Paper 3188-2019

# The Bayesians are Coming! The Bayesians are Coming! The Bayesians are Coming to Time Series!

Aric LaBarr, Institute for Advanced Analytics at North Carolina State University

## ABSTRACT

With the computational advances over the past few decades, Bayesian analysis approaches are starting to be fully appreciated. Forecasting and time series also have Bayesian approaches and techniques, but most people are unfamiliar with them due to the immense popularity of Exponential Smoothing and ARIMA classes of models. However, Bayesian modeling and time series analysis have a lot in common! Both are based on using historical information to help inform future modeling and decisions. This talk will compare the classical Exponential Smoothing and ARIMA class models to Bayesian models with autoregressive components. It will compare results from each of the classes of models on the same data set as well as discuss how to approach Bayesian time series models in SAS.

## INTRODUCTION

Forecasting has applications across all industries. From needing to predict future values of sales for a product line, energy usage for a power company, to volatility of a portfolio of assets to hedge against risk, forecasting provides needed answers to decision makers. Time series models are extensively used in industry for this very purpose. Time series models and the underlying data are different in design than traditional cross-sectional modeling techniques. Typical cross-sectional modeling techniques are calculated across many individuals or objects usually under some assumption of independence. Whether they are traditional statistical models or newer machine learning algorithms, the techniques ignore, or are calculated regardless of, any time component. A simple example of this is analyzing energy usage across buildings on a major university campus compared to the temperature outside to understand the relationship between the two. Time series models on the other hand use the evolutionary nature of a set of data across time to help with the forecasting of future values. Their data typically consists of series of observations measured on the same individual or object across many points in time. A similar example for time series modeling is following the progress of energy usage for a building across time and use the previous values of energy usage to forecast future ones.

Popular approaches to time series forecasting are exponential smoothing models and combinations/variations on autoregressive and moving average models called ARIMA models. Both of these families of approaches are typically done with frequentist statistical methodology compared to Bayesian statistical methodology. Bayesian statistical methodology is named after Thomas Bayes, an English statistician, philosopher, and Presbyterian minister. The overly simplified difference between Bayesian and frequentist statistical methodology revolves around the estimation of unknown population parameters such as coefficients in a regression model. To a frequentist, these coefficients are fixed quantities that we are trying to estimate with a single point. Variability exists in these point estimates, but they are still single point estimates. To a Bayesian, these coefficients are random variables that follow a probability distribution, so they estimate entire distributions instead of single points. To do this, Bayesians make assumptions on the prior state of coefficients to inform their future thought on the coefficients' distributions. Each are making assumptions on the parameters of interest, the frequentist assuming the parameter doesn't move, and the Bayesian assuming that not only does the parameter move, but they know how the parameter moves.

Theoretical arguments aside, practitioners benefit from having knowledge of both frequentist and Bayesian time series modeling approaches. The more techniques a practitioner has, the better chance they have at providing the best solution to the decision maker using the forecasts, which is the true end goal. This paper will review the traditional frequentist approaches to time series – exponential smoothing models, ARIMA models, and VAR models. Next, the paper will introduce Bayesian time series approaches – Bayesian autoregressive models and Bayesian VAR models – hoping to build out the readers tools for forecasting real world problems using SAS® software.

## CLASSICAL TIME SERIES MODELING

At its heart, time series analysis basically tries to break down a series of data into two primary components – signal and noise. We extract the signal from the data and repeat this signal into the future while using the noise to estimate variation in our signal. Specifically, in time series we rely on the assumption that the observations at a certain point in time depend on previous observations in time. There are two extremes to this approach. The first is the naïve model which depends only on the previous observation,

$$\hat{Y}_{t+h} = Y_t$$

where $Y_t$ is the last observed data point, and $\hat{Y}_{t+h}$ is the forecast going $h$ observations into the future. The second approach is the overall average that weights each observation in time equally:

$$\hat{Y}_{t+h} = \frac{1}{T}\sum_{t=1}^{T} Y_t$$

Exponential smoothing models as well as ARIMA class models are a balance between these two extremes. Throughout this paper, we will try to forecast the percentage change in quarterly United States personal consumption expenditure (PCE) – essentially household buying habits based on the price changes in consumer goods and services. Figure 1 displays

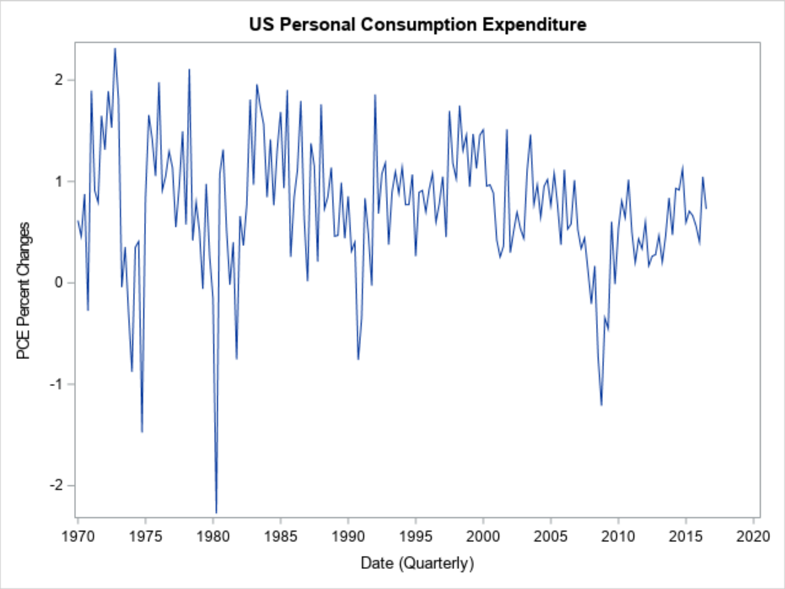

**Figure 1: US Personal Consumption Expenditures**

the quarterly US PCE from Q1 1970 through Q3 2016. For all of the analysis in this paper, the last seven observations – 2015 and 2016 – were removed and used as a hold-out sample for model accuracy reporting.

## EXPONENTIAL SMOOTHING MODELS

Exponential smoothing models (ESMs) became popular after WWII because of their accurate forecasting with minimal hand calculation. They are still popular today because those same principles are important to business modeling today. ESMs emphasize more recent observations in their weighting scheme for forecasting observations. There are many different types and classes of ESMs, but this paper uses the simple exponential smoothing model that doesn't include trend or seasonality as our data doesn't exhibit either. The simple (or single) ESM applies a weighting scheme on observations that decreases exponentially the further back in time we go,

$$\hat{Y}_{t+1} = \theta Y_t + \theta(1-\theta)Y_{t-1} + \theta(1-\theta)^2 Y_{t-2} + \theta(1-\theta)^3 Y_{t-3} + \cdots$$

where $\theta$ is bounded between 0 and 1. The larger the value of $\theta$, the more that the most recent observation is emphasized as seen in Figure 2.

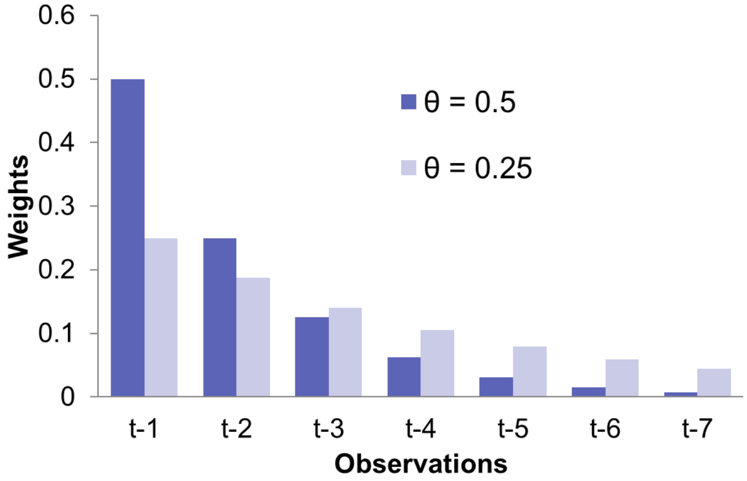

**Figure 2: Comparing Weights Across Theta Values**

The above exponentially decreasing weights simplify to the following equation:

$$\hat{Y}_{t+1} = \theta Y_t + (1-\theta)\hat{Y}_t$$

These models minimize the sum of the squared errors on the one-step ahead forecasts, essentially optimizing themselves to forecast one time period into the future.

The following SAS procedure computes the simple ESM:

```
proc esm data = train print = all plot = all lead = 7 outfor = for_esm;
  forecast consumption / model = simple;
run;
```

The optimized value of $\theta = 0.334$. The forecasts of a simple ESM are rather boring as they are a horizontal line at the forecast for the next time period. Since the simple ESM is optimized to predicting one time period into the future, it uses that one forecast infinitely into the future as seen in the forecast charts in Figure 3.

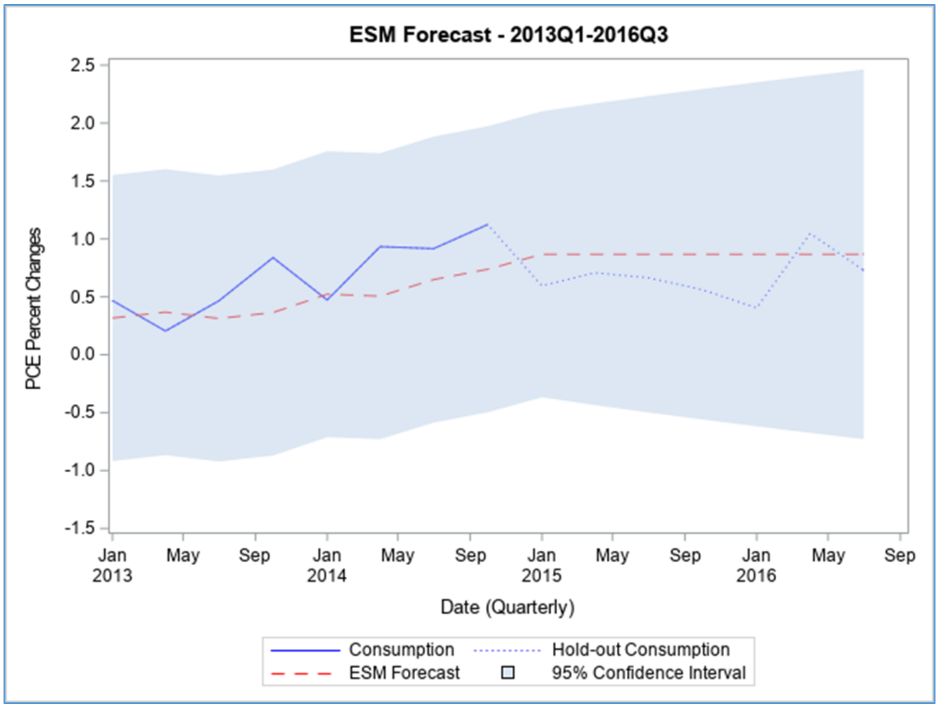

**Figure 3: ESM Forecast for US Personal Consumption Expenditure**

The code to subset the needed data and produce the previous figure is the following:

```
data esm_chart;
  set us_econ(rename=(Consumption=Full_Consumption));
  set for_esm;
run;

proc sgplot data=esm_chart(where=('01JAN2013'd <= Index)) noautolegend;
  title 'ESM Forecast - 2013Q1-2016Q3';
  series x = Index y = ACTUAL / lineattrs = (color = blue) name = "train"
    legendlabel = "Consumption";
  series x = Index y = Full_Consumption / lineattrs = (color = blue
    pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
  series x = Index y = PREDICT / lineattrs = (color = red pattern = dash)
    name = "for" legendlabel = "ESM Forecast";
  band x = Index upper=UPPER lower=LOWER / transparency=.5 name = "CI"
    legendlabel = "95% Confidence Interval";
  yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
  xaxis label='Date (Quarterly)';
  keylegend "train" "test" "for" "CI";
run;
```

ESMs predict fairly well since that is their design, however, they may still not find all of the signal in a series.

## ARIMA MODELS

Exponential smoothing models limit the structure on how previous observations impact future forecasts to only exponentially decreasing. There is no guarantee that that specific structure must hold true. Autoregressive (AR) models forecast a series of observations based solely on past values. The main difference from ESMs is that AR models allow each lag, or previous observation, to have values not necessarily decreasing exponentially the further back in time you go. In fact, there may come a point when previous lags values of a series of data no longer impacts the current observation and the AR model stops at that lag and doesn't continue going further back in time. AR models are long memory models where the effects of observations dissipate over many time periods.

The following is the AR model that has $p$ lags:

$$\hat{Y}_t = \hat{\alpha}_0 + \hat{\alpha}_1 Y_{t-1} + \hat{\alpha}_2 Y_{t-2} + \cdots + \hat{\alpha}_p Y_{t-p}$$

To help mitigate large values of $p$, combining AR models with moving average (MA) models to form ARIMA models can reduce the parameterization of a model. MA models are short memory models that forecast future values based solely on errors from previous time points. The MA model with $q$ previous error terms is the following:

$$\hat{Y}_t = \hat{\alpha}_0 + \hat{\beta}_1 \varepsilon_{t-1} + \hat{\beta}_2 \varepsilon_{t-2} + \cdots + \hat{\beta}_q \varepsilon_{t-q}$$

Combining these models together form the ARIMA model. The I in ARIMA stands for integrated and is not covered here in this paper as it deals with making the data stationary, which our example data is.

Three available model selection techniques in SAS are the minimum information criterion (MINIC), the smallest canonical correlation (SCAN), and the extended sample autocorrelation function (ESACF). The technical details for each of these is not provided here in this paper, but each provides an estimate of the number of AR and MA lags needed to best fit the data. The following code runs these selection techniques looking up to 4 lags into the past:

```
proc arima data=train plot=all;
  identify var=consumption nlag=10 minic scan esacf P=(0:4) Q=(0:4);
run;
quit;
```

Two of the techniques – the MINIC and SCAN – agree that the AR(3) model with no MA terms is the best model:

$$\hat{Y}_t = \hat{\alpha}_0 + \hat{\alpha}_1 Y_{t-1} + \hat{\alpha}_2 Y_{t-2} + \hat{\alpha}_3 Y_{t-3}$$

The following code builds the AR(3) model and forecasts seven observations forward:

```
proc arima data=train plot=all;
  identify var=consumption nlag=10;
  estimate p=3 method=ML;
  forecast lead=7 out=for_arima;
run;
quit;
```

The forecast by the AR(3) model suggests a slight downward projection for future values of PCE as seen in Figure 4.

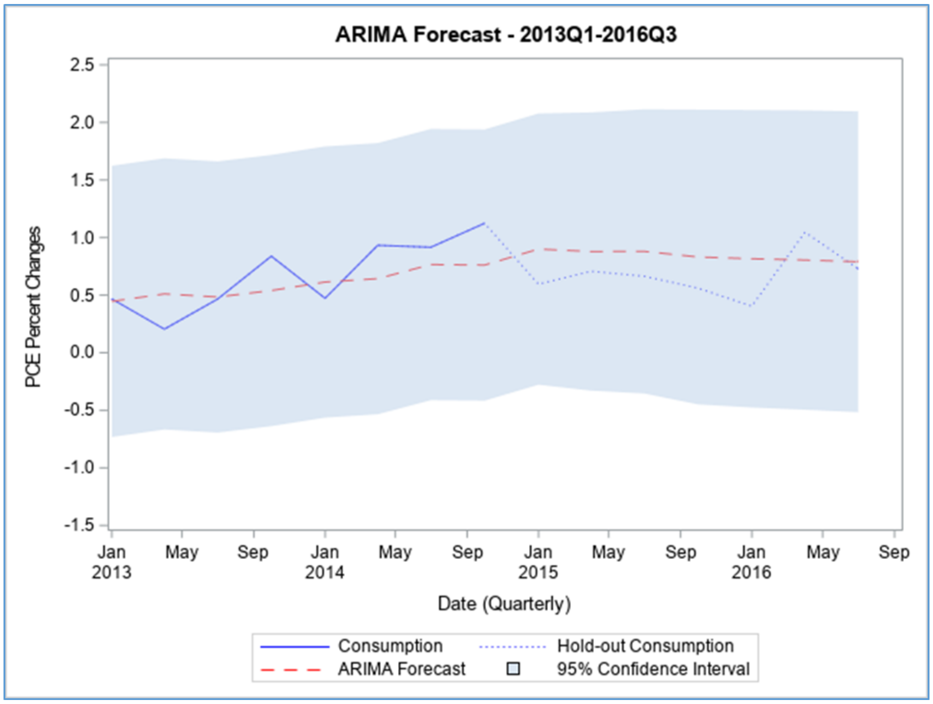

**Figure 4: ARIMA Forecast for US Personal Consumption Expenditure**

The code to subset the needed data and produce the previous figure is the following:

```
data arima_chart;
  set us_econ(rename=(Consumption=Full_Consumption));
  set for_arima;
run;

proc sgplot data=arima_chart(where=('01JAN2013'd <= Index)) noautolegend;
  title 'ARIMA Forecast - 2013Q1-2016Q3';
  series x = Index y = Consumption / lineattrs = (color = blue)
    name = "train" legendlabel = "Consumption";
  series x = Index y = Full_Consumption / lineattrs = (color = blue
    pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
  series x = Index y = Forecast / lineattrs = (color = red pattern = dash)
    name = "for" legendlabel = "ARIMA Forecast";
  band x = Index upper=U95 lower=L95 / transparency=.5 name = "CI"
    legendlabel = "95% Confidence Interval";
  yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
  xaxis label='Date (Quarterly)';
  keylegend "train" "test" "for" "CI";
run;
```

Both ARIMA models and ESMs are univariate classes of models, predicting only one series at a time. However, what if PCE moves with changes in personal income? Forecasting both of these simultaneously might provide further information.

## VECTOR AUTOREGRESSIVE MODELS

Multivariate analysis tries to predict many different response variables at the same time in the same model. Instead of single variables, we use vectors of variables. Instead of single coefficients, matrices of coefficients are used. An example of a vector AR(1) model looks like the following:

$$\begin{bmatrix} Y_{t,1} \\ Y_{t,2} \end{bmatrix} = \begin{bmatrix} \alpha_{0,1} \\ \alpha_{0,2} \end{bmatrix} + \begin{bmatrix} \alpha_{11,1} & \alpha_{12,1} \\ \alpha_{21,1} & \alpha_{22,1} \end{bmatrix} \begin{bmatrix} Y_{t-1,1} \\ Y_{t-1,2} \end{bmatrix} + \begin{bmatrix} e_{t,1} \\ e_{t,2} \end{bmatrix}$$

By expanding out the matrices the single vector equation becomes two equations, one for each of the response variables:

$$Y_{t,1} = \alpha_{0,1} + \alpha_{11,1}Y_{t-1,1} + \alpha_{12,1}Y_{t-1,2} + e_{t,1}$$

$$Y_{t,2} = \alpha_{0,2} + \alpha_{21,1}Y_{t-1,1} + \alpha_{22,1}Y_{t-1,2} + e_{t,2}$$

The benefit may be better seen through this view of the models as it shows that each response variable depends not only on its previous lagged values, but also the lagged values of the other response variable. In our data set, this means that previous values of changes in PCE as well as previous values of changes in personal income impact the forecasts of PCE going forward. Due to our previous analysis using an AR(3) model for predicting univariate PCE, we use a VAR(3) model with percentage changes in personal income as the other response variable. The VAR(3) model is defined as follows:

$$\boldsymbol{\hat{Y}_t = \hat{\alpha}_0 + \hat{\alpha}_1 Y_{t-1} + \hat{\alpha}_2 Y_{t-2} + \hat{\alpha}_3 Y_{t-3}}$$

There is a different SAS procedure to deal with multivariate time series as compared to univariate time series:

```
proc varmax data = train plot=all;
  id index interval=quarter;
  model consumption income / p=3 lagmax=5 print=(estimates diagnose);
  output out=for_var lead=7;
run;
```

Similar to the AR(3) model, the VAR(3) model also forecasts a slight downward trend over the next seven time periods as seen in Figure 5.

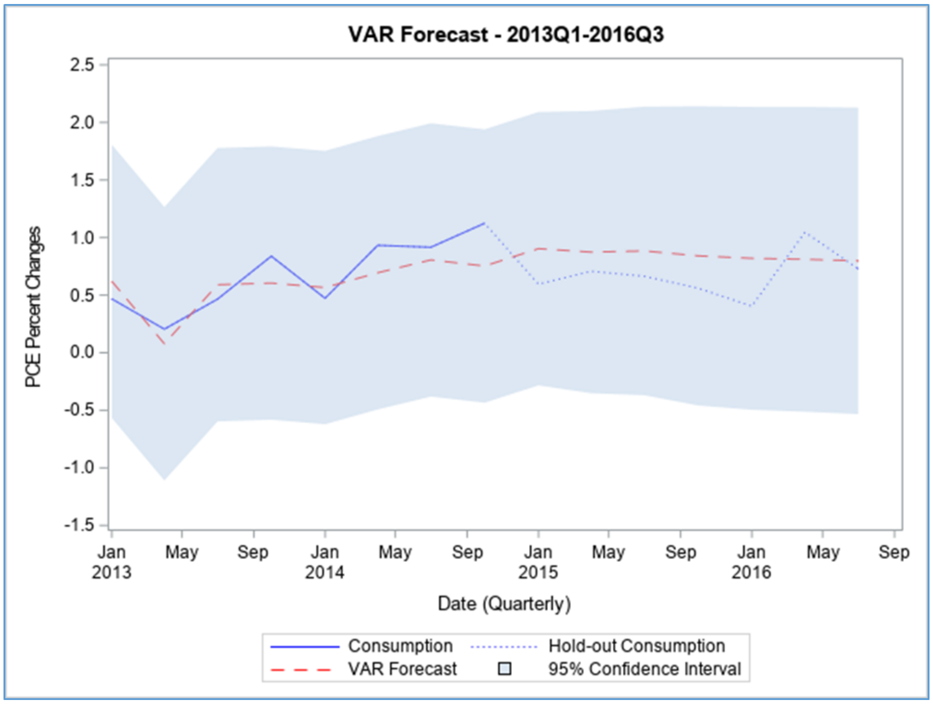

**Figure 5: VAR Forecast for US Personal Consumption Expenditures**

The code to subset the needed data and produce the previous figure is the following:

```
data var_chart;
  set us_econ(rename=(Consumption=Full_Consumption));
  set for_var;
run;

proc sgplot data=var_chart(where=('01JAN2013'd <= Index)) noautolegend;
  title 'VAR Forecast - 2013Q1-2016Q3';
  series x = Index y = Consumption / lineattrs = (color = blue)
    name = "train" legendlabel = "Consumption";
  series x = Index y = Full_Consumption / lineattrs = (color = blue
    pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
  series x = Index y = For1 / lineattrs = (color = red pattern = dash)
    name = "for" legendlabel = "VAR Forecast";
  band x = Index upper=UCI1 lower=LCI1 / transparency=.5 name = "CI"
    legendlabel = "95% Confidence Interval";
  yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
  xaxis label='Date (Quarterly)';
  keylegend "train" "test" "for" "CI";
run;
```

This section has served as a brief review of popular time series models. All of the previous models take a frequentist approach to modeling time series. Time series models inherently depend on previous knowledge using lagged variables. The Bayesian framework of modeling relies on previous assumptions about data, which fits in perfectly with time series.

## BAYESIAN TIME SERIES MODELING

The wonderful part about Bayesian time series modeling is that the structures of the models are mostly identical to frequentist models. The main difference is the assumptions. Instead of just taking the inputs into our model – the previous data and parameters – as fixed values we are either using or estimating, we assume that they are random variables with corresponding distributions.

### BAYESIAN AUTOREGRESSIVE MODELS

We are going to fit a Bayesian AR(3) model since the AR(3) structure seemed to fit our data the best according to model selection techniques. Estimating the distribution of input variables, here the lagged values of consumption, is rather easy to do. Since the residuals

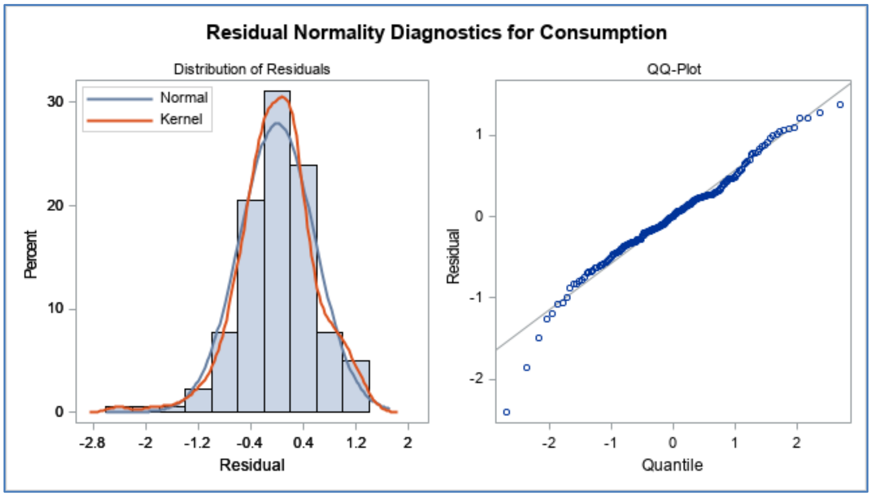

**Figure 6: Residuals from AR(3) Model**

from our original AR(3) model are approximately normal as seen in Figure 6, with only a little difference occurring with some outliers in the tail, we will assume the consumption model also follows a normal distribution with a mean of the AR(3) estimated AR(3) model and variance of $\sigma^2$. However, we must also assume distributions on all of the parameters from the AR(3) model – $\alpha_1, \alpha_2, \alpha_3, \sigma^2$. Table 1 lists all of the assumed distributions for the model.

| **Random Variables** | **Assumed Distribution** |
|---|---|
| Consumption | Normal<br>Mean = AR(3), S.D. = $\sigma^2$ |
| Consumption Initial Value 0 | Normal<br>Mean = 0, S.D. = 1 |
| Consumption Initial Value 1 | Normal<br>Mean = 0, S.D. = 1 |
| Consumption Initial Value 2 | Normal<br>Mean = 0, S.D. = 1 |
| $\alpha_1$ – AR(1) Parameter | Normal<br>Mean = 0, S.D. = 1 |
| $\alpha_2$ – AR(2) Parameter | Normal<br>Mean = 0, S.D. = 1 |
| $\alpha_3$ – AR(3) Parameter | Normal<br>Mean = 0, S.D. = 1 |
| $\sigma^2$ – Model Variance | Gamma<br>Shape = 3/10, Scale = 10/3 |

**Table 1: Prior Distributions in Bayesian AR(3) Model**

In Bayesian times series analysis, we estimated the posterior distributions of the final forecasts through Markov Chain Monte Carlo (MCMC) techniques. Specifically, PROC MCMC in SAS uses the random walk Metropolis algorithm to calculate the posterior distribution. The mathematical details of this algorithm is not covered in this paper, but the general idea is that we want to simulate stepping through a series of probabilistic events connected to one another. If we draw enough samples from this series of events, we will eventually get a distribution that resembles the posterior distribution that we desire – forecasted future values. For each forecasted future value we actually have a distribution of possible future values at that time point. We take the mean of the distribution of possible values at each of the seven forecasted time points developing our forecasts over the next seven quarters. SAS easily does this with PROC MCMC:

```
proc mcmc data = train nmc = 100000 seed = 100 nthreads = 8 propcov=quanew;
  parms alpha_1 alpha_2 alpha_3;
  parms sigma2 1;
  parms Y_0 Y_1 Y_2;

  prior alpha_: ~ normal(0,var = 1);
  prior sigma2 ~ igamma(shape = 3/10, scale = 10/3);
  prior Y_: ~ normal(0, var = 1 );

  mu = alpha_1*consumption.l1 + alpha_2*consumption.l2 +
       alpha_3*consumption.l3;
  model consumption ~ normal(mu, var = sigma2) icond=(Y_2 Y_1 Y_0);
```

```
    preddist outpred=predicted statistics=brief;

    ods output PredSumInt=for_bar;
  run;
```

with the NMC option specifying the number of MCMC iterations. The remaining pieces of the code define the prior distributions and the form of the model to be forecasted. The variables consumption.l1, consumption.l2, and consumption.l3 are lag values created by PROC MCMC, which makes the managing of data much easier.

The Bayesian AR(3) model seems to have the closest fit yet of the forecasted models for consumption as seen in Figure 7.

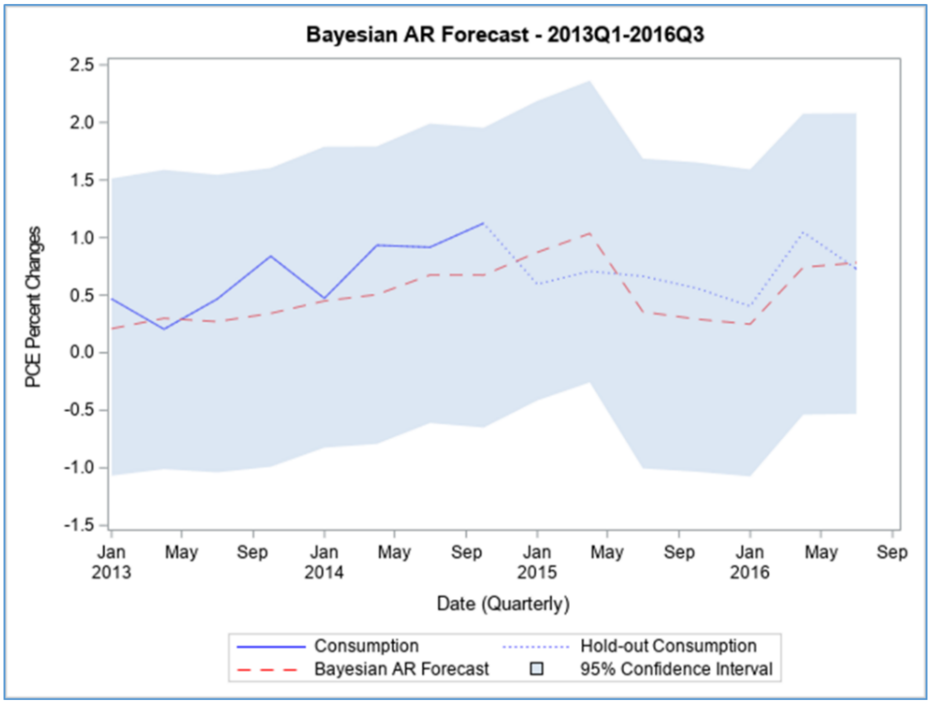

**Figure 7: Bayesian AR(3) Forecast of US Personal Consumption Expenditures**

The code to subset the needed data and produce the previous figure is the following:

```
  data bar_chart;
    set us_econ(rename=(Consumption=Full_Consumption));
    set train;
    set for_bar;
  run;

  proc sgplot data=bar_chart(where=('01JAN2013'd <= Index)) noautolegend;
    title 'Bayesian AR Forecast - 2013Q1-2016Q3';
    series x = Index y = Consumption / lineattrs = (color = blue)
      name = "train" legendlabel = "Consumption";
    series x = Index y = Full_Consumption / lineattrs = (color = blue
      pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
    series x = Index y = Mean / lineattrs = (color = red pattern = dash)
      name = "for" legendlabel = "Bayesian AR Forecast";
    band x = Index upper=hpdupper lower=hpdlower / transparency=.5
```

```
      name = "CI" legendlabel = "95% Confidence Interval";
    yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
    xaxis label='Date (Quarterly)';
    keylegend "train" "test" "for" "CI";
  run;
```

If frequentist AR models can be extended into the multivariate space, we can do the same thing with Bayesian AR models as well. Again, we will try to forecast both changes in PCE and personal income simultaneously to try and improve our model.

## BAYESIAN VECTOR AUTOREGRESSIVE MODELS

Similar to the extension of the Bayesian AR model to the frequentist AR model framework, the Bayesian extension to vector AR models is rather straight-forward. The model structure for the Bayesian VAR model is the same as the frequentist VAR model, but with additional assumptions on the parameters and known lagged values of the data. In PROC VARMAX, the multivariate normal distribution is assumed for the needed data and parameters. The only options to control the fit of the distribution are the hyperparameters in these Bayesian VAR models with trial and error being the best approach.

Similar to the previous models, three lagged values are used in the model – a Bayesian VAR(3):

```
  proc varmax data = train plot=all;
    id index interval=quarter;
    model consumption income / p=3 lagmax=5
          print=(estimates diagnose)
          prior=(lambda=0.9 theta=0.1);
    output out=for_bvar lead=7;
  run;
```

The Bayesian VAR(3) model is very similar in forecasts as the VAR(3) model as seen in Figure 8. The univariate models seemed to improve with the addition of the Bayesian framework, but the multivariate models seemed to remain approximately the same.

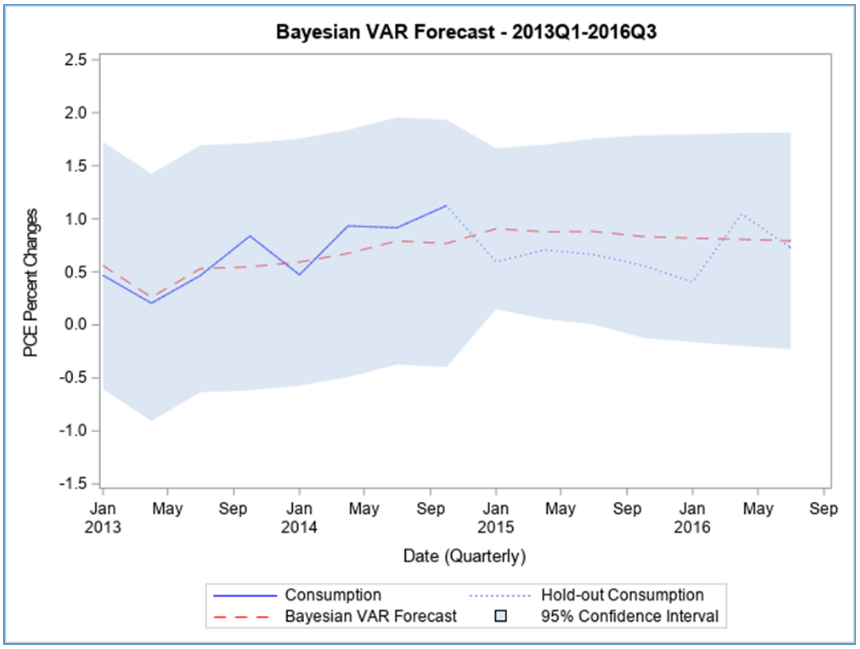

**Figure 8: BVAR(3) Forecast of US Personal Consumption Expenditures**

The code to subset the needed data and produce the previous figure is the following:

```
data var_chart;
  set us_econ(rename=(Consumption=Full_Consumption));
  set for_var;
run;

proc sgplot data=var_chart(where=('01JAN2013'd <= Index)) noautolegend;
  title 'VAR Forecast - 2013Q1-2016Q3';
  series x = Index y = Consumption / lineattrs = (color = blue)
    name = "train" legendlabel = "Consumption";
  series x = Index y = Full_Consumption / lineattrs = (color = blue
    pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
  series x = Index y = For1 / lineattrs = (color = red pattern = dash)
    name = "for" legendlabel = "VAR Forecast";
  band x = Index upper=UCI1 lower=LCI1 / transparency=.5 name = "CI"
    legendlabel = "95% Confidence Interval";
  yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
  xaxis label='Date (Quarterly)';
  keylegend "train" "test" "for" "CI";
run;
```

Each of the models had varying forecasts for the future seven quarters of US PCE. Combining some of the forecasts into an ensemble might improve them even further.

## CONCLUSION

This paper summarized and introduced some basic tools that every forecasting practitioner should know. In each problem, it is best to compare across modeling techniques in a hold-out sample to know which one is best to use. Instead of just visualizing the results from above, we calculated the Mean Absolute Percentage Error for each forecast on the hold-out data set:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right|$$

Table 2 contains the MAPE values for each of the models proposed above.

| Model | Mean Absolute Percentage Error |
|---|---|
| Bayesian Autoregressive(3) | 37.4% |
| Autoregressive(3) | 41.3% |
| Bayesian Vector Autoregressive(3) | 41.6% |
| Vector Autoregressive(3) | 41.8% |
| Simple Exponential Smoothing Model | 43.4% |

**Table 2: MAPE Values From Proposed Models**

The univariate Bayesian AR(3) model performed the best when compared using MAPE on the hold-out sample of seven quarters. One last tool that any practitioner in forecasting should have is the technique of ensembling forecasts together. Visual inspection of both Figure 4 and Figure 7 show that our two top performing models predict in opposite directions. The AR(3) model overpredicts change in PCE, while the Bayesian AR(3) underpredicts it. Averaging these forecasts together proves to outperform either of the forecasts individually. The MAPE from this ensemble of forecasts – straight average of the two – is **22.4%**. This is a definite improvement easily seen in Figure 9.

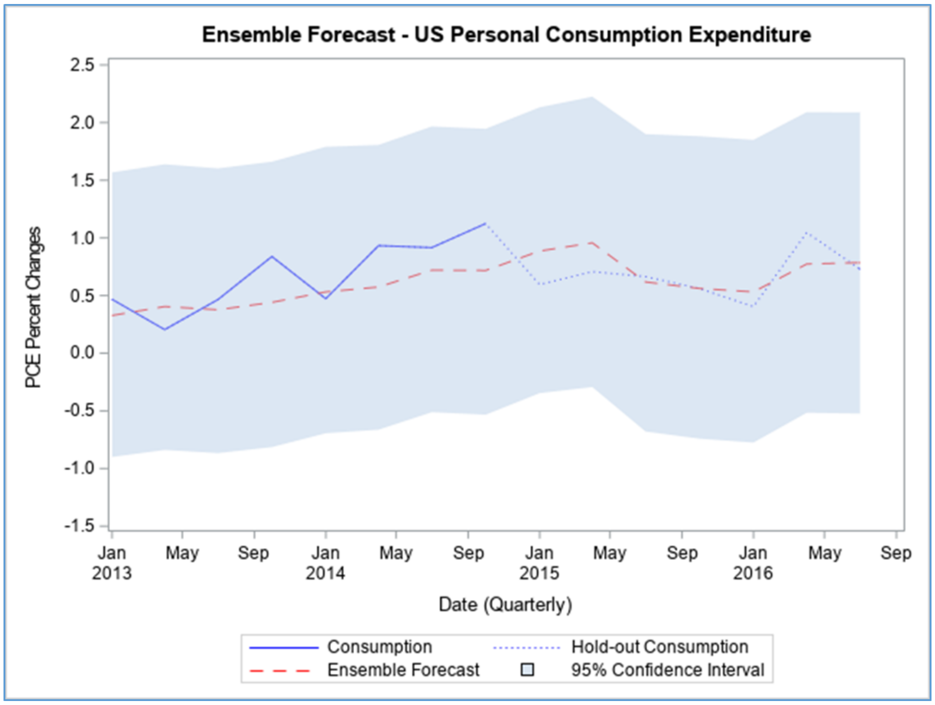

**Figure 9: Ensemble Forecast of US Personal Consumption Expenditures**

Both frequentist and Bayesian approaches to time series are valuable tools for any practitioner in forecasting. Each of the approaches contains their own theoretical foundation, which we will not argue for one way or the other as each forecasting problem is unique and there are no clear favorites to approaches. Every approach is a viable one for forecasting problems that arise, especially when you can combine them through ensembling of forecasts. Only the practitioner of the problem would know best. Hopefully now the practitioner has more tools in their tool belt!

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Aric LaBarr
Institute for Advanced Analytics at North Carolina State University
919-513-4076
aric_labarr@ncsu.edu
http://www.ariclabarr.com