Practise 1 - Ej1

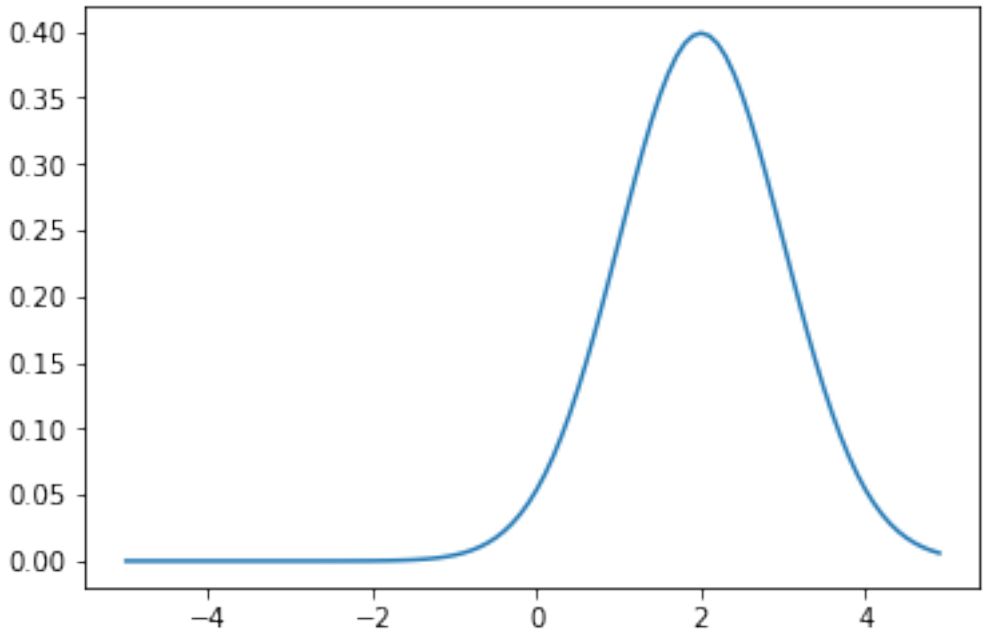
October 15, 2018

```
In [1]: import numpy as np
        import math
        import matplotlib.pyplot as plt

        %matplotlib inline
```

1.- Escribe un programa en matlab que dibuje una gaussiana, para valores ţ=2 y = 1, en el intervalo [-5, 5] y con incrementos de 0.1. Implementa para ello tu propia f(x).

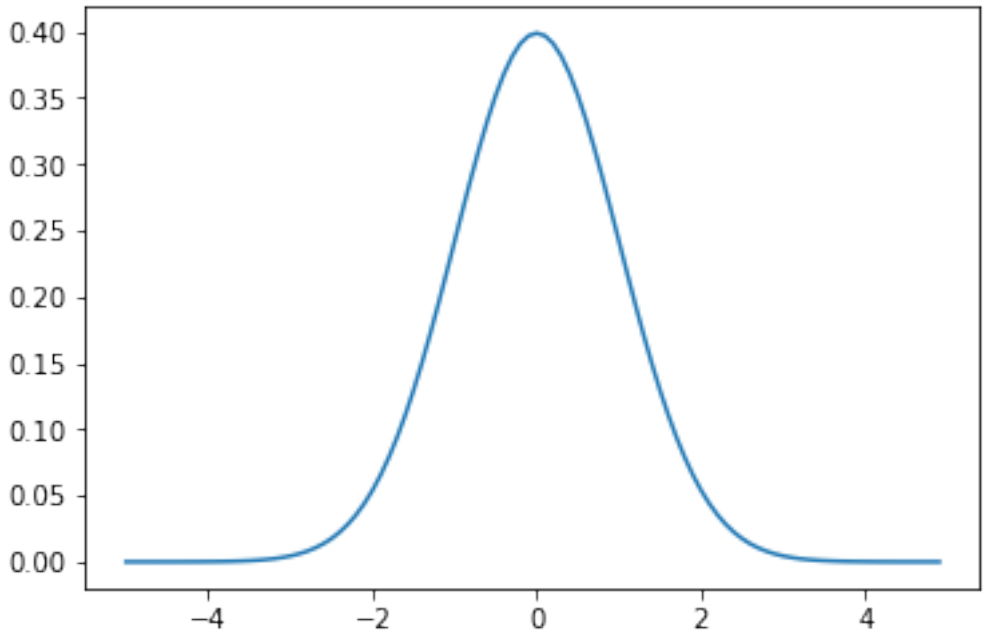
```
In [146]: def f(x, mu = 2, sigma = 1):
              return (1/math.sqrt(2*math.pi*(sigma**2)))*math.exp(-((x-mu)**2)/(2*(sigma**2)))
```

```
In [147]: x = np.arange(-5, 5, 0.1)
          plt.plot(x, list(map(f, x)));
```



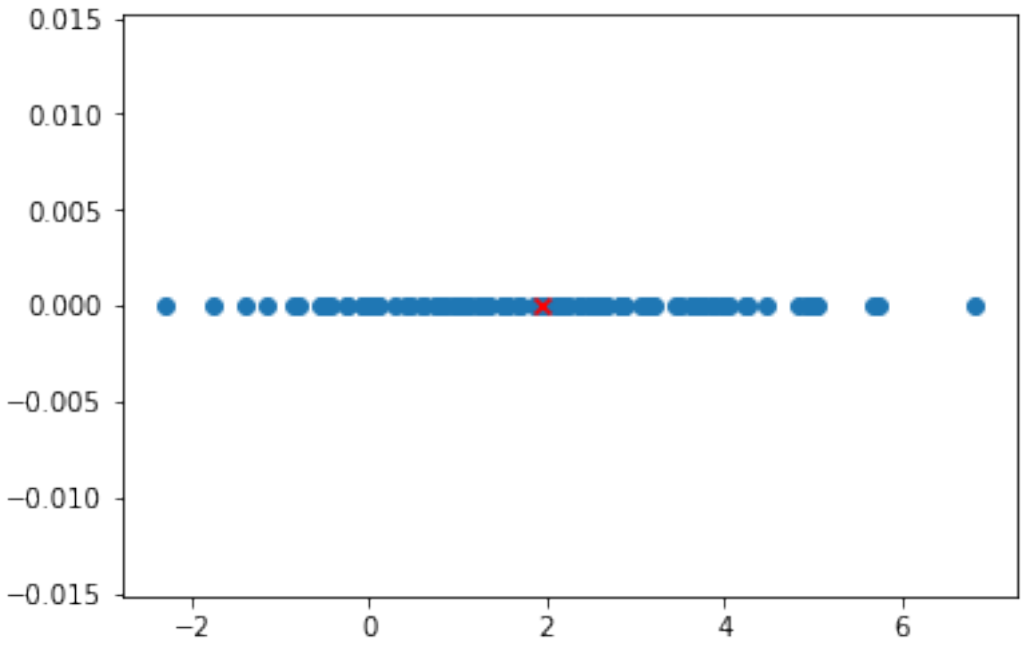
2.- Implementa una función llamada evaluate_gaussian que tome como parámetros la media y la desviación de una distribución normal y el valores de x en la que se evalúa. Desde la consola, plotea f(x) para con incrementos de 0.1, usando la media y desviaciones típicas del apartado anterior.

```
In [148]: def evaluate_gaussian(x, mu, sigma):
              plt.plot(x, list(map(lambda x:f(x, mu, sigma), x)));

In [159]: x = np.arange(-5, 5, 0.1)
          evaluate_gaussian(x, 0, 1)
```



3.- Muestrear consiste en obtener un valor aleatorio obedeciendo una distribución de probabilidad. Empleando la función randn, obtiene y dibuja muestras sobre el eje x de una distribución normal de parámetros ţ=2 y = 2.

```
In [3]: def randn(size, mu, sigma):
            x = np.random.randn(size)
            x = x*sigma + mu
            return x

In [4]: x = randn(100, 2, 2)
        plt.scatter(x, np.zeros(100));
        plt.scatter(np.mean(x), np.zeros(1), marker='x', c='r')

Out[4]: <matplotlib.collections.PathCollection at 0x7f1b0b283748>
```



Razona las siguientes cuestiones:

£En torno a qué valores se concentran?£Por qué?

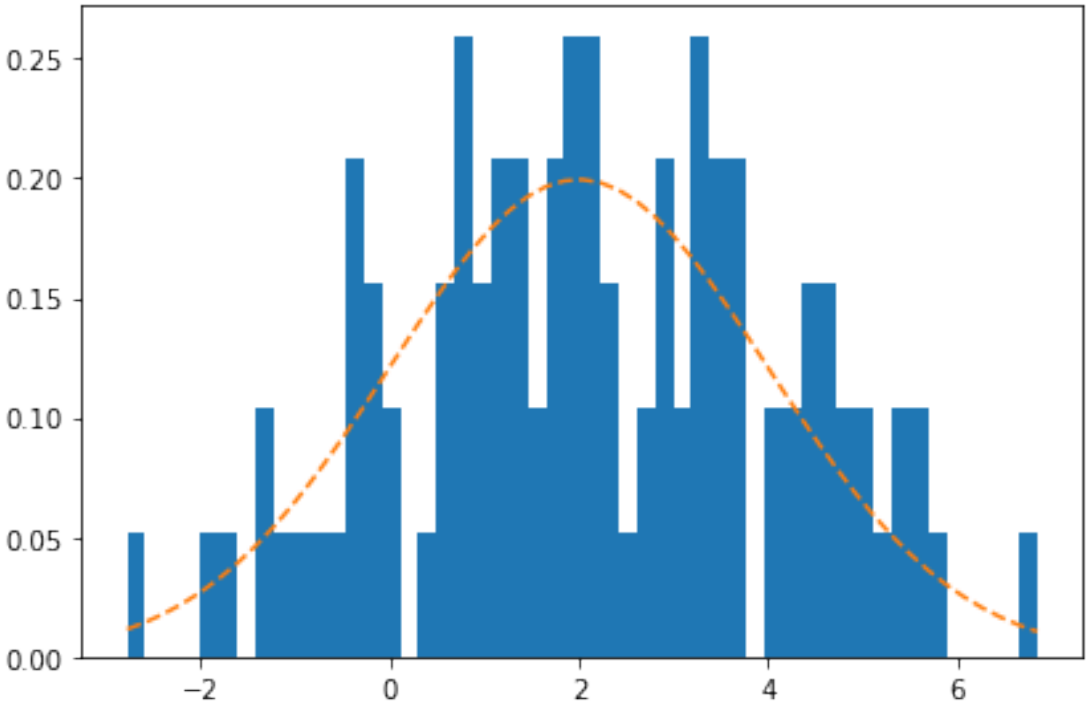
Se concentran entorno a la media, ya que es más probable que aparezcan allí.

£Por qué hay cada vez menos muestras cuanto más nos alejamos del origen de coordenadas?

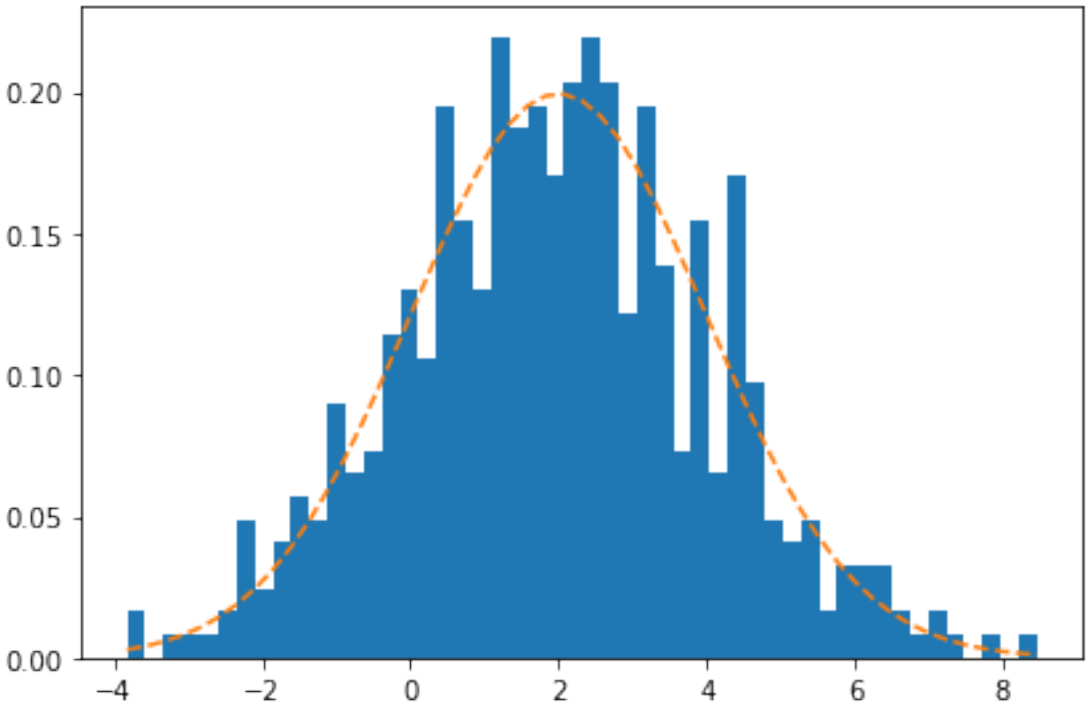
Porque es menos probable que tomen estos valores

4.- Dibuja el histograma para distintos valores de muestras (100,500,1000), empleando la media y la desviación típica del apartado anterior.

```
In [170]: def plot_randn(size, mu, sigma):
              x = randn(size, mu, sigma)
              fig, ax = plt.subplots()
              num_bins = 50
              # the histogram of the data
              n, bins, patches = ax.hist(x, num_bins, density=1)

              # add a 'best fit' line
              y = ((1 / (np.sqrt(2 * np.pi) * sigma)) *
                   np.exp(-0.5 * (1 / sigma * (bins - mu))**2))
              ax.plot(bins, y, '--')

              # Tweak spacing to prevent clipping of ylabel
              fig.tight_layout()
              plt.show()

In [171]: plot_randn(100, 2, 2)
```




```
In [181]: plot_randn(500, 2, 2)
```



```
In [177]: plot_randn(1000, 2, 2)
```