Last week we  [wrote about time-series cross-validation](http://freerangestats.info/blog/2019/07/20/time-series-cv), and mentioned that my original motivation was forecasting unemployment. Actually, I’m interested in *nowcasting* unemployment rates – that is, estimating a value for a current or recently past period, before the official statistic becomes available. In particular, I’ve been wondering about what leading indicators might be available to help with this.

“Leading” in this case will have to mean pretty fast, because the official unemployment stats in Australia come out from the Australian Bureau of Statistics (ABS) with admirable promptitude given the complexity of managing the Labour Force Survey. ABS Series 6202.0 – the monthly summary from the Labour Force Survey – comes out around two weeks after the reference month. Only a few economic variables of interest are available faster than that. In this blog post I look at two candidates for leading information that are readily available in more or less real time – interest rates and stock exchange prices.

One big change in the past decade in this sort of short-term forecasting of unemployment has been to model the transitions between participation, employed and unemployed people, rather than direct modelling of the resulting proportions. This innovation comes from [an interesting 2012 paper by Barnichon and Nekarda](https://www.brookings.edu/wp-content/uploads/2012/09/2012b_Barnichon.pdf). I’ve only skimmed this paper, but I’d like to look into how much of the gains they report comes from the focus on workforce transitions, and how much from their inclusion of new information in the form of vacancy postings and claims for unemployment insurance. My suspicion is that these latter two series have powerful new information. I will certainly be returning to vacancy information and job adverts at a later time – these are items which feature prominently for me in my day job at Nous Group in analysing the labour market.

**Unemployment stats**

Here’s the Australian unemployment rate that we’re trying to forecast. I’ve added the party of federal government as context:

The data go back to 1978 and are published as their “original” survey estimates, seasonally adjusted and trend. “Trend” is nearly visually identical to “seasonally adjusted” so I’ve not included it in the chart above.

If I’m to be forecasting this I’ll want to forecast both the original and the seasonally adjusted values, so it’s worth checking out if I can replicate (or get close to) the ABS’ seasonal adjustment. Here’s a comparison of a seasonally adjusted series out of the box using X13-SEATS-ARIMA and the default settings provided by Christoph Sax’s excellent seasonal package (which includes testing for and including-if-helpful trading days and Easter seasonal adjustments).

There are 13 observations where we are 0.1 of a percentage point or more out, mostly featuring a difference of opinion between my model and the ABS’ on the seasonality of the winter months. This would be worth further exploring in a future post; perhaps the ABS’ parameters are published somewhere, and/or it would be interesting to try to reverse-engineer them. But it’s close enough for today’s purposes.

Here’s the R code so far. Getting hold of and tidying ABS time series data is a cinch now thanks to Matt Cowgill et al’s readabs package. BTW, I’m behind the times on how I manage time series data – I know that some of what I’m doing below can be done a bit easier with the help of [tsibble](https://cran.r-project.org/package=tsibble) and related R packages. But I haven’t got around to learning how yet. Which is a great example of why continuous learning is an essential part of modern statistics and data science.

*Post continues after code excerpt*

library(quantmod)

library(tidyverse)

library(scales)

library(readabs)

library(lubridate)

library(forecast)

library(vars)

library(rvest)

library(seasonal)

library(ggdag)

the\_caption <- "Analysis by http:://freerangestats.info"

#------------unemployment------------------

lfs1 <- read\_abs("6202.0", tables = 1)

govt <- tibble(end = as.Date(c("1975-12-13", "1983-03-05", "1996-03-02", "2007-11-24", "2013-09-07", "2022-06-30")),

party = c("ALP", "Lib/Nat", "ALP", "Lib/Nat", "ALP", "Lib/Nat")) %>%

mutate(start = c(as.Date("1972-12-02"), end[-n()]))

lfs1 %>%

filter(series == "Unemployment rate ; Persons ;" & series\_type != "Trend") %>%

ggplot() +

geom\_rect(data = govt, alpha = 0.8,

aes(xmin =start, xmax = end, ymin = -Inf, ymax = Inf, fill = party)) +

geom\_line(aes(x = date, y = value / 100, colour = series\_type)) +

scale\_y\_continuous(label = percent\_format(accuracy = 1)) +

labs(x = "",

y = "Unemployment rate",

colour = "Unemployment series:",

title = "Unemployment in Australia, 1978 to present",

subtitle = "Monthly series 6202.0 from the Australian Bureau of Statistics",

caption = the\_caption) +

scale\_fill\_manual("Federal government:", values = c("ALP" = "#E53440", "Lib/Nat" = "#1C4f9C")) +

scale\_colour\_manual(values = c("Original" = "black", "Seasonally Adjusted" = "white")) +

theme(legend.key = element\_rect(fill = "grey"))

# prepare data frame for later with columns for each series

unemp <- lfs1 %>%

filter(series == "Unemployment rate ; Persons ;" &

series\_type %in% c("Seasonally Adjusted", "Original")) %>%

dplyr::select(date, value, series\_type) %>%

spread(series\_type, value) %>%

rename(unemployment = Original,

unemployment\_sa = `Seasonally Adjusted`) %>%

mutate(unemployment\_my\_sa = ts(unemployment, start = c(1978, 2), frequency = 12) %>%

seas() %>%

final())

#--------------Comparison of seasonal adjustments------------------

col\_labs <- seq(from = 1979, to = 2019, by = 10) %>%

paste0("-01-01") %>%

as.Date(format = "%Y-%m-%d")

my\_col\_scale <- scale\_colour\_viridis\_c(breaks = as.numeric(col\_labs),

labels = year(col\_labs),

option = "D",

direction = -1)

ggplot(unemp, aes(x = unemployment\_sa, y = unemployment\_my\_sa, colour = as.numeric(date))) +

geom\_point() +

coord\_equal() +

labs(x = "Seasonally adjusted series published by the ABS",

y = "Seasonally adjusted from seasonal",

title = "Comparison of seasonally adjusted unemployment series",

subtitle = "Vertical axis uses the defaults from the {seasonal} R package front-end to X13-SEATS-ARIMA",

caption = the\_caption,

colour = "") +

my\_col\_scale +

theme(legend.position = "right")

# check for the individual highest points

unemp %>%

mutate(d = abs(unemployment\_my\_sa - unemployment\_sa)) %>%

arrange(desc(d)) %>%

slice(1:20)

**Stock prices and interest rates**

The Australian Securities Exchange publishes an index based on the price of the “ASX 200” list of companies. Daily data are available from the likes of Yahoo Finance (easily accessed from the R quantmod package), but the longest series I could find was from the ASX themselves. Unfortunately, even this series only goes back to the early 2000s. There must be more historical data out there somewhere, so please feel free to make a comment on this post about where to find it.

I grabbed the historical interest rates series form the Reserve Bank of Australia (RBA). This series goes all the way back to the 1970s.

Here’s what all the time series I’ve got so far look like – unemployment (both original and seasonally adjusted), interest rates and stock prices:

**Relationships**

So what’s the relationship between these different series when we put them into scatter plots, with one axis for each series? First, let’s look at them on their original scales. We see there’s strongly negative relationships between both interest rates and stock prices on unemployment:

Of course, this doesn’t mean that interest rates and stock prices going up leads to a fall in unemployment. Rather, there are some complicated things going on here:

* When unemployment is low or expected to be low, the authorities become more worried about inflation and tend to increase interest rates to control that.
* When the economy is doing well, unemployment goes low and stock prices tend to grow.
* When investors think that interest rates are likely to go up, they pay less for stocks and stock prices go down (and vice versa). So a decrease in unemployment can lead to “pessimism” on the part of stock market players, fearing a rise in interest rates despite (or because of) the good news for the economy that the decrease in unemployment is. And vice versa – an anticipated increase in unemployment leads to expectations of interest rate cuts, which can lead to a surge in stock prices.
* Complicating these relationships, stock prices are definitely a non-stationary time series (ie despite their ups and downs, they have definitely grown on average since the 2000s) – and interest rates and unemployment rates are probably non-stationary (that is, their average values change over time in historic level shifts, not just oscillating around a constant mean). Comparing non-stationary time-series leads to spurious correlations.

We can control to a degree for the non-stationarity problem by looking at growth rates or first differences rather than the original series, and the following set of scatter plots does just that:

Now we’re getting somewhere. The apparent relationship between the stock prices and unemployment was probably a spurious artefact of the secular change they were both undergoing. On the other hand, there is definitely a negative relationship between change in interest rates and change in unemployment rates. What we’re seeing here is the dominance of one way of the causality direction – as unemployment goes up, interest rates go down in response. This trumps the presumed causal effect in the opposite direction (as interest rates go down, it boosts the economy, eventually leading to a decrease in unemployment), which is probably only observeable, if at all, at quite some lag.

In fact, the situation was sufficiently complex that I had a go at drawing a direct acyclic graph (DAG), as is now all the rage in thinking about causality. Here’s my greatly simplified description of what is happening here:

This provides a framework for thinking about what we’re observing. Basically, the connection from unemployment to interest rates via the RBA is realtively strong and quick, whereas the causality in the other direction is complex, delayed and difficult to observe. And the price of stocks is impacted on by so many things in such weird ways (I haven’t tried to show all the “animal spirits” and other influences, rational and otherwise) that it isn’t that surprising that it’s little or no use as a predictor of unemployment.

Here’s R code to date, probably of particular interest for people either in interested in grabbing this historical data from the ASX and the RBA, or in drawing DAGs (for which I’m using the R package “ggdag: Analyze and Create Elegant Directed Acyclic Graphs” by Malcolm Barrett).

*Post continues after code excerpt*

#-----------------------ASX 200 from Yahoo--------------------

# First attempt at stock price data, not enough historical to be useful

axjo <- getSymbols("^AXJO", auto.assign = FALSE)

# exploratory plot (not shown in blog)

autoplot(axjo$AXJO.Close) +

labs(x = "",

title= "ASX 200 stock price index",

subtitle = "Sourced from Yahoo Finance via quantmod; data only available back to 2007",

caption = the\_caption)

#---------------------ASX 200 from ASX------------------

url <- "https://www.asx.com.au/about/historical-market-statistics.htm"

asx <- read\_html(url)

# Read the 18 separate HTML tables in as elements of a list

asx\_l <- lapply(1:18, function(i){ html\_table(asx)[[i]]})

# convert to a data frame. Note that `date` is going to use the first day of the

# month even though the values are actually month-end values. This is for joining

# to other data that use a different convention later on:

asx\_df <- do.call("rbind", asx\_l) %>%

as\_tibble() %>%

# convert the messy and inconsistent Month column into an actual date:

mutate(mon = str\_sub(Month, 1, 3),

yr = str\_extract(Month, "[0-9]\*$"),

date = as.Date(paste("01", mon, yr, sep = "-"), format = "%d-%b-%y")) %>%

dplyr::select(-mon, -yr, -Month) %>%

# convert numbers with comma big marks into numeric values:

mutate(`Dom. Equity Mkt cap $m` = as.numeric(gsub(",", "", `Dom. Equity Mkt cap $m`)))

# exploratory plot (not shown in blog)

asx\_df %>%

gather(variable, value, -date) %>%

ggplot(aes(x = date, y = value)) +

facet\_wrap(~variable, scales = "free\_y", ncol = 1) +

geom\_line() +

scale\_y\_continuous(label = comma) +

labs(title = "Various stock market indicators direct from the ASX",

subtitle = "Month-end data available back to 2002",

x = "",

y= "",

caption = paste("Source: https://www.asx.com.au/about/historical-market-statistics.htm\n",

the\_caption))

# make a data frame of just the series we're interested in:

asx\_end\_month <- asx\_df %>%

dplyr::select(date,

asx200 = `S&P/ASX 200 Price Index`)

#--------------------Interest rates---------------------

f11 <- read\_csv("https://www.rba.gov.au/statistics/tables/csv/f1.1-data.csv?v=2019-06-08-13-25-32",

skip = 10)

interest <- f11 %>%

mutate(date = as.Date(paste0("01-", `Series ID`), format = "%d-%b-%Y")) %>%

dplyr::select(date, interest = FIRMMCRI) %>%

filter(!is.na(interest))

#===============Analysis===============

# create a combined data set of all the series so far

combined <- asx\_end\_month %>%

full\_join(unemp, by = "date") %>%

full\_join(interest, by = "date") %>%

mutate(label = ifelse(month(date) == 1, year(date), "")) %>%

ungroup() %>%

arrange(date) %>%

mutate(date\_n = as.numeric(date)) %>%

dplyr::select(-unemployment\_my\_sa)

# combined time series plot

combined %>%

gather(variable, value, -date, -date\_n, -label) %>%

ggplot(aes(x = date, y = value)) +

facet\_wrap(~variable, scales = "free\_y", ncol = 1) +

geom\_line() +

labs(x = "", y = "",

title = "Stock prices, interest rates and unemployment considered together",

subtitle = "Sourced from ASX, RBA and ABS",

caption = the\_caption)

d1 <- combined %>%

dplyr::select(-unemployment) %>%

filter(!is.na(asx200) & !is.na(unemployment\_sa)) %>%

gather(variable, value, -date, -date\_n, -label, -unemployment\_sa)

col\_labs <- seq(from = 2002, to = 2019, by = 3) %>%

paste0("-01-01") %>%

as.Date(format = "%Y-%m-%d")

my\_col\_scale <- scale\_colour\_viridis\_c(breaks = as.numeric(col\_labs),

labels = year(col\_labs),

option = "D",

direction = -1)

# first connected scatter plot

d1 %>%

ggplot(aes(x = value, y = unemployment\_sa / 100, colour = date\_n)) +

facet\_wrap(~variable, scales = "free\_x") +

geom\_smooth(method = "gam", colour = "brown") +

geom\_point(data = filter(d1, label == "")) +

geom\_text(aes(label = label)) +

geom\_path() +

theme(legend.position = "right") +

labs(x = "Raw value",

y = "Seasonally adjusted unemployment rate",

colour = "",

caption = the\_caption,

title = "Inverse relationship between unemployment and both stock prices and interest rates",

subtitle = "Unemployment impacts on interest rates, rather than the other way around. And stock prices are undergoing long term structural change.") +

scale\_x\_continuous(label = comma) +

scale\_y\_continuous(label = percent) +

my\_col\_scale

d2 <- combined %>%

dplyr::select(-unemployment) %>%

mutate(asx200 = c(NA, diff(log(asx200))),

interest = c(NA, diff(interest)) / 100,

unemployment\_sa = c(NA, diff(log(unemployment\_sa)))) %>%

filter(!is.na(asx200) & !is.na(unemployment\_sa)) %>%

gather(variable, value, -date, -label, -unemployment\_sa, -date\_n)

# Second connected scatter plot

d2 %>%

ggplot(aes(x = value, y = unemployment\_sa / 100, colour = date\_n)) +

facet\_wrap(~variable, scales = "free\_x") +

geom\_smooth(method = "gam", colour = "brown") +

geom\_point(data = filter(d2, label == "")) +

geom\_text(aes(label = label)) +

geom\_path() +

theme(legend.position = "right") +

labs(x = "Growth in other variable",

y = "Growth in unemployment rate",

title = "Changes in interest rates and stock prices, compared to unemployment rate - monthly changes",

subtitle = "Unemployment and ASX200 shown as diff(log(x)); change in interest rate shown in percentage points.

Increase in unemployment comes with decrease in interest rate; no real relationship to stock prices.",

colour = "") +

scale\_x\_continuous(label = percent) +

scale\_y\_continuous(label = percent) +

my\_col\_scale

#--------------Digression to think about causality-----------------------

dagified <- dagify(unemployment ~ economy,

interest ~ RBA,

RBA ~ inflation + unemployment,

govt ~ voters,

voters ~ unemployment + stocks + inflation + interest,

economy ~ govt + inflation + interest,

inflation ~ economy + interest,

stocks ~ economy + interest,

outcome = "unemployment")

# Draw DAG:

set.seed(123)

tidy\_dagitty(dagified) %>%

ggplot(aes(x = x, y = y , xend = xend, yend = yend)) +

geom\_dag\_edges\_arc(edge\_colour = "grey50") +

geom\_dag\_node(colour ="grey80") +

geom\_dag\_text(colour = "steelblue", family = main\_font) +

theme\_void(base\_family = main\_font) +

labs(title = "Simplified causal diagram of key economic actors and variables in Australia",

caption = "Only primary impacts are shown, rather than impacts that depend on anticipating intermediaries.")

**Vector auto regression as an exploratory modelling stage**

The data here seem appropriate for exploring with a vector auto-regression model (VAR). A VAR models the changes in multiple time series together and is a way to understand the impacts of each (and their lagged versions) on themselves and eachother.

One of the most useful outputs from fitting a VAR is the plot of the Impulse Response Function, which lets you visualise the ongoing impact on all series of a shock to one series. In our case, here’s what happens if there is an upwards shock to interest rates:

The biggest impact is on interest rates, they go high, increase further for a bit, and basically stay high – the shock leads to a level shift upwards in the interest rates. But the model also implies that this level shift up in interest rates leads to a level shift downwards in unemployment. Clearly, interpreting this causally would be a mistake, for the reasons discussed earlier.

Also, it’s interesting to note that the interest rate shift upwards has little evidence of any impact on stock prices (the red lines are 95% confidence intervals, and basically include zero, suggesting no clear impact one way or another).

For completeness, here are the charts showing the impact of shocks to the other two series.

None of these are as polished as I usually like my graphics, but I’m treating these as outputs from exploratory modelling rather than core parts of my results dissemination. Actually, I have no intent of using a VAR for my actual nowcasting, as the whole point of using leading indicators is that I don’t need to forecast all three series together. But a VAR is a useful thing to fit to add a bit more discipline in thinking about multi-directional [Granger causality](https://en.wikipedia.org/wiki/Granger_causality) to what has so far been just a bunch of scatter plots.

Here’s the code that fits those VARs, and also which puts together the data in a form for what I’ll be actually using in a jiffy. I’m mostly using the first-differenced stock price series, and original unemployment and interest rates series. My rationale for doing this is the instinct that prices are fundamentally a different, more absolute measure than are either of the other two, which are expressed as rates or proportions. Definitely if I were doing this for real, this would be one of my more substantive decisions that would need some serious thinking, and it’s one we’ll come back to.

*Post continues after code excerpt*

#-----------------Subsets of data for modelling---------------

# a subset of the data that is complete for when we have the growth in ASX prices, and its lag, and unemployment

d4 <- combined %>%

mutate(asx200 = c(NA, diff(log(asx200))),

asx200\_l = lag(asx200)) %>%

filter(!is.na(unemployment) & !is.na(asx200\_l))

asx200\_ts <- ts(d4$asx200, start = c(2002, 6), frequency = 12)

asx200\_ts1 <- ts(d4$asx200\_l, start = c(2002, 6), frequency = 12)

unemp\_ts <- ts(d4$unemployment, start = c(2002, 6), frequency = 12)

unemp\_sa\_ts <- ts(d4$unemployment\_sa, start = c(2002, 6), frequency = 12)

interest\_ts <- ts(d4$interest, start = c(2002, 6), frequency = 12)

#-----------------Vector auto-regression----------

# For the VAR, we'll use seasonally adjusted unemployment

comb\_ts\_d <- scale(cbind(asx200\_ts, unemp\_sa\_ts, interest\_ts)[-1, ])

mod\_var <- VAR(comb\_ts\_d, p = 2)

plot(irf(mod\_var, impulse = "interest\_ts"), main = "Impact of a shock upwards to interest rates",

sub = "We can see that this model is based on a misinterpretation of causality")

plot(irf(mod\_var, impulse = "asx200\_ts"), main = "Impact of a shock upwards to stock price")

plot(irf(mod\_var, impulse = "unemp\_sa\_ts"), main = "Impact of a shock upwards to unemployment")

**Finally, creating some viable forecasting models**

I fit four potential ARIMA (auto-regressive integrated moving average) models, in each case specifying level one of integration ie the original time series is to be first-differenced. I used the original observations, not seasonally adjusted – my idea is that if I ever do this for real, I will nowcast the actual observed unemployment, and turn it into a seasonally adjusted value myself, so I have both numbers to present.

My four models differ in the explanatory “x” regressor variables they include:

* none
* just interest rates
* interest rates and stock prices
* interest rates, stock prices, and stock prices at lag 1

Here’s the performance of each of these models on the basis of Akaike’s Information Criterion:

|  | **df** | **AIC** |
| --- | --- | --- |
| mod\_aa0 | 6 | -109.8404 |
| mod\_aa1 | 8 | -128.5164 |
| mod\_aa2 | 9 | -126.6400 |
| mod\_aa3 | 10 | -124.8471 |

So historically, the best performing model for predicting unemployment (the lowest value of AIC) is the one with interest rates in it as an explanatory variable, but no stock prices.

Here’s rough confidence intervals for the values of all the parameters in that model. We’re most interest in the xreg estimate, which is the direction and strenght of the impact of interest rates in forecasting uinemployment. Remember, this doesn’t imply causality – just that it’s useful for forecasting purposes.

|  | **2.5 %** | **97.5 %** |
| --- | --- | --- |
| ar1 | -0.05 | 0.64 |
| ar2 | -0.26 | 0.14 |
| ma1 | -0.96 | -0.31 |
| sar1 | -2.65 | 1.70 |
| sma1 | -2.43 | 1.96 |
| sma2 | -1.84 | 1.26 |
| xreg | -0.38 | -0.21 |

The negative value means that when interest rates are high, it’s a sign that the change in unemployment rates is about to decrease. How are we to interpret this? Recall this simple chart:

We see that at times of substantial economic change, there is a very strong negative link between interest rates and unemployment. This is particularly apparent in the early 1990s and in the period around the 2008 global economic crises up to 2010. The 1990s are excluded from my modelling data in order to have four comparable models, because I don’t have stock prices to include; so I suspect it’s the 2007 to 2009 period that is being picked up in the model.

I am profoundly skeptical of whether this indicates that we really have found a useful forecasting explanatory regressor variable. Hence I wanted to check the AIC quality of fit figures with time series cross-validation of the one-period-out forecasts. AIC is actually asymptotically equivalent to one-period-out cross-validation, but how much data do we need for this property to kick in? Funny things are the rule rather than exception with time series, and spurious correlation lurks everywhere. I’d feel more comfortable if brute force showed me that I would have been better off to include interest rates than not, in a thought experiment where I was doing monthly nowcasts of unemployment rates since the 2000s. Hence my post last week on time-series cross-validation, to be sure with simulated data that I was doing it right.

The reason I wanted to be sure (and my digression last week into simulated data) was that in this case, the cross-validation comes up with different results from the AIC. Here’s the accuracy summary of the one-period-out cross-validation results for comparing models mod\_aa0 (first differenced unemployment rate, no explanatory variables) and mod\_aa1 (with interest rates as an explanatory variable). The no-explanatory-variables model wins hands-down:

| **model** | **ME** | **RMSE** | **MAE** | **MPE** | **MAPE** | **ACF1** | **Theil’s U** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| With interest rates | 0.0285642 | 0.4014554 | 0.1882691 | 0.6239319 | 3.491464 | 0.0617393 | 1.2358698 |
| Univariate | 0.0061069 | 0.2200771 | 0.1631800 | 0.2062656 | 3.032230 | -0.0147829 | 0.6987458 |

The univariate mode has lower bias (mean error) as well as lower root mean square error, mean absolute error, mean percentage error, mean absolute percentage error, autocorrelation of the residuals, and Theil’s U. The apparent superiority of the interest rates model from comparing AICs turns out to be a mistake.

So after all that, what have we found? If I want to nowcast unemployment rates, I am best off (on the potential leading indicators used so far), ignoring other variables and simply fitting a univariate time series model to the unemployment rate. This actually doesn’t surprise me, but it was worth looking around.

I still have hopes of vacancy rates or job advert numbers as a leading variable. Unlike interest rates, these are basically measures of a very similar, directly related concept to unemployment, without any significant causality complications. But this will need to wait for another post.

Here’s the R code for fitting the various ARIMA models and doing the time-series cross-validation. The [cross-validation code is explained in last week’s post](http://freerangestats.info/blog/2019/07/20/time-series-cv):

*Post continues after code excerpt*

#----------------ARIMA modelling---------------

mod\_aa0 <- auto.arima(unemp\_ts, d = 1)

mod\_aa1 <- auto.arima(unemp\_ts, d=1,

xreg = interest\_ts)

mod\_aa2 <- auto.arima(unemp\_ts, d=1,

xreg = cbind(asx200\_ts, interest\_ts))

mod\_aa3 <- auto.arima(unemp\_ts, d=1,

xreg = cbind(asx200\_ts, asx200\_ts1, interest\_ts))

knitr::kable(AIC(mod\_aa0, mod\_aa1, mod\_aa2, mod\_aa3))

knitr::kable(round(confint(mod\_aa1), 2))

# The fact that higher interest rates are a good forecast of lower unemployment suggests the RBA (or banks?)

# are taking extra information into account in setting their interest rates - they know something about

# the direction the economy is going that lets them set the interest rate level even before it has

# manifested itself in the unemployment

#-------------Time series cross-validation------------

# The model chosen by AIC (ie mod\_aa1, which includes interest rates and is better than the univariate model)

# should in theory be the same as what we'd get with leave-one-out cross validation. See:

# Stone M. (1977) An asymptotic equivalence of choice of model by cross-validation and Akaike’s criterion. Journal of the Royal Statistical Society Series B. 39, 44–7.

# But funny things can happen with time series, and actually our sample is not vast. So let's check that out:

#' auto.arima forecast function for time series cross validation

#'

#' adapted from https://gist.github.com/robjhyndman/d9eb5568a78dbc79f7acc49e22553e96

aafc <- function(y, h, xreg = NULL, ...){

if(!is.null(xreg)){

ncol <- NCOL(xreg)

X <- matrix(xreg[1:length(y), ], ncol = ncol)

if(NROW(xreg) < length(y) + h)

stop("Not enough xreg data for forecasting")

newX <- matrix(xreg[length(y) + (1:h), ], ncol = ncol)

fit <- auto.arima(y, xreg=X, ...)

return(forecast(fit, xreg = newX, h = h))

} else {

fit <- auto.arima(y, ...)

return(forecast(fit, h = h))

}

}

# this CV takes about 20 minutes

system.time({

aa1\_cv <- tsCV(unemp\_ts, aafc, xreg = interest\_ts, d = 1)

aa0\_cv <- tsCV(unemp\_ts, aafc, d=1)

})

rbind(accuracy(unemp\_ts - aa1\_cv, unemp\_ts),

accuracy(unemp\_ts - aa0\_cv, unemp\_ts)) %>%

as.data.frame() %>%

mutate(model = c("With interest rates", "Univariate")) %>%

dplyr::select(model, everything())

**Final clincher – testing on a longer time series**

So far in my modelling, I’ve been working with a cut-down dataset, only going back to 2002 so I can include stock prices as a candidate variable. If I’m satisfied that stock prices have nothing to offer in nowcasting unemployemnt (and I am now), then I might as well refit my two univariate and interest rates models with the full set of data available back to the late 1970s.

This time, the interest rate model is inferior even on the basis of the AIC (which of course has a longer series to allow its asymptotic properties to kick in).:

|  | **df** | **AIC** |
| --- | --- | --- |
| mod\_aa0\_long | 6 | -106.2901 |
| mod\_aa1\_long | 7 | -104.3239 |

This is confirmed by the time-series cross-validation:

| **model** | **ME** | **RMSE** | **MAE** | **MPE** | **MAPE** | **ACF1** | **Theil’s U** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| With interest rates | -0.0022767 | 0.3287481 | 0.1980351 | 0.0378736 | 2.959823 | -0.0400820 | 0.8890656 |
| Univariate | 0.0064120 | 0.2404348 | 0.1847038 | 0.1837424 | 2.755187 | 0.0177967 | 0.6408704 |

Other than mean percentage error (which I dislike as a metric of accuracy anyway), the univariabte model is the better of the two.

*Post concludes after code excerpt*

#---------longer series------

unemp\_long <- ts(unemp$unemployment, frequency = 12, start = c(1978, 2))

interest\_long <- ts(interest$interest, frequency = 12, start = c(1976, 5)) %>%

window(start = c(1978, 2))

mod\_aa0\_long <- auto.arima(unemp\_long, d = 1)

mod\_aa1\_long <- auto.arima(unemp\_long, d=1, xreg = interest\_long)

AIC(mod\_aa0\_long, mod\_aa1\_long)

system.time({

aa1\_cv\_long <- tsCV(unemp\_long, aafc, xreg = interest\_long, d = 1)

aa0\_cv\_long <- tsCV(unemp\_long, aafc, d=1)

})

rbind(accuracy(unemp\_long - aa1\_cv\_long, unemp\_long),

accuracy(unemp\_long - aa0\_cv\_long, unemp\_long)) %>%

as.data.frame() %>%

mutate(model = c("With interest rates", "Univariate")) %>%

dplyr::select(model, everything()) %>%

knitr::kable()

**Unfinished business**

So, a nice, clear conclusion today – don’t bother with interest rates or stock prices in trying to nowcast unemployment rates.

I have three things that need further investigation here, but as this post is now well over 500 lines long these are definitely for another time:

* more systematic thinking and exploration of whether the interest rate explanatory regressor should be first differenced, not in its absolute values. For the curious, I did fit my models with differenced interest rates, and reach the same (no effect) eventual conclusions as reported in the post. In fact, is possible that the non-stationary undifferenced interest rate series is what’s responsible for the suprious AIC finding that interest rates are worth including in the model in the first place.
* replicating the ABS’ seasonal adjustment.
* testing other possible explanatory variables, particularly vacancy rates or job advert numbers.

That’s all for now, folks.