Sorry ARIMA, but I'm Going Bayesian

When people think of "data science" they probably think of algorithms that scan large datasets to predict a customer's next move or interpret unstructured text. But what about models that utilize small, time-stamped datasets to forecast dry metrics such as demand and sales? Yes, I'm talking about good old time series analysis, an ancient discipline that hasn't received the cool "data science" re-branding enjoyed by many other areas of analytics.

Yet, analysis of time series data presents some of the most difficult analytical challenges: you typically have the least amount of data to work with, while needing to inform some of the most important decisions. For example, time series analysis is frequently used to do demand forecasting for corporate planning, which requires an understanding of seasonality and trend, as well as quantifying the impact of known business drivers. But herein lies the problem: you rarely have sufficient historical data to estimate these components with great precision. And, to make matters worse, validation is more difficult for time series models than it is for classifiers and your audience may not be comfortable with the imbedded uncertainty.

So, how does one navigate such treacherous waters? You need business acumen, luck, and *Bayesian structural time series models*. In my opinion, these models are more transparent than ARIMA – which still tends to be the go-to method. They also facilitate better handling of uncertainty, a key feature when planning for the future. In this post I will provide a gentle intro the `bsts` R package written by Steven L. Scott at Google. Note that the `bsts` package is also being used by the `CausalImpact` package written by Kay Brodersen, which I discussed in this post from January.

Airline Passenger Data

An ARIMA Model

First, let's start by fitting a classical ARIMA model to the famous airline passenger dataset. The ARIMA model has the following characteristics:

- First order differencing ($I = 1$) and a moving average term ($q = 1$)
- Seasonal differencing and a seasonal MA term.
- The year of 1960 was used as the holdout period for validation.
- Using a log transformation to model the growth rate.

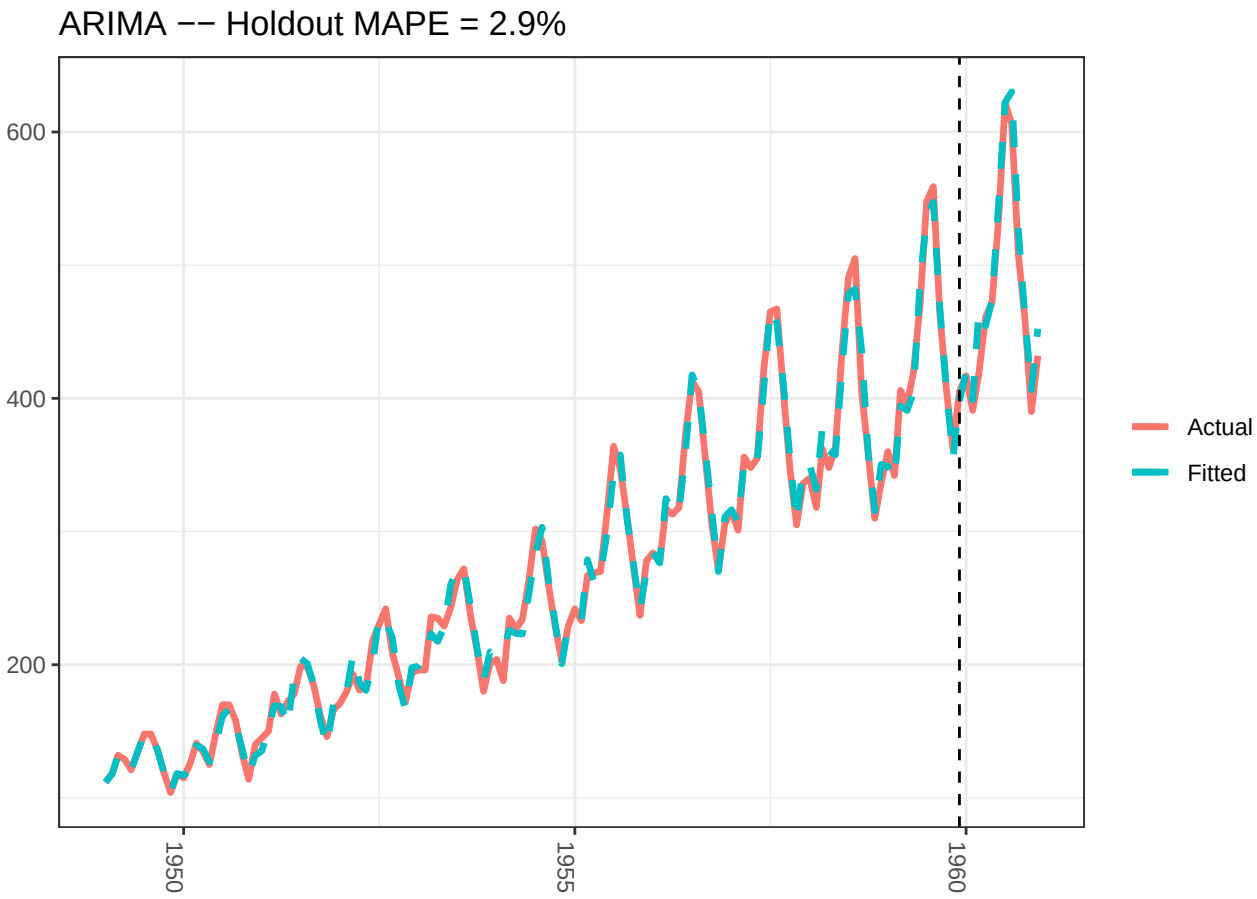
```
library(lubridate)
library(bsts)
library(dplyr)
library(ggplot2)
library(forecast)
library(Boom)

data("AirPassengers")
Y <- window(AirPassengers, start=c(1949, 1), end=c(1959,12))
arima <- arima(log10(Y),
               order=c(0, 1, 1),
               seasonal=list(order=c(0,1,1), period=12))

d1 <- data.frame(c(10^as.numeric(fitted(arima)), # fitted and predicted
                   10^as.numeric(predict(arima, n.ahead = 12)$pred)),
                   as.numeric(AirPassengers), #actual values
                   as.Date(time(AirPassengers)))
names(d1) <- c("Fitted", "Actual", "Date")

MAPE <- filter(d1, year(Date)>1959) %>% summarise(MAPE=mean(abs(Actual-Fitted)/Actual))
```

```
ggplot(data=d1, aes(x=Date)) +
  geom_line(aes(y=Actual, colour = "Actual"), size=1.2) +
  geom_line(aes(y=Fitted, colour = "Fitted"), size=1.2, linetype=2) +
  theme_bw() + theme(legend.title = element_blank()) +
  ylab("") + xlab("") +
  geom_vline(xintercept=as.numeric(as.Date("1959-12-01")), linetype=2) +
  ggtitle(paste0("ARIMA -- Holdout MAPE = ", round(100*MAPE,2), "%")) +
  theme(axis.text.x=element_text(angle = -90, hjust = 0))
```



This model predicts the holdout period quite well as measured by the MAPE (mean absolute percentage error). However, the model does not tell us much about the time series itself. In other words, we cannot visualize the "story" of the model. All we know is that we can fit the data well using a combination of moving averages and lagged terms.

A Bayesian Structural Time Series Model

A different approach would be to use Bayesian structural time series model with unobserved components. This technique is more transparent than ARIMA models and deals with uncertainty in a more elegant manner. It is more transparent because it does not rely on differencing, lags and moving averages to fit the data. You can visually inspect the underlying components of the model. It deals with uncertainty in a better way because you can quantify the posterior uncertainty of the individual components, control the variance of the components, and impose prior beliefs on the model. Last, but not least, any ARIMA model can be recast as a structural model.

Generally, we can write a Bayesian structural model like this:

$$Y_t = \mu_t + x_t\beta + S_t + e_t, e_t \sim N(0, \sigma_e^2)$$
$$\mu_{t+1} = \mu_t + \nu_t, \nu_t \sim N(0, \sigma_\nu^2).$$

Here x_t denotes a set of regressors, S_t represents seasonality, and μ_t is the *local level* term. The local level term defines how the latent state evolves over time and is often referred to as the *unobserved trend*. Note that the regressor coefficients, seasonality and trend are estimated *simultaneously*, which helps avoid strange coefficient estimates due to spurious relationships (similar in spirit to Granger causality). In addition, this approach facilitates model averaging across many smaller regressions, as well as coefficient shrinkage to promote sparsity. Also, given that the slope coefficients, β, are random we can specify outside priors for the regressor slopes in case we're not able to get meaningful estimates from the historical data.

The airline passenger dataset does not have any regressors, and so we'll fit a simple Bayesian structural model:

- 500 MCMC draws.
- Use 1960 as the holdout period.
- Trend and seasonality.
- Forecast created by averaging across the MCMC draws.
- Credible interval generated from the distribution of the MCMC draws.
- Discarding the first MCMC iterations (burn-in).
- Using a log transformation to make the model multiplicative