

Distribuciones continuas (e.j. gaussiana)

Recordemos que la distribución de probabilidad normal o gaussiana está dada por:

$$P(X) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)^2\right\}$$

donde:

- μ : media de la distribución
- σ : desviación estandar de la distribución

Ejemplo: considerando una variable aleatoria que sigue una distribución normal con media $\mu=4$ y desviación estándar $\sigma=0.3$, la probabilidad de que dicha variable tome el valor de 0.2 está dada por:

$$P(0.2) = \frac{1}{0.3 \sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{0.2-4}{0.3}\right)^2\right\}$$

Lo cual en Python se traduce en:

```
from scipy.stats import norm
```

```
norm(mu, sigma).pdf(X)
```

Así también, la distribución de probabilidad acumulada correspondiente está dada por:

$$C(X) = \int_{-\infty}^X P(x)dx = \int_{-\infty}^X P(X)dX$$

teniendo en cuenta que $\text{Dom}(X) = (-\infty, \infty)$.

Ejemplo: considerando una variable aleatoria que sigue una distribución normal con media $\mu=4$ y desviación estándar $\sigma=0.3$, la probabilidad de que dicha variable tome el valor de 0.2 o menos está dada por:

$$C(0.2) = \int_{-\infty}^{0.2} P(x)dx = \int_{-\infty}^{0.2} \frac{1}{0.3 \sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{X-4}{0.3}\right)^2\right\} dX$$

La cual se calcula en Python como:

```
from scipy.stats import norm
```

```
norm(mu, sigma).cdf(X)
```

Es importante recordar que la función de probabilidad acumulada de la distribución gaussiana no se puede calcular de forma exacta, ya que la integral anterior no tiene una expresión cerrada conocida. Es decir, los métodos de integración conocidos no funcionan para resolver esta integral.

Ejercicios (bloque 1)

Considerando una variable aleatoria que sigue una distribución normal con media $\mu = 4$ y desviación estándar $\sigma = 0.1$, calcula las siguientes probabilidades (usando Python):

1. $P(X = 4)$
2. $P(X = -10)$
3. $P(X = 10)$
4. $P(X \leq 4)$
5. $P(X \geq 4)$

▼ De la binomial a la gaussiana

En la clase 8 vimos como generar secuencias aleatorias de experimentos binomiales ([aquí](#)), donde cada experimento era lanzar un cierto número de monedas.

¿Qué sucede si el número consecutivo de monedas que lanzamos en cada experimento (trial) es muy largo?

La función `generate_binomial_trials()` nos muestra lo que sucede si graficamos los resultados de muchos experimentos de lanzar 100 monedas en cada intento, con `pyplot`:

```
import numpy as np
from numpy.random import binomial
import matplotlib.pyplot as plt

def generate_binomial_trials(trials=1000, coin_toss=100):
    '''
    el resultado de esta funcion es generar un conjuntos
    de experimentos binomiales (trials) y de cada uno obtener
    las cantidades de exitos en cada secuencia (e.j. lanzar monedas).

    * trial: es una secuencia de <coin_toss> lanzamientos de moneda

    * coin_toss: es el numero de monedas lanzadas en cada trial
    '''
    arr = []
    for _ in range(trials):
        arr.append(binomial(coin_toss, 0.5))
    values, dist = np.unique(arr, return_counts=True)

    return values, dist

values, dist = generate_binomial_trials(100000)
plt.bar(values, dist)
```

¿Se parece a algo conocido? Tal vez una distribución normal se ajuste, para verificarlo haremos el siguiente ejercicio:

▼ Ejercicios (bloque 2)

1. Con los resultados anteriores guardados en `values`, `dist` ajusta, usando el método de estimación paramétrica, una distribución gaussiana donde la media y desviación estándar correspondan a lo calculado a partir de los datos, de la misma manera que se hizo en el notebook de la clase 9 ([aquí](#)).

escribe tu código aquí:

Si todo salió bien, habrás notado que en efecto una distribución normal se ajusta perfectamente a los datos. Esto se conoce como el **teorema del límite central**, el cual establece que en muchas situaciones conocidas, cuando variables aleatorias independientes se combinan, su total tiende a seguir una distribución normal cuando el número de variables que se combinan es muy grande $n \rightarrow \infty$.

