ASSIGNMENT #3: due TODAY

if you need extension, contact me

ASSIGNMENT #4: due FRIDAY 3/18

PROBABILITY II

Chapters 1+2 of Practical Statistics for Astronomers

Bayesian Inferences with Probability

GOAL: estimating the parameters of assumed probability distributions, i.e., we are assuming a model for our data and wish to find out how this model is characterized. In other words, we are *data modeling*.

We have a probability distribution (the likelihood) $f(data|\bar{\alpha})$ and we wish to know the parameter vector α . In the Bayesian route, we need to compute the posterior distribution of α

Bayesian Inferences with Probability

GOAL: estimating the parameters of assumed probability distributions, i.e., we are assuming a model for our data and wish to find out how this model is characterized. In other words, we are *data modeling*.

We have a probability distribution (the likelihood) $f(data|\bar{\alpha})$ and we wish to know the parameter vector α . In the Bayesian route, we need to compute the posterior distribution of α

EXAMPLE 1: Suppose we have N data X_i, drawn from a Gaussian of known standard deviation (σ) but unknown mean (μ), which we want to estimate.

Bayesian Inferences with Probability

GOAL: estimating the parameters of assumed probability distributions, i.e., we are assuming a model for our data and wish to find out how this model is characterized. In other words, we are *data modeling*.

We have a probability distribution (the likelihood) $f(data|\bar{\alpha})$ and we wish to know the parameter vector α . In the Bayesian route, we need to compute the posterior distribution of α

EXAMPLE 1: Suppose we have N data X_i, drawn from a Gaussian of known standard deviation (σ) but unknown mean (μ), which we want to estimate.

We need a prior on the mean: diffuse prior prob(μ)=constant over a wide range of μ

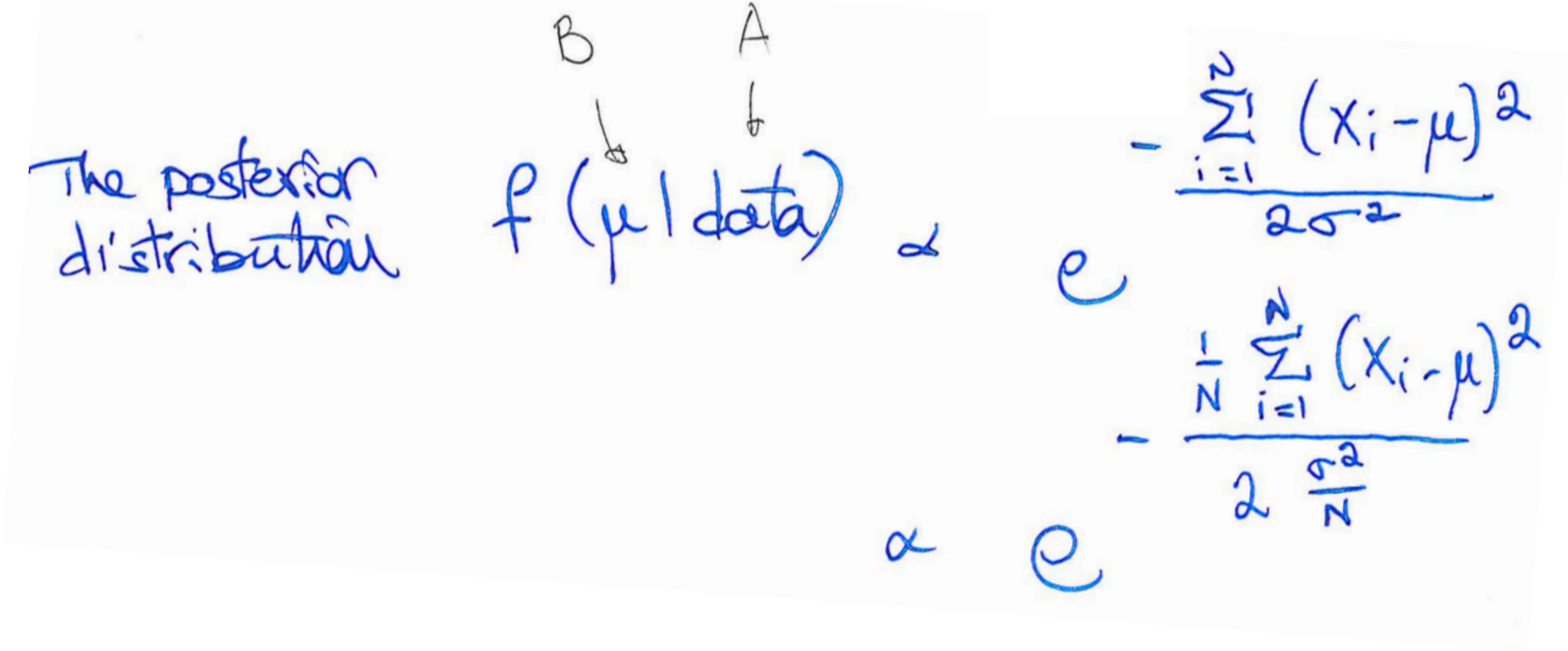
Bayesian Inferences with Probability

GOAL: estimating the parameters of assumed probability distributions, i.e., we are assuming a model for our data and wish to find out how this model is characterized. In other words, we are *data modeling*.

We have a probability distribution (the likelihood) $f(data|\bar{\alpha})$ and we wish to know the parameter vector α . In the Bayesian route, we need to compute the posterior distribution of α

EXAMPLE 1: Suppose we have N data X_i, drawn from a Gaussian of known standard deviation (σ) but unknown mean (μ), which we want to estimate.

We need a prior on the mean: diffuse prior prob(μ)=constant over a wide range of μ



The posterior distribution $f(\overset{B}{\mu} \mid \overset{A}{data}) \propto e^{-\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{2\sigma^2}}$

$$\propto e^{-\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}{2\frac{\sigma^2}{N}}}$$

So that the average of the data is distributed around the mean μ with variance σ^2/N

Bayesian Inferences with Probability

GOAL: estimating the parameters of assumed probability distributions, i.e., we are assuming a model for our data and wish to find out how this model is characterized. In other words, we are *data modeling*.

We have a probability distribution (the likelihood) $f(data|\bar{\alpha})$ and we wish to know the parameter vector α . In the Bayesian route, we need to compute the posterior distribution of α

EXAMPLE 1: Suppose we have N data X_i, drawn from a Gaussian of known standard deviation (σ) but unknown mean (μ), which we want to estimate.

We need a prior on the mean: diffuse prior prob(μ)=constant over a wide range of μ

The posterior distribution

$$f(\overset{B}{\mu}|\overset{A}{data}) \propto e^{-\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{2\sigma^2}}$$

$$\propto e^{-\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}{2\frac{\sigma^2}{N}}}$$

So that the average of the data is distributed around the mean μ with variance σ^2/N

This method is related to the classical technique of MAXIMUM LIKELIHOOD. If the prior is diffuse, then the posterior probability is proportional to the likelihood term $f(data|\bar{\alpha})$. Maximum likelihood picks out the mode (i.e., the peak) of the posterior, i.e., the value of α which maximizes the likelihood. We will learn more on this later…

EXAMPLE 2: Suppose we make an observation at a randomly selected position in the sky. Our model of the data, an event D consisting of a single measured flux density f, is that it is distributed in a Gaussian way about the true flux density S with variance σ^2 .

The extensive body of source counts tells us the a-priori distribution of S, prob(S)=KS$^{-5/2}$ (this is the prior) describing our prior state of knowledge. K normalizes the counts to 1, i.e., there is presumed to be one source in the beam at some flux-density level.

EXAMPLE 2: Suppose we make an observation at a randomly selected position in the sky. Our model of the data, an event D consisting of a single measured flux density f, is that it is distributed in a Gaussian way about the true flux density S with variance σ^2 .
The extensive body of source counts tells us the a-priori distribution of S, prob(S)=KS$^{-5/2}$ (this is the prior) describing our prior state of knowledge. K normalizes the counts to 1, i.e., there is presumed to be one source in the beam at some flux-density level.

The probability of observing f when the true value is S is the likelihood: $e^{-\frac{1}{2\sigma^2}(f-S)^2}$

EXAMPLE 2: Suppose we make an observation at a randomly selected position in the sky. Our model of the data, an event D consisting of a single measured flux density f, is that it is distributed in a Gaussian way about the true flux density S with variance σ^2 .
The extensive body of source counts tells us the a-priori distribution of S, prob(S)=KS$^{-5/2}$ (this is the prior) describing our prior state of knowledge. K normalizes the counts to 1, i.e., there is presumed to be one source in the beam at some flux-density level.

The probability of observing f when the true value is S is the likelihood: $e^{-\frac{1}{2\sigma^2}(f-S)^2}$

Bayes' theorem —> posterior probability = likelihood * prior:

$$prob(S|D) = K' e^{-\frac{1}{2\sigma^2}(f_i-S)^2} S^{-5/2}$$

EXAMPLE 2: Suppose we make an observation at a randomly selected position in the sky. Our model of the data, an event D consisting of a single measured flux density f, is that it is distributed in a Gaussian way about the true flux density S with variance σ^2 .
The extensive body of source counts tells us the a-priori distribution of S, prob(S)=KS$^{-5/2}$ (this is the prior) describing our prior state of knowledge. K normalizes the counts to 1, i.e., there is presumed to be one source in the beam at some flux-density level.

The probability of observing f when the true value is S is the likelihood: $e^{-\frac{1}{2\sigma^2}(f-S)^2}$

Bayes' theorem —> posterior probability = likelihood * prior:

$$prob(S|D) = K' e^{-\frac{1}{2\sigma^2}(f_i-S)^2} S^{-5/2}$$

If we are able to obtain n independent measurements of the flux f$_i$, then

$$prob(S|D) = K'' e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(f_i-S)^2} S^{-5/2}$$

EXAMPLE 2: Suppose we make an observation at a randomly selected position in the sky. Our model of the data, an event D consisting of a single measured flux density f, is that it is distributed in a Gaussian way about the true flux density S with variance σ^2 .
The extensive body of source counts tells us the a-priori distribution of S, prob(S)=KS$^{-5/2}$ (this is the prior) describing our prior state of knowledge. K normalizes the counts to 1, i.e., there is presumed to be one source in the beam at some flux-density level.

The probability of observing f when the true value is S is the likelihood: $e^{-\frac{1}{2\sigma^2}(f-S)^2}$

Bayes’ theorem —> posterior probability = likelihood * prior:

$$prob(S|D) = K'e^{-\frac{1}{2\sigma^2}(f_i-S)^2}S^{-5/2}$$

If we are able to obtain n independent measurements of the flux f$_i$, then

$$prob(S|D) = K''e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(f_i-S)^2}S^{-5/2}$$

Suppose that the source counts extend from 1 to 100 units, the noise level was $\sigma = 1$, and the data were 2, 1.3, 3, 1.5, 2, 1.8, then determine the posterior probability of the flux for the first 2, 4, and 6 measurements.
NOTE: The increase in data gradually overwhelms the prior but the prior affects the conclusions markedly when there are a few measurements.

EXAMPLE 2: Suppose we make an observation at a randomly selected position in the sky. Our model of the data, an event D consisting of a single measured flux density f, is that it is distributed in a Gaussian way about the true flux density S with variance σ^2 .

The extensive body of source counts tells us the a-priori distribution of S, prob(S)=KS$^{-5/2}$ (this is the prior) describing our prior state of knowledge. K normalizes the counts to 1, i.e., there is presumed to be one source in the beam at some flux-density level.

The probability of observing f when the true value is S is the likelihood: $e^{-\frac{1}{2\sigma^2}(f-S)^2}$

Bayes' theorem —> posterior probability = likelihood * prior:

$$prob(S|D) = K'e^{-\frac{1}{2\sigma^2}(f_i-S)^2}S^{-5/2}$$

If we are able to obtain n independent measurements of the flux f$_i$, then

$$prob(S|D) = K''e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(f_i-S)^2}S^{-5/2}$$

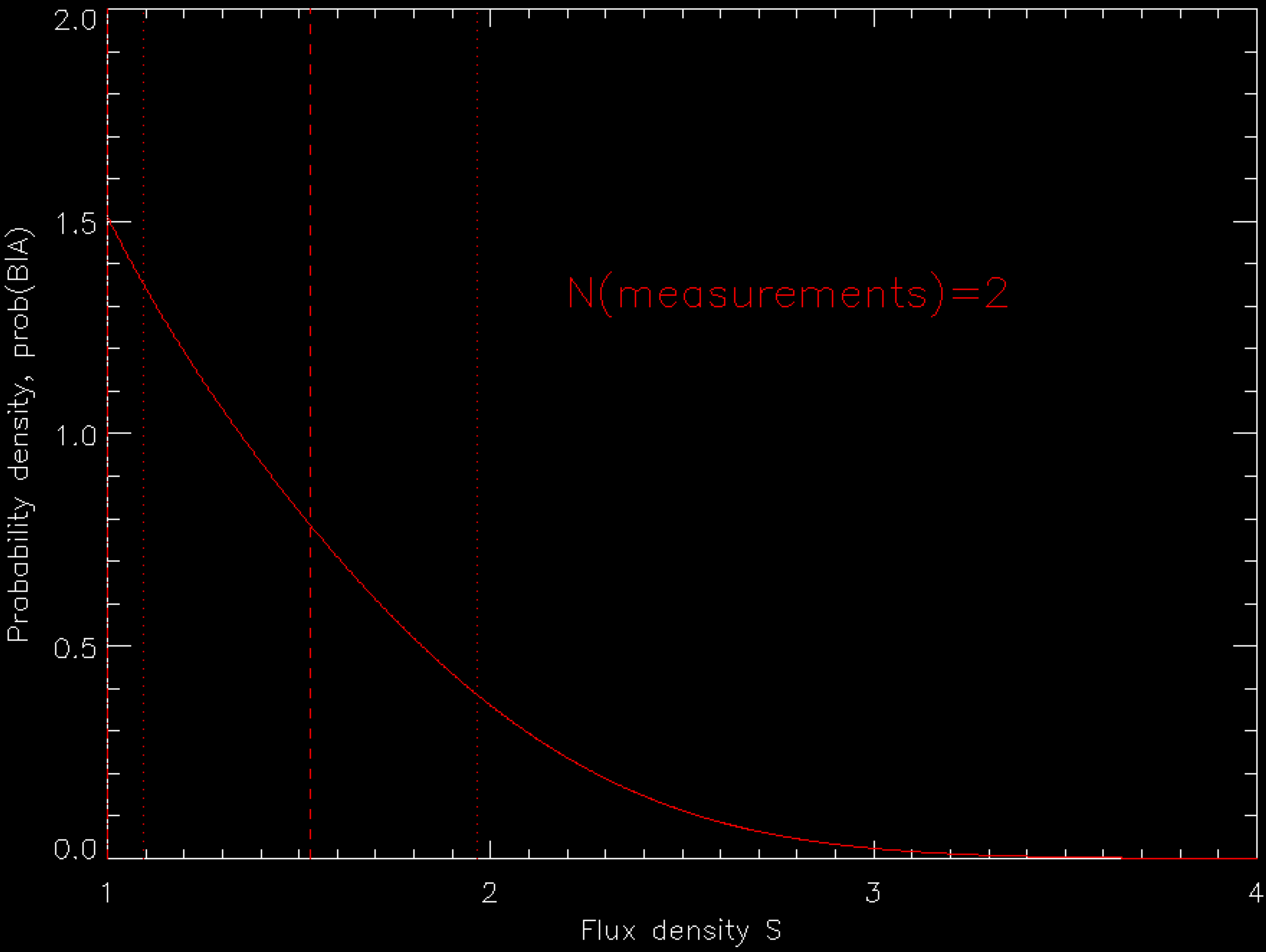
Suppose that the source counts extend from 1 to 100 units, the noise level was $\sigma = 1$, and the data were 2, 1.3, 3, 1.5, 2, 1.8, then determine the posterior probability of the flux for the first 2, 4, and 6 measurements.

NOTE: The increase in data gradually overwhelms the prior but the prior affects the conclusions markedly when there are a few measurements.

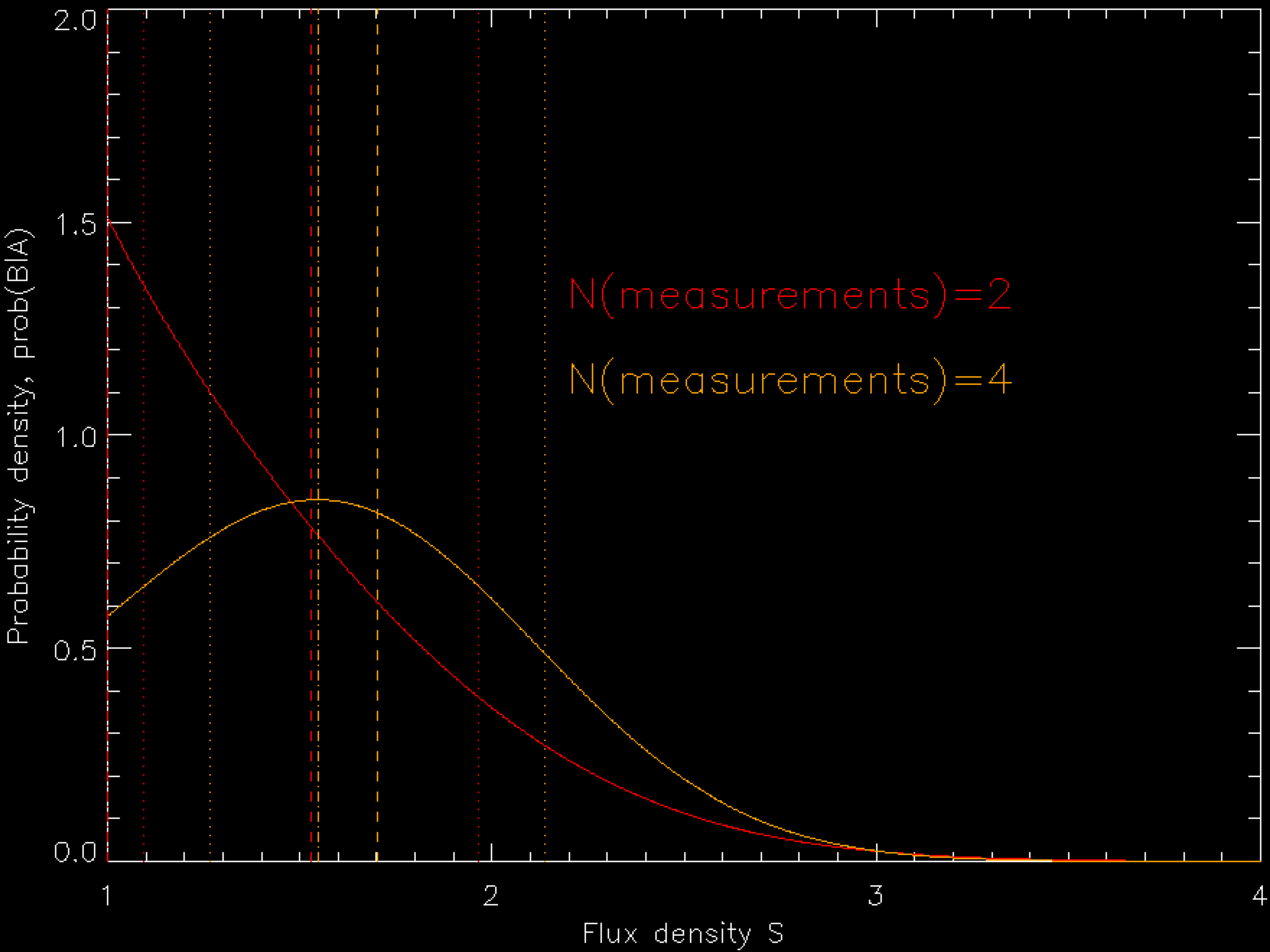
NOTE: If I knew nothing about the prior, the mean and sigma of the measurements [2, 1.3, 3, 1.5, 2, 1.8] are: $\mu = 1.93 \quad \sigma = 0.59$. From the posterior probability f(x):

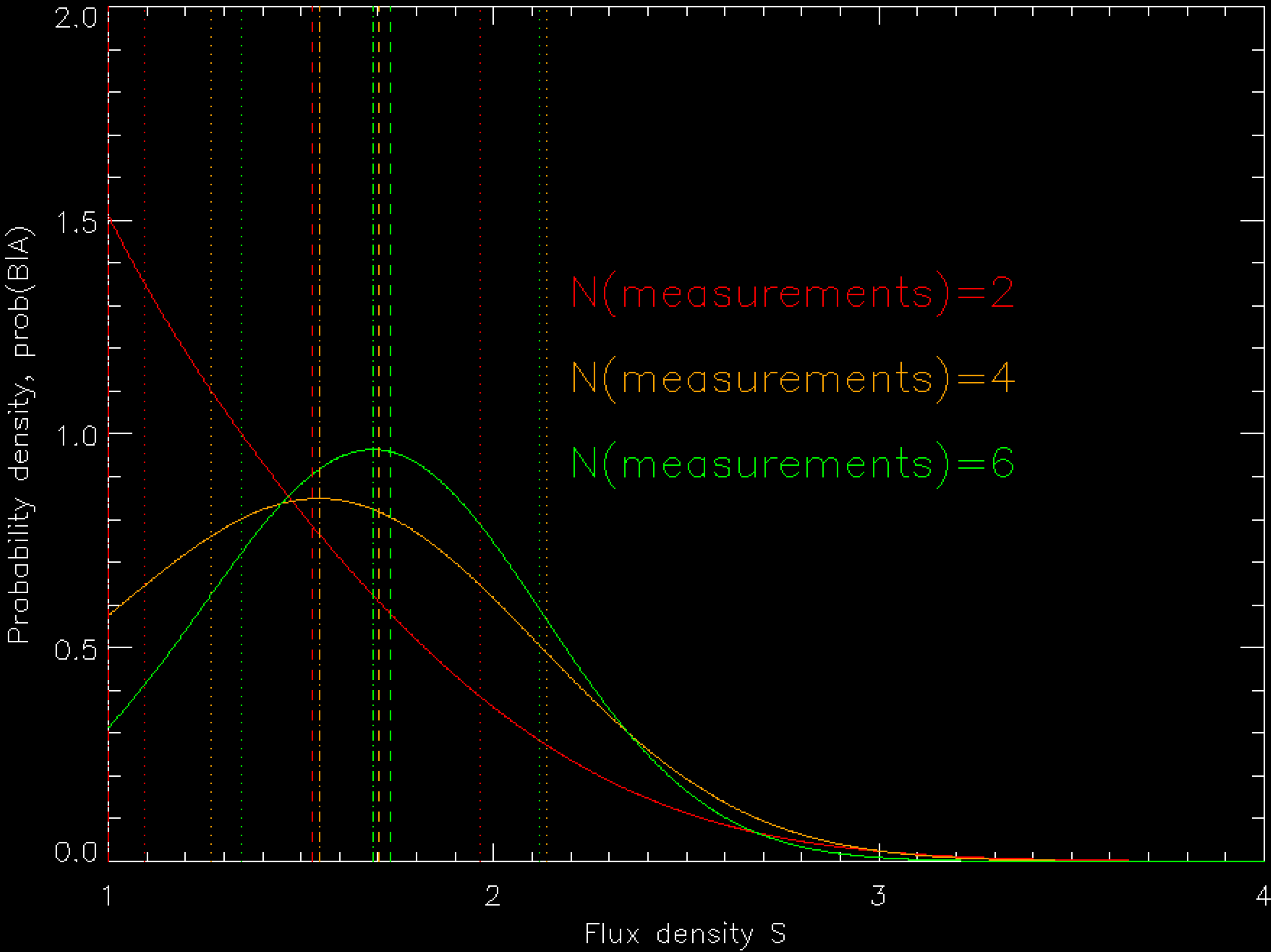
$$peak = 1.69 \quad \mu = 1.93 \quad \sigma = 0.39$$

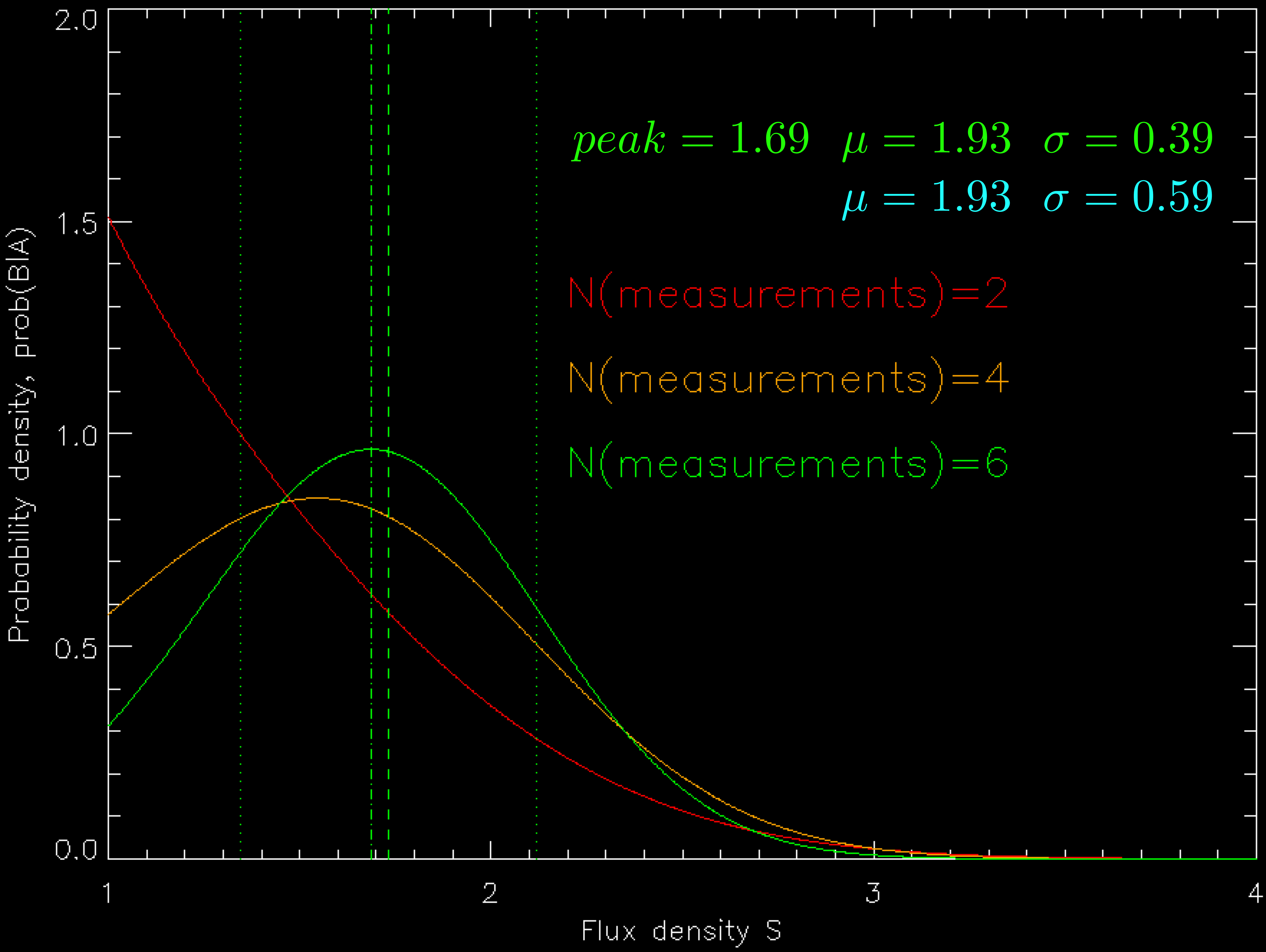
$$\mu = \int_{-\infty}^{+\infty} xf(x)dx \qquad \sigma^2 = \int_{-\infty}^{+\infty} (x-\mu)^2 f(x)dx$$



N(measurements)=2
Probability density, prob(B|A)
Flux density S
0.0
0.5
1.0
1.5
2.0
1
2
3
4


2.0
1.5
1.0
0.5
0.0
Probability density, prob(B|A)
1
2
3
4
Flux density S
N(measurements)=2
N(measurements)=4


2.0
1.5
1.0
0.5
0.0
Probability density, prob(B|A)
1
2
3
4
Flux density S
N(measurements)=2
N(measurements)=4
N(measurements)=6


$peak = 1.69 \quad \mu = 1.93 \quad \sigma = 0.39$
$\mu = 1.93 \quad \sigma = 0.59$
N(measurements)=2
N(measurements)=4
N(measurements)=6
Probability density, prob(B|A)
Flux density S
2.0
1.5
1.0
0.5
0.0
1
2
3
4

MONTE CARLO (random number) GENERATORS

There are frequent occasions in probability calculations, hypothesis testing and model fitting when it is essential to use a set of numbers distributed how we guess the data might be. We need random numbers.

MONTE CARLO (random number) GENERATORS

There are frequent occasions in probability calculations, hypothesis testing and model fitting when it is essential to use a set of numbers distributed how we guess the data might be. We need random numbers.

There are a number of methods to generate random numbers. Key issuesL

1. How long is it before the pseudo-random cycle is repeated? Or how many random numbers do you need? —> need to understand the characteristics of the generator
2. Follow the prescribed implementation precisely
3. The routines generate pseudo-random numbers, i.e., run them again from the same starting point and you will get the same set of numbers.

MONTE CARLO (random number) GENERATORS

There are frequent occasions in probability calculations, hypothesis testing and model fitting when it is essential to use a set of numbers distributed how we guess the data might be. We need random numbers.

There are a number of methods to generate random numbers. Key issuesL

1. How long is it before the pseudo-random cycle is repeated? Or how many random numbers do you need? —> need to understand the characteristics of the generator
2. Follow the prescribed implementation precisely
3. The routines generate pseudo-random numbers, i.e., run them again from the same starting point and you will get the same set of numbers.

Q: what is a random number produced by the computer?
A: after all, a computer will produce an output following a deterministic algorithm. The way out of this contradiction is that computer generated random numbers are not strictly random, are pseudo-random.

MONTE CARLO (random number) GENERATORS

There are frequent occasions in probability calculations, hypothesis testing and model fitting when it is essential to use a set of numbers distributed how we guess the data might be. We need random numbers.

There are a number of methods to generate random numbers. Key issuesL

1. How long is it before the pseudo-random cycle is repeated? Or how many random numbers do you need? —> need to understand the characteristics of the generator
2. Follow the prescribed implementation precisely
3. The routines generate pseudo-random numbers, i.e., run them again from the same starting point and you will get the same set of numbers.

Q: what is a random number produced by the computer?
A: after all, a computer will produce an output following a deterministic algorithm. The way out of this contradiction is that computer generated random numbers are not strictly random, are pseudo-random.

We need to have a pragmatic approach: a sequence of numbers would be considered random if so appears relative to a set of statistical tests which are aimed at pointing out correlations.

MONTE CARLO (random number) GENERATORS

There are frequent occasions in probability calculations, hypothesis testing and model fitting when it is essential to use a set of numbers distributed how we guess the data might be. We need random numbers.

There are a number of methods to generate random numbers. Key issuesL

1. How long is it before the pseudo-random cycle is repeated? Or how many random numbers do you need? —> need to understand the characteristics of the generator
2. Follow the prescribed implementation precisely
3. The routines generate pseudo-random numbers, i.e., run them again from the same starting point and you will get the same set of numbers.

Q: what is a random number produced by the computer?
A: after all, a computer will produce an output following a deterministic algorithm. The way out of this contradiction is that computer generated random numbers are not strictly random, are pseudo-random.

We need to have a pragmatic approach: a sequence of numbers would be considered random if so appears relative to a set of statistical tests which are aimed at pointing out correlations.

As a result, what is random for one application may not be random enough for another one.

MONTE CARLO (random number) GENERATORS

There are frequent occasions in probability calculations, hypothesis testing and model fitting when it is essential to use a set of numbers distributed how we guess the data might be. We need random numbers.

There are a number of methods to generate random numbers. Key issuesL

1. How long is it before the pseudo-random cycle is repeated? Or how many random numbers do you need? —> need to understand the characteristics of the generator
2. Follow the prescribed implementation precisely
3. The routines generate pseudo-random numbers, i.e., run them again from the same starting point and you will get the same set of numbers.

Q: what is a random number produced by the computer?
A: after all, a computer will produce an output following a deterministic algorithm. The way out of this contradiction is that computer generated random numbers are not strictly random, are pseudo-random.

We need to have a pragmatic approach: a sequence of numbers would be considered random if so appears relative to a set of statistical tests which are aimed at pointing out correlations.

As a result, what is random for one application may not be random enough for another one.

All computer generated random numbers have a certain statistical range of validity.

MONTE CARLO (random number) GENERATORS

There are frequent occasions in probability calculations, hypothesis testing and model fitting when it is essential to use a set of numbers distributed how we guess the data might be. We need random numbers.

There are a number of methods to generate random numbers. Key issuesL

1. How long is it before the pseudo-random cycle is repeated? Or how many random numbers do you need? —> need to understand the characteristics of the generator
2. Follow the prescribed implementation precisely
3. The routines generate pseudo-random numbers, i.e., run them again from the same starting point and you will get the same set of numbers.

Q: what is a random number produced by the computer?
A: after all, a computer will produce an output following a deterministic algorithm. The way out of this contradiction is that computer generated random numbers are not strictly random, are pseudo-random.

We need to have a pragmatic approach: a sequence of numbers would be considered random if so appears relative to a set of statistical tests which are aimed at pointing out correlations.

As a result, what is random for one application may not be random enough for another one.

All computer generated random numbers have a certain statistical range of validity.

The simplest distribution function for random numbers is a **constant probability distribution, a.k.a., uniform deviate**. Uniform deviates are the building blocks of random number generation and Monte Carlo techniques.

Never use a random number generator which has not been tested!

Never use a random number generator which has not been tested!

Numerical Recipes has a good variety of random number generators (all uniform deviates):

Never use a random number generator which has not been tested!

Numerical Recipes has a good variety of random number generators (all uniform deviates):

ran0: very basic and fast, period of correlation 2^{31}~10^{9}, i.e., you start to see correlation after 10^{9} numbers; computational cost cc=1.0 (normalized)

Never use a random number generator which has not been tested!

Numerical Recipes has a good variety of random number generators (all uniform deviates):

ran0: very basic and fast, period of correlation 2^{31}~10^{9}, i.e., you start to see correlation after 10^{9} numbers; computational cost cc=1.0 (normalized)

ran1: statistically better than ran0, but with smaller period ~10^{8}; cc=1.3

Never use a random number generator which has not been tested!

Numerical Recipes has a good variety of random number generators (all uniform deviates):

ran0: very basic and fast, period of correlation 2^{31}~10^9, i.e., you start to see correlation after 10^9 numbers; computational cost cc=1.0 (normalized)

ran1: statistically better than ran0, but with smaller period ~10^8; cc=1.3

ran2: statistically almost perfect; period ~10^{18} ~ infinity; cc=2.0

Never use a random number generator which has not been tested!

Numerical Recipes has a good variety of random number generators (all uniform deviates):

ran0: very basic and fast, period of correlation 2^{31}~10^9, i.e., you start to see correlation after 10^9 numbers; computational cost cc=1.0 (normalized)

ran1: statistically better than ran0, but with smaller period ~10^8; cc=1.3

ran2: statistically almost perfect; period ~10^{18} ~ infinity; cc=2.0

ranq1: quick and dirty, period ~10^4-10^6, cc=0.1, i.e., very fast
ranq2: quick and dirty, period ~10^4-10^6, cc=0.25, i.e., very fast

Never use a random number generator which has not been tested!

Numerical Recipes has a good variety of random number generators (all uniform deviates):

ran0: very basic and fast, period of correlation $2^{31}\sim10^{9}$, i.e., you start to see correlation after 10^{9} numbers; computational cost cc=1.0 (normalized)

ran1: statistically better than ran0, but with smaller period $\sim10^{8}$; cc=1.3

ran2: statistically almost perfect; period $\sim10^{18}$ ~ infinity; cc=2.0

ranq1: quick and dirty, period $\sim10^{4}$-10^{6}, cc=0.1, i.e., very fast

ranq2: quick and dirty, period $\sim10^{4}$-10^{6}, cc=0.25, i.e., very fast

Q: why use random numbers?

A: random numbers are the stepping stones of generating distributions functions (a.k.a., deviates)

Never use a random number generator which has not been tested!

Numerical Recipes has a good variety of random number generators (all uniform deviates):

ran0: very basic and fast, period of correlation 2^{31}~10^9, i.e., you start to see correlation after 10^9 numbers; computational cost cc=1.0 (normalized)

ran1: statistically better than ran0, but with smaller period ~10^8; cc=1.3

ran2: statistically almost perfect; period ~10^{18} ~ infinity; cc=2.0

ranq1: quick and dirty, period ~10^4-10^6, cc=0.1, i.e., very fast

ranq2: quick and dirty, period ~10^4-10^6, cc=0.25, i.e., very fast

Q: why use random numbers?
A: random numbers are the stepping stones of generating distributions functions (a.k.a., deviates)

Example: usually, a uniform deviate generator produces N random numbers between 0 and 1 from a uniform distribution. If I want N random numbers between 0 and 10 from a uniform distribution, I multiply those generated from the previous example by 10. If I want N random numbers between 2 and 12 from a uniform distribution, I multiply those generated from the first example by 10 and then I add 2.

So far, we have considered random number generators that produce uniform deviates.

How do we draw a set of random numbers following a given frequency distribution?

So far, we have considered random number generators that produce uniform deviates.

How do we draw a set of random numbers following a given frequency distribution?

Suppose that we have a way of producing random deviates that are uniformly distributed over the range 0-1 in the variable a, i.e., $a \in (0,1)$, and we have a functional form for our frequency distribution $\frac{dn}{dx} = f(x)$, and we want to draw random numbers from this f(x).

So far, we have considered random number generators that produce uniform deviates.

How do we draw a set of random numbers following a given frequency distribution?

Suppose that we have a way of producing random deviates that are uniformly distributed over the range 0-1 in the variable a, i.e., $a \in (0,1)$, and we have a functional form for our frequency distribution $\frac{dn}{dx} = f(x)$, and we want to draw random numbers from this f(x).

We need a transformation x=x(a) to distort the uniformity of a to follow f(x).

So far, we have considered random number generators that produce uniform deviates.

How do we draw a set of random numbers following a given frequency distribution?

Suppose that we have a way of producing random deviates that are uniformly distributed over the range 0-1 in the variable a, i.e., $a \in (0,1)$, and we have a functional form for our frequency distribution $\frac{dn}{dx} = f(x)$, and we want to draw random numbers from this f(x).

We need a transformation x=x(a) to distort the uniformity of a to follow f(x).

We know that $$f(x) = \frac{dn}{dx} = \frac{dn}{da}\frac{da}{dx} \underset{\big\uparrow}{=} \frac{da}{dx}$$

since $\frac{dn}{da}$ is uniform by assumption

So far, we have considered random number generators that produce uniform deviates.

How do we draw a set of random numbers following a given frequency distribution?

Suppose that we have a way of producing random deviates that are uniformly distributed over the range 0-1 in the variable a, i.e., $a \in (0,1)$, and we have a functional form for our frequency distribution $\dfrac{dn}{dx} = f(x)$, and we want to draw random numbers from this f(x).

We need a transformation x=x(a) to distort the uniformity of a to follow f(x).

We know that
$$f(x) = \frac{dn}{dx} = \frac{dn}{da}\frac{da}{dx} \underset{\uparrow}{=} \frac{da}{dx}$$

since $\dfrac{dn}{da}$ is uniform by assumption

Then $a(x) = \displaystyle\int^{x} f(x)dx$ ★

From whence the required transformation x=x(a), the inverse of ★ , i.e., solve ★ for x.

So far, we have considered random number generators that produce uniform deviates.

How do we draw a set of random numbers following a given frequency distribution?

Suppose that we have a way of producing random deviates that are uniformly distributed over the range 0-1 in the variable a, i.e., $a \in (0,1)$, and we have a functional form for our frequency distribution $\frac{dn}{dx} = f(x)$, and we want to draw random numbers from this f(x).

We need a transformation x=x(a) to distort the uniformity of a to follow f(x).

We know that $$f(x) = \frac{dn}{dx} = \frac{dn}{da}\frac{da}{dx} \underset{\uparrow}{=} \frac{da}{dx}$$

since $\frac{dn}{da}$ is uniform by assumption

Then $$a(x) = \int^{x} f(x)dx \quad ★$$

From whence the required transformation x=x(a), the inverse of ★ ,
i.e., solve ★ for x.

$a \in (0,1)$ → x=x(a) other random distribution

random uniform

EXAMPLE 1: $f(x)dx = -1.5x^{-2.5}dx$

EXAMPLE 1: $f(x)dx = -1.5x^{-2.5}dx$

➡ $$a(x) = \int^{x} f(x)dx = x^{-1.5}$$

EXAMPLE 1: $f(x)dx = -1.5x^{-2.5}dx$

➡ $$a(x) = \int^{x} f(x)dx = x^{-1.5}$$

➡ $$x = a^{-1/1.5} \qquad a \in (0,1)$$

EXAMPLE 1: $f(x)dx = -1.5x^{-2.5}dx$

$$\Rightarrow \quad a(x) = \int^{x} f(x)dx = x^{-1.5}$$

$$\Rightarrow \quad x = a^{-1/1.5} \qquad a \in (0,1)$$

EXAMPLE 2: exponential deviate

Suppose you want numbers randomly drawn from a distribution that is inversely exponential

EXAMPLE 1: $f(x)dx = -1.5x^{-2.5}dx$

$$\Rightarrow \quad a(x) = \int^{x} f(x)dx = x^{-1.5}$$

$$\Rightarrow \quad x = a^{-1/1.5} \qquad a \in (0,1)$$

EXAMPLE 2: exponential deviate

Suppose you want numbers randomly drawn from a distribution that is inversely exponential

$$f(x) = e^{-x} \quad \Rightarrow \quad x = -\ln a$$

EXAMPLE 1: $f(x)dx = -1.5x^{-2.5}dx$

$$\Rightarrow a(x) = \int^{x} f(x)dx = x^{-1.5}$$

$$\Rightarrow x = a^{-1/1.5} \qquad a \in (0,1)$$

EXAMPLE 2: exponential deviate

Suppose you want numbers randomly drawn from a distribution that is inversely exponential

$$f(x) = e^{-x} \quad \Rightarrow \quad x = -\ln a$$

```
function        expdev(seed)
                 tmp = ran1(seed)              —> this produces a uniform deviate
                 exdev = -ln(tmp)             —> this produces the exponential deviate
end function
```

EXAMPLE 1: $f(x)dx = -1.5x^{-2.5}dx$

➡ $a(x) = \int^{x} f(x)dx = x^{-1.5}$

➡ $x = a^{-1/1.5} \qquad a \in (0,1)$

EXAMPLE 2: exponential deviate

Suppose you want numbers randomly drawn from a distribution that is inversely exponential

$f(x) = e^{-x}$ ➡ $x = -\ln a$

```
function      expdev(seed)
                tmp = ran1(seed)            —> this produces a uniform deviate
                exdev = -ln(tmp)            —> this produces the exponential deviate
end function
```

In general,

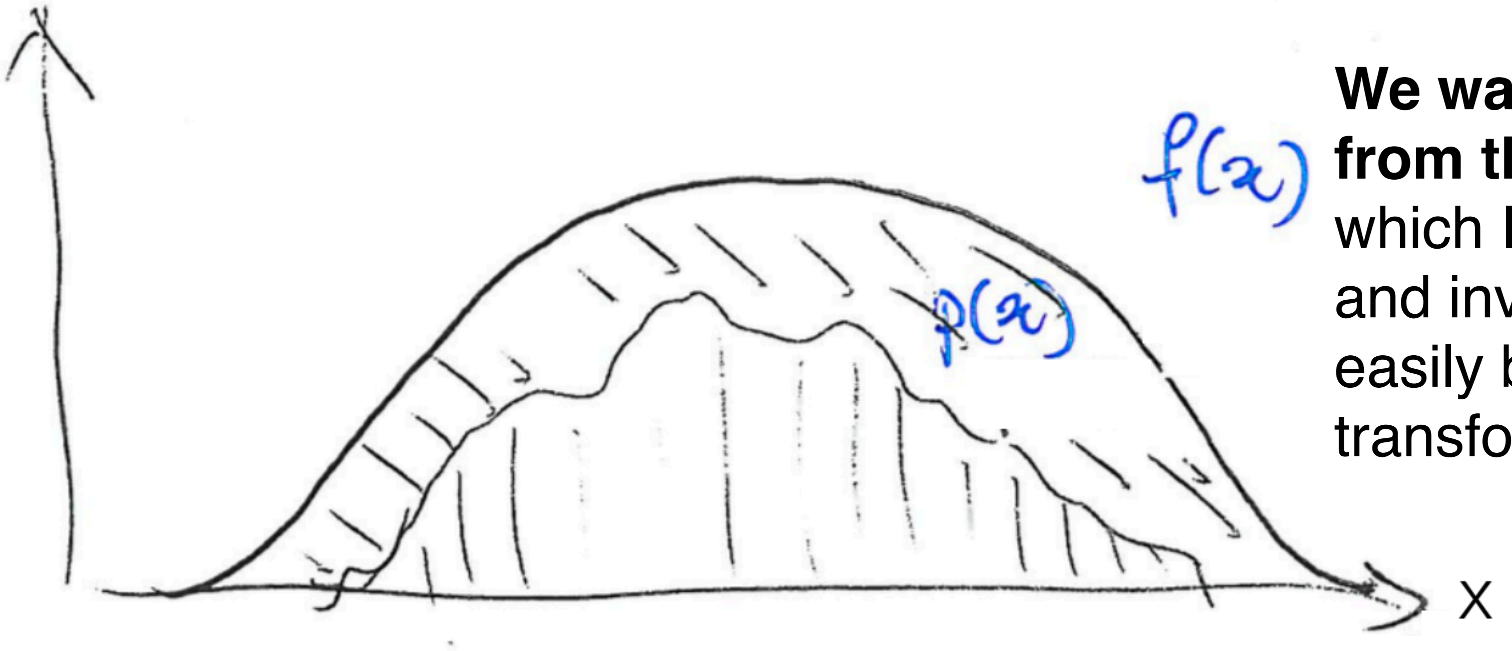
you want a set of random numbers drawn from a distribution f(x) ➡ $\left|\frac{da}{dx}\right| = f(x)$

Taking the integral, $a(x) = \int^{x} f(x)dx = F(x)$ primitive function

So that, x=x(a), x=F^{-1}(a) inverse function of the primitive (ex., F = e^{-} ; F^{-1}=-ln)

The transformation method has a limited validity: it is limited by the knowledge of $F^{-1}(a)$; this is known analytically for the exponential and a normal (Gaussian) deviates.

Q: What if $F^{-1}(a)$ cannot be calculated?
A: We use the rejection method (general, but not as efficient as the transformation method)



We want to produce random numbers drawn from the distribution p(x). f(x) is a Gaussian which I know to construct (it should be integrable and invertible, so that a random sample can easily be obtained from f(x) through the previous transformation method).

If I can construct a distribution function that follows f(x) and that incorporates p(x), then I can reject the excess and be left with the desired deviate.

Of course, there is an overhead = rejected points = $\int f(x)dx - \int p(x)dx$

The problem therefore is that of generating random numbers below f(x).

STEP 1: choose a random number with uniform deviate $\bar{a} \in [0, A]$

STEP 2: calculate $\bar{x}$ $so\ that$ $\int_0^{\bar{x}} f(x)dx = \bar{a} = F(\bar{x})$

STEP 3: once $f(\bar{x})$ in known, I choose a random number $\bar{a}$ from a uniform deviate between 0 and $A = f(\bar{x})$, i.e., $\bar{a} \in [0, A = f(\bar{x})]$

STEP 4: if $\bar{a} \leq p(\bar{x})$ → KEEP

if $\bar{a} > p(\bar{x})$ → REJECT

STEP 1: choose a random number with uniform deviate $\bar{a} \in [0, A]$

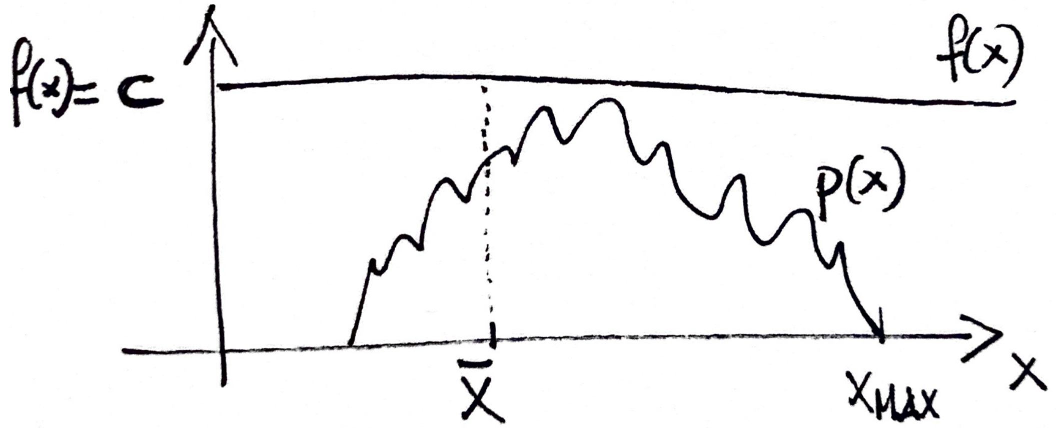
STEP 2: calculate $\bar{x}$ *so that* $\int_0^{\bar{x}} f(x)dx = \bar{a} = F(\bar{x})$

STEP 3: once $f(\bar{x})$ in known, I choose a random number $\bar{a}$ from a uniform deviate between 0 and $A = f(\bar{x})$, i.e., $\bar{a} \in [0, A = f(\bar{x})]$

STEP 4: if $\bar{a} \leq p(\bar{x})$ → KEEP

if $\bar{a} > p(\bar{x})$ → REJECT

EXAMPLE:



STEP 1: I pick a number $\bar{x}$ from a uniform deviate between 0 and x_{max}

STEP 2: I pick a random number from a uniform deviate between 0 and C, which I call $\bar{a}$

STEP 3: If

$$\bar{a} \leq p(\bar{x}) \quad \rightarrow \quad accept$$
$$\bar{a} > p(\bar{x}) \quad \rightarrow \quad reject$$

STEP 1: choose a random number with uniform deviate $\bar{a} \in [0, A]$

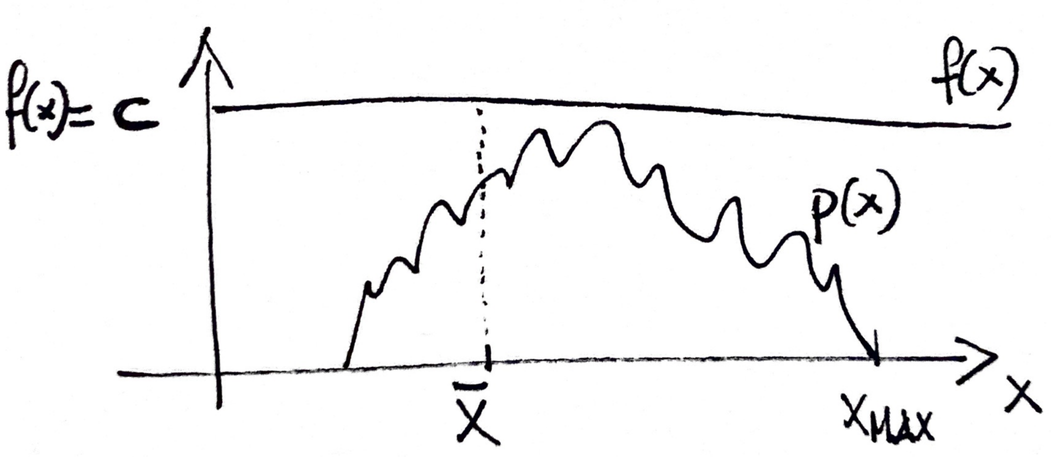
STEP 2: calculate $\bar{x}$ $\;so\ that\;$ $\int_0^{\bar{x}} f(x)dx = \bar{a} = F(\bar{x})$

STEP 3: once $f(\bar{x})$ in known, I choose a random number $\bar{a}$ from a uniform deviate between 0 and $A = f(\bar{x})$, i.e., $\bar{a} \in [0, A = f(\bar{x})]$

STEP 4: if $\bar{a} \leq p(\bar{x})$ → KEEP

if $\bar{a} > p(\bar{x})$ → REJECT

EXAMPLE:



STEP 1: I pick a number $\bar{x}$ from a uniform deviate between 0 and x_{max}

STEP 2: I pick a random number from a uniform deviate between 0 and C, which I call $\bar{a}$

STEP 3: If

$$\begin{aligned} \bar{a} \leq p(\bar{x}) &\rightarrow accept \\ \bar{a} > p(\bar{x}) &\rightarrow reject \end{aligned}$$

The REJECTION METHOD is easy to implement, but it can have large overheads, and the smarter f(x) is chosen, the less overheads it will have.