Accuratezza del test

70%

Accuratezza del test

71%

Accuratezza del test

99%

Accuratezza del test

100%

Il reverendo Bayes è tuo amico

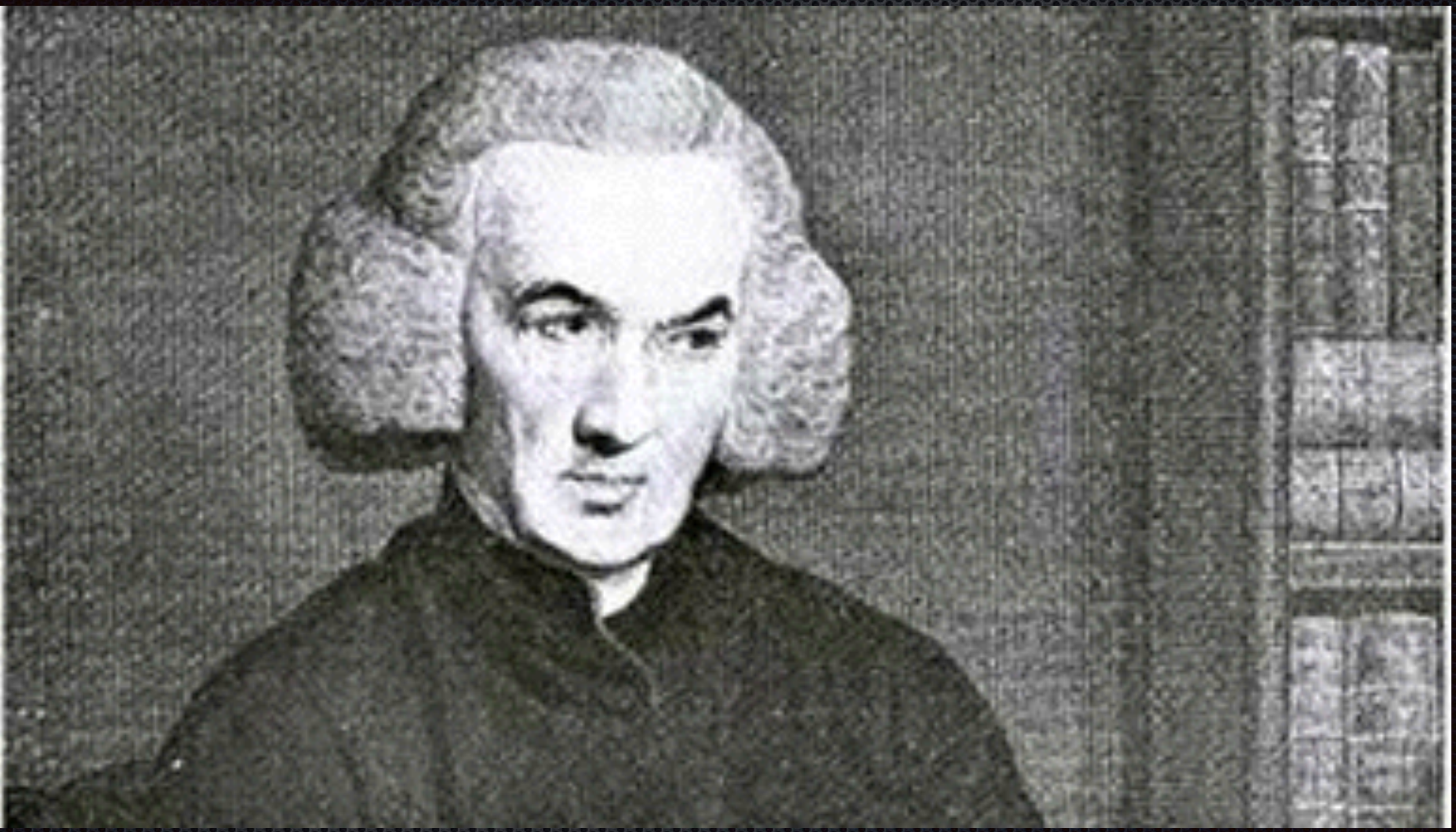
Emmanuele Somma

Londra, 1702 – Royal Tunbridge Wells, 17 aprile 1761

- *Divine Benevolence, or an Attempt to Prove That the Principal End of the Divine Providence and Government is the Happiness of His Creatures* (1731)
- *An Introduction to the Doctrine of Fluxions, and a Defence of the Mathematicians Against the Objections of the Author of the Analyst* (1736)

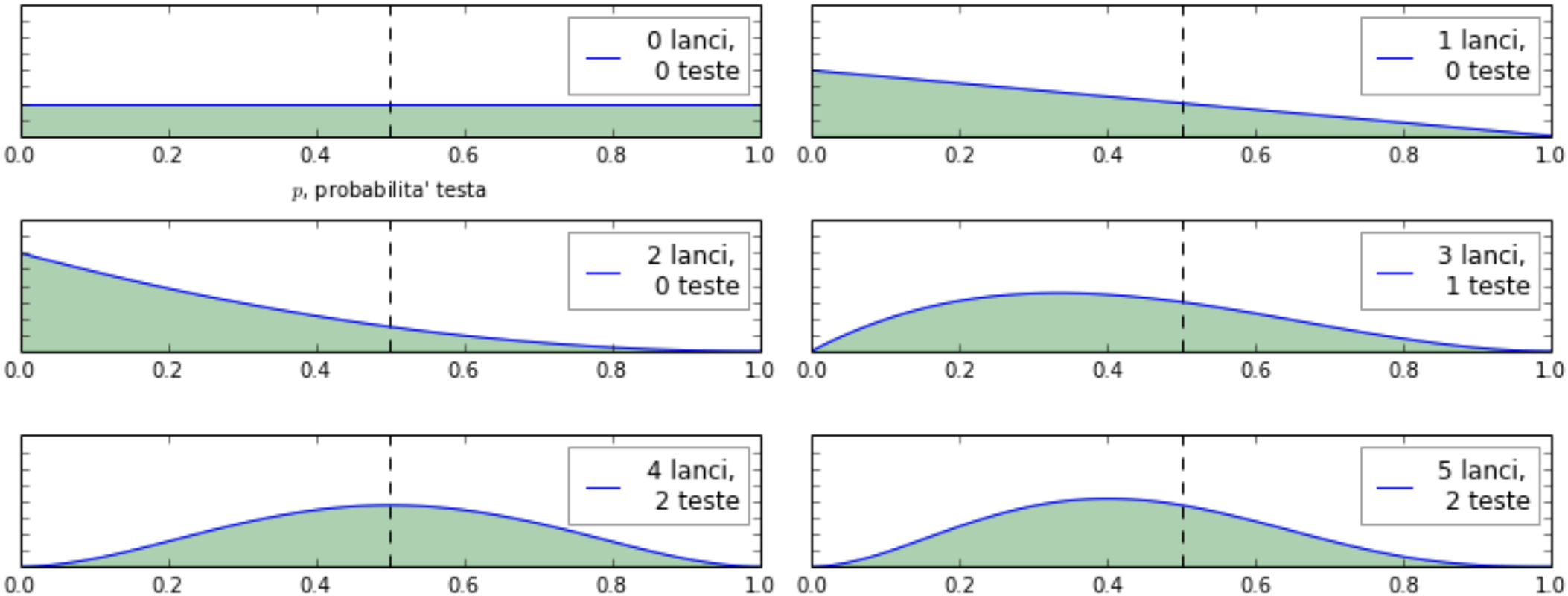
T. Bayes.

Richard Price (23 February 1723 – 19 April 1791)



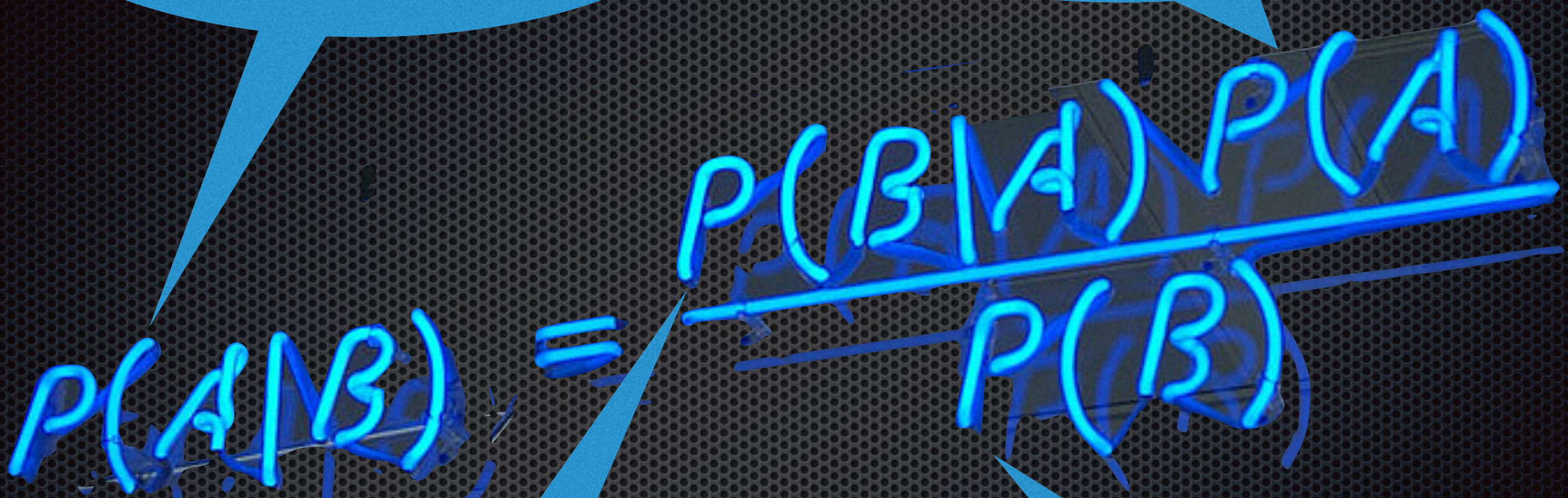

UNITED STATES OF AMERICA
E PLURIBUS
UNUM
W
E
1 OZ. FINE SILVER–ONE DOLLAR

probabilita' a posteriori bayesiane



Non ci sono bug nel programma se tutti i test passano?

I programmi sono corretti?



I test passano se il programma è corretto?

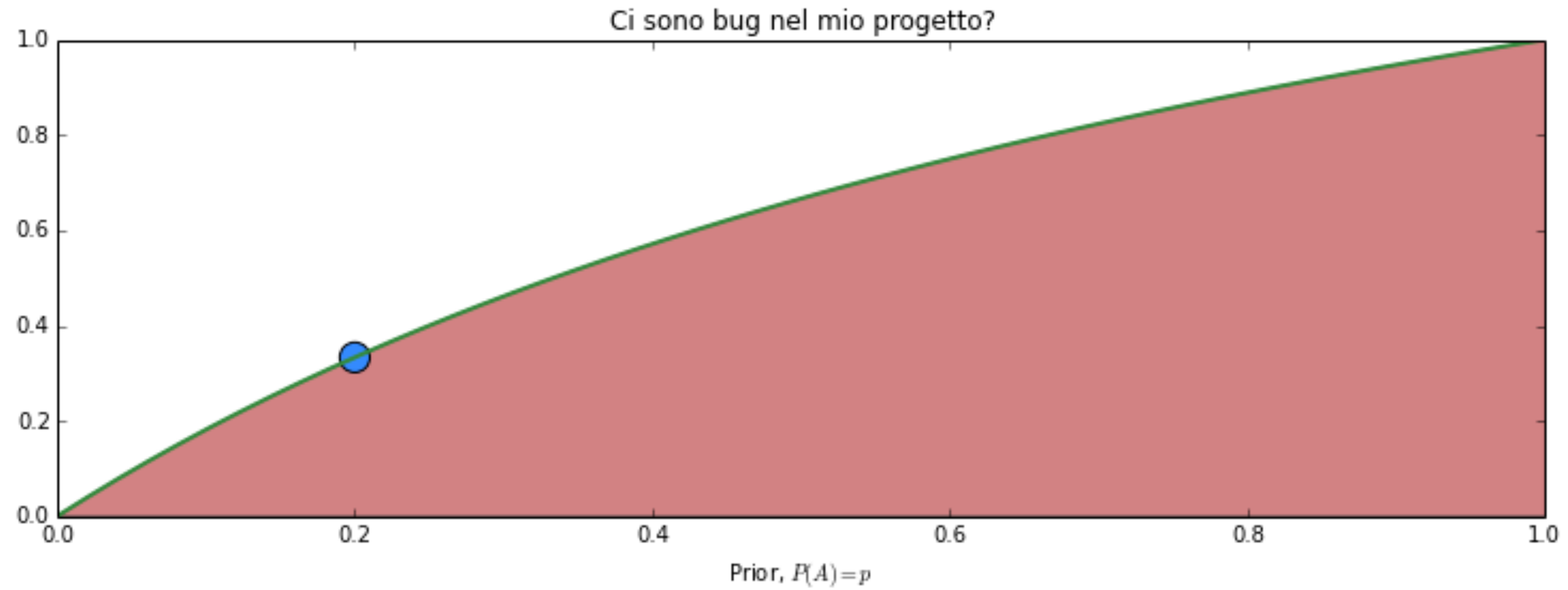
I test sono corretti?

http://commons.wikimedia.org/wiki/File:Bayes'_Theorem_MMB_01.jpg

$$\frac{2p}{1+p}$$

1

p



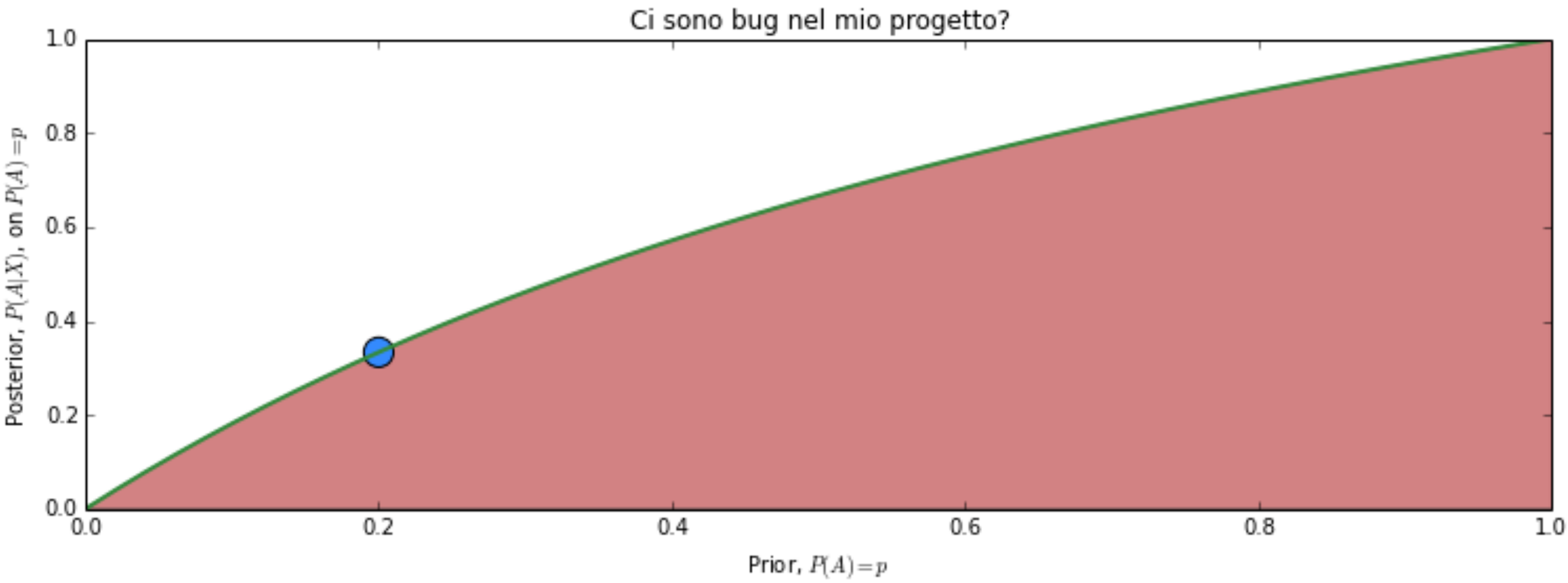
$$=P(B|A)P(A)+P(B|\sim A)P(\sim A)$$

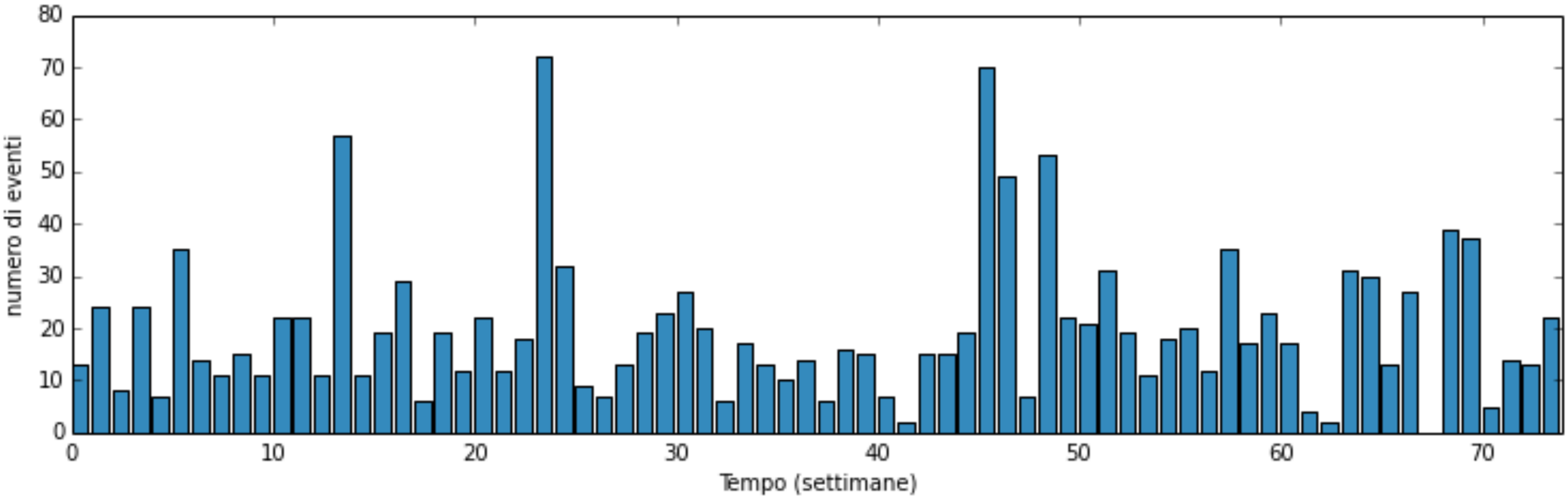
$$=P(X|A)p+P(X|\sim A)(1-p)$$

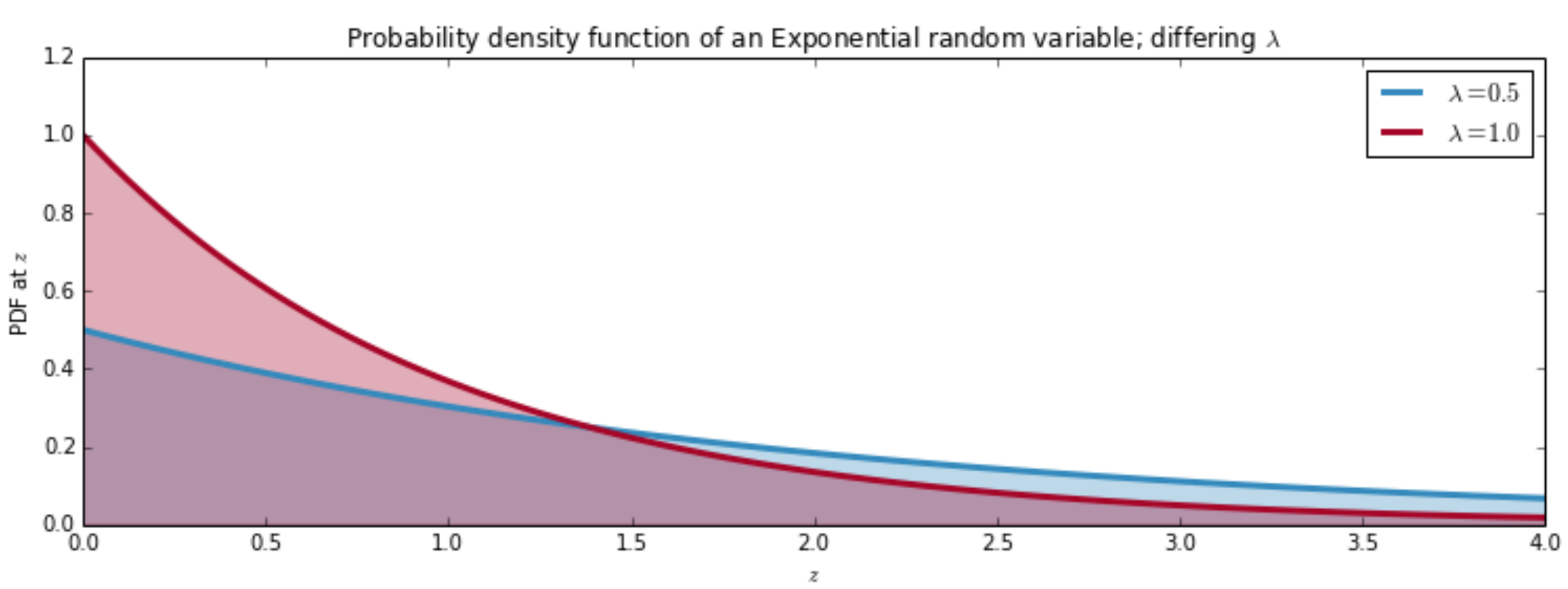
$$=p+0{,}5(1-p)$$

http://commons.wikimedia.org/wiki/File:Bayes'_Theorem_MMB_01.jpg

```
%matplotlib inline
from IPython.core.pylabtools import figsize
import numpy as np
from matplotlib import pyplot as plt
figsize(12.5, 4)
p = np.linspace(0, 1, 50)
plt.plot(p, 2 * p / (1 + p), color="#348A3D", lw=2)
plt.fill_between(p, 2*p/(1+p), alpha=.5, facecolor=["#A60608"])
plt.scatter(0.2, 2 * (0.2) / 1.2, s=200, c="#348AFD")
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel("Prior, $P(A) = p$")
plt.ylabel("Posterior, $P(A|X)$, su $P(A) = p$")
plt.title("Ci sono bug nel mio progetto?")
```




numero di eventi
Tempo (settimane)
0
10
20
30
40
50
60
70
80


Probability density function of an Exponential random variable; differing λ
$\lambda = 0.5$
$\lambda = 1.0$
PDF at z
z
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Probability mass function of a Poisson random variable; differing λ values

probability of k

$\lambda = 1.5$

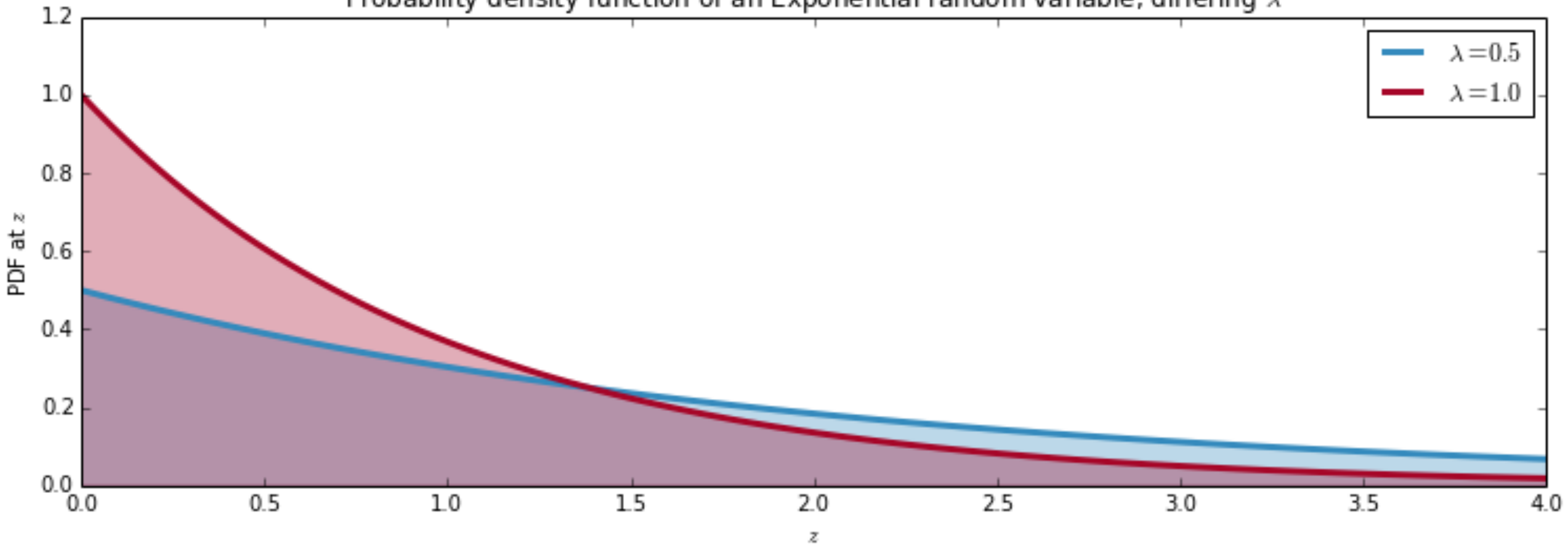
$\lambda = 4.2$

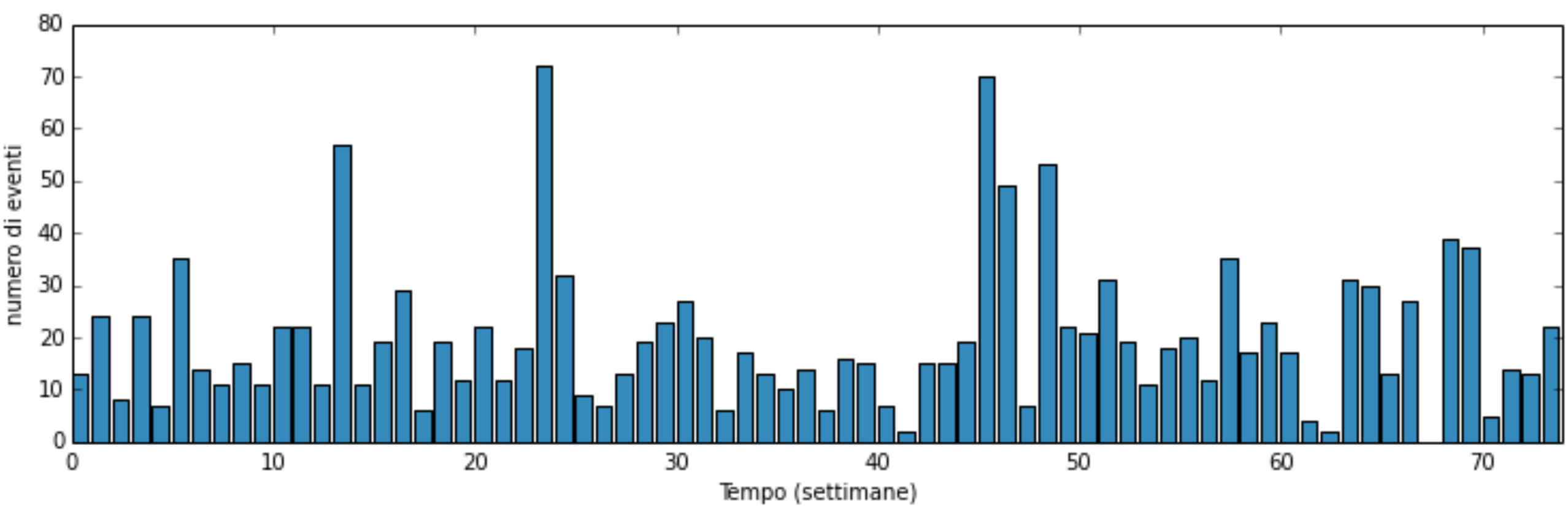
k

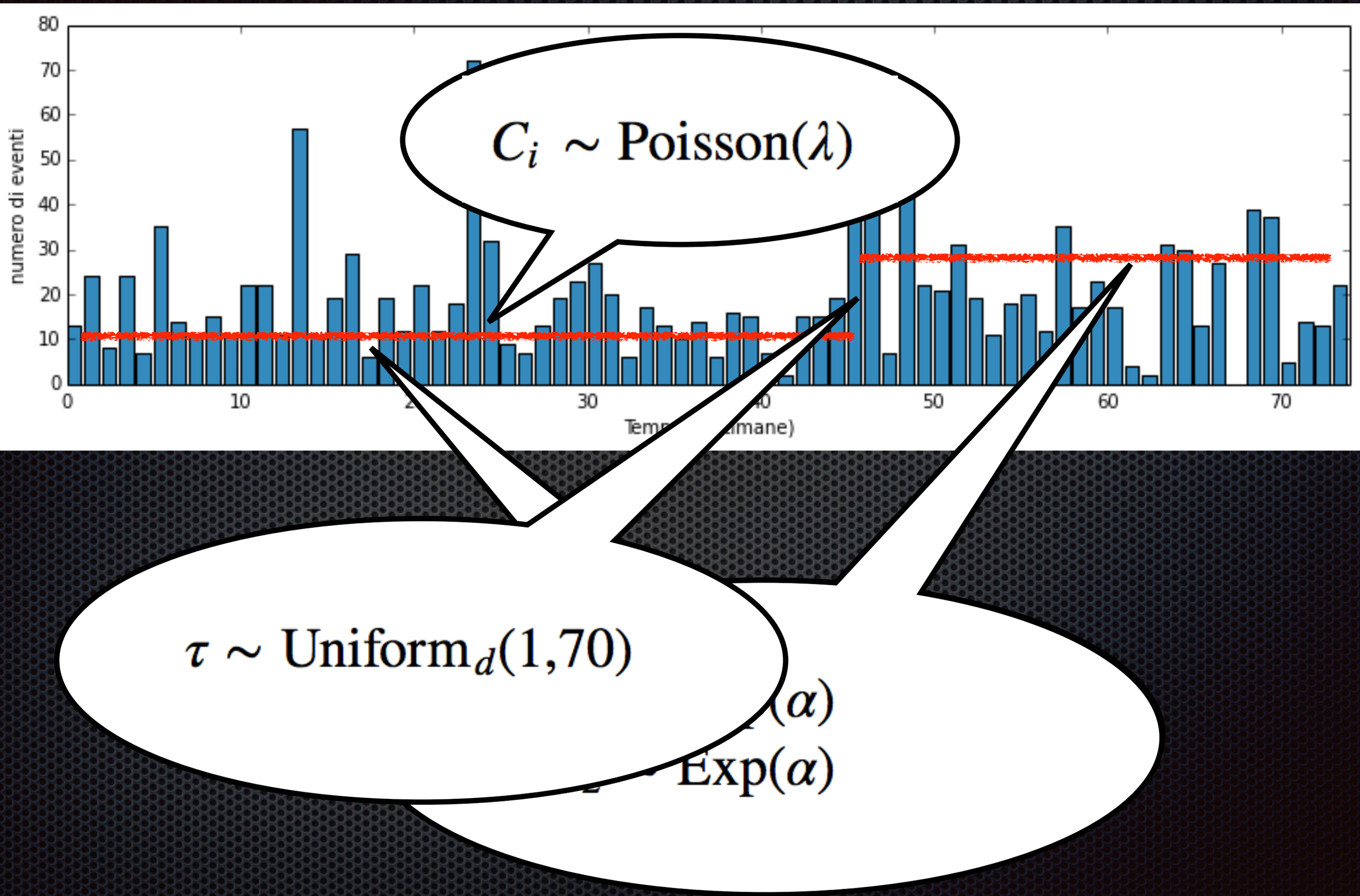
numero di eventi

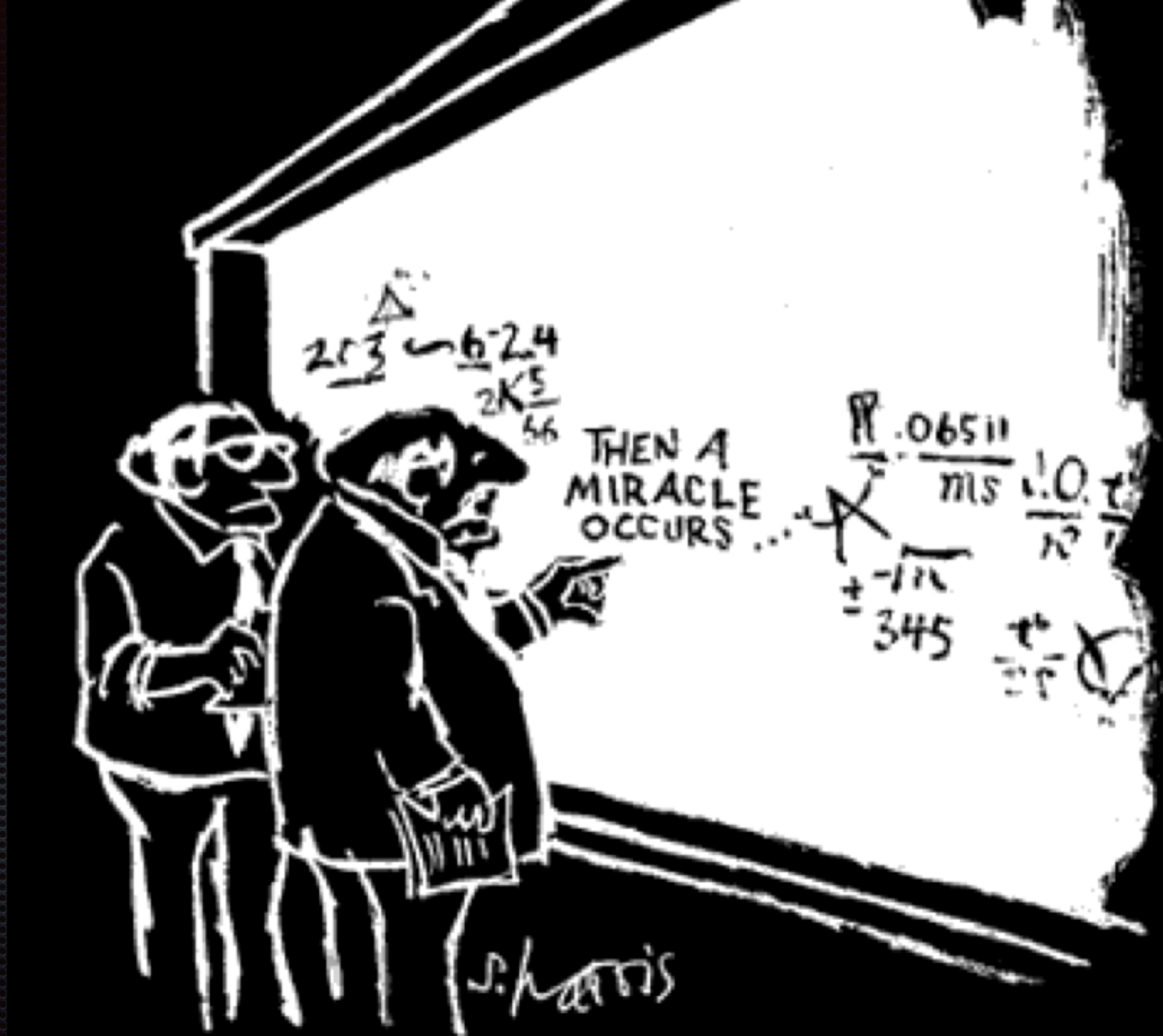
Tempo (settimane)

z


Probability density function of an Exponential random variable; differing λ
$\lambda = 0.5$
$\lambda = 1.0$
PDF at z
z
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


numero di eventi
Tempo (settimane)
0
10
20
30
40
50
60
70
80


numero di eventi
0
10
20
30
40
50
60
70
80
0
10
30
50
60
70
$C_i \sim \text{Poisson}(\lambda)$
$\tau \sim \text{Uniform}_d(1,70)$
(α)
$\text{Exp}(\alpha)$


THEN A MIRACLE OCCURS ...
S. Harris

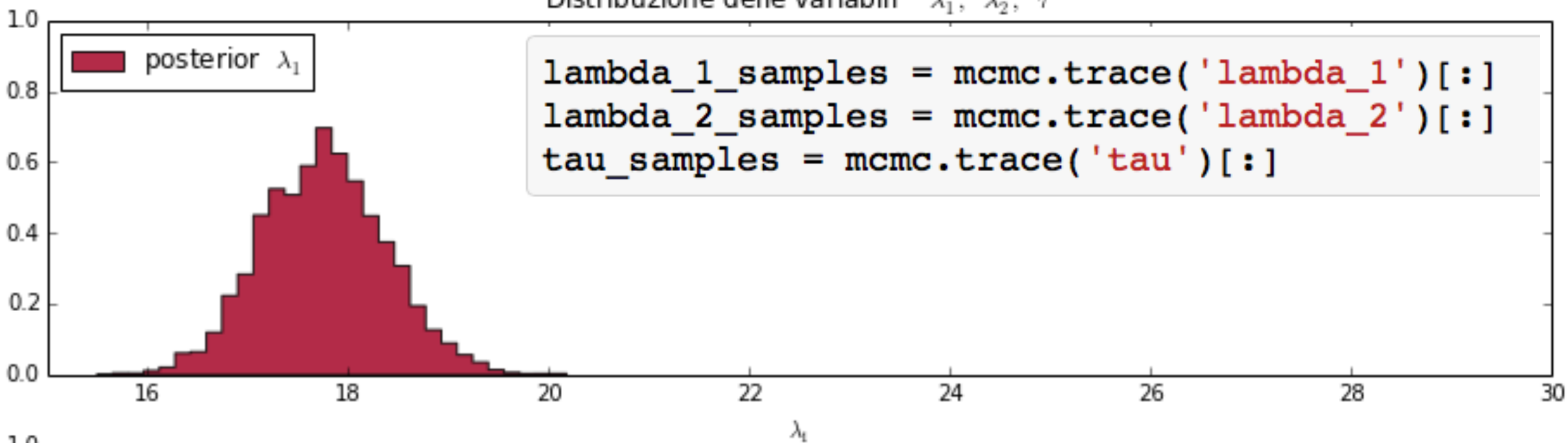
```
import pymc as pm
import numpy as np

count_data = np.loadtxt("data/txtdata.csv")
n_count_data = len(count_data)

alpha = 1.0 / count_data.mean()

lambda_1 = pm.Exponential("lambda_1", alpha)
lambda_2 = pm.Exponential("lambda_2", alpha)

tau = pm.DiscreteUniform("tau", lower=0, upper=n_count_data)
```

```
@pm.deterministic
def lambda_(tau=tau, lambda_1=lambda_1, lambda_2=lambda_2):
    out = np.zeros(n_count_data)
    out[:tau] = lambda_1
    out[tau:] = lambda_2
    return out
```

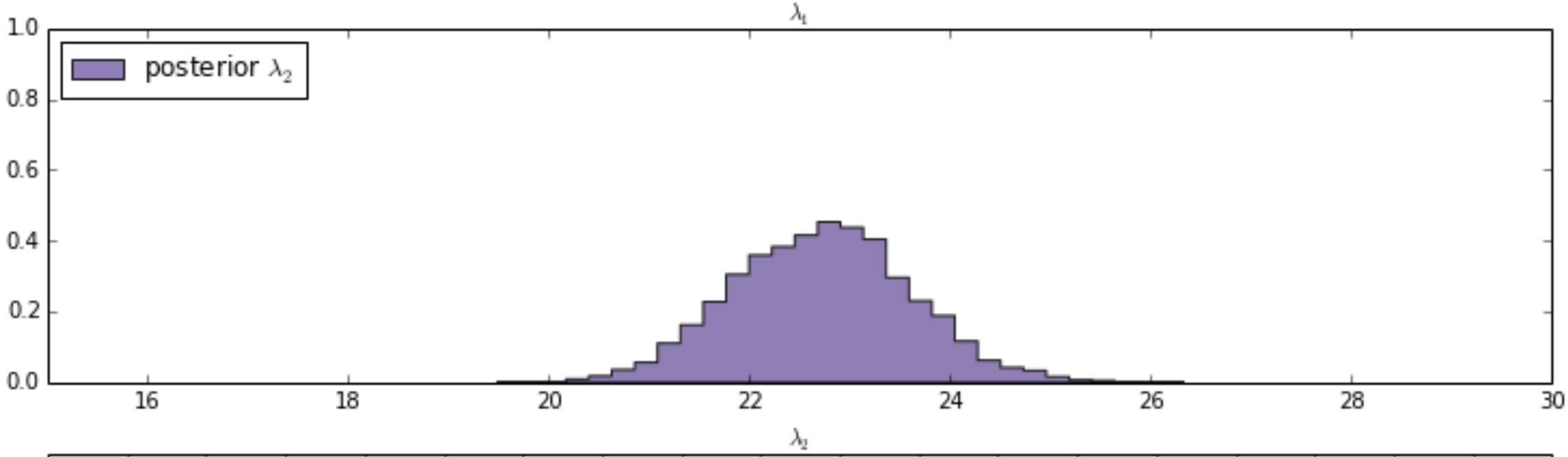
```
observation = pm.Poisson("obs", lambda_,value=count_data,observed=True)
model = pm.Model([observation, lambda_1, lambda_2, tau])
```

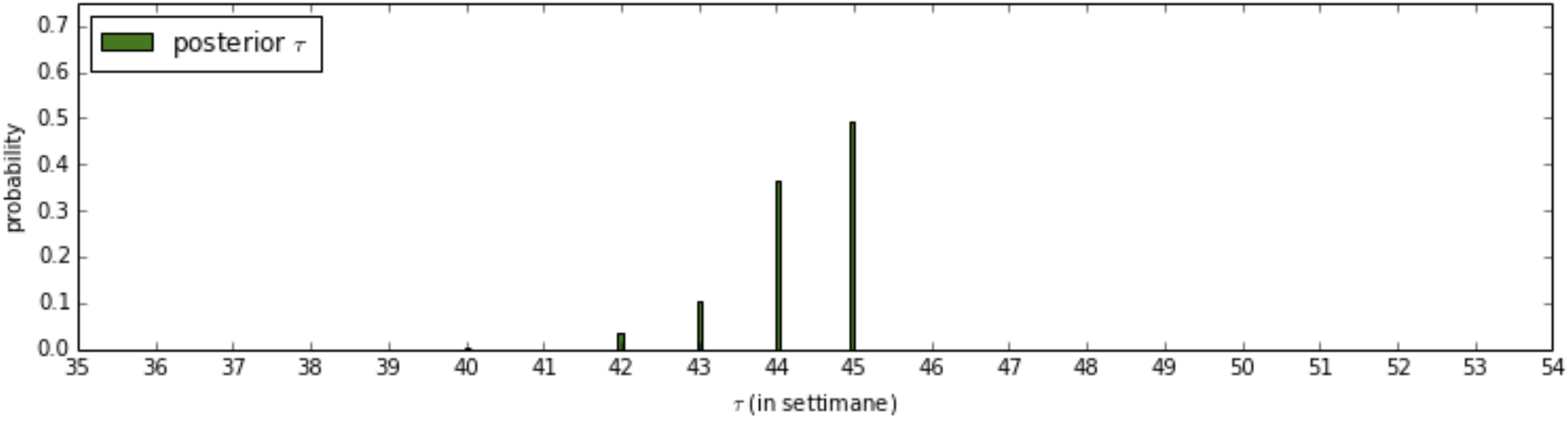
```
mcmc = pm.MCMC(model)
mcmc.sample(40000, 10000, 1)
```

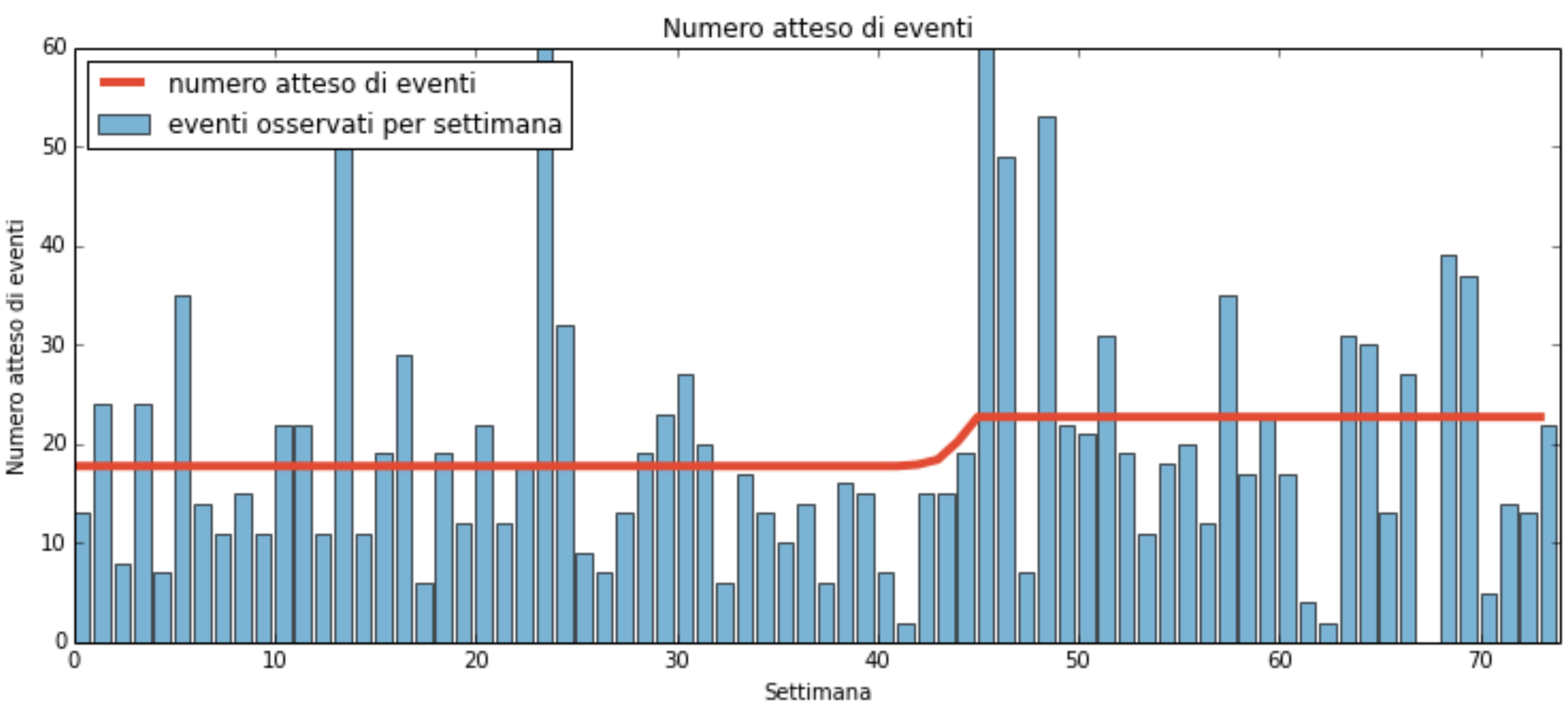
Distribuzione delle variabili λ_1, λ_2, τ




```
lambda_1_samples = mcmc.trace('lambda_1')[:]
lambda_2_samples = mcmc.trace('lambda_2')[:]
tau_samples = mcmc.trace('tau')[:]
```







$$\lambda = \begin{cases} \lambda_1 & = 17.7545366 \\ \lambda_2 & = 22.71789065 \end{cases}$$

Grazie

exedre@gmail.com