**<u>First Principle Models</u>**
12/02/2025

Climate science models can be complicated and confusing largely because the models are the products of the "standard approach" of many "ready-made" packages. However, these packages hind the fundamentals deep in the convoluted workflow of "model training and testing", as if the model itself is indeed a blackbox. This article intends to open up the "blackbox" using the first principles. The example used is the sea level changes over time at Key West, FL.

The first principle approach is to divide the model into three components: trend, seasonality, and autoregression. The components are additive so that one modeling step is used for each component and the final model is the sum of the three components. In fact, each step decomposes the response variable data into the component model plus the residual term which is then modeled with the next component plus residual term until all three component models are complete.

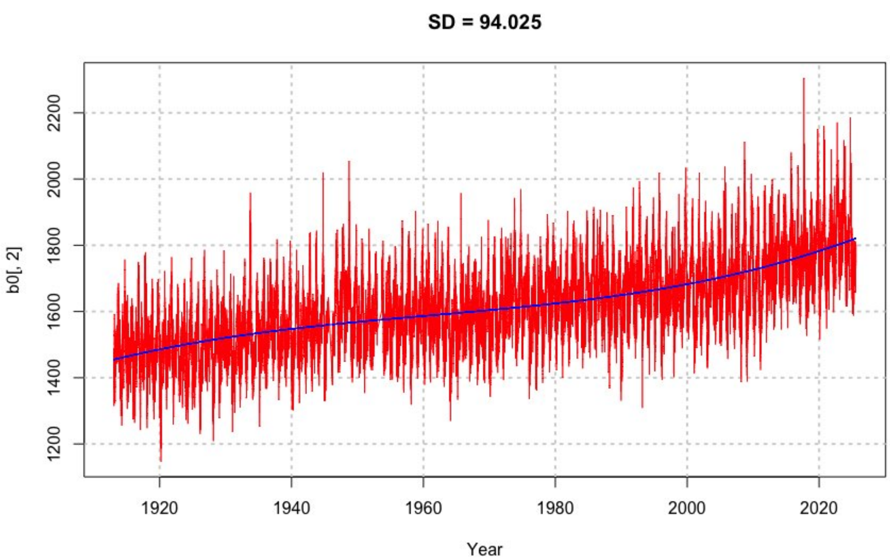


*Figure 1, Sea Level Measurements at Key West, FL.*

<u>Trend</u>

Figure 1 shows cubic linear model fitting the Key West data (data file d242.csv). A higher order linear model does not improve the root mean error (RME). The trend model is done with function "detrend()".

```
detrend=function(t,y){lmout=lm(y~t+I(t^2)+I(t^3)); return(lmout$fitted.values)}
```

<u>Seasonality</u>

A general approach to modeling seasonality is to use the Fourier transformation and ensure the model includes all important frequency components. Figure 2 shows the spectrogram of Fourier transformation of the residual term from the trend model.

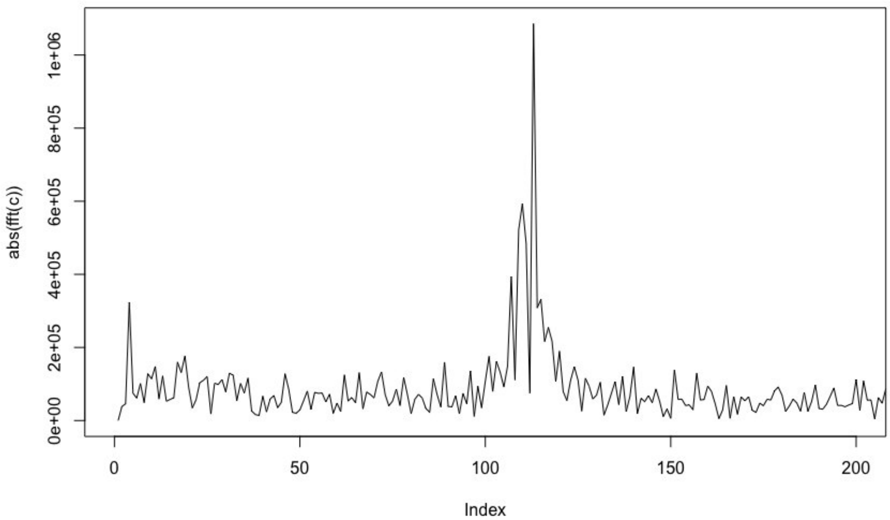


*Figure 2, Fourier transformation of the residual term from the trend model.*

Several major frequency components are located approximately at the frequency index range of 100 to 125. As such, the seasonality model must include all the frequency component below 150. Figure 3 shows the modeling results. The seasonality model is accomplished by function ftr():

```
ftr=function(y,pe=pe){ftr=fft(y);for (i in pe:length(ftr)) {ftr[i]=0};
       iff=Re(fft(ftr,inverse=T))/length(ftr); return(2*iff)}
```

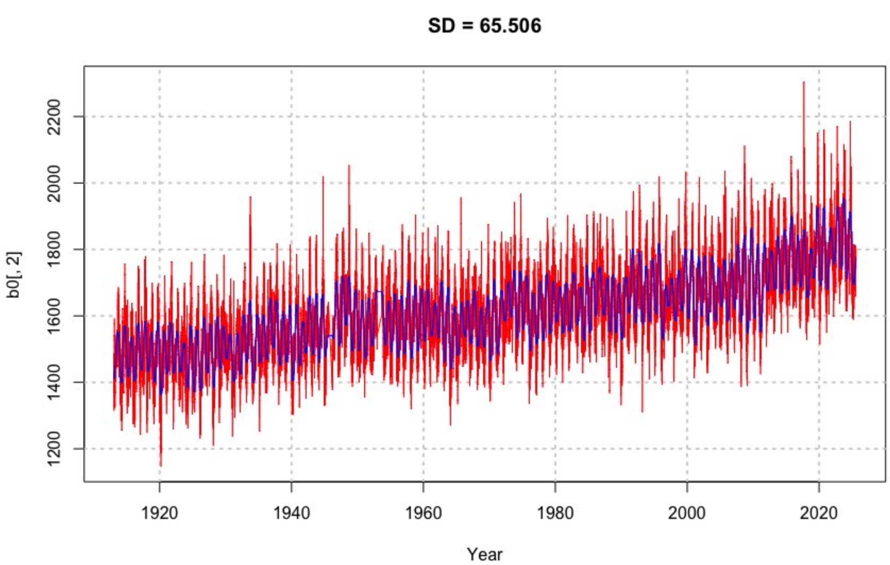


*Figure 3, Sea Level Measurements at Key West, FL and Blue Line Model of both the Trend and Seasonality.*

where parameter pe is the number of frequency components in the model. It is clear that the seasonality model reduces the RME or SD from 94.025 to 65.506, a significant improvement.

## Autoregression

While the trend model only uses time as the “explanatory variable”, the seasonality model uses frequencies as the explanatory variables. Autoregression, on the other hand, treats some previous response variable as the explanatory variables. A “lag” operator generates the corresponding variables which is determined using the partial autocorrelation function (PACF) of the residual term (d) from the seasonality model as shown in Figure 4. Eight lagged terms are to be used.

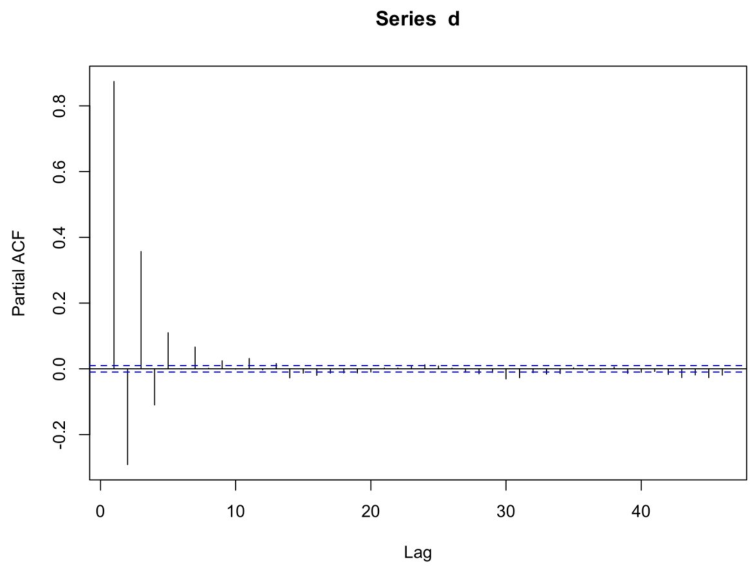


*Figure 4, Partial Autocorrelation Function of the Residual Term (d) from the Seasonality Model.*

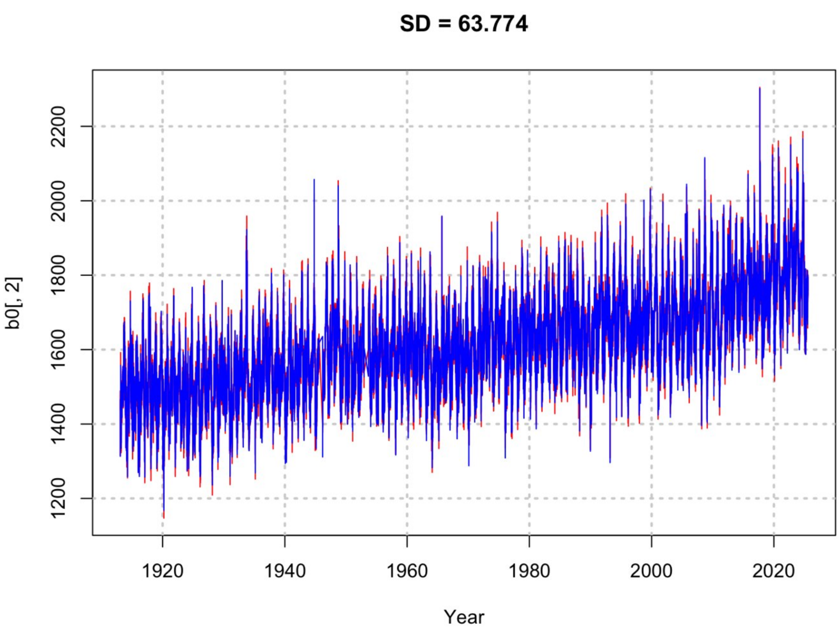


*Figure 5, Combined Model with Trend, Seasonality, and Autoregression Components.*

The lag operator and autoregression are made using lag(), auto() and relag() functions.

```
lag=function(x,n=1){c(rep(NA,n),x[-((length(x)-n+1):length(x))])}
auto=function(x,n){y=matrix(x,ncol=1);for(i in 1:n)
{y=cbind(y,lag(y[,i]))};return(as.data.frame(y))}
delag=function(x){lmout=lm(V1~. ,data=x); return(lmout$fitted.values)}
```

The combined model is shown in Figure 5. The RMSE or SD of the combined model is reduced to 63.774 from 65.506. A closeup view of the model is also in Figure 6.

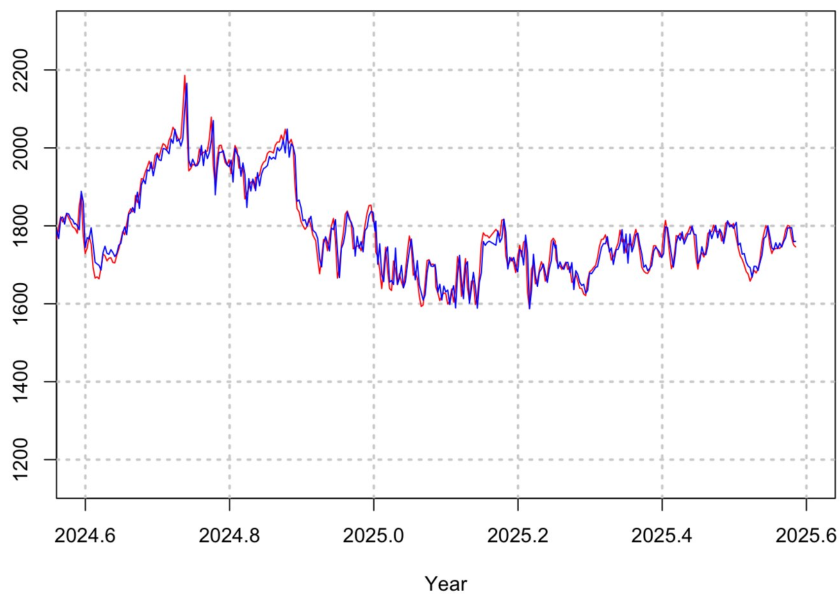


*Figure 6, A Closeup View of Figure 5 of Actual Measurements (red line) and Model Prediction (blue line)*

Appendix R Script

```
pe=200
input=function(fn){y=read.table(fn,header=F,sep=",");data.frame(date=y[,1]+(y[,2]-1)/
12+y[,3]/365,level=y[,4])}
lag=function(x, n=1){c(rep(NA, n),x[-((length(x)-n+1):length(x))])}
detrend=function(t,y){lmout=lm(y~t+I(t^2)+I(t^3)); return(lmout$fitted.values)}
auto=function(x,n){y=matrix(x,ncol=1);for(i in 1:n)
{y=cbind(y,lag(y[,i]))};return(as.data.frame(y))}
#delag=function(l1,l2,l3,x){lmout=lm(x~l1+l2+l3); return(lmout$fitted.values)}
#delag=function(l1,l2,x){lmout=lm(x~l1+l2); return(lmout$fitted.values)}
delag=function(x){lmout=lm(V1~. ,data=x); return(lmout$fitted.values)}
ftr=function(y,pe=pe){ftr=fft(y);for (i in pe:length(ftr)) {ftr[i]=0};
      iff=Re(fft(ftr,inverse=T))/length(ftr); return(2*iff)}
dec=function(y){return(sprintf("%.3f",y))}

b0=input("d242.csv");
b0=b0[b0[,2]>0,]; yr=b0[,1];

## trending
t=detrend(yr,b0[,2])
c=b0[,2]-t

## seasonality
plot(abs(fft(c)), type="l", xlim=c(0,200))
s=ftr(c,pe)
d=c-s

## autocorrelation
pacf(d)
y=auto(d,8)

a=delag(y)
s=e[-(1:8)]; t=t[-(1:8)]; yr=yr[-(1:8)]; d=d[-(1:8)]
h=d-a;

## combined
f=t+s+a
plot(b0[,1],b0[,2], type="n", xlab = "Year", main=paste("SD =",dec(sd(h)))); grid(lwd = 2)
lines(b0[,1],b0[,2],col="red")
lines(yr,f,col="blue")
```