

Hypothesis Testing

Applied Multi-Messenger Astronomy
Hans Niederhausen

TUM - winter term 2020/21

Summary of last lecture

- A **random phenomenon** is a phenomenon whose outcome is unpredictable
- The **Frequentist probability** is the frequency of an outcome for a large (infinite) number of trials
- A **random variable** X is a variable whose possible values x are the outcomes of a random phenomenon. It is a variable in the sense that the frequency of its outcomes depends on the properties of the phenomenon (aka the parameters of a models). It is random in the sense that the outcome of the process is random, ergo unpredictable;

Summary of last lecture

- A **random phenomenon** is a phenomenon whose outcome is unpredictable
- The **Frequentist probability** is the frequency of an outcome for a large (infinite) number of trials
- A **random variable** X is a variable whose possible values x are the outcomes of a random phenomenon. It is a variable in the sense that the frequency of its outcomes depends on the properties of the phenomenon (aka the parameters of a models). It is random in the sense that the outcome of the process is random, ergo unpredictable;
- The **probability function** of a random variable $X \sim f(x)$ is a function that associates a probability value for each possible outcome to occur. If the function is continuous, it is called probability density function (PDF) if it is discrete it is called probability mass function (PMF)
- A **statistical model** is defined by a set of parameters, a set of basic random variables, and their probability (density) functions

Summary of last lecture

- A **random phenomenon** is a phenomenon whose outcome is unpredictable
- The **Frequentist probability** is the frequency of an outcome for a large (infinite) number of trials
- A **random variable** X is a variable whose possible values x are the outcomes of a random phenomenon. It is a variable in the sense that the frequency of its outcomes depends on the properties of the phenomenon (aka the parameters of a models). It is random in the sense that the outcome of the process is random, ergo unpredictable;
- The **probability function** of a random variable $X \sim f(x)$ is a function that associates a probability value for each possible outcome to occur. If the function is continuous, it is called probability density function (PDF) if it is discrete it is called probability mass function (PMF)
- A **statistical model** is defined by a set of parameters, a set of basic random variables, and their probability (density) functions
- The parameters of a model can be grouped in:
 - **parameters of interest:** those on which we want to do statistical inference
 - **nuisance parameters:** those derived from the data and known with a limited accuracy
 - **fixed parameters:** universal constants or predictions from theory without uncertainties)

Summary of last lecture

- The probability of a composite outcome (i.e. the simultaneous occurrence of outcomes of independent random processes) is the product of the probability of each outcome:
 $P(X, Y) = P(X) \cdot P(Y)$. The combined probability is called **joint probability**
- The **joint probability function of a model** is the composite probability of the probability functions of all random variables and represent the overall probability of an event given the model

Summary of last lecture

- The probability of a composite outcome (i.e. the simultaneous occurrence of outcomes of independent random processes) is the product of the probability of each outcome:
 $P(X, Y) = P(X) \cdot P(Y)$. The combined probability is called **joint probability**
- The **joint probability function of a model** is the composite probability of the probability functions of all random variables and represent the overall probability of an event given the model
- The **likelihood** is formally exactly the joint probability function but it is viewed as a function in which the outcome is fixed (defined by an experimental result) and the model parameters are a variable

Summary of last lecture

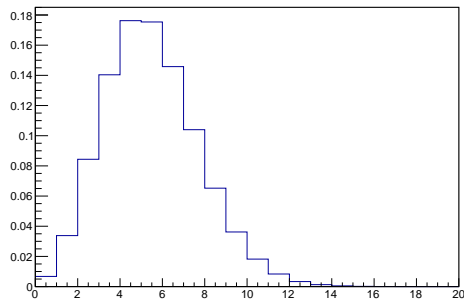
- The probability of a composite outcome (i.e. the simultaneous occurrence of outcomes of independent random processes) is the product of the probability of each outcome:
 $P(X, Y) = P(X) \cdot P(Y)$. The combined probability is called **joint probability**
- The **joint probability function of a model** is the composite probability of the probability functions of all random variables and represent the overall probability of an event given the model
- The **likelihood** is formally exactly the joint probability function but it is viewed as a function in which the outcome is fixed (defined by an experimental result) and the model parameters are a variable
- Before doing the experiment
 - what is the probability that I get a particular outcome?
 - the probability for an outcome is given by the probability function
- after running the experiment and having observed a data set
 - what is the model that gives my observed data with the largest probability?
 - the one that has a higher likelihood

Joint probability function vs likelihood

Consider a Poisson process and a counting experiment (random variable is N). The expected number of counts in the data set is $\lambda = 5$, the observed number of counts is 5:

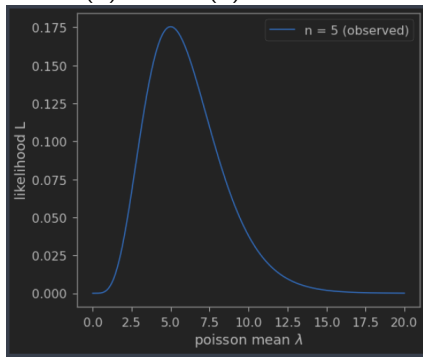
$$f_N(n; \lambda) = \mathcal{L}(\lambda; n) = \frac{e^{-\lambda} \lambda^n}{n!}$$

$f_N : n \rightarrow P(n)$ for $\lambda = 5$



what is the probability of an outcome given the model?

$\mathcal{L}(\lambda) : \lambda \rightarrow \mathcal{L}(\lambda)$ for $n = 5$



what is the model that gives my observed data with the largest probability?

Joint probability function vs likelihood

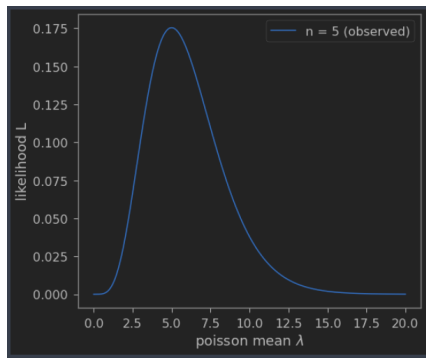
- Be careful to the pitfalls: the likelihood is not a probability function!
- A probability function is defined for a random variable, i.e. the outcome of a random process
- The parameters of a model are not random variables:
 - either the Higgs boson exists or it does not
 - if it exist its mass value is either one or an other
- That's why it's wrong to use the likelihood has the probability function of a model parameter
- Still, it is okay to ask what is probability of an outcome given a value of the model parameters. This is what is used to perform statistical inference

Joint probability vs likelihood

If we compare the likelihood at two points and find that

$$P_{\vec{\theta}_1}(\vec{X} = \vec{x}) = L(\vec{\theta}_1; \vec{x}) > L(\vec{\theta}_2; \vec{x}) = P_{\vec{\theta}_2}(\vec{X} = \vec{x})$$

then the observed data are more likely to have occurred if $\vec{\theta} = \vec{\theta}_1$ than if $\vec{\theta} = \vec{\theta}_2$, which can be interpreted as saying that $\vec{\theta}_1$ is a more plausible value for the true value of $\vec{\theta}$ than $\vec{\theta}_2$.



Summary of our statistical model

Parameters of models:

λ_s and λ_b , i.e. the expectation for the total numbers of signal and background events

Elementary random variables:

N number of events (signal and background); X energy of an event

Probability density function for a single event energy:

$$X \sim f_X(x; \mu, \sigma, \lambda_s, \lambda_b) = \frac{1}{\lambda_s + \lambda_b} \left(\lambda_s \cdot f_X^s(x; \mu, \sigma) + \lambda_b \cdot f_X^b(x) \right) = \frac{1}{\lambda_s + \lambda_b} \left[\lambda_s \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} + \lambda_b \cdot \frac{1}{20} \right]$$

Probability mass function for the number of events: background events:

$$N \sim f_N(n; \lambda_s + \lambda_b) = \frac{e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n}{n!}$$

Likelihood function for our reference example

For a given data set in which $N = n$ and $\vec{X} = \vec{x}$:

$$\begin{aligned}\mathcal{L}(\lambda_s, \lambda_b; n, \vec{x}) &= f_{N, \vec{x}}(n, \vec{x}; \mu, \sigma, \lambda_s, \lambda_b) \\ &= \frac{e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n}{n!} \cdot \prod_{i=1}^N \left\{ \frac{1}{\lambda_s + \lambda_b} \left[\lambda_s \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} + \lambda_b \cdot \frac{1}{20} \right] \right\}\end{aligned}$$

this is called an “extended unbinned” likelihood function

- it is unbinned in the sense that the energy of each events is used
- it is extended in the sense that the total number of events observed is not fixed and it is described by the Poisson term

Maximum likelihood Point Estimators

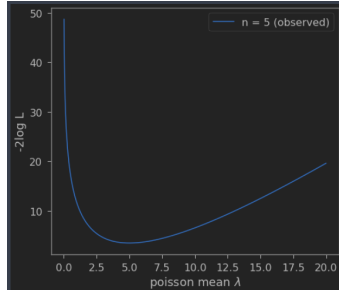
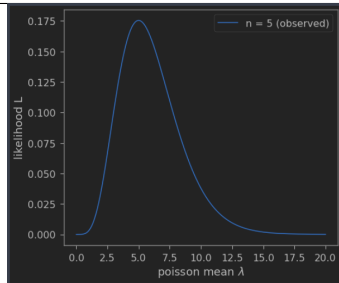
- Point Estimators are rules for calculating a single value which is to serve as a "best guess" or "best estimate" of an unknown parameter
- For a given outcome \vec{x} , the maximum likelihood estimator (MLE) of a parameter θ is the parameter value that maximizes $\mathcal{L}(\theta; \vec{x})$ considered as a function of θ with \vec{x} held fixed. We will denote the MLE with $\hat{\theta}(\vec{x})$

Maximum likelihood Point Estimators

- Point Estimators are rules for calculating a single value which is to serve as a "best guess" or "best estimate" of an unknown parameter
- For a given outcome \vec{x} , the maximum likelihood estimator (MLE) of a parameter θ is the parameter value that maximizes $\mathcal{L}(\theta; \vec{x})$ considered as a function of θ with \vec{x} held fixed. We will denote the MLE with $\hat{\theta}(\vec{x})$
- the maximum can be found with computational methods. To this purpose, the problem of finding the maximum of the likelihood is converted into finding the minimum of the negative log-likelihood:

$$\hat{\theta} = \left\{ \max_{\theta \in \Theta} \mathcal{L}(\theta; \mathbf{x}) \right\} = \left\{ \min_{\theta \in \Theta} -2 \log \mathcal{L}(\theta; \mathbf{x}) \right\}$$

The logarithm transforms products in sums that reduce numerical approximations



Questions?

Interlude: Multiple Random Variables and Correlations

Joint probability distributions

If two random variables are uncorrelated, the probability of a composite outcome is given by the product of the probability of each outcome:

$$P(X, Y) = P(X) \cdot P(Y)$$

Similarly, the probability distribution function for a composite outcome is

$$f_{X,Y}(x, y) = f_X(x) \cdot f_Y(y)$$

In general, if we have a vector of N random variables $\vec{X} = \{X_1, X_2, \dots, X_N\}$, their joint probability distribution is:

$$f_{\vec{X}}(\vec{x}) = \prod_{i=1}^N f_{X_i}(x_i)$$

Joint probability distributions

If two random variables are *correlated*, the formula's from last slide do not apply.
Let's develop some terminology.

Consider 2 events: Event A and Event B.

We will need to describe things like "*Probability for Event A and Event B to happen*", "*Probability for Event A given that B did not happen*", etc.

Joint probability distributions

If two random variables are *correlated*, the formula's from last slide do not apply.
Let's develop some terminology.

Consider 2 events: Event A and Event B.

We will need to describe things like "*Probability for Event A and Event B to happen*", "*Probability for Event A given that B did not happen*", etc.

Let's introduce random variables X and Y. In our example we have:

- $x = 0$: A happens.
- $x = 1$: A does not happen.
- $y = 0$: B happens.
- $y = 1$: B does not happen.

Joint probability distributions

The **joint probability** $P(x, y)$ describes the probability for the overall outcomes of both random processes. In our example there are four possible outcomes (X, Y) . The sample space is $S = \{(x = 0, y = 0), (x = 0, y = 1), (x = 1, y = 0), (x = 1, y = 1)\}$

Joint probability distributions

The **joint probability** $P(x, y)$ describes the probability for the overall outcomes of both random processes. In our example there are four possible outcomes (X, Y) . The sample space is $S = \{(x = 0, y = 0), (x = 0, y = 1), (x = 1, y = 0), (x = 1, y = 1)\}$

The **conditional probability** describes the probability for one random variable to assume a value given that a value for the other has been observed already. It's is denoted $P(x|y)$

Joint probability distributions

The **joint probability** $P(x, y)$ describes the probability for the overall outcomes of both random processes. In our example there are four possible outcomes (X, Y) . The sample space is $S = \{(x = 0, y = 0), (x = 0, y = 1), (x = 1, y = 0), (x = 1, y = 1)\}$

The **conditional probability** describes the probability for one random variable to assume a value given that a value for the other has been observed already. It's denoted $P(x|y)$

The **marginal probabilities** $P(x)$ and $P(y)$ describe the overall probability for each process to yield a certain observation.

They are related as follows:

$$P(x, y) = P(x|y) \cdot P(y) = P(y|x) \cdot P(x)$$

Joint probability distributions

This can be re-arranged to compute conditional probabilities from joint probabilities:

$$P(x|y) = \frac{P(x, y)}{P(y)}$$

The **marginal probabilities** can be computed from the **joint probabilities** via **marginalization**

$$P(x) = \sum_y p(x, y)$$

$$P(y) = \sum_x p(x, y)$$

Joint probability distributions

This can be re-arranged to compute conditional probabilities from joint probabilities:

$$P(x|y) = \frac{P(x, y)}{P(y)}$$

The **marginal probabilities** can be computed from the **joint probabilities** via **marginalization**

$$P(x) = \sum_y p(x, y)$$

$$P(y) = \sum_x p(x, y)$$

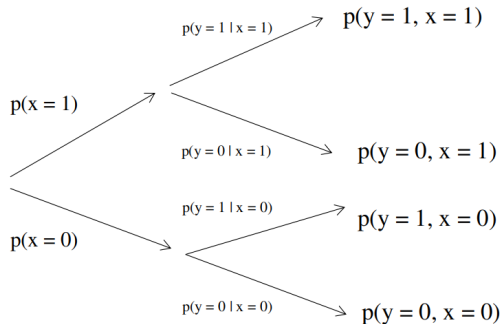
This produces **Bayes rule**

$$P(x|y) = \frac{P(y|x)P(x)}{\sum_x P(y|x)P(x)}$$

Example:

Event A: a person got infected by some new virus ($x=1$)

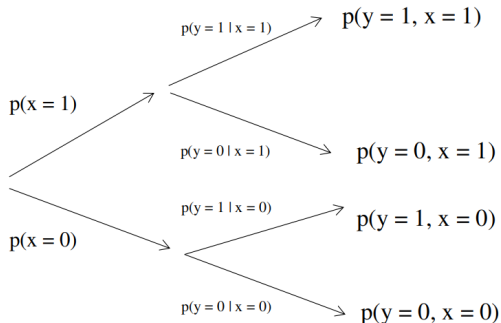
Event B: a person tests positive for the new virus ($y=1$)



Example:

Event A: a person got infected by some new virus ($x=1$)

Event B: a person tests positive for the new virus ($y=1$)



What's the probability to be infected given that one has tested positive?

Assume some test properties

$p(y=1|x=1) = 0.80$ (probability to test positive if infected)

$p(y=1|x=0) = 0.01$ (probability to test positive if not infected)

```
In [10]: import numpy as np
import matplotlib.pyplot as plt

def p_sick_given_positive(p_sick):
    p_healthy = 1. - p_sick

    # characteristics of test
    p_positive_given_sick = 0.8
    p_negative_given_sick = 1. - p_positive_given_sick
    p_positive_given_healthy = 0.01
    p_negative_given_healthy = 1. - p_positive_given_healthy

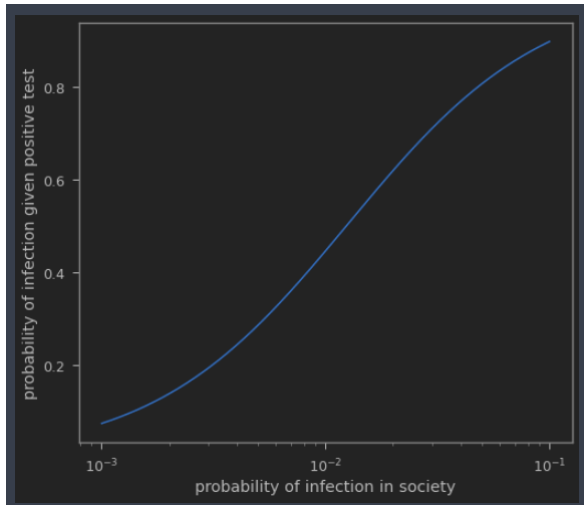
    # Bayes rule
    p_positive = p_positive_given_healthy * p_healthy + p_positive_given_sick * p_sick
    p_sick_given_positive = p_sick * p_positive_given_sick / p_positive

    return p_sick_given_positive
```

I didn't specify the probability for a random person to be infected, $p(x = 1)$.

We can answer the question as function of this rate of infection by using **Bayes rule**.

```
In [17]: p_sick_vals = np.logspace(-3,-1, 1000)
p_sick_given_positive_vals = [p_sick_given_positive(p_sick) for p_sick in p_sick_vals]
plt.plot(p_sick_vals, p_sick_given_positive_vals)
plt.xscale('log')
plt.xlabel('probability of infection in society')
plt.ylabel('probability of infection given positive test')
```



From last lecture: Random sampling in Toy Example

- Random sampling means to generate values $\{x_1, \dots, x_N\}$ for a random variable $X \sim f_X(x)$
- generate random values from a uniform distribution $U \sim \text{uniform}(0,1)$ (and other standard distributions) is relatively easy, the problem is to sample from non-standard distributions.
- Sampling from *mixture-models* can be done in two steps:

A

generate total counts from poisson distribution with expectation $\lambda_{tot} = \lambda_s + \lambda_b$

From last lecture: Random sampling in Toy Example

- Random sampling means to generate values $\{x_1, \dots, x_N\}$ for a random variable $X \sim f_X(x)$
- generate random values from a uniform distribution $U \sim \text{uniform}(0,1)$ (and other standard distributions) is relatively easy, the problem is to sample from non-standard distributions.
- Sampling from *mixture-models* can be done in two steps:

A

generate total counts from poisson distribution with expectation $\lambda_{tot} = \lambda_s + \lambda_b$

- step 1: for each count, randomly assign type (signal or bkg) by flipping a coin with probability of success given by probability to find signal type ($p_s = \lambda_s / \lambda_{tot}$)

From last lecture: Random sampling in Toy Example

- Random sampling means to generate values $\{x_1, \dots, x_N\}$ for a random variable $X \sim f_X(x)$
- generate random values from a uniform distribution $U \sim \text{uniform}(0,1)$ (and other standard distributions) is relatively easy, the problem is to sample from non-standard distributions.
- Sampling from *mixture-models* can be done in two steps:

A

generate total counts from poisson distribution with expectation $\lambda_{tot} = \lambda_s + \lambda_b$

- step 1: for each count, randomly assign type (signal or bkg) by flipping a coin with probability of success given by probability to find signal type ($p_s = \lambda_s / \lambda_{tot}$)
- step 2: for each count generate energy from either gaussian (if type is signal) or uniform (if type is background)

From last lecture: Random sampling in Toy Example

- Random sampling means to generate values $\{x_1, \dots, x_N\}$ for a random variable $X \sim f_X(x)$
- generate random values from a uniform distribution $U \sim \text{uniform}(0,1)$ (and other standard distributions) is relatively easy, the problem is to sample from non-standard distributions.
- Sampling from *mixture-models* can be done in two steps:

A

generate total counts from poisson distribution with expectation $\lambda_{tot} = \lambda_s + \lambda_b$

- step 1: for each count, randomly assign type (signal or bkg) by flipping a coin with probability of success given by probability to find signal type ($p_s = \lambda_s / \lambda_{tot}$)
- step 2: for each count generate energy from either gaussian (if type is signal) or uniform (if type is background)
- this procedure guarantees that the sample is equivalent to being produced directly from mixture model

From last lecture: Random sampling in Toy Example

```
In [7]: import numpy as np
import matplotlib.pyplot as plt

In [8]: lambda_s = 200
lambda_b = 2000

mu = 10
sigma = 1
x_min = 0
x_max = 20

rv_seed = 0
np.random.seed(rv_seed)

ntot = np.random.poisson(lambda_s + lambda_b)
```

```
# First step generate random numbers between 0 and 1 and attribute each
# event to signal or background using based on the ratio between
# signal rate and total rate
alpha = lambda_s / (lambda_s + lambda_b)
temp_rvars = np.random.uniform(0,1, ntot)
idx_s = np.where(temp_rvars < alpha)[0]
idx_b = np.where(temp_rvars >= alpha)[0]

# Second step. Store in "samples" the energy value of each event
samples = np.zeros(ntot)
# draw samples that are realized as background events from the background distribution
samples[idx_b] = np.random.uniform(x_min, x_max, len(idx_b))
# draw samples that are realized as signal events from the signal distribution
samples[idx_s] = np.random.normal(mu, sigma, len(idx_s))

bin_edges = np.linspace(x_min, x_max, 40)
plt.hist(samples, bins=bin_edges, label="Run 1 (total)")
plt.hist(samples[idx_s], bins=bin_edges, label="Run 1 (signal)", alpha=0.4)
plt.hist(samples[idx_b], bins=bin_edges, label="Run 1 (bkg)", alpha=0.4)
plt.xlabel("energy [keV]")
plt.ylabel("counts")
plt.legend(fontsize=16)
plt.show()
```


We sample the total number of events n from a poisson distribution: $n \sim \text{poisson}(\lambda_{tot})$.

Does the series of n coin tosses yield a number of signal events n_s that follows the expected poisson distribution $n_s \sim \text{poisson}(\lambda_s)$ as expected? Let's do the math!

We sample the total number of events n from a poisson distribution: $n \sim \text{poisson}(\lambda_{tot})$.

Does the series of n coin tosses yield a number of signal events n_s that follows the expected poisson distribution $n_s \sim \text{poisson}(\lambda_s)$ as expected? Let's do the math!

Realize that n coin tosses with success probability $\alpha = \frac{\lambda_s}{\lambda_{tot}}$ form a binomial process

$$n_s | n \sim \text{binom}(n, \alpha)$$

We sample the total number of events n from a poisson distribution: $n \sim \text{poisson}(\lambda_{tot})$.

Does the series of n coin tosses yield a number of signal events n_s that follows the expected poisson distribution $n_s \sim \text{poisson}(\lambda_s)$ as expected? Let's do the math!

Realize that n coin tosses with success probability $\alpha = \frac{\lambda_s}{\lambda_{tot}}$ form a binomial process

$$n_s | n \sim \text{binom}(n, \alpha)$$

Thus, we have

$$p(n; \lambda_{tot}) = \frac{\lambda_{tot}^n}{n!} \exp(-\lambda_{tot})$$

We sample the total number of events n from a poisson distribution: $n \sim \text{poisson}(\lambda_{\text{tot}})$.

Does the series of n coin tosses yield a number of signal events n_s that follows the expected poisson distribution $n_s \sim \text{poisson}(\lambda_s)$ as expected? Let's do the math!

Realize that n coin tosses with success probability $\alpha = \frac{\lambda_s}{\lambda_{\text{tot}}}$ form a binomial process

$$n_s | n \sim \text{binom}(n, \alpha)$$

Thus, we have

$$p(n; \lambda_{\text{tot}}) = \frac{\lambda_{\text{tot}}^n}{n!} \exp(-\lambda_{\text{tot}})$$

and

$$p(n_s | n, \alpha) = \frac{n!}{n_s! (n - n_s)!} \alpha^{n_s} (1 - \alpha)^{n - n_s}$$

We sample the total number of events n from a poisson distribution: $n \sim \text{poisson}(\lambda_{\text{tot}})$.

Does the series of n coin tosses yield a number of signal events n_s that follows the expected poisson distribution $n_s \sim \text{poisson}(\lambda_s)$ as expected? Let's do the math!

Realize that n coin tosses with success probability $\alpha = \frac{\lambda_s}{\lambda_{\text{tot}}}$ form a binomial process

$$n_s | n \sim \text{binom}(n, \alpha)$$

Thus, we have

$$p(n; \lambda_{\text{tot}}) = \frac{\lambda_{\text{tot}}^n}{n!} \exp(-\lambda_{\text{tot}})$$

and

$$p(n_s | n, \alpha) = \frac{n!}{n_s!(n - n_s)!} \alpha^{n_s} (1 - \alpha)^{n - n_s}$$

Need to find the marginal probability of n_s .

$$p(n_s) = \sum_{n=0}^{\infty} p(n_s, n) = \sum_{n \geq n_s}^{\infty} p(n_s, n)$$

$$p(n_s, n) = p(n_s | n) \cdot p(n) = \frac{n!}{n_s!(n - n_s)!} \alpha^{n_s} (1 - \alpha)^{n - n_s} \frac{\lambda_{tot}^n}{n!} \exp(-\lambda_{tot})$$

$$p(n_s, n) = p(n_s | n) \cdot p(n) = \frac{n!}{n_s!(n - n_s)!} \alpha^{n_s} (1 - \alpha)^{n - n_s} \frac{\lambda_{tot}^n}{n!} \exp(-\lambda_{tot})$$

$$p(n_s) = \sum_{n \geq n_s}^{\infty} \frac{\alpha^{n_s}}{n_s!(n - n_s)!} (1 - \alpha)^{n - n_s} \lambda_{tot}^n \exp(-\lambda_{tot})$$

$$p(n_s, n) = p(n_s | n) \cdot p(n) = \frac{n!}{n_s!(n - n_s)!} \alpha^{n_s} (1 - \alpha)^{n - n_s} \frac{\lambda_{tot}^n}{n!} \exp(-\lambda_{tot})$$

$$p(n_s) = \sum_{n \geq n_s}^{\infty} \frac{\alpha^{n_s}}{n_s!(n - n_s)!} (1 - \alpha)^{n - n_s} \lambda_{tot}^n \exp(-\lambda_{tot})$$

$$p(n_s) = \left(\frac{\alpha}{1 - \alpha} \right)^{n_s} \frac{1}{n_s!} \sum_{n \geq n_s}^{\infty} \frac{(1 - \alpha)^n}{(n - n_s)!} \lambda_{tot}^n \exp(-\lambda_{tot})$$

$$p(n_s, n) = p(n_s | n) \cdot p(n) = \frac{n!}{n_s!(n - n_s)!} \alpha^{n_s} (1 - \alpha)^{n - n_s} \frac{\lambda_{tot}^n}{n!} \exp(-\lambda_{tot})$$

$$p(n_s) = \sum_{n \geq n_s}^{\infty} \frac{\alpha^{n_s}}{n_s!(n - n_s)!} (1 - \alpha)^{n - n_s} \lambda_{tot}^n \exp(-\lambda_{tot})$$

$$p(n_s) = \left(\frac{\alpha}{1 - \alpha} \right)^{n_s} \frac{1}{n_s!} \sum_{n \geq n_s}^{\infty} \frac{(1 - \alpha)^n}{(n - n_s)!} \lambda_{tot}^n \exp(-\lambda_{tot})$$

Substitute $n \rightarrow n' = n - n_s$ (aka $n = n' + n_s$), then the sum starts at $n' = 0$.

$$p(n_s, n) = p(n_s | n) \cdot p(n) = \frac{n!}{n_s!(n - n_s)!} \alpha^{n_s} (1 - \alpha)^{n - n_s} \frac{\lambda_{tot}^n}{n!} \exp(-\lambda_{tot})$$

$$p(n_s) = \sum_{n \geq n_s}^{\infty} \frac{\alpha^{n_s}}{n_s!(n - n_s)!} (1 - \alpha)^{n - n_s} \lambda_{tot}^n \exp(-\lambda_{tot})$$

$$p(n_s) = \left(\frac{\alpha}{1 - \alpha} \right)^{n_s} \frac{1}{n_s!} \sum_{n \geq n_s}^{\infty} \frac{(1 - \alpha)^n}{(n - n_s)!} \lambda_{tot}^n \exp(-\lambda_{tot})$$

Substitute $n \rightarrow n' = n - n_s$ (aka $n = n' + n_s$), then the sum starts at $n' = 0$.

$$p(n_s) = \left(\frac{\alpha}{1 - \alpha} \right)^{n_s} \frac{1}{n_s!} \sum_{n'=0}^{\infty} \frac{(1 - \alpha)^{(n'+n_s)}}{(n' + n_s - n_s)!} \lambda_{tot}^{(n'+n_s)} \exp(-\lambda_{tot})$$

$$p(n_s, n) = p(n_s | n) \cdot p(n) = \frac{n!}{n_s!(n - n_s)!} \alpha^{n_s} (1 - \alpha)^{n - n_s} \frac{\lambda_{tot}^n}{n!} \exp(-\lambda_{tot})$$

$$p(n_s) = \sum_{n \geq n_s}^{\infty} \frac{\alpha^{n_s}}{n_s!(n - n_s)!} (1 - \alpha)^{n - n_s} \lambda_{tot}^n \exp(-\lambda_{tot})$$

$$p(n_s) = \left(\frac{\alpha}{1 - \alpha} \right)^{n_s} \frac{1}{n_s!} \sum_{n \geq n_s}^{\infty} \frac{(1 - \alpha)^n}{(n - n_s)!} \lambda_{tot}^n \exp(-\lambda_{tot})$$

Substitute $n \rightarrow n' = n - n_s$ (aka $n = n' + n_s$), then the sum starts at $n' = 0$.

$$p(n_s) = \left(\frac{\alpha}{1 - \alpha} \right)^{n_s} \frac{1}{n_s!} \sum_{n'=0}^{\infty} \frac{(1 - \alpha)^{(n'+n_s)}}{(n' + n_s - n_s)!} \lambda_{tot}^{(n'+n_s)} \exp(-\lambda_{tot})$$

Factors in powers of n_s can be moved in front of the sum. We get some nice cancellations.

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot})$$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot})$$

Insert a factor of $1 = \exp(-\alpha \lambda_{tot}) \cdot \exp(\alpha \lambda_{tot})$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot}) \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot}) \exp(\alpha \lambda_{tot})$$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot})$$

Insert a factor of $1 = \exp(-\alpha \lambda_{tot}) \cdot \exp(\alpha \lambda_{tot})$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot}) \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot}) \exp(\alpha \lambda_{tot})$$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot}) \sum_{n'=0}^{\infty} \frac{(\lambda_{tot} (1-\alpha))^{n'}}{n'!} \exp(-(\lambda_{tot}(1-\alpha)))$$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot})$$

Insert a factor of $1 = \exp(-\alpha \lambda_{tot}) \cdot \exp(\alpha \lambda_{tot})$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot}) \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot}) \exp(\alpha \lambda_{tot})$$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot}) \sum_{n'=0}^{\infty} \frac{(\lambda_{tot} (1-\alpha))^{n'}}{n'!} \exp(-(\lambda_{tot}(1-\alpha)))$$

The pesky sum equals unity (because poisson probabilities sum to 1). Thus we have

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot})$$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot})$$

Insert a factor of $1 = \exp(-\alpha \lambda_{tot}) \cdot \exp(\alpha \lambda_{tot})$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot}) \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot}) \exp(\alpha \lambda_{tot})$$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot}) \sum_{n'=0}^{\infty} \frac{(\lambda_{tot} (1-\alpha))^{n'}}{n'!} \exp(-(\lambda_{tot}(1-\alpha)))$$

The pesky sum equals unity (because poisson probabilities sum to 1). Thus we have

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot})$$

and we can recognize that this means $n_s \sim \text{poisson}(\alpha \lambda_{tot})$, with $\alpha \lambda_{tot} = \frac{\lambda_s}{\lambda_{tot}} \lambda_{tot} = \lambda_s$.

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot})$$

Insert a factor of $1 = \exp(-\alpha \lambda_{tot}) \cdot \exp(\alpha \lambda_{tot})$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot}) \sum_{n'=0}^{\infty} \frac{(1-\alpha)^{n'}}{n'!} \lambda_{tot}^{n'} \exp(-\lambda_{tot}) \exp(\alpha \lambda_{tot})$$

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot}) \sum_{n'=0}^{\infty} \frac{(\lambda_{tot} (1-\alpha))^{n'}}{n'!} \exp(-(\lambda_{tot}(1-\alpha)))$$

The pesky sum equals unity (because poisson probabilities sum to 1). Thus we have

$$p(n_s) = \frac{(\alpha \lambda_{tot})^{n_s}}{n_s!} \exp(-\alpha \lambda_{tot})$$

and we can recognize that this means $n_s \sim \text{poisson}(\alpha \lambda_{tot})$, with $\alpha \lambda_{tot} = \frac{\lambda_s}{\lambda_{tot}} \lambda_{tot} = \lambda_s$. Hence

$$n_s \sim \text{poisson}(\lambda_s)$$

Multiple continuous random variables

Math is kind. The same formulas can be used with continuous random variables and their PDFs.

joint pdf:

$$f(x, y) = f(x | y) \cdot f(y) = f(y | x) \cdot f(x)$$

marginal pdf:

$$f(x) = \int dy f(x, y)$$

and

$$f(y) = \int dx f(x, y)$$

conditional pdf:

$$f(x | y) = f(x, y) / f(y)$$

The multi-variate normal distribution

One common example is the **multi-variate normal distribution**.

$$f(\vec{x}) = \left(\frac{1}{2\pi}\right)^{d/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu}) \Sigma^{-1} (\vec{x} - \vec{\mu})^T\right)$$

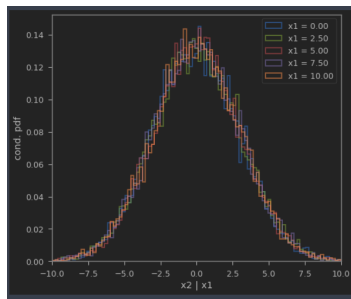
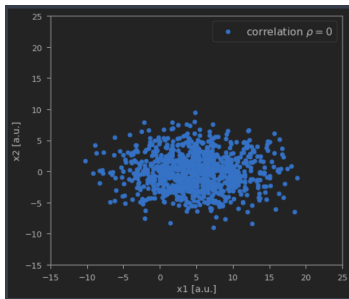
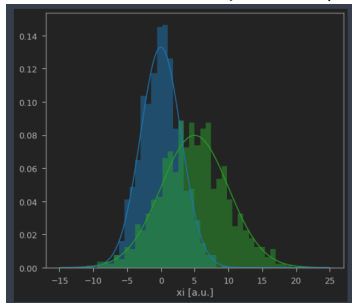
for $d = 2$ the covariance matrix Σ can be expressed as

$$\Sigma = \begin{vmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{vmatrix}$$

and $0 < \rho < 1$ measured the amount of correlation between the components x_1 and x_2 .

The multi-variate normal distribution

examples with $\rho = 0$ (independent case, identical to 2 standard normal RVS)

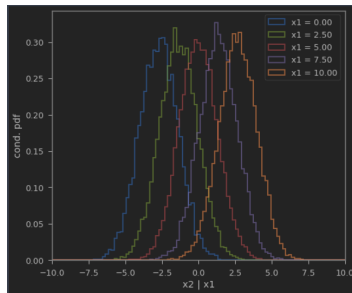
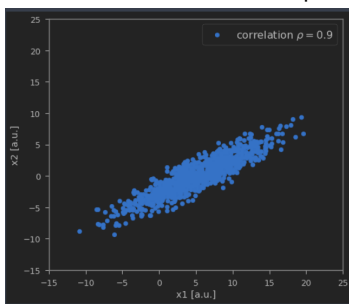
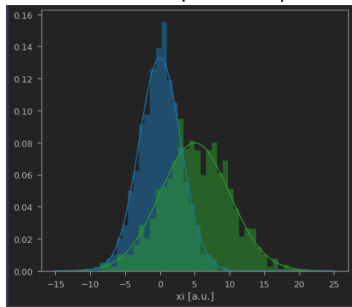


Note how the conditional distribution $f(x_2 | x_1)$ is independent from the value of x_1 .

See also jupyter notebook.

The multi-variate normal distribution

examples with $\rho = 0.9$ (can not be created from 2 independent standard normal RVS)



Note how the conditional distribution $f(x_2 | x_1)$ is depends on the value of x_1 .

See also jupyter notebook.

Example for continuous marginalization

will post a jupyter notebook.

(calculate reconstructed energy pdf from monochromatic neutrino beam in a neutral-current only universe).

- new task of statistical inference!
- The goal of hypothesis testing is to decide, based on an observed outcome and a model, which of two complementary hypothesis is true.
- ingredients needed:
 - definition of hypothesis
 - a test statistic
 - an acceptance/rejection rule

What's an hypothesis?

- A hypothesis is a statement about a model or about a parameter of a model
- The two complementary hypotheses in a hypothesis testing problem are called:
 - null hypothesis H_0
 - alternative hypothesis H_1
- **complementary** hypotheses means that
 - either H_0 is true and H_1 is false
 - or H_0 is false and H_1 is true

What's an hypothesis?

- A hypothesis is a statement about a model or about a parameter of a model
 - The two complementary hypotheses in a hypothesis testing problem are called:
 - null hypothesis H_0
 - alternative hypothesis H_1
 - **complementary** hypotheses means that
 - either H_0 is true and H_1 is false
 - or H_0 is false and H_1 is true
 - if θ denotes a parameter of a model, the general form of the null and alternative hypothesis is:
 - $H_0 : \theta \in \Theta_0$
 - $H_1 : \theta \in \Theta_0^c$
- where Θ_0 is some subset of the parameter space allowed for θ and Θ_0^c is its complement

$$\Theta_0 \cup \Theta_0^c = \Theta$$

Examples

Example 1: experimentation of a new drug

- θ denotes the average change in a patient's blood pressure after taking an experimental drug
- question of interest:
 - $H_0 : \theta = 0$ (the treatment has no effect)
 - $H_1 : \theta \neq 0$ (the treatment has an effect)
- this common situation in which H_0 states that a treatment has no effect led to the term “null” hypothesis

Examples

Example 1: experimentation of a new drug

- θ denotes the average change in a patient's blood pressure after taking an experimental drug
- question of interest:
 - $H_0 : \theta = 0$ (the treatment has no effect)
 - $H_1 : \theta \neq 0$ (the treatment has an effect)
- this common situation in which H_0 states that a treatment has no effect led to the term “null” hypothesis

Example 2: proportion of defective items

- question of interest:
 - $H_0 : \theta \geq \theta_0$
 - $H_1 : \theta < \theta_0$
- the value θ_0 is the maximum proportion of defective items acceptable by the producer

Hypothesis testing procedure

- A Hypothesis testing procedure is a rule that specifies:
 - for which set of outcomes the decision is made to accept H_0 as true
 - for which set of outcomes H_0 is rejected and H_1 is accepted as true
- The subset of the outcome space for which H_0 will be rejected is called **rejection region**. Its complement is called the **acceptance region**

Hypothesis testing procedure

- A Hypothesis testing procedure is a rule that specifies:
 - for which set of outcomes the decision is made to accept H_0 as true
 - for which set of outcomes H_0 is rejected and H_1 is accepted as true
- The subset of the outcome space for which H_0 will be rejected is called **rejection region**. Its complement is called the **acceptance region**
- A hypothesis test is typically specified in terms of a test statistic

$$T(X_1, \dots, X_n) : \vec{X} \rightarrow \mathbb{R}$$

i.e. a function that maps an outcome into a scalar number

Hypothesis testing procedure

- A Hypothesis testing procedure is a rule that specifies:
 - for which set of outcomes the decision is made to accept H_0 as true
 - for which set of outcomes H_0 is rejected and H_1 is accepted as true
- The subset of the outcome space for which H_0 will be rejected is called **rejection region**. Its complement is called the **acceptance region**
- A hypothesis test is typically specified in terms of a test statistic

$$T(X_1, \dots, X_n) : \vec{X} \rightarrow \mathbb{R}$$

i.e. a function that maps an outcome into a scalar number

- Example 1: an example of rule to reject H_0 is when the mean change of blood pressure is above 10 mmHg (mean over K patients). Indicating with x_i the measured pressure change for the i -th patient, this rule can be formally expressed giving a test statistic:

$$T(X_1, \dots, X_k) = \vec{X}_{\text{mean}} = \sum_{i=1}^K X_i / K$$

and a rejection region:

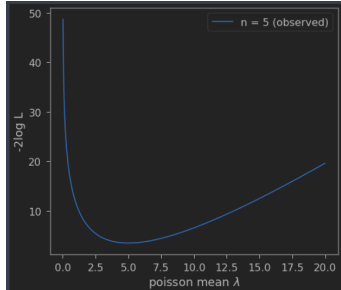
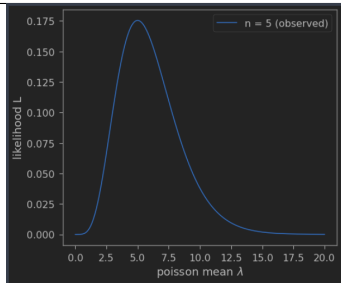
$$\{(x_1, \dots, x_k) : T(x_1, \dots, x_k) = \vec{X}_{\text{mean}} > 10 \text{ mmHg}\}$$

Likelihood Ratio Test Statistics

- there are different methods of finding test procedures and test statistics
- comparing the likelihoods given different model parameters seems pretty natural considering the relationship with the joint probability of the model

Likelihood Ratio Test Statistics

- there are different methods of finding test procedures and test statistics
- comparing the likelihoods given different model parameters seems pretty natural considering the relationship with the joint probability of the model
- let's consider a simple experiment that measures a number of counts N from a process that according to your theoretical colleague can have an expectation $\lambda = 1$ or $\lambda = 5$. The probability function is $N \sim \text{Poisson}(n; \lambda)$. After observing $n = 5$ we test:
 - $H_0 : \lambda = 1$
 - $H_1 : \lambda = 5$
- $T(n = 5) = \frac{\mathcal{L}(\lambda = 1, N = 5)}{\mathcal{L}(\lambda = 5, N = 5)}$



Likelihood Ratio Test Statistics

- let's consider a simple experiment that measures a number of counts N from a process that according to your theoretical colleague can have an expectation $\lambda = 1$ or $\lambda = 5$. The probability function is $N \sim \text{Poisson}(n; \lambda)$. After observing $n = 5$ we test:
 - $H_0 : \lambda = 1$
 - $H_1 : \lambda = 5$

$$T(n = 5) = \frac{\mathcal{L}(\lambda = 1, N = 5)}{\mathcal{L}(\lambda = 5, N = 5)}$$

Likelihood Ratio Test Statistics

- let's consider a simple experiment that measures a number of counts N from a process that according to your theoretical colleague can have an expectation $\lambda = 1$ or $\lambda = 5$. The probability function is $N \sim \text{Poisson}(n; \lambda)$. After observing $n = 5$ we test:
 - $H_0 : \lambda = 1$
 - $H_1 : \lambda = 5$

$$T(n = 5) = \frac{\mathcal{L}(\lambda = 1, N = 5)}{\mathcal{L}(\lambda = 5, N = 5)}$$

$$t(n = 5) = -2 \log T = -2 \log \frac{\mathcal{L}(\lambda = 1, N = 5)}{\mathcal{L}(\lambda = 5, N = 5)}$$

Likelihood Ratio Test Statistics

- let's consider a simple experiment that measures a number of counts N from a process that according to your theoretical colleague can have an expectation $\lambda = 1$ or $\lambda = 5$. The probability function is $N \sim \text{Poisson}(n; \lambda)$. After observing $n = 5$ we test:
 - $H_0 : \lambda = 1$
 - $H_1 : \lambda = 5$

$$T(n = 5) = \frac{\mathcal{L}(\lambda = 1, N = 5)}{\mathcal{L}(\lambda = 5, N = 5)}$$

$$t(n = 5) = -2 \log T = -2 \log \frac{\mathcal{L}(\lambda = 1, N = 5)}{\mathcal{L}(\lambda = 5, N = 5)}$$

$$t(n = 5) = -2 \log \mathcal{L}(\lambda = 1, N = 5) - (-2 \log \mathcal{L}(\lambda = 5, N = 5))$$

LRT: counting with known signal and background

Let's make the previous example a bit more realistic:

- The experiment measures a number of counts

$$N \sim \text{Poisson}(n; \lambda_s + \lambda_b)$$

- λ_s is the signal expectation
- λ_b is the background expectation
- $\mathcal{L}(\lambda_s, \lambda_b; n) = e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n / n!$

LRT: counting with known signal and background

Let's make the previous example a bit more realistic:

- The experiment measures a number of counts
 $N \sim \text{Poisson}(n; \lambda_s + \lambda_b)$
- λ_s is the signal expectation
- λ_b is the background expectation
- $\mathcal{L}(\lambda_s, \lambda_b; n) = e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n / n!$
- hypotheses (both signal and background expectations are fixed):
 - $H_0 : \lambda_s = 0, \lambda_b = 5$
 - $H_1 : \lambda_s = 3, \lambda_b = 5$

LRT: counting with known signal and background

Let's make the previous example a bit more realistic:

- The experiment measures a number of counts
 $N \sim \text{Poisson}(n; \lambda_s + \lambda_b)$
- λ_s is the signal expectation
- λ_b is the background expectation
- $\mathcal{L}(\lambda_s, \lambda_b; n) = e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n / n!$
- hypotheses (both signal and background expectations are fixed):
 - $H_0 : \lambda_s = 0, \lambda_b = 5$
 - $H_1 : \lambda_s = 3, \lambda_b = 5$
- $$\begin{aligned} t(n) &= -2 \log \mathcal{L}(\lambda_s = 0, \lambda_b = 5; n) - (-2 \log \mathcal{L}(\lambda_s = 3, \lambda_b = 5; n)) \\ &= -2(-5 + n \log(5) - \log(n!)) + 2(-8 + n \log(8) - \log(n!)) \\ &= -2n(\log(5) - \log(8)) - 6 \end{aligned}$$

LRT: counting with known signal and background

Let's make the previous example a bit more realistic:

- The experiment measures a number of counts
 $N \sim \text{Poisson}(n; \lambda_s + \lambda_b)$
- λ_s is the signal expectation
- λ_b is the background expectation
- $\mathcal{L}(\lambda_s, \lambda_b; n) = e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n / n!$
- hypotheses (both signal and background expectations are fixed):
 - $H_0 : \lambda_s = 0, \lambda_b = 5$
 - $H_1 : \lambda_s = 3, \lambda_b = 5$
- $$t(n) = -2 \log \mathcal{L}(\lambda_s = 0, \lambda_b = 5; n) - (-2 \log \mathcal{L}(\lambda_s = 3, \lambda_b = 5; n))$$
$$= -2(-5 + n \log(5) - \log(n!)) + 2(-8 + n \log(8) - \log(n!))$$
$$= -2n(\log(5) - \log(8)) - 6$$

```
In [20]: def test_statistic_example(counts_obs, lambda_s = 3, lambda_b = 5):  
         l11 = poisson.logpmf(counts_obs, lambda_s + lambda_b)  
         l10 = poisson.logpmf(counts_obs, lambda_b)  
         return -2 * (l10 - l11)  
  
In [26]: xvals = np.arange(10)  
         yvals = test_statistic_example(xvals)  
         plt.plot(xvals, yvals, "bo", label='$\lambda_s=3, \lambda_b=5$')  
         plt.legend(fontsize=16)  
         plt.xlabel("observed counts n", fontsize=18)  
         plt.ylabel("test-statistic t(n)", fontsize=18)  
         plt.xticks(fontsize=16)  
         plt.yticks(fontsize=16)  
         plt.show()
```

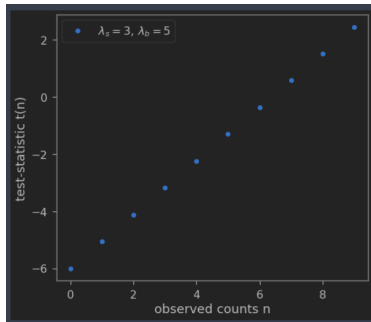
LRT: counting with known signal and background

Let's make the previous example a bit more realistic:

- The experiment measures a number of counts
 $N \sim \text{Poisson}(n; \lambda_s + \lambda_b)$
- λ_s is the signal expectation
- λ_b is the background expectation
- $\mathcal{L}(\lambda_s, \lambda_b; n) = e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n / n!$
- hypotheses (both signal and background expectations are fixed):
 - $H_0 : \lambda_s = 0, \lambda_b = 5$
 - $H_1 : \lambda_s = 3, \lambda_b = 5$
- $t(n) = -2 \log \mathcal{L}(\lambda_s = 0, \lambda_b = 5; n) - (-2 \log \mathcal{L}(\lambda_s = 3, \lambda_b = 5; n))$
 $= -2(-5 + n \log(5) - \log(n!)) + 2(-8 + n \log(8) - \log(n!))$
 $= -2n(\log(5) - \log(8)) - 6$

```
In [20]: def test_statistic_example(counts_obs, lambda_s = 3, lambda_b = 5):  
        ll1 = poisson.logpmf(counts_obs, lambda_s + lambda_b)  
        ll0 = poisson.logpmf(counts_obs, lambda_b)  
        return -2 * (ll0 - ll1)
```

```
In [26]: xvals = np.arange(10)  
        yvals = test_statistic_example(xvals)  
        plt.plot(xvals, yvals, "bo", label='$\lambda_s=3, \lambda_b=5$')  
        plt.legend(fontsize=16)  
        plt.xlabel("observed counts n", fontsize=18)  
        plt.ylabel("test-statistic t(n)", fontsize=18)  
        plt.xticks(fontsize=16)  
        plt.yticks(fontsize=16)  
        plt.show()
```




```

In [40]: # generate background counts
lambda_b = 5
lambda_s = 3

np.random.seed(0)
counts_bkg = np.random.poisson(lambda_b, 10000)
counts_sig_bkg = np.random.poisson(lambda_s + lambda_b, 10000)

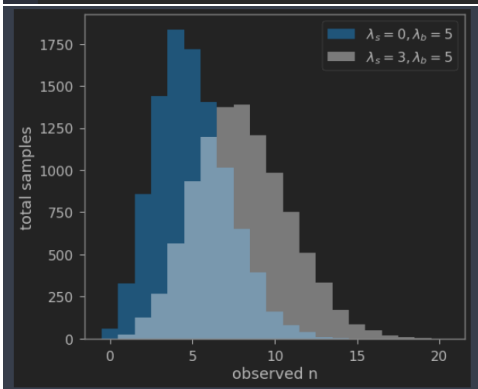
edges = np.linspace(-0.5, 20.5, 22)
plt.hist(counts_bkg, bins=edges, alpha=0.6, color='tab:blue', label='$\lambda_s=0, \lambda_b=5$')
plt.hist(counts_sig_bkg, bins=edges, alpha=0.4, color='white', label='$\lambda_s=3, \lambda_b=5$')

plt.ylabel("total samples", fontsize=18)
plt.xlabel("observed n", fontsize=18)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)

plt.legend(fontsize=16)

plt.show()

```



```

In [42]: tstats_bkg = test_statistic_example(counts_bkg)
tstats_sig_bkg = test_statistic_example(counts_sig_bkg)

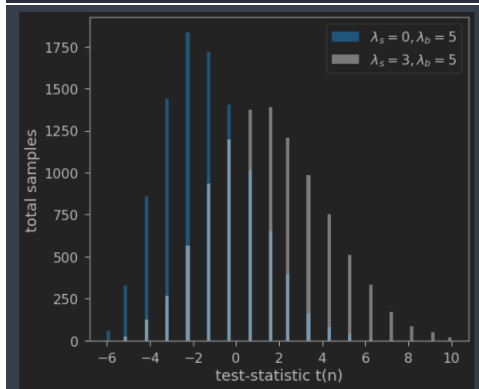
edges = np.linspace(-6, 10, 101)
plt.hist(tstats_bkg, bins=edges, alpha=0.6, color='tab:blue', label='$\lambda_s=0, \lambda_b=5$')
plt.hist(tstats_sig_bkg, bins=edges, alpha=0.4, color='white', label='$\lambda_s=3, \lambda_b=5$')

plt.ylabel("total samples", fontsize=18)
plt.xlabel("test-statistic t(n)", fontsize=18)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)

plt.legend(fontsize=16)

plt.show()

```



LRT: counting with unknown signal and known background

- The experiment still measures a number of counts
 $N \sim \text{Poisson}(n; \lambda_s + \lambda_b)$ where λ_s is the signal expectation and λ_b is the background expectation
- $\mathcal{L}(\lambda_s, \lambda_b; n) = e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n / n!$
- hypotheses (no signal vs some signal):
 - $H_0 : \lambda_s = 0, \lambda_b = 5$
 - $H_1 : \lambda_s \geq 0, \lambda_b = 5$

LRT: counting with unknown signal and known background

- The experiment still measures a number of counts
 $N \sim \text{Poisson}(n; \lambda_s + \lambda_b)$ where λ_s is the signal expectation and λ_b is the background expectation
- $\mathcal{L}(\lambda_s, \lambda_b; n) = e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n / n!$
- hypotheses (no signal vs some signal):
 - $H_0 : \lambda_s = 0, \lambda_b = 5$
 - $H_1 : \lambda_s \geq 0, \lambda_b = 5$
- $t(n) = -2 \log \mathcal{L}(\lambda_s = 0, \lambda_b = 5; n) - (-2 \log \mathcal{L}(\lambda_s = \hat{\lambda}_s, \lambda_b = 5; n))$

$$\hat{\lambda}_s = \begin{cases} 0 & \text{if } n \leq \lambda_b \\ n - \lambda_b & \text{if } n > \lambda_b \end{cases}$$

$$t(n) = \begin{cases} 0 & \text{if } n \leq \lambda_b \\ 2\lambda_b - 2n(\log \lambda_b + 1 - \log n) & \text{if } n > \lambda_b \end{cases}$$

LRT: counting with unknown signal and known background

- The experiment still measures a number of counts
 $N \sim \text{Poisson}(n; \lambda_s + \lambda_b)$ where λ_s is the signal expectation and λ_b is the background expectation

- $\mathcal{L}(\lambda_s, \lambda_b; n) = e^{-(\lambda_s + \lambda_b)} (\lambda_s + \lambda_b)^n / n!$

- hypotheses (no signal vs some signal):

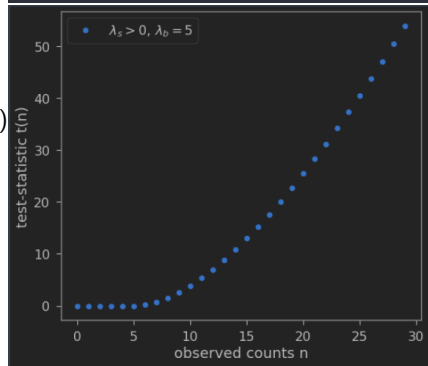
- $H_0 : \lambda_s = 0, \lambda_b = 5$
- $H_1 : \lambda_s \geq 0, \lambda_b = 5$

- $t(n) = -2 \log \mathcal{L}(\lambda_s = 0, \lambda_b = 5; n) - (-2 \log \mathcal{L}(\lambda_s = \hat{\lambda}_s, \lambda_b = 5; n))$

$$\hat{\lambda}_s = \begin{cases} 0 & \text{if } n \leq \lambda_b \\ n - \lambda_b & \text{if } n > \lambda_b \end{cases}$$

$$t(n) = \begin{cases} 0 & \text{if } n \leq \lambda_b \\ 2\lambda_b - 2n(\log \lambda_b + 1 - \log n) & \text{if } n > \lambda_b \end{cases}$$

```
def test_statistic_one_sided(counts, lambda_b = 5):  
    test_statistics = np.zeros(len(counts))  
    idx = counts > lambda_b  
  
    test_statistics[idx] = 2 * lambda_b  
    test_statistics[idx] += 2 * counts[idx] * (np.log(lambda_b) + 1 - np.log(counts[idx]))  
  
    return test_statistics  
  
xvals = np.arange(0, 30)  
yvals = test_statistic_one_sided(xvals)  
  
plt.plot(xvals, yvals, "bo", label=f"$\lambda_s > 0, \lambda_b = 5$")  
plt.legend(fontsize=16)  
plt.xlabel("observed counts n", fontsize=18)  
plt.ylabel("test-statistic t(n)", fontsize=18)  
plt.xticks(fontsize=16)  
plt.yticks(fontsize=16)  
plt.show()
```



```

# generate background counts
lambda_b = 5
lambda_s = 3
lambda_s2 = 10

np.random.seed(0)
counts_bkg = np.random.poisson(lambda_b, 10000)
counts_sig_bkg = np.random.poisson(lambda_s + lambda_b, 10000)
counts_sig_bkg2 = np.random.poisson(lambda_s2 + lambda_b, 10000)

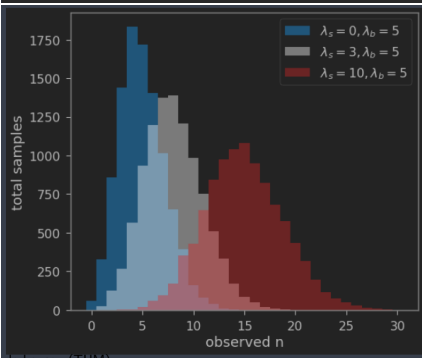
edges = np.linspace(-0.5, 30.5, 32)
plt.hist(counts_bkg, bins=edges, alpha=0.6, color='tab:blue', label='$\lambda_s=0, \lambda_b=5$')
plt.hist(counts_sig_bkg, bins=edges, alpha=0.4, color='white', label='$\lambda_s=3, \lambda_b=5$')
plt.hist(counts_sig_bkg2, bins=edges, alpha=0.4, color='tab:red', label='$\lambda_s=10, \lambda_b=5$')

plt.ylabel("total samples", fontsize=18)
plt.xlabel("observed n", fontsize=18)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)

plt.legend(fontsize=16)

plt.show()

```



```
# generate background counts
lambda_b = 5
lambda_s = 3
lambda_s2 = 10

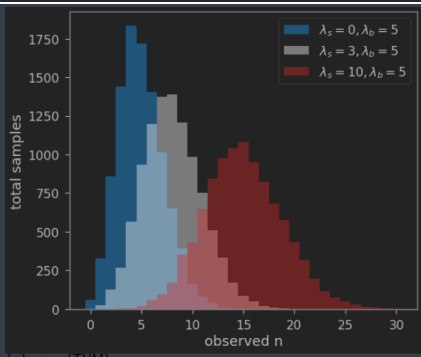
np.random.seed(0)
counts_bkg = np.random.poisson(lambda_b, 10000)
counts_sig_bkg = np.random.poisson(lambda_s + lambda_b, 10000)
counts_sig_bkg2 = np.random.poisson(lambda_s2 + lambda_b, 10000)

edges = np.linspace(-0.5, 30.5, 32)
plt.hist(counts_bkg, bins=edges, alpha=0.6, color='tab:blue', label=r'$\lambda_s=0, \lambda_b=5$')
plt.hist(counts_sig_bkg, bins=edges, alpha=0.4, color='white', label=r'$\lambda_s=3, \lambda_b=5$')
plt.hist(counts_sig_bkg2, bins=edges, alpha=0.4, color='tab:red', label=r'$\lambda_s=10, \lambda_b=5$')

plt.ylabel("total samples", fontsize=18)
plt.xlabel("observed n", fontsize=18)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)

plt.legend(fontsize=16)

plt.show()
```

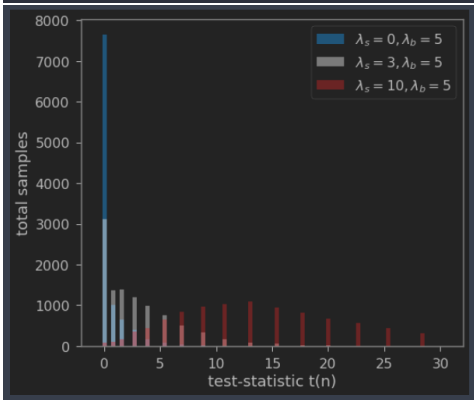


```
edges = np.linspace(-0.5, 30.5, 32)
plt.hist(test_statistic_one_sided(counts_bkg), bins=edges, alpha=0.6, color='tab:blue', label=f)
plt.hist(test_statistic_one_sided(counts_sig_bkg), bins=edges, alpha=0.4, color='white', label=f)
plt.hist(test_statistic_one_sided(counts_sig_bkg2), bins=edges, alpha=0.4, color='tab:red', label=f)

plt.ylabel("total samples", fontsize=18)
plt.xlabel("test-statistic t(n)", fontsize=18)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)

plt.legend(fontsize=16)

plt.show()
```



LRT formal definition

In general the two hypotheses tested are expressed as $H_0 : \vec{\theta} \in \Theta_0$ and $H_1 : \vec{\theta} \in \Theta_0^c$ where $\vec{\theta} = \{\theta_1, \dots, \theta_k\}$ is a vector of model parameters, Θ_0 is some subset of the parameter space allowed for $\vec{\theta}$ and Θ_0^c is its complement. In this case the test statistic can be expressed as

$$t(x) = -2 \log \frac{\mathcal{L}(\hat{\theta}_0; x)}{\mathcal{L}(\hat{\theta}; x)}$$

where $\hat{\theta}$ is the MLE for $\vec{\theta} \in \Theta$ and $\hat{\theta}_0$ is the MLE for $\vec{\theta} \in \Theta_0$.

LRT formal definition

In general the two hypotheses tested are expressed as $H_0 : \vec{\theta} \in \Theta_0$ and $H_1 : \vec{\theta} \in \Theta_0^c$ where $\vec{\theta} = \{\theta_1, \dots, \theta_k\}$ is a vector of model parameters, Θ_0 is some subset of the parameter space allowed for $\vec{\theta}$ and Θ_0^c is its complement. In this case the test statistic can be expressed as

$$t(x) = -2 \log \frac{\mathcal{L}(\hat{\theta}_0; x)}{\mathcal{L}(\hat{\theta}; x)}$$

where $\hat{\theta}$ is the MLE for $\vec{\theta} \in \Theta$ and $\hat{\theta}_0$ is the MLE for $\vec{\theta} \in \Theta_0$.

Considering that the MLE is the value of the parameter that maximized the likelihood in the allowed parameter space, the LRT can be in general expressed as

$$t(x) = -2 \log \frac{\sup_{\vec{\theta} \in \Theta_0} \mathcal{L}(\vec{\theta}; x)}{\sup_{\vec{\theta} \in \Theta} \mathcal{L}(\vec{\theta}; x)}$$

Examples

In our example with the unknown signal and known background:

- $\vec{\theta} = \{\lambda_s\}$
- $\Theta_0 : \{\lambda_s : \lambda_s = 0\}$
- $\Theta : \{\lambda_s : \lambda_s \geq 0\}$

Examples

In our example with the unknown signal and known background:

- $\vec{\theta} = \{\lambda_s\}$
- $\Theta_0 : \{\lambda_s : \lambda_s = 0\}$
- $\Theta : \{\lambda_s : \lambda_s \geq 0\}$

$$\begin{aligned} t(n) &= -2 \log \mathcal{L}(\lambda_s = 0, \lambda_b = 5; n) - (-2 \log \mathcal{L}(\lambda_s = \hat{\lambda}_s, \lambda_b = 5; n)) \\ &= -2 \log \frac{\sup_{\lambda_s=0} \mathcal{L}(\lambda_s, \lambda_b = 5; n)}{\sup_{\lambda_s \geq 0} \mathcal{L}(\lambda_s, \lambda_b = 5; n)} \end{aligned}$$

This generalizes to the case in which there are more parameters. E.g. if the background expectation is considered as not known and becomes a nuisance parameters:

- $\vec{\theta} = \{\lambda_s, \lambda_b\}$
- $\Theta_0 : \{\lambda_s, \lambda_b : \lambda_s = 0, \lambda_b \geq 0\}$
- $\Theta : \{\lambda_s, \lambda_b : \lambda_s \geq 0, \lambda_b \geq 0\}$

$$t(n) = -2 \log \frac{\sup_{\lambda_s=0, \lambda_b \geq 0} \mathcal{L}(\lambda_s, \lambda_b; n)}{\sup_{\lambda_s \geq 0, \lambda_b \geq 0} \mathcal{L}(\lambda_s, \lambda_b; n)}$$

The definition of the test statistic extended to nuisance parameters is also called **profile likelihood ratio**

Questions?

How to define the acceptance/rejection regions for the test – 1

Given a LRT statistic:

$$t(x) = -2 \log \frac{\sup_{\vec{\theta} \in \Theta_0} \mathcal{L}(\vec{\theta}; x)}{\sup_{\vec{\theta} \in \Theta} \mathcal{L}(\vec{\theta}; x)}$$

the acceptance region is defined as $\{\vec{x} : t(\vec{x}) \leq c\}$ where c is any positive real number.

How to define the acceptance/rejection regions for the test – 1

Given a LRT statistic:

$$t(x) = -2 \log \frac{\sup_{\vec{\theta} \in \Theta_0} \mathcal{L}(\vec{\theta}; x)}{\sup_{\vec{\theta} \in \Theta} \mathcal{L}(\vec{\theta}; x)}$$

the acceptance region is defined as $\{\vec{x} : t(\vec{x}) \leq c\}$ where c is any positive real number.

But how to choose the value of c ? For any chosen value:

- the test will have some probability to reject H_0 even if it true (type I error)
- the test will have some probability to accept H_0 even if it is false (type II error)

How to define the acceptance/rejection regions for the test – 1

Given a LRT statistic:

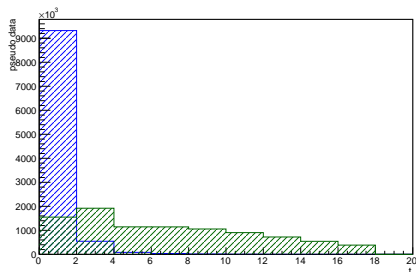
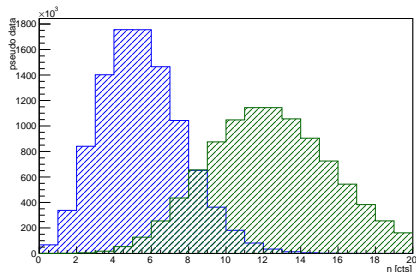
$$t(x) = -2 \log \frac{\sup_{\vec{\theta} \in \Theta_0} \mathcal{L}(\vec{\theta}; x)}{\sup_{\vec{\theta} \in \Theta} \mathcal{L}(\vec{\theta}; x)}$$

the acceptance region is defined as $\{\vec{x} : t(\vec{x}) \leq c\}$ where c is any positive real number.

But how to choose the value of c ? For any chosen value:

- the test will have some probability to reject H_0 even if it true (type I error)
- the test will have some probability to accept H_0 even if it is false (type II error)

Typically we fix the probability for a type I error. Test statistic based on likelihood ratios have the property to minimize the probability for type II errors in a large class of problems



How to define the acceptance/rejection region for the test – 2

The probability of rejecting the null hypothesis when it is true is indicated with α and it is typically small e.g. 10% , 5%, 1%

How to define the acceptance/rejection region for the test – 2

The probability of rejecting the null hypothesis when it is true is indicated with α and it is typically small e.g. 10% , 5%, 1%

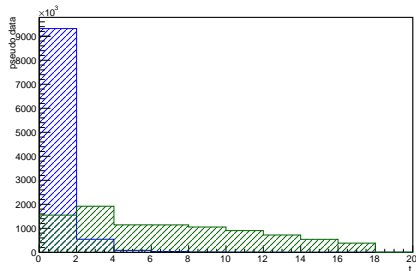
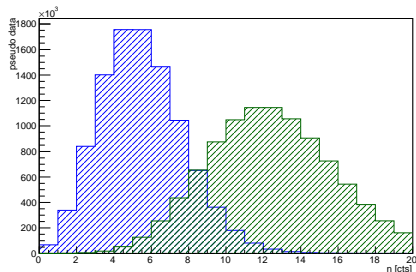
Physicist use to call $1-\alpha$ the confidence level (C.L.) of a test.
A test with type I error probability of 5% is called a 95% CL test.

How to define the acceptance/rejection region for the test – 2

The probability of rejecting the null hypothesis when it is true is indicated with α and it is typically small e.g. 10% , 5%, 1%

Physicist use to call $1-\alpha$ the confidence level (C.L.) of a test. A test with type I error probability of 5% is called a 95% CL test.

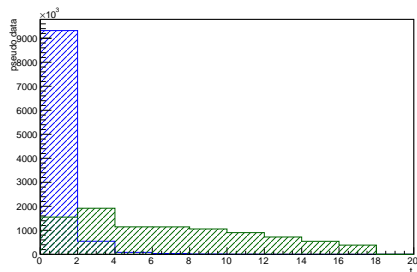
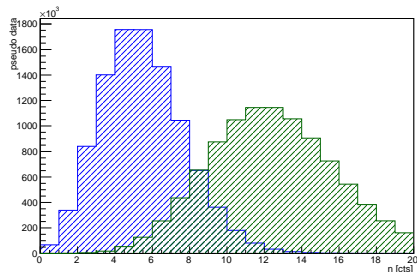
To map values of α in values of c , the probability distribution of the test statistic assuming the null hypothesis is needed. The probability distribution can be created by random sampling!



How to define the acceptance/rejection region for the test – 3

How to build the probability distribution of the statistic assuming the null hypothesis:

- 1) create a pseudo-data set by sampling a value for each random variables of the model according to the joint probability distribution for the null hypothesis parameters
- 2) compute and store the value of the test statistic for the pseudo-data set
- 3) repeat step 1 and 2 multiple times
- 4) plot a normalized histogram that will give the probability of observing a particular test statistic value given the model (the test statistic is actually a random variable by itself)



Summary of hypothesis testing

- 1) define the null and alternative hypothesis

Summary of hypothesis testing

- 1) define the null and alternative hypothesis
- 2) once the two hypotheses are defined, a LRT statistic emerges naturally by maximizing the likelihood in the parameter space allowed by the two hypotheses. The maximization of the likelihood (i.e. minimize of the negative log likelihood) can be done using numerical techniques

Summary of hypothesis testing

- 1) define the null and alternative hypothesis
- 2) once the two hypotheses are defined, a LRT statistic emerges naturally by maximizing the likelihood in the parameter space allowed by the two hypotheses. The maximization of the likelihood (i.e. minimize of the negative log likelihood) can be done using numerical techniques
- 3) construct the probability function of the test statistic assuming the null hypotheses by generating pseudo data and computing the test statistic value for each of them

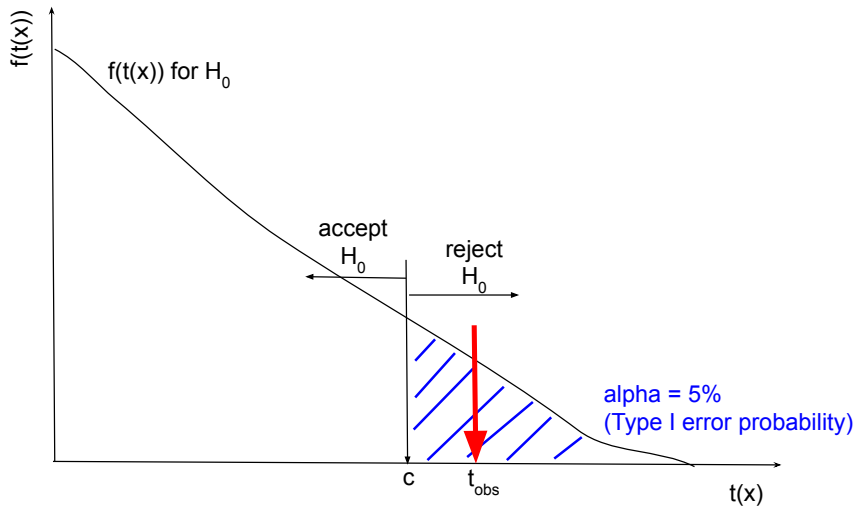
Summary of hypothesis testing

- 1) define the null and alternative hypothesis
- 2) once the two hypotheses are defined, a LRT statistic emerges naturally by maximizing the likelihood in the parameter space allowed by the two hypotheses. The maximization of the likelihood (i.e. minimize of the negative log likelihood) can be done using numerical techniques
- 3) construct the probability function of the test statistic assuming the null hypotheses by generating pseudo data and computing the test statistic value for each of them
- 4) define the C.L. of the test and calculate the rejection region (hint: `np.percentile`)

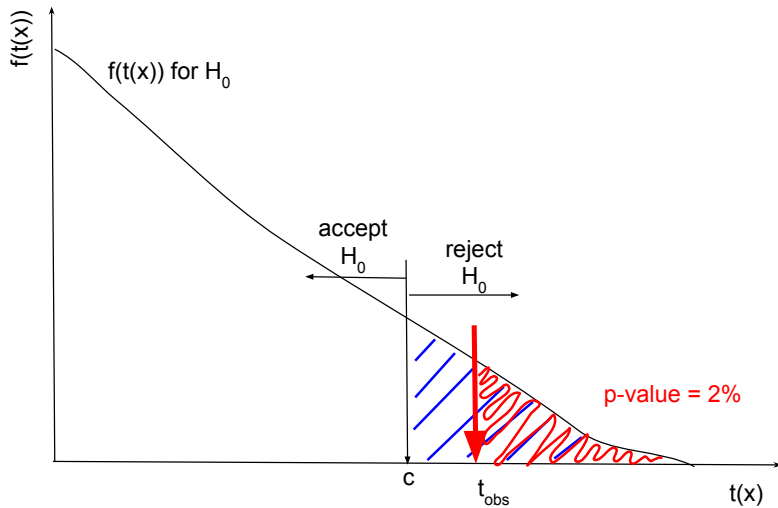
Summary of hypothesis testing

- 1) define the null and alternative hypothesis
- 2) once the two hypotheses are defined, a LRT statistic emerges naturally by maximizing the likelihood in the parameter space allowed by the two hypotheses. The maximization of the likelihood (i.e. minimize of the negative log likelihood) can be done using numerical techniques
- 3) construct the probability function of the test statistic assuming the null hypotheses by generating pseudo data and computing the test statistic value for each of them
- 4) define the C.L. of the test and calculate the rejection region (hint: `np.percentile`)
- 5) take the data from your experiment (i.e. the **observed** data), compute the test statistic on them and check whether its value is in the rejection or acceptance region

p-values



p-values



Questions?