



Función de distribución. Hipergeométrica

Adriana Calvo - Misael Enrique Peraza Luis - Carolina
Yanes Rivero

Técnicas Experimentales. 1^{er} curso. 2^{do} semestre

Grupo 3F

Facultad de Matemáticas

Universidad de la Laguna

La Laguna 11 de Mayo de 2014

Resumen

El propósito de esta exposición consiste en describir de manera objetiva los distintos procedimientos y herramientas utilizadas para llevar a cabo la implementación de la función de distribución de la variable aleatoria hipergeométrica en Python.

El objetivo principal de este proyecto consiste en aprender a manejar un lenguaje de programación denominado Python, además de usarlo como una herramienta básica en el ámbito matemático. En este trabajo también se encuentran otros objetivos como el de familiarizarse con \LaTeX y Beamer, herramientas básicas en la redacción y exposición de informes y proyectos.

Índice general

1. Motivación y objetivos	1
2. Fundamentos teóricos	2
2.1. Variables aleatorias	2
2.1.1. Variable aleatoria continua	2
2.1.2. Variable aleatoria discreta	3
2.2. Variable aleatoria hipergeométrica :	3
3. Procedimiento Experimental	4
3.1. Descripción de los experimentos	4
3.1.1. Poker de Ases	4
3.1.2. Color	5
3.2. Descripción del material	5
3.3. Resultados obtenidos	6
3.4. Análisis de los resultados	7
4. Conclusiones	8
A. Algoritmo Poker de Ases	9
B. Algoritmo Color	10
C. Algoritmo F.D Hipergeométrica	11
D. F.D Hipergeométrica Implementada	13

Capítulo 1

Motivación y objetivos

Durante el curso se han visto diferentes métodos matemáticos con el fin de entenderlos y hacer uso de ellos, formulando programas en Python que permitan usarlos de una manera más rápida, sencilla y eficaz. Con este trabajo se pretende profundizar en el aprendizaje de \LaTeX , Beamer y Python utilizando un tema relacionado con el campo de las matemáticas, en este caso de la *función de distribución* de la variable aleatoria **hipergeométrica**. Se explicará en que consiste, se dará un código de programación de Python para hacer uso de este método y se comprobará que dicho programa sirve con un ejemplo.

En el subcampo matemático de la probabilidad, la distribución de variable aleatoria hipergeométrica trata de medir el número de éxitos cuando se extrae una muestra de tamaño n sin reemplazamiento.

Capítulo 2

Fundamentos teóricos

Llamamos **variable aleatoria** a toda función que asocia a cada elemento del espacio muestral Ω ¹ un número real.

Se utilizan letras mayúsculas X, Y, \dots para designar variables aleatorias, y las respectivas minúsculas (x, y, \dots) para designar valores concretos de las mismas. Antes de seguir hablando de las variables aleatorias, definamos dos conceptos que nos ayudarán más adelante:

- **Esperanza:** Dice aproximadamente mediante una media las probabilidades de una variable aleatoria.
- **Varianza:** Muestra la variabilidad de una distribución, indicando por medio de un número, si las diferentes puntuaciones de una variable están muy alejadas de la media. Cuánto mayor sea ese valor, mayor será la variabilidad, cuanto menor sea, más homogénea será a la media. Así se sabe si todos los casos son parecidos o varían mucho entre ellos.

2.1. Variables aleatorias

2.1.1. Variable aleatoria continua

Una variable aleatoria continua es aquella que puede tomar todos los valores posibles dentro de un cierto intervalo de la recta real.

Altura de los alumnos de una clase, horas de duración de una pila,...

¹Conjunto de todos los posibles resultados individuales de un experimento aleatorio. Cada posible resultado se denomina suceso

2.1.2. Variable aleatoria discreta

Una variable aleatoria discreta es aquella que sólo puede tomar valores enteros.

Número de hijos de una familia, puntuación al lanzar un dado,...

Cada variable aleatoria discreta tiene una función de probabilidad o distribución que se denota:

$$P_{\xi}(k) = P(\omega \in \Omega / \xi(\omega) = k), \forall k \in \mathbf{N}$$

Continuando con este tipo de variable aleatoria, existen diferentes tipos:

- Uniforme
- Bernouilli
- Binomial
- Binomial Negativa
- Poisson
- Geométrica
- Hipergeométrica

2.2. Variable aleatoria hipergeométrica :

Dados N , A y B números naturales, se llama **variable aleatoria hipergeométrica** de parámetros (n, A, B) a una variable que trata de medir el número de "éxitos" cuando se extrae una muestra de tamaño n sin reemplazamiento. Su función de probabilidad es:

$$P[\xi = k] = \frac{\binom{A}{k} \binom{B}{n-k}}{\binom{N}{n}}$$

$\forall k = 0, 1, \dots, n$ y se denota por $\xi \sim H(n, A, B)$. Se tiene que:

$$E[\xi] = n \cdot \frac{A}{N}$$

$$V(\xi) = n \cdot \frac{A}{N} \cdot \frac{B}{N} \cdot \frac{N-n}{N-1}$$

Capítulo 3

Procedimiento Experimental

3.1. Descripción de los experimentos

Partiendo de la función de distribución hipergeométrica:

$$P[\xi = k] = \frac{\binom{A}{x} \binom{B}{n-x}}{\binom{N}{n}}$$

$$A + B = N$$

Aplicándola a dos ejemplos en particular:

De una baraja española de 40 cartas se extraen 5 al azar. Calcúlese la probabilidad de obtener:

3.1.1. Poker de Ases

Poker de Ases : 4 cartas con el mismo valor de la extracción.

Se clasificará la baraja en dos grupos, uno con 4 ases, y otro con las 36 cartas restantes que no son as. Sea la variable ξ (número de ases obtenidos en la extracción de 5 cartas), que se distribuye según la ley hipergeométrica. Denominando:

- N_1 = Número de ases en la baraja.
- N_2 = Número de no ases en la baraja.
- $N = N_1 + N_2$ = Número de cartas en la baraja.
- n = Número de cartas extraídas de la baraja.
- x = Número de ases en las 5 extracciones.
- $n - x$ = Número de cartas extraídas no ases.

3.1.2. Color

Color : La "mano" del mismo palo.

Teniendo en cuenta que son 4 palos diferentes. Se denomina:

- N_1 = Número de cartas de un determinado palo.
- N_2 = Número de cartas de los otros tres palos.
- $N = N_1 + N_2$ = Número de cartas en la baraja.
- n = Número de cartas extraídas de la baraja.
- x = Número de cartas extraídas de un determinado palo.
- $n - x$ = Número de cartas extraídas de los otros tres palos.

3.2. Descripción del material

En esta sección se mostrará el material utilizado, es decir, los datos de la máquina:

- Linux PROA 3.2.0-40-generic 64-Ubuntu SMP Mon Mar 25 21:22:26 UTC 2013 i686 i686 i386 GNU/Linux
- Python 2.7.3
- ('default', 'Aug 1 2012 05:16:07')
- Intel(R) Core(TM)2 Duo CPU T7250 @ 2.00GHz
- GenuineIntel
- 800.000 Hz
- 2048 KB

3.3. Resultados obtenidos

Para el poker de ases, se tiene que:

$$P(\xi = x) = \frac{\binom{N_1}{x} \binom{N_2}{n-x}}{\binom{N}{n}}$$

que, en este caso, se traduce en:

$$P(\xi = 4) = \frac{\binom{4}{4} \binom{36}{1}}{\binom{40}{5}} = 0,000055$$

Para el color la probabilidad de sacarlo de un determinado palo será:

$$P(\xi = 5) = \frac{\binom{10}{5} \binom{30}{0}}{\binom{40}{5}} = 0,000383$$

donde ξ es la variable (número de cartas extraídas de un determinado palo).

Pero como la baraja tiene cuatro palos, la probabilidad de obtener color será:

$$P(color) = 4 \cdot 0,000383 = 0,001532.$$

A continuación se estimará el tiempo que ha tardado en calcularse cada ejemplo con el algoritmo que se ha creado y el implementado:

Poker Ases		
	Algoritmo	Implementado
Tiempo	0.000291109085083 s	0.00150394439697 s
Valor	5.62303193882e-5	5.47105810264e-5
Error tiempo = 0.001212835 s		
Error valor = 1.519738362e-6		

Cuadro 3.1: Poker de Ases

Color		
	Algoritmo	Implementado
Tiempo	0.000288724899292 s	0.00144481658936 s
Valor	0.000382974067185	0.000382974067185
Error tiempo = 1.15609169e-3 s		
Error valor = 0		

Cuadro 3.2: Color

A continuación, estudiaremos las probabilidades y tiempo mediante una gráfica de otra muestra dando valores enteros ($A = 40, B = 10, n = 5, x = 1, \dots, n$).

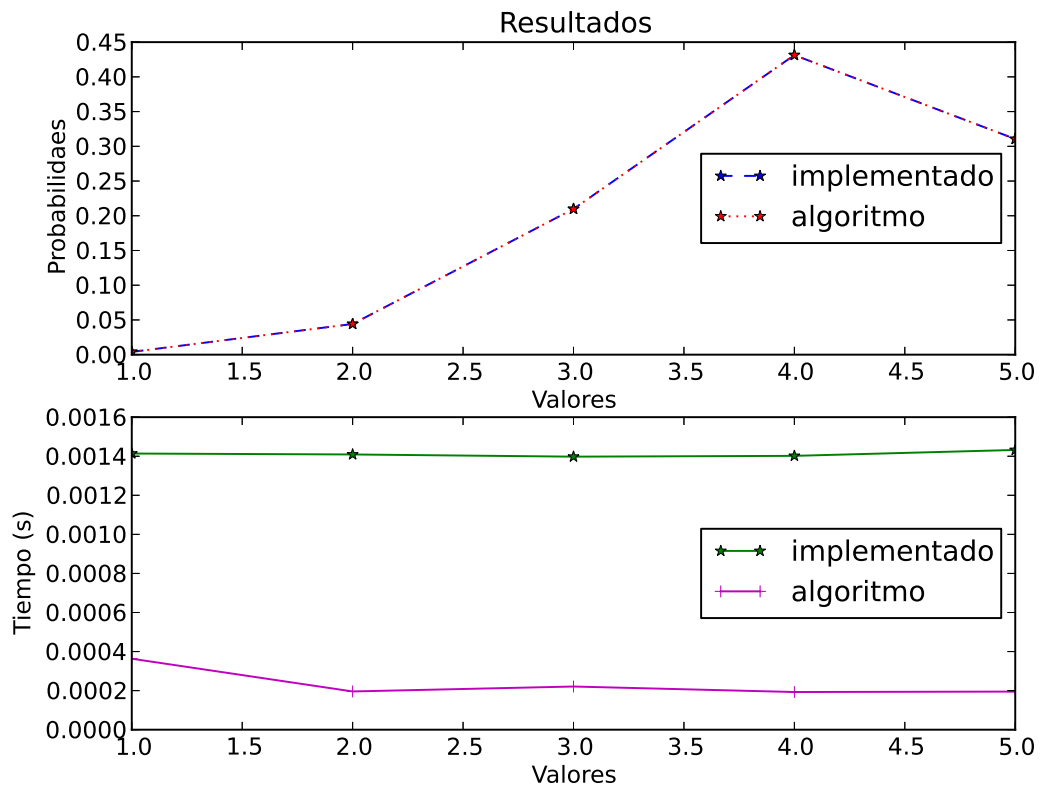


Figura 3.1: Probabilidades y tiempo

3.4. Análisis de los resultados

En cualquiera de las muestras la función implementada tarda más en compilar el programa que el algoritmo ya que tiene que cargar más paquetes, comandos, etc. mientras que el algoritmo es muy básico y bastante eficaz.

Respecto a las probabilidades, dependiendo de los valores que se tomen, puede salir hasta exactamente igual en la función implementada que la creada. En la mayoría de casos, sale aproximada.

Capítulo 4

Conclusiones

En resumen, se definió las variables aleatorias y la función, se ha creado cuatro algoritmos sobre la función de distribución hipergeométrica, dos de ellos con ejemplos, uno genérico, y uno con la función implementada. Se ha creado un módulo para obtener el tiempo empleado en cada algoritmo y los errores de cálculo entre el programa implementado y el genérico.

Este informe detalla brevemente en lo que ha consistido el proyecto. En él, se ha identificado los propósitos del mismo. Se ha recopilado información sobre la función de distribución hipergeométrica y se ha implementado este aspecto matemático en el lenguaje de programación de esta asignatura, Python. Por último, se ha adquirido más destreza en el uso de programas como \LaTeX y Beamer para la creación de textos científicos.

En este apéndice se incluirá los algoritmos Python que se han utilizado en el procedimiento experimental.

Apéndice A

Algoritmo Poker de Ases

```
#!/encoding:UTF-8
#!/usr/bin/python
```

```
import math
```

```
def fact(p):
    a=1
    if p<0:
        return 0
    elif p==0:
        return 1
    else:
        for i in range(2,p+1):
            a=a*i
        return (a)
```

```
N1=int(raw_input("Introduzca el número de ases de las baraja (N1 > x):"))
N2=int(raw_input("Introduzca el número de no ases en la baraja (N2 > n-x):"))
n=int(raw_input("Introduzca el número de cartas extraídas de la baraja (n < N1+N2):"))
x=int(raw_input("Introduzca el número de ases en las 5 extracciones :"))
N = N1 + N2
```

```
comb1 = float(fact(N1)/(fact(x)*fact(N1-x)))
comb2 = float(fact(N2)/(fact(n-x)*fact(N2-n+x)))
comb3 = float(fact(N)/(fact(n)*fact(N-n)))
```

```
probabilidad = (comb1+comb2)/comb3
print "La probabilidad es : ", probabilidad
```

Apéndice B

Algoritmo Color

```
#!/encoding:UTF-8
#!/usr/bin/python
```

```
import math
```

```
def fact(p):
    a=1
    if p<0:
        return 0
    elif p==0:
        return 1
    else:
        for i in range(2,p+1):
            a=a*i
    return (a)
```

```
N1=int(raw_input("Introduzca el número de cartas de determinado palo (N1 > x):"))
N2=int(raw_input("Introduzca el número de cartas de los otros tres palos (N2 > n)"))
n=int(raw_input("Introduzca el número de cartas extraídas de la baraja (n < N1+N2)"))
x=int(raw_input("Introduzca el número de cartas extraídas de un determinado palo (x < N1)"))
N=N1+N2
```

```
comb1=float(fact(N1)/(fact(x)*fact(N1-x)))
comb2=float(fact(N2)/(fact(n-x)*fact(N2-n+x)))
comb3=float(fact(N)/(fact(n)*fact(N-n)))
```

```
probabilidad=(comb1*comb2)/comb3
print"La probabilidad es : ",probabilidad
```

Apéndice C

Algoritmo F.D Hipergeométrica

```
#!/encoding:UTF-8
#!/usr/bin/python

import math
import time

start1 = time.time()
def fact(p):
    a=1
    if p<0:
        return 0
    elif p==0:
        return 1
    else:
        for i in range(2,p+1):
            a=a*i
        return (a)

finish1 = time.time() - start1

A=int(raw_input("Introduzca el A (A>x): "))
B=int(raw_input("Introduzca el B (B>n-x): "))
n=int(raw_input("Introduzca el n: (n<A+B)"))
x=int(raw_input("Introduzca la x: "))

N=A+B
```

```
start2 = time.time()
comb1=float(fact(A)/(fact(x)*fact(A-x)))
comb2=float(fact(B)/(fact(n-x)*fact(B-n+x)))
comb3=float(fact(N)/(fact(n)*fact(N-n)))

probabilidad=(comb1*comb2)/comb3

print"La probabilidad es :",probabilidad

finish2 = time.time() - start2

finish = finish1 + finish2

print finish
```

Apéndice D

F.D Hipergeométrica Implementada

```
from scipy.stats import hypergeom
import numpy as np
import time

M=int(raw_input("Introduzca el valor de M: "))
n=int(raw_input("Introduzca el valor de n: "))
N=int(raw_input("Introduzca el valor de N: "))
x=int(raw_input("Introduzca el valor de x: "))
start = time.time()

rv = hypergeom(M, n, N)
pmf_cartas = rv.pmf(x)

print pmf_cartas

finish=time.time()-start

print finish
```


Bibliografía

- [1] Francisco Javier Martín-Pliego, José María Montero Lorezno, Luis Ruiz-Maya Pérez. Problemas de Probabilidad
- [2] Guido Rossum. Python reference manual. Technical report, Amsterdam, The Netherlands, 1995.
- [3] Juan José Salazar y Marta López Yurda. Ejercicios Resueltos de probabilidad.
- [4] ACM LaTeX Style. *[http : //www.acm.org/publications/latex_style/](http://www.acm.org/publications/latex_style/)*.