So You Just Got 300 New Series You Need to Seasonally Adjust...

Brian C. Monsell and Osbert Pang

Brian.C.Monsell@census.gov

SAPW 2016

November 4, 2016

Disclaimer

- Any views expressed are those of the author(s) and not necessarily those of the U.S. Census Bureau.

Outline

- The problem
  - 364 Business Formation Series
  - Our planned procedure
- Creating a diagnostic summary in R
  - Seasonal package
  - The `udg( )` function

The problem

- The Center for Economic Studies is planning on publishing seasonally adjusted estimates for quarterly Business Formation series
  - 7 types of series
  - In each type, there is an estimate for each state, the District of Columbia, and the total for the US
- Plan a quick turnaround time

What was needed?

- A way to do a quick triage of the series
  - Examine plots of the series
  - Run all the series with default options
  - Flag those that seemed problematic
    - Do more extensive modeling and option checking for the problematic series
  - Do a final check with the final models and plots of components

Plots of the time series

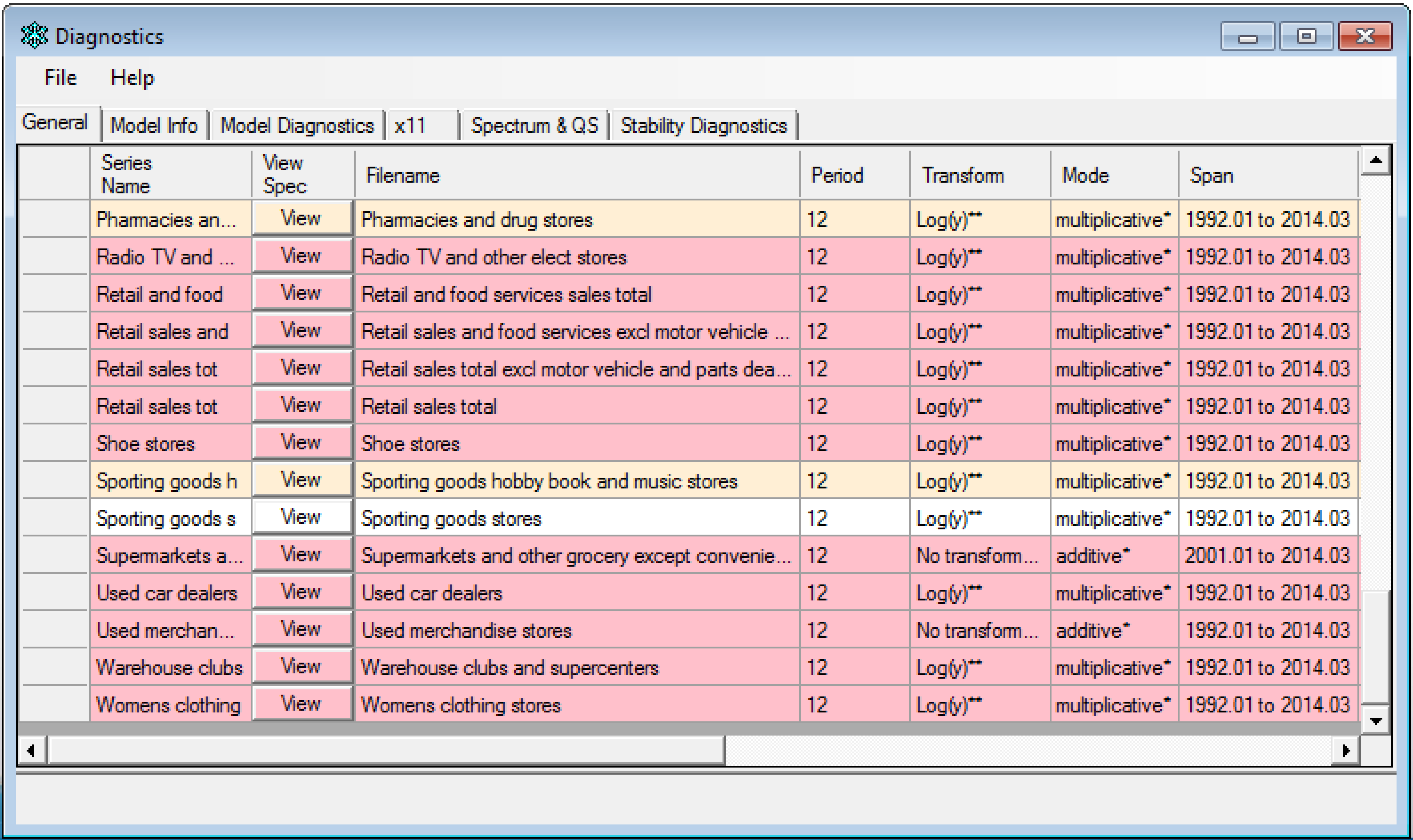
- Helpful in determining
  - Span of modeling
  - Transformation of the series
  - Possible outliers

- Since data not yet released, no plots in this presentation

Diagnostic summary

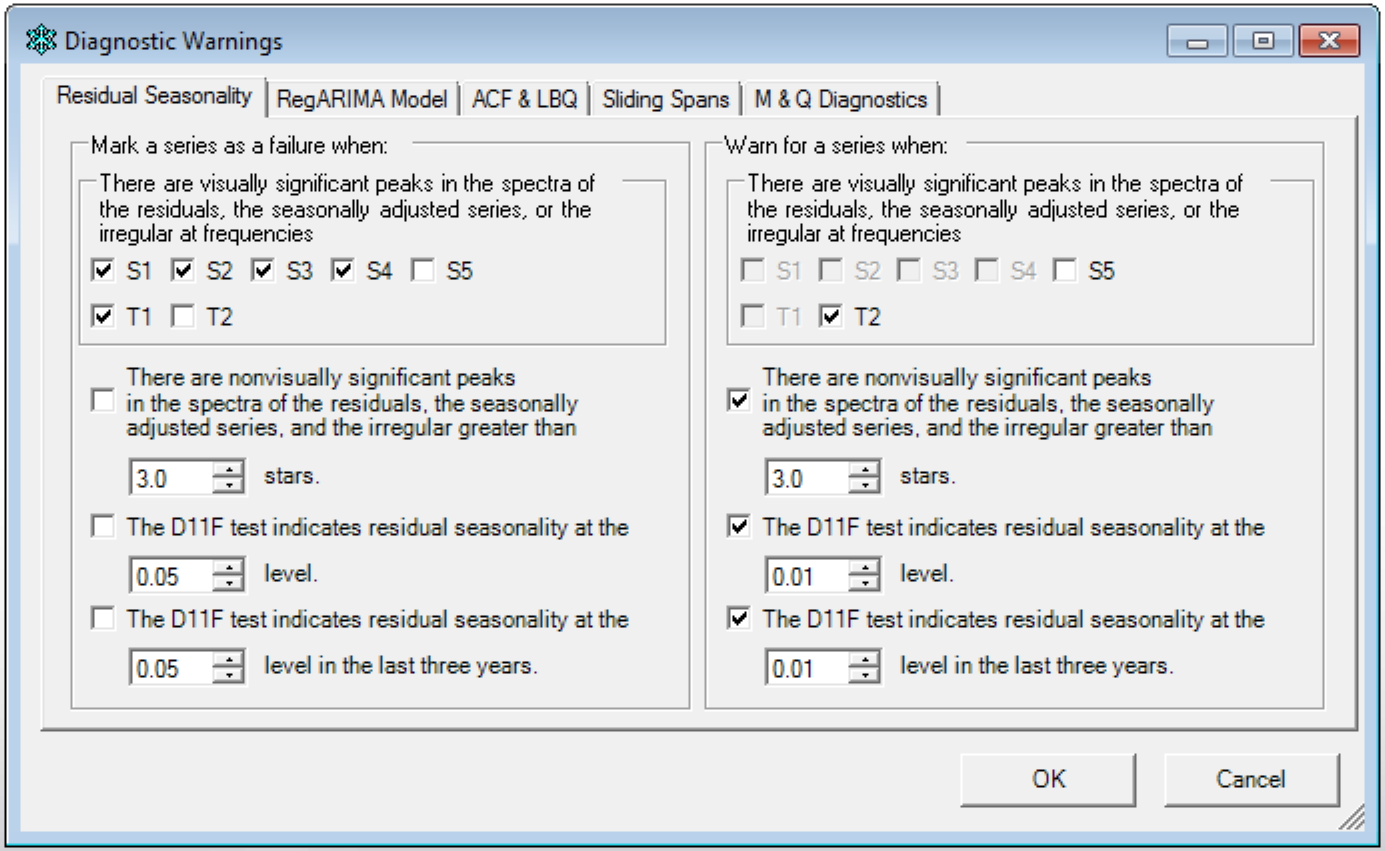
- Win X-13 produces an excellent summary of available diagnostics and other model information
- Allows the user to set limits for which diagnostics to flag and at what level
- Win X-13 could also be used to generate the spec files

Win X-13 diagnostic summary


Diagnostics
File Help
General | Model Info | Model Diagnostics | x11 | Spectrum & QS | Stability Diagnostics

Series Name	View Spec	Filename	Period	Transform	Mode	Span
Pharmacies an...	View	Pharmacies and drug stores	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Radio TV and ...	View	Radio TV and other elect stores	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Retail and food	View	Retail and food services sales total	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Retail sales and	View	Retail sales and food services excl motor vehicle ...	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Retail sales tot	View	Retail sales total excl motor vehicle and parts dea...	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Retail sales tot	View	Retail sales total	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Shoe stores	View	Shoe stores	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Sporting goods h	View	Sporting goods hobby book and music stores	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Sporting goods s	View	Sporting goods stores	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Supermarkets a...	View	Supermarkets and other grocery except convenie...	12	No transform...	additive*	2001.01 to 2014.03
Used car dealers	View	Used car dealers	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Used merchan...	View	Used merchandise stores	12	No transform...	additive*	1992.01 to 2014.03
Warehouse clubs	View	Warehouse clubs and supercenters	12	Log(y)**	multiplicative*	1992.01 to 2014.03
Womens clothing	View	Womens clothing stores	12	Log(y)**	multiplicative*	1992.01 to 2014.03

Diagnostic threshold



Diagnostic Warnings

Residual Seasonality | RegARIMA Model | ACF & LBQ | Sliding Spans | M & Q Diagnostics

Mark a series as a failure when:

There are visually significant peaks in the spectra of the residuals, the seasonally adjusted series, or the irregular at frequencies

[✓] S1 [✓] S2 [✓] S3 [✓] S4 [ ] S5

[✓] T1 [ ] T2

[ ] There are nonvisually significant peaks in the spectra of the residuals, the seasonally adjusted series, and the irregular greater than 3.0 stars.

[ ] The D11F test indicates residual seasonality at the 0.05 level.

[ ] The D11F test indicates residual seasonality at the 0.05 level in the last three years.

Warn for a series when:

There are visually significant peaks in the spectra of the residuals, the seasonally adjusted series, or the irregular at frequencies

[ ] S1 [ ] S2 [ ] S3 [ ] S4 [ ] S5

[ ] T1 [✓] T2

[✓] There are nonvisually significant peaks in the spectra of the residuals, the seasonally adjusted series, and the irregular greater than 3.0 stars.

[✓] The D11F test indicates residual seasonality at the 0.01 level.

[✓] The D11F test indicates residual seasonality at the 0.01 level in the last three years.

OK Cancel

However

- For security reasons we needed to run the series from a specific drive
- Generating output files from X-13ARIMA-SEATS (particularly HTML output files) caused storage problems

R seasonal package

- Allows users to run X-13ARIMA-SEATS with R
- Eliminates many of the external files generated by X-13ARIMA-SEATS
- Data and diagnostic information can be stored in efficient data structures within R

Example

```
# load seasonal package
library("seasonal")
Sys.setenv(X13_PATH = "h:/x13ashtml")
checkX13()

# run Airline Series,
# do X-11 seasonal adjustment
m <- seas(AirPassengers, x11="")
# examine output file for run
out(m)
```

R seasonal package

- Have access to series and diagnostics information for X-13 runs within R
- A new function for accessing the diagnostic information is the `udg( )` function
  - Allows access to information from the .udg file generated from X-13ARIMA-SEATS
  - Can pull out all the output, or output for individual keywords

```
date: Sep 22, 2016
time: 09.07.36
version: 1.1
build: 34
output: html
srstit: X-13ARIMA-SEATS run of airline
srsnam: airline
freq:    12
span:  1st month,1949 to 12th month,1960
constant:     0.0000000000E+00
transform: Log(y)
nfcst:    60
ciprob:     0.950000
lognormal: no
mvval:     0.1000000000E+10
iqtype: ljungbox
samode: multiplicative seasonal adjustment
```

```
airNfcst <- udg(m,"nfcst")
# nfcst = 60, a number

airOutput <- udg(m,"output")
# airOutput = "html", a string

airQSori <- udg(m,"qsori")
# airQSori = 167.64858    0.00000,
#             a numeric vector
```

Running multiple series

- First, we'll store the data in a list object
  - An object with named sets of other objects
  - thisData$series01
- Use `lapply()` to apply the `seas()` function to each element of the data list
  - Similar to running X-13ARIMA-SEATS in data metafile mode

```
setwd("N:/timeSeriesCSRM")
ahq.data.list <- list(
  state01 = import.ts("ahq_state01.dat"),
  state02 = import.ts("ahq_state02.dat"),
  state04 = import.ts("ahq_state04.dat"),
  state05 = import.ts("ahq_state05.dat"),
  us = import.ts("ahq_us.dat"))
#
#  ahq.data.list$state01 and
#  ahq.data.list[[1]] are equivalent
#
```

```
ahq.lauto <- lapply(ahq.data.list,
   function(x) try(seas(x, x11 = "")))

# Result is a list of seas objects that
# can be used with udg() and other
# functions to get diagnostic information
#
# Example: to view output for state1 –

out(ahq.lauto$state1)
```

Construct diagnostic summary

- Use similar criteria as Win X-13
  - Slightly modified for quarterly series
  - Create a number of R functions that use the `udg( )` function

Series of diagnostic tests

- Significant Seasonality using QS diagnostic
- Basic regARIMA Model diagnostics
- ACF and PACF diagnostics
- Residual Seasonality
  - regARIMA residuals (QS)
  - Seasonally adjusted series and irregular series (QS)
  - D11 F-test

Series of diagnostic tests

- Seasonal Adjustment Diagnostics
  - Sliding Spans Diagnostic
  - Q2, M7 Diagnostic

- Note – if these were monthly series, we would also want to check
  - Spectral peak results
  - Presences of calendar effects

R functions for diagnostics

- For each set of diagnostics we want to test, we have two types of functions
  - A function that returns a value of “pass”, “fail” or “warn” for each series in the list, depending on the criteria (Example: `QS.test()`)
  - A function that returns a text string that gives the reason why a series failed or got a warning (Example: `QS.test.why()`)

R functions for diagnostics

- Again, we'll use the `lapply()` function to apply these functions to each series

```
ahq.qs.test <- lapply(ahq.lauto,
   function(x) try(QS.test(x, testspan=FALSE)))

ahq.qs.fail <- UDGmatch(ahq.qs.test,"fail")
if (ahq.qs.fail[[1]] != "none") {
  ahq.qs.fail.why <- lapply(ahq.lauto[ahq.qs.fail],
     function(x) try(QS.test.why(x)))
} else { ahq.qs.fail.why <- "none" }

ahq.qs.warn <- UDGmatch(ahq.qs.test,"warn")
if (ahq.qs.warn[[1]] != "none") {
  ahq.qs.warn.why <- lapply(ahq.lauto[ahq.qs.warn],
     function(x) try(QS.test.why(x)))
} else { ahq.qs.warn.why <- "none" }
```

Final diagnostic summary

- The `diagDF()` function (Osbert Pang)
- Takes the output from all the tests and put them into one data table
  - First column gives the series name
  - Each column is a different test
  - If a test doesn't pass for all series, there is a column after that column showing the reason for the fail or warn state

Example diagnostic file

- Used the `write.csv()` function to store the diagnostic summary into a separate file

What next?

- Identify series that need extra attention
- View the X-13ARIMA-SEATS output using the `out()` function
- Rerun seas with updated options
  - Function called `saveSpecFile()` will save the seas function call used to generate a given m object into a separate file

Example

```
saveSpecFile("ahq","us")
#  contents of ahq.us.r is below:
```

```
x <- ahq.data.list$us
m.us <-
seas(x = x, transform.function = "log", x11 = "",
    slidingspans = "", forecast.maxlead = 8,
    check.print = "pacf",
    regression.variables = "ls2007.4",
    arima.model = "(1 1 0)(0 1 1)",
    regression.aictest = NULL,
    outlier = NULL)
```

Save final options

- Once we have a final set of options:
  - Store the final seas object into the list of seas objects
  - Use the `static()` function to create a set of final `seas` objects that can be used when new data are added to the data list

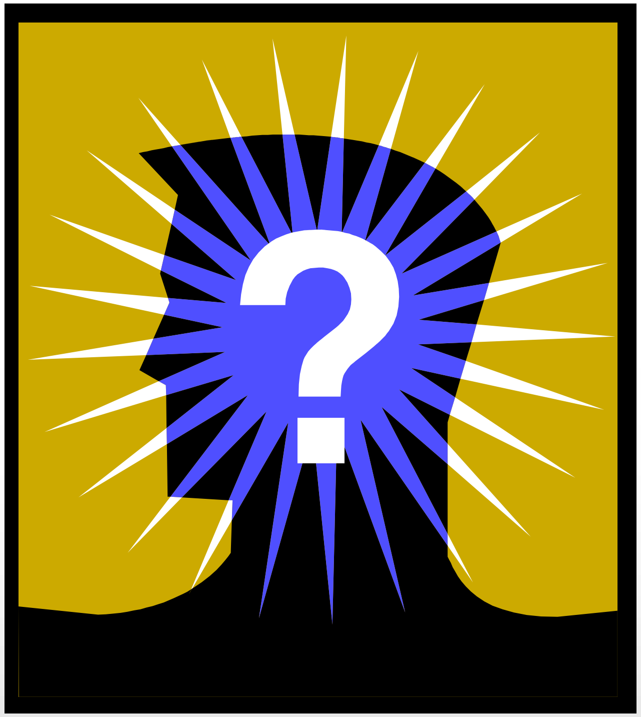
Save final options

```
ahq.lauto$us <- m.ahq.us

ahq.lcall <- lapply(ahq.lauto, static,
   x11.filter = TRUE, test = FALSE)

# re-evaluate static calls with new data
Map(function(x, call) eval(call),
   x = ahq.lnewdta,
   call = ahq.lcall lcall)
```

Future work

- Go back over the functions
  - Simpler
  - More modular
  - Not dependent on naming conventions
- Integrate this with what James is doing

Questions?



Brian C. Monsell
Email : brian.c.monsell@census.gov