Propagation of Errors and the Variance-Covariance Matrix

John Quinn
Advanced Lab.

Errors on a measured quantity, x

- Assume we have uncertainties on some measured quantities and repeated measurements give a spread.

- We characterise the spread by the variance and standard deviation (σ) of the parent population:

$$\mathrm{Var(x)} = \sigma^2 = \lim_{N\to\infty} \frac{1}{N} \sum_{i=i}^{N} (x_i - \bar{x})^2$$

- If the mean is derived from the data itself, then the sample variance is defined as:

$$s^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2$$

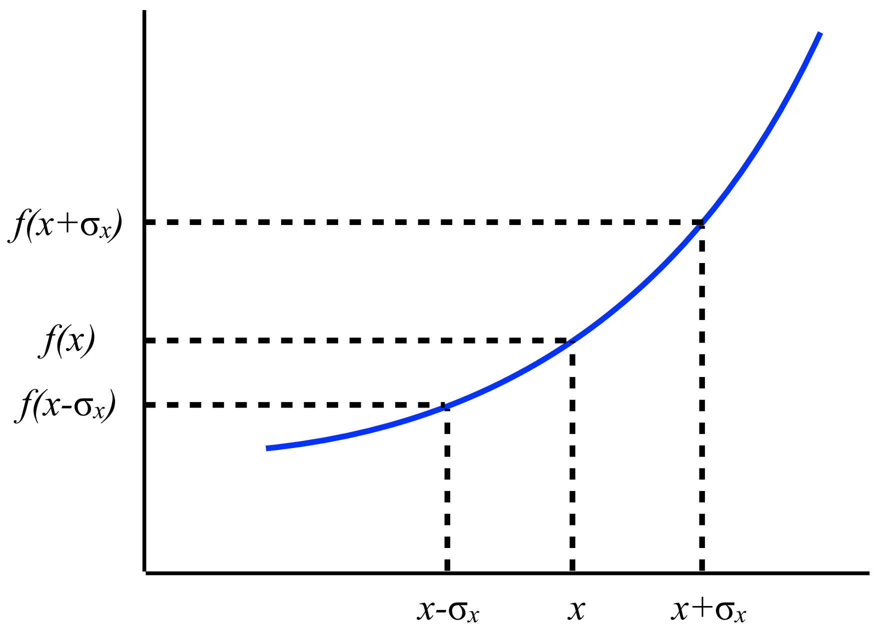
- s and σ are often used interchangeably....

- Rule: if the mean is derived from the data then divide by *(N-1)*.

- "Var" and "σ" will be used in rest of slides....

Error on $f(\mathbf{x})$

- We want to know how the spread propagates through a function $f(x)$,
  - i.e. what is $f(x \pm \sigma_x)$?
  - or, for multiple variables, what is $f(x_1 \pm \sigma_{x1}, x_2 \pm \sigma_{x2}, \ldots)$?
- For function of a single variable, we could calculate $f(x), f(x + \sigma_x), f(x - \sigma_x)$ and use these value to quote the best estimate of $f(x)$ and the confidence region.



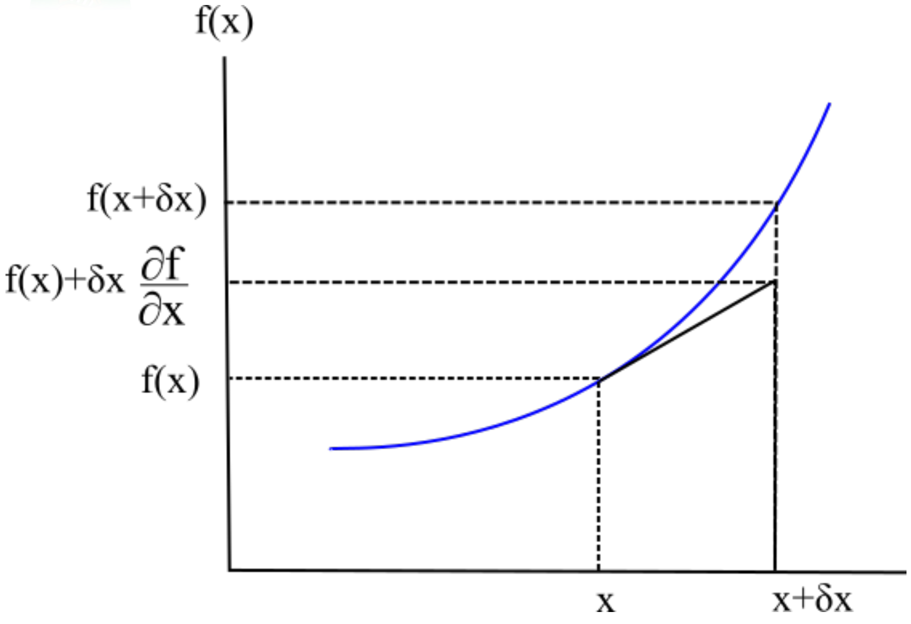
Advantages:

- Easy to do.
- For non-linear functions we can get asymmetric error bars.

Disadvantages:

- Does not scale to functions of multiple variables
- Does not handle correlations between variables.

Error on $f(\mathbf{x})$ - the standard way!



- Use Taylor Expansion:
  - for function of one variable:

$$f(x+\delta x) = f(x) + \delta x \frac{\partial f}{\partial x} + \frac{\delta x^2}{2!}\frac{\partial^2 f}{\partial x^2} + \ldots$$

- if δx is small:

$$f(x+\delta x) \approx f(x) + \delta x \frac{\partial f}{\partial x}$$

- Extend to function of two variables?
  - e.g. for $f(x,y)$: $$f(x+\delta x, y+\delta y) \approx f(x,y) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y$$

Error on $f(\mathbf{x})$: Vector Notation

$$f(x + \delta x, y + \delta y) \approx f(x, y) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y$$

- For multiple variables (using vector notation):

$$f(\mathbf{x} + \delta \mathbf{x}) \approx f(\mathbf{x}) + \mathbf{g}(\mathbf{x})^T \delta \mathbf{x} + \ldots$$

Note: f() is scalar-valued function of a vector

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \qquad \delta \mathbf{x} = \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_n \end{bmatrix}$$

$\mathbf{g}(\mathbf{x})$ is the Gradient vector (also called the Jacobian)

- e.g. for $f(x_1,x_2)$:

$$f(x_1 + \delta x_1, x_2 + \delta x_2) \approx f(x_1, x_2) + \frac{\partial f}{\partial x_1}\delta x_1 + \frac{\partial f}{\partial x_2}\delta x_2$$

Error on $f(\mathbf{x})$

- Say we have N measurements (x_i, y_i) and some function $f(x_i, y_i)$ of them.

$$f(x+\delta x, y+\delta y) \approx f(x,y) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y$$

- Assume that: $\overline{f(x,y)} = f(\bar{x}, \bar{y})$

$$\mathrm{Var(x)} = \sigma^2 = \lim_{N\to\infty} \frac{1}{N}\sum_{i=i}^{N} [x_i - \bar{x}]^2$$

- Now, let's calculate the variance of $u = f(x, y)$

$$\delta x_i = x_i - \bar{x}$$

$$\sigma_u^2 = \frac{1}{N}\sum_{i=1}^{N}\left[f(x_i, y_i) - \overline{f(x,y)}\right]^2 = \frac{1}{N}\sum_{i=1}^{N}\left[f(x_i, y_i) - f(\bar{x}, \bar{y})\right]^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[f(\bar{x}+\delta x_i, \bar{y}+\delta y_i) - f(\bar{x}, \bar{y})\right]^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[f(\bar{x}, \bar{y}) + \frac{\partial f}{\partial x}\delta x_i + \frac{\partial f}{\partial y}\delta y_i - f(\bar{x}, \bar{y})\right]^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[\left(\frac{\partial f}{\partial x}\right)^2 \delta x_i^2 + \left(\frac{\partial f}{\partial y}\right)^2 \delta y_i^2 + 2\left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial f}{\partial y}\right)\delta x_i \delta y_i\right]$$

Error on $f(\mathbf{x})$

$$\sigma_u^2 = \frac{1}{N}\sum_{i=1}^{N}\left[\left(\frac{\partial f}{\partial x}\right)^2 \delta x_i^2 + \left(\frac{\partial f}{\partial y}\right)^2 \delta y_i^2 + 2\left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial f}{\partial y}\right)\delta x_i \delta y_i\right]$$

Recall:

$$\delta x_i = (x_i - \bar{x}) \qquad \sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

$$\delta y_i = (y_i - \bar{y}) \qquad \sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

Define the covariance term:

$$\sigma_{xy}^2 \equiv \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$$

Note: despite it being written as σ_{xy}^2 it can be, and often is, negative!

Hence:

$$\sigma_u^2 = \left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + 2\left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial f}{\partial y}\right)\sigma_{xy}^2$$

Error on $f(\mathbf{x})$

Can extend to more variables:

If $u = f(x, y, \ldots)$, then the error in u is

$$\sigma_u^2 = \left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + \ldots + 2\left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial f}{\partial y}\right)\sigma_{xy}^2 + \ldots$$

The Propagation of Errors Formula

Notes:

- always gives symmetric error bars
- for non-linear functions σ_x, σ_y, $\ldots$ need to be sufficiently small

The Variance-Covariance Matrix

- Software packages (e.g. `numpy.cov()`) usually return the covariances in the variance-covariance matrix.

- For example, for data consisting of pairs of (x, y) measurements:

$$\mathbf{C} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{yx}^2 & \sigma_y^2 \end{bmatrix}$$

- The
  - diagonal elements are the variances and
  - the off-diagonal elements are the covariances

Numpy: std() and cov()

Default: numpy.std() divides by N, but numpy.cov() divides by N−1.

numpy.std(a, axis=None, dtype=None, out=None, ddof=0, keepdims=False)
Compute the standard deviation along the specified axis.

ddof : *int, optional*
Means *Delta Degrees of Freedom*. The divisor used in calculations is `N - ddof`, where `N` represents the number of elements. By default *ddof* is zero.

numpy.cov(m, y=None, rowvar=1, bias=0, ddof=None)
Estimate a covariance matrix, given data.
Covariance indicates the level to which two variables vary together. If we examine N-dimensional samples, $X=[x_1, x_2, x_3, \ldots x_N]^T$, then the covariance matrix element C_{ij} is the covariance of x_i and x_j. The element C_{ij} is the variance of x_i.

bias : *int, optional*
Default normalization is by `(N - 1)`, where `N` is the number of observations given (unbiased estimate). If *bias* is 1, then normalization is by `N`. These values can be overridden by using the keyword `ddof` in numpy versions >= 1.5.

ddof : *int, optional*
New in version 1.5.
If not `None` normalization is by `(N - ddof)`, where `N` is the number of observations; this overrides the value implied by `bias`. The default value is `None`.

Example: apply to area calculation

$$\sigma_u^2 = \left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + 2\left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial f}{\partial y}\right)\sigma_{xy}^2$$

Function: $A = w \times l$

$$\sigma_A^2 = \left(\frac{\partial A}{\partial w}\right)^2 \sigma_w^2 + \left(\frac{\partial A}{\partial l}\right)^2 \sigma_l^2 + 2\left(\frac{\partial A}{\partial w}\right)\left(\frac{\partial A}{\partial l}\right)\sigma_{wl}^2$$

$$= l^2\sigma_w^2 + w^2\sigma_l^2 + 2lw\sigma_{wl}^2$$

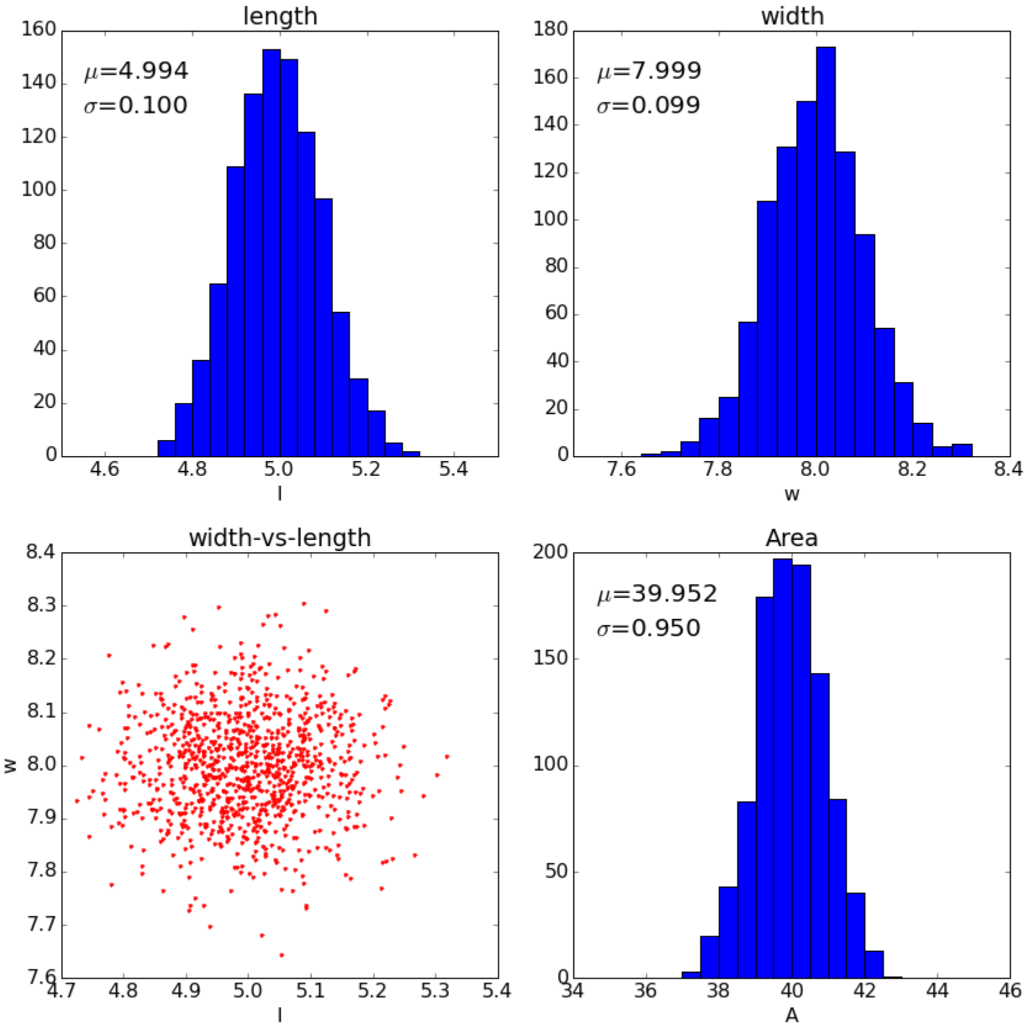
$$\left(\frac{\sigma_A}{A}\right)^2 = \left(\frac{\sigma_w}{w}\right)^2 + \left(\frac{\sigma_l}{l}\right)^2 + \frac{2\sigma_{wl}^2}{wl}$$

$$\sigma_A = A\sqrt{\left(\frac{\sigma_w}{w}\right)^2 + \left(\frac{\sigma_l}{l}\right)^2 + \frac{2\sigma_{wl}^2}{wl}}$$

Now, let's simulate and compare to the above formula: l=5.0±0.1 cm, w=8.0±0.1 cm

Example: Area Estimation (no correlation)

```
l=5+0.1*np.random.randn(1000)
w=8+0.1*np.random.randn(1000)
a=w*l
```




```
c=np.cov(w,l)
```

```
array([[ 0.00979411,  0.00016841],
       [ 0.00016841,  0.01007087]])
```

```
print(np.sqrt(c[0][0]),
      np.sqrt(c[1][1]))
```

```
0.099 0.100
```

$$\sigma_A = A\sqrt{\left(\frac{\sigma_w}{w}\right)^2 + \left(\frac{\sigma_l}{l}\right)^2} = 0.943$$

$$\sigma_A = A\sqrt{\left(\frac{\sigma_w}{w}\right)^2 + \left(\frac{\sigma_l}{l}\right)^2 + 2\frac{\sigma_{wl}^2}{w\,l}} = 0.950$$

Example: Area Estimation (correlated)

```
l_s=np.sort(l)
w_s=np.sort(w)
a_s=w_s*l_s
```

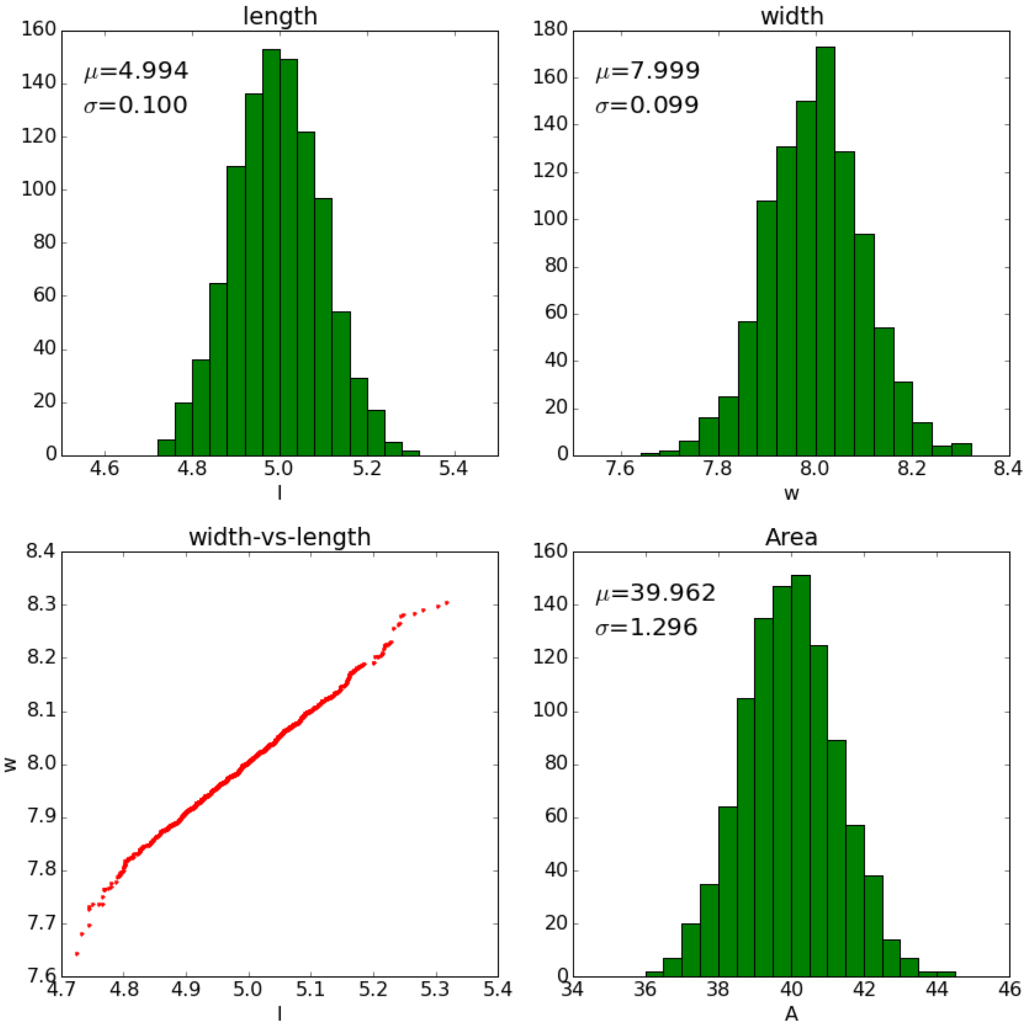
Same points but sorted, hence correlated!

```
c_s=np.cov(w_s,l_s)

array([[ 0.00979411,  0.00991393],
       [ 0.00991393,  0.01007087]])

print(np.sqrt(c_s[0][0]),
      np.sqrt(c_s[1][1]))

0.100 0.099
```



$$\sigma_A = A\sqrt{\left(\frac{\sigma_w}{w}\right)^2 + \left(\frac{\sigma_l}{l}\right)^2} = 0.937$$

$$\sigma_A = A\sqrt{\left(\frac{\sigma_w}{w}\right)^2 + \left(\frac{\sigma_l}{l}\right)^2 + 2\frac{\sigma_{wl}^2}{w\,l}} = 1.292$$

Example: Area Estimation (anti-correlated)

```
l_s=np.sort(l)
w_s=-np.sort(-w)
a_s=w_s*l_s
```

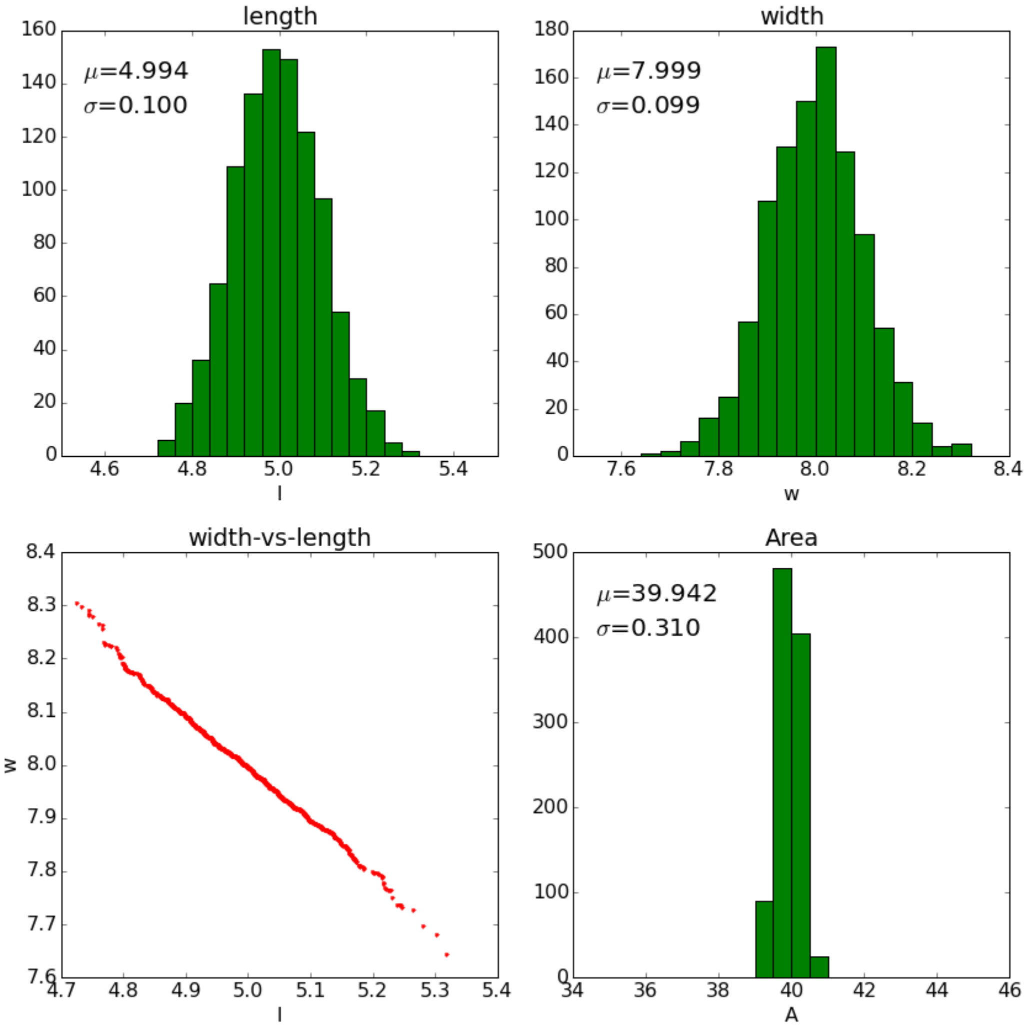
Same points but reverse sorted, hence anti-correlated!

```
c_s=np.cov(w_s,l_s)
```

```
array([[ 0.00979411, -0.00991637],
       [-0.00991637,  0.01007087]])
```

```
print(np.sqrt(c_s[0][0]),
      np.sqrt(c_s[1][1]))
```

```
0.101 0.098
```



$$\sigma_A = A\sqrt{\left(\frac{\sigma_w}{w}\right)^2 + \left(\frac{\sigma_l}{l}\right)^2} = 0.937$$

$$\sigma_A = A\sqrt{\left(\frac{\sigma_w}{w}\right)^2 + \left(\frac{\sigma_l}{l}\right)^2 + 2\frac{\sigma_{wl}^2}{w\,l}} = 0.293$$

Note: σ_{wl}^2 is negative!

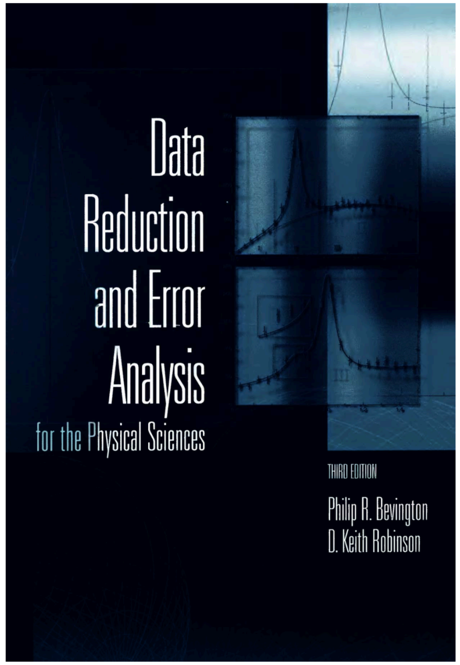
Covariance Terms

- We usually drop/ignore the covariance term (!) since:
  - we assume we have independent measurements that are not correlated.
  - we often do not have enough measurements to calculate the covariance terms
- If there is a chance of measurement fluctuations being correlated then it should be included for an accurate estimate of the error.
- However, when we fit a function to data (next week!), the function parameters are usually highly correlated:
  - the fitting package returns the full variance-covariance matrix
  - the covariance terms should be included in generating any confidence intervals.

Specific Formulas

Specific formulas:

$x = au + bv$	$\sigma_x^2 = a^2\sigma_u^2 + b^2\sigma_v^2 + 2ab\sigma_{uv}^2$
$x = auv$	$\frac{\sigma_x^2}{x^2} = \frac{\sigma_u^2}{u^2} + \frac{\sigma_v^2}{v^2} + 2\frac{\sigma_{uv}^2}{uv}$
$x = \frac{au}{v}$	$\frac{\sigma_x^2}{x^2} = \frac{\sigma_u^2}{u^2} + \frac{\sigma_v^2}{v^2} - 2\frac{\sigma_{uv}^2}{uv}$
$x = au^b$	$\frac{\sigma_x}{x} = b\frac{\sigma_u}{u}$
$x = ae^{bu}$	$\frac{\sigma_x}{x} = b\sigma_u$
$x = a^{bu}$	$\frac{\sigma_x}{x} = (b \ln a)\sigma_u$
$x = a \ln(bu)$	$\sigma_x = ab\frac{\sigma_u}{u}$
$x = a \cos(bu)$	$\sigma_x = -\sigma_u ab \sin(bu)$
$x = a \sin(bu)$	$\sigma_x = \sigma_u ab \cos(bu)$



Linear Correlation Coefficient

- The Linear Correlation Coefficient between two variables is defined as:

$$\rho_{xy} = \frac{\sigma_{xy}^2}{\sigma_x \sigma_y}$$

$-1 \leq \rho \leq 1$
ρ close to 0 indicates no correlation,
$|\rho|$ close to 1 means highly correlated.

- For the three simulated data sets:

Data Set	ρ_{wl}
Unsorted	0.017
Sorted	+1.0
Reverse Sorted	-1.0

Relative Contributions

- Sometimes when lots of measurements go into calculating a final result it is impractical to propagate every error through.
- It that case it is justified (and encouraged) to identify those which make the biggest contribution and neglect those which make small contributions.
- Example: measure area:
  - L=22.1 ± 0.1 cm, W=7.3 ± 0.1 cm.

$$\frac{\sigma_L}{L} = 0.005 \qquad \frac{\sigma_W}{W} = 0.014$$

$$\begin{aligned} \sigma_A &= A\sqrt{0.014^2 + 0.005^2} \\ &\approx 0.014\,A\left(1 + \frac{1}{2}\left(\frac{0.005}{0.014}\right)^2\right) \\ &\approx 0.014\,A\,(1 + 0.06) \\ &= 0.015\,A \end{aligned}$$

So, the uncertainty on the length makes only a 6% contribution to the overall uncertainty on the area an it would have been justified to neglect it.

Numerical Propagation of Errors

$$\sigma_u^2 = \left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + 2\left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial f}{\partial y}\right)\sigma_{xy}^2$$

- For complicated functions it can be tedious to calculate all of the derivatives/
- Alternatively, one can use a computer to numerically propagate errors through any formula:

$$u = f(x, y)$$

First, calculate:

$$du_x = f(x + \sigma_x, y) - f(x, y)$$

$$du_y = f(x, y + \sigma_y) - f(x, y)$$

Then,

$$\sigma_u = \sqrt{du_x^2 + du_y^2}$$ w/o covariance

$$\sigma_u = \sqrt{du_x^2 + du_y^2 + 2\frac{du_x}{\sigma_x}\frac{du_y}{\sigma_y}\sigma_{xy}^2}$$ with covariance

Numerical Propagation of Errors

In [7]:

```
ml=np.mean(l_s)  # mean of sorted lengths
mw=np.mean(w_s)  # mean of sorted widths

A=ml*mw

c=np.cov(l_s,w_s)  # covariance matrix of the sorted parameters
σl=np.sqrt(c[0,0])
σw=np.sqrt(c[1,1])
σlw2=c[0,1]       # covariance term

print(A*np.sqrt((σl/ml)**2 + (σw/mw)**2))
print(A*np.sqrt((σl/ml)**2 + (σw/mw)**2 + 2*σlw2/A))
```

```
0.9202593612148566
1.268357589122689
```

Using Equation derived using Propagation of Errors Formula

$$\sigma_A = A\sqrt{\left(\frac{\sigma_w}{w}\right)^2 + \left(\frac{\sigma_l}{l}\right)^2 + \frac{2\sigma_{wl}^2}{wl}}$$

In [8]:

```
def A(l,w):
    return l*w

ml=np.mean(l_s)  # mean of sorted lengths
mw=np.mean(w_s)  # mean of sorted widths

c=np.cov(l_s,w_s)
σl=np.sqrt(c_s[0,0])
σw=np.sqrt(c_s[1,1])
σlw2=c_s[0,1]

dal=A(ml+σl,mw)-A(ml,mw)
daw=A(ml,mw+σw)-A(ml,mw)

print(np.sqrt(dal**2 + daw**2))
print(np.sqrt(dal**2 + daw**2 + 2 * dal/σl * daw/σw * σlw2))
```

```
0.9211158132746902
1.2689791264816905
```

Using Numerical Propagation of Errors

$$\sigma_u = \sqrt{du_x^2 + du_y^2 + 2\frac{du_x}{\sigma_x}\frac{du_y}{\sigma_y}\sigma_{xy}^2}$$

Conclusions

- The standard formula for propagation of errors is:

$$\sigma_u^2 = \left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + \ldots + 2\left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial f}{\partial y}\right)\sigma_{xy}^2 + \ldots$$

  - the covariance terms can be dropped if the errors on the variables are uncorrelated.
  - produces symmetric error bars
  - the errors on parameters must be sufficiently small.

- Be familiar with error propagation formula for standard functions (reference table)

- Propagating errors through complicated functions:
  - Identify the biggest contributors and ignore those which do no contribute significantly.
  - Use Numerical Propagation of Errors, even as a cross-check.