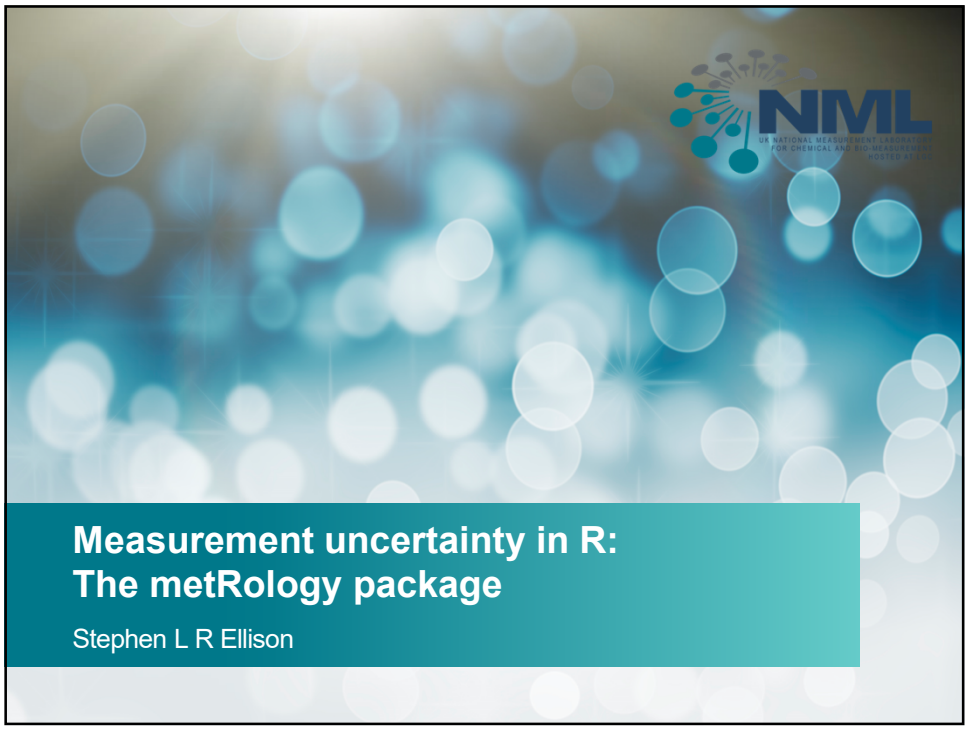


Introduction

- What is R?
- What is the metRology package?
- Functionality: What does it do?
- Philosophy: Who owns it?
- Location: Where is it?

What is …

- R
  - "R is an integrated suite of software facilities for data manipulation, calculation and graphical display"
    - Free, open source package for statistical analysis and programming
    - Extensible via "packages"
- The metRology package
  - An R package for statistics applied to metrology
  - metrology: The science of measurement

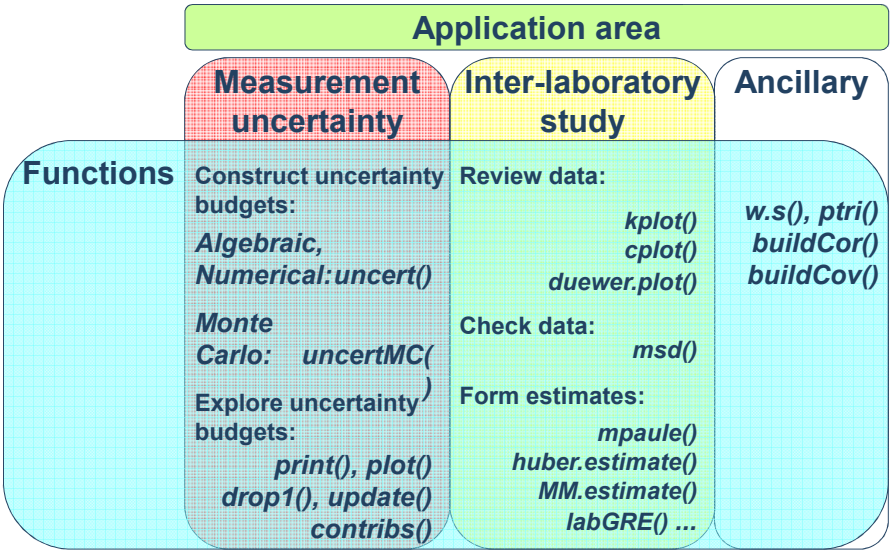
Functionality: What does it do?

"metRology provides classes and calculation and plotting functions for metrology applications, including measurement uncertainty estimation and inter-laboratory metrology comparison studies."

https://r-forge.r-project.org/projects/metrology/
25 September 2017

- Measurement uncertainty estimation
- Support for interlaboratory studies

metRology: Functions



Measurement uncertainty

1. ISO Guide to the Expression of Uncertainty in Measurement
   - First-order error propagation from a 'measurement model'

$$y = f(x_1, x_2, \ldots x_n)$$

$$u(y) = \sqrt{\sum_{i=1}^{n}\left(\frac{\partial y}{\partial x_i}u(x_i)\right)^2 + 2\sum_{i=1}^{n}\sum_{j>i}^{n}\frac{\partial y}{\partial x_i}\frac{\partial y}{\partial x_j}\mathrm{cov}(x_i, x_j)}$$

   - Correlation allowed for

Uncertainty implementations in metRology

- Direct combination of contributions $u_i(y) = u(x_i)\dfrac{\partial y}{\partial x_i}$
- Algebraic differentiation of $f(x_1, ...)$
- Numerical differentiation
  - Kragten's method

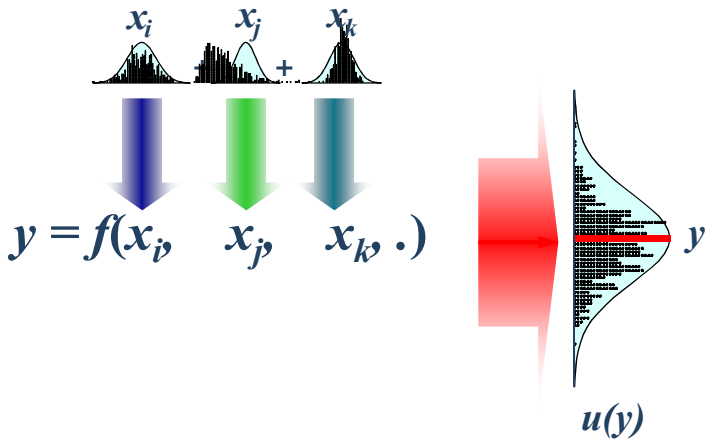
$$u(x_i)\partial y/\partial x_i \approx f(x_i + u(x_i)) - f(x_i)$$

  - Symmetric finite difference

$$\partial y/\partial x_i \approx [f(x_i + \delta x_i) - f(x_i - \delta x_i)]/2\,\delta x_i$$

Uncertainty implementations in metRology

2. Monte Carlo (GUM Supplement 1)



Examples

```
expr <- expression(a+b*2+c*3+d/2)
       # a=1(0.1), b=3(0.3), c=2(0.2), d=11(1.1)
u.expr<-uncert(expr, x, u, method="NUM")
```

```
#Compare with default:
uncert(u=c(0.1, 0.3, 0.2, 1.1), c=c(1.0, 2.0, 3.0,
  0.5))

#... or with function method
 f <- function(a,b,c,d) a+b*2+c*3+d/2
 u.fun<-uncert(f, x, u, method="NUM")

#.. or with the formula method
 u.form<-uncert(~a+b*2+c*3+d/2, x, u, method="NUM")
```

Examples (cont.)

```
Uncertainty evaluation
Call:
  uncert.expression(expr = expr, x = x, u = u, method =
  "NUM")
Expression: a + b * 2 + c * 3 + d/2
```

```
Evaluation method:  NUM
Uncertainty budget:
  x  u   c   u.c
a  1 0.1 1.0 0.10
b  3 0.3 2.0 0.60
c  2 0.2 3.0 0.60
d 11 1.1 0.5 0.55
```

```
   y:  18.5
u(y):  1.01612
```

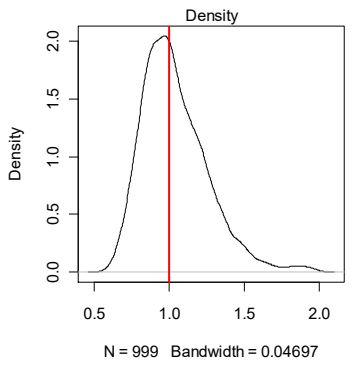
A more interesting Monte carlo case

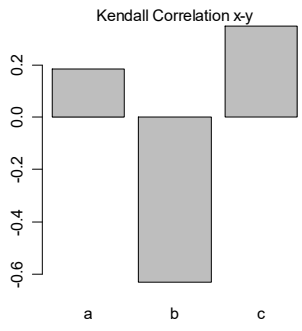
```
expr <- expression(a/(b-c))
x <- list(a=1, b=3, c=2)
Expression: a/(b - c)
   y:  1
u(y):  0.2187031

Monte Carlo evaluation using 999 replicates:
   y:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.6047  0.8845  1.0020  1.0410  1.1630  1.9540
```

A more interesting Monte carlo case

```
> par(mfrow=c(2,2))
> plot(u.invexpr, which=1:4, pch=20, method="k")
                  # method="k" gives Kendall correlation
```





Other MU diagnostics and utilities

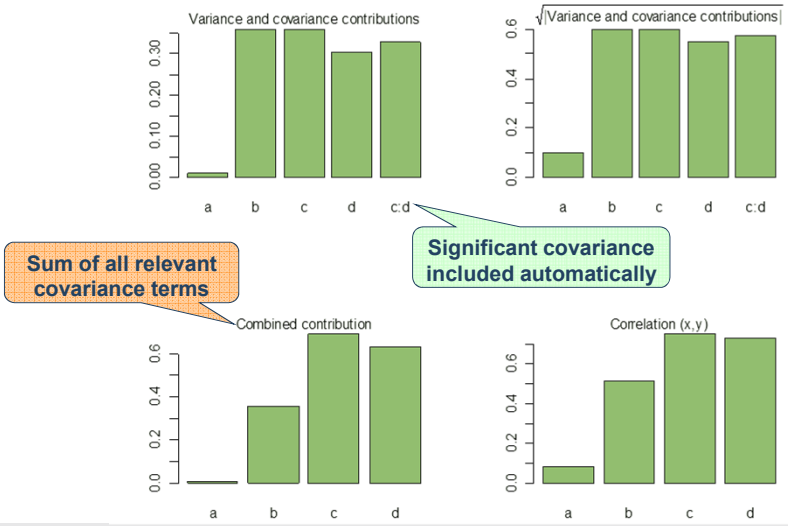
```
#An example with correlation
>  u.cor<-diag(1,4)
>  u.cor[3,4]<-u.cor[4,3]<-0.5
>  u.formc<-uncert(~a+b*2+c*3+d/2, x, u, method="NUM",
+
>
```

Builds a correlation matrix

```
#Which uncertainties matter most?
> par(mfrow=c(2,2)
> plot(u.formc)
```

Plots for exploring uncertainty budgets

Other MU diagnostics and utilities: plot.uncert



Other MU diagnostics and utilities

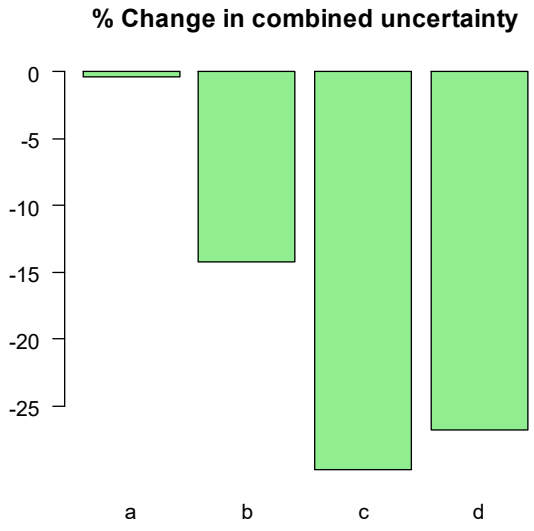
```
#An example with correlation
>  u.cor<-diag(1,4)
>  u.cor[3,4]<-u.cor[4,3]<-0.5
>  u.formc<-uncert(~a+b*2+c*3+d/2, x, u, method="NUM",
+
>
```

```
#Which uncertainties matter most?
> par(mfrow=c(2,2)
> plot(u.formc)

#What happens if we reduce one uncertainty?
> barplot( drop1(u.formc) )
```

Drop each term successively

Other MU diagnostics and utilities: drop1



Other MU diagnostics and utilities

- update
  - Modifies uncertainty budgets (for example, changing an individual uncertainty or the method of evaluation)
- w.s, welch.satterthwaite
  - Welch-Satterthwaite effective degrees of freedom
- buildCov, buildCor
  - simplifies assembly of correlation or covariance matrices by taking a short list of labelled off-diagonal terms

Philosophy: Who owns metRology?

- metRology is an open source project
- No single ‘owner’
- Contributions invited
- Licence is GPL
  - Copyright is held by code contributor
  - Contribution is conditional on granting full permissions under the GPL:
    - right to distribute and modify under the same terms

Location: Where is metRology?

- CRAN:

```
install.packages("metRology")
```

- R-Forge: metRology
  https://r-forge.r-project.org/projects/metrology/
- Code access via SubVersion (svn)
  - Tortoise SVN for Windows

Acknowledgements

- NIST Statistical Engineering Division
  - Support and code contribution