Estudio de la función de Distribución Geométrica

María Baeza López, Juan Jesús Dóniz Labrador y Jesús Rodríguez Falcón

Matemáticas

La Laguna,15 de Mayo de 2014

Indice

Indice

Indice

Indice

Indice

Introducción

En este trabajo hemos elaborado un función Python que resuelva problemas matemáticos asociados a la función de distribución geométrica. Para ello hemos recopilado información sobre dicha función,también hemos probado y verificado que la función que se ha implementado funciona correctamente,además de comprobar la eficacia de la misma.

Distribución Geométrica

La distribución geométrica es un modelo adecuado para aquellos procesos en los que se repiten pruebas consecutivamente hasta obtener el resultado correcto o deseado (éxito). Esta distribución se puede expresar de las siguientes maneras:

- → La distribución de probabilidad del número X necesaria para obtener un éxito, contenido en el conjunto $\{1, 2, 3, ...\}$

Distribución Geométrica

La distribución geométrica es un modelo adecuado para aquellos procesos en los que se repiten pruebas consecutivamente hasta obtener el resultado correcto o deseado (éxito). Esta distribución se puede expresar de las siguientes maneras:

- $\rightarrow$ La distribución de probabilidad del número X necesaria para obtener un éxito, contenido en el conjunto $\{1, 2, 3, ...\}$
- $\rightarrow$ La distribución del número $Y = X - 1$ de fallos del primer éxito,contenido en el conjunto $\{1, 2, 3, ...\}$

Función de Probabilidad

La función de probabilidad de esta distribución se puede escribir de dos maneras posibles, dependiendo la definición de la función de distribución:

$\rightarrow$ Si la probabilidad de éxito en cada caso es p, entonces la probabilidad de que x ensayos sean necesarios para obtener un éxito es:

Función de Probabilidad

La función de probabilidad de esta distribución se puede escribir de dos maneras posibles, dependiendo la definición de la función de distribución:

→ Si la probabilidad de éxito en cada caso es p, entonces la probabilidad de que x ensayos sean necesarios para obtener un éxito es:

$$P(X = x) = (1 - p)^{x-1} \cdot p \ \ , \text{ para } \ x = 0, 1, 2, 3, \ldots$$

Función de Probabilidad

La función de probabilidad de esta distribución se puede escribir de dos maneras posibles, dependiendo la definición de la función de distribución:

→ Si la probabilidad de éxito en cada caso es p, entonces la probabilidad de que x ensayos sean necesarios para obtener un éxito es:

$$P(X = x) = (1 - p)^{x-1} \cdot p \quad , \text{ para } \; x = 0, 1, 2, 3, ...$$

→ Equivalentemente, la probabilidad de que halla x fallos antes del primer éxito es:

Función de Probabilidad

La función de probabilidad de esta distribución se puede escribir de dos maneras posibles, dependiendo la definición de la función de distribución:

→ Si la probabilidad de éxito en cada caso es p, entonces la probabilidad de que x ensayos sean necesarios para obtener un éxito es:

$$P(X = x) = (1 - p)^{x-1} \cdot p \ \ , \text{ para } \ x = 0, 1, 2, 3, \ldots$$

→ Equivalentemente, la probabilidad de que halla x fallos antes del primer éxito es:

$$P(X = x) = (1 - p)^{x} \cdot p \ \ , \text{ para } \ x = 0, 1, 2, 3, \ldots$$

Propiedades de la función Geométrica

→ La media:

$$E(X) = \frac{1}{p} \quad \text{o} \quad E(Y) = \frac{1-p}{p}$$

Propiedades de la función Geométrica

→ La media:

$$E(X) = \frac{1}{p} \quad \text{o} \quad E(Y) = \frac{1-p}{p}$$

→ La varianza:

$$V(X) = V(Y) = \frac{1-p}{p^2}$$

Propiedades de la función Geométrica

→ La media:

$$E(X) = \frac{1}{p} \quad \text{o} \quad E(Y) = \frac{1-p}{p}$$

→ La varianza:

$$V(X) = V(Y) = \frac{1-p}{p^2}$$

→ La moda:es el valor de la variable que tiene asociada la mayor probabilidad. Es fácil ver que:

$$P(x_i) \leq P(X = 1) \ \forall x_i$$

Propiedades de la función Geométrica

→ La media:

$$E(X) = \frac{1}{p} \quad \text{o} \quad E(Y) = \frac{1-p}{p}$$

→ La varianza:

$$V(X) = V(Y) = \frac{1-p}{p^2}$$

→ La moda:es el valor de la variable que tiene asociada la mayor probabilidad. Es fácil ver que:

$$P(x_i) \leq P(X = 1)\ \forall x_i$$

→ La mediana se define como:

$$Me = \frac{-ln(2)}{ln(q)}$$

Funciones utilizadas en Python

En primer lugar implementamos dos funciones que resolvieran los diversos problemas relacionados con la distribución geométrica: ▸ Código de funciones
Después de comprobar que ambas funcionaban correctamente, comenzamos a implementar el código principal que cubría todos los casos posibles, quedando como sigue: ▸ Código principal
Por último, hemos comprobado la exactitud del programa buscando problemas concretos, que podamos realizar tanto "a mano" (o con una calculadora especifica de internet) como con nuestra función, y la eficacia del código al comprobar los tiempos de ejecución.

Funciones utilizadas en Python

En primer lugar implementamos dos funciones que resolvieran los diversos problemas relacionados con la distribución geométrica: ▸ Código de funciones
Después de comprobar que ambas funcionaban correctamente, comenzamos a implementar el código principal que cubría todos los casos posibles, quedando como sigue: ▸ Código principal
Por último, hemos comprobado la exactitud del programa buscando problemas concretos, que podamos realizar tanto "a mano" (o con una calculadora especifica de internet) como con nuestra función, y la eficacia del código al comprobar los tiempos de ejecución.

Ejemplo

Problema 2: Tantos en baloncesto

Un jugador de baloncesto no cesa en su intento de lanzar pelotas a la canasta que se halla situada a 2 metros de altura hasta que consiga introducir una de éstas a través del aro. Si se supone que sus tiros son independientes y que la probabilidad de anotar una canasta es de 0.8,¿ cuál es la probabilidad de que el baloncestista necesite realizar dos tiros?¿ y de que sean tres tiros, cuatro tiros, cinco tiros, etc. hasta deducir la fórmula para n tiros?¿ cuál es la probabilidad de necesitar como mucho cinco tiros?

Solución:

Supongamos que tenemos la siguiente variable aleatoria:

X = número de tiros necesarios por el baloncestista hasta anotar una canasta

Ahora bien, sabemos que los valores de p y q son los siguientes:

$$p = 0{,}8; \; q = 1 - p = 1 - 0{,}8 = 0{,}2$$

Ejemplo

Procedamos por tanto, a calcular dichas probabilidades:

$\rightarrow$ $P[X = 1] = p = 0{,}8$

Ejemplo

Procedamos por tanto, a calcular dichas probabilidades:

- $\rightarrow$ $P[X = 1] = p = 0{,}8$
- $\rightarrow$ $P[X = 2] = q \cdot p = 0{,}8 \cdot 0{,}2 = 0{,}16$

Ejemplo

Procedamos por tanto, a calcular dichas probabilidades:

- $\rightarrow$ $P[X = 1] = p = 0{,}8$
- $\rightarrow$ $P[X = 2] = q \cdot p = 0{,}8 \cdot 0{,}2 = 0{,}16$
- $\rightarrow$ $P[X = 3] = q \cdot q \cdot p = 0{,}8 \cdot 0{,}8 \cdot 0{,}2 = 0{,}032$

Ejemplo

Procedamos por tanto, a calcular dichas probabilidades:

- $\rightarrow$ $P[X = 1] = p = 0{,}8$
- $\rightarrow$ $P[X = 2] = q \cdot p = 0{,}8 \cdot 0{,}2 = 0{,}16$
- $\rightarrow$ $P[X = 3] = q \cdot q \cdot p = 0{,}8 \cdot 0{,}8 \cdot 0{,}2 = 0{,}032$
- $\rightarrow$ $P[X = 4] = q \cdot q \cdot q \cdot p = 0{,}8 \cdot 0{,}8 \cdot 0{,}8 \cdot 0{,}2 = 0{,}0064$

Ejemplo

Procedamos por tanto, a calcular dichas probabilidades:

- $\rightarrow$ $P[X = 1] = p = 0{,}8$
- $\rightarrow$ $P[X = 2] = q \cdot p = 0{,}8 \cdot 0{,}2 = 0{,}16$
- $\rightarrow$ $P[X = 3] = q \cdot q \cdot p = 0{,}8 \cdot 0{,}8 \cdot 0{,}2 = 0{,}032$
- $\rightarrow$ $P[X = 4] = q \cdot q \cdot q \cdot p = 0{,}8 \cdot 0{,}8 \cdot 0{,}8 \cdot 0{,}2 = 0{,}0064$
- $\rightarrow$ $P[X = 5] = q \cdot q \cdot q \cdot q \cdot p = 0{,}8 \cdot 0{,}8 \cdot 0{,}8 \cdot 0{,}8 \cdot 0{,}2 = 0{,}00128$

Luego, de aquí es fácil apreciar que:

Fórmula

$$P[X = n] = q^{n-1} \cdot p$$

Ejemplo

Procedamos por tanto, a calcular dichas probabilidades:

- $\rightarrow$ $P[X = 1] = p = 0{,}8$
- $\rightarrow$ $P[X = 2] = q \cdot p = 0{,}8 \cdot 0{,}2 = 0{,}16$
- $\rightarrow$ $P[X = 3] = q \cdot q \cdot p = 0{,}8 \cdot 0{,}8 \cdot 0{,}2 = 0{,}032$
- $\rightarrow$ $P[X = 4] = q \cdot q \cdot q \cdot p = 0{,}8 \cdot 0{,}8 \cdot 0{,}8 \cdot 0{,}2 = 0{,}0064$
- $\rightarrow$ $P[X = 5] = q \cdot q \cdot q \cdot q \cdot p = 0{,}8 \cdot 0{,}8 \cdot 0{,}8 \cdot 0{,}8 \cdot 0{,}2 = 0{,}00128$

Luego, de aquí es fácil apreciar que:

Fórmula

$$P[X = n] = q^{n-1} \cdot p$$

Veamos ahora cuál es la probabilidad de necesitar como máximo cinco tiros para encestar en la canasta. Atendiendo a los resultados del apartado anterior nos queda:

$$P[X \leq 5] = 0{,}8 + 0{,}16 + 0{,}032 + 0{,}0064 + 0{,}00128 = 0{,}99968$$

Ejemplo

En Python se calcularía:

→ En primer lugar meteríamos en Konsole la sentencia: **python geo_ sol.py n 0.8**, donde n va variando entre 1 hasta 5

→ Seguidamente el programa te manda a elegir una opción de las ofertadas: para las cinco primeras veces escogeremos la **opción 0** y para la última la **opcion 1**:

- La **opción 0** es la más sencilla y solo utiliza la función: **def calcular_ geo (n,p)**
- La **opcion 1** necesita de las dos funciones implementadas: la anterior llamada dentro de la función **def calcular_ geo1 (n,p)**

→ Por último, el programa te dará los resultados y el tiempo que ha tardado en calcularlo.

En la siguiente tabla se muestran los resultados obtenidos durante la comprobación de la función:

Problemas:	**p**	**Resultado Python (1)**	**Resultado "a mano"(2)**	**Calculadora Online**(3)	**Error entre (1) y (2)**	**Error entre (1) y (3)**
1	0.5	$P[X \geq 2] = 0{,}25$	$\frac{1}{4}$	0.25	0	0
2	0.8	$P[X = 1] = 0{,}8$	0.8	0.8	0	0
	0.8	$P[X = 2] = 0{,}16$	0.16	0.16	0	0
	0.8	$P[X = 3] = 0{,}032$	0.032	0.0032	0	0
	0.8	$P[X = 4] = 0{,}0064$	0.0064	0.0064	0	0
	0.8	$P[X = 5] = 0{,}001280$	0.0013	0.00128	$2 \cdot 10^{-6}$	0
	0.8	$P[X \leq 0] = 0{,}999680$	0.99968	0.99968	0	0
3	0.16666666	$P[X = 3] = 0{,}115741$	0.1157	0.115773796296	$4{,}1 \cdot 10^{-5}$	$3{,}04 \cdot 10^{-6}$
4	0.4	$P[X < 3] = 0{,}64$	0.64	0.64	0	0

Como se puede observar los errores que se producen son nulos o cantidades mínimas, debido sobre todo a las aproximaciones de los valores p introducidos al programa.

Tiempos obtenidos

En la siguiente gráfica podemos observar los tiempos (en segundos) obtenidos al hacer los cálculos con el valor $p = 0{,}3$:

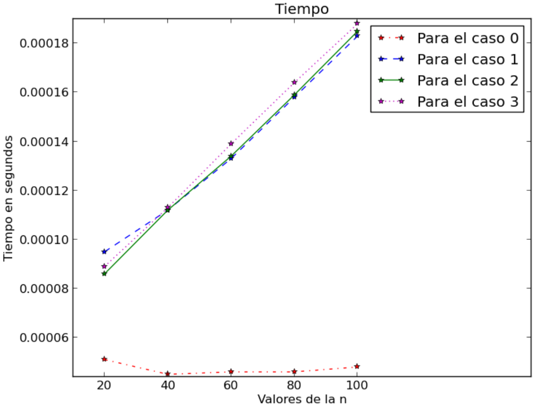

Figura : Gráfica de resultados

Conclusiones

Podemos sacar como conclusión final que:

1. Debemos ejecutar varias veces un programa, hasta purificarlo y sacar un programa eficiente y operativo.

Conclusiones

Podemos sacar como conclusión final que:

1. Debemos ejecutar varias veces un programa, hasta purificarlo y sacar un programa eficiente y operativo.
2. Además, gracias a nuestra búsqueda de información hemos podido refrescar y ampliar conocimientos sobre probabilidades, más concretamente sobre la distribución geométrica.

Conclusiones

Podemos sacar como conclusión final que:

1. Debemos ejecutar varias veces un programa, hasta purificarlo y sacar un programa eficiente y operativo.
2. Además, gracias a nuestra búsqueda de información hemos podido refrescar y ampliar conocimientos sobre probabilidades, más concretamente sobre la distribución geométrica.
3. Nos hemos dado cuenta del enorme potencial que tiene la utilización de LaTeX, Beamer y Python, para la elaboración documental, de presentación y de creación de algoritmos, respectivamente. Nos será de gran ayuda en la elaboración de nuestros escritos, en el marco de nuestra formación académica universitaria y laboral.

Bibliografía

- Ejercicios resueltos de probabilidad (Año 2001) Salazar González JJ y López Yurda M
- Estadistica I.Probabilidad y distribuciones (Año 2000) Casas Sánchez JM y Zamora Sanz A
- Calculadora de la distribución geométrica (2011) *http://www.elektro-energetika.cz/calculations/distrgeo.php*
- Distribución geométrica (Wikipedia) *http://es.wikipedia.org*

Código funciones

```
def calcular_geo (n,p):
  if (n>1):
    q=1-p
    probabilidad=(q**(n-1))*p
  else:
    probabilidad=p
  return (probabilidad)

def calcular_geo1 (n,p):
  probabilidad=0
  for i in range (1,n+1):
    sumaprobabilidad=calcular_geo(i,p)
    probabilidad+=sumaprobabilidad
  return (probabilidad)
```

Funciones utilizadas en Python

Código funciones

```
def calcular_geo (n,p):
  if (n>1):
    q=1-p
    probabilidad=(q**(n-1))*p
  else:
    probabilidad=p
  return (probabilidad)

def calcular_geo1 (n,p):
  probabilidad=0
  for i in range (1,n+1):
    sumaprobabilidad=calcular_geo(i,p)
    probabilidad+=sumaprobabilidad
  return (probabilidad)
```

Funciones utilizadas en Python

Código principal

```
# Menu principal
argumentos = sys.argv[1:]
if (len(argumentos) == 2):
  n = int(argumentos[0])
  p = float(argumentos[1])
else:
    print "Introduzca el nº de pruebas necesarias para
    obtener un exito (n>0):"
    n = int (raw_input())
    print "Introduzca el valor p (p>0):"
    p = float(raw_input())
if (n > 0):
  print "¿Que tipo de probabilidad vas a hallar?
  (0=P(X=n), 1=P(X<=n), 2=P(X<n), 3=P(X>=n))"
  respuesta=int(raw_input())
```

Código principal

```
# Menu principal
argumentos = sys.argv[1:]
if (len(argumentos) == 2):
  n = int(argumentos[0])
  p = float(argumentos[1])
else:
    print "Introduzca el nº de pruebas necesarias para
    obtener un exito (n>0):"
    n = int (raw_input())
    print "Introduzca el valor p (p>0):"
    p = float(raw_input())
if (n > 0):
  print "¿Que tipo de probabilidad vas a hallar?
  (0=P(X=n), 1=P(X<=n), 2=P(X<n), 3=P(X>=n))"
  respuesta=int(raw_input())
```

Código principal

```
if (respuesta==0):
  start=time.time()
  probabilidad=mod_geo.calcular_geo (n,p)
  finish=time.time()-start
  print "La probabilidad P[X= %d] con p= %f es: %f
  y he tardado %f segundos en calcularlo"
  %(n,p,probabilidad,finish)
elif (respuesta==1):
  start=time.time()
  probabilidad=mod_geo.calcular_geo1(n,p)
  finish=time.time()-start
  print "La probabilidad P[X<= %d] con p= %f es: %f
  y he tardado %f segundos en calcularlo"
  %(n,p,probabilidad,finish)
```

Código principal

```
  elif (respuesta == 2):
    start=time.time()
    probabilidad=mod_geo.calcular_geo1(n-1,p)
    finish=time.time()-start
    print "La probabilidad P[X< %d] con p= %f es: %f
    y he tardado %f segundos en calcularlo" %(n,p,probabilidad
  else:
    start=time.time()
    prob=mod_geo.calcular_geo1(n,p)
    probabilidad = 1 - prob
    finish=time.time()-start
    print "La probabilidad P[X>= %d] con p= %f es: %f
    y he tardado %f segundos en calcularlo" %(n,p,probabilidad
else:
    print "No podemos hallar la probabilidad"
```

Funciones utilizadas en Python

Código principal

```
  elif (respuesta == 2):
    start=time.time()
    probabilidad=mod_geo.calcular_geo1(n-1,p)
    finish=time.time()-start
    print "La probabilidad P[X< %d] con p= %f es: %f
    y he tardado %f segundos en calcularlo" %(n,p,probabilidad
  else:
    start=time.time()
    prob=mod_geo.calcular_geo1(n,p)
    probabilidad = 1 - prob
    finish=time.time()-start
    print "La probabilidad P[X>= %d] con p= %f es: %f
    y he tardado %f segundos en calcularlo" %(n,p,probabilidad
else:
    print "No podemos hallar la probabilidad"
```

Funciones utilizadas en Python