Deutsches Elektronen-Synchrotron DESY

Terascale Statistics School 2023 June, 2023

Roman Kogler

Exercise: Poisson Coverage

In the lectures we learned about the coverage probability of uncertainties (short: coverage). The coverage $C(\mu)$ for an unknown parameter μ is defined as the probability that the true value of μ lies within the interval defined by the measured value μ_m and the uncertainties σ_1 and σ_2,

$$\mu_m - \sigma_1 \leq \mu \leq \mu_m + \sigma_2, \tag{1}$$

or in other words, the probability that the true value of μ lies within the interval $[\mu_m - \sigma_1, \mu_m + \sigma_2]$. Our best estimate of the parameter μ is thus given by μ_m.

In this exercise we will calculate the coverage for Poisson distributed data. Assume that we have measured n events. If the data follow a Poisson distribution, it is common to assume that the parameter

$$\mu_m = n. \tag{2}$$

Commonly the uncertainty is obtained from the square-root of the variance of the Poisson distribution,

$$\sigma_{1,2} = \sqrt{V[x]} = \sqrt{\mu_m} = \sqrt{n}. \tag{3}$$

Now investigate how good the coverage of this estimate is by calculating the coverage probability. The code can be found at `https://github.com/rkogler/StatSchool23`.

Open the program `coverage_poisson.cxx` with your favourite editor. You can compile the program using:

```
gcc coverage_poisson.cxx -o coverage_poisson
```

Then run the code by executing the resulting executable `coverage_poisson`.

1. Poisson distribution

   First implement a function that calculates the Poisson probability of obtaining n events for a given value of μ,

   $$p(n, \mu) = \frac{\mu^n e^{-\mu}}{n!}. \tag{4}$$

   The function should take two arguments, n and μ, and return the probability $p(n, \mu)$:

   ```
   double prob(double n, double mu)
   ```

   Hint: You can use the implementation of the Γ-function to calculate $n!$: `std::tgamma(n+1)`

   Check that you have implemented it correctly by comparing some values with reference ones (see for example `http://stattrek.com/online-calculator/poisson.aspx`).

2. Coverage

Calculate the coverage as

$$C(\mu) = \sum_{n \text{ if } \mu \in [a(n), b(n)]} p(n, \mu), \tag{5}$$

where the interval $[a(n), b(n)] = [\mu_m(n) - \sigma_1(n), \mu_m(n) + \sigma_2(n)]$. Implement this in a function which takes μ as its single argument,

```
double Cmu(double mu)
```

The function should return the coverage for a given value of μ. In the `main()` routine, calculate the coverage for some values of μ and record those in the table.

μ	0	1	3	10
$C(\mu)$				

What do you observe and what do you conclude?

3. Plot the coverage

Now create a plot of the coverage as a function of μ. For this, calculate $C(\mu)$ for 1000 values of μ in the range $[0, 20]$. Write out the value pairs μ and $C(\mu)$ into a file `cmu.txt`. You can do this by using

```
std::ofstream fout("cmu.txt");
for (...){
  ...
  fout << mu << " ";
  fout << coverage << " " << std::endl;
}
fout.close();
```

The code is already in the `main()` routine, and you only need to comment it in. This needs the header `#include <fstream>`, which is included already.
Afterwards start `gnuplot` in a terminal and plot the coverage with the command

```
plot "cmu.txt" using 1:2 with points
```

Alternatively, you can use the ROOT macro provided to plot the coverage:

```
root -l plot_cov.C
```

What do you observe?