Seasonal adjusment on the fly with X-13ARIMA-SEATS, seasonal and ggplot2

At a glance:

I show how to seasonally adjust published electronic card transactions spend in New Zealand using the US Census Bureau's excellent X-13ARIMA-SEATS software, the Spanish SEATS algorithm and Christoph Sax's seasonal R package; and how to build a new "stat" for ggplot2 to make it easy to do seasonal adjustment on the fly for a graphic of a time series split by various grouping dimensions.

10 Oct 2015

Calling seasonal adjustment software from R

I recently explored for the first time (having languished on the “check this out later” list) Christoph Sax (http://www.christophsax.com/)’s excellent `seasonal` R package (https://cran.r-project.org/web/packages/seasonal/index.html). It makes it super easy for R users to engage with X-13ARIMA-SEATS, the latest industry standard software for time series analysis and in particular seasonal adjustment of official statistics series. It’s a huge step forward in ease of use from the `x12` package which was a good interface to X-12ARIMA but never felt to me as R-native as `seasonal` does; I was always conscious of X-12ARIMA in the background. For example, to control for Chinese New Year (important at my work), you had to save a text file with the relevant dates encoded in it, rather than pass it directly to the function in an R-native way.

`seasonal` calls on the latest US Census Bureau software X-13ARIMA-SEATS which has an excellently informative website (https://www.census.gov/srd/www/x13as/) and free downloads. See the definitive reference manual (https://www.census.gov/ts/x13as/docX13ASHTML.pdf).

To avoid confusion, some terminology:

- ARIMA stands for “autoregressive integrated moving average” and is one of a class of models for time series
- X11, originally the name of software by the US Census Bureau and taken up by Statistics Canada, now usually refers to the X11 *method* for seasonal adjustment (or other inference) via ARIMA modelling first developed in the 1960s (Shiskin, Young and Musgrave in 1967)
- SEATS is Signal Extraction in ARIMA Time Series and is an alternative *method* for seasonal adjustment (or other inference) via ARIMA modelling, developed by Gomez and Maravall at Bank of

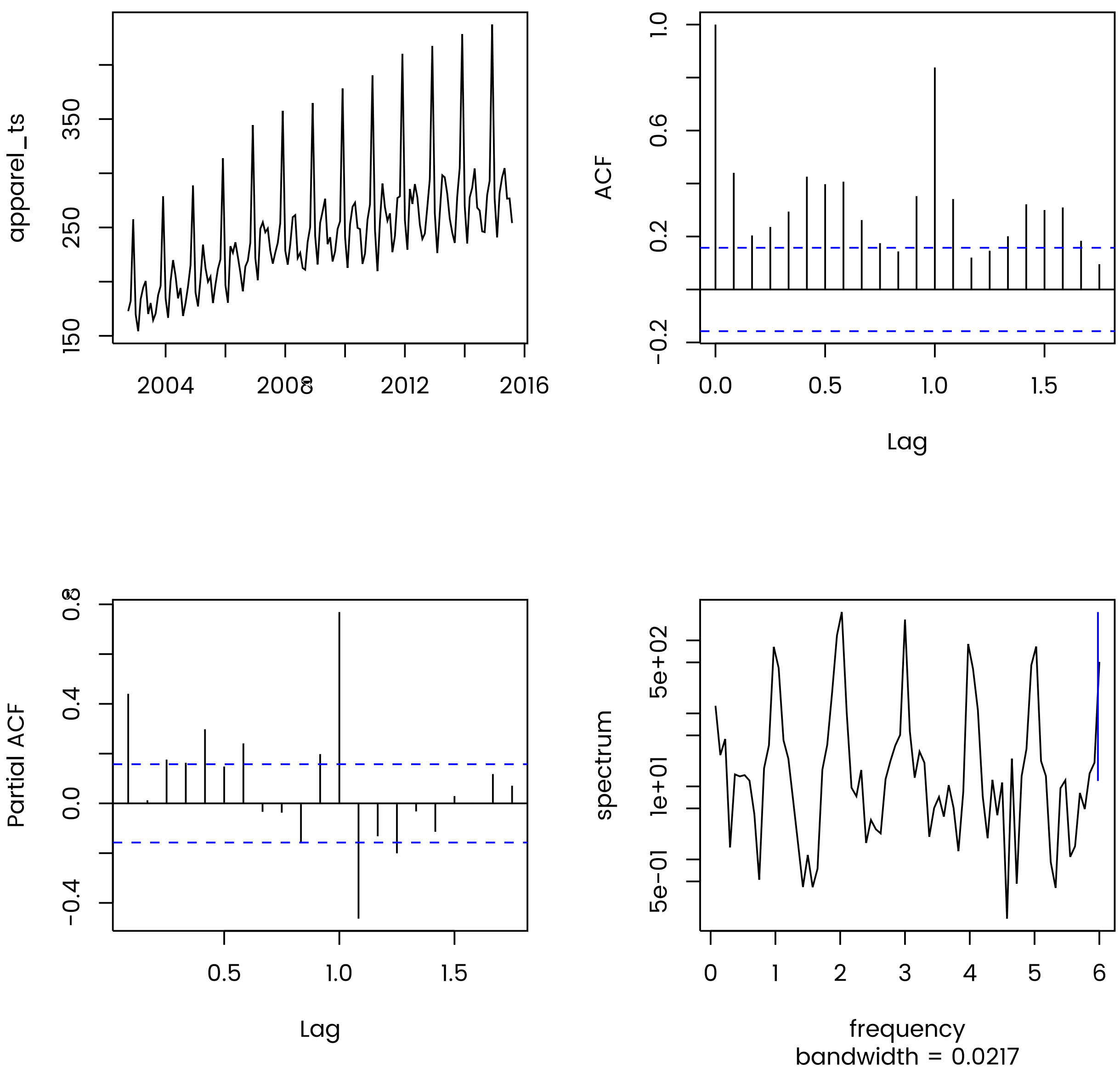
Spain (various 1990s references)

- X13-ARIMA-SEATS is the US Census Bureau's latest program that implements both the X11 and SEATS methods plus some additional diagnostic and model strategy tools
- `seasonal` is an R package that acts as a front end to X13-ARIMA-SEATS and gives all the needed functionality including both the X11 and SEATS methods without leaving the R environment

Electronic card transactions in New Zealand

I'll demonstrate this functionality generously made available by the US Census Bureau and Sax with electronic card transactions spend in New Zealand, published by Statistics New Zealand (http://www.stats.govt.nz/browse_for_stats/businesses/business_characteristics/electronic-card-transactions-info-releases.aspx). The data are published on a monthly basis from October 2002 to (at the time of writing) August 2015.

To get started I look at a single subset of the data, spend in millions of New Zealand dollars on apparel. Here's the basic data, its autocorrelation function, partial autocorrelation function, and spectrum.



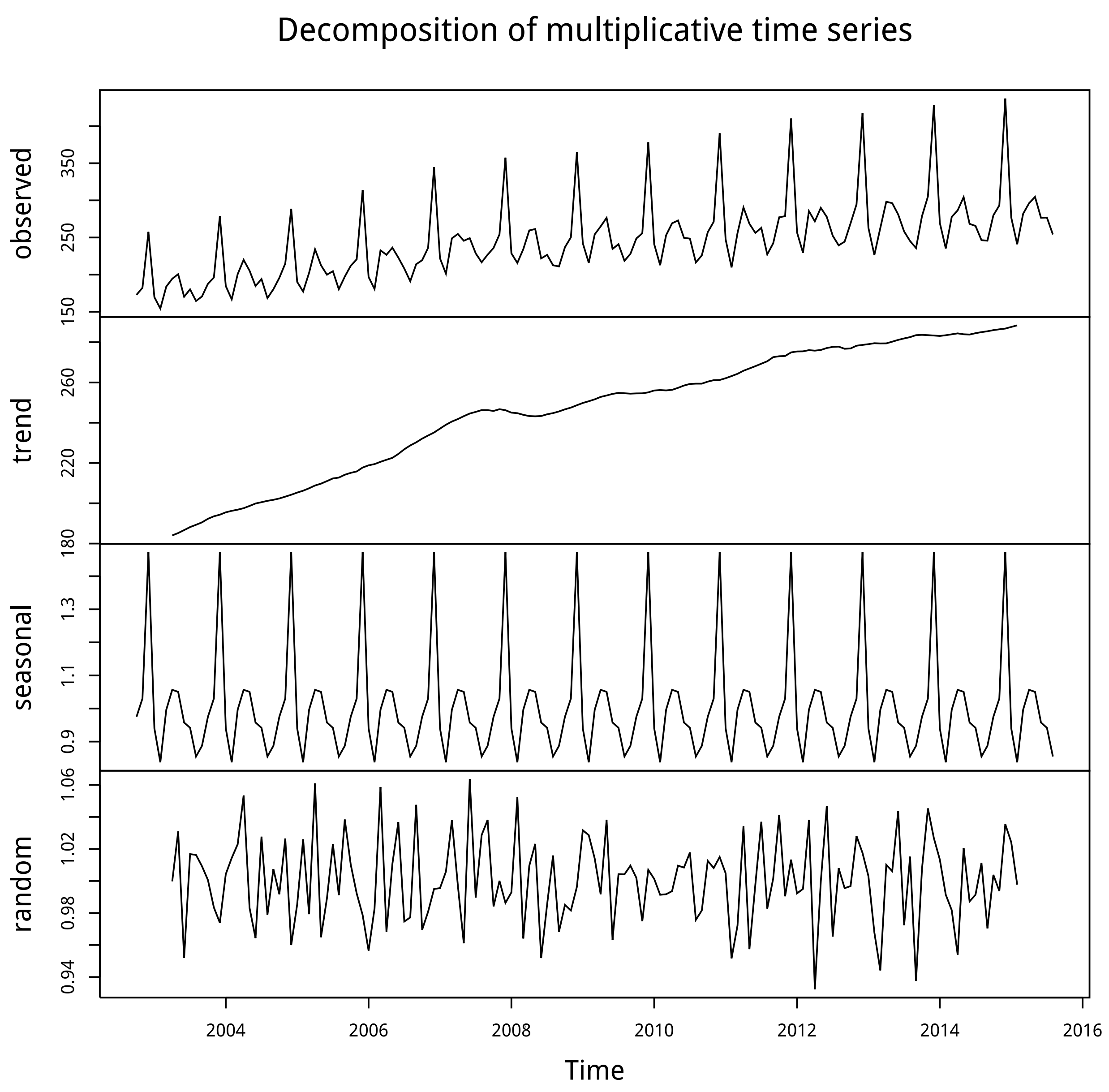
Key characteristics include:

- there's an obvious and unsurprising trend upwards - it's not a stationary time series
- the variance increases as the series increases, so some form of transformation (probably logarithm, which I checked and works well) needed if we want methods where the variance doesn't vary with the trend
- there's a strong partial autocorrelation at 1.0 (ie annual, showing spend in a particular month is correlated with spend in the same month a year before) and at 1/12 (showing spend in a particular month is also correlated with spend in the immediately previous month).

Here's the code for downloading this data from where I've stashed a clean version (available for you all) and producing those basics:

```
library(seasonal)
library(tidyr)
library(dplyr)
library(showtext)
library(ggplot2)
library(scales)

# set up fonts etc
font.add.google("Poppins", "myfont")
showtext.auto()
theme_set(theme_light(base_family = "myfont"))

# set path to X13-SEATS and check it's working
Sys.setenv(X13_PATH = "c:/winx13/x13as")
checkX13()

# download file with data ultimately from Statistics New Zealand's Infoshare, prepared
# earlier: "Values - Electronic card transactions A/S/T by industry group (Monthly)"
#
download.file("https://github.com/ellisp/ellisp.github.io/blob/master/data/Electronic card tr
              mode = "wb", # ie binary
              destfile = "tmp.rda")
load("tmp.rda")
head(ect)

apparel <- ect %>%
   filter(group == "Apparel")

apparel_ts <- ts(apparel$Value, start = c(2002, 10), frequency = 12)

par(mfrow = c(2, 2), family = "myfont")
plot(apparel_ts, xlab = "")
acf(apparel_ts, main = "")
pacf(apparel_ts, main = "")
spectrum(apparel_ts, main = "")
```

Here’s the classic decomposition into trend, seasonal and random components that can be multiplied together to form the original series. I use multiplicative decomposition because of the way the variance increases as the mean increases.




```
plot(decompose(apparel_ts, type = "multiplicative"))
```

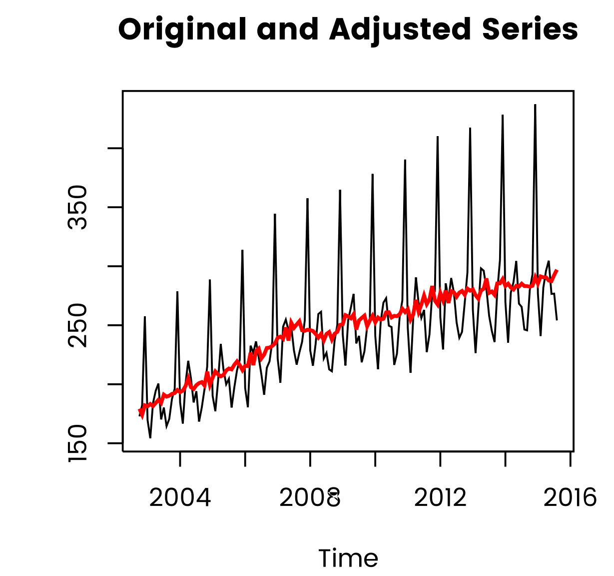
A simple form of seasonal adjustment can be performed by multiplying that trend value in the plot above by the random value which can be seen to hover around 1.0 (or, equivalently, dividing the original value by the seasonal values). This is a bit simplistic however; most notably it means the multiplier for a month stays the same over time, which is unlikely to be true over longer periods (although it might be a reasonable approximation for this relatively short series).

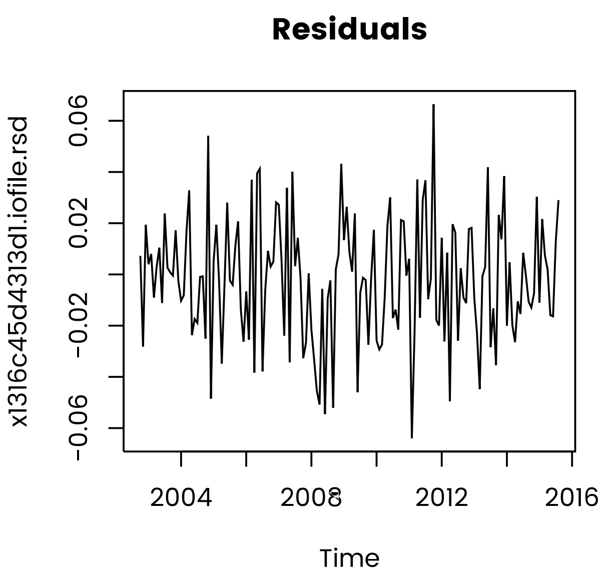
SEATS seasonal adjustment

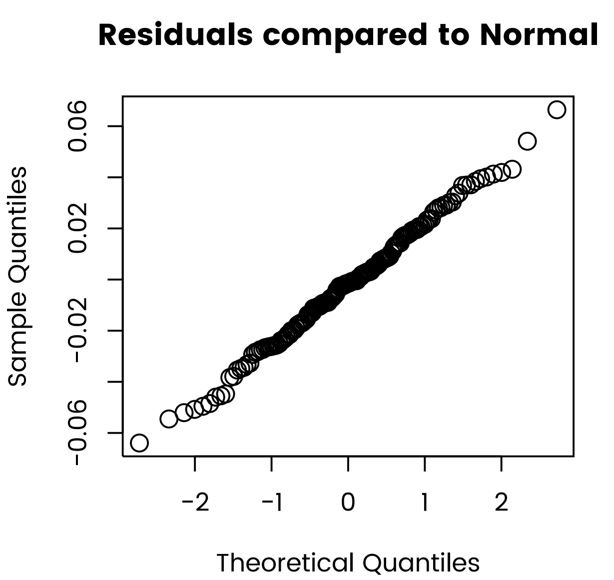
To perform a more sophisticated seasonal adjustment it’s best to go to the best in breed software, which is the US Census’ X-13ARIMA-SEATS. Sax’s seasonal package makes it super-easy to fit a model and extract the seasonally adjusted values and residuals for diagnostic purposes. In fact it’s a one liner - a call to `seas()` , which gives us very sensible defaults using the SEATS method:

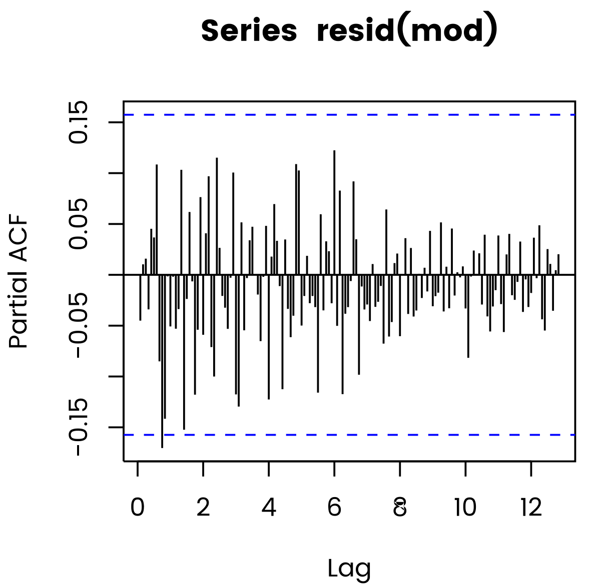
- automatically fits a seasonal ARIMA model based on the frequency defined in the `ts` object it is fed
- detects outliers and in effect leaves them out of the calculation of the seasonal effects (turns out not to be important in this dataset)
- detects any impact of number of trading days per month and Easter, and creates external regressors for them if necessary
- detects if a transformation is necessary and chooses a good one if necessary (in this case it opts for logarithmic as we'd guessed)

```
mod <- seas(apparel_ts)

plot(mod)
plot(resid(mod), main = "Residuals")
qqnorm(resid(mod), main = "Residuals compared to Normal")
pacf(resid(mod), "Partial ACF of residuals")
```







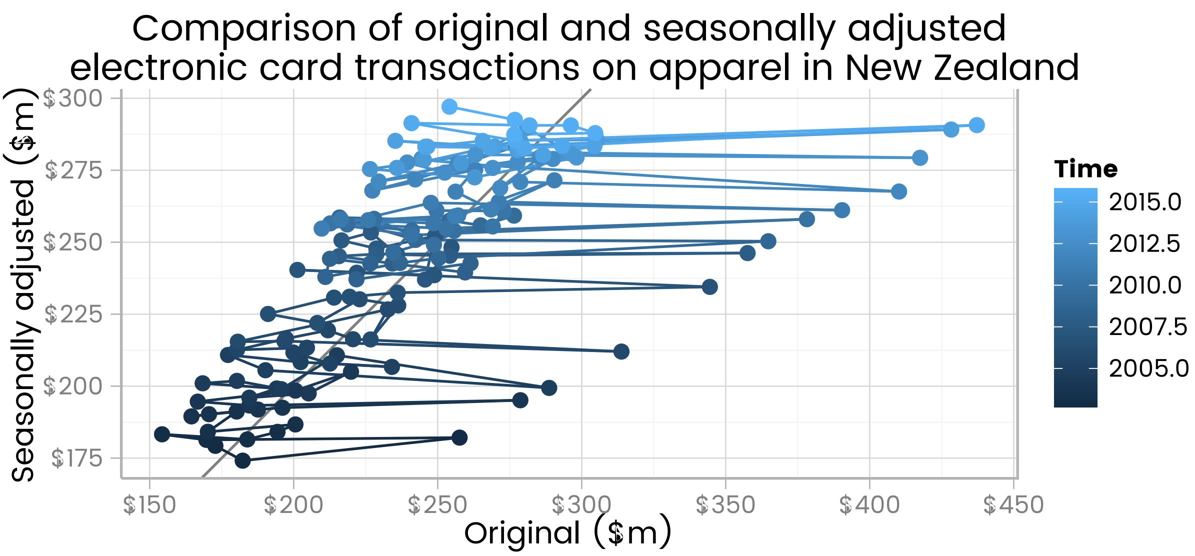


Super simple.

The main reasons we're doing this from R rather than hand coding a X-13 .spc file are:

- its data management ease and flexibility,
- access to a wide class of analytical functions (as used in my initial explorations), and
- graphics.

So the basic workflow will be to take the results from our SEATS model and manipulate them in R for re-use, whether performing diagnostic statistical tests or just visuals to understand the data. For example, the connected scatter plot below lets us see how the original values get adjusted down (those in the right bottom triangle of the plot) or up (those in the left upper triangle) in the seasonal adjustment process.



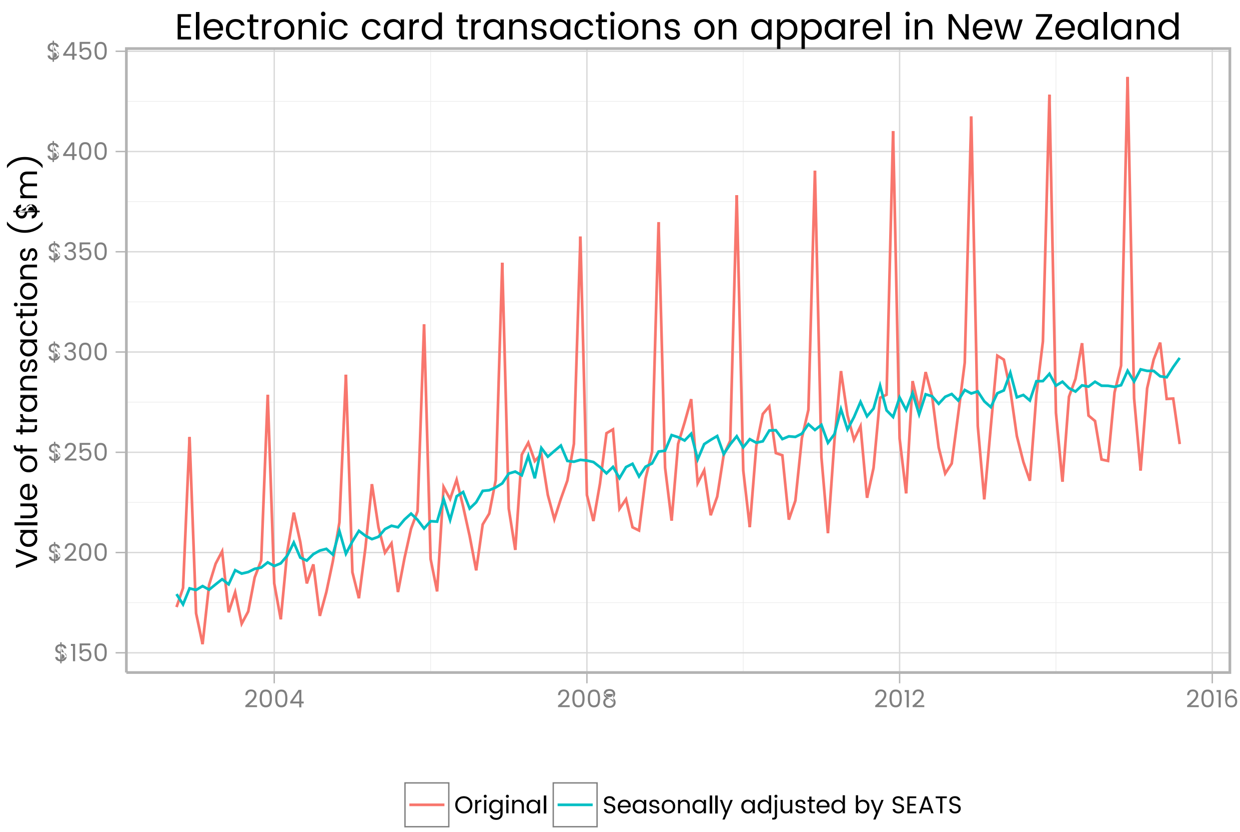

```
apparel_sa <- data_frame(
   Time = time(apparel_ts),
   Original = apparel_ts,
   SA = final(mod))

ggplot(apparel_sa, aes(x = Original, y = SA, colour = Time)) +
   geom_abline(intercept = 0, slope = 1, colour = "grey50") +
   geom_path() +
   geom_point() +
   coord_equal() +
   scale_x_continuous("Original ($m)", label = dollar) +
   scale_y_continuous("Seasonally adjusted ($m)", label = dollar) +
   ggtitle("Comparison of original and seasonally adjusted\n electronic card transactions on
```

In fact, Sax provides a super-useful `inspect()` function that spins up a Shiny app that lets you interact with all the key settings and diagnostics. I can't show that here though as I don't have access to a server with X-13ARIMA-SEATS and Shiny Server on it (couldn't run it on shinyapps.io at this point for example, because of the requirement for the X-13ARIMA-SEATS software).

For those interested in the actual results, the standard summary is shown below. Months with more Fridays and Saturdays in them get increased spend on apparel (something I hadn't expected, but I guess I hadn't thought about it very much); months with Easter in them get less spending; and the randomness can be modelled adequately with integration of order 1 and a moving average for both the month to month change, and the year-on-year month comparison:

```
> summary(mod)

Call:
seas(x = apparel_ts)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
Mon              -0.0042437  0.0042013  -1.010 0.312446
Tue              -0.0005243  0.0041054  -0.128 0.898375
Wed              -0.0049353  0.0040414  -1.221 0.222013
Thu              -0.0037071  0.0041312  -0.897 0.369540
Fri               0.0150563  0.0040701   3.699 0.000216 ***
Sat               0.0153981  0.0041104   3.746 0.000180 ***
Easter[1]        -0.0407969  0.0085273  -4.784 1.72e-06 ***
MA-Nonseasonal-01  0.6739188  0.0611324  11.024  < 2e-16 ***
MA-Seasonal-12     0.6326280  0.0671190   9.425  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

SEATS adj.  ARIMA: (0 1 1)(0 1 1)  Obs.: 155  Transform: log
AICc: 949.3, BIC: 977.1  QS (no seasonality in final):    0
Box-Ljung (no autocorr.):  23.7   Shapiro (normality): 0.9963
```

Here's how I turn the output into a `ggplot2` graphic:




```
apparel_sa %>%
   gather("variable", "value", -Time) %>%
   mutate(variable = gsub("SA", "Seasonally adjusted by SEATS", variable)) %>%
   ggplot(aes(x = Time, y = value, colour = variable)) +
   geom_line() +
   labs(colour = "", x = "") +
   scale_y_continuous("Value of transactions ($m)", label = dollar) +
   ggtitle("Electronic card transactions on apparel in New Zealand") +
   theme(legend.position = "bottom")
```

Create a new ggplot2 stat

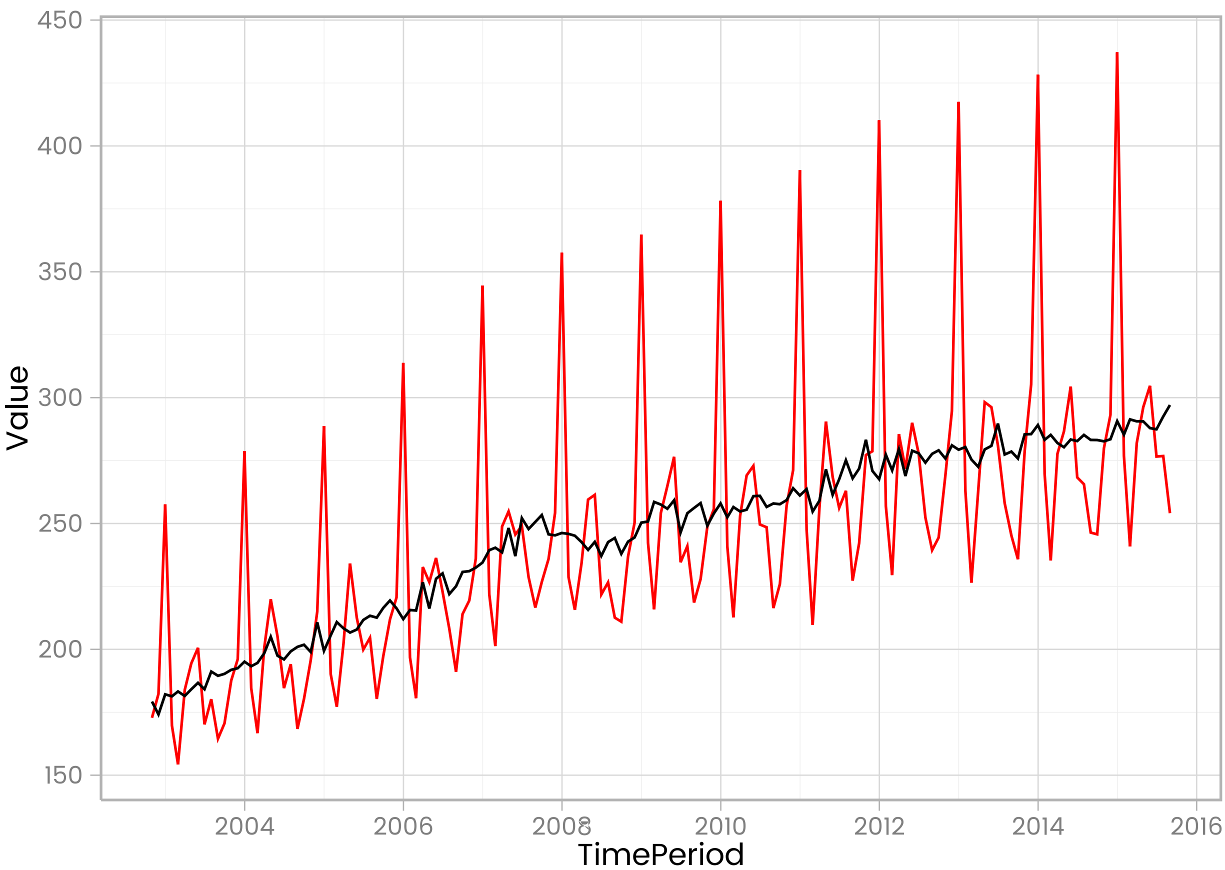
So that's nice, but to make this feel super useful and integrate into my graphics-based data exploration workflow, I'm going to want to seasonally adjust on the fly a dataset that is sliced and diced by different dimensions. I want something that works like `geom_smooth()`. Luckily, Hadley Wickham's amazing `ggplot2` package is designed to allow exactly this sort of extension. We just need to create a new statistical transformation, with the help of the `proto` package which lets us briefly treat R as though it were an object-oriented programming language.

In the code below, `StatSeas` is an object that creates functions, and `stat_seas(...)` is a function that works just like `stat_smooth()`, except instead of fitting a scatter plot smoother it performs a call to X-13ARIMA-SEATS with whatever arguments the user has passed through with the `...`. The user has to specify the frequency and the starting point of the time series, and the approach isn't robust to missing values (ie all the time series after slicing into groups need to begin at the same date).

```
library(proto)

StatSeas <- proto(ggplot2:::Stat, {
   required_aes <- c("x", "y")
   default_geom <- function(.) GeomLine
   objname <- "seasadj"
   calculate_groups <- function(., data, scales, ...){
      .super$calculate_groups(., data, scales, ...)
   }
   calculate <- function(., data, scales, frequency, start, ...) {
      y_ts <- ts(data$y, frequency = frequency, start = start)
      y_sa <- seasonal::final(seasonal::seas(y_ts, ...))
      result <- data.frame(x = data$x, y = as.numeric(y_sa))
      return(result)
   }
})
stat_seas <- StatSeas$new
```

Here's the new function in action:

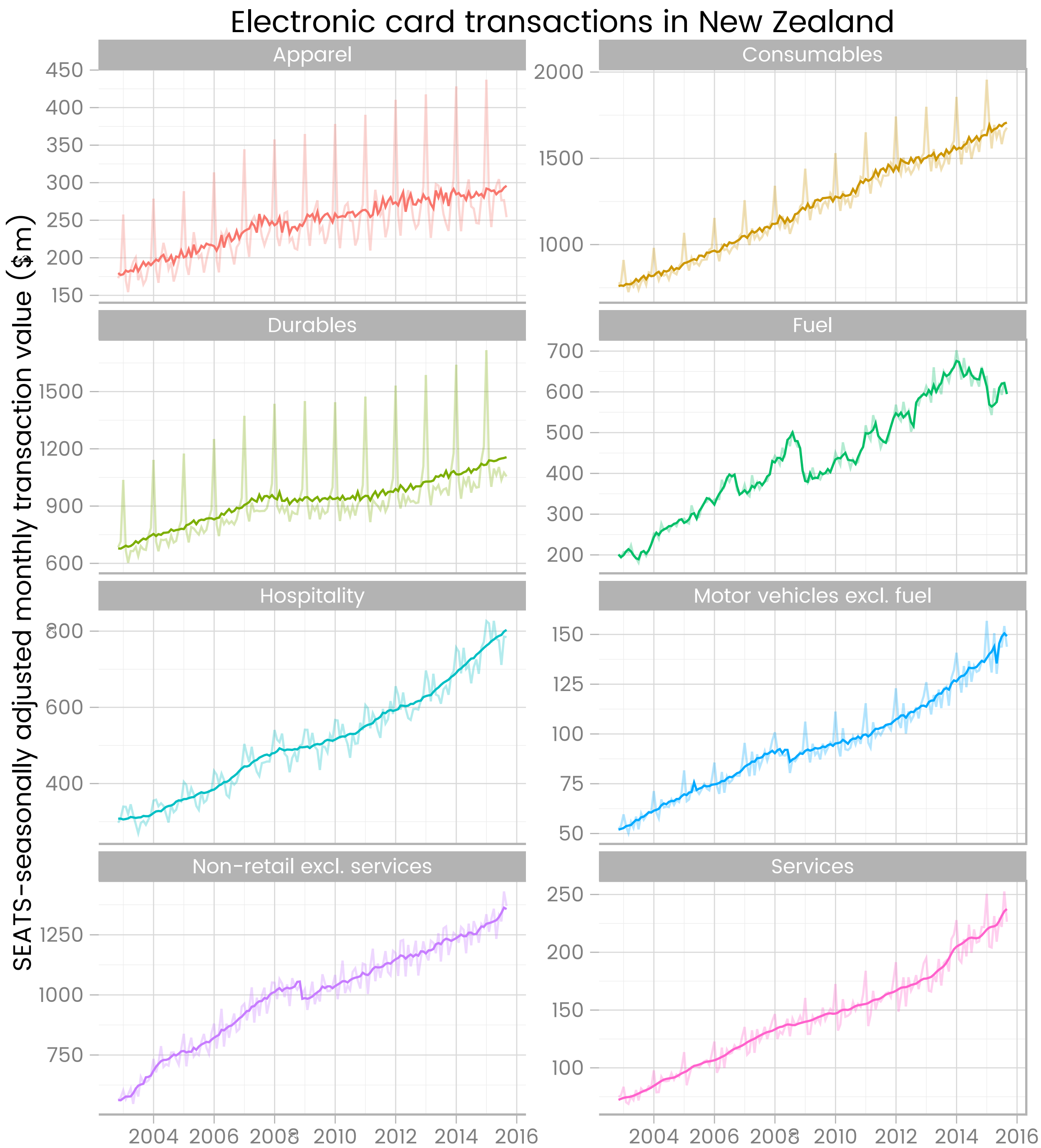
```
ggplot(apparel, aes(x = TimePeriod, y = Value)) +
   # original:
   geom_line(colour = "red") +
   # seasonally adjusted:
   stat_seas(frequency = 12, start = c(2002, 10))
```



Nice and easy!

But the real beauty is that we can use this on data that is grouped by aesthetics like colour or linetype, or facets. I demo this by going back to the full set of data of which spend on apparel was just one element. The eight lines of code below take the original data, fit SEATS models and seasonally adjust for every spend category, and produce a nicely polished plot. Not bad!

Thanks Hadley Wickham for `ggplot2`, Christoph Sax for `seasonal`, the US Census Bureau for X-13ARIMA-SEATS, and the Bank of Spain for the SEATS method.



```
ect %>%
   ggplot(aes(x = TimePeriod, y = Value, colour = group)) +
   geom_line(alpha = 0.3) +
   stat_seas(frequency = 12, start = c(1978, 4)) +
   facet_wrap( ~ group, scales = "free_y", ncol = 2) +
   labs(x = "", y = "SEATS-seasonally adjusted monthly transaction value ($m)",
        title = "Electronic card transactions in New Zealand") +
   theme(legend.position = "none")
```

Note - Statistics New Zealand publish an official seasonally adjusted series from this data, making the exercise above redundant other than as a demonstration. Where their results differ from mine (which is only in tiny details - I checked but decided too boring to include in here), use their’s not mine.

← Previous post

Recruiting Analysts for dynamic cutting edge public sector team (/blog/2015/10/04/recruiting)

Next post →

Silver flows in the seventeenth and eighteenth centuries (/blog/2015/10/25/silver)