Esercizio1

July 19, 2018

1 Generatore di numeri casuali secondo una Breit-Wigner

Si costruisca un generatore di numeri casuali distribuiti secondo una Breit-Wigner e si discuta dei possibili test di casualità per i numeri prodotti.

Per costruire il generatore si è utilizzata la tecnica della funzione inversa. Si generano quindi dei numeri casuali distribuiti uniformemente in $[0, 1]$.

Un primo e più semplice test eseguibile su questi numeri è una verifica delle loro media ($\mu_{attesa} = \frac{1}{2}$) e varianza ($V_{attesa} = \frac{1}{12}$).

```
In [1]: import numpy as np
        import scipy.integrate as integrate
        import matplotlib.pyplot as plt
        import math
        from scipy import stats

        np.random.seed()
        M_0 = 776.
        Gamma = 70.
        Numero_Eventi = int( 1e7 )

        def Breit_Wigner (x):    # Definisco la pdf di Breit-Wigner
                return Gamma/(2.*np.pi) * 1. / (Gamma**2/4. + (x - M_0)**2)


        ### Genero i numeri secondo una distribuzione uniforme
        Eventi = np.random.uniform(0.,1.,Numero_Eventi)

        ### Eseguo il primo test
        mean = np.mean(Eventi)     # Calcolo la media
        variance = np.var(Eventi, ddof=1) # Calcolo la varianza
        errmean = math.sqrt(variance/Numero_Eventi) # Calcolo l'errore sulla media
        errvar = variance * math.sqrt (2 / (Numero_Eventi - 1)) # Calcolo errore sulla varianz

        # Stampo i risultati
        print("\nTest sulla distribuzione uniforme")
        print("Media attesa: ", 0.5)
        print("Media: \t", mean, "+/-", errmean)
```

```
print("Varianza attesa: ", 1./12.)
print("Varianza: \t", variance, "+/-", errvar)
```

```
Test sulla distribuzione uniforme
Media attesa:  0.5
Media:          0.4999418976816775 +/- 9.130871579375001e-05
Varianza attesa:  0.08333333333333333
Varianza:          0.08337281579903813 +/- 3.728545858471652e-05
```

Sia la distribuzione di Breit-Wigner definita come:

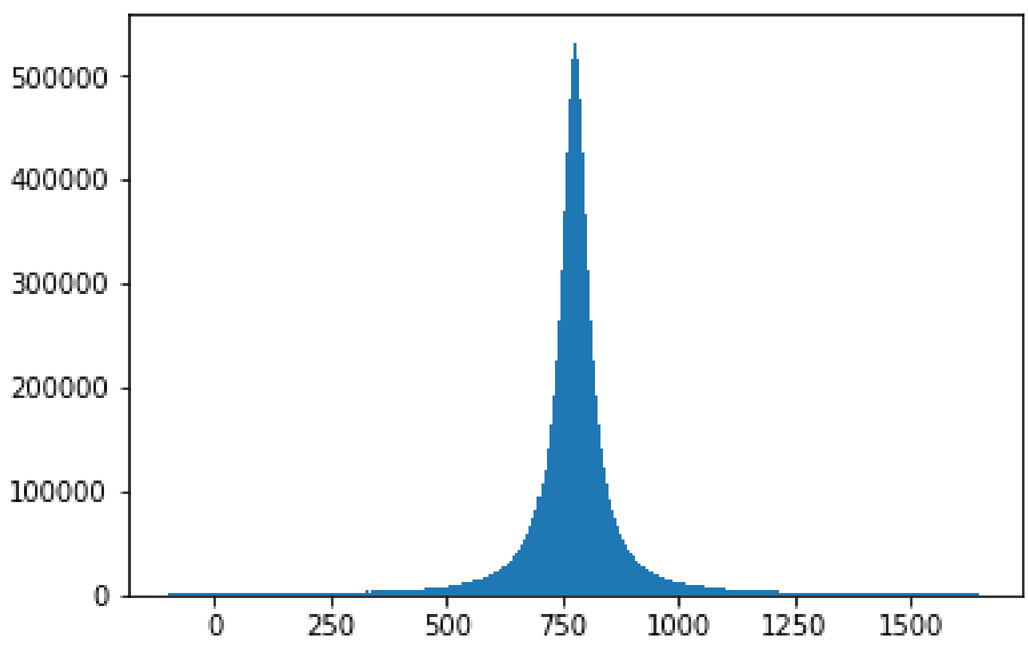
$$f(x, \Gamma, M_0) = \frac{1}{\pi} \frac{\Gamma/2}{\Gamma^2/4 + (x - M_0)^2}$$

è possibile campionare numeri random distribuiti come una Breit-Wigner invertendo la funzione cumulativa F a valori in $[0, 1]$:

$$x = M_0 + \frac{\Gamma}{2} \tan\left(\pi\left(F - \frac{1}{2}\right)\right)$$

```
In [2]: # Sfrutto la possibilità di eseguire operazioni su tutti gli
        # elementi di un array di python
        Eventi = M_0 + (Gamma/2.) * np.tan(np.pi * (Eventi - 0.5))

        # Disegno l'istogramma dei dati
        bins = np.linspace(M_0 - 25*Gamma/2, M_0 + 25*Gamma/2, 300) # creo i bin
        Histogram = plt.hist(Eventi, bins)
        conteggi, bins, bars = Histogram
```



Il primo test eseguibile per verificare l'effettiva distribuzione dei numeri secondo una Breit-Wigner è quello del χ^2. Calcolo il valore:

$$\chi^2 = \sum_i^k \frac{(O_i - NP_i)^2}{NP_i}$$

Con N numero di eventi misurati, $P_i = \int_{bin_i} f(x)dx$ frazione di eventi attesa nel bin i-esimo e O_i eventi osservati nel bin i-esimo.

Il risultato può essere confrontato con i valori della distribuzione χ^2 con gradi di libertà pari al numero di bin.

```
In [3]: prob = [] # Lista in cui inserirò i valori P_i
        for i in range(len(bins) - 1):   # calcolo i P_i
                x = (bins[i] + bins[i+1]) / 2.
                risultato, incertezza = integrate.quad(Breit_Wigner,  bins[i], bins[i+1],)
                prob.append(risultato)

        prob = np.array(prob)      # Trasformo la lista in array

        # Calcolo gli scarti quadratici medi per ogni bin
        Chi2_array = (conteggi - sum(conteggi) * prob)**2 / sum(conteggi) * prob

        # Sommo gli scarti quadratici e ottengo il Chi2
        Chi2 = sum(Chi2_array)
        Chi2_ridotto = Chi2/(len(bins)-1)  # Calcolo il Chi2 ridotto

        # Stampo i risultati
        print("\nTest sulla distribuzione Breit-Wigner")
        print("- Chi Quadro -")
        print("Chi2: ", Chi2,"\tChi2 ridotto: ", Chi2_ridotto)
        print("P-value: ", 1 - stats.chi2.cdf(Chi2_ridotto, len(bins-1)))
```

```
Test sulla distribuzione Breit-Wigner
- Chi Quadro -
Chi2:  6.776054952795765         Chi2 ridotto:  0.022662391146474132
P-value:  1.0
```

Un secondo test per verificare che la distribuzione sia quella prescelta è il test di Kolmogorov-Smirnov. In questo test confronto la Cumulative Distribution Function $F(x)$ della distribuzione attesa con la cumulativa degli N eventi ottenuti $S_N(x)$. Si calcola il valore:

$$D = \sup |S_N(x) - F(x)|$$

Si verifica quindi che questa sia inferiore al valore critico di rigetto per un certo livello di confidenza α. Per far ciò si è usufruito della funzione *kstest()* della libreria *stats* di Python.

```
In [4]: print("\n- Kolmogorov Smirnov -")

        D, p = stats.kstest(Eventi, "cauchy", args=(M_0, Gamma/2.)) # Eseguo il test
        D_critica = 1.36/math.sqrt(Numero_Eventi)          # Calcolo il valore critico

        # Stampo i risultati
        print("Valore D critico per alpha = 5%","\tValore D ottenuto dal test")
        print(D_critica,"\t\t\t", D)


        plt.show()
```

```
- Kolmogorov Smirnov -
Valore D critico per alpha = 5%         Valore D ottenuto dal test
0.0004300697617828996                           0.000335682620303146
```

I risultati confermano quindi che i numeri estratti sono distribuiti secondo una Breit-Wigner con $\Gamma = 70.$ e $M_0 = 776.$