units + errors = quantities

Quantity Calculus for R Vectors

Iñaki Úcar, postdoctoral fellow @ UC3M-Santander Big Data Institute (IBiDat)

Edzer Pebesma, full professor @ Institute for Geoinformatics, University of Münster

November 15-16, 2019

Quantity Calculus

Two quotes from Cobb and Moore (1997):

> Data are not just numbers, they are numbers with a **context**.
>
> In data analysis, context provides meaning.

Quantity Calculus

Two quotes from Cobb and Moore (1997):

> Data are not just numbers, they are numbers with a **context**.
>
> In data analysis, context provides meaning.

Examples of context in R:

- Names of `data.frame` columns
- Dimensions of an array
- Levels of `factor`
- Time-related objects `POSIXt`, `Date`, `difftime`…

Quantity Calculus

Two quotes from Cobb and Moore (1997):

> Data are not just numbers, they are numbers with a **context**.
>
> In data analysis, context provides meaning.

From the International Vocabulary of Metrology (VIM):

> [A **quantity** is] a property of a phenomenon, body, or substance, where the property has a magnitude that can be expressed as a number and a reference.

Examples of context in R:

- Names of `data.frame` columns
- Dimensions of an array
- Levels of `factor`
- Time-related objects `POSIXt`, `Date`, `difftime`…

Quantity Calculus

Two quotes from Cobb and Moore (1997):

> Data are not just numbers, they are numbers with a **context**.
>
> In data analysis, context provides meaning.

Examples of context in R:

- Names of `data.frame` columns
- Dimensions of an array
- Levels of `factor`
- Time-related objects `POSIXt`, `Date`, `difftime`...

From the International Vocabulary of Metrology (VIM):

> [A **quantity** is] a property of a phenomenon, body, or substance, where the property has a magnitude that can be expressed as a number and a reference.

$$\overbrace{10.45}^{\text{value}}\overbrace{\underbrace{\pm\ 0.03}_{\text{uncertainty}}\quad\underbrace{\mathrm{km}}_{\text{unit}}}^{\text{context}}$$

units

The `units` package [1]:

- Support for measurement units in R vectors and arrays
- Automatic propagation, conversion, simplification
- Raising errors in case of unit incompatibility
- Compatible with the POSIXct, Date and difftime classes
- Uses Unidata's `UDUNITS-2` library and database

[1] Pebesma E, Mailund T, Hiebert J (2016). "Measurement Units in R." *R Journal*, 8(2), 486-494. doi: 10.32614/RJ-2016-061.

units

The `units` package [1]:

- Support for measurement units in R vectors and arrays
- Automatic propagation, conversion, simplification
- Raising errors in case of unit incompatibility
- Compatible with the POSIXct, Date and difftime classes
- Uses Unidata's `UDUNITS-2` library and database

```r
library(units)

x ← set_units(rnorm(3), m)
y ← set_units(rnorm(3), s)
sum(x)
```

```
## 1.169389 [m]
```

```r
x / y
```

```
## Units: [m/s]
## [1]  2.166281 -1.396840 -3.421720
```

```r
x + y
```

```
## Error: cannot convert s into m
```

[1] Pebesma E, Mailund T, Hiebert J (2016). "Measurement Units in R." *R Journal*, 8(2), 486-494. doi: 10.32614/RJ-2016-061.

units

The `units` package [1]:

- Support for measurement units in R vectors and arrays
- Automatic propagation, conversion, simplification
- Raising errors in case of unit incompatibility
- Compatible with the POSIXct, Date and difftime classes
- Uses Unidata's `UDUNITS-2` library and database

Specifics:

- S3 implementation
- Group Generics `Math`, `Ops`, `Summary` and more
- `?units_options` controls printing, parsing, autoconversion and simplification
- Installation of new units

```r
library(units)

x ← set_units(rnorm(3), m)
y ← set_units(rnorm(3), s)
sum(x)
```

```
## 1.169389 [m]
```

```r
x / y
```

```
## Units: [m/s]
## [1]  2.166281 -1.396840 -3.421720
```

```r
x + y
```

```
## Error: cannot convert s into m
```

[1] Pebesma E, Mailund T, Hiebert J (2016). "Measurement Units in R." *R Journal*, 8(2), 486-494. doi: 10.32614/RJ-2016-061.

errors

The `errors` package [1]:

- Support for uncertainties in R vectors and arrays
- Automatic propagation (first-order TSM, as recommended by BIPM's GUM)
- Pretty printing (also following BIPM's GUM)

[1] **Ucar I**, Pebesma E, Azcorra A (2018). "Measurement Errors in R." *R Journal*, 10(2), 549-557. doi: 10.32614/RJ-2018-075.

errors

The `errors` package [1]:

- Support for uncertainties in R vectors and arrays
- Automatic propagation (first-order TSM, as recommended by BIPM's GUM)
- Pretty printing (also following BIPM's GUM)

```r
library(errors)

x ← set_errors(rnorm(3), 0.1)
y ← set_errors(rnorm(3), 0.1)
sum(x)
```

```
## 1.2(2)
```

```r
print(sum(x), notation="plus-minus", digits=2)
```

```
## 1.17 ± 0.17
```

```r
x / y
```

```
## Errors: 0.3770097 0.4249392 3.3591220
## [1]  2.166281 -1.396840 -3.421720
```

[1] **Ucar I**, Pebesma E, Azcorra A (2018). "Measurement Errors in R." *R Journal*, 10(2), 549-557. doi: 10.32614/RJ-2018-075.

errors

The `errors` package [1]:

- Support for uncertainties in R vectors and arrays
- Automatic propagation (first-order TSM, as recommended by BIPM's GUM)
- Pretty printing (also following BIPM's GUM)

Specifics:

- S3 implementation
- Correlations/covariances across objects
- Group Generics `Math`, `Ops`, `Summary` and more
- Options for printing control, by default
    - `errors.notation = "parenthesis"`
    - `errors.digits = 1`

```
library(errors)

x ← set_errors(rnorm(3), 0.1)
y ← set_errors(rnorm(3), 0.1)
sum(x)
```

```
## 1.2(2)
```

```
print(sum(x), notation="plus-minus", digits=2)
```

```
## 1.17 ± 0.17
```

```
x / y
```

```
## Errors: 0.3770097 0.4249392 3.3591220
## [1]  2.166281 -1.396840 -3.421720
```

[1] **Ucar I**, Pebesma E, Azcorra A (2018). "Measurement Errors in R." *R Journal*, 10(2), 549-557. doi: 10.32614/RJ-2018-075.

quantities

The `quantities` framework [1]:

- Project funded by the R Consortium [2]
- Integrates `units` and `errors`
- Complete Quantity Calculus system for R

[1] R-Quantities organization on GitHub: https://github.com/r-quantities/

[2] Announcement: https://www.r-consortium.org/announcement/2018/02/22/announcing-second-round-isc-funded-projects-2017

quantities

The `quantities` framework [1]:

- Project funded by the R Consortium [2]
- Integrates `units` and `errors`
- Complete Quantity Calculus system for R

```
library(quantities)

x <- set_quantities(rnorm(3), m, 0.1)
y <- set_quantities(rnorm(3), s, 0.1)
sum(x)
```

```
## 1.2(2) [m]
```

```
x / y
```

```
## Units: [m/s]
## Errors: 0.3770097 0.4249392 3.3591220
## [1]  2.166281 -1.396840 -3.421720
```

```
x + y
```

```
## Error: cannot convert s into m
```

[1] R-Quantities organization on GitHub: https://github.com/r-quantities/

[2] Announcement: https://www.r-consortium.org/announcement/2018/02/22/announcing-second-round-isc-funded-projects-2017

quantities

The `quantities` framework [1]:

- Project funded by the R Consortium [2]
- Integrates `units` and `errors`
- Complete Quantity Calculus system for R

Specifics:

- S3 implementation as a super-class
- Again, Group Generics and other stuff
- Support all conversions, resolve conflicts
- Data wrangling guide (see vignette)
- Parsers (see vignette)

```r
library(quantities)

x ← set_quantities(rnorm(3), m, 0.1)
y ← set_quantities(rnorm(3), s, 0.1)
sum(x)
```

```
## 1.2(2) [m]
```

```r
x / y
```

```
## Units: [m/s]
## Errors: 0.3770097 0.4249392 3.3591220
## [1]  2.166281 -1.396840 -3.421720
```

```r
x + y
```

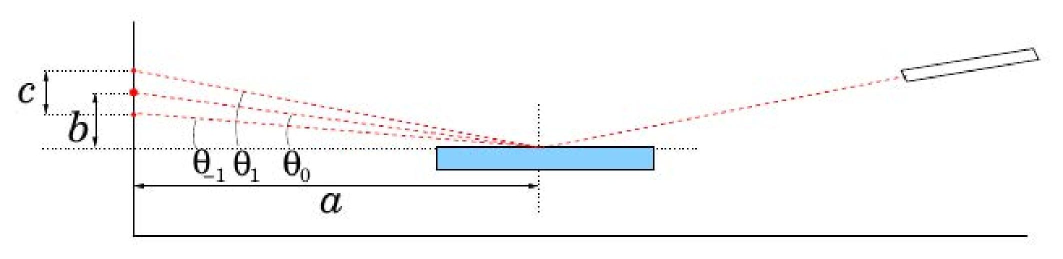
```
## Error: cannot convert s into m
```

[1] R-Quantities organization on GitHub: https://github.com/r-quantities/

[2] Announcement: https://www.r-consortium.org/announcement/2018/02/22/announcing-second-round-isc-funded-projects-2017

Example: a physics experiment

Surface tension in liquids



- Dispersion relation for capillary waves

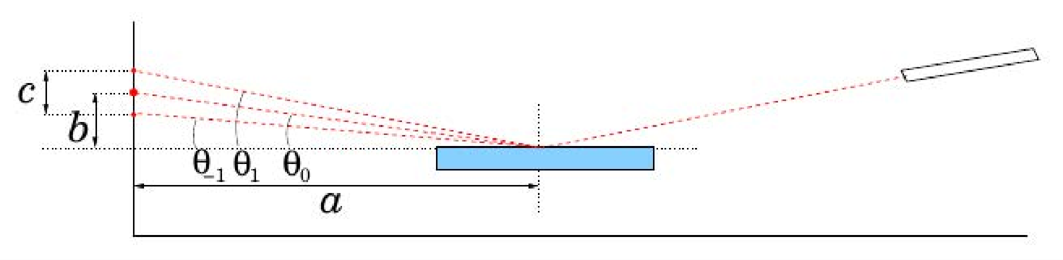
$$w^2 = \frac{\sigma}{\rho}|k|^3, \quad k = \frac{\pi}{n\lambda}\frac{c_n}{a}\sin\left(\frac{b}{a}\right)$$

```
str(df)
```

```
## 'data.frame':    65 obs. of  3 variables:
##  $ f  : num  100 110 120 130 140 150 160 170 180 190 ...
##  $ n  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ c_n: num  11 12 13 13 14 15 16 16 17 17 ...
```

Example: a physics experiment

Surface tension in liquids



- Dispersion relation for capillary waves

$$w^2 = \frac{\sigma}{\rho}|k|^3, \quad k = \frac{\pi}{n\lambda}\frac{c_n}{a}\sin\left(\frac{b}{a}\right)$$

```
str(df)
```

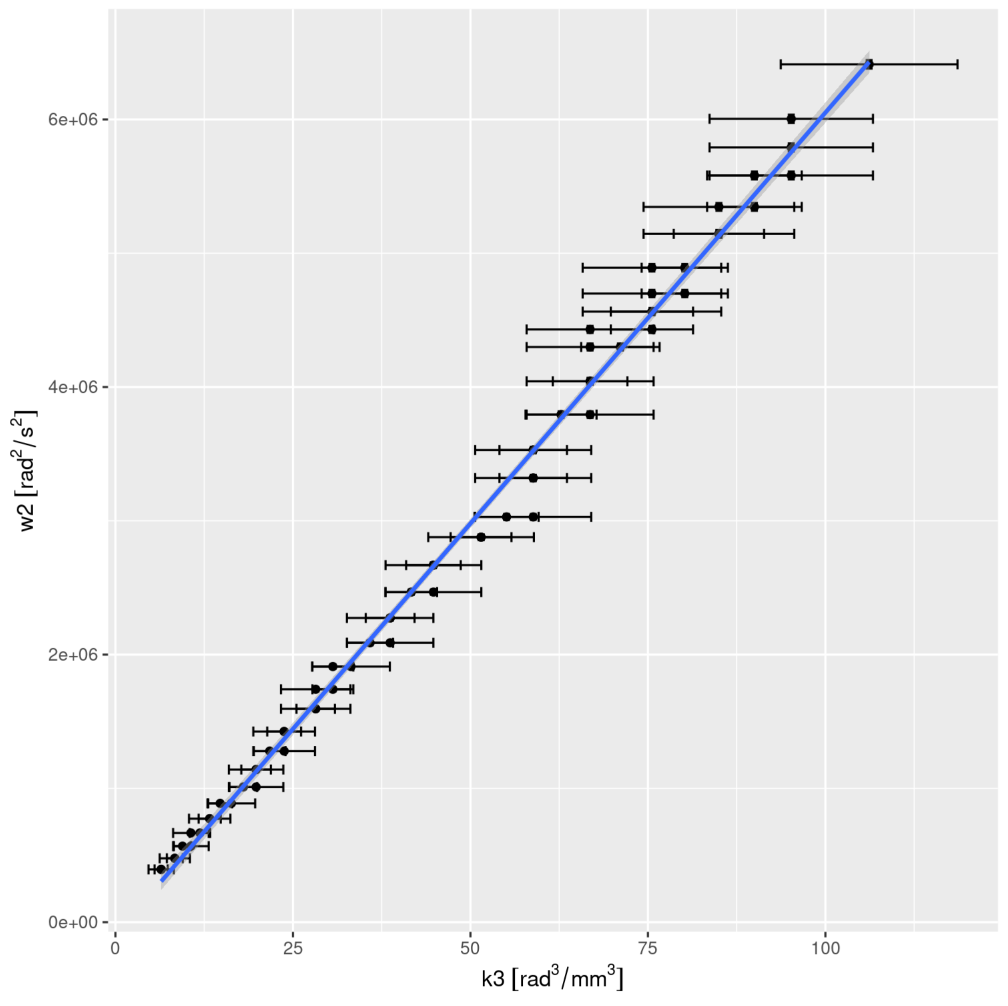
```
## 'data.frame':    65 obs. of  3 variables:
##  $ f  : num  100 110 120 130 140 150 160 170 180 190 ...
##  $ n  : int  1 1 1 1 1 1 1 1 1 1  ...
##  $ c_n: num  11 12 13 13 14 15 16 16 17 17  ...
```

```
library(quantities)

a ← set_quantities(3095, mm, 10)
b ← set_quantities(414+116-203, mm, 1+1+3)
l ← set_quantities(633, nm, 0)

df ← within(df, {
  f ← set_quantities(f, Hz, 1)
  c_n ← set_quantities(c_n, mm, 1)

  w2 ← set_units(2 * pi * f, rad/s)^2
  k3 ← set_units(
    pi/l/n * c_n/a * sin(set_units(b/a, rad)),
    rad/mm)^3
})
head(df[, c("w2", "k3")], 3)
```

```
##                        w2                k3
## 1 3.95(8)e5 [rad^2/s^2]  6(2) [rad^3/mm^3]
## 2 4.78(9)e5 [rad^2/s^2]  8(2) [rad^3/mm^3]
## 3 5.68(9)e5 [rad^2/s^2] 11(3) [rad^3/mm^3]
```

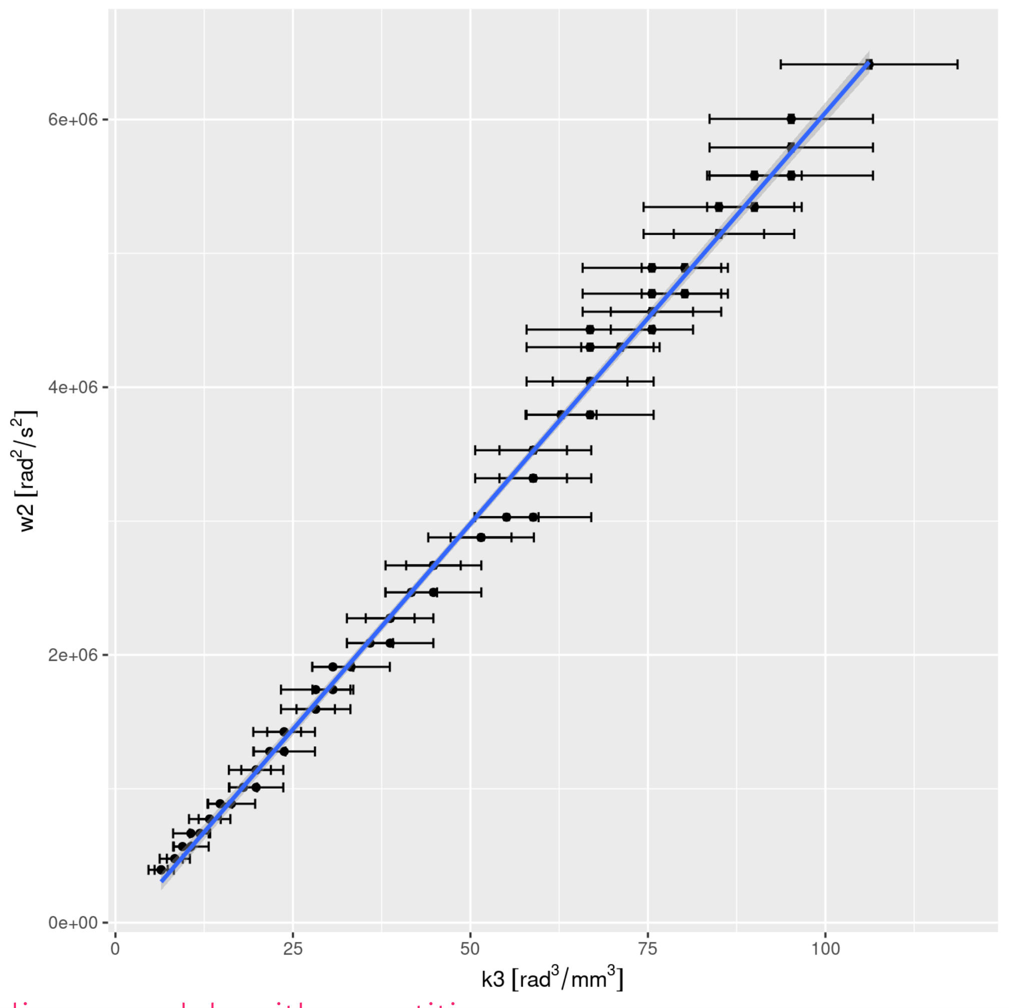
Example: a physics experiment

```r
library(ggplot2)
library(ggforce)

ggplot(df) + aes(k3, w2) +
  geom_errorbar(
    aes(ymin=errors_min(w2), ymax=errors_max(w2))) +
  geom_errorbarh(
    aes(xmin=errors_min(k3), xmax=errors_max(k3))) +
  geom_point() + geom_smooth(method="lm")
```



Example: a physics experiment

```r
library(ggplot2)
library(ggforce)

ggplot(df) + aes(k3, w2) +
  geom_errorbar(
    aes(ymin=errors_min(w2), ymax=errors_max(w2))) +
  geom_errorbarh(
    aes(xmin=errors_min(k3), xmax=errors_max(k3))) +
  geom_point() + geom_smooth(method="lm")
```



Measure the surface tension:

```r
fit ← qlm(w2 ~ k3, df) # [1]
rho ← set_quantities(997, kg/m^3, 1)
sigma ← coef(fit)$k3 * rho
set_units(sigma, dyne/cm)
```

```
## 61.2(6) [dyne/cm]
```

[1] See https://www.r-spatial.org/r/2018/08/31/quantities-final.html#fitting-linear-models-with-quantities

Summary

- Transparent (almost) support for quantity calculus (i.e., computations with context) in R
  - Assign units and/uncertainty metadata and go
  - Automatic propagation, conversion, simplification and reporting

Summary

- Transparent (almost) support for quantity calculus (i.e., computations with context) in R
    - Assign units and/uncertainty metadata and go
    - Automatic propagation, conversion, simplification and reporting
- Published in the R Journal [1, 2], following BIPM's recommendations

[1] Pebesma E, Mailund T, Hiebert J (2016). "Measurement Units in R." *R Journal*, 8(2), 486-494. doi: 10.32614/RJ-2016-061.

[2] **Ucar I**, Pebesma E, Azcorra A (2018). "Measurement Errors in R." *R Journal*, 10(2), 549-557. doi: 10.32614/RJ-2018-075.

Summary

- Transparent (almost) support for quantity calculus (i.e., computations with context) in R
  - Assign units and/uncertainty metadata and go
  - Automatic propagation, conversion, simplification and reporting
- Published in the R Journal [1, 2], following BIPM's recommendations
- `units` (reference) and `errors` (uncertainty) developed in separate packages
- Integrated within the `quantities` framework

[1] Pebesma E, Mailund T, Hiebert J (2016). "Measurement Units in R." *R Journal*, 8(2), 486-494. doi: 10.32614/RJ-2016-061.

[2] **Ucar I**, Pebesma E, Azcorra A (2018). "Measurement Errors in R." *R Journal*, 10(2), 549-557. doi: 10.32614/RJ-2018-075.

Thanks!

https://github.com/r-quantities