



EXTENDING CRAN PACKAGES WITH BINARIES: THE EXAMPLE OF X13BINARY

Dirk Eddelbuettel and Christoph Sax

useR! 2016 at Stanford, CA

X13BINARY (AND SEASONAL)

WHAT IS SEASONAL ADJUSTMENT?

trend / cycle

- long term trend

seasonal

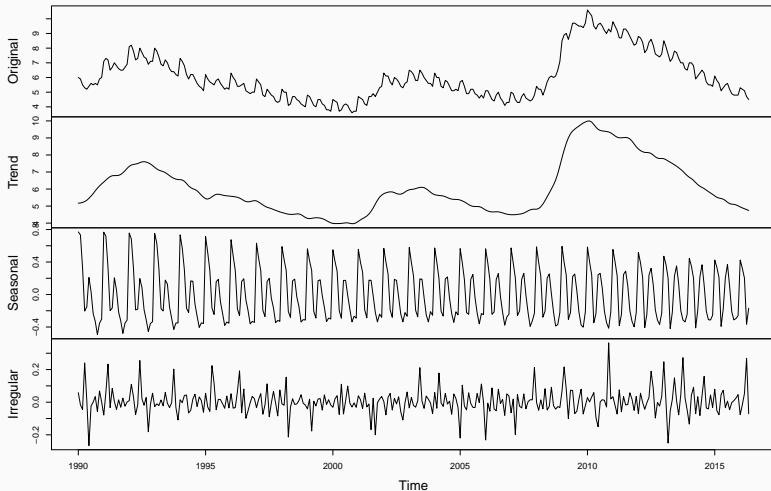
- intra-year fluctuations repeated regularly

irregular

- Random fluctuations not explained by previous components

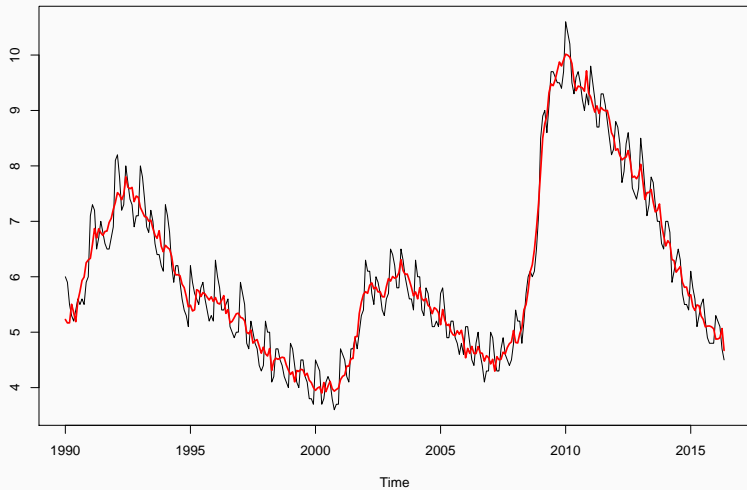
EXAMPLE: US UNEMPLOYMENT RATE I

decomposition



EXAMPLE: US UNEMPLOYMENT RATE II

Original and Adjusted Series



Mainly

- To recover the underlying core 'signal'
- Remove *predictable* seasonal pattern

Why do this? Why not y/y comparisons?

- Turning points
- Calendar effects

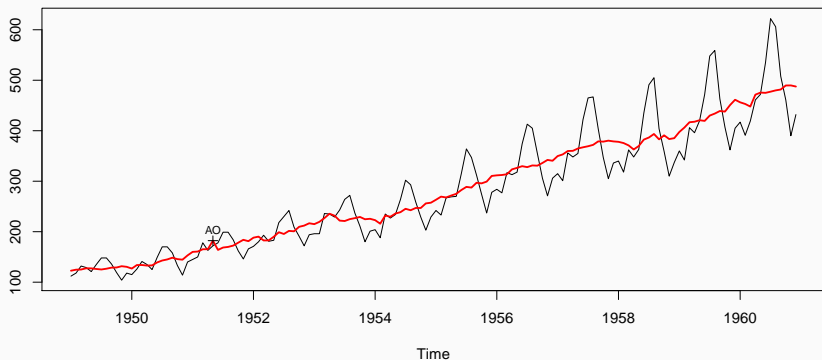
Key Aspects

- Seasonal Adjustment Software by the US-Census Bureau
- Combination of X-12 and TRAMO/SEATS (Bank of Spain)
- Written in Fortran, delivered as an executable (and its source)
- X-13 (and predecessors) used in almost all (official) statistical offices around the world

A MINIMAL EXAMPLE

```
library(seasonal)           # depends on x13binary  
m <- seas(AirPassengers); plot(m)
```

Original and Adjusted Series



Online Seasonal Adjustment: www.seasonal.website

seasonal.website

SEASONAL R INTERFACE TO X-13

Workbench Introduction Examples Examples Up/Download

Options

Adjustment Method: SEATS

Pre-Transformation: AIC Test

Arima Model: Auto Search

Outlier: Auto Critical Value

Holiday: AIC Test Easter

Trading Days: AIC Test

Output

Original and Adjusted Series

Jan 1955: original: 242 adjusted: 263.27

R Call

```
seas(  
  x = AirPassengers  
)
```

Summary

Coefficients		Stats		Tests	
Weekday	-0.00	Adjustment	SEATS	QS	0
Easter[1]	0.02	ARIMA	(0 1 1)(0 1 1)	H0: no seasonality in final series	
AO1951.May	0.10	Obs.	144	Box-Ljung	26.65
MA-Nonseasonal-01	0.12	Transform	log	H0: no residual autocorrelation	

GITHUB COMPLEMENTING CRAN

In a nutshell

- `seasonal` package makes it easy to work with `X13-ARIMA`
- However, it *still* requires the user to do manual installation
- How large a drop-off in users does this create?
- Now `CRAN` is *wonderful*:
 - quality, curation, checks, availability
 - we just press a button or run an installer function
 - and it all *just works*

Our insight

- Leverage the strength of [CRAN](#) and R
- Provide an *installer package* which when installed will complement with a 'hidden' binary
- Other key idea: [GitHub](#) as a (web, file) server

Key Steps

- A GitHub repository [x13prebuilt](#) is set up containing the binaries of `x13ashtml` for the three key OSs we care about
- A CRAN package [x13binary](#) (with corresponding code in a [GitHub repo](#)) then relies on the former and copies binaries as needed (upon package creation via the **configure** step)
- We added additional bells and whistles for checks once installed etc pp

Simple, really

- `x13binary` is now a CRAN package, available globally and in the repository system
- *Clients* such as `seasonal` (and therefore second-degree clients like `gunsales` and `ggseas`) just depend on it
- `x13binary` then utilizes `x13prebuilt` to provide `x13ashtml` during 'compilation' of the `x13binary` package
- CRAN does not offer (externally-built) binaries (for security reasons) but allows access to them during compilation step
- *And it all just works*

SUMMARY

Takeways

- De-seasonalisation is very powerful, very widely used, and implemented *once* by US Census
- Use from R was already easy and convenient thanks to [seasonal](#)
- [but manual / tedious to set up](#)
- By leveraging CRAN and GitHub infrastructure all steps can now be automated

Takeways

- De-seasonalisation is very powerful, very widely used, and implemented *once* by US Census
- Use from R was already easy and convenient thanks to [seasonal](#)
- By leveraging CRAN and GitHub infrastructure all steps can now be automated