

¿Cómo migrar de Excel a R?

Valery Zúñiga - Carolina Montoya

Motivación

- Un mercado que demanda análisis de datos de forma eficiente.
- Automatización de procesos.
- ¿Por qué R? Costo de oportunidad de lo tradicional.

Caso de estudio.

Provisión matemática. *Actuarial Mathematics*, Bowers et al (1984)

$${}_kV_x = A_{x+k} - P_x \ddot{a}_{x+k}$$

- Periodicidad del cálculo.
- Flexibilidad al cambio de parámetros. (Ejemplo: la tasa de interés.)
- Eficencia.

Paquetes recomendados: tidyxl & unpivotr

A	B	C	D	E	F	G	H	I	J	K
Année 2014, semaine et weekend		Population âgée de 10 à 74 ans			Selon le genre					
		Ensemble			Hommes			Femmes		
		Total	Population ayant réalisée l'activité		Total	Population ayant réalisée l'activité		Total	Population ayant réalisée l'activité	
		Durée	Durée %		Durée	Durée %		Durée	Durée %	
0	Besoins physiologiques	11:35	11:35	100,0	11:28	11:28	100,0	11:43	11:43	100,0
01	Dormir	08:46	08:46	99,9	08:44	08:45	99,9	08:48	08:48	100,0
02	Manger et boire	01:53	01:54	99,7	01:52	01:53	99,7	01:54	01:54	99,7
03	Soins personnels et médicaux à domicile	00:56	00:57	97,5	00:51	00:53	97,2	01:01	01:02	97,9

```
## # A tibble: 6 x 21
##   sheet address row col is_blank data_type error logical numeric
##   <chr> <chr>   <int> <int> <lgl>   <chr>   <chr> <lgl>   <dbl>
## 1 Index A1      1     1 FALSE   character <NA>   NA      NA      NA
## 2 Index B1      1     2 TRUE    blank    <NA>   NA      NA      NA
## 3 Index C1      1     3 TRUE    blank    <NA>   NA      NA      NA
## 4 Index D1      1     4 TRUE    blank    <NA>   NA      NA      NA
## 5 Index E1      1     5 TRUE    blank    <NA>   NA      NA      NA
## 6 Index F1      1     6 TRUE    blank    <NA>   NA      NA      NA
## # ... with 12 more variables: date <dtm>, character <chr>,
## #   character_formatted <list>, formula <chr>, is_array <lgl>,
## #   formula_ref <chr>, formula_group <int>, comment <chr>, height <dbl>,
## #   width <dbl>, style_format <chr>, local_format_id <int>
```

Transición Excel - R: librerías de ayuda.

Paquetes utilizados	Descripción
dplyr	Manipulación de datos.
data.table	Como alternativa al data.frame.
lubridate	Trabajo con fechas.
readxl	Lectura de archivos de excel.
openxlsx	Impresión en excel.
readr	Lectura de archivos csv y txt.

Herramienta de conversión: VBA a R.

#VBA

```
Public Function prob_Sobrev(prob_muerte As Range,  
                             x, inic, omega)
```

```
For j = 1 To omega - x
```

```
p(j + 1) = p(j) * (1 - prob_muerte(x + j - inic))
```

```
Next
```

```
End Function
```

#R

```
prob_Sobrev <- function(prob_muerte , x, inic, omega){
```

```
for(j in 1:(omega - x)){
```

```
p(j + 1) = p(j) * (1 - prob_muerte(x + j - inic))
```

```
}
```

```
}
```

Herramienta de conversión: VBA a R.

#VBA

```
k = Fix(ant)           'Antigüedad entera
x = ed + k             'Edad actual del cliente
```

```
If (k < n) And (x <= omega) Then
```

#R

```
k = floor(ant)         #Antigüedad entera
x = ed + k             #Edad actual del cliente
```

```
if((k < n)&(x <= omega)){
```

Herramienta de conversión: VBA a R.

#VBA

```
Prov_gasto = WorksheetFunction.Max((A + B * C * ((1 + jp) ^  
    Fix(WorksheetFunction.Min(ant, 60 - ed - 1), 0)))  
Else  
Prov_gasto = 0  
End If  
End Function
```

#R

```
Prov_gasto = max((A + B * C * ((1 + jp) ^  
    floor(min(ant, 60 - ed - 1), 0)))  
}  
Prov_gasto = 0  
}  
}
```


Programación vectorial

¿Qué entendemos por programación vectorial?

#No vectorial

```
Prov_gasto = max((A + B * C * ((1 + jp)
                  ^ floor(max(min(ant, 60 - ed - 1), 0))))
```

#Vectorial

```
Prov_gasto = pmax((A + B * C * ((1 + jp)
                  ^ floor(pmax(pmin(ant, 60 - ed - 1), 0))))
```

Programación vectorial

#No vectorial

```
prob_Sobrev <- function(prob_muerte , x, inic, omega){  
  p <- c()  
  p[1] <- 1  
  for(j in 1:(omega - x)){  
    p(j + 1) = p(j) * (1 - prob_muerte(x + j - inic))  
  }  
}
```

#Vectorial

```
prob_Sobrev_Vec <- function(prob_muert , x, inic, omega){  
  j <- mapply(":", 1, omega - x)  
  sapply(1:length(omega),function(i)  
    c(1, cumprod(1 - prob_muert[[i]][x[i] + j[[i]] - inic[i]])))  
}
```

Estructura de la programación.

- Cálculo por fila:

```
df %>%  
  rowwise() %>%  
  mutate(provision = funcion(var1, var2...)) %>%  
  ungroup()
```

- Paralelo:

```
vector <- unlist(mclapply(1:nrow(df),  
                          function(i) funcion(var1[i], var2[i]...),  
                          mc.cores = 4))
```

Estructura de la programación.

- pmap:

```
vector <- pmap_dbl(list(var1, var2...), funcion)
```

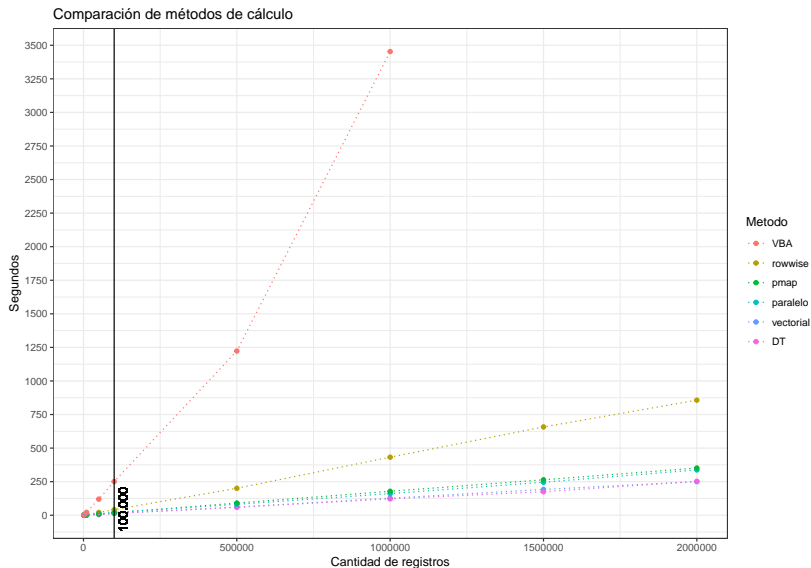
- Vectorial con dplyr:

```
df %>%  
  mutate(provision = funcion_vectorial(var1, var2...))
```

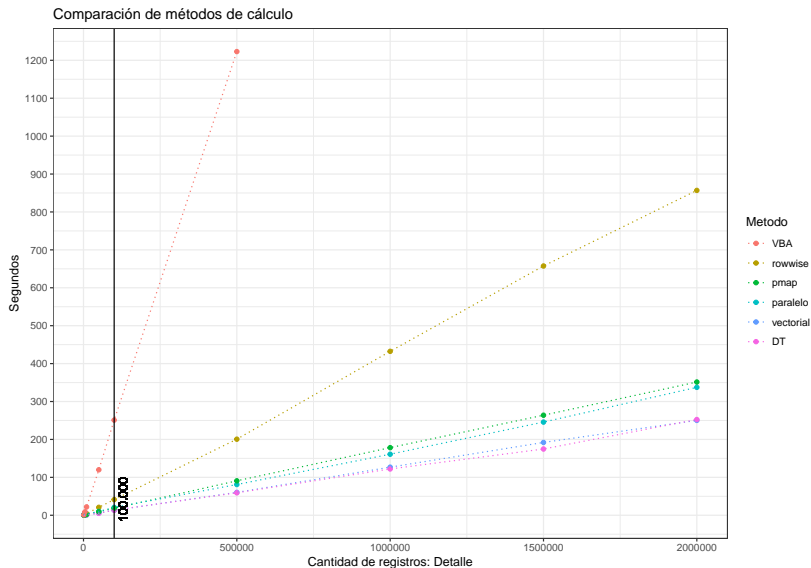
- Vectorial con DT:

```
DT[,provision := funcion_vectorial(var1, var2...)]
```

Tiempos de ejecución: VBA vs métodos en R.

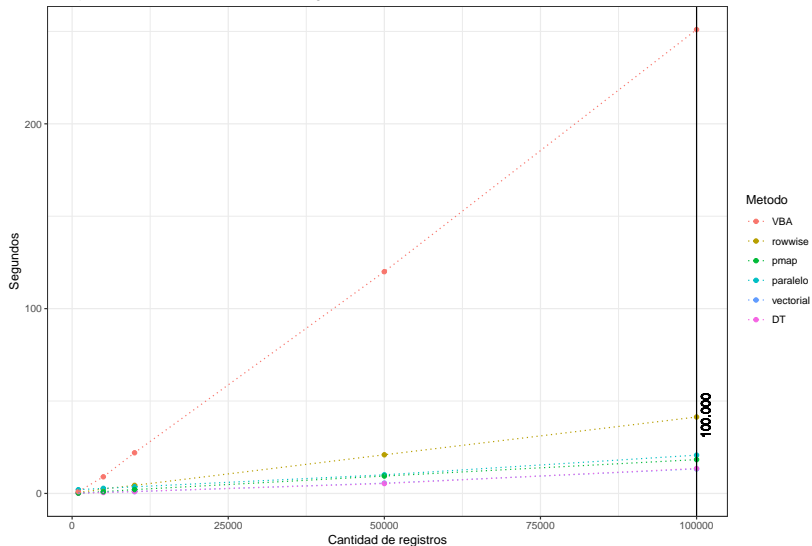


VBA vs métodos en R. Escala reducida.



VBA vs métodos en R. Hasta 100.000 registros.

Comparación de métodos de cálculo: Registros menores a 100.000



Ahora de R a Excel: openxlsx

```
wb <- loadWorkbook("Provision.xlsx")

addWorksheet(wb, "Provision calculada")

writeData(df, sheet = "Provision calculada",
          startCol = 1,
          startRow = 2)

saveWorkbook(wb, file = "Provision.xlsx", overwrite = TRUE)

openXL("Provision.xlsx")
```


Referencias.

- Spreadsheet Munging Strategies, Duncan Garmonsway.
- Going from a human readable Excel file to a machine-readable csv with {tidyxl}, Bruno Rodrigues.
- <https://regexone.com/>
- Advanced R, Hadley Wickman.
- Parallel R, Q. Etahn McCallun & Stephen Weston.