Statistics 435/535/711 6 April 2022
P. Shaman

Seasonal ARIMA Model Fits—Dynamic Seasonal Estimation

A. Monthly U.S. beer production, January 1987—December 2017, revisited

The purpose of this reexamination of the beer data is to explore seasonal index estimates which arise from ARIMA estimation. We'll estimate dynamic seasonal structure (which ARIMA estimation allows) and also construct static seasonal estimates from the ARIMA model. And we'll compare the ARIMA static seasonal estimates to those obtained from regression.

Recall that in the regression fit for a decomposition model the seasonal component was modeled to be perfectly periodic. For example, for monthly series with an annual period, there were 12 seasonal indices, and these indices remained constant from one year to the next. That is, the estimation produced only a static seasonal estimate. By contrast, a seasonal ARIMA model fit allows estimation of a seasonal component that varies from one cycle (one year) to the next. That is, each seasonal cycle has its own set of indices. We will see that the seasonal indices do vary from one cycle to another in the two examples presented in these notes.

To begin, we recalculate and save the static seasonal estimates obtained from the regression model shown on pages 2–3 of the 24 January notes. The R command for the regression model is

```
lm(beer~time+I(time^2)+I(time^3)+I(time^4)+fmonth)
```

The static seasonal estimates derived from this model follow:

```
> seas

(Intercept)     fmonth2     fmonth3     fmonth4     fmonth5     fmonth6
 -0.7902230  -1.4325268   0.7884912   0.6568310   1.8049441   2.0721854
    fmonth7     fmonth8     fmonth9    fmonth10    fmonth11    fmonth12
  1.5725549   1.1594398  -0.5064181  -0.7808250  -2.0540390  -2.4904147
```

Note that the regression model used to obtain these static seasonal estimates does not include the calendar trigonometric pairs with frequencies 0.348 and 0.432, which were seen to be significant in the beer data (see page 3 of the 4 April notes). However, failure to include these variables in the regression model does not bias estimation of the static seasonal indices.

Next, we turn to fitting of a seasonal ARIMA model for the beer data. We repeat the development shown in the 4 April notes. First we use regression and calculation of residuals to remove calendar effects from the data (page 3 of the 4 April notes):

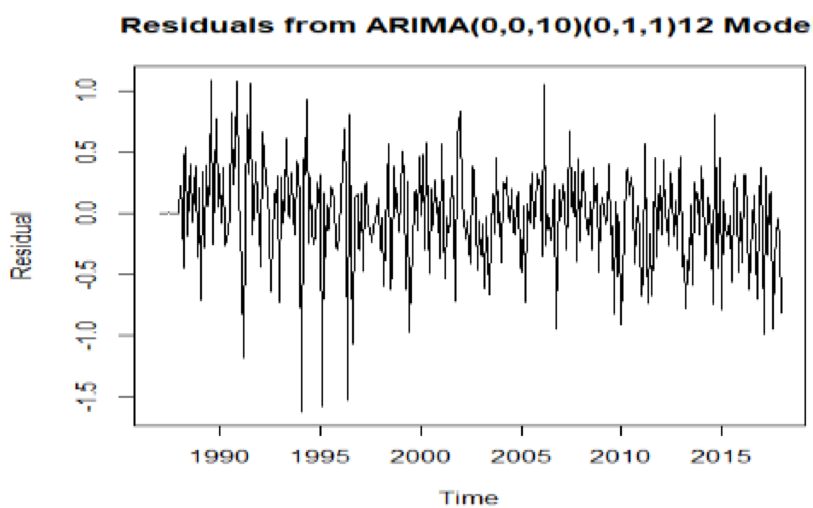
```
> model1<-lm(beer~c348+s348+c432+s432)
> modbeer<-resid(model1)
```

Next we fit the ARIMA(0,0,10)$(0,1,1)_{12}$ model to these residuals (page 6 of the 4 April notes):

```
> modelma<-
arima(modbeer,order=c(0,0,10),seasonal=list(order=c(0,1,1),period=12))
```

Although this model does not include ordinary differencing to remove trend, the residual analysis in the 4 April notes confirms the model achieves reduction to white noise. That is, the trend is not strong enough to warrant inclusion of ordinary differencing in the ARIMA model.

Recall that the first 12 residuals from this ARIMA model calculated by the R program do not have appropriate residual structure, and that we need to delete them for subsequent analysis. The plot of the residuals, repeated from page 7 of the 4 April notes, which follows, shows a shelf formed at the far left by the first 12 residual values.

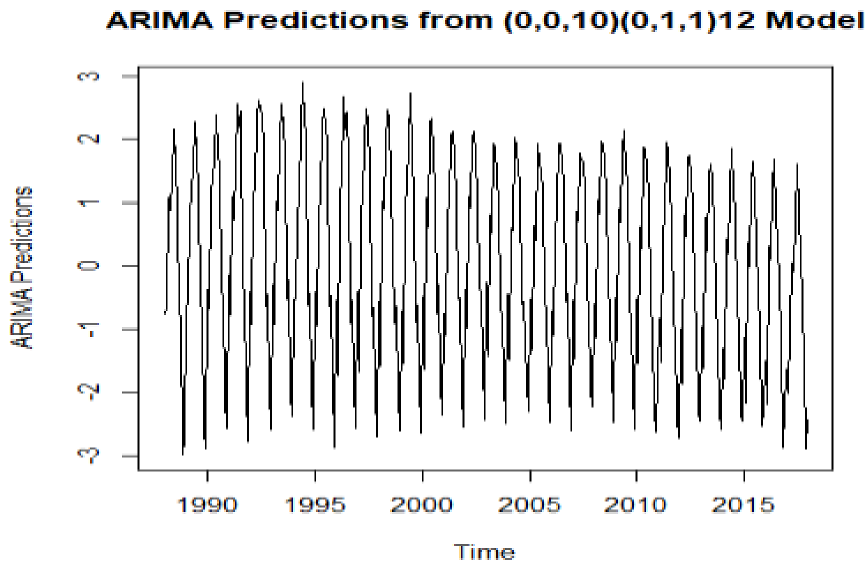


Next, we form the predicted values from the ARIMA model. They are predictions stripped of the effects of the calendar pairs with frequencies 0.348 and 0.432 in the data series. To obtain the predicted values, we take the residuals from the initial regression model which fits only the calendar trigonometric pairs and subtract from them the residuals from the ARIMA model. In doing this, we delete the first 12 residuals in each set.

```
> sel<-1:12
> arimapred<-resid(model1)[-sel]-resid(modelma)[-sel]
```

The plot of the ARIMA predicted values follows. The predicted values run from January 1988 through December 2017.

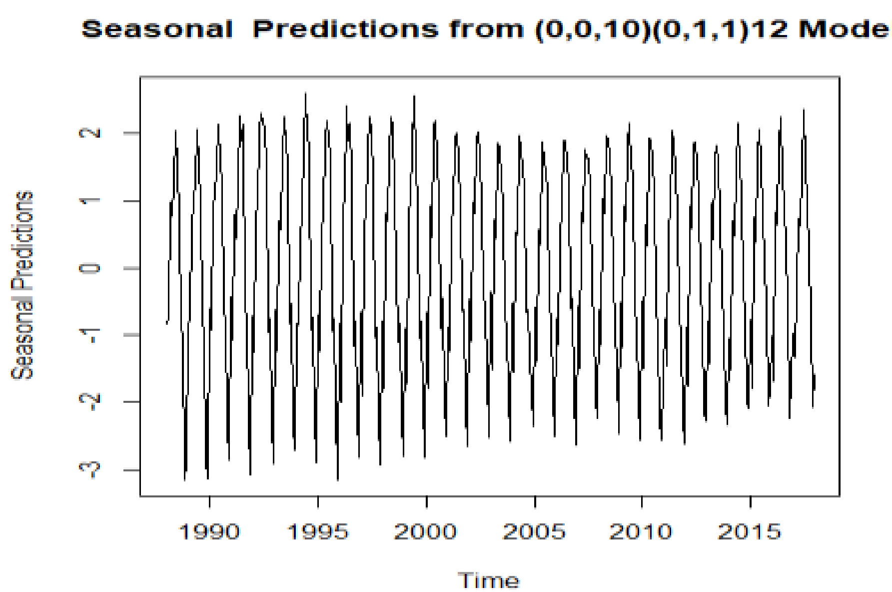
```
> arimapred.ts<-ts(arimapred,start=c(1988,1),freq=12)
> plot(arimapred.ts,xlab="Time",ylab="ARIMA Predictions",main="ARIMA
Predictions from (0,0,10)(0,1,1)12 Model")
```



The plot shows some remaining trend structure in the predicted values. We need to remove this trend, to leave only a representation of the seasonal structure. The removal will be accomplished with a regression.

```
> time<-as.numeric(1:360)
> arimapred2<-resid(lm(arimapred~poly(time,4)))
```

As stated, the residuals from this calculation represent predictions of just the seasonal structure in the beer time series. Both trend and calendar structures have now been removed. Here is the plot of the modified predicted values:



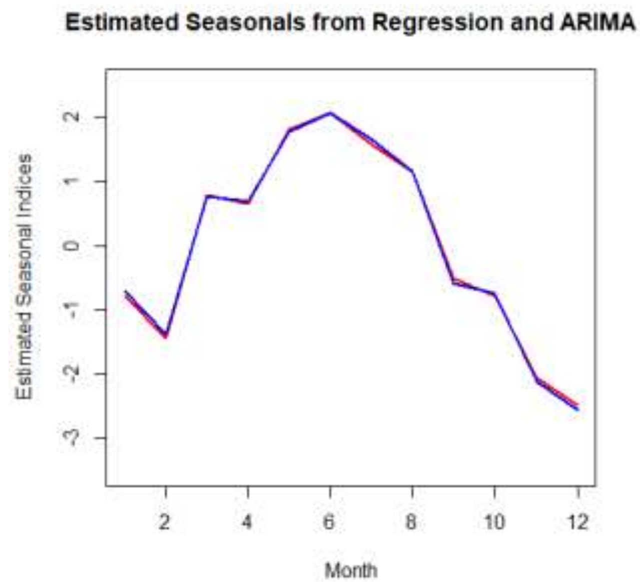
Estimation of static seasonal indices for the time range of the modified predicted values follows. These are static estimates obtained from the ARIMA model fit.

```
> monmeans<-tapply(arimapred2,month[-sel],mean)
> seas2<-monmeans-mean(monmeans)
> seas2
         1          2          3          4          5          6          7
-0.7143384 -1.3783248  0.7619614  0.6855102  1.7813141  2.0660444  1.6645069
         8          9         10         11         12
 1.1605749 -0.5971080 -0.7532279 -2.1177081 -2.5592048
```

Note the method used is calculation of monthly means, followed by adjustment so that the values add to 0.

The table and plot which follow compare the estimated static seasonal indices obtained from the regression and ARIMA models.

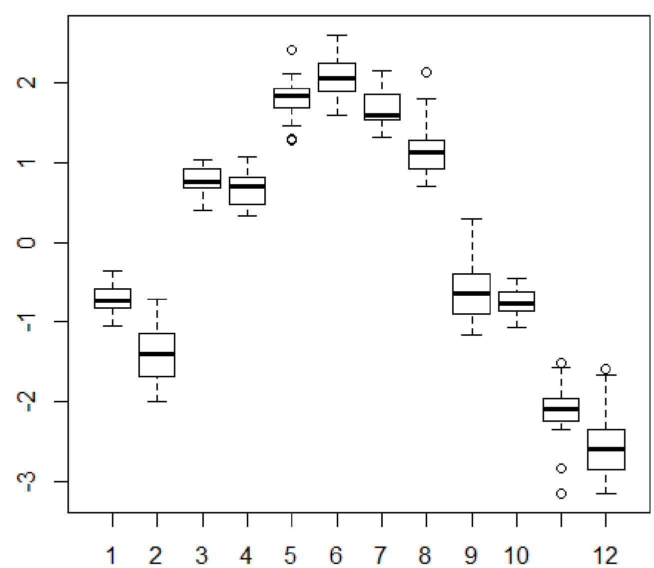
```
> cbind(1:12,seas,seas2)

Month Regression  ARIMA

  1 -0.7902230 -0.7143384
  2 -1.4325268 -1.3783248
  3  0.7884912  0.7619614
  4  0.6568310  0.6855102
  5  1.8049441  1.7813141
  6  2.0721854  2.0660444
  7  1.5725549  1.6645069
  8  1.1594398  1.1605749
  9 -0.5064181 -0.5971080
 10 -0.7808250 -0.7532279
 11 -2.0540390 -2.1177081
 12 -2.4904147 -2.5592048

> plot(ts(seas),xlab="Month",ylab="Estimated Seasonal
Indices",main="Estimated Seasonals from Regression and ARIMA",ylim=c(-
3.5,2.5),lty=1,lwd=2,col="red")
> lines(ts(seas2),lty=1,lwd=2,col="blue")
```



The regression estimates are in red and the ARIMA estimates in blue. The estimated seasonal indices from the two methods do differ somewhat for several of the months.

It is useful to look at side-by-side box plots for the ARIMA modified predicted values by month. For the plots the time range 1988 through 2017 is used.

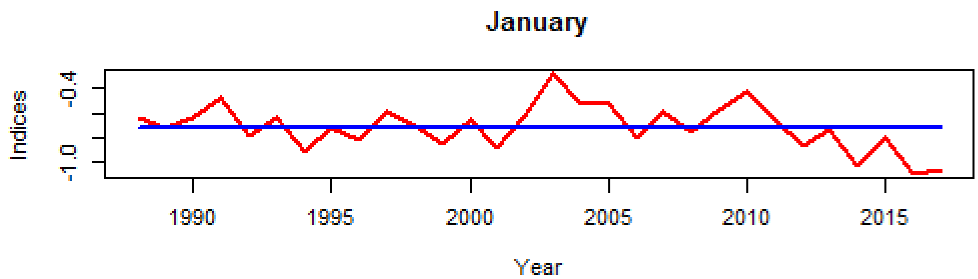
```
> arimapred2.ts<-ts(arimapred2,start=c(1988,1),freq=12)
> boxplot(arimapred2.ts~cycle(arimapred2.ts))
```

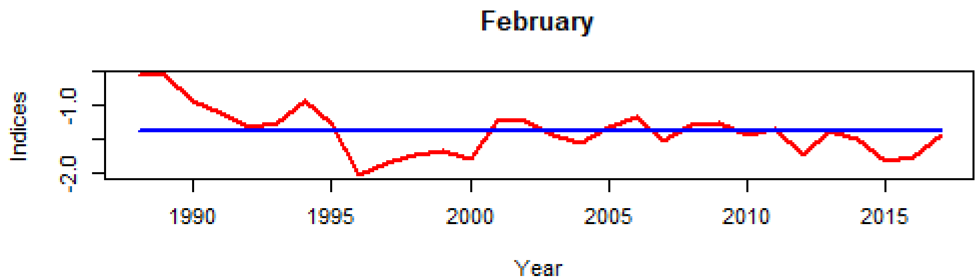


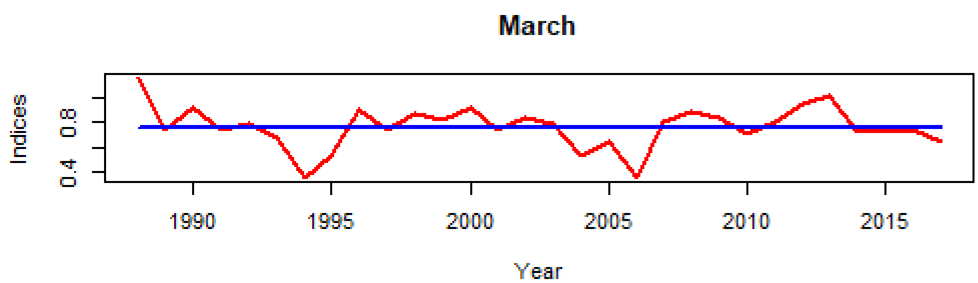
As we see, there is variation across the years for each month. This will permit estimation of a dynamic seasonal structure for the years 1988 through 2017.

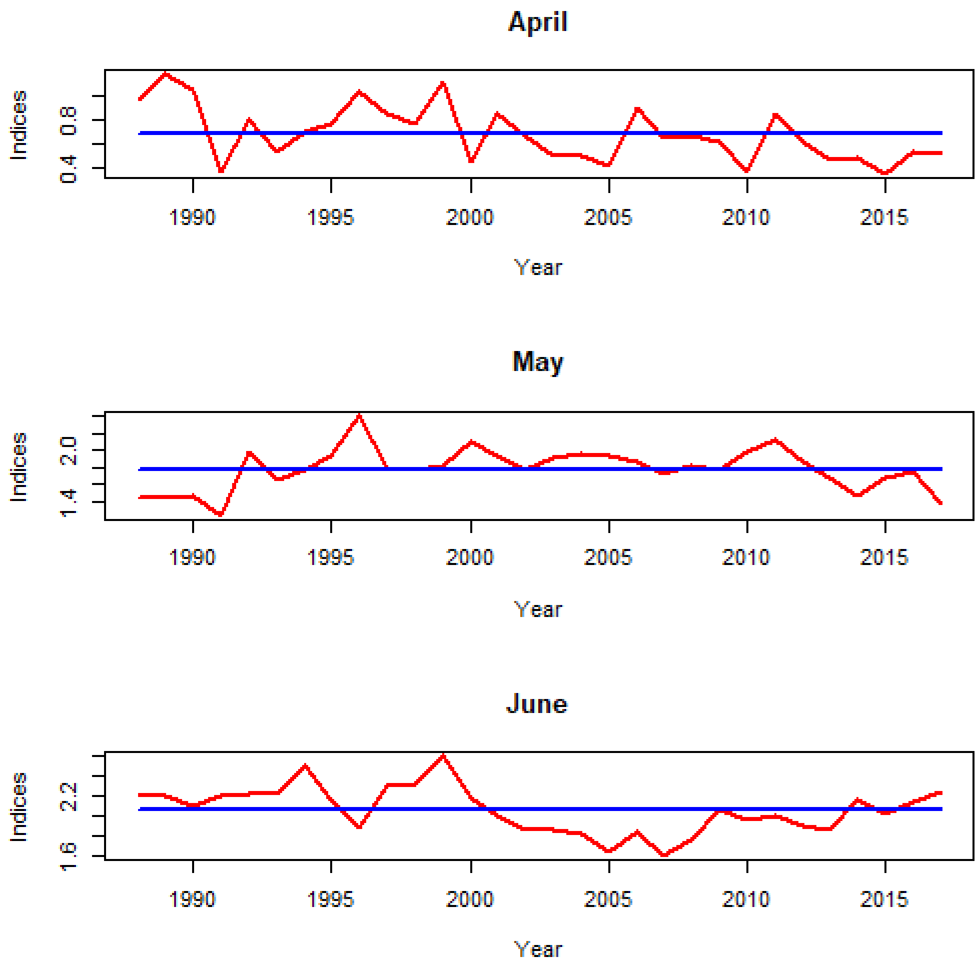
Twelve monthly plots follow, in the order January, February, …, December. Each red curve shows the dynamic estimation of the seasonal structure, giving evolution of the estimated seasonal index from 1988 to 2017 for a given month, as implied by the seasonal ARIMA model. And for each month the blue curve is the static estimate of the seasonal index obtained from the ARIMA model.

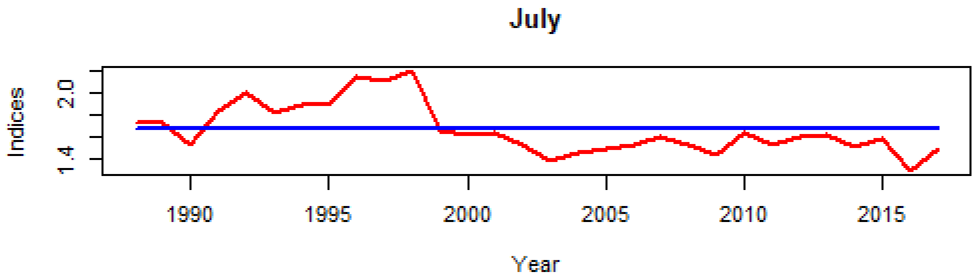
```
> y<-arimapred2
> seasm<-matrix(rep(0,360),ncol=30)
> j<--11
> for(i in 1:30){
+ j<-j+12;j1<-j+11
+ seasm[,i]<-y[j:j1]-mean(y[j:j1])
+ }
> year<-seq(1988,2017)
> seas2m<-matrix(rep(seas2,30),ncol=30)

> name<-
c("January","February","March","April","May","June","July","August","Se
ptember","October","November","December")
> par(mfrow=c(3,1))
> for(i in 1:3){
+
plot(year,seasm[i,],xlab="Year",ylab="Indices",main=name[i],type="l",lw
d=2,col="red")
+ lines(year,seas2m[i,],lty=1,lwd=2,col="blue")
+ }
```

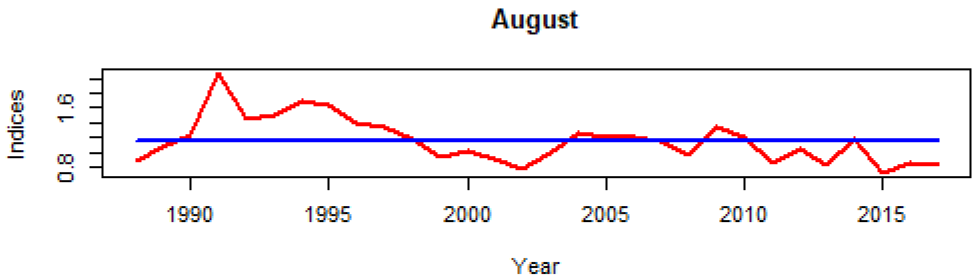

January
Indices
-0.4
-1.0
1990
1995
2000
2005
2010
2015
Year

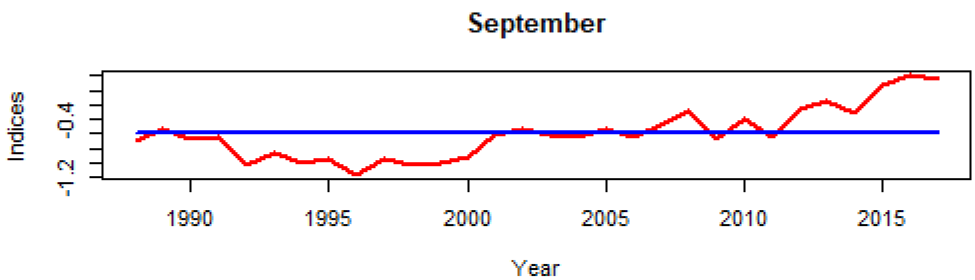

February
Indices
-1.0
-2.0
1990
1995
2000
2005
2010
2015
Year

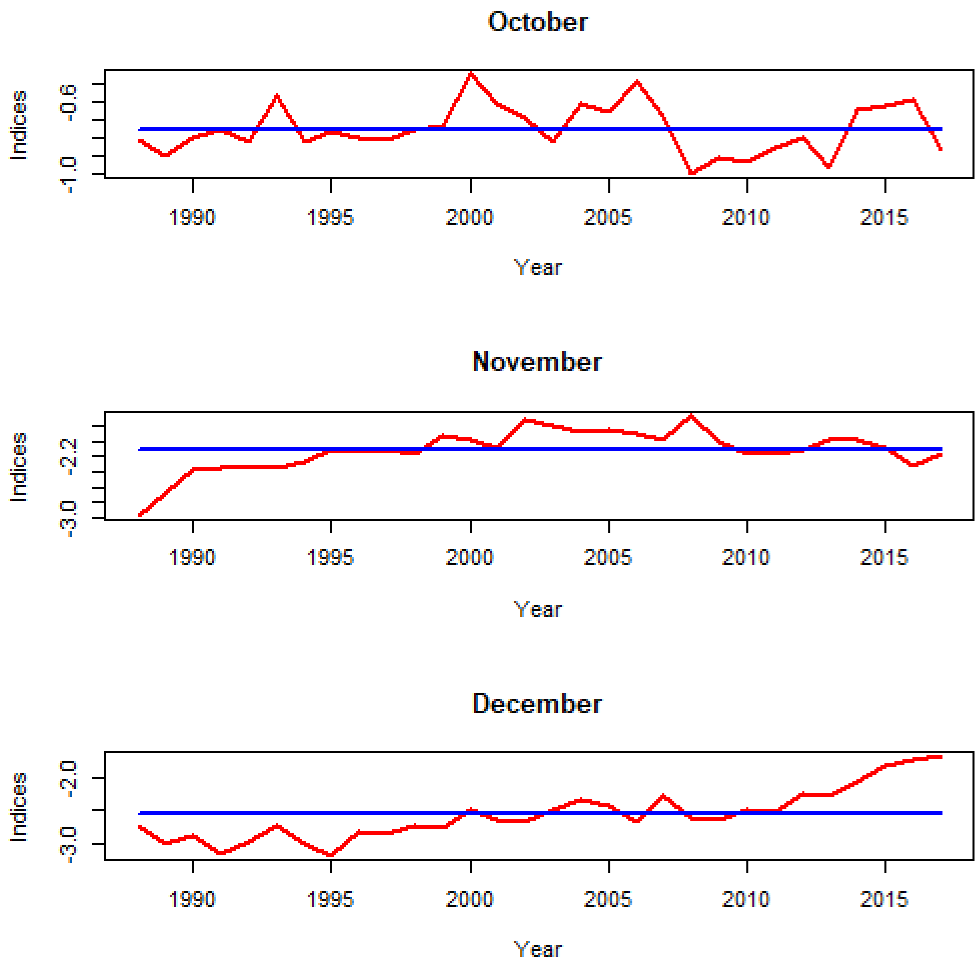

March
Indices
0.4
0.8
1990
1995
2000
2005
2010
2015
Year


April
Indices
0.4
0.8
1990
1995
2000
2005
2010
2015
Year
May
Indices
1.4
2.0
1990
1995
2000
2005
2010
2015
Year
June
Indices
1.6
2.2
1990
1995
2000
2005
2010
2015
Year


July
Indices
1.4
2.0
1990
1995
2000
2005
2010
2015
Year


August
Indices
0.8
1.6
1990
1995
2000
2005
2010
2015
Year


September
Indices
-1.2
-0.4
1990
1995
2000
2005
2010
2015
Year



The results show variation across time of the seasonal pattern. In later notes I'll address the Australian beer data similarly.

B. Monthly international airline passenger data, 1949-1960, revisited.

First let's fit a multiplicative model with regression and look at the residuals. With this model we'll record estimates of seasonal indices.

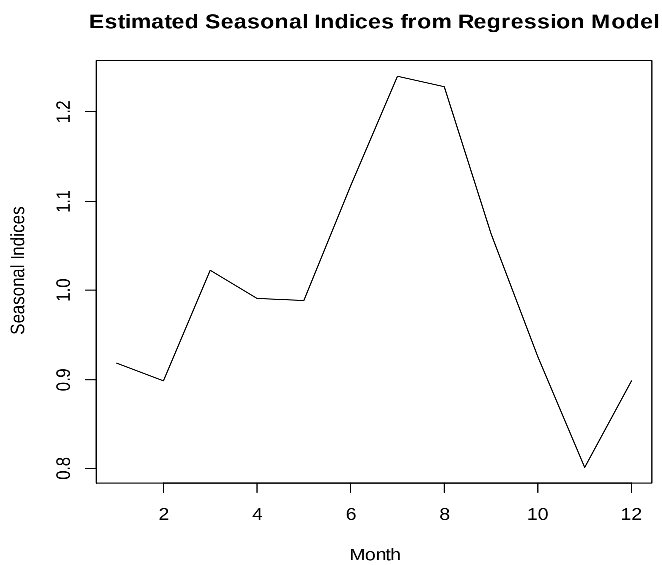
The regression fit which follows employs a quadratic polynomial for the trend and the month dummies to pick up the (static) seasonal component. The response is the log of the passenger count (count in thousands).

```
> lpass<-log(passengers)
> month<-rep(1:12,12)
> fmonth<-as.factor(month)
> time<-1:144

> model<-lm(lpass~time+I(time^2)+fmonth);summary(model)

Call:
lm(formula = lpass ~ time + I(time^2) + fmonth)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.12748 -0.03709  0.00418  0.03197  0.11529 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.651e+00  1.786e-02 260.410  < 2e-16 ***
time         1.318e-02  3.892e-04  33.877  < 2e-16 ***
I(time^2)   -2.148e-05  2.599e-06  -8.265 1.41e-13 ***
fmonth2     -2.227e-02  1.968e-02  -1.132 0.259839    
fmonth3      1.078e-01  1.968e-02   5.477 2.15e-07 ***
fmonth4      7.639e-02  1.968e-02   3.882 0.000164 ***
fmonth5      7.393e-02  1.968e-02   3.756 0.000259 ***
fmonth6      1.960e-01  1.968e-02   9.959  < 2e-16 ***
fmonth7      3.000e-01  1.969e-02  15.238  < 2e-16 ***
fmonth8      2.907e-01  1.969e-02  14.765  < 2e-16 ***
fmonth9      1.462e-01  1.969e-02   7.423 1.33e-11 ***
fmonth10     8.145e-03  1.970e-02   0.414 0.679912    
fmonth11    -1.354e-01  1.970e-02  -6.873 2.36e-10 ***
fmonth12    -2.132e-02  1.971e-02  -1.082 0.281286    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0482 on 130 degrees of freedom
Multiple R-squared:  0.9892,    Adjusted R-squared:  0.9881 
F-statistic: 912.7 on 13 and 130 DF,  p-value: < 2.2e-16
```

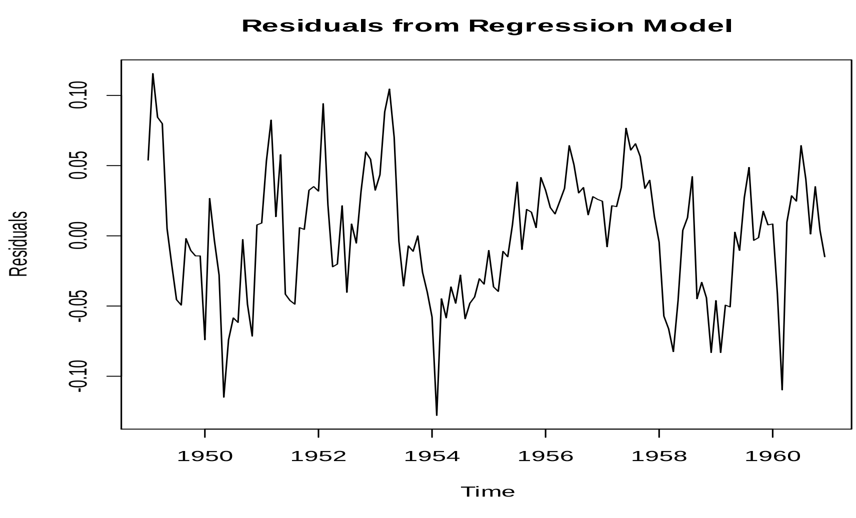
Let's construct the seasonal index estimates.

```
> b1<-coef(model)[1]
> b2<-coef(model)[4:14]+b1
> b3<-c(b1,b2)
> seasreg<-exp(b3-mean(b3))
```

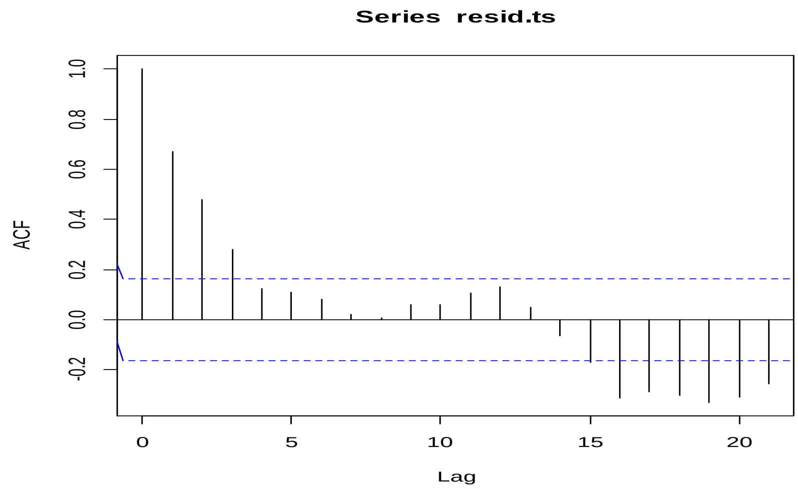
```
> seasreg

(Intercept)     fmonth2     fmonth3     fmonth4     fmonth5     fmonth6
  0.9185000   0.8982714   1.0230335   0.9914116   0.9889771   1.1174163
    fmonth7     fmonth8     fmonth9    fmonth10    fmonth11    fmonth12
  1.2398141   1.2283964   1.0630700   0.9260117   0.8021864   0.8991239

> plot(ts(seasreg),xlab="Month",ylab="Seasonal Indices",main="Estimated
Seasonal Indices from Regression Model")
```



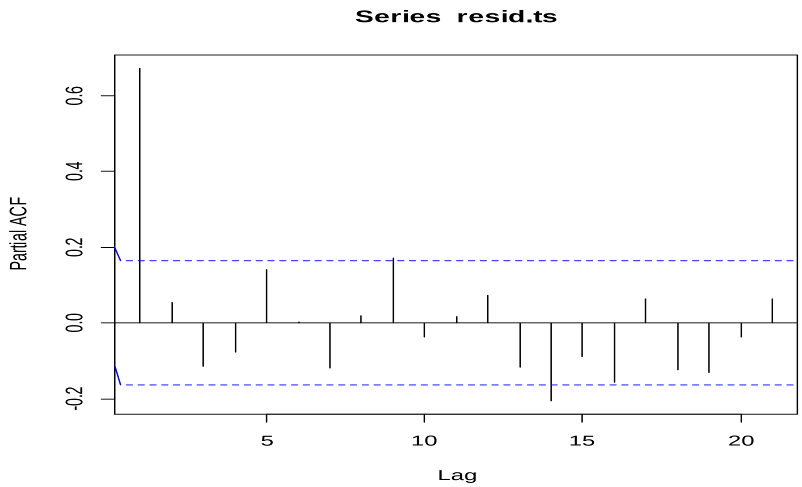
Next, let's look at the regression residuals.



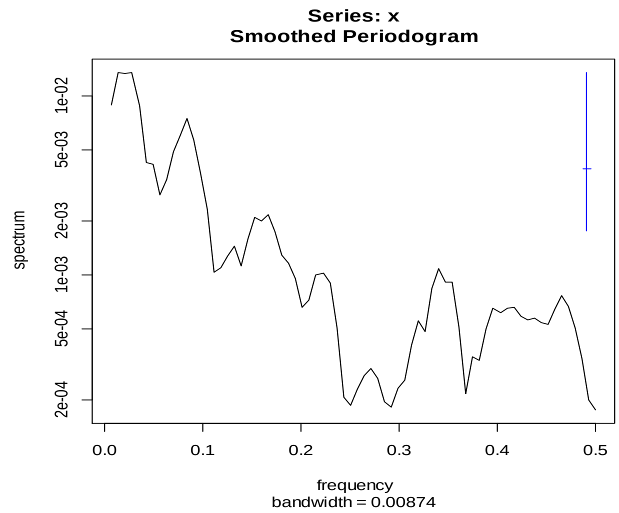

```
> resid.ts<-ts(resid(model))
> acf(resid.ts)
```



> pacf(resid.ts)



> spectrum(resid.ts,span=4)



There are spectral peaks at frequencies 0.220 and 0.348, both of which arise from calendar effects. These peaks are modest, though, in comparison to the prominent low frequency structure present in the residuals, attributable to remaining trend structure.

Next, let's first fit an ARIMA(0,1,1)(0,1,1)$_{12}$ model (the airline model) to the logged passenger data. Following this, we fit a regression to the ARIMA *predicted values*. The regression explanatory variables are a quadratic trend, and trigonometric terms to account for calendar components.

For the calculations following the fit of the airline ARIMA model, we will need to exclude the first 12 ARIMA predicted values. Note that in these example we are reversing some of the steps employed in the above analysis of the beer production data.

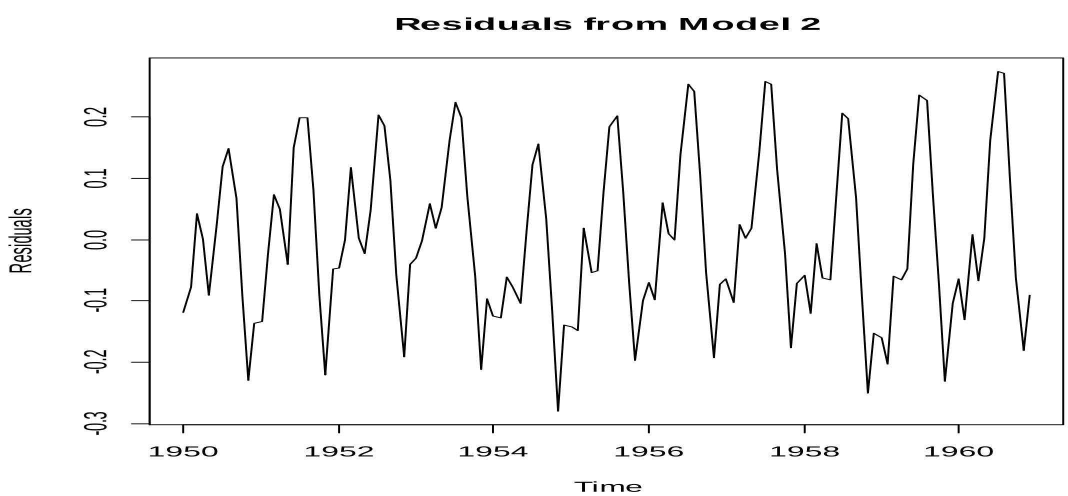
```
> lpass.ts<-ts(lpass)
> airlinemodel<-
arima(lpass.ts,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12))

> #calculate predicted values from ARIMA airline model
> sel<-1:12
> arimapred<-(lpass-resid(airlinemodel))[-sel]

> time<-1:132
> c220<-cos(0.44*pi*time);s220<-sin(0.44*pi*time)
> c348<-cos(0.696*pi*time);s348<-sin(0.696*pi*time)
> c432<-cos(0.864*pi*time);s432<-sin(0.864*pi*time)

> #fit the regression to the ARIMA predicted values
> model2<-
lm(arimapred~time+I(time^2)+c220+s220+c348+s348+c432+s432);summary(mode
l2)

Call:
lm(formula = arimapred ~ time + I(time^2) + c220 + s220 + c348 +
    s348 + c432 + s432)

Residuals:
     Min       1Q   Median       3Q      Max
-0.27940 -0.09108 -0.02203  0.08440  0.27426

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.859e+00  3.588e-02 135.442  < 2e-16 ***
time         1.375e-02  1.245e-03  11.045  < 2e-16 ***
I(time^2)   -2.857e-05  9.068e-06  -3.150  0.00205 **
c220         3.530e-03  1.666e-02   0.212  0.83252
s220         4.078e-04  1.665e-02   0.024  0.98050
c348        -8.936e-03  1.665e-02  -0.537  0.59239
s348         7.547e-04  1.666e-02   0.045  0.96394
c432         1.202e-03  1.668e-02   0.072  0.94267
s432         5.787e-03  1.663e-02   0.348  0.72845
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.1353 on 123 degrees of freedom
Multiple R-squared:  0.8949,    Adjusted R-squared:  0.8881
F-statistic: 130.9 on 8 and 123 DF,  p-value: < 2.2e-16
```

The residuals from this regression will consist of the ARIMA airline model structure with trend and calendar components stripped out. Although the calendar terms are not significant in the model, we need to remove the calendar structure, and the above procedure does so. (The calendar structure is weak relative to the seasonal structure and is therefore not detected by the *t*-tests. We want to remove the calendar structure so it doesn't contaminate the estimation of the seasonal structure developed next.)

The next plot shows the residuals from this regression. These residuals are estimates of the seasonal structure of the airline passenger data. One can see that the seasonal structure is strong, and that it is clearly fluctuating with time. As noted, these residuals are devoid of trend and calendar structure.

```
> model2resid.ts<-ts(resid(model2),start=c(1950,1),freq=12)
> plot(model2resid.ts,xlab="Time",ylab="Residuals",main="Residuals from
Model 2")
```



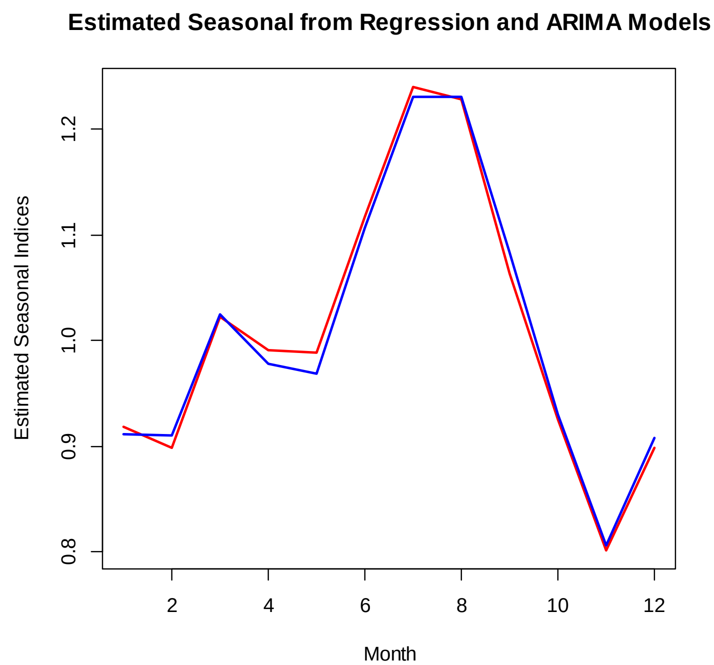
The plot provides clear visual evidence of dynamic seasonal structure.

Next we form one set of seasonal indices from these residuals. These are static estimates of the seasonal structure.

```
> monmeans<-tapply(model2resid.ts,month[-sel],mean)
> seasarima<-exp(monmeans-mean(monmeans))
> seasarima

        1         2         3         4         5         6         7         8
0.9118573 0.9104608 1.0254300 0.9783093 0.9687778 1.1068674 1.2307393 1.2307892
        9        10        11        12
1.0836615 0.9306037 0.8067760 0.9085497
```

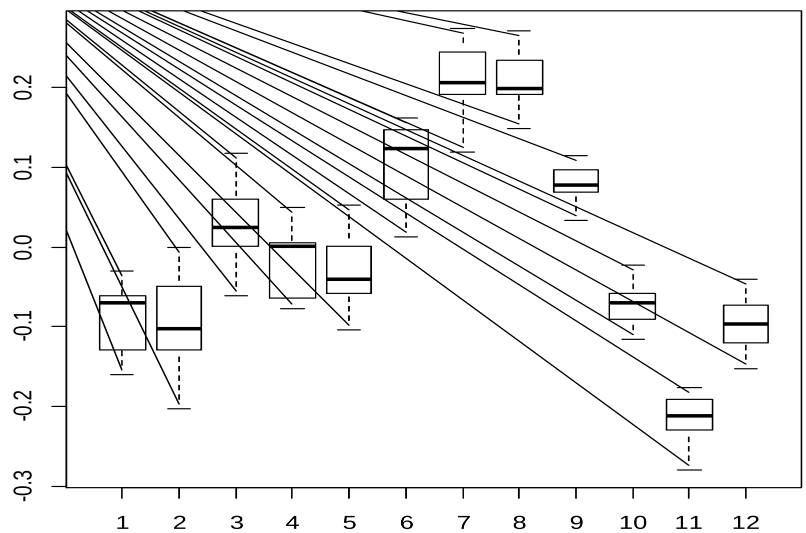
The table and plot following compare the estimated seasonal indices obtained from the regression and ARIMA models.

```
> cbind(1:12,seasreg,seasarima)
Month Regression     ARIMA

  1    0.9185000   0.9118573
  2    0.8982714   0.9104608
  3    1.0230335   1.0254300
  4    0.9914116   0.9783093
  5    0.9889771   0.9687778
  6    1.1174163   1.1068674
  7    1.2398141   1.2307393
  8    1.2283964   1.2307892
  9    1.0630700   1.0836615
 10    0.9260117   0.9306037
 11    0.8021864   0.8067760
 12    0.8991239   0.9085497

> plot(ts(seasreg),xlab="Month",ylab="Estimated Seasonal
Indices",main="Estimated Seasonal from Regression and ARIMA
Models",lty=1,lwd=2,col="red")
> lines(ts(seasarima),lty=1,lwd=2,col="blue")
```



Next, we show the side-by-side boxplots of the ARIMA predictions stripped of trend and calendar components, boxplots for all the months.

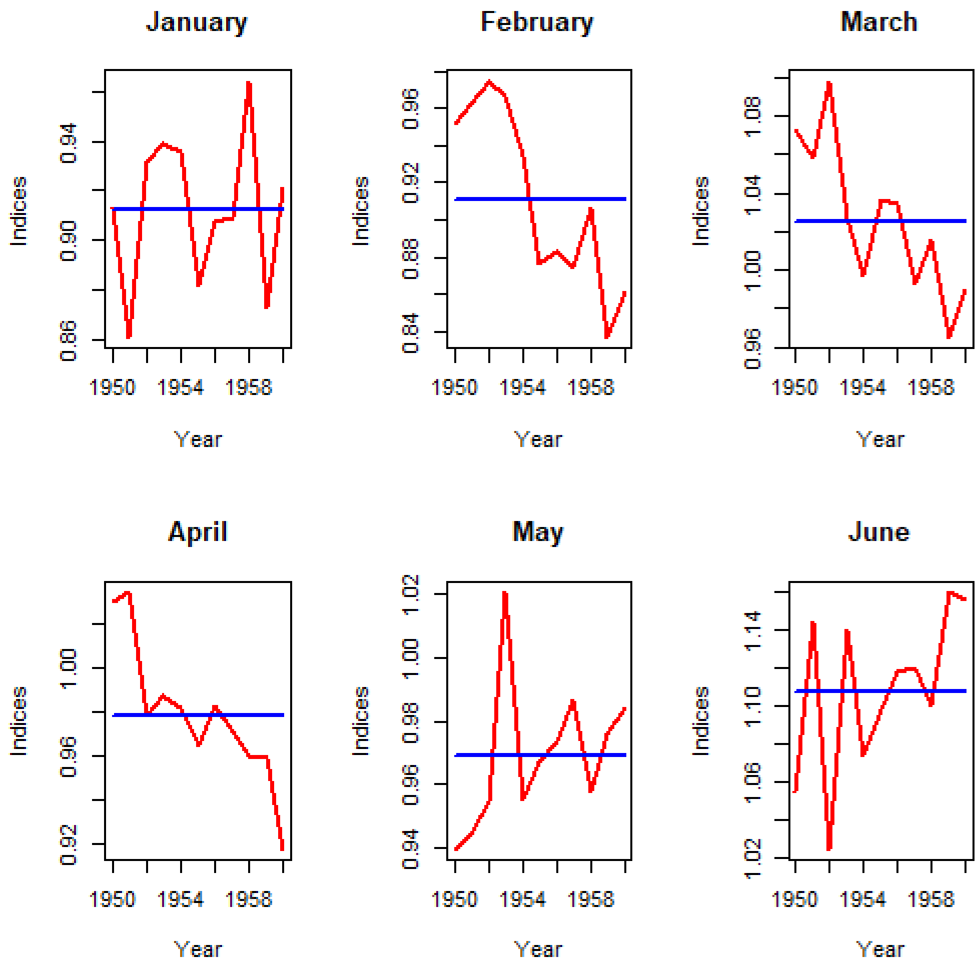
```
> boxplot(model2resid.ts~cycle(model2resid.ts))
```

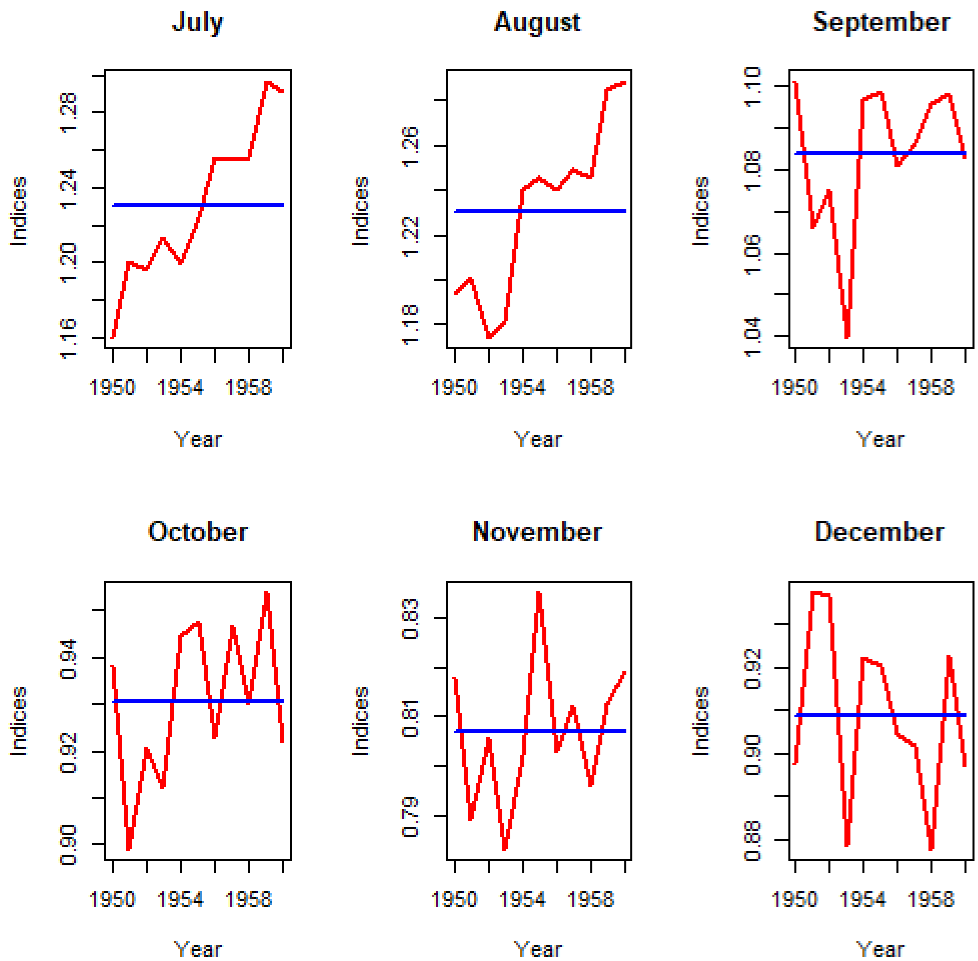


As with the beer data, the variation of seasonal indices across the years is evident in the boxplots.

Finally, calculation of the dynamic seasonal index estimates follows.

```
> y<-model2resid.ts
> seasm<-matrix(rep(0,132),ncol=11)
> j<--11
> for(i in 1:11){
+ j<-j+12;j1<-j+11
+ seasm[,i]<-exp(y[j:j1]-mean(y[j:j1]))
+ }
> year<-seq(1950,1960)
> seasarimam<-matrix(rep(seasarima,11),ncol=11)

> name<-
c("January","February","March","April","May","June","July","August","Se
ptember","October","November","December")
> par(mfrow=c(2,3))
> for(i in 1:6){
+
plot(year,seasm[i,],xlab="Year",ylab="Indices",main=name[i],type="l",lw
d=2,col="red")
+ lines(year,seasarimam[i,],lty=1,lwd=2,col="blue")
+ }
```


January
February
March
April
May
June
Indices
Year
1950
1954
1958
0.86
0.90
0.94
0.84
0.88
0.92
0.96
1.00
1.04
1.08
0.98
1.02
1.06
1.10
1.14



Each red curve shows the dynamic estimation of the seasonal structure, giving evolution of the estimated seasonal index from 1950 to 1960 for a given month, as implied by the ARIMA model. And for each month the blue curve is the static estimate of the seasonal index from the ARIMA model. The change in seasonal index behavior over time is most noticeable for February, March, and April, which show a decline over time, and July and August, which exhibit an increase over time.

Summary and additional remarks

1. The U.S. beer and airline passenger time series addressed in the 5 April notes are revisited. The purpose is to obtain seasonal index estimates from ARIMA models fit to these series.

2. The static seasonal index estimates derived from the ARIMA models are shown to be similar to those obtained from regression methodology for both of the time series.

3. A seasonal ARIMA model permits estimation of a dynamic seasonal pattern. That is, a set of seasonal index estimates is derived for each cycle (each year when the data are monthly), and plots allow one to see how the seasonal pattern drifts over time.

4. In these notes, two different methods have been employed to obtain seasonal index estimation from ARIMA seasonal model estimation.

(i) For the U.S. beer data a regression with calendar trigonometric pairs was initially fit to the time series. Next, a seasonal ARIMA model was fit to the residuals from this regression. Then trend structure was removed via regression from the predicted values from this seasonal ARIMA fit. Finally, centered monthly averages of the modified predicted values were calculated, giving the seasonal index estimates.

(ii) For the airline data, a seasonal ARIMA model was initially fit to the log of the time series. Next, a regression with time trend and calendar trigonometric pairs was fit to the predicted values from this ARIMA model, and the regression residuals were formed. Finally, centered monthly averages of the regression residuals were calculated, then exponentiated, giving the seasonal index estimates.

Thus, the two methods reverse some of the operations of regression and seasonal ARIMA fitting.