Who knew Australia could brew? Modeling and Forecasting the Monthly Beer Production in Australia

A Time Series Analysis By: Nicolas Lanker and Jorge Mejia

April 16, 2015

Abstract

In this time series analysis, we analyze the monthly beer production in Australia between 1970 and 1990. The original series was determined to be non-stationary and required both a first and seasonal difference to make it stationary. After stationarity was reached, the SARIMA parameter's were found by using the auto.arima function in the r package "forecast". By optimizing this function according to our differenced model, the function was able to filter through many different candidate models and selected the one with the lowest AIC and BIC. Our best candidate model was found to be a $(2,1,2),(2,1,0)_{12}$ SARIMA model with all parameter coefficients being significant. The model diagnostics weren't perfect as the model failed the Ljung-Box test which tested whether the model was appropriate. Nevertheless both the normality and independence of the standardized residuals were upheld according to the Shapiro-Wilks and the Runs-test respectively. Since no other model we tested could pass the Ljung-Box test we stuck with the best model so far which was the SARIMA $(2,1,2),(2,1,0)_{12}$. The Forecasts using this model were very accurate the first year and the actual data also stayed within the 95% confidence interval for all four years. This model and it's forecasts might be of use to Australian brewers and economists.

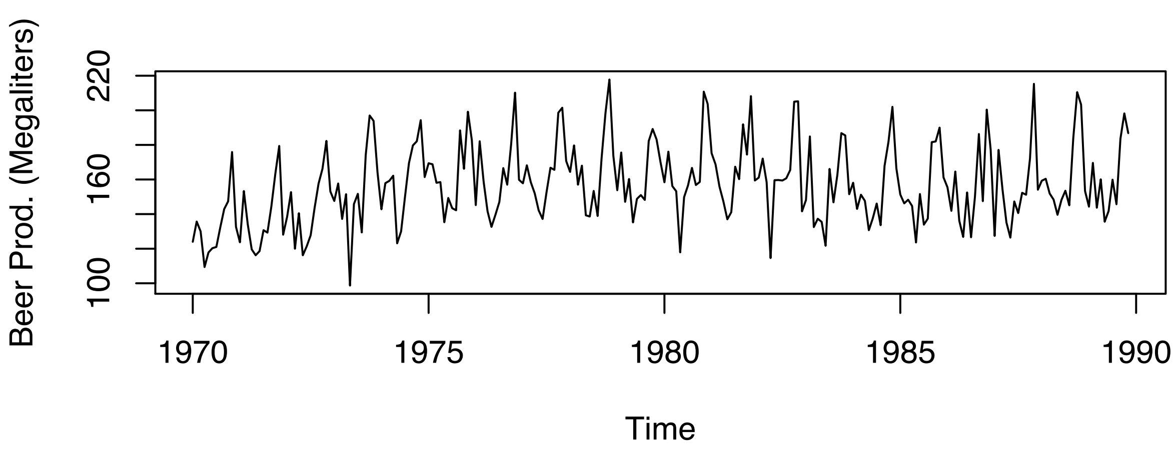
Introduction

Is it surprising that Australians are heavier drinkers than the English? Not in the slightest considering many of Australias first settlers not only happened to be sailors but also happened to be English. Double the trouble if you ask me! It's also not surprising considering Captain Cook used to prefer beer to water on his voyages through the Pacific because it kept *better*[2]. Nevertheless, as a Yankee, I never really paid much attention to Australian beer considering I've only had Foster's. Yet despite my ignorance, Australia has one of the highest per capita beer consumption in the world, and apparently they brew some great beer *too*[2]. It's not surprising then that they export their subpar Foster's to us Yanks.

For Aussies, the Lager is their beer of choice, especially in the Summer *months*[3]. We were initially confused when we first went over a time series of the monthly production of beer in Australia; Why were the Aussies producing so much more beer in October and November and then cutting their production by more than 25% percent as soon December rolled around? It's then that we realized that Australia's basically on the opposite side of the planet and as such have their summer months from December to February. Thus a spike in Beer production right before Summer would make sense. As we digged deeper into the analysis of this series, we also discovered something else interesting. Namely that the one time that December had production levels close to that of November and October, was coincidentally the same month that the song "Down Under" became the #1 song in Australia ($Dec, 1981$)[4]. Perhaps Australians, already on an early summer drinking spree because the songs good vibes, needed to up production in December to make sure everyone had their cellars stocked.

After seeing such an odd coincidence in the data set we were hooked. Not because modeling beer production was particularly interesting, but because maybe if we modeled the series correctly, we could then discover the successor to "Down Under". To be clear, we were never able to discover such a song, since the nature of math that governs time series models tends to make every December just as similar as the ones before it. We thus concluded after analyzing this data set that the next big discovery in Time Series Analysis should make it possible to discover when such one-hit wonders could materialize. Although our hopes were dashed somewhere in the early stages of analysis, we continued to analyze the data in hopes that someday, somewhere, a random brewery or some bored economist picked up our model, and ended up putting it to better use than we did.

Thus begins our analysis of this time series of Australian beer production. We will begin by attempting to find the parameters needed to model our time series, and follow that up with estimating coefficients for those parameters. After finding the model that we think fits best, we will defend it with some diagnostic tests as well as present a forecast of the next four years. The data we will be using comes from the time series database by Rob J Hyndman. We truncated the data to go from January 1st 1970 to December of 1989, and witheld the last four years of data for a comparison with our forecast. The response variable is the megaliters of beer produced that month. To put that into perspective, one megaliter is 4 million cans of 8 oz. beer. The *data*[4] below is the original time series plot of the monthly beer production in Australia between 1970 and 1990.

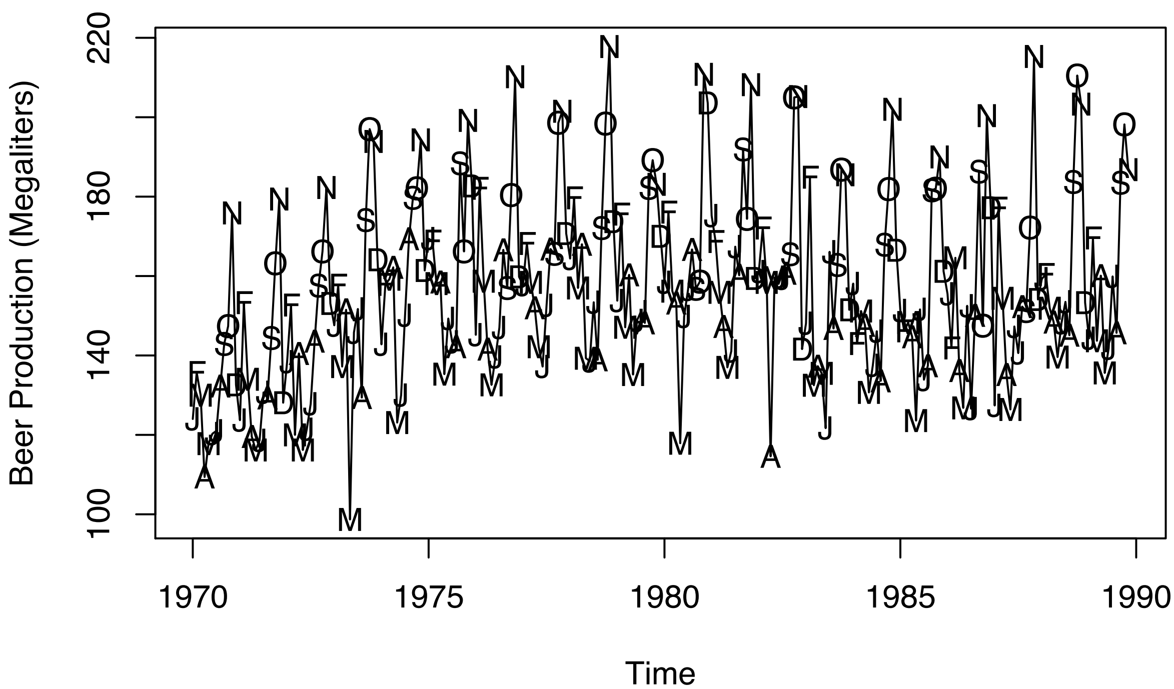


Model Specification

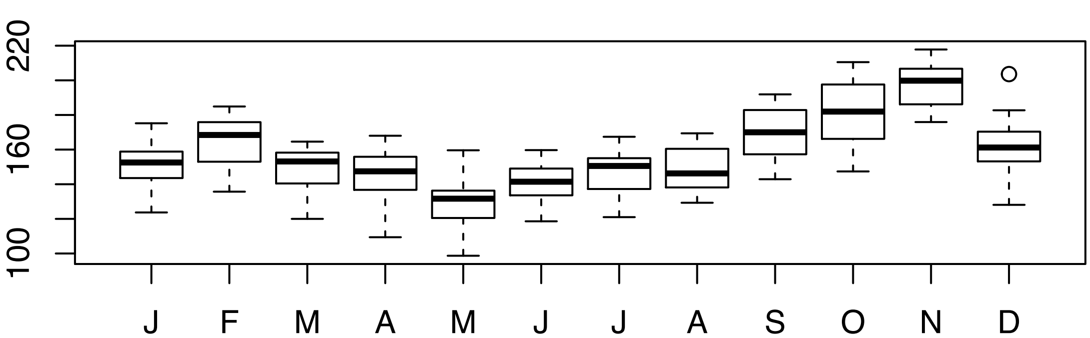
In this section we will attempt to find a time series model that sufficiently characterizes the production of beer Australia, and use that model to forecast the next few years of beer production in Australia. We begin our analysis by observing the production of beer graphically.

As was discussed in the introduction, the production of beer seems to stay about the same year over year, but the monthly production of beer varies fairly predictibly with October and November being very popular months for production, followed by a very significant drop off in December. This monthly trend can be seen in the in both Time Series plot of the data and the monthly boxplots of the data shown below.

Monthly Beer Production in Australia between 1970 and 1990

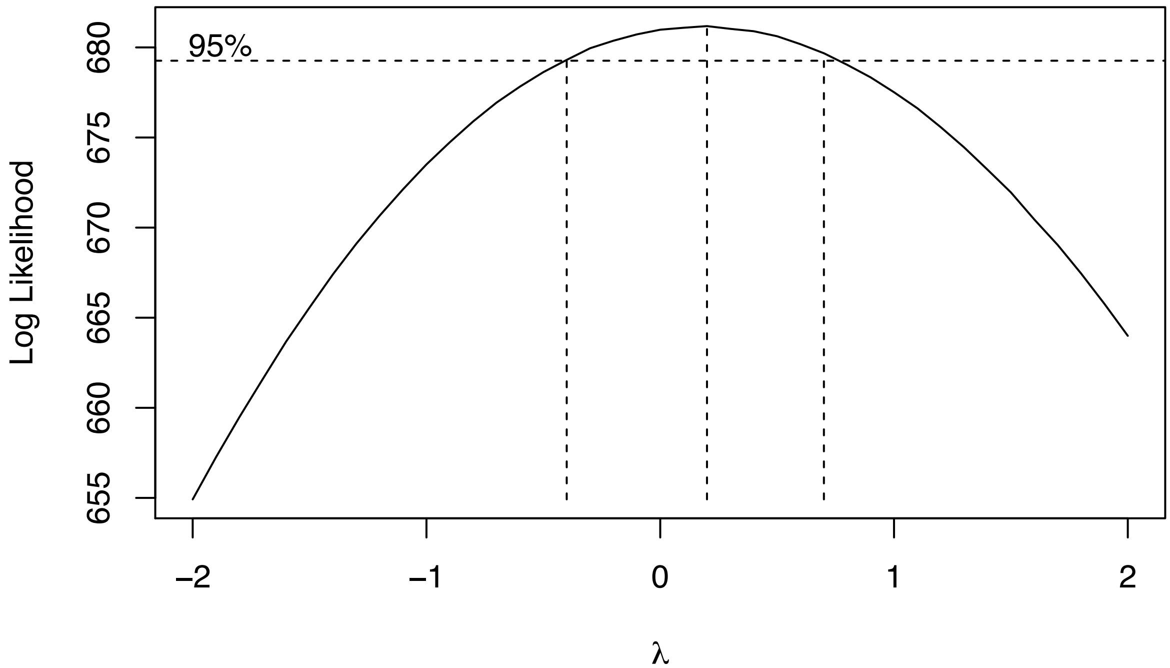


Monthly boxplots of Beer Production in Australia



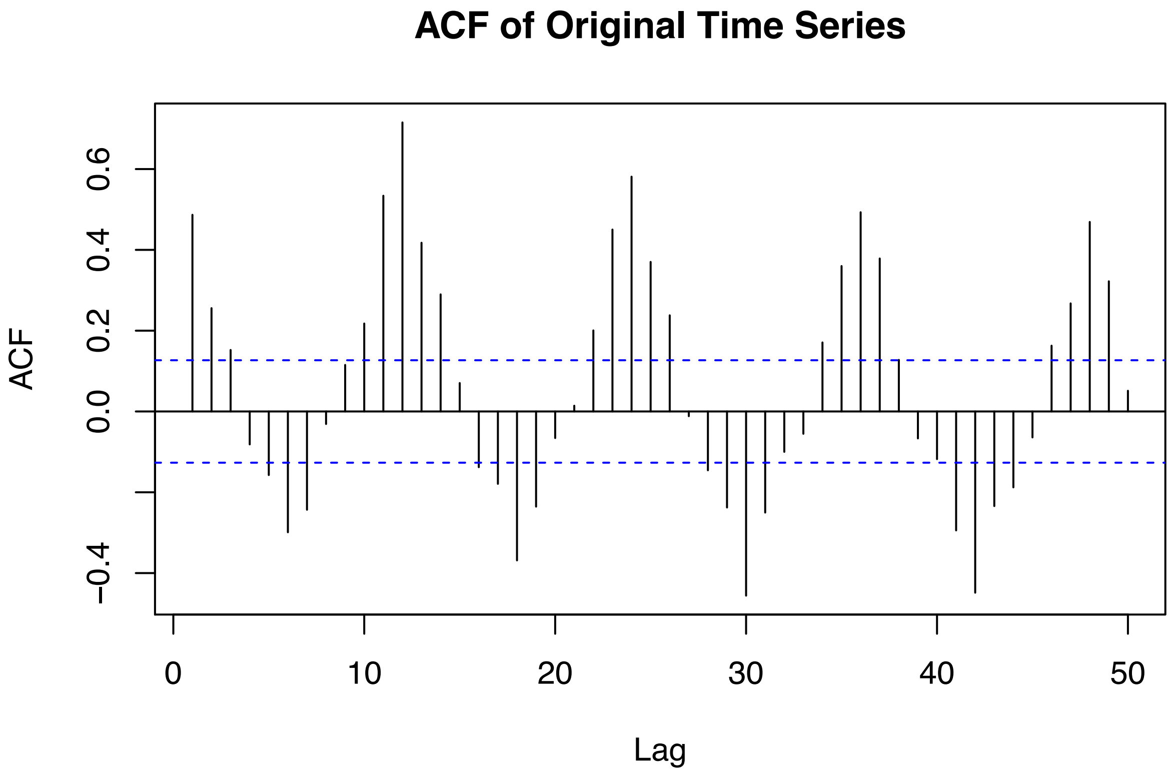
Adjusting for heterscedasticity using BoxCox

To test whether or not the series exhibits heteroscedasticity we need to look at whether or not BoxCox wants us to transform the data initially to reduce heteroscedasticity. Below is the BoxCox lambda estimation for a power transformation.



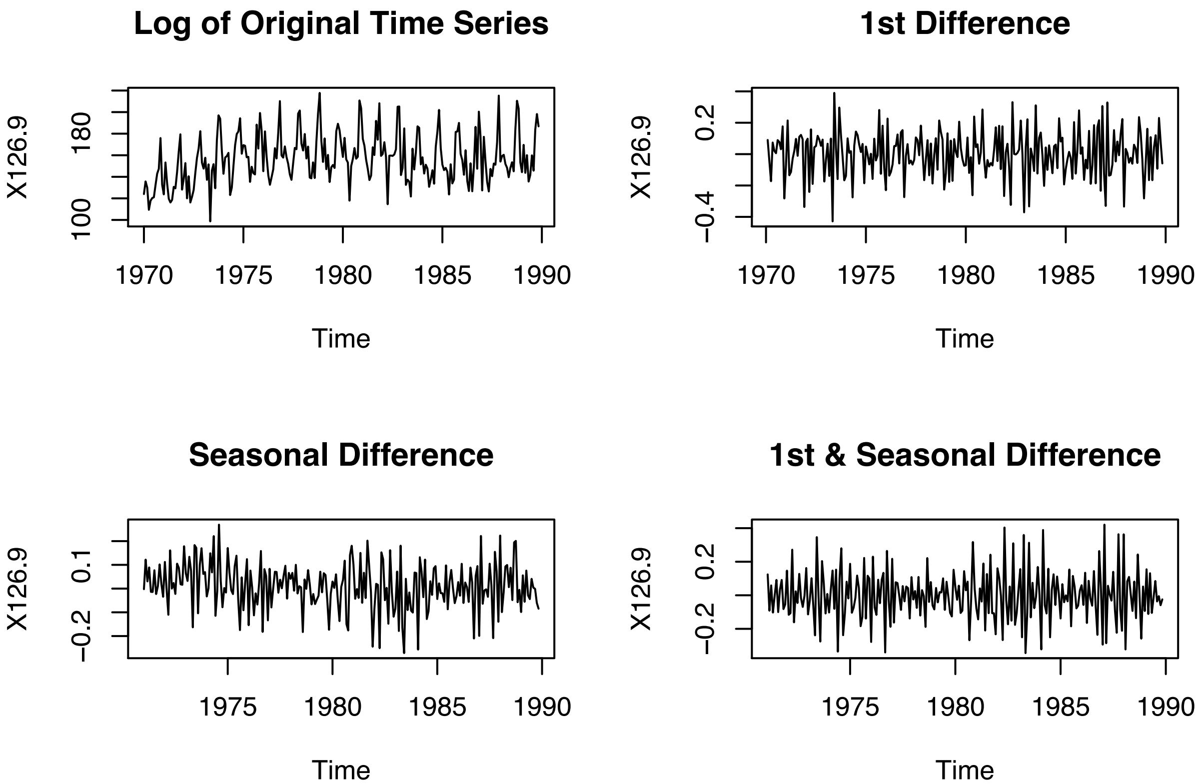
The BoxCox results show that $\lambda = 0$ is within the 95th confidence intervals for the lambda estimation while $\lambda = 1$ is not. Since the confidence intervals are quite wide and are near $\lambda = 1$, a transformation might not be necessary, however seeing as they are outside this interval and a transformation will only help with heteroscedasticity and not hurt it, we decided to use log transformation of our data since λ is centered close to 1.

A plot of the log transformed values can be seen in the appendix.



As was discussed before the data shows a definate seasonal trend, and as seen above this seasonal trend is also apparent in the autocorrelation function (ACF) for the series. The ACF has a sinusoidal pattern to it that peaks at lags 12, 24, 36 and onwards which is indicitave of a yearly or 12 month seasonal trend. Since the peaks decay very slowly, this indicates that the series is probably seasonally nonstationary and so taking the 12th difference, or seasonal difference, might help make this series more stationary. At this point however, it's unclear whether a first difference, a seasonal difference, or a combined seasonal and first difference is necessary. Thus the next step is to determine which differences will make this series stationary.

Plots of the 1st, Seasonal, and Combined Differences



Here we can see the different plots of the log of the Original Time Series, as well as the log of the First, Seasonal and Combined Differences. The original time series is definately nonstationary, whereas it's still hard to know whether any of the differenced plots exhibit nonstationarity.

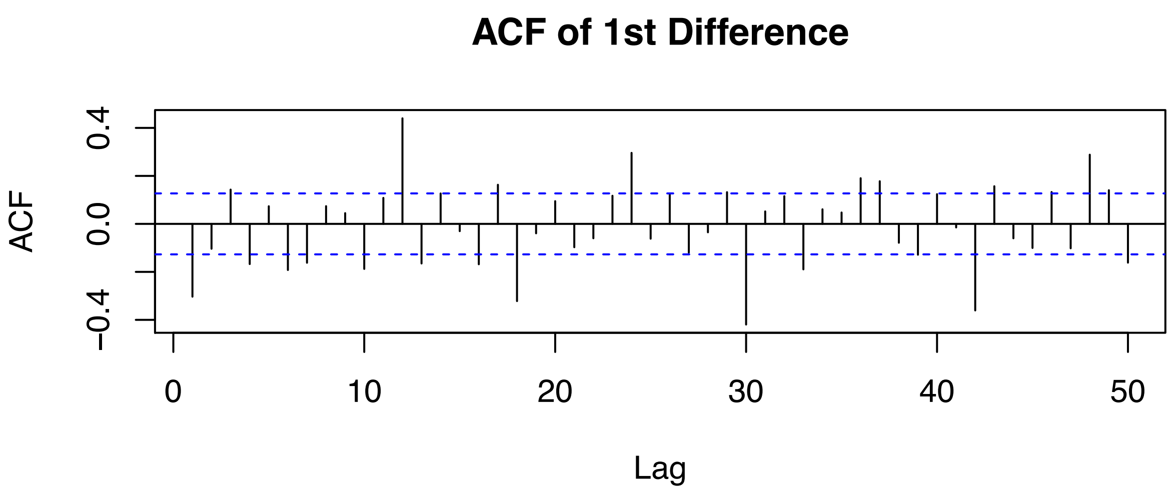
The first difference plot definately still has a seasonal component to it but could still be stationary.

The seasonally differenced model seems to have little nonstationarity, since the points don't hang together too much. However the seasonal differenced plot does not have the same mean level over time.

The combined seasonal and first difference plot does seem to exhibit a stationary process. Since it's not certain they're all stationary we will also take a look at the ACF's and PACF's of each as well as apply a Dickey-Fuller Test to each one.

Testing For Stationarity of the First, Seasonal, and Combined Differences

1st Difference




```
##
## Call:
## ar(x = beer.diff)
##
## Coefficients:
##       1        2        3        4        5        6        7        8
## -0.8338  -0.8003  -0.6415  -0.6079  -0.4080  -0.5502  -0.6139  -0.5140
##       9       10       11       12       13       14       15
## -0.4893  -0.5372  -0.2616   0.2404   0.1996   0.2792   0.1743
##
## Order selected 15  sigma^2 estimated as  0.008262
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 15
##   STATISTIC:
##     Dickey-Fuller: -5.031
##   P VALUE:
##     0.01
##
## Description:
##  Fri May  1 16:24:54 2015 by user:
```

The first difference seems to be stationary since the Dickey Fuller test (with k=15) rejects the null hypothesis of nonstationarity. It does have signficant lags at lag k = 12, 18, 24, 30, 36, 42, and 48 indicating, like before, that this series has a seasonal component to it. The PACF for this series is in the appendix.

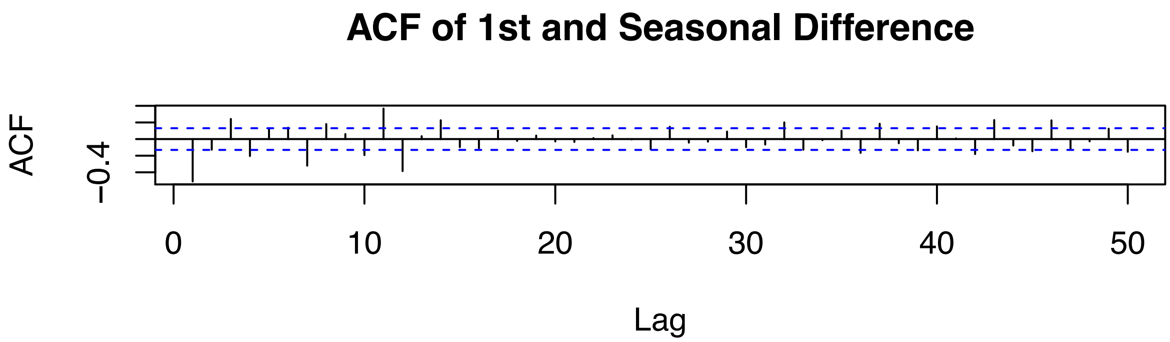
Seasonal Difference

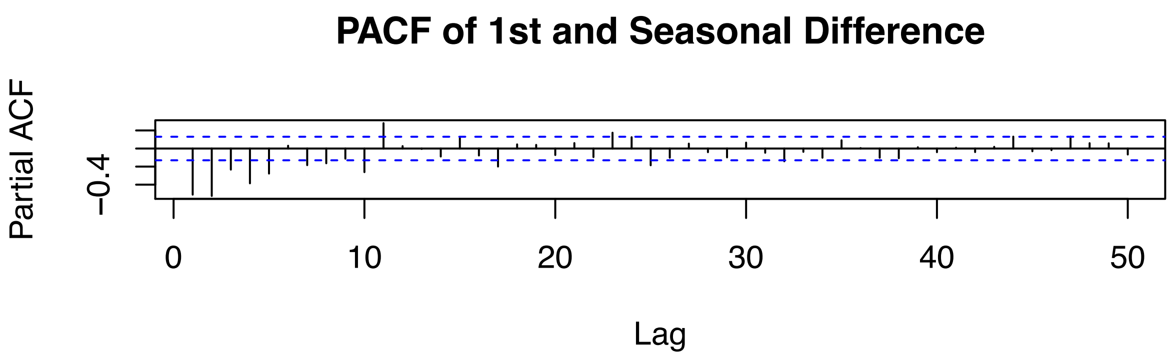
```
##
## Call:
## ar(x = beer.se)
##
## Coefficients:
##       1        2        3        4        5        6        7        8
## -0.1582  -0.1424   0.1221   0.0423   0.1966   0.2099  -0.0421   0.1463
##       9       10       11       12       13       14       15       16
##  0.1441   0.0612   0.1842  -0.3471  -0.0860  -0.0594   0.0280  -0.0928
##      17       18
##  0.0916   0.2006
##
## Order selected 18  sigma^2 estimated as  0.006939
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 18
##   STATISTIC:
##     Dickey-Fuller: -2.1521
##   P VALUE:
##     0.03206
##
## Description:
##  Fri May  1 16:24:55 2015 by user:
```

Since the seasonally differenced series failed to reject the null hypothesis of nonstationarity in the Dickey-Fuller test, we can disregard this series. The ACF and PACF for this seasonal difference are in the appendix.

First and Seasonal Difference






```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 17
##   STATISTIC:
##     Dickey-Fuller: -5.1142
##   P VALUE:
##     0.01
##
## Description:
##  Fri May  1 16:24:55 2015 by user:
```

Here we can see that the combined first difference and seasonal difference is likely stationary since it rejects the null hypothesis of nonstationarity in the Dickey-Fuller test. Furthermore, unlike ACF plot of the first difference only, there's really only one peak at lag k = 12, and not out to lag k = 60 or even further. Thus since teh ACF plot looks better in the first and seasonal difference I will continue with using this combined model.

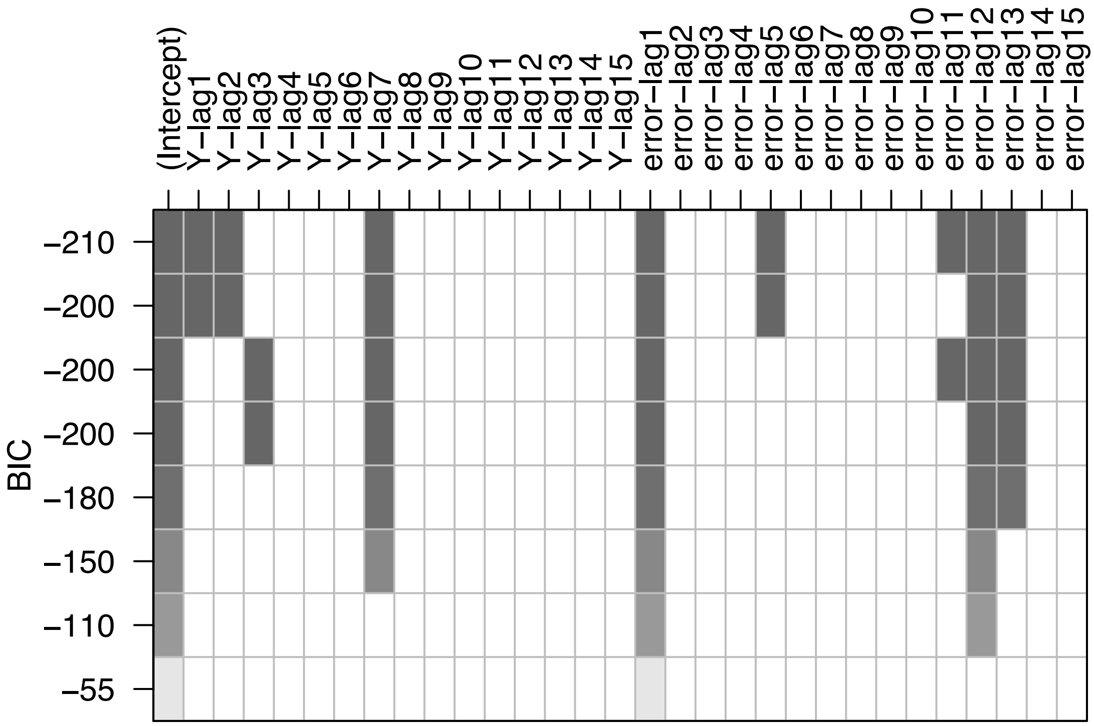
We now have d=1 (First Difference) and D=1 (Seasonal Difference) in a multiplicative SARIMA model Our goal now is to find the four other parameters in our model, the MA(p) and AR(q) parameters of our nonseasonal component and the SMA(P) and SAR(Q) of our seasonal componenet.

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o x x o x x x o x x  x  o  x
## 1 x x x o o o x x o o x  x  o  x
## 2 x x x o x o x o o o o  x  x  o
## 3 x x o x x o x o x o o  x  x  x
## 4 x x x x x o o o o o x  x  x  o
## 5 o x x x x o o o o o o  x  x  o
## 6 o x x x x o x o o o o  x  x  o
## 7 x o x o x o x o o o o  x  x  o
```

Looking at the ACF, PACF, or the EACF it's very hard to tell which order each parameter should have. The EACF is useless at this point. However the ACF does show strong autocorrelation at lag k = 1 indicating an MA(1) process (p=1). Furthermore the peak at lag k = 12 indicates an SMA(1) process (P=1), and since there is some "noise" around this peak, this supports an MA(1) process as well. The PACF really can't tell us anything interesting except that the slow decay after lag k = 1 hints at an MA(1) process. Thus one model we could analyze is the $(1,1,0),(1,1,0)_{12}$ process. However, this model may not be the only candidate and so we go onto other methods of finding param

Arma Subsets using BIC

Another way of finding a good fitting model would be to observe the Arma Subsets that returns SARIMA parameters according to the lowest BIC value. Below is the ARMA subset for the First and Seasonally Differenced Process.



As we can see above it's again going to be hard to analyze this subset since the best model it suggests is one with Yt-1, Yt-2, Yt-7, et-1, et-5, et-11, et-12, et-13. This model is far too complex and as is evident the next several are also too complex to use, especially considering they all would require taking a awkward lag k=7 autoregressive parameter which doesn't fit into our current seasonal difference operator which is s = 12. Although the BIC gave us models that were too complex, it also suggests a SARIMA $(1, 1, 0), (1, 1, 0)_{12}$ at the very bottom of the table which we previously estimated from the ACF. Since the ARMA subsets were pretty unhelpful we ended up using the auto.arima function to estimate our parameters

Using the "auto.arima()" function to estimate ARIMA and SARIMA parameters

Since the only model we currently have by using the previous parameter estimation methods is the SARIMA $(1,1,0),(1,1,0)_{12}$ model, we decided to use the auto.arima function to help us with estimating the parameters for this model. Since we already know our parameters d=1 and D=1 we added some constraints to the auto.arima function that would help it's estimation. These constraints are that the model must contain a seasonal and first difference, and for the first function it must maximize AIC and for the second it must maximize BIC. The input data is the log transformation of the original series. The function would give the same output had we inputted the first and seasonal differenced series and had removed the d=1 and D=1 parameters. I thought it was necessary to see the code behind this so here it is:

```
auto.arima(beer.log, d=1, D=1, seasonal=TRUE, approximation=FALSE, ic="aic")
```

```
## Series: beer.log
## ARIMA(2,1,2)(2,1,0)[12]
##
## Coefficients:
##           ar1      ar2      ma1      ma2     sar1     sar2
##       -0.7856  -0.4101  -0.4093  -0.4103  -0.5219  -0.2923
## s.e.   0.1254   0.0653   0.1277   0.1185   0.0656   0.0657
##
## sigma^2 estimated as 0.006832:  log likelihood=239.24
## AIC=-464.49   AICc=-463.97   BIC=-440.54
```

As we can see above, according to function the best model to choose is the one with the lowest AIC value, which would be a $(2,1,2),(2,1,0)_{12}$ SARIMA model. Since this model has the lowest AIC value and since the $(1,1,0),(1,1,0)_{12}$ model ends up being worse at forecasting and doesn't do better with model diagnostics, I will only continue with the $(2,1,2),(2,1,0)_{12}$ model. These $(1,1,0),(1,1,0)_{12}$ forecasts and select model diagnostics can be found in the appendix as well.

I also ran another auto.arima function with the only changed parameter being that it chose the one with the lowest BIC value and this model was also the lowest in that category, tied with it's close runner-up the SARIMA $(2,1,1),(2,1,0)_{12}$. This "BIC" auto.arima function can also be found in the appendix.

Fitting and Diagnostics

Model Estimation

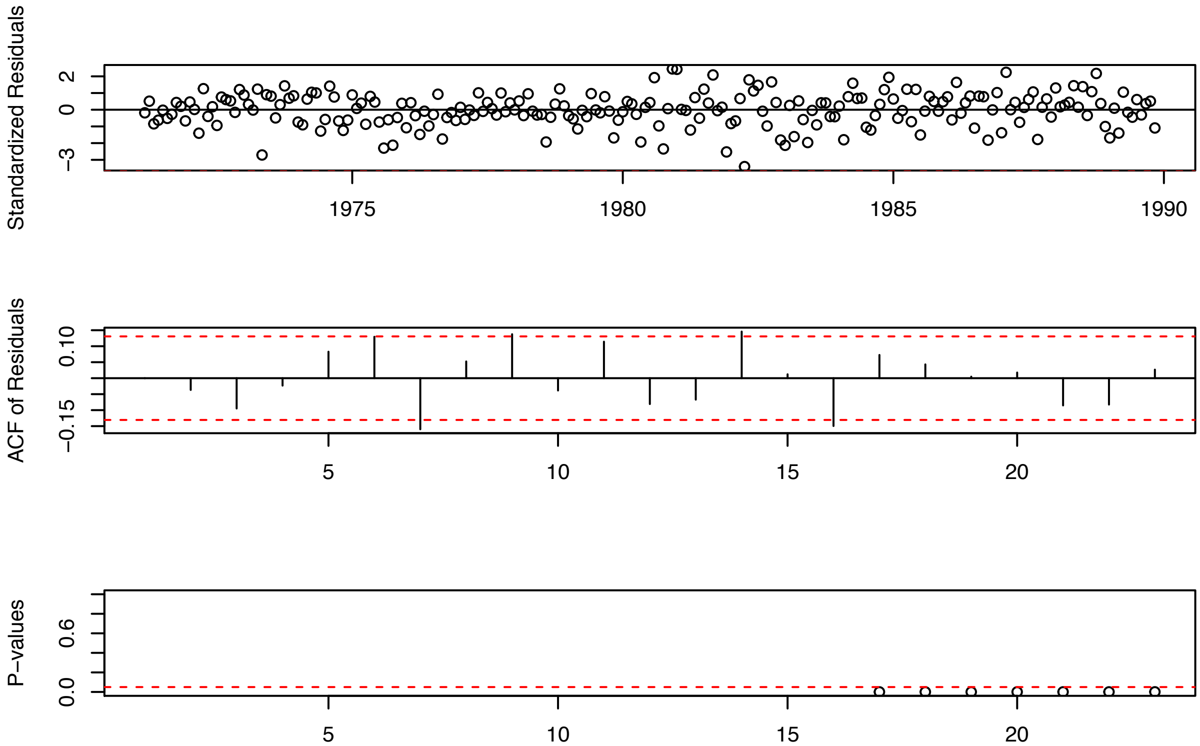
The next step is to estimate the parameter coefficients in my model. The auto.arima function already does this for me but I will still run the arima function below to list the coefficients.

```
##
## Call:
## arima(x = beer.log, order = c(2, 1, 2), seasonal = list(order = c(2, 1, 0),
##     period = 12))
##
## Coefficients:
##           ar1      ar2      ma1      ma2     sar1     sar2
##       -0.7856  -0.4101  -0.4093  -0.4103  -0.5219  -0.2923
## s.e.   0.1254   0.0653   0.1277   0.1185   0.0656   0.0657
##
## sigma^2 estimated as 0.006832:  log likelihood = 239.24,  aic = -466.49
```

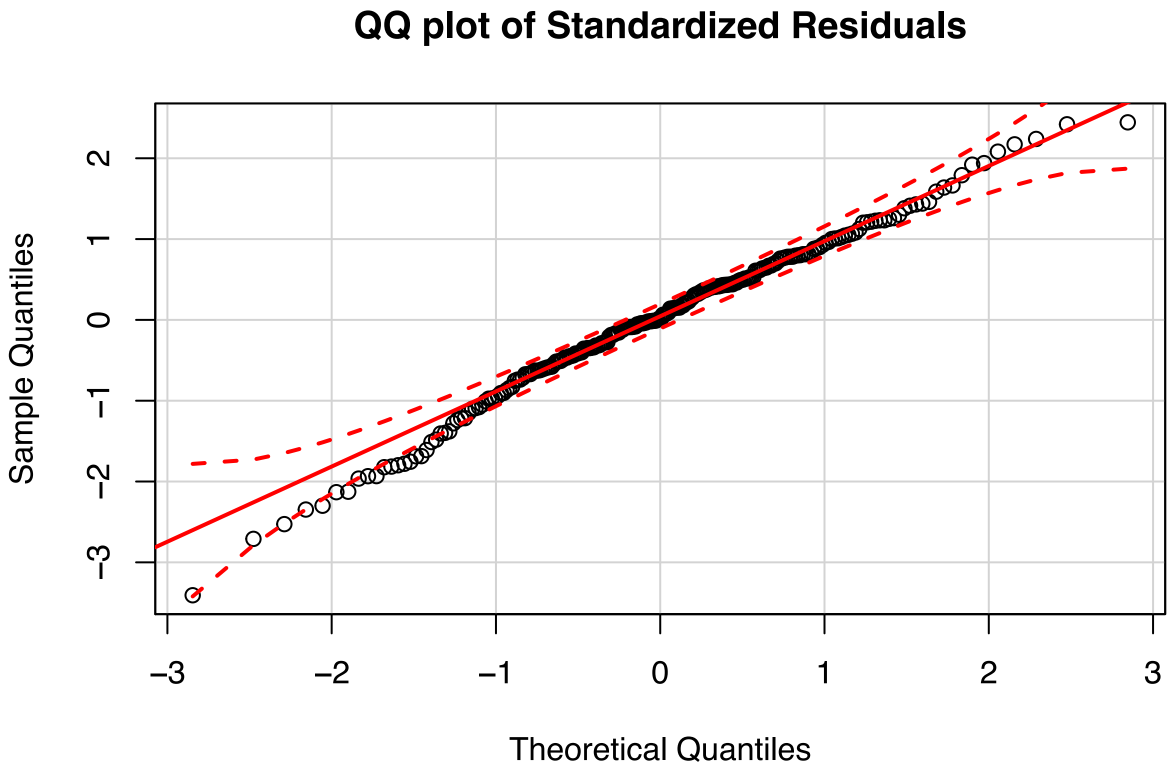
As is seen above all of the parameter coefficients are significant since they all are at least three standard deviations away from zero. We now go onto the model diagnostics.

Model Diagnostics

At this point we have chosen a model and chosen it's parameter coefficients. After fitting this model, the next step is to look at the model diagnostics. Now, before I go further, having tested numerous previous models and checking their model diagnostics, there were only slight differences here and there but all of them rejected the Ljung-Box null hypothesis that the selected SARIMA model is appropriate at all lags k. Unfortunately there is only so much that can be done to model a time series and so trying to find a model that passed the Ljung-Box test was futile. The tsdiag function below displays some of the more important diagnostic plots.



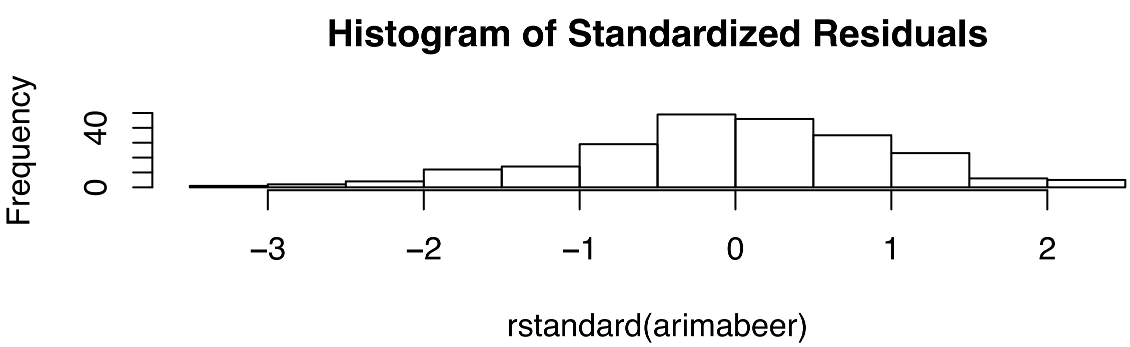
In the first plot there is some minor trumpeting of the residuals as time goes on which indicates some non-constant variance. The ACF has very few and very minor signficant autocorrelations which is good. Again the results from the different lags of the LJung-Box test shows that our model is unsatisfactory, but we will disregard this for the reasons stated before. A larger version of the acf can be seen in the appendix.



The qqPlot above shows some flaring at the ends especially on the lower half. It would seem then that our model doesn't take into account some of the larger drops in beer production that occur throughout the time series. Thus it would seem that the normality of the residuals is not certain yet.

A further analysis of the normality of our residuals using both the histogram and the shapiro wilks test is represented below.

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(arimabeer)
## W = 0.9892, p-value = 0.08891
```



Although the histogram doesn't look perfectly normally distributed, it doesn't having any glaring inconsisincies that would reject normality. Furthermore the shapiro-wilks test narrowly fails to reject normality ($p = 0.089 > 0.05$), which is great news for our model. So, normality holds, now onto the runs test to see whether our standardized residuals are independent.

Runs Test

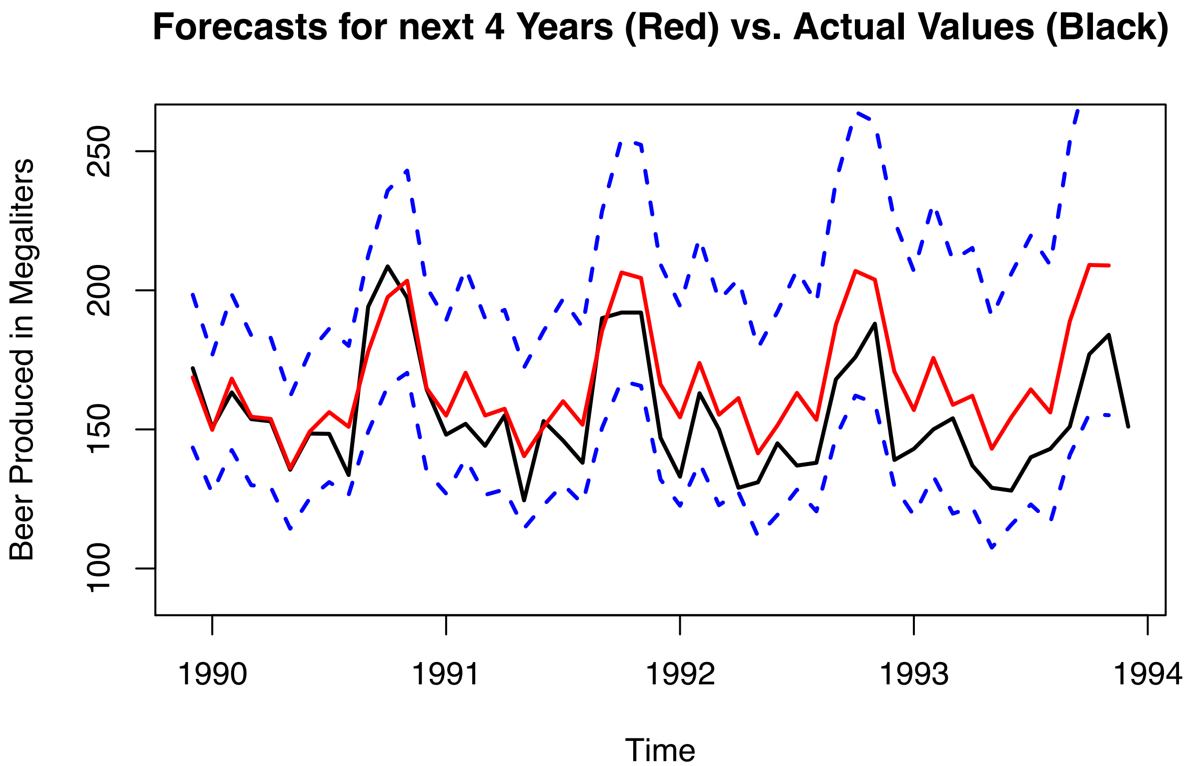
```
## $pvalue
## [1] 0.461
##
## $observed.runs
## [1] 120
##
## $expected.runs
## [1] 113.9646
##
## $n1
## [1] 111
##
## $n2
## [1] 115
##
## $k
## [1] 0
```

The runs test also fails to reject the independence of the residuals assumption since p = 0.461 >> 0.05), and so we can conclude that our residuals are very independent of eachother which is good news for our model.

Thus, although the model did not do so well on the Ljung-Box test it did pass for both the normality and independence of the standardized residuals and so we can feel comfortable moving onto the Forecasting.

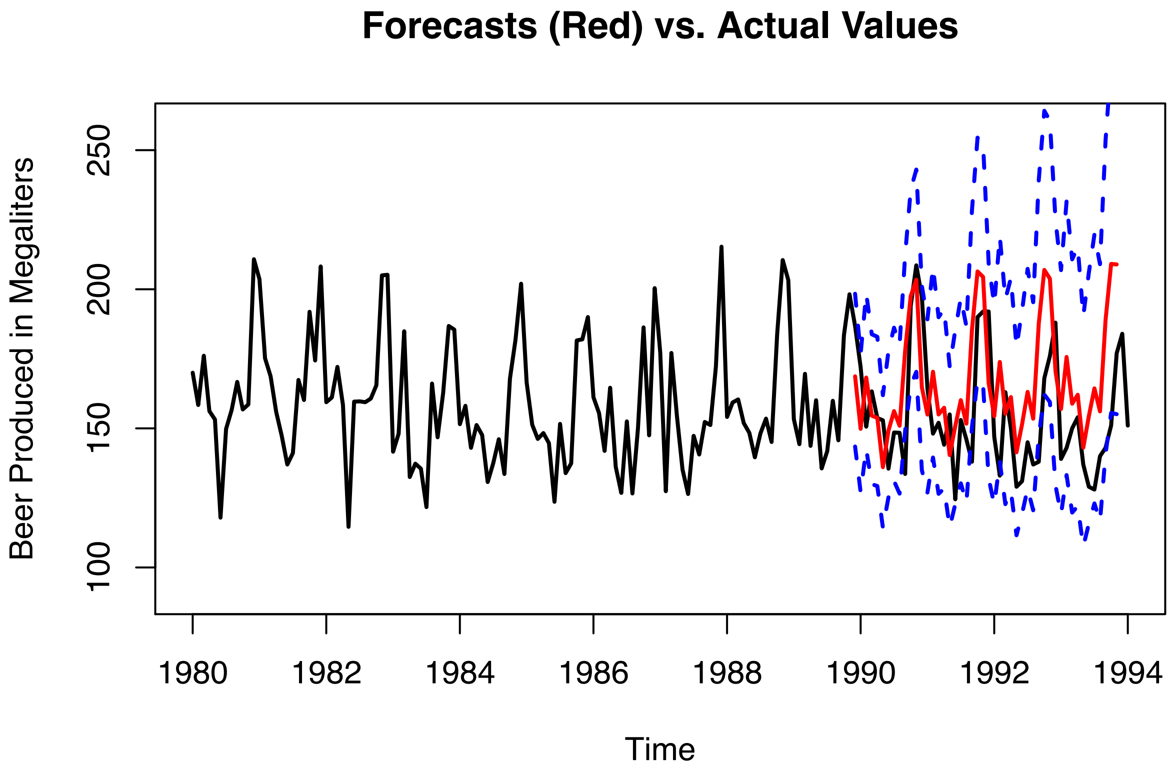
Forecasting

Fortunately R automates the forecasting process. Shown below is some of the r code used to forecast this series. We will fist look at only how the forecasted series compares to the actual values that were initially witheld.



Above we can see the actual values in black compared to the forecasted values in red. The forecast does a very good job forecasting the initial year and does a decent job with the second year as well. You will notice that as time goes on the forecast deviates from the actual values, yet it still seems to keep the shape of the process. The predicted values rise slightly up in their mean values likely because that slight upward slope was seen in the initial time series. Instead of going up however, the actual values tend to decrease in value as peaks have a noticeable decline in size. Furthermore the confidence intervals do a good job at both “hugging” the data (in that they aren’t too far from the actual values), while also containing all of the actual values. As is predicted, the confidence intervals also widen towards the end of the Time Series, but not obscenly so. Since the predicted values seem to be very near the actual values for at least the first year then I would say this model is succesful at accurately modeling beer production in australia for up to a year and can confidently (at least within 95% confidence) model the production for the next four years. A reevaluation of the model and a possible tweaking of its parameters and coefficients could be warranted every year to keep the forecasts very accurate.

It’s also necessary to see the forecast as a continuation of the previous years. In the plot below you will see the actual values from 1980 until 1994 with the forecast starting at 1990.



One issue that's apparent when seeing this overview of the forecasted values is that the confidence intervals tend to rise very sharply at the end of the 3rd and 4th year forecast. Otherwise the red forecast looks great compared to the previous values and the actual values it sits on.

Discussion

At first glance, this time series displays a fairly predictable yet jagged seasonal pattern. Fitting a model to what seemed like a simple seasonal pattern proved to be difficult as many steps were taken along the way that could have made the model go one way or another. It might not have been necessary to take the seasonal difference after the first difference but after plenty of experimenting it was clear that both a seasonal and first difference were necessary to make it both stationary and improve the diagnostics and forecasts. Since attempting to find a model by looking at the ACF's and PACF's as well as the ARMA subsets was unsatisfactory as well, we turned to fiddling with the auto.arima function that optimized our model for the best AIC and BIC values. The model diagnostics turned out good for the normality and independence of the residuals but the Ljung-Box test was rejected at all lags indicating that our model didn't perfectly fit the series after all. Nevertheless, the best model we came up with ended up giving a great forecast for the first year and a fairly accurate forecast that was within the 95% confidence interval for the next few years.

The main problems we encountered in this data analysis were the unhelpful ACF's and PACF's after the first and seasonal differencing, as well as the weird parameters that the BIC ARMA subsets suggested. It was thus hard to figure out any model parameters without using something that automated the search for the lowest AIC/BIC for us, namely auto.arima. Especially frustrating was seeing a large amount of significant autocorrelations at random lags, such as at lag $k = 7$, even after first and seasonal differencing. Even after fitting numerous models that seemed appropriate, they all failed the LJung-Box test. This is probably because ACF of the residuals had too many significant values (which is what the Ljung-Box uses in it's test).

The forecasts that our model produced were very useful for the first few months and could be useful for brewers who might use the forecasts to plan ahead production amounts. Economists or brewers from outside the country might also use these forecasts to make decisions about entering the Australian beer production market. This Australian beer production model might not be easy to generalize since the seasons are different in Australia than the rest of the world, however New Zealand might be interested in these forecasts in order to forecast their own beer production. If the months could be adjusted, then maybe the model could be applicable in other countries, yet economic and cultural differences between Australia and other countries will likely make this model useless in other countries. Thus, although this model and the forecasts might be useful to the brewing business, the time series model isn't very relevant to the average Joe(y)*.

*I'm really hoping this joke made sense

Acknowledgements

Jorge and Nic would like to thanks Professor Talithia Williams and the Harvey Mudd Math Departement for supplying us with knowledge of Time Series.

Bibliography

1. Source for Data: https://datamarket.com/data/set/22xr/monthly-beer-production-in-australia-megalitres -include-ale -and-stout-does-not-include-beverages-with-alcohol-percentage-less-than -115-ja-1956-aug-1995#!ds=22xr&display=line

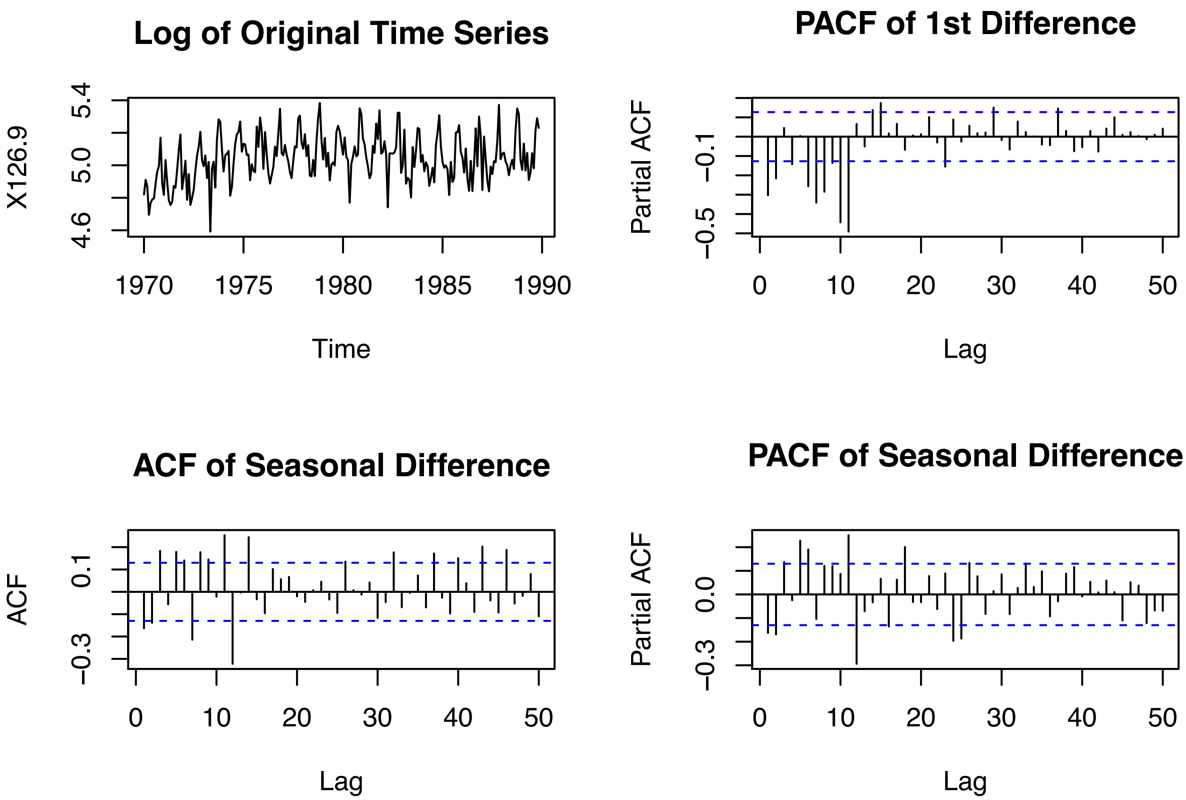
2. http://www.australianbeers.com/

3. http://en.wikipedia.org/wiki/Beer_in_Australia

4. http://en.wikipedia.org/wiki/Down_Under_%28song%29

Appendix

A list of plots and outputs mentioned in the text but not displayed:




```
##
## Call:
## ar(x = beer.se.diff)
##
## Coefficients:
##       1        2        3        4        5        6        7        8
## -1.0925  -1.1711  -0.9826  -0.8946  -0.6606  -0.4218  -0.4422  -0.2724
##       9       10       11       12       13       14       15       16
## -0.1075  -0.0340   0.1684  -0.1684  -0.2152  -0.2460  -0.2024  -0.2925
##      17
## -0.1998
##
## Order selected 17  sigma^2 estimated as  0.007412
```

```
##
##  ARIMA(2,1,2)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(0,1,0)[12]                    : -204.917
##  ARIMA(1,1,0)(1,1,0)[12]                    : -297.0952
##  ARIMA(0,1,1)(0,1,1)[12]                    : Inf
##  ARIMA(1,1,0)(0,1,0)[12]                    : -267.7035
##  ARIMA(1,1,0)(2,1,0)[12]                    : -306.9351
##  ARIMA(1,1,0)(2,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(2,1,0)[12]                    : -237.9077
##  ARIMA(2,1,0)(2,1,0)[12]                    : -393.4842
```

```
##  ARIMA(2,1,1)(2,1,0)[12]                    : -440.879
##  ARIMA(3,1,2)(2,1,0)[12]          : Inf *
##  ARIMA(2,1,1)(2,1,0)[12]                    : -440.879
##  ARIMA(2,1,1)(1,1,0)[12]                    : -429.4227
##  ARIMA(2,1,1)(2,1,1)[12]                    : Inf
##  ARIMA(1,1,1)(2,1,0)[12]                    : -428.3269
##  ARIMA(3,1,1)(2,1,0)[12]                    : -437.1133
##  ARIMA(2,1,2)(2,1,0)[12]                    : -440.5415
##
##  Best model: ARIMA(2,1,1)(2,1,0)[12]
```

```
## Series: beer.log
## ARIMA(2,1,1)(2,1,0)[12]
##
## Coefficients:
##          ar1      ar2      ma1     sar1     sar2
##      -0.3796  -0.3013  -0.8570  -0.4980  -0.2799
## s.e.   0.0702   0.0684   0.0388   0.0654   0.0660
##
## sigma^2 estimated as 0.007001:  log likelihood=236.7
## AIC=-461.4   AICc=-461.02   BIC=-440.88
```