

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



Two quotes from Cobb and Moore (1997):

Data are not just numbers, they are numbers with a **context**.

In data analysis, context provides meaning.



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` ...



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.



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.

value context
 $10.45 \pm 0.03 \text{ km}$
 uncertainty unit



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.



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

```
library(units)
```

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

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

```
x / y
```

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

```
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.

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

```
library(units)
```

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

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

```
x / y
```

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

```
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.



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.



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)

```
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.



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.



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>



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>



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)

```
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
```

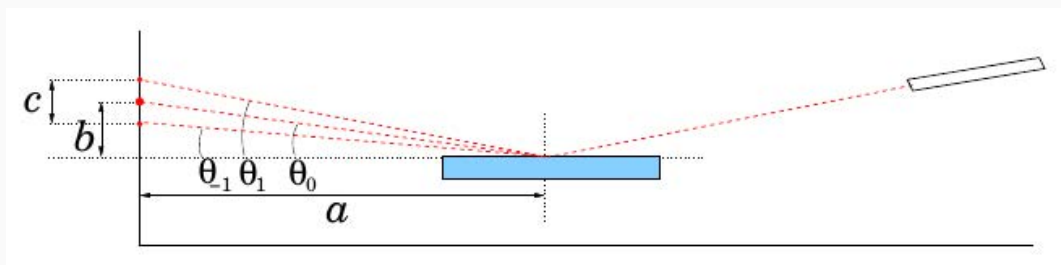
```
x + y
```

```
## 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>

Surface tension in liquids



- Dispersion relation for capillary waves

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

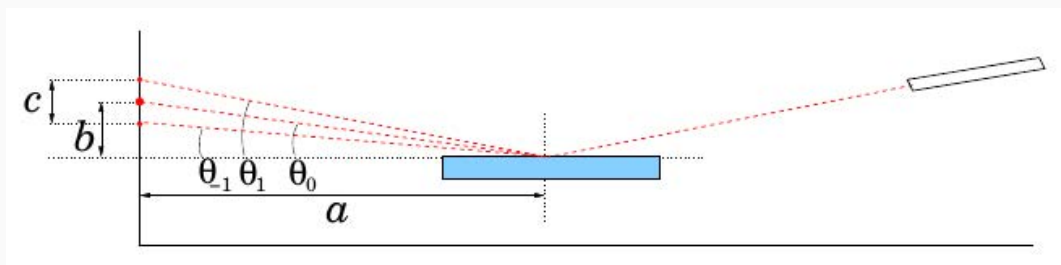
```
str(df)
```

```
## '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

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

```
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)
```

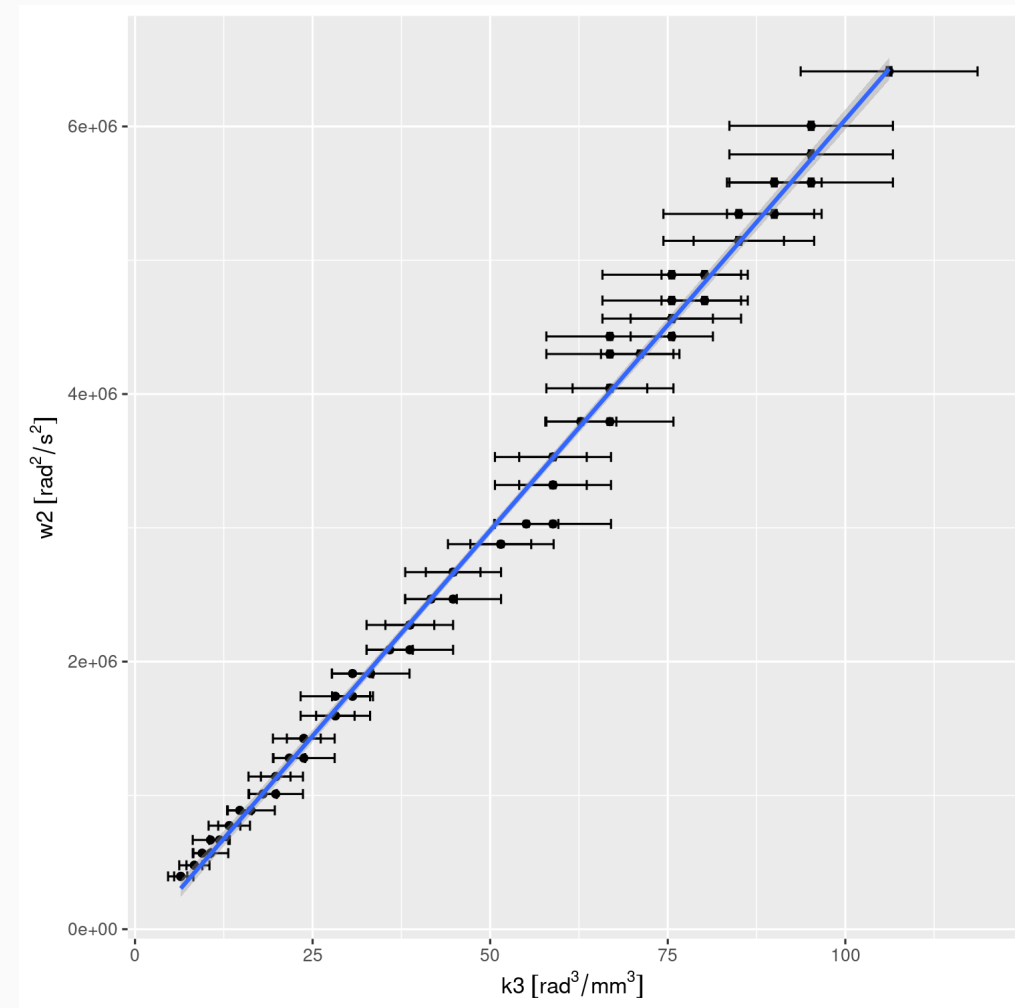
```
##                w2                k3
## 1 3.95(8)e5 [rad^2/s^2]  6(2) [rad^3/mm^3]
## 2 4.78(9)e5 [rad^2/s^2]  8(2) [rad^3/mm^3]
## 3 5.68(9)e5 [rad^2/s^2] 11(3) [rad^3/mm^3]
```


Example: a physics experiment



```
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



```
library(ggplot2)
library(ggforce)

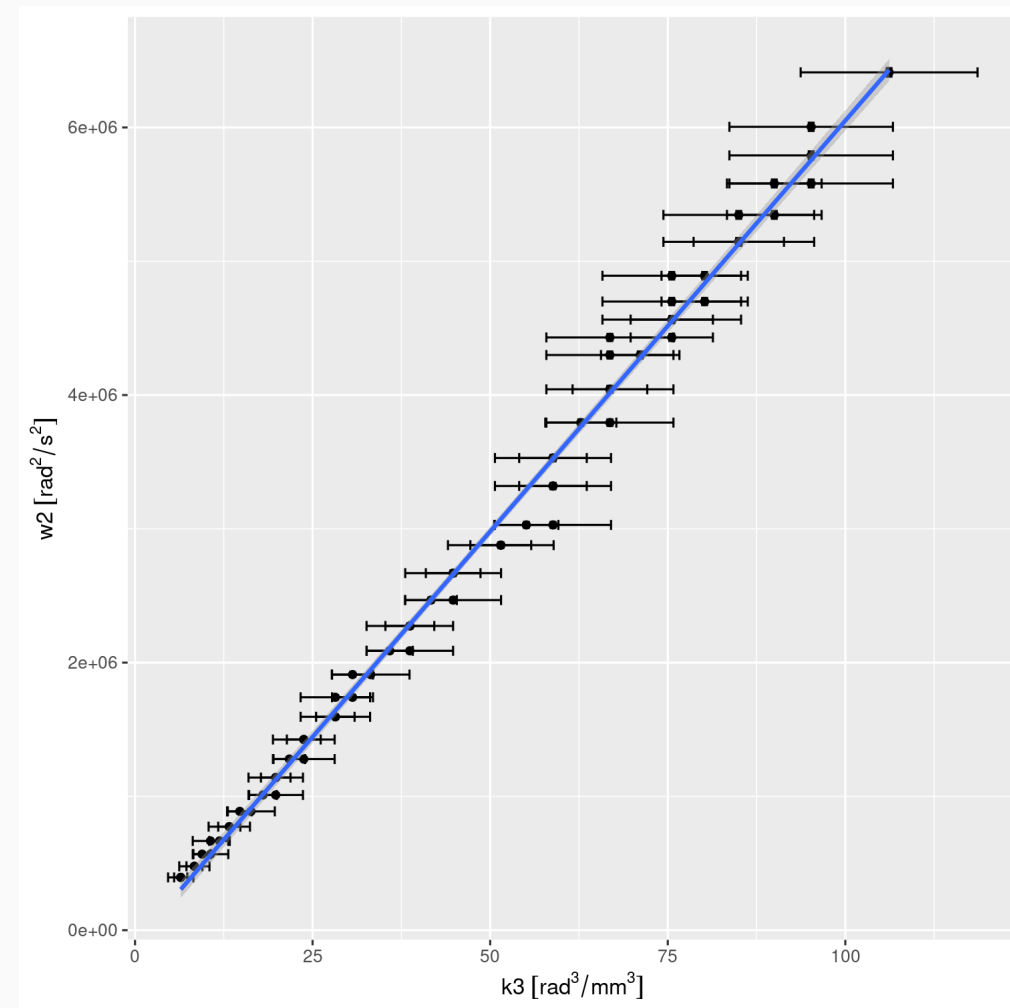
ggplot(df) + aes(k3, w2) +
  geom_errorbar(
    aes(ymin=errors_min(w2), ymax=errors_max(w2))) +
  geom_errorbarh(
    aes(xmin=errors_min(k3), xmax=errors_max(k3))) +
  geom_point() + geom_smooth(method="lm")
```

Measure the surface tension:

```
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>





- 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



- 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.



- 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>



Descarga de datos del Instituto Nacional de Estadística con R usando el servicio API JSON

Daniel Redondo-Sánchez, Miguel Ángel Luque Fernández, Miguel Rodríguez Barranco, Pablo Fernández-Navarro, María José Sánchez Pérez

Instituto de Investigación Biosanitaria de Granada (ibs.GRANADA),
Universidad de Granada

Registro de Cáncer de Granada, Escuela Andaluza de Salud Pública
CIBER de Epidemiología y Salud Pública



Escuela Andaluza de Salud Pública
CONSEJERÍA DE SALUD Y FAMILIAS



"Una manera de hacer Europa"

Objetivo

Describir un **método de descarga de información** del Instituto Nacional de Estadística (INE) usando su servicio API y el software R.

El código es totalmente reproducible y está disponible en GitHub:

github.com/danielredondo/INE_R

Utilización del servicio API del INE

Utilizamos el servicio API (*Application Programming Interface*) del INE para realizar la tarea de conexión e intercambio de datos.

1. **Obtenemos la dirección web** válida para la descarga, en función del tipo de información a descargar (por ejemplo, si es una tabla con número determinado, o si es un fichero *PCAxis*).
2. Después, procedemos a la **descarga de información** usando el comando GET del paquete **httr** (v1.4.0). El contenido se descarga en formato JSON (*JavaScript Object Notation*).
3. Se **procesa la información** con **dplyr** (v0.8.3), **rlist** (v0.4.6.1) y **data.table** (v1.12.4) hasta obtener un objeto data.frame para su fácil manipulación en R.

1. Obtención de URL




```
url <- "http://servicios.ine.es/wstempus/js/ES/DATOS_TABLA/ . . ."
```

1. Obtención de URL

Tipo 1: Número de tabla

URL descarga:
`http://ine.es/jaxiT3/Tabla.htm?t=9687`

URL API:
`http://servicios.ine.es/wstempus/js/ES/DATOS_TABLA/9687`



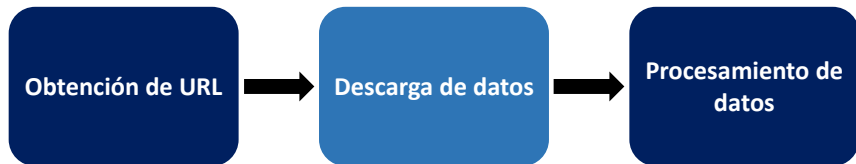
Tipo 2: Fichero PC-AXIS

URL descarga:
`http://ine.es/jaxi/Tabla.htm?path=/t15/p417/a2017/10/&file=01007.px`

URL API:
`http://servicios.ine.es/wstempus/js/ES/DATOS_TABLA/t15/p417/a2017/10/01007.px`



2. Descarga de información



```
library(httr)
datos_json <- GET(url)
datos_json$status_code
```



Si empieza por:	Suele indicar:
2 ó 3	Éxito
4	Error (del código)
5	Error (de la web)

3. Procesamiento de datos



```
datos <- content(datos_json) +
```

```
%>%  
  rlist::list.select  
  rlist::list.stack  
data.table::rbindlist
```

```
= data.frame
```

Ejemplo: procesamiento de defunciones fetales tardías

```
n <- length(defunciones_contenido)

for(i in 1:n){
  dato.i <- defunciones_contenido[[i]]
  defunciones.i <- list.select(dato.i$Data, Valor) %>% list.stack %>%
    cbind(codigo = dato.i$MetaData[[1]]$Codigo) %>%
    cbind(sexo = dato.i$MetaData[[2]]$Codigo) %>%
    cbind(s_gest = dato.i$MetaData[[3]]$Codigo)
  ifelse(i == 1,
    defunciones <- defunciones.i,
    defunciones <- rbindlist(list(defunciones, defunciones.i)))
}

head(defunciones)
```

##	Valor	codigo	sexo	s_gest
## 1:	1274	0193ixxiitodaslascausas	ambossexos	total
## 2:	196	0193ixxiitodaslascausas	ambossexos	menosde28semanas
## 3:	229	0193ixxiitodaslascausas	ambossexos	de28a31semanas
## 4:	355	0193ixxiitodaslascausas	ambossexos	de32a36semanas
## 5:	322	0193ixxiitodaslascausas	ambossexos	de37a41semanas
## 6:	0	0193ixxiitodaslascausas	ambossexos	42semanasyas

Aplicabilidad de uso de los datos descargados

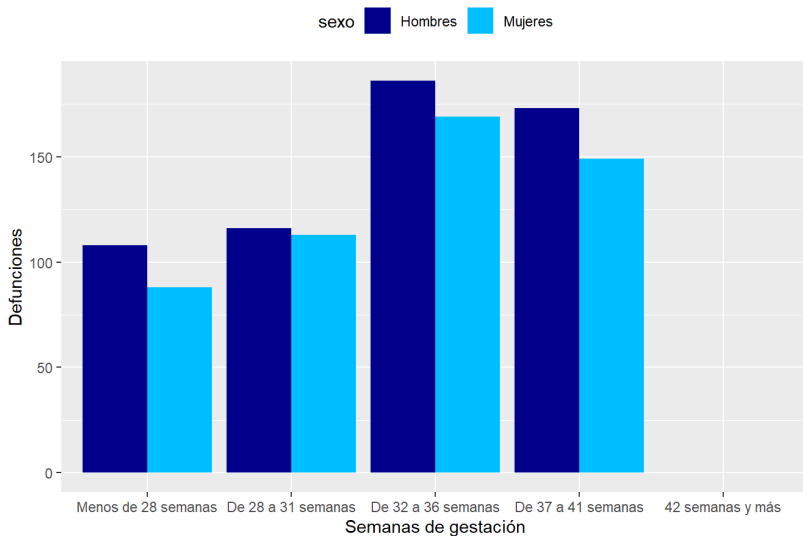
```
library(ggplot2)

datos <- subset(defunciones, defunciones$codigo == "01-93 I-XXII.Todas las causas" &
  defunciones$sexo != "Ambos sexos" &
  ! defunciones$s_gest %in% c("Total", "No consta"))

ggplot(datos, aes(x = s_gest, y = Valor, fill = sexo)) +
  scale_fill_manual(values=c("darkblue", "deepskyblue")) +
  geom_bar(stat = "identity", position = "dodge") +
  ylab("Defunciones") + xlab("Semanas de gestación") +
  ggtitle("Defunciones según semanas de gestación, por sexos") +
  theme(legend.position="top")
```

Aplicabilidad de uso de los datos descargados

Defunciones según semanas de gestación, por sexos



Fortaleza: **Fácil implementación**

En GitHub (github.com/danielredondo/INE_R) está disponible un **tutorial** (Rmd/html) con dos ejemplos de descarga:

- Defunciones fetales tardías por causas (lista perinatal), sexo y semanas de gestación, 2017.
- Población por provincias, por edad simple, 2017-2019.

Fortaleza: **Reproducibilidad**

Esta descarga permite **reproducibilidad** en análisis posteriores, algo importante para lograr **transparencia** en la publicación de resultados científicos.

Fortaleza: Descarga masiva de información

Haciendo uso de **156 URLs diferentes** (52 provincias \times 3 años) realizamos una descarga automática de **+6.000.000 de filas** (población por edad simple por secciones censales de los años 2010, 2011 y 2012).

```
[1] "Se han descargado 17556 filas de Araba/Álava"  
[1] "Se ha anexado la información de Araba/Álava"  
[1] "Se han descargado 19866 filas de Albacete"  
[1] "Se ha anexado la información de Albacete"  
[1] "Se han descargado 80982 filas de Alicante/Alacant"  
[1] "Se ha anexado la información de Alicante/Alacant"  
[1] "Se han descargado 27984 filas de Almería"  
[1] "Se ha anexado la información de Almería"  
[1] "Se han descargado 20724 filas de Ávila"  
[1] "Se ha anexado la información de Ávila"  
[1] "Se han descargado 36498 filas de Badajoz"  
[1] "Se ha anexado la información de Badajoz"
```

Limitación: Tiempo

Largos tiempos de espera en la descarga y procesamiento de información. Recomendable el uso de RStudio con **Google Cloud** o **Amazon Web Services (AWS)**.

¡Gracias!

Daniel Redondo Sánchez

✉ daniel.redondo.easp@juntadeandalucia.es

🌐 danielredondo.com

🔄 github.com/danielredondo

🐦 [@dredondosanchez](https://twitter.com/dredondosanchez)

Financiación: Instituto de Salud Carlos III (FIS PI18/01593 EU-FEDER)
Subprograma de Vigilancia Epidemiológica del Cáncer del CIBERESP



Escuela Andaluza de Salud Pública
CONSEJERÍA DE SALUD Y FAMILIAS



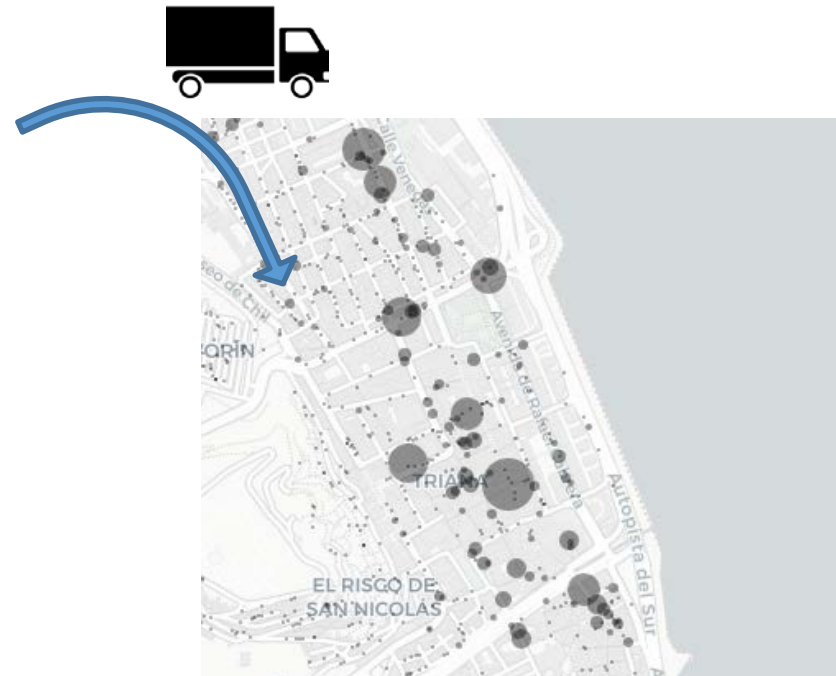
"Una manera de hacer Europa"

XI Jornadas de usuarios de R

Ajuste de modelos predictivos de series temporales para estimar los pedidos de un producto de consumo energético en una ciudad

Introducción

- Este es un proyecto relacionado con una empresa distribuidora de un producto relacionado con la energía.
- En este trabajo se muestra un estudio de modelos de predicción del número de pedidos realizados por clientes.



Modelos analizados: TBATS

- Se han analizado diversos modelos de series temporales que permiten predecir el número de pedidos.
- En primer lugar se ha analizado los modelos TBATS de series temporales (De Livera, Hyndman & Snyder, 2011).
- Estos modelos están implementado en la librería **forecast** (<https://github.com/robjhyndman/forecast>)
- TBATS permiten combinar de forma automatizada un modelo de regresión dinámica con términos de Fourier, que funciona como un modelo de espacio de estados de suavizado exponencial, y una transformación de Box-Cox.
- Los errores o residuales se modelizan como procesos ARIMA.
- Permiten introducir variables exógenas.
- También es posible ajustar múltiples periodos estacionales

De Livera, A.M., Hyndman, R.J. and Snyder, R.D. (2011) Forecasting Time Series with Complex Seasonal Patterns Using Exponential Smoothing. Journal of the American Statistical Association, 106, 1513-1527.

<https://doi.org/10.1198/jasa.2011.tm09771>

Modelos analizados: TBATS

- El modelo TBATS se expresa de la forma siguiente:

$$y_t = \gamma x_t + \sum_{k=1}^K \alpha_k \sin\left(\frac{2\pi kt}{m}\right) + \beta_k \cos\left(\frac{2\pi kt}{m}\right) + \varepsilon_t$$
$$\phi(B)\nabla^d \varepsilon_t = \theta(B)z_t$$

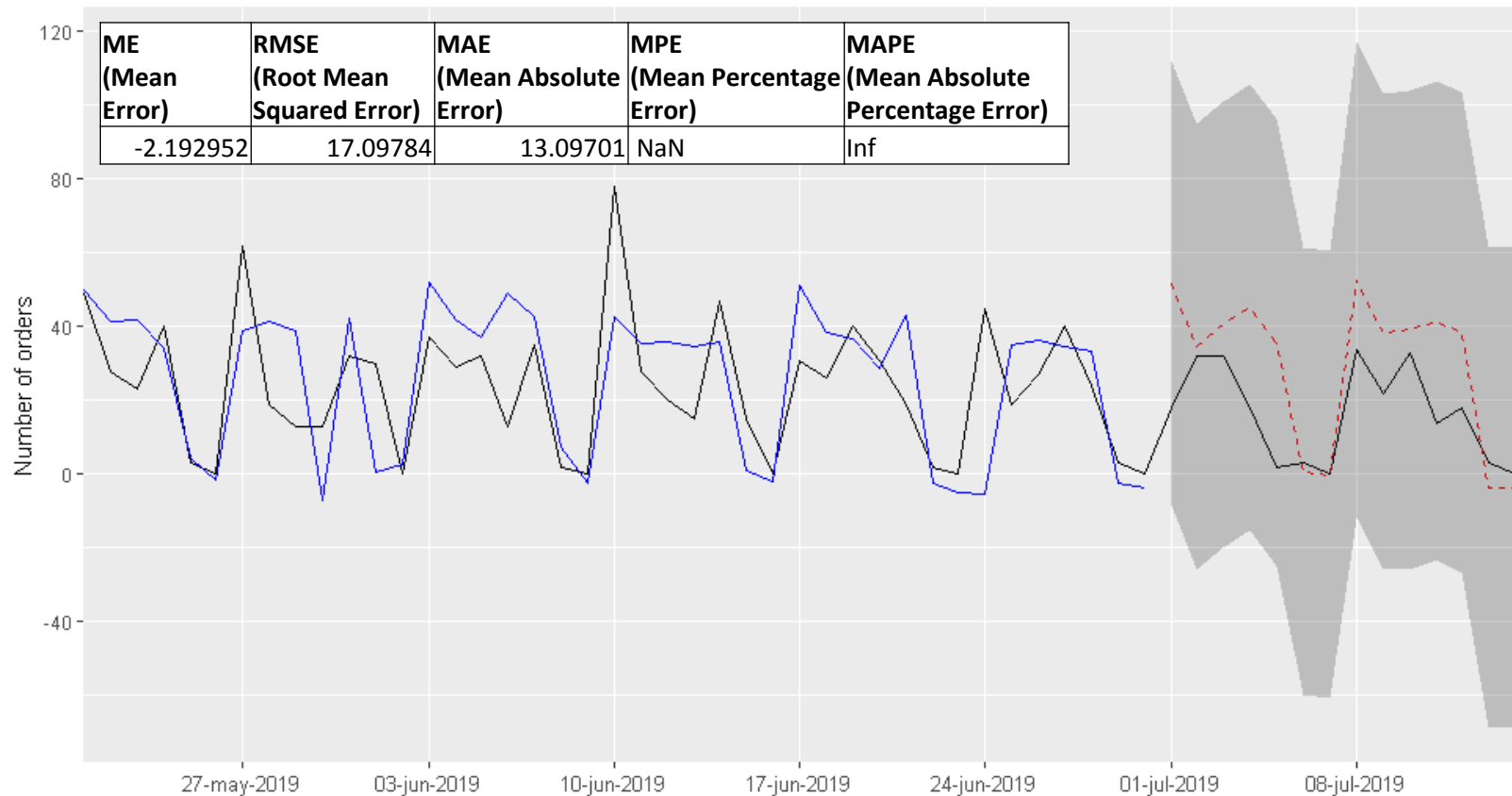
- m es el parámetro de periodo estacional, $\phi(B)$ y $\theta(B)$ son los términos AR y MA, z_t es ruido blanco y x_t denota la covariable en tiempo t .
- Nosotros hemos ajustado estos modelos considerando el efecto calendario (festivos y fines de semana).
- Buscamos (de forma iterativa) el número de términos de Fourier óptimo que permita minimizar el criterio AIC

Ejemplo de ajuste con TBATS

- En un municipio determinado y para un tipo de producto dado

Predicting number of orders of XXX product in municipality WWW

Regression with ARIMA(0,1,1) errors and weekly-monthly ($m=7$ and $m=30$) periodicity



Modelos analizados: Prophet

- También hemos utilizado el procedimiento Prophet, de Facebook (<https://facebook.github.io/prophet/>).
- Prophet, que está disponible tanto en R como Python (Taylor & Letham, 2018), está basado en modelos aditivos que ajusta las tendencias no lineales con estacionalidades anuales, semanales y diarias y considera también el efecto calendario (festivos).
- Trabaja bien con series que tienen fuertes efectos estacionales y varias temporadas de datos históricos.
- Prophet es bastante robusto a datos perdidos y desviaciones en la tendencia, y maneja bastante bien los outliers.
- En R el procedimiento está implementado en la librería **prophet**.

Sean J. Taylor, Benjamin Letham (2018) Forecasting at scale. The American Statistician 72(1):37-45
<https://peerj.com/preprints/3190.pdf>.

Modelos analizados: Prophet

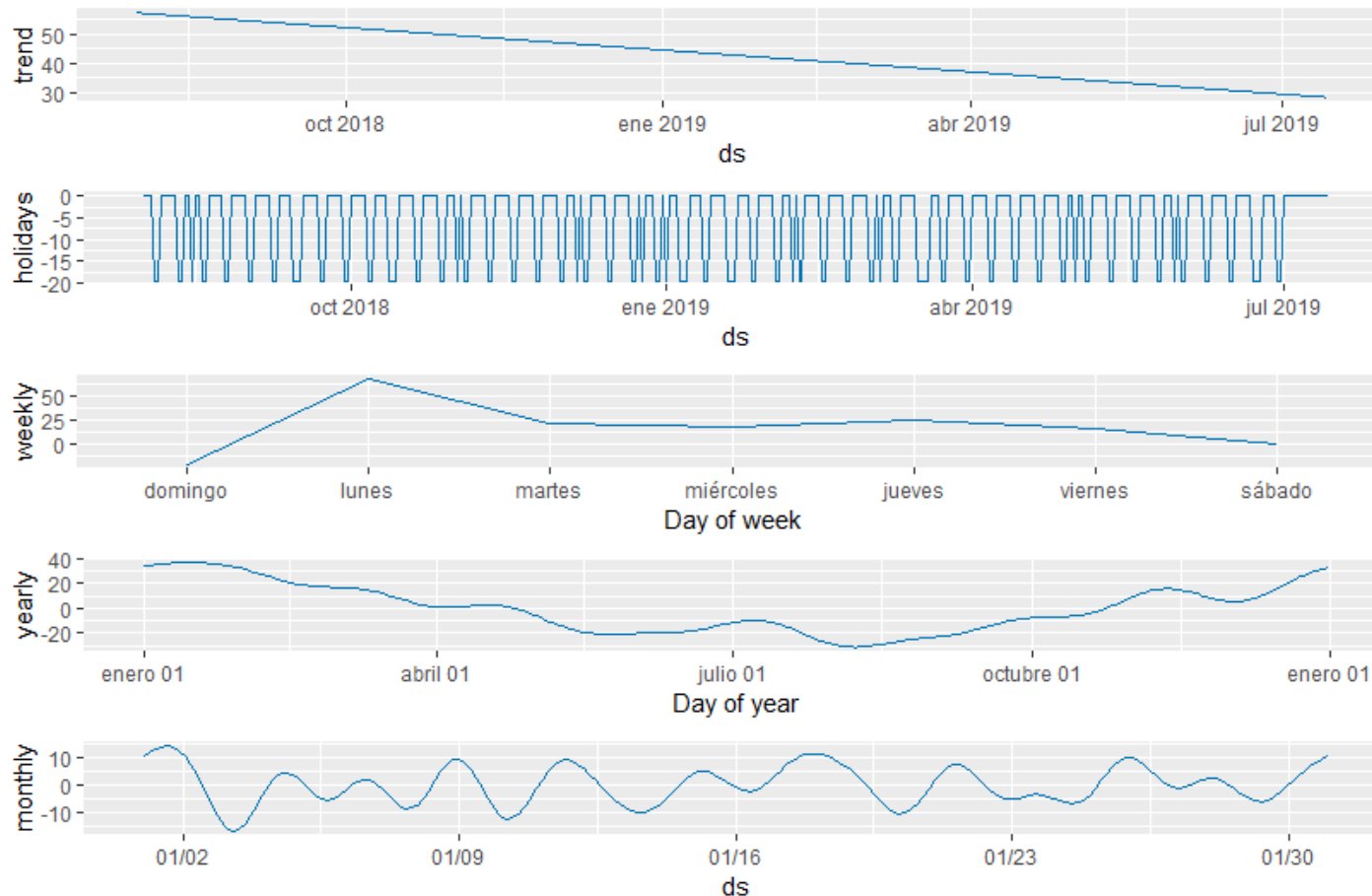
- En este caso, la expresión de carácter aditivos sería

$$y_t = g_t + s_t + h_t + \varepsilon_t$$

- donde g_t modeliza la componente de la tendencia con los cambios no periódicos (por ejemplo, puede ser una componente de regresión con respecto a t), s_t representa la componente estacional (modelizada también como series de Fourier) y h_t recoge la información del efecto calendario (vacaciones, festivos, eventos,...).
- Es un procedimiento muy utilizado para series de valores diarios.
- Además, permite introducir puntos temporales de cambio abrupto en la tendencia de las series (changepoints).

Modelos analizados: Prophet

- La librería dispone de métodos para representar las distintos componentes de la serie.

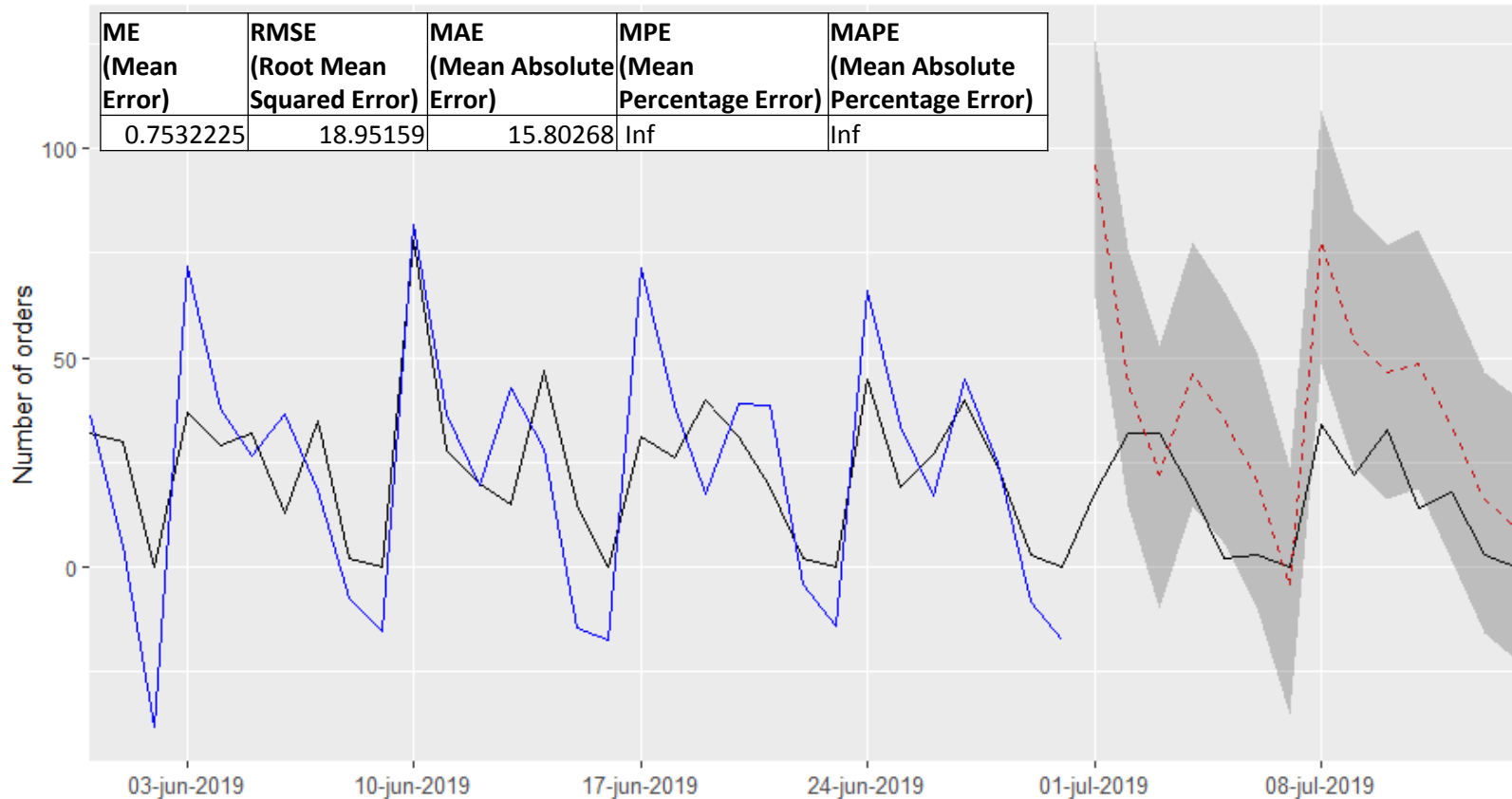


Ejemplo de ajuste con Prophet

- Considerando el mismo ejemplo de antes

Predicting number of orders of XXX product in municipality WWW

Regression with Prophet library and weekly-monthly periodicity



Líneas de trabajo

- De momento, tenemos resultados similares en los modelos que ajustan una serie de valores diarios de un año entero.
- De forma progresiva se dispondrá de más datos adicionales.
- También queremos estudiar los modelos LSTM (Long Short Term Memory) con el fin de evaluar y comparar los resultados obtenidos hasta el momento.
- LSTM (Sherstinsky, 2018) tiene la ventaja de utilizar Tensorflow con Keras y ya hay librería en R que lo implementan.
- La posible ventaja que tiene el uso de un red neuronal es que podríamos capturar efectos no lineales e interacciones entre variables.

Sherstinsky, Alex (2018). “Fundamentals of Recurrent Neural Network (Rnn) and Long Short-Term Memory (Lstm) Network,” August. URL: <http://arxiv.org/abs/http://arxiv.org/abs/1808.03314v4>.

Muchas gracias

Ponente: Carlos Pérez González (cpgonzal@ull.es)