calcularemos y visualizaremos la correlación entre variables de una base de datos.

3.2 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 problemalema 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 $A_1, A_2, ..., A_{100}$ el promedio de edades obtenido en cada respectivo intento.

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

$$\bar{A}_{100} = \frac{A_1 + ... + A_{100}}{100} \tag{3.1}$$

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, $A_1, A_2, ..., A_{100}$ estarán distribuidos de manera normal.

Observación. 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.

3.3 *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*.

3.4 Estadísticos Z y t

1. Suponga que el valor del parámetro asumido en la hipótesis nula es *Ao*.

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:

$$Z = \frac{A - A_0}{\sigma / \sqrt{n}} \tag{3.2}$$

donde σ 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 a 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 (3.2) 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 estandarizadas 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. [1]

[1] Wikipedia: Distribución *t* de Student

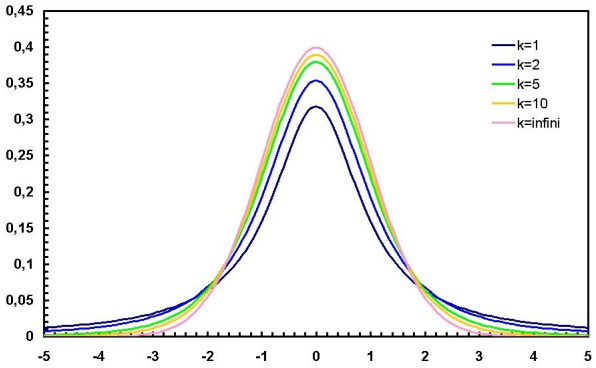


Figura 3.1: De The original uploader was Thorin de Wikipedia en francés - Transferido desde fr.wikipedia a Commons., CC BY-SA 1.0, https://commons.wikimedia.org/w/index.php?curid=18

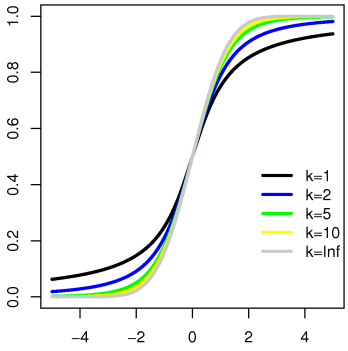
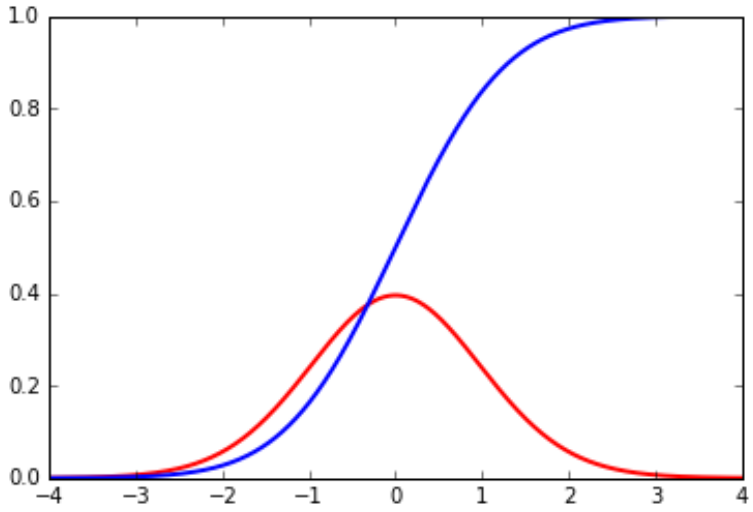


Figura 3.2: De Desconocido, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=78

```
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):
8  return stats.t.cdf(x, df=nu)
9  x = np.arange(-4,4,0.01)
10 yd = ft(x,30)
```

```
yc = Ft(x,30)

fig, ax = plt.subplots()
plt.plot(x, yd, 'r', linewidth=2)
plt.plot(x, yc, 'b', linewidth=2)
plt.ylim(ymin=0)
plt.show()
```

Listing 3.1: Distribución t en `Python`

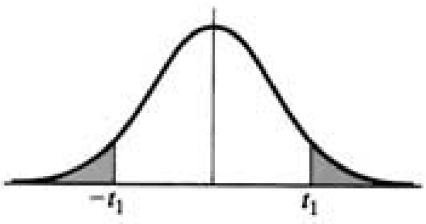


El parámetro `df` se le conoce como *grados de libertad* y generalmente se denota como ν (la letra `nu` griega).

Si una variable aleatoria X tiene distribución t con ν grados de libertad, entonces

$$\mu_X = 0,\ \sigma_X^2 = \frac{\nu}{\nu - 2} \tag{3.3}$$

Ejemplo 3.4.1. 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

```
from scipy import stats
import numpy as np
```

```
import matplotlib.pyplot as plt

def tp(x, nu):
return stats.t.ppf(x, df=nu)

print tp(0.05, 9)
##-1.83311293265
print tp(1-0.05, 9)
##1.83311293265
```

Varianza muestral

$$S^2 = \sum \frac{(A_i - A_0)^2}{n-1} \tag{3.4}$$

Estadístico t

$$t = \frac{(A - A_0)}{S/\sqrt{n}} \tag{3.5}$$

3.5 Intervalos de confianza, niveles de significación y valores p

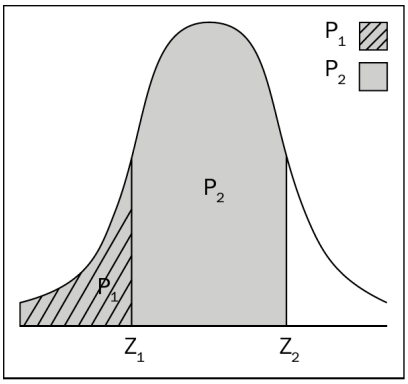


Figura 3.3: Una distribución típica normal con valores p.

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

En otras palabras

$$P(X > Z_1) = p_1 \tag{3.6}$$
$$P(X > Z_2) = p_2 \tag{3.7}$$

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 μ y desviación estándar σ, 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* cómo 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*

$$\beta = 1 - p, \tag{3.8}$$

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

β 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*)

$$p - \texttt{valor} = P\left(X > Z\right) \tag{3.9}$$

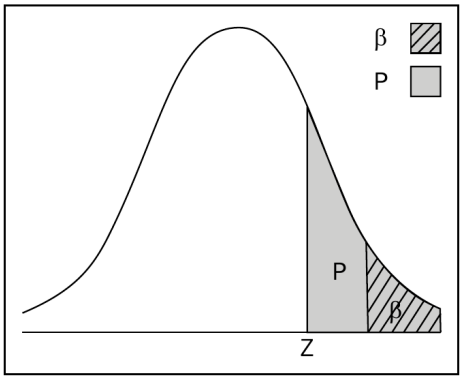


Figura 3.4: Una distribución normal típica con $p-$valores y nivel de significación.

Criterio

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

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

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

Cola izquierda

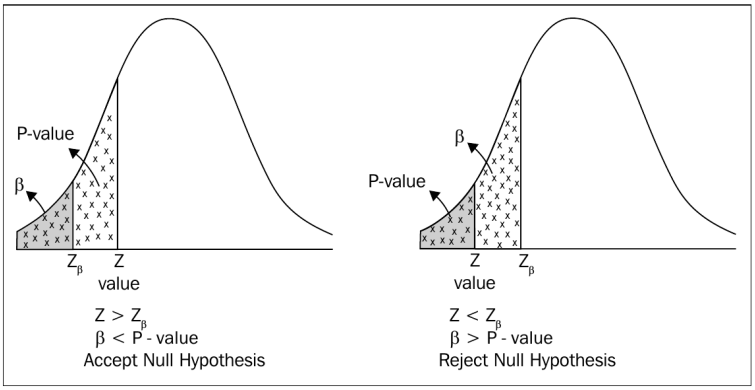


Figura 3.5: Prueba de hipótesis: Cola izquierda

Cola derecha

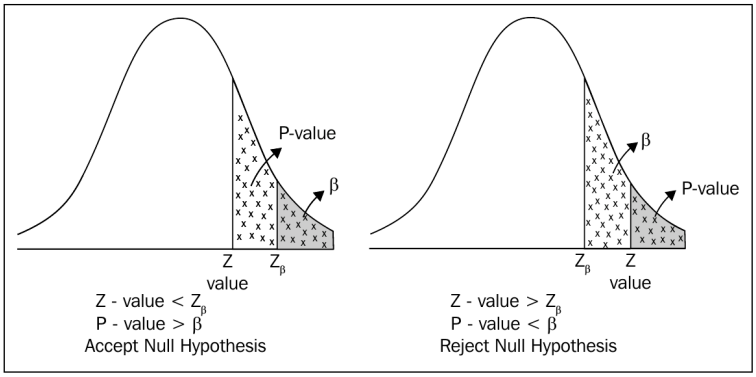


Figura 3.6: Prueba de hipótesis: Cola derecha

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.

3.6 *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

$$Z = \frac{A_m - A_0}{\sigma / \sqrt{n}} \tag{3.10}$$

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

La probleamibilidad 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.

Ejemplo

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:

$$Z = \frac{21.2 - 20}{3/\sqrt{64}} = 3.2 \tag{3.11}$$

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

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

$$\texttt{valor-p} = 1 - F(3.2) \tag{3.12}$$
$$= 1 - 0.999312862062 \tag{3.13}$$
$$= 0.000687137937916 \tag{3.14}$$

Esto significa que si la media fuera $A_0 = 20$, la problemaabilidad 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 $H_0 : A_0 = 20$.

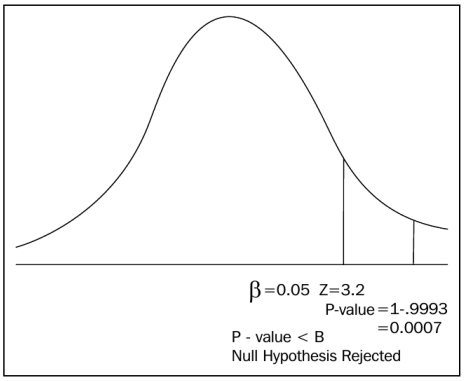


Figura 3.7: La hipótesis nula se rechaza porque el $p-$valor es menor al nivel de significación β.

3.7 *Prueba $\chi-$cuadrada*

La *prueba* χ^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:

$$\texttt{valor}\ \chi^2\texttt{:}g = \sum \frac{(O-E)^2}{E}, \tag{3.15}$$

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.
- Comproblemaar 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 reverso o un sol? La respuesta adecuada sería 5. Ahora bien, ¿qué pasaría si realizamos este experimento 1000 veces y registramos los números de reversos y soles.

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

$$H_0 : \text{La proporción de soles y reversos es } 0.5$$
$$H_a : \text{La proporción no es } 0.5$$

	Soles	reversos
Observado	553	447
Esperado	500	500

Calculemos el valor χ^2 :

$$g = \frac{\left((553-500)^2 + (447-500)^5\right)}{500} \approx 11.236 \tag{3.16}$$

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

La Distribución χ^2

Sean $X_1, X_2, ..., X_\nu$ variables aleatorias independientes $N(0,1)$. Consideremos la variable aleatoria

$$\chi^2 = X_1^2 + ... + X_\nu^2 \tag{3.17}$$

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

Propiedades de χ^2

$$\mu = \nu,\ \sigma = 2\nu \tag{3.18}$$

[]scipy.stats.chi2

```
from scipy.stats import chi2
import numpy as np
import matplotlib.pyplot as plt
fig, ax = plt.subplots(1, 1)

df = 55

x = np.linspace(chi2.ppf(0.01, df),
chi2.ppf(0.99, df), 100)
ax.plot(x, chi2.pdf(x, df),'r-',
lw=5, alpha=0.6, label='chi2 pdf')
```

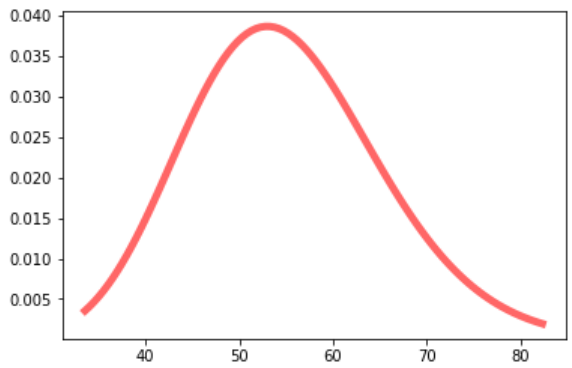
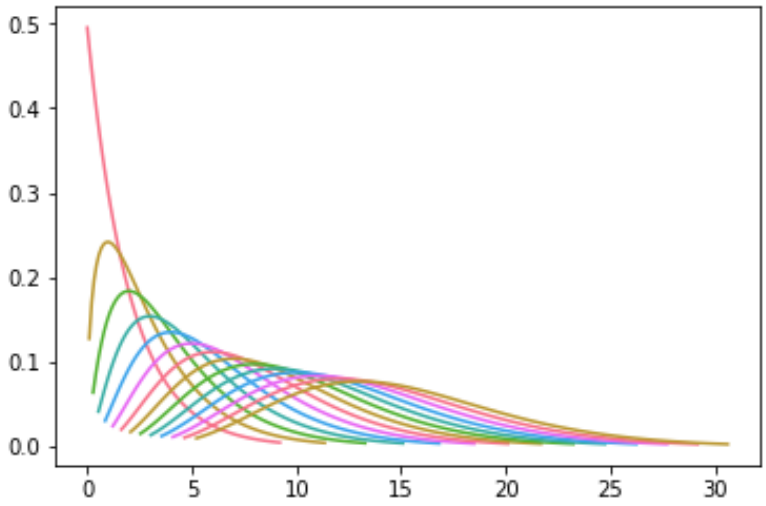


Figura 3.8: Función de densidad de distribución χ^2 con $\nu = 55$

[]statsChi2.py

```
from scipy.stats import chi2
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

sns.set_palette("husl")
fig, ax = plt.subplots(1, 1)

for df in range(2,15+1):
x = np.linspace(chi2.ppf(0.01, df),
chi2.ppf(0.99, df), 100)
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 $\beta = 0.05$.

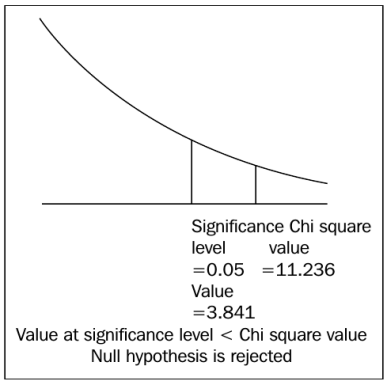


Figura 3.9: La hipótesis nula se rechaza porque el valor del estadístico χ^2 al nivel de significación es menor que el valor del estadístico.

[]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, en-

tonces 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 χ^2 se obtiene al sumar todos estos valores.

Conclusiones (del profesor)

Como $\chi^2 = 4.99$ y el valor del estadístico χ^2 a un nivel se significación es 11.07, la hipótesis nula se acepta.

De manera equivalente

$$\texttt{valor-}p = 1 - F_{\chi^2}(4.99) = 0.416991040312 > \beta = 0.05, \tag{3.19}$$

obtenemos la misma conclusión:

La elección de materias es independiente del género.

3.8 *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.

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:

$$Y = f(X) \tag{3.20}$$

Ejemplos de correlación

- Correlación lineal

$$y = ax + b \tag{3.21}$$

- Correlación exponencial

$$y = be^{ax} \tag{3.22}$$

Definición de correlación

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

$$\rho(X,Y) = \frac{\sum_{x_i,y_j} \left((x_i - \bar{x})\,(y_j - \bar{y})\right)}{\sqrt{\sum_{x_i,y_j} (x_i - \bar{x})^2 \sum_{x_i,y_j} (y_i - \bar{y})^2}} \tag{3.23}$$

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. ¡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 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

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import pandas as pd

advert = pd.read_csv("./dataBases/Advertising.csv")
print(advert.head())

import numpy as np

advert["dX*dY"] = ( advert["TV"] -
np.mean(advert["TV"])) * ( advert["Sales"]
np.mean(advert["Sales"]))
advert["dX**2"] = ( advert["TV"] -
np.mean(advert["TV"])) ** 2
advert["dY**2"] = ( advert["Sales"] -
np.mean(advert["Sales"])) ** 2

sxy = advert.sum()["dX*dY"]
sxx = advert.sum()["dX**2"]
syy = advert.sum()["dY**2"]

r = sxy/np.sqrt(sxx*syy)
```

[,]Salida de la pantalla

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

[,]advertising2.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import pandas as pd

advert = pd.read_csv("./dataBases/Advertising.csv")
print(advert.head())

import numpy as np

def rCoef(df, var1, var2):
df["dX*dY"] = (df[var1]-np.mean(df[var1]))*(df[var2]-np.
  mean(df[var2]))
df["dX**2"] = (df[var1]-np.mean(df[var1]))**2
df["dY**2"] = (df[var2]-np.mean(df[var2]))**2
sxy = df.sum()["dX*dY"]
sxx = df.sum()["dX**2"]
syy = df.sum()["dY**2"]
r = sxy/np.sqrt(sxx*syy)
return r

tvVsSales = rCoef(advert, "TV", "Sales")
## 0.782224424862
```

```
radioVsSales = rCoef(advert, "Radio", "Sales")
## 0.576222574571
```

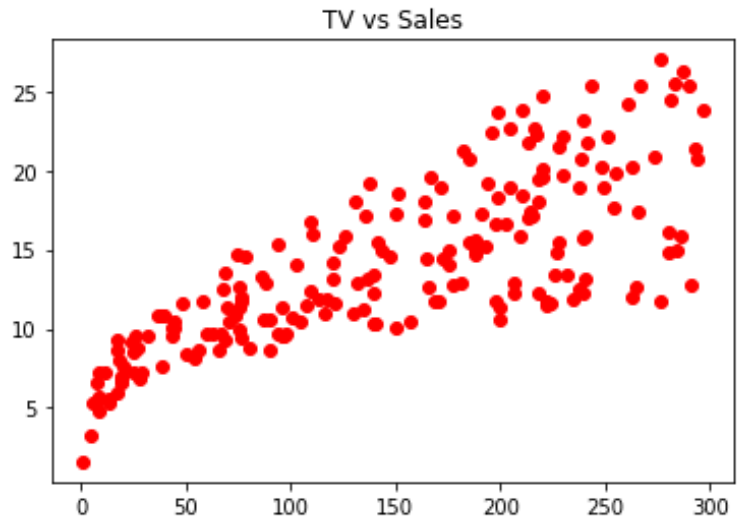
	TV	Radio	Newspaper	Sales
TV	1	0.05	0.06	0.78
Radio	0.05	1	0.35	0.57
Newspaper	0.06	0.35	1	0.23
Sales	0.78	0.57	0.23	1

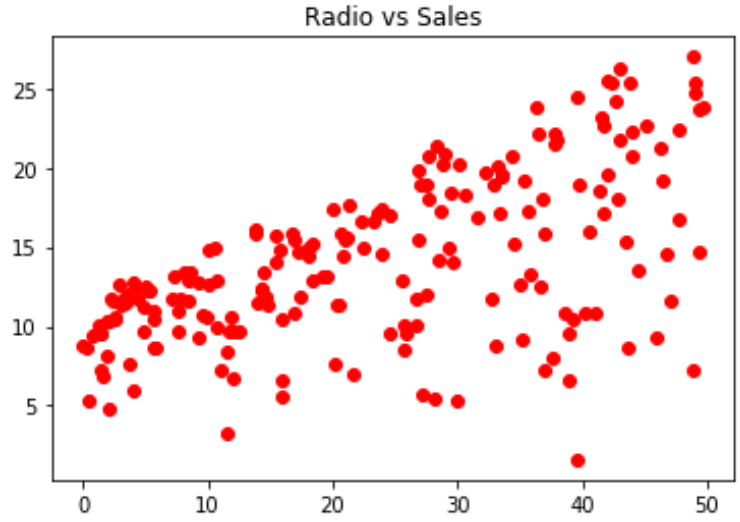
Figura 3.10: Matriz de correlación

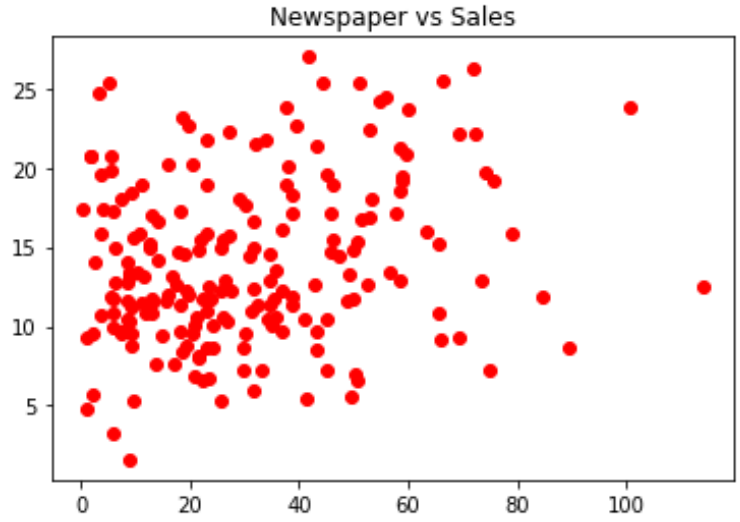
Veamos la naturaleza de esta correlación graficando las variables `TV`, `Radio` y `Newspaper` vs `Sales` del *data frame* `advert`.

[,]tvVsSales.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import pandas as pd
advert = pd.read_csv("./dataBases/Advertising.csv")

import matplotlib.pyplot as plt

plt.plot(advert['TV'],advert['Sales'],'ro')
plt.title('TV vs Sales')
plt.show()

plt.plot(advert['Radio'],advert['Sales'],'ro')
plt.title('Radio vs Sales')
plt.show()

plt.plot(advert['Newspaper'],advert['Sales'],'ro')
plt.title('Newspaper vs Sales')
plt.show()
```




Radio vs Sales
25
20
15
10
5
0
10
20
30
40
50


Newspaper vs Sales
25
20
15
10
5
0
20
40
60
80
100

4 Regresiones lineales

4.1 Introducción

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

$$P = a_1 \times S + a_2 \times A + a_3 \times T \tag{4.1}$$

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.

4.2 *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

$$Y_e = \alpha + \beta X, \tag{4.2}$$

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

Observación. El propósito de la regresión lineal es encontrar valores α, β 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

$$Y_e = 2 + 0.03X. \tag{4.3}$$

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

$$Y_e = 2 + 0.03(900) = 29. \tag{4.4}$$

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

$$\epsilon = Y - Y_e. \tag{4.5}$$

Nuestro objetivo es minimizar

$$\sum \epsilon^2 = \sum (Y - Y_e)^2 \tag{4.6}$$

$$= \sum (Y - (\alpha + \beta X))^2 \tag{4.7}$$

respecto de los parámetros α, β.

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

$$\beta = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2} \tag{4.8}$$

$$\alpha = \bar{y} - \beta \times \bar{x} \tag{4.9}$$

4.3 *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

$$Y = \alpha + \beta X + K. \tag{4.10}$$

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 α y β, 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 $\sigma = 0.5$.

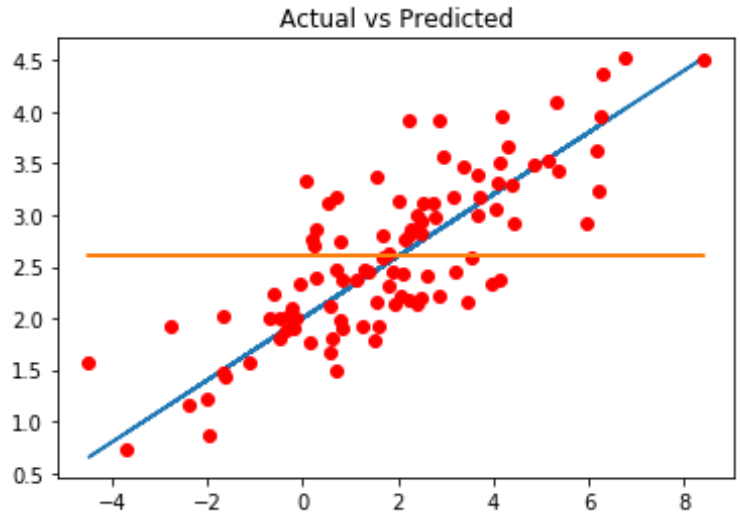
[,]`fittingLinearRegression.py`

```
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
```

```
res=.5*np.random.randn(100)+0
ypred=2+.3*x
yact=2+.3*x+res
xlist=x.tolist()
ypredlist=ypred.tolist()
yactlist=yact.tolist()
df=pd.DataFrame({'Input_Variable(X)':xlist,'
  Predicted_Output(ypred)':ypredlist,'Actual_Output(yact)':
  yactlist})
print(df.head())


import matplotlib.pyplot as plt

% x=2.5*np.random.randn(100)+1.5
% res=.5*np.random.randn(100)+0
% ypred=2+.3*x
% yact=2+.3*x+res

ymean=np.mean(yact)
yavg=[ymean for i in range(1,len(xlist)+1)]

plt.plot(x,ypred)
plt.plot(x,yact,'ro')
plt.plot(x,yavg)
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 linea 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 nuestro modelos.

Si usted pasa cualquier dato conteniendo dos variables, una de entrada y otra de salida, el programa de estadística generara algunos valores α, β.

¿Pero cómo entender que esos valores que se nos están dando son un buen modelo?

Suma de Cuadrados Total

$$SST = \sum (Y_i - \bar{Y})^2 \tag{4.11}$$

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

Suma de Cuadrados de Regresión

$$SSR = \sum (Y_{e,i} - \bar{Y})^2 \tag{4.12}$$

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

Suma de Cuadrados de Diferencia

$$SSD = \sum (Y_i - Y_{e,i})^2 \tag{4.13}$$

Recordemos que $Y_e = \alpha + \beta X$, con β definida por (4.8) y α por (4.9).

Utilizando estas identidad se puede demostrar que

$$SST = SSR + SSD. \tag{4.14}$$

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

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

Coeficiente de determinación

`R-cuadrado:`

$$R^2 = \frac{SSR}{SST} \tag{4.15}$$

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`

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

4.4 *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 siguiere, es la variable de salida real.

Utilizando estas dos variables, podemos calcular los valores de α y β de acuerdo a las fórmulas (4.9) y (4.8). En el siguiente script, implementaremos estas fórmulas para obtener un modelo optimo de regresión lineal.

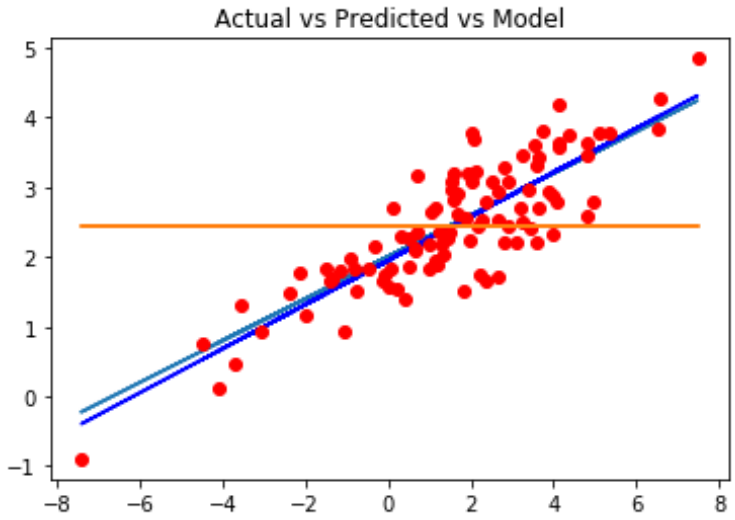
[,]`optimalValue.py`

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 23 00:37:36 2017

@author: jdk2py
"""

import pandas as pd
import numpy as np

np.random.seed(1234)

x=2.5*np.random.randn(100)+1.5
res=.5*np.random.randn(100)+0
ypred=2+.3*x
yact=2+.3*x+res
xlist=x.tolist()
ypredlist=ypred.tolist()
yactlist=yact.tolist()
```

```
df=pd.DataFrame({'Input_Variable(X)':xlist,'
  Predicted_Output(ypred)':ypredlist,'Actual_Output(yact)':
  yactlist})

ymean=np.mean(yact)
yavg=[ymean for i in range(1,len(xlist)+1)]

import matplotlib.pyplot as plt

xmean=np.mean(df['Input_Variable(X)'])
ymean=np.mean(df['Actual_Output(yact)'])
df['betan']=(df['Input_Variable(X)']-xmean)*(df['
  Actual_Output(yact)']-ymean)
df['xvar']=(df['Input_Variable(X)']-xmean)**2
betan=df.sum()['betan']
betad=df.sum()['xvar']
beta=betan/betad
alpha=ymean-beta*xmean
print(beta,alpha)

df['ymodel']=beta*df['Input_Variable(X)']+alpha

df['SSR']=(df['ymodel']-ymean)**2
df['SST']=(df['Actual_Output(yact)']-ymean)**2
SSR=df.sum()['SSR']
SST=df.sum()['SST']
R2 = SSR/SST

print(df.head())

plt.plot(x,ypred)
plt.plot(x,df['ymodel'], "b")
plt.plot(x,yact,'ro')
plt.plot(x,yavg)
plt.title('Actual vs Predicted vs Model')
```

[,]

```
Actual_Output(yact)  Input_Variable(X)  Predicted_Output(
  ypred)     betan  \
0              2.949179           2.678588
  2.803576  0.543137
1              1.840035          -1.477439
  1.556768  1.873530
2              3.776326           5.081767
  3.524530  4.629773
3              2.358159           0.718370
  2.215511  0.080941
4              2.151703          -0.301472
  1.909558  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 otras estadísticos y parámetros que uno necesita mira para hacer lo siguiente:

1. Seleccionar algunas variables y deseche 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 los estadísticos que ayudan a abordar los Problemas discutido anteriormente.

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

La prueba de hipotética es acerca si el valor β 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

$$\texttt{Hipótesis Nula } H_0 : \beta = 0 \tag{4.16}$$

$$\texttt{Hipótesis alternativa } H_a : \beta \neq 0 \tag{4.17}$$

En general, si se realiza una regresión lineal y β 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

$$\beta > 0 \ \texttt{o} \ \beta < 0, \tag{4.18}$$

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

En caso contrario, nos permitirían no rechazar la hipótesis nula, de manera que β 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 β y cada uno de estos indicará una estimación.

En tal caso, aparte de problemaar 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:

$$\texttt{Hipótesis nula} \ H_0 : \beta_1 = \beta_2 = ... = \beta_n = 0 \tag{4.19}$$

$$\texttt{Hipótesis alternativa} \ H_a : \exists \beta_i \neq 0 \tag{4.20}$$

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

$$F = \frac{(SST - SSD)\,/p}{SSD/\,(n-p-1)} \tag{4.21}$$

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:

$$RSE = \sqrt{\frac{1}{n-2}SSD} \tag{4.22}$$

donde n es el número de datos puntuales.

En general,

$$RSE = \sqrt{\frac{SSD}{n-p-1}} \tag{4.23}$$

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`:

[]

```
n = len(df['Input_Variable(X)'])
df['SSD']=(df['Actual_Output(yact)']-df['ymodel'])**2
SSD=df.sum()['SSD']
RSE=np.sqrt(SSD/(n-2))
```

El valor `RSE` resultante en este caso es ≈ 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.

4.5 *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 supuestos 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

$$\alpha = 7.032594,\ \beta = 0.047537 \tag{4.24}$$

y la ecuación de nuestro modelo será

$$\texttt{Ventas} = 7.032 + 0.047 * \texttt{TV} \tag{4.25}$$

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

```
print(model1.pvalues)

Intercept    1.406300e-35
TV           1.467390e-42
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:

```
print(model1.rsquared)

0.61187505085
```

[] Si requerimos todos los parámetros del modelo en un sólo paso, podemos ocupar la siguiente línea de código

```
print(model1.summary())
```

De lo cuál obtenemos

[]

```
OLS Regression Results

  ==============================================================================

Dep. Variable:                   Sales   R-squared:
                0.612
Model:                             OLS   Adj. R-squared:
                0.610
Method:                  Least Squares   F-statistic:
                312.1
Date:                 Sun, 29 Oct 2017   problema (F-
  statistic):           1.47e-42
Time:                         04:11:12   Log-Likelihood:
              -519.05
No. Observations:                  200   AIC:
                1042.
Df Residuals:                      198   BIC:
                1049.
```

```
Df Model:                           1
Covariance Type:            nonrobust

==============================================================================
coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      7.0326      0.458     15.360      0.000       6.130       7.935
TV             0.0475      0.003     17.668      0.000       0.042       0.053
==============================================================================

Omnibus:                        0.531   Durbin-Watson:                   1.935
problema(Omnibus):                  0.767   Jarque-Bera (JB):                 0.669
Skew:                          -0.089   problema(JB):                     0.716
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.

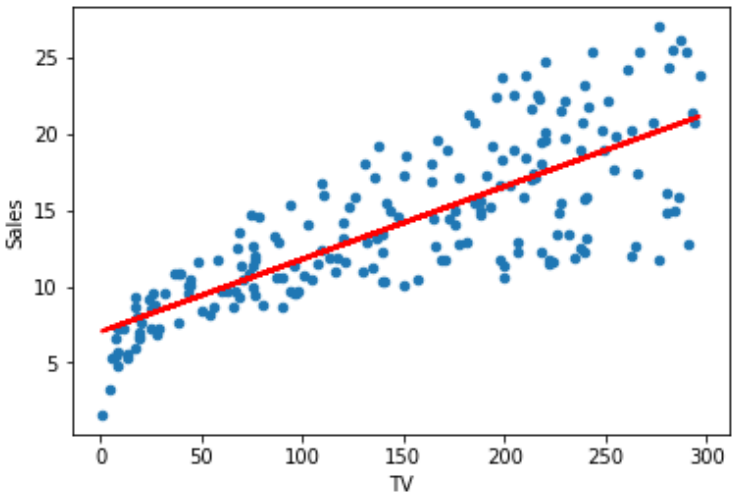
[] Ahora predigamos el valor de las ventas basados en la ecuación que acabamos de encontrar. Esto podemos hacer de la siguiente manera

```
sales_pred=model1.predict(pd.DataFrame(advert['TV']))
print(sales_pred.head())

0    17.970775
1     9.147974
2     7.850224
3    14.234395
4    15.627218
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:

```
import matplotlib.pyplot as plt
advert.plot(kind='scatter', x='TV', y='Sales')
plt.plot(pd.DataFrame(advert['TV']),sales_pred,c='red',
  linewidth=2)
plt.show()
```



[] Ahora calculemos el valor `RSE`

```
advert['sales_pred']=0.047537*advert['TV']+7.03
advert['RSE']=(advert['Sales']-advert['sales_pred'])**2
SSD=advert.sum()['RSE']
n = len(advert["Sales"])
RSE=np.sqrt(SSD/(n-2))
salesmean=np.mean(advert['Sales'])
error=RSE/salesmean
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.

4.6 *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 β_i asociadas con cada predictor.

El modelo se representaría como sigue:

$$Y = \alpha + \beta_1 X_1 + ... + \beta_n X_n \tag{4.26}$$

Cada β_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. 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 vació (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)
```

[,]

```
Intercept    5.774948
TV           0.046901
Newspaper    0.044219
dtype: float64
Intercept    3.145860e-22
TV           5.507584e-44
Newspaper    2.217084e-05
dtype: float64
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

$$\texttt{Ventas} = 5.77 + 0.046 * \texttt{TV} + 0.04 * \texttt{Periódico} \tag{4.27}$$

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:

```
sales_pred=model2.predict(advert[['TV','Newspaper']])
print(sales_pred.head())
```

[,]

```
0    19.626901
1     9.856348
2     9.646055
3    15.467318
4    16.837102
dtype: float64
```

[,] Para calcular el valor `RSE`, utilizamos las siguientes líneas

```
#RSE
import numpy as np
advert['sales_pred']= 5.77 + 0.046901*advert['TV'] + \
0.044219*advert['Newspaper']
advert['SSD'] = (advert['Sales']- \
advert['sales_pred'])**2
SSD=advert.sum()['SSD']
n = len(advert["Sales"])
print("n",n)
p = 2
RSE=np.sqrt(SSD/(n-p-1))
print("RSE", RSE)
salesmean=np.mean(advert['Sales'])
print("salesmean", salesmean)
error=RSE/salesmean
print("error", error)
```

[,]

```
n 200
RSE 3.12072391442
salesmean 14.022500000000003
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

```
print(model2.summary())
```

obtenemos la siguiente tabla que es el resumen del modelo:

[,]

```

OLS Regression Results

  ==============================================================================

Dep. Variable:                   Sales   R-squared:
                0.646
Model:                            OLS   Adj. R-squared:
                0.642
Method:                 Least Squares   F-statistic:
                179.6
Date:                Fri, 03 Nov 2017   problema (F-
  statistic):           3.95e-45
Time:                        21:57:12   Log-Likelihood:
              -509.89
No. Observations:                 200   AIC:
                1026.
Df Residuals:                     197   BIC:
                1036.
Df Model:                           2
Covariance Type:            nonrobust

  ==============================================================================

coef    std err          t      P>|t|      [0.025
  0.975]

  ------------------------------------------------------------------------------

Intercept      5.7749      0.525     10.993      0.000
    4.739       6.811
TV             0.0469      0.003     18.173      0.000
    0.042       0.052
Newspaper      0.0442      0.010      4.346      0.000
    0.024       0.064

  ==============================================================================

Omnibus:                        0.658   Durbin-Watson:
                1.969
problema(Omnibus):                  0.720   Jarque-Bera (JB
  ):                0.415
Skew:                          -0.093   problema(JB):
                0.813
Kurtosis:                       3.122   Cond. No.
                 410.

  ==============================================================================


Warnings:
```

```
27  [1] Standard Errors assume that the covariance matrix of
      the 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
```

```
4    13.223908
dtype: float64
```

[,]

```
#print(model3.summary())
OLS Regression Results

  ==============================================================================

Dep. Variable:                  Sales   R-squared:
                0.897
Model:                            OLS   Adj. R-squared:
                0.896
Method:                 Least Squares   F-statistic:
                859.6
Date:                Sun, 05 Nov 2017   problema (F-
  statistic):           4.83e-98
Time:                        20:09:44   Log-Likelihood:
              -386.20
No. Observations:                 200   AIC:
                778.4
Df Residuals:                     197   BIC:
                788.3
Df Model:                           2
Covariance Type:            nonrobust

  ==============================================================================

coef    std err          t      P>|t|      [0.025
  0.975]

  ------------------------------------------------------------------------------

Intercept      2.9211      0.294      9.919      0.000
    2.340       3.502
TV             0.0458      0.001     32.909      0.000
    0.043       0.048
Radio          0.1880      0.008     23.382      0.000
    0.172       0.204

  ==============================================================================

Omnibus:                       60.022   Durbin-Watson:
                2.081
problema(Omnibus):              0.000   Jarque-Bera (JB):
             148.679
Skew:                          -1.323   problema(JB):
                5.19e-33
Kurtosis:                       6.292   Cond. No.
                425.

  ==============================================================================


Warnings:
[1] Standard Errors assume that the covariance matrix of
  the errors is correctly specified.
```

Observemos que el valor R^2 se ha incrementado considerablemente debido a la adición de la radio al modelo. De la misma manera, el

estadístico$-F$ se incrementado considerablemente del último modelo indicando un modelo altamente eficiente.

El valor `RSE` puede ser calculado usando el mismo método descrito anteriormente:

[,]`advertisingModel3.py` (continuación)

```
#RSE
import numpy as np

advert['SSD'] = (advert['Sales']- \
advert['sales_pred'])**2
SSD=advert.sum()['SSD']
n = len(advert["Sales"])
print("n",n)
p = 2
RSE=np.sqrt(SSD/(n-p-1))
print("RSE", RSE)
salesmean=np.mean(advert['Sales'])
print("salesmean", salesmean)
error=RSE/salesmean
print("error", error)
```

[,]

```
n 200
RSE 1.68136091251
salesmean 14.022500000000003
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 4.6.1. Desarrolle un modelo para

`"Ventas"∼"TV+Radio+Periódico";`

haga un análisis de los estadísticos asociados. Con base en estas observaciones, trata de responde 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 problemalema 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 problemalema 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 problemalema.

`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:

$$X_i = \sum_{j \neq i} a_j X_j \tag{4.28}$$

2. Calcule el valor R^2 para este modelo y denótelo por R_i^2. El valor `VIF` para X_i está dado por

$$\texttt{VIF} = \frac{1}{1 - R_i^2} \tag{4.29}$$

3. - $VIF = 1$: Los predictores no están correlacionados.
   - $1 < VIF < 5$: Las predictores estás 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`

```
modelN = smf.ols(formula = "Newspaper~TV+Radio", data =
  advert).fit()
r2N = modelN.rsquared
VIFN = 1/(1-r2N)
print("VIF(Newspaper):", VIFN)

modelT = smf.ols(formula = "TV~Newspaper+Radio", data =
  advert).fit()
r2T = modelT.rsquared
VIFT = 1/(1-r2T)
print("VIF(TV):", VIFT)

modelR = smf.ols(formula = "Radio~TV+Newspaper", data =
  advert).fit()
r2R = modelR.rsquared
VIFR = 1/(1-r2R)
print("VIF(Radio):", VIFR)
```

[,] Del cual obtenemos los siguientes resultados

```
VIF(Newspaper): 1.14518737872
VIF(TV): 1.00461078494
VIF(Radio): 1.14495191711
```

Los predictores `Newpaper` 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 `Newpaper` del modelo y escoger el modelo 3 como el mejor candidato para el modelo final:

$$\texttt{Ventas} = 2.92 + 0.45 * \texttt{TV} + 0.18 * \texttt{Radio} \tag{4.30}$$

4.7 *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 contante todas fuentes de datos similares o no.

Esto nos ayuda a detectar algún `problemalema del 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 diviendo 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`

```
import pandas as pd
import statsmodels.formula.api as smf

advert = pd.read_csv("./dataBases/Advertising.csv")
model3=smf.ols(formula='Sales~TV+Radio',data=advert).fit()
sales_pred=model3.predict(advert[['TV','Radio']])
print(sales_pred.head())

print(model3.summary())

import numpy as np

N = len(advert)
arr =  np.arange(N)
np.random.shuffle(arr)
check = arr < 0.8*N
training=advert[check].copy()
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:

[,]

```
import statsmodels.formula.api as smf
model5=smf.ols(formula='Sales~TV+Radio',data=training).fit()
print(model5.summary())
```

[,]print(model3.summary())

```
OLS Regression Results

==============================================================================

Dep. Variable:                  Sales   R-squared:
                0.897
Model:                            OLS   Adj. R-squared:
                0.896
Method:                 Least Squares   F-statistic:
                859.6
Date:                Sun, 12 Nov 2017   problema (F-
  statistic):           4.83e-98
Time:                        22:40:26   Log-Likelihood:
              -386.20
No. Observations:                 200   AIC:
                778.4
Df Residuals:                     197   BIC:
                788.3
Df Model:                           2
Covariance Type:            nonrobust

==============================================================================

coef    std err          t      P>|t|      [0.025
  0.975]

------------------------------------------------------------------------------

Intercept      2.9211      0.294      9.919      0.000
    2.340       3.502
TV             0.0458      0.001     32.909      0.000
    0.043       0.048
Radio          0.1880      0.008     23.382      0.000
    0.172       0.204

==============================================================================

Omnibus:                       60.022   Durbin-Watson:
                2.081
problema(Omnibus):                0.000   Jarque-Bera (JB
  ):              148.679
Skew:                          -1.323   problema(JB):
                5.19e-33
Kurtosis:                       6.292   Cond. No.
                425.

==============================================================================
```

```
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:
                0.906
4   Model:                            OLS   Adj. R-squared:
                0.904
5   Method:                 Least Squares   F-statistic:
                753.7
6   Date:                Sun, 12 Nov 2017   problema (F-
      statistic):           3.22e-81
7   Time:                        22:40:26   Log-Likelihood:
              -299.60
8   No. Observations:                 160   AIC:
                605.2
9   Df Residuals:                     157   BIC:
                614.4
10  Df Model:                           2
11  Covariance Type:            nonrobust
12
      ==============================================================================
13  coef    std err          t      P>|t|      [0.025
      0.975]
14
      ------------------------------------------------------------------------------
15  Intercept      2.8708      0.318      9.035      0.000
      2.243       3.498
16  TV             0.0448      0.001     30.797      0.000
      0.042       0.048
17  Radio          0.1959      0.009     22.803      0.000
      0.179       0.213
18
      ==============================================================================
19  Omnibus:                       11.888   Durbin-Watson:
                2.158
20  problema(Omnibus):                0.003   Jarque-Bera (JB
      ):               13.175
21  Skew:                          -0.696   problema(JB):
                0.00138
22  Kurtosis:                       2.807   Cond. No.
                438.
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:

$$\texttt{Ventas} \sim 2.86 + 0.04 * \texttt{TV} + 0.17 * \texttt{Radio} \tag{4.31}$$

[,] Ahora, predigamos los valores de las ventas para los valores de prueba:

```
sales_pred=model5.predict(training[['TV','Radio']])
sales_pred
```

El valor `RSE` para esta predicción en el conjunto de datos de prueba puede ser calculadas usando el siguiente pedazo de código:

[,]

```
testing['sales_pred']=2.86 + 0.04*testing['TV'] + 0.17*
  testing['Radio']
n = len(testing)
p = 2
testing['SSD']=(testing['Sales']-testing['sales_pred'])**2
SSD=testing.sum()['SSD']
RSE=np.sqrt(SSD/(n-p-1))
salesmean=np.mean(testing['Sales'])
error=RSE/salesmean
print(RSE,salesmean,error)
##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.

4.8 *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	$1.47e^{-42}$	3.25 (23%)
Model 2	Sales ~ TV+Newspaper	0.646/0.642	179.6	$3.95e^{-45}$	3.12(22%)
Model 3	Sales ~ TV+Radio	0.897/0.896	859.6	$4.83e^{-98}$	1.71(12%)
Model 4	Sales ~ TV+Radio+Newspaper	0.897/0.896	570.3	$1.58e^{-96}$	1.80(13%)

Figura 4.1: 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`.

4.9 *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`

```
advert = pd.read_csv("./dataBases/Advertising.csv")
feature_cols = ['TV', 'Radio']
X = advert[feature_cols]
Y = advert['Sales']
trainX,testX,trainY,testY = train_test_split(X,Y, test_size
   = 0.2)
lm = LinearRegression()
lm.fit(trainX, trainY)
```

[,] El siguiente código nos devuelve los parámetros

```
print(lm.intercept_)
for _ in zip(feature_cols, lm.coef_):
print(_)
```

[,] El valor R^2 se obtiene de la siguiente manera

```
print("R2 =", lm.score(trainX, trainY))
```

[,] Un resultado típico de este código sería

```
2.73465274245
('TV', 0.046830472078714387)
('Radio', 0.18642021416992249)
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:

```
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.

Veamos cómo se puede hacer una selección de funciones en `scikit-learn`:

[,]`RFE.py`

```
import pandas as pd
advert = pd.read_csv("./dataBases/Advertising.csv")

from sklearn.feature_selection import RFE
from sklearn.svm import SVR
feature_cols = ['TV', 'Radio','Newspaper']
X = advert[feature_cols]
Y = advert['Sales']
estimator = SVR(kernel="linear")
selector = RFE(estimator,2,step=1)
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:

```
print(selector.support_)
##[ True  True False]
```

En nuestro caso, `X` consta de tres variables: `TV`, `Radio` y `Newpaper`. 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:

```
print(selector.ranking_)
##[1 1 2]
```

Observación. 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.

4.10 *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`

```
import pandas as pd
df=pd.read_csv('./dataBases/EcomExpense.csv')
print(df.head())
```

[,] Deberíamos obtener los siguientes resultados:

```
Transaction ID  Age    Items   Monthly Income  Transaction
  Time  Record  \
0          TXN001    42       10            7313
  627.668127       5
1          TXN002    24        8           17747
  126.904567       3
2          TXN003    47       11           22845
  873.469701       2
```

```
3          TXN004     50        11            18552
  380.219428        7
4          TXN005     60         2            14439
  403.374223        2

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

$$Y_{\texttt{model}} = \alpha + \sum \beta_i X_i \tag{4.32}$$

en la cual algunas X_i pueden ser categóricas. Digamos que X_g es tal variable.

En nuestro ejemplo,

$$X_g = \begin{cases} 1 & \texttt{cliente masculino} \\ 0 & \texttt{cliente femenino} \end{cases} \tag{4.33}$$

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:

$$X_{t1} = \begin{cases} 1 & \texttt{"City Tier"} = 1 \\ 0 & \texttt{"City Tier"} \neq 1 \end{cases} \tag{4.34}$$

$$X_{t2} = \begin{cases} 1 & \texttt{"City Tier"} = 2 \\ 0 & \texttt{"City Tier"} \neq 2 \end{cases} \tag{4.35}$$

Entonces, el modelo puede ser alguno de los siguientes

$$Y = \begin{cases} \alpha + \beta_1 X_1 + ... + \beta_{t1} X_{t1} + ... + b_n X_n & \texttt{"City Tier"=1} \\ \alpha + \beta_1 X_1 + ... + \beta_{t1} X_{t2} + ... + b_n X_n & \texttt{"City Tier"=2} \\ \alpha + \beta_1 X_1 + ... + + b_n X_n & \texttt{"City Tier"=3} \end{cases} \tag{4.36}$$

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. 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:

[,]

```
dummy_gender=pd.get_dummies(df['Gender'],prefix='Sex')
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`:

[,]

```
City_Tier 1  City_Tier 2  City_Tier 3
0            1            0            0
1            0            1            0
2            0            1            0
3            1            0            0
4            0            1            0
```

[,] `dummy_gender` es similar a la siguiente tabla:

```
Sex_Female  Sex_Male
0           1         0
1           1         0
2           0         1
3           1         0
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:

[,]

```
Transaction ID  Age    Items   Monthly Income  Transaction
  Time  Record  \
0         TXN001    42       10            7313
  627.668127       5
1         TXN002    24        8           17747
  126.904567       3
2         TXN003    47       11           22845
  873.469701       2
3         TXN004    50       11           18552
  380.219428       7
4         TXN005    60        2           14439
  403.374223       2

Gender City Tier  Total Spend  Sex_Female  Sex_Male
  City_Tier 1  \
0  Female    Tier 1  4198.385084           1         0
         1
1  Female    Tier 2  4134.976648           1         0
         0
2    Male    Tier 2  5166.614455           0         1
         0
```

```
3  Female     Tier 1  7784.447676            1          0
        1
4  Female     Tier 2  3254.160485            1          0
        0

City_Tier 2  City_Tier 3
0            0            0
1            1            0
2            1            0
3            0            0
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.

Veamos 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:

[,]

```
from sklearn.linear_model import LinearRegression
feature_cols = ['Monthly Income','Transaction Time','
  City_Tier 1',\
'City_Tier 2','City_Tier 3','Sex_Female','Sex_Male']
X = df2[feature_cols]
Y = df2['Total Spend']
lm = LinearRegression()
lm.fit(X,Y)
```

[,] Los parámetros del modelo se pueden encontrar de la siguiente manera:

```
print(lm.intercept_)
for _ in zip(feature_cols, lm.coef_):
print(_)
```

[,]

```
3655.72940769
('Monthly Income', 0.15297824609320512)
('Transaction Time', 0.12372608642619998)
('City_Tier 1', 119.66325160390086)
('City_Tier 2', -16.67901800799039)
('City_Tier 3', -102.9842335959104)
('Sex_Female', -94.157798830320132)
('Sex_Male', 94.157798830320118)
```

[,] El valor R^2 para este modelo se puede encontrar escribiendo lo siguiente:

```
R2 = lm.score(X,Y)
print(R2)
## 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:

$$\texttt{Total\_Spend} = 3655.72 + 0.12 * \texttt{Transaction Time} \tag{4.37}$$

$$+ 0.15 * \texttt{Monthly Income} + 119 * \texttt{City\_Tier 1} \tag{4.38}$$

$$- 16 * \texttt{City\_Tier 2} - 102 * \texttt{City\_Tier 3} \tag{4.39}$$

$$- 94 * \texttt{Sex\_Female} + 94 * \texttt{Sex\_Male} \tag{4.40}$$

[,] El RSE se puede calcular de la siguiente manera:

```
import numpy as np
df2['total_spend_pred']=3720.72940769 + 0.12*df2['
  Transaction Time']+ \
0.15*df2['Monthly Income']+119*df2['City_Tier 1']-16*df2['
  City_Tier 2']
-102*df2['City_Tier 3']-94*df2['Sex_Female']+94*df2['
  Sex_Male']
df2['RSE']=(df2['Total Spend']-df2['total_spend_pred'])**2
RSEd=df2.sum()['RSE']
RSE=np.sqrt(RSEd/2354)
salesmean=np.mean(df2['Total Spend'])
error=RSE/salesmean
print(RSE,salesmean,error)
##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	*Total_Spend=3655.72 + 0.12*Transaction Time + 0.15*Monthly Income + 119*City_Tier 1 +94*Sex_Male*
Male	2	*Total_Spend=3655.72 + 0.12*Transaction Time + 0.15*Monthly Income -16*City_Tier 2 +94*Sex_Male*
Male	3	*Total_Spend=3655.72 + 0.12*Transaction Time + 0.15*Monthly Income - 102*City_Tier 3 +94*Sex_Male*
Female	1	*Total_Spend=3655.72 + 0.12*Transaction Time + 0.15*Monthly Income + 119*City_Tier 1 - 94*Sex_Female*
Female	2	*Total_Spend=3655.72 + 0.12*Transaction Time + 0.15*Monthly Income -16*City_Tier 2 - 94*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`

```
import pandas as pd
data = pd.read_csv('./dataBases/Auto.csv')
print(data.head())
```

[,]

```
mpg  cylinders  displacement  horsepower  weight
  acceleration  \
0  18.0          8         307.0       130.0    3504
     12.0
1  15.0          8         350.0       165.0    3693
     11.5
2  18.0          8         318.0       150.0    3436
     11.0
3  16.0          8         304.0       150.0    3433
     12.0
4  17.0          8         302.0       140.0    3449
     10.5

model year  origin                   car name
0          70       1  chevrolet chevelle malibu
1          70       1          buick skylark 320
2          70       1         plymouth satellite
3          70       1              amc rebel sst
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:

```
import matplotlib.pyplot as plt
#%matplotlib inline
data['mpg']=data['mpg'].dropna()
data['horsepower']=data['horsepower'].dropna()
plt.plot(data['horsepower'],data['mpg'],'ro')
plt.xlabel('Horsepower')
plt.ylabel('MPG (Miles Per Gallon)')
```

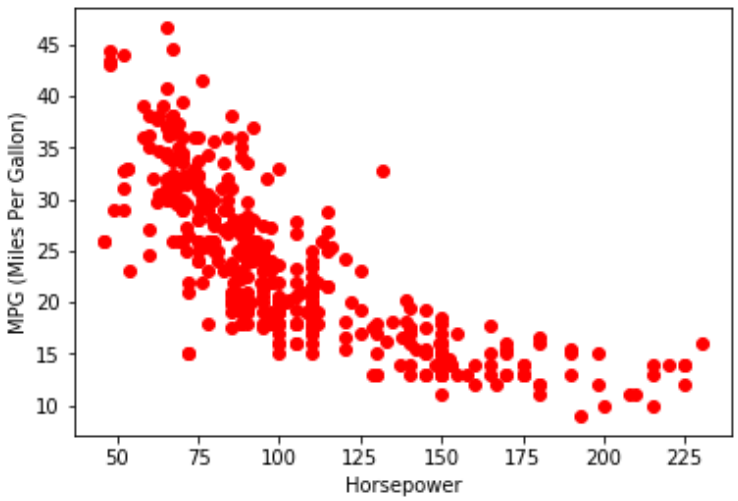


Figura 4.2: `HP vs MPG`

Aunque hemo supuesto que el modelo es lineal

$$mpg = c_0 + c_1 * \texttt{hp} \tag{4.41}$$

en realidad parece más un *modelo cuadrático*

$$mpg = c_0 + c_1 * \texttt{hp} + c_2 \texttt{hp}^2 \tag{4.42}$$

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 utilizado en el modelo. También simultáneamente, creemos un modelo asumiendo un lineal relación entre mpg y cuadrado de potencia:

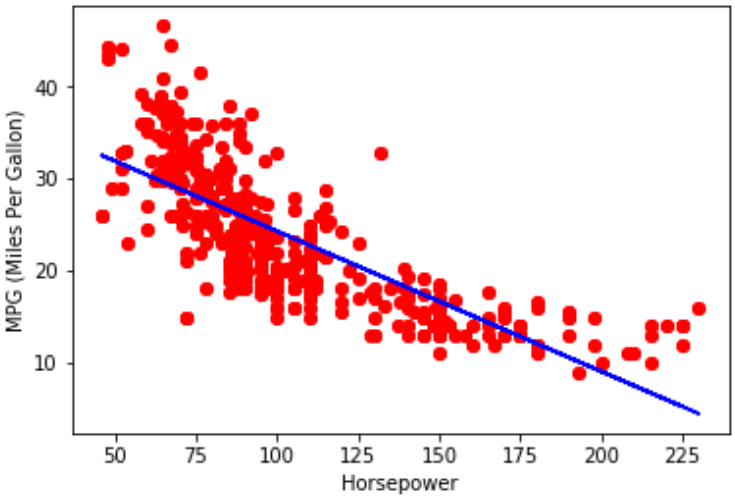
[,]

```
import numpy as np
from sklearn.linear_model import LinearRegression
X=data['horsepower'].fillna(data['horsepower'].mean())
Y=data['mpg'].fillna(data['mpg'].mean())
lm=LinearRegression()
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:
[,]

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot(data['horsepower'],data['mpg'],'ro')
plt.plot(X,lm.predict(X[:,np.newaxis]),color='blue')
```



[,]

```
R2 = lm.score(X[:,np.newaxis],Y)
print(R2)
## 0.574653340645

RSEd=(Y-lm.predict(X[:,np.newaxis]))**2
RSE=np.sqrt(np.sum(RSEd)/389)
ymean=np.mean(Y)
error=RSE/ymean
print(RSE,error)
## 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`

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn import linear_model
X=data['horsepower'].fillna(data['horsepower'].mean())
Y=data['mpg'].fillna(data['mpg'].mean())
poly = PolynomialFeatures(degree=2)
X_ = poly.fit_transform(X[:,np.newaxis])
```

```
clf = linear_model.LinearRegression()
clf.fit(X_, Y)

print(clf.intercept_)
##55.0261924471
print(clf.coef_)
##[ 0.          -0.43404318   0.00112615]
```

El modelo se puede escribir entonces como

$$\texttt{mpg} = 55.02 - 0.43 * \texttt{hp} + 0.001 * \texttt{hp}^2 \tag{4.43}$$

[,] Con el siguiente fragmento de código, podemos visualizar el modelo cuadrático:

```
plt.plot(data['horsepower'],data['mpg'],'ro')
plt.plot(X,clf.predict(X_), "bo")
plt.show()
```