Statistical Data Analysis
Discussion slides – week 11

- Problem sheet 8
- Software for Gamma Variance Model (errors-on-errors)
- Example from 2019 Exam (Gamma Variance Model)

Gaussian signal on exponential background

Consider a pdf for continuous random variable x, (truncate and renormalize in $0 \leq x \leq x_{\max}$)

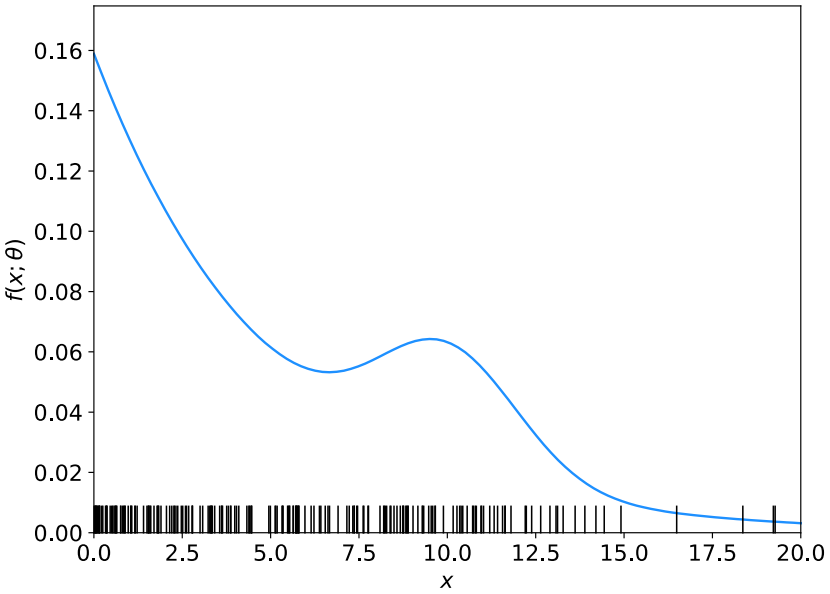
$$f(x;\theta,\xi) = \theta \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} + (1-\theta)\frac{1}{\xi} e^{-x/\xi}$$

θ = parameter of interest ,
gives signal rate.

Depending on context, take ξ, μ, σ as nuisance parameters or fixed.

Generate i.i.d. sample x_1,..., x_n.

Estimate θ (and other params.)



A quick look at mlFit.py

```
# Example of maximum-likelihood fit with iminuit version 2.
# pdf is a mixture of Gaussian (signal) and exponential (background),
# truncated in [xMin,xMax].
# G. Cowan / RHUL Physics / December 2022


import numpy as np
import scipy.stats as stats
from scipy.stats import truncexpon
from scipy.stats import truncnorm
from scipy.stats import chi2
import iminuit
from iminuit import Minuit
import matplotlib.pyplot as plt
from matplotlib import container
plt.rcParams["font.size"] = 14
print("iminuit version:", iminuit.__version__)  # need 2.x


# define pdf and generate data
np.random.seed(seed=1234567)       # fix random seed
theta = 0.2                        # fraction of signal
mu = 10.                           # mean of Gaussian
sigma = 2.                         # std. dev. of Gaussian
xi = 5.                          # mean of exponential
xMin = 0.
xMax = 20.
```

Define the fit function

```
def f(x, par):
    theta   = par[0]
    mu      = par[1]
    sigma   = par[2]
    xi      = par[3]
    fs = stats.truncnorm.pdf(x, a=(xMin-mu)/sigma, b=(xMax-mu)/sigma,
            loc=mu, scale=sigma)
    fb = stats.truncexpon.pdf(x, b=(xMax-xMin)/xi, loc=xMin, scale=xi)
    return theta*fs + (1-theta)*fb
```

Generate the data

```
numVal = 200
xData = np.empty([numVal])
for i in range (numVal):
    r = np.random.uniform();
    if r < theta:
        xData[i] = stats.truncnorm.rvs(a=(xMin-mu)/sigma, b=(xMax-mu)/sigma,
                                       loc=mu, scale=sigma)
    else:
        xData[i] = stats.truncexpon.rvs(b=(xMax-xMin)/xi, loc=xMin, scale=xi)
```

Set up the fit

```
# Function to be minimized is negative log-likelihood
def negLogL(par):
    pdf = f(xData, par)
    return -np.sum(np.log(pdf))


# Initialize Minuit and set up fit:
parin   = np.array([theta, mu, sigma, xi]) # initial values (here = true values)
parname = ['theta', 'mu', 'sigma', 'xi']
parname_latex = [r'$\theta$', r'$\mu$', r'$\sigma$', r'$\xi$']
parstep = np.array([0.1, 1., 1., 1.])      # initial setp sizes
parfix  = [False, True, True, False]       # change these to fix/free parameters
parlim  = [(0.,1), (None, None), (0., None), (0., None)]    # set limits
m = Minuit(negLogL, parin, name=parname)
m.errors = parstep
m.fixed = parfix
m.limits = parlim
m.errordef = 0.5                    # errors from lnL = lnLmax - 0.5
```

Do the fit, get errors, extract results

```
# Do the fit, get errors, extract results
m.migrad()                          # minimize -logL
MLE = m.values                      # max-likelihood estimates
sigmaMLE = m.errors                 # standard deviations
cov = m.covariance                  # covariance matrix
rho = m.covariance.correlation()            # correlation coeffs.

print(r"par index, name, estimate, standard deviation:")
for i in range(m.npar):
    if not m.fixed[i]:
        print("{:4d}".format(i), "{:<10s}".format(m.parameters[i]), " = ",
         "{:.6f}".format(MLE[i]), " +/- ", "{:.6f}".format(sigmaMLE[i]))

print()
print(r"free par indices, covariance, correlation coeff.:")
for i in range(m.npar):
    if not(m.fixed[i]):
        for j in range(m.npar):
            if not(m.fixed[j]):
                print(i, j, "{:.6f}".format(cov[i,j]),
            "{:.6f}".format(rho[i,j]))
```

Make some plots...

Comment on the $\ln L = \ln L_{\max} - ½$ contour

In the lectures, we saw that the standard deviations of fitted parameters are found from the tangent lines (planes) to the contour

$$\ln L = \ln L_{\max} - \frac{1}{2}$$

A similar procedure can be used to find a confidence region in the parameter space that will cover the true parameter with probability CL = 1 – α (the "confidence level"). This uses the contour

$$\ln L = \ln L_{\max} - \frac{1}{2} F_{\chi^2}^{-1}(1-\alpha; N) \ , \qquad N = \text{number of parameters}$$

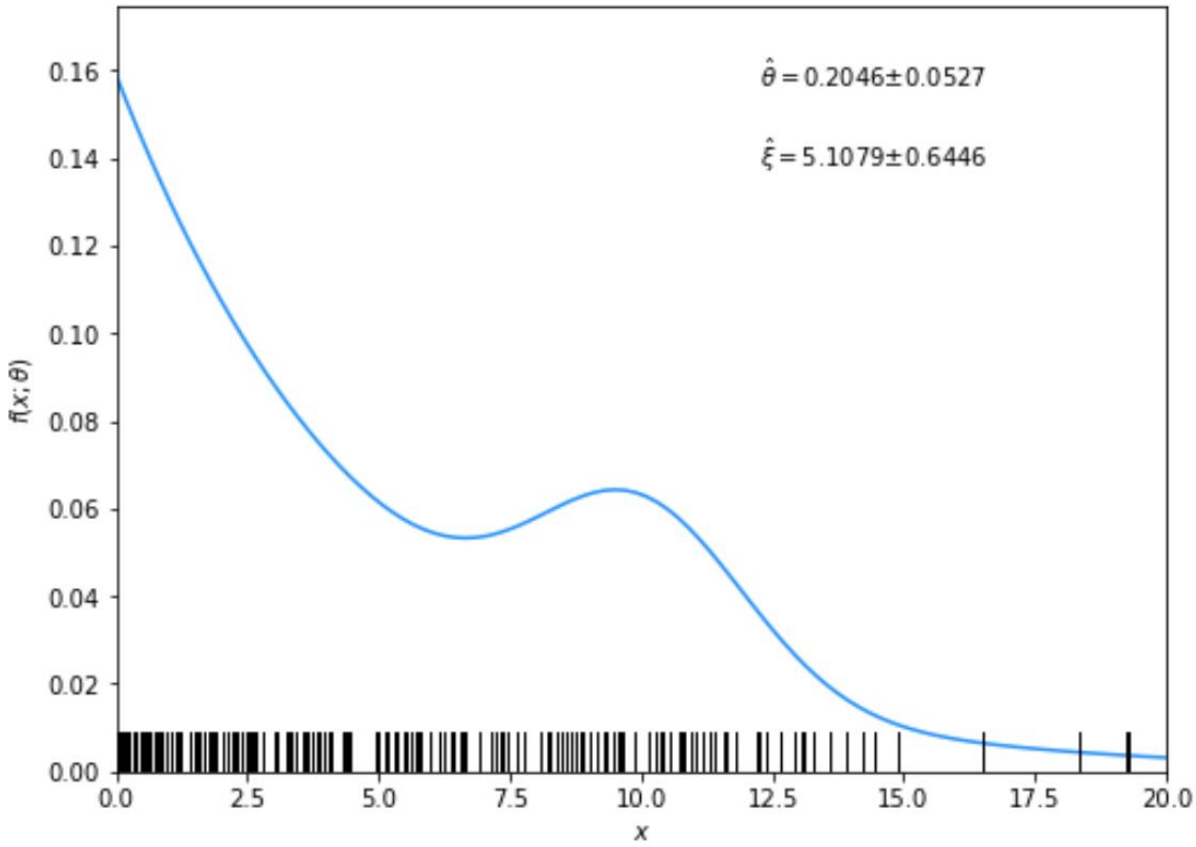
If you want the contour $\ln L = \ln L_{\max} - ½$ in iminuit, you need to choose CL (= 1 – α) such that $F_{\chi 2}^{-1}(1-\alpha, N) = 1$, i.e.,

$$\mathrm{CL} = F_{\chi^2}(1; N) = \texttt{stats.chi2.cdf(1.,N)}$$

Solutions to exercises

1a) Running the program mlFit.py produces the following plots:

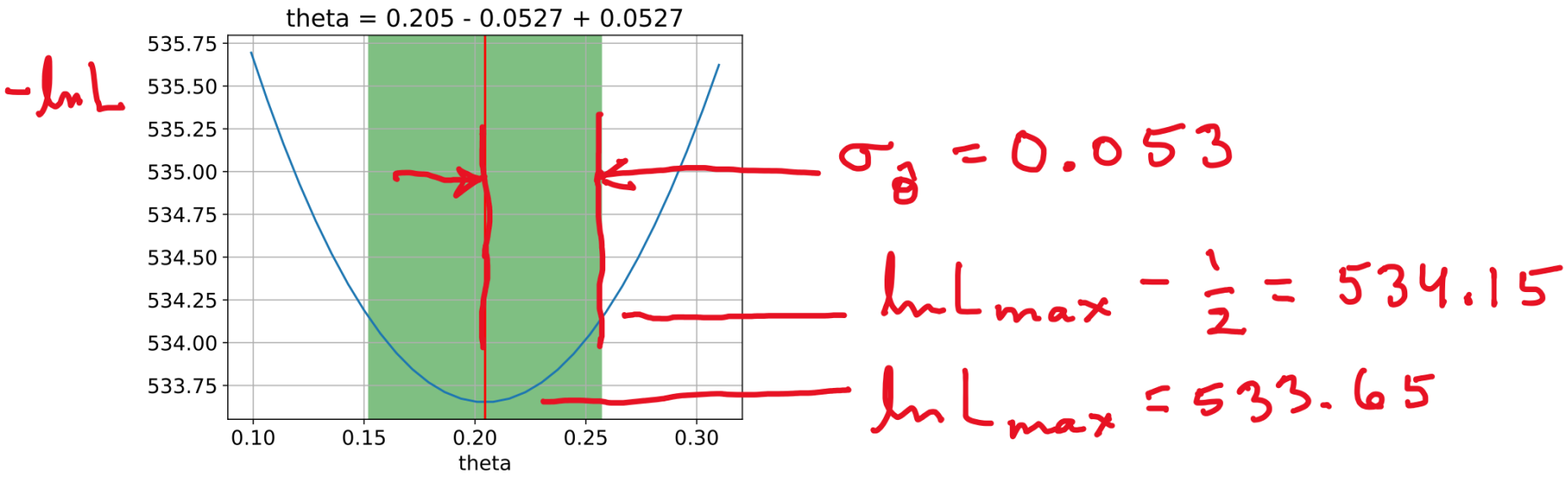
A fit of the pdf:



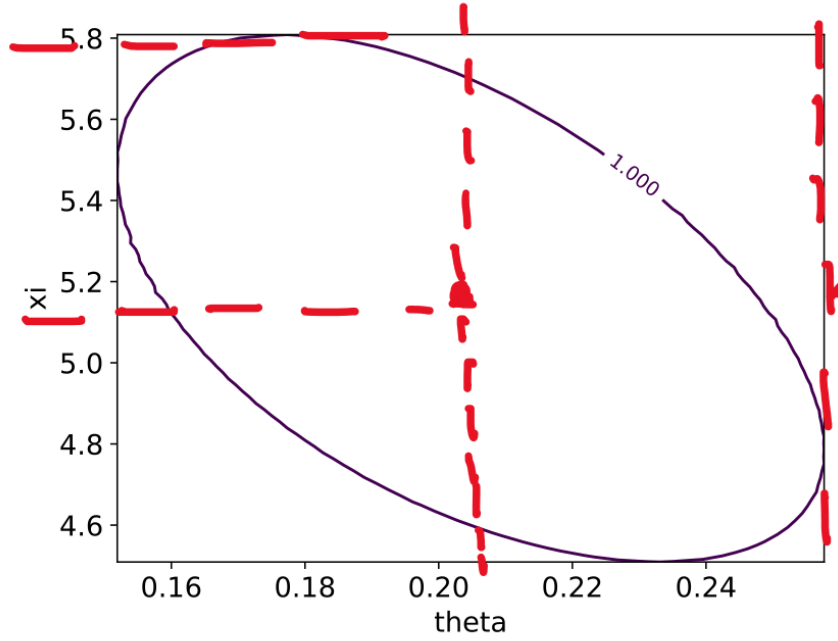
From running program:

```
par index, name, estimate, standard deviation:
    0 theta        =   0.204551   +/-   0.052736
    3 xi           =   5.107878   +/-   0.644563
```

A scan of -lnL versus theta:



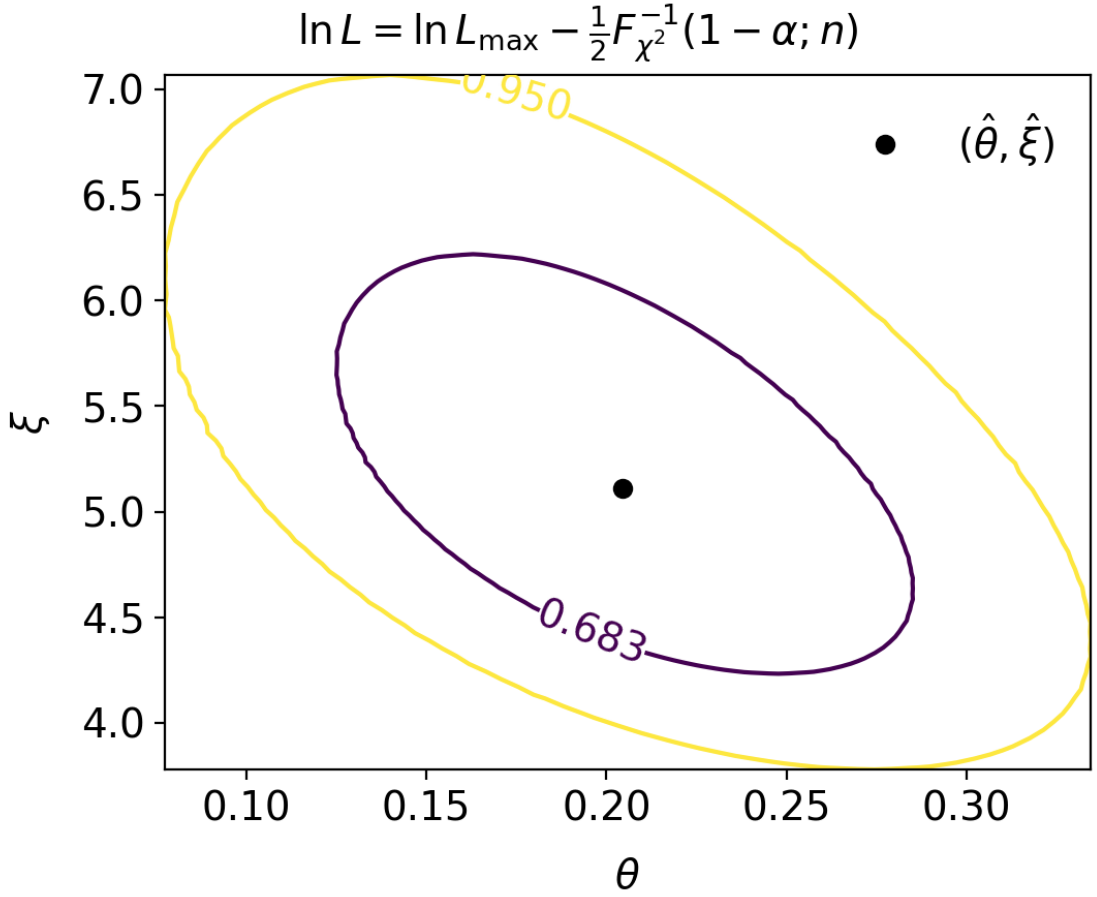
A contour of lnL = lnLmax – ½:



Confidence regions at CL = 68.3% and 95%

In iminuit v2, user can set $\mathrm{CL} = 1 - \alpha$

```
m.draw_mncontour('theta', 'xi', cl=[0.683, 0.95], size=200)
```



1b) Assume i.i.d. data sample, so $L(\boldsymbol{\theta}) = P(\mathbf{x}|\boldsymbol{\theta}) = \prod_{k=1}^{n} P(x_k|\boldsymbol{\theta})$

Assume inverse covariance from Fisher Information (large sample):

$$V_{ij}^{-1} = -E\left[\frac{\partial^2 \ln L}{\partial\theta_i \partial\theta_j}\right] = -\int \frac{\partial^2 \ln L}{\partial\theta_i \partial\theta_j} P(\mathbf{x}|\boldsymbol{\theta})\, d\mathbf{x}$$

Since $\ln L(\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln P(x_k|\boldsymbol{\theta})$ we find

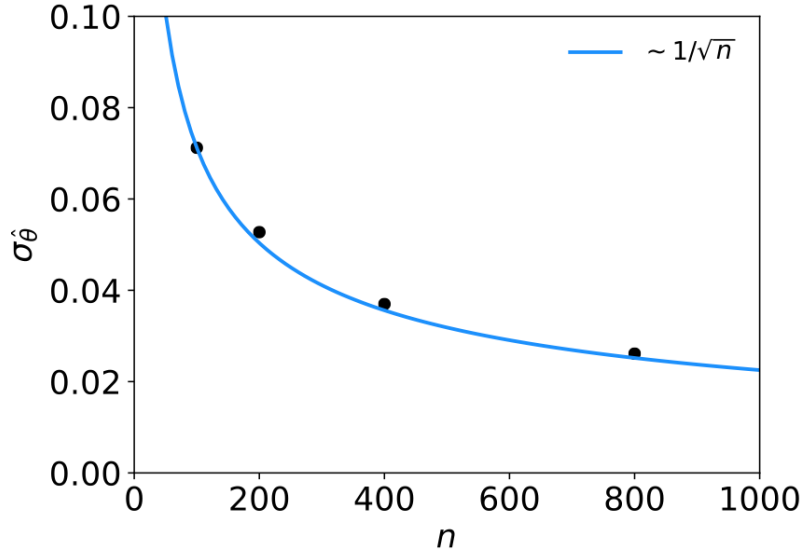
$$V_{ij}^{-1} = -\sum_{k=1}^{n} \int \frac{\partial^2 \ln P(x_k|\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j} P(x_k|\boldsymbol{\theta})\, dx_k = -n \int \frac{\partial^2 \ln P(x|\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j} P(x|\boldsymbol{\theta})\, dx$$

But $V^{-1}V = I$ so if $V^{-1} \propto n$, then $V \propto 1/n$, and so from the square roots of the diagonal elements $\sigma_{\hat{\theta}_i} \propto 1/\sqrt{n}$

1(c) Running mlFit.py with different numbers of events gave:

numVal	thetaHat	sigma_thetaHat
100	0.197218	0.071219
200	0.204551	0.052736
400	0.160808	0.036985
800	0.198224	0.026129

A plot of sigma_thetaHat versus numVal is shown below. The standard deviation of the estimator is seen to decrease as $1/\sqrt{n}$, as expected.



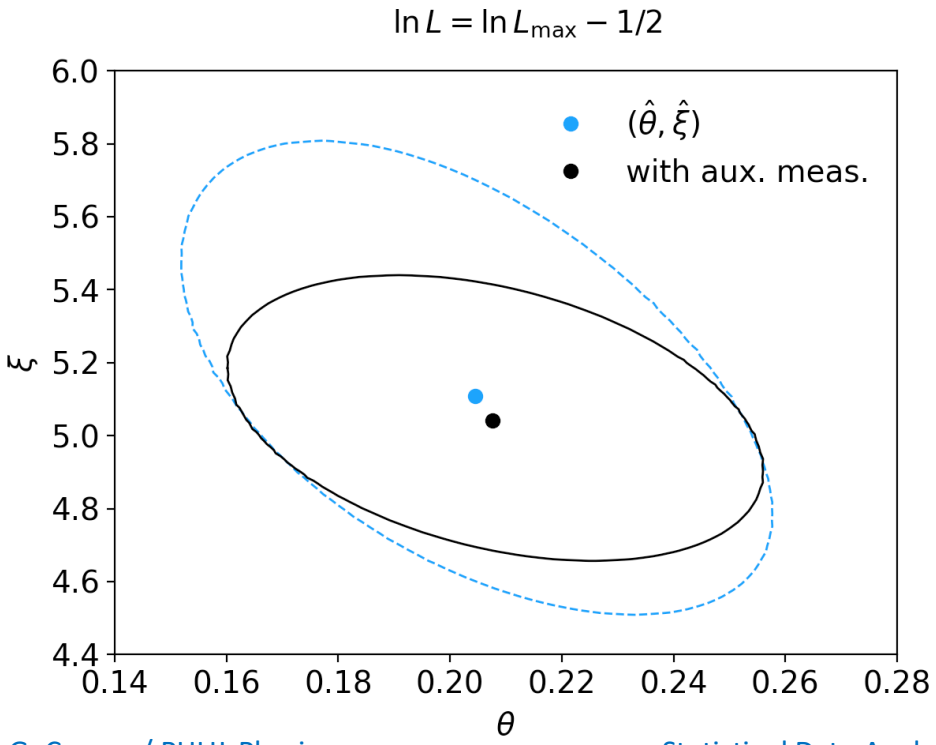
1(d) The results of the fit with different combinations of parameters adjustable are:

Free	Fixed	sigma_thetaHat
theta	mu, sigma, xi	0.044535
theta, xi	mu, sigma	0.052736
theta, xi, sigma	mu	0.064456
theta, xi, sigma, mu	--	0.085786

As can be seen, the standard deviation of the estimator of theta increases when it is fitted simultaneously with an increasing number of other adjustable parameters.

Extra part (not on problem sheet)

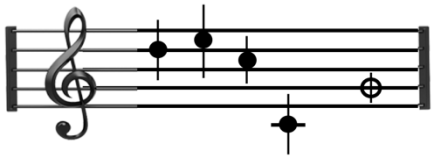
1(e) Consider the case where θ and ξ are adjustable and σ and μ are fixed. Suppose that one has an independent estimate u of the parameter ξ in addition to the $n = 200$ values of x. Treat u as Gaussian distributed with a mean ξ and standard deviation $\sigma_u = 0.5$ and take the observed value $u = 5$. Find the log-likelihood function that includes both the primary measurements $(x_1, \ldots, x_n)$ and the auxiliary measurement u and modify the fitting program accordingly. Investigate how the uncertainties of the MLEs for θ and ξ are affected by including u.



1(e) By including the auxiliary measurement u one uses more information about ξ and thus the covariance ellipse shrinks. This reduces the standard deviations of the MLEs for both ξ and θ.

```
par index, name, estimate, standard deviation:
   0 theta       =  0.204551  +/-  0.052736
   3 xi          =  5.107878  +/-  0.644563
```

Tutorial 4: Student's *t* average



Software: stave.py

The program stave.py implements the Gamma Variance Model (GVM) described in Lecture 3 for averaging N measurements.

For details see G. Cowan, EPJC (2019) 79:133.

In this version the model does not distinguish between statistical and systematic errors.

Confidence interval for the mean μ becomes sensitive to goodness-of-fit (increases if data internally inconsistent).

Estimated mean less sensitive to outliers.

Least Squares vs Gamma Variance Model

Quadratic terms from Least Squares replaced by logarithmic ones:

$$\frac{(y_i - \mu)^2}{\sigma_i^2} \quad \rightarrow \quad \left(1 + \frac{1}{2r_i^2}\right) \ln\left[1 + 2r_i^2 \frac{(y_i - \mu)^2}{v_i}\right]$$

where

y_i = measured value

$v_i = s_i^2$ = estimated variance

r_i = relative uncertainty on estimate of variance

Equivalent to replacing Gauss pdf for measurements by Student's t, number of degrees of freedom $= 1/2r_i^2$

A quick look at stave.py

Set measured values, estimates of std. dev., errors on errors:

```
y = np.array([17., 19., 15., 3.])       # measured values
s = np.array([1.5, 1.5, 1.5, 1.5])      # estimates of std. dev
v = s**2                                # estimates of variances
r = np.array([0.2, 0.2, 0.2, 0.2])      # relative errors on errors
```

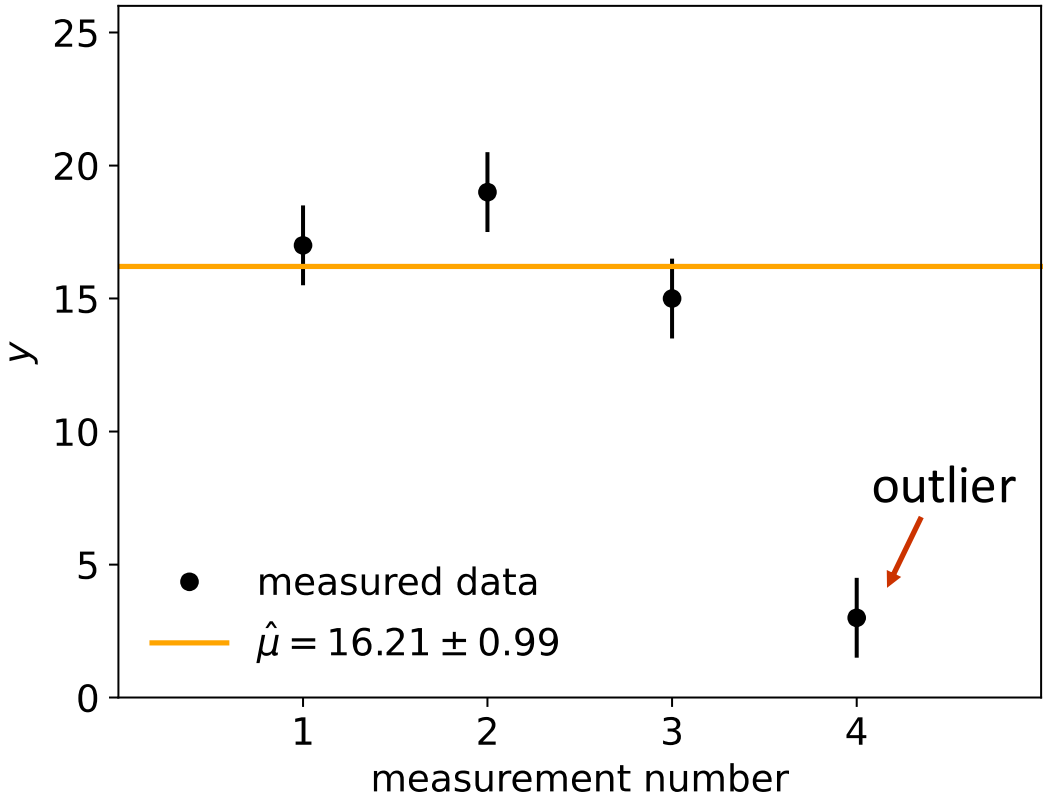
log-likelihood:

```
class NegLogL:

    def __init__(self, y, s, r):
        self.setData(y, s, r)

    def setData(self, y, s, r):
        self.data = y, s, r

    def __call__(self, mu):
        y, s, r = self.data
        v = s ** 2
        lnf = -0.5*(1. + 1./(2.*r**2))*np.log(1. + 2.*(r*(y-mu))**2/v)
        return -np.sum(lnf)
```

Example average with GVM

Suppose four measurements of the parameter μ.

Each reports an estimated standard dev. of $s = 1.5$ and a "relative error on the error" $r = 0.2$.



Suggested exercise:

Experiment with different numbers of measurements, different levels of internal consistency, different values for the std. dev. and error on error.

2019 Exam Question 3(a)

3. Suppose that the outcome of a measurement consists of two independent values, y and v, which follow

$$f_y(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-\mu)^2/2\sigma^2} ,$$

$$f_v(v) = \frac{\left(\frac{\nu}{2\sigma^2}\right)^{\nu/2}}{\Gamma(\nu/2)} v^{\nu/2-1} e^{-\nu v/2\sigma^2} .$$

Here Γ is the Euler gamma function and ν is a known constant. Suppose the parameters μ and σ^2 are both unknown; μ is the parameter of interest and σ^2 is a nuisance parameter.

(a) Show that the log-likelihood function can be written as

$$\ln L(\mu, \sigma^2) = -\frac{1}{2}\left[(1+\nu)\ln\sigma^2 + \frac{(y-\mu)^2}{\sigma^2} + \frac{\nu v}{\sigma^2}\right] + C ,$$

where C represents terms that do not depend on the adjustable parameters μ or σ^2. [6]

2019 Exam Question 3(a)

3(a) [6 marks] The random variables y and v are independent, and therefore the likelihood function is the product of their pdfs:

$$L(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-\mu)^2/2\sigma^2} \frac{\left(\frac{\nu}{2\sigma^2}\right)^{\nu/2}}{\Gamma(\nu/2)} v^{\nu/2-1} e^{-\nu v/2\sigma^2} \; .$$

The log-likelihood is therefore

$$\begin{aligned} \ln L(\mu, \sigma^2) &= -\frac{1}{2}\ln\sigma^2 - \frac{1}{2}\frac{(y-\mu)^2}{\sigma^2} + \alpha \ln\beta - \beta v + C \\ &= -\frac{1}{2}\left[(1+\nu)\ln\sigma^2 + \frac{(y-\mu)^2}{\sigma^2} + \frac{\nu v}{\sigma^2}\right] + C' \; , \end{aligned}$$

where C and C' represent terms that do not depend on μ or σ^2.

2019 Exam Question 3(b)

(b) Show that the maximum-likelihood estimators $\hat{\mu}$ and $\widehat{\sigma^2}$ and the profiled estimator $\widehat{\widehat{\sigma^2}}(\mu)$ are

$$\begin{aligned} \hat{\mu} &= y\,, \\ \widehat{\sigma^2} &= \frac{\nu v}{1+\nu}\,, \\ \widehat{\widehat{\sigma^2}}(\mu) &= \frac{\nu v + (y-\mu)^2}{1+\nu}\,. \end{aligned}$$

[10]

3(b) [10 marks] Setting the derivative of $\ln L$ with respect to σ^2 to zero gives

$$\frac{\partial \ln L}{\partial \sigma^2} = -\frac{1}{2}\left[\frac{1+\nu}{\sigma^2} - \frac{(y-\mu)^2}{(\sigma^2)^2} - \frac{\nu v}{(\sigma^2)^2}\right] = 0\,. \qquad (1)$$

2019 Exam Question 3(b)

Solving for σ^2 as a function of μ gives the profiled value

$$\widehat{\widehat{\sigma^2}}(\mu) = \frac{\nu v + (y-\mu)^2}{1+\nu} \,.$$

Setting the derivative of $\ln L$ with respect to μ to zero gives

$$\frac{\partial \ln L}{\partial \mu} = \frac{y-\mu}{\sigma^2} = 0 \,. \qquad (2)$$

Solving Eqs. (1) and (2) simultaneously for μ and σ^2 give the maximum-likelihood estimators

$$\begin{aligned} \hat{\mu} &= y \,, \\ \widehat{\sigma^2} &= \frac{\nu v}{1+\nu} \,. \end{aligned}$$

2019 Exam Question 3(c)

(c) Suppose we use the statistic

$$t_\mu = -2 \ln \frac{L(\mu, \widehat{\widehat{\sigma^2}}(\mu))}{L(\hat{\mu}, \widehat{\sigma^2})}$$

to test hypothesized values of μ. Using the ingredients found above, show that t_μ is

$$t_\mu = (1+\nu) \ln \left[1 + \frac{1}{\nu} \frac{(y-\mu)^2}{v} \right] .$$

[8]

The parametric form of $f_v(v)$ was chosen such that the expectation value and variance of v are $E[v] = \sigma^2$ and $V[v] = 2\sigma^4/\nu$, i.e., v is an estimate of the variance σ^2. Let $s = \sqrt{v}$ be the corresponding estimator for σ.

3(c) [8 marks] Using the ingredients found in (a) and (b) we can find the profile log-likelihood,

$$\ln L(\mu, \widehat{\widehat{\sigma^2}}) = -\frac{1}{2} \left[(1+\nu) \ln \frac{\nu v + (y-\mu)^2}{1+\nu} + \frac{(y-\mu)^2(1+\nu)}{\nu v + (y-\mu)^2} + \frac{\nu v (1+\nu)}{\nu v + (y-\mu)^2} \right] + C$$

2019 Exam Question 3(c)

and also the maximum of the log-likelihood,

$$\ln L(\hat{\mu}, \widehat{\sigma^2}) = -\frac{1}{2}(1+\nu)\left[1 + \ln\frac{v\nu}{1+\nu}\right] + C \ ,$$

where the constants C are the same in both expressions above. The statistic t_μ is therefore

$$\begin{aligned} t_\mu &= -2\ln\frac{L(\mu, \widehat{\widehat{\sigma^2}})}{L(\hat{\mu}, \widehat{\sigma^2})} = (1+\nu)\ln\frac{\nu v + (y-\mu)^2}{1+\nu}\frac{1+\nu}{\nu v} + \frac{(1+\nu)[(v\nu + (y-\mu)^2]}{v\nu + (y-\mu)^2} - (1+\nu) \\ &= (1+\nu)\ln\left[1 + \frac{1}{\nu}\frac{(y-\mu)^2}{v}\right] \ . \end{aligned}$$

2019 Exam Question 3(d)

(d) Assuming that one approximates $E[v] \approx (E[s])^2$, show using error propagation that the ratio of the standard deviation of s to its mean is

$$\frac{\sigma_s}{E[s]} \approx \frac{1}{\sqrt{2\nu}} \,.$$

[6]

3(d) [6 marks] The estimate s of σ is $s = v^{1/2}$. Using error propagation to find the standard deviation of s gives

$$\sigma_s = \left|\frac{\partial s}{\partial v}\right|_{E[v]} \sigma_v = \left|\frac{1}{2} v^{-1/2}\right|_{E[v]} \sigma_v = \frac{1}{2}\frac{\sigma_v}{\sqrt{E[v]}} \,.$$

Using $E[v] = \sigma^2$, $V[v] = 2\sigma^4/\nu$ and $E[s] \approx \sqrt{E[v]}$ (all given) we find

$$\frac{\sigma_s}{E[s]} \approx \frac{\sigma_s}{\sqrt{E[v]}} = \frac{1}{2}\frac{\sigma_v}{E[v]} = \frac{1}{2}\sqrt{\frac{2}{\nu}}\sigma^2 \frac{1}{\sigma^2} = \frac{1}{\sqrt{2\nu}} \,.$$

2019 Exam Question 3(e)

(e) Starting from the pdf $f_v(v)$ given above, find the pdf for s in terms of the parameters ν and σ^2. [5]

3(e) [5 marks] The pdf of s is

$$g(s) = \left|\frac{dv}{ds}\right| f(v(s)) \,,$$

where the pdf for v is

$$f(v) = \frac{\left(\frac{\nu}{2\sigma^2}\right)^{\nu/2}}{\Gamma(\nu/2)} v^{\nu/2-1} e^{-\nu v/2\sigma^2} \,.$$

Using $v = s^2$ we have $dv/ds = 2s$ and therefore

$$g(s) = 2s \frac{\left(\frac{\nu}{2\sigma^2}\right)^{\nu/2}}{\Gamma(\nu/2)} s^{2(\nu/2-1)} e^{-\nu s^2/2\sigma^2} = 2 \frac{\left(\frac{\nu}{2\sigma^2}\right)^{\nu/2}}{\Gamma(\nu/2)} s^{\nu-1} e^{-\nu s^2/2\sigma^2} \,.$$

2019 Exam Question 3(f)

(f) Show that in the limit where ν is very large, the statistic t_μ becomes

$$t_\mu \approx \frac{(y-\mu)^2}{\sigma^2} \, .$$

[5]

3(f) [5 marks] Having $\nu \to \infty$ means $\sigma_v^2 = 2\sigma^4/\nu \to 0$, i.e., the estimate v is always equal to σ^2. Expanding the logarithm using $\ln(1+\varepsilon) \approx \varepsilon$ for small ϵ and replacing v by σ^2 gives

$$t_\mu = (1+\nu) \ln \left[1 + \frac{1}{\nu} \frac{(y-\mu)^2}{v} \right] \to \frac{(y-\mu)^2}{\sigma^2} \, .$$