Forecasting and Plots

Group4

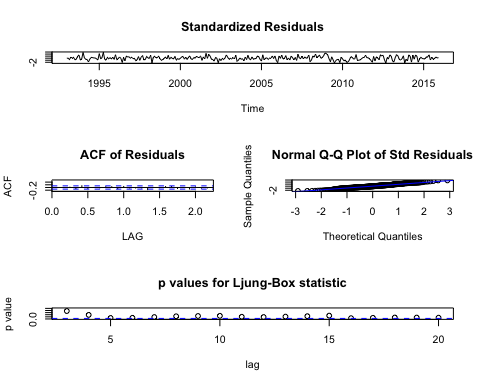
July 22, 2016

# Thank you for all the following hard work

From all the models discussed thus far, I vote for ARIMA(1, 2, 1). It had the lowest AIC, best diagnostics, and greatest parsimony.

sarima(unem, p = 1, d = 2, q = 1)

## initial value -1.454782   
## iter 2 value -1.756280  
## iter 3 value -1.798959  
## iter 4 value -1.806922  
## iter 5 value -1.812862  
## iter 6 value -1.814496  
## iter 7 value -1.815632  
## iter 8 value -1.815971  
## iter 9 value -1.815979  
## iter 10 value -1.815979  
## iter 10 value -1.815979  
## iter 10 value -1.815979  
## final value -1.815979   
## converged  
## initial value -1.817240   
## iter 2 value -1.817260  
## iter 3 value -1.817288  
## iter 4 value -1.817288  
## iter 4 value -1.817288  
## iter 4 value -1.817288  
## final value -1.817288   
## converged



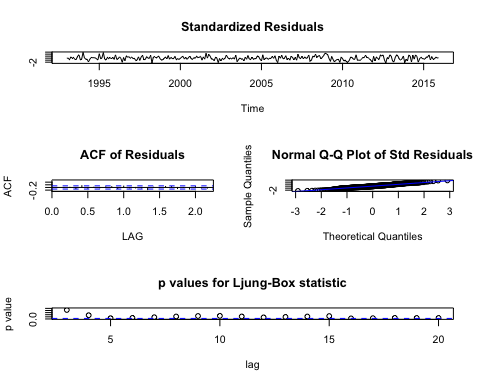
## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,   
## REPORT = 1, reltol = tol))  
##   
## Coefficients:  
## ar1 ma1  
## -0.2021 -0.8078  
## s.e. 0.0688 0.0433  
##   
## sigma^2 estimated as 0.02626: log likelihood = 109.15, aic = -212.3  
##   
## $AIC  
## [1] -2.625197  
##   
## $AICc  
## [1] -2.617631  
##   
## $BIC  
## [1] -3.598962

I then checked if our extra x regressors would help basic ARIMA(1, 2, 1) by using forward selection.

# None of these regressors are significant

sarima(unem, p = 1, d = 2, q = 1, xreg = ipi)

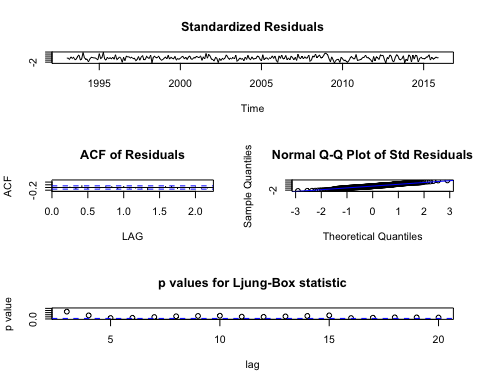
## initial value -1.454955   
## iter 2 value -1.755856  
## iter 3 value -1.798190  
## iter 4 value -1.799051  
## iter 5 value -1.811310  
## iter 6 value -1.815699  
## iter 7 value -1.817959  
## iter 8 value -1.818208  
## iter 9 value -1.818241  
## iter 10 value -1.818254  
## iter 11 value -1.818260  
## iter 12 value -1.818265  
## iter 13 value -1.818265  
## iter 13 value -1.818265  
## iter 13 value -1.818265  
## final value -1.818265   
## converged  
## initial value -1.820049   
## iter 2 value -1.820058  
## iter 3 value -1.820186  
## iter 4 value -1.820194  
## iter 5 value -1.820198  
## iter 5 value -1.820198  
## iter 5 value -1.820198  
## final value -1.820198   
## converged



## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = xreg, optim.control = list(trace = trc, REPORT = 1,   
## reltol = tol))  
##   
## Coefficients:  
## ar1 ma1 xreg  
## -0.1974 -0.8284 -0.0168  
## s.e. 0.0677 0.0434 0.0134  
##   
## sigma^2 estimated as 0.0261: log likelihood = 109.95, aic = -211.89  
##   
## $AIC  
## [1] -2.624139  
##   
## $AICc  
## [1] -2.616358  
##   
## $BIC  
## [1] -3.584787

sarima(unem, p = 1, d = 2, q = 1, xreg = orders)

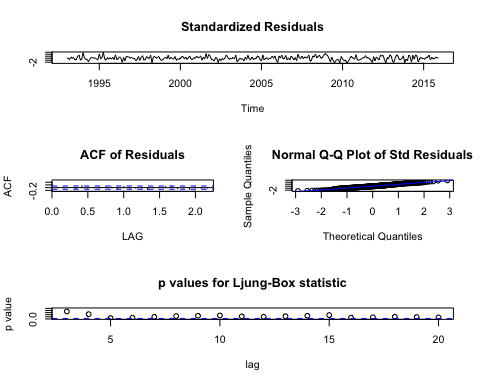
## initial value -1.455655   
## iter 2 value -1.758641  
## iter 3 value -1.800605  
## iter 4 value -1.805766  
## iter 5 value -1.812832  
## iter 6 value -1.815829  
## iter 7 value -1.816599  
## iter 8 value -1.816666  
## iter 9 value -1.816683  
## iter 10 value -1.816699  
## iter 11 value -1.816701  
## iter 12 value -1.816703  
## iter 12 value -1.816703  
## iter 12 value -1.816703  
## final value -1.816703   
## converged  
## initial value -1.817632   
## iter 2 value -1.817666  
## iter 3 value -1.817685  
## iter 4 value -1.817696  
## iter 5 value -1.817698  
## iter 6 value -1.817699  
## iter 6 value -1.817699  
## iter 6 value -1.817699  
## final value -1.817699   
## converged



## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = xreg, optim.control = list(trace = trc, REPORT = 1,   
## reltol = tol))  
##   
## Coefficients:  
## ar1 ma1 xreg  
## -0.2054 -0.8026 3e-04  
## s.e. 0.0695 0.0455 7e-04  
##   
## sigma^2 estimated as 0.02624: log likelihood = 109.26, aic = -210.52  
##   
## $AIC  
## [1] -2.6187  
##   
## $AICc  
## [1] -2.610919  
##   
## $BIC  
## [1] -3.579348

sarima(unem, p = 1, d = 2, q = 1, xreg = house)

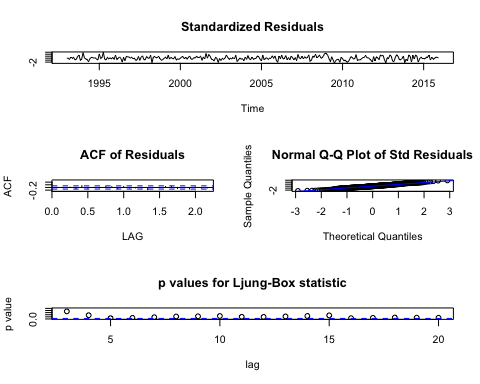
## initial value -1.456005   
## iter 2 value -1.761364  
## iter 3 value -1.804513  
## iter 4 value -1.811421  
## iter 5 value -1.818716  
## iter 6 value -1.818795  
## iter 7 value -1.819478  
## iter 8 value -1.819499  
## iter 9 value -1.819499  
## iter 10 value -1.819499  
## iter 10 value -1.819499  
## iter 10 value -1.819499  
## final value -1.819499   
## converged  
## initial value -1.820356   
## iter 2 value -1.820384  
## iter 3 value -1.820385  
## iter 4 value -1.820395  
## iter 4 value -1.820395  
## iter 4 value -1.820395  
## final value -1.820395   
## converged



## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = xreg, optim.control = list(trace = trc, REPORT = 1,   
## reltol = tol))  
##   
## Coefficients:  
## ar1 ma1 xreg  
## -0.2150 -0.7957 0.0165  
## s.e. 0.0686 0.0435 0.0126  
##   
## sigma^2 estimated as 0.0261: log likelihood = 110, aic = -212  
##   
## $AIC  
## [1] -2.624037  
##   
## $AICc  
## [1] -2.616255  
##   
## $BIC  
## [1] -3.584684

sarima(unem, p = 1, d = 2, q = 1, xreg = constr)

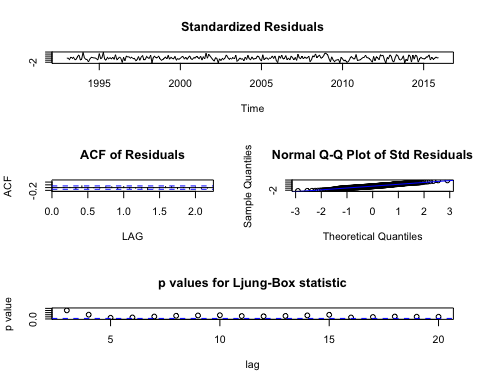
## initial value -1.461000   
## iter 2 value -1.756478  
## iter 3 value -1.799997  
## iter 4 value -1.808408  
## iter 5 value -1.814872  
## iter 6 value -1.816337  
## iter 7 value -1.817805  
## iter 8 value -1.817931  
## iter 9 value -1.817933  
## iter 10 value -1.817934  
## iter 10 value -1.817934  
## iter 10 value -1.817934  
## final value -1.817934   
## converged  
## initial value -1.818759   
## iter 2 value -1.818792  
## iter 3 value -1.818798  
## iter 4 value -1.818804  
## iter 5 value -1.818804  
## iter 6 value -1.818804  
## iter 6 value -1.818804  
## iter 6 value -1.818804  
## final value -1.818804   
## converged



## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = xreg, optim.control = list(trace = trc, REPORT = 1,   
## reltol = tol))  
##   
## Coefficients:  
## ar1 ma1 xreg  
## -0.2005 -0.8036 0.0058  
## s.e. 0.0689 0.0436 0.0063  
##   
## sigma^2 estimated as 0.02618: log likelihood = 109.56, aic = -211.13  
##   
## $AIC  
## [1] -2.620898  
##   
## $AICc  
## [1] -2.613117  
##   
## $BIC  
## [1] -3.581546

sarima(unem, p = 1, d = 2, q = 1, xreg = recession)

## initial value -1.461187   
## iter 2 value -1.761096  
## iter 3 value -1.802223  
## iter 4 value -1.814318  
## iter 5 value -1.816731  
## iter 6 value -1.818908  
## iter 7 value -1.819200  
## iter 8 value -1.819245  
## iter 9 value -1.819249  
## iter 10 value -1.819249  
## iter 11 value -1.819251  
## iter 11 value -1.819251  
## iter 11 value -1.819251  
## final value -1.819251   
## converged  
## initial value -1.820455   
## iter 2 value -1.820479  
## iter 3 value -1.820499  
## iter 4 value -1.820500  
## iter 4 value -1.820500  
## iter 4 value -1.820500  
## final value -1.820500   
## converged

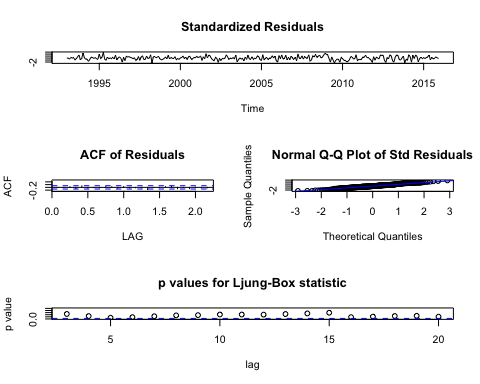


## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = xreg, optim.control = list(trace = trc, REPORT = 1,   
## reltol = tol))  
##   
## Coefficients:  
## ar1 ma1 xreg  
## -0.2042 -0.8003 0.1021  
## s.e. 0.0695 0.0453 0.0767  
##   
## sigma^2 estimated as 0.0261: log likelihood = 110.03, aic = -212.06  
##   
## $AIC  
## [1] -2.624255  
##   
## $AICc  
## [1] -2.616474  
##   
## $BIC  
## [1] -3.584903

When retail is added, it is significant, and the AIC is decreased, so I put it in the model.

sarima(unem, p = 1, d = 2, q = 1, xreg = retail)

## initial value -1.478523   
## iter 2 value -1.769227  
## iter 3 value -1.810050  
## iter 4 value -1.814411  
## iter 5 value -1.823510  
## iter 6 value -1.827035  
## iter 7 value -1.828141  
## iter 8 value -1.828242  
## iter 9 value -1.828244  
## iter 10 value -1.828248  
## iter 11 value -1.828248  
## iter 12 value -1.828249  
## iter 13 value -1.828249  
## iter 13 value -1.828249  
## iter 13 value -1.828249  
## final value -1.828249   
## converged  
## initial value -1.828254   
## iter 2 value -1.828264  
## iter 3 value -1.828278  
## iter 4 value -1.828280  
## iter 5 value -1.828281  
## iter 5 value -1.828281  
## iter 5 value -1.828281  
## final value -1.828281   
## converged



## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = xreg, optim.control = list(trace = trc, REPORT = 1,   
## reltol = tol))  
##   
## Coefficients:  
## ar1 ma1 xreg  
## -0.1911 -0.7975 0.0030  
## s.e. 0.0700 0.0451 0.0012  
##   
## sigma^2 estimated as 0.0257: log likelihood = 112.16, aic = -216.32  
##   
## $AIC  
## [1] -2.639681  
##   
## $AICc  
## [1] -2.6319  
##   
## $BIC  
## [1] -3.600329

Continued with forward selection by adding additional regressors along with retail. None were significant.

\*Travis, for some reason I can't get these to run on my system it says:

"Error in stats::arima(xdata, order = c(p, d, q), seasonal = list(order = c(P, : lengths of 'x' and 'xreg' do not match Calls: ... withVisible -> eval -> eval -> sarima -> Execution halted"

I'll try it on my PC tomorrow but just wanted to give you a heads up.

#sarima(unem, p = 1, d = 2, q = 1, xreg = c(retail, ipi))  
#sarima(unem, p = 1, d = 2, q = 1, xreg = c(retail, orders))  
#sarima(unem, p = 1, d = 2, q = 1, xreg = c(retail, house))  
#sarima(unem, p = 1, d = 2, q = 1, xreg = c(retail, constr))  
#sarima(unem, p = 1, d = 2, q = 1, xreg = c(retail, recession))

Even though ipi is not significant, its presence does decrease the AIC. So I figured I would keep it around just in case.

Adding anything beyond retail and ipi doesn't improve the model. Retail is always significant whenever it's in the model, and it decreases the AIC, so it's probably the most important predictor. However, its coefficient, while significant, is very small (0.0030).

Someone can do backward selection if they want. I couldn't figure out how to do any selection with the stepAIC( ) or stepwise( ) functions, so I just did it manually.

### To summarize, after looking at regressors, I think we have three potential models:

#### ARIMA(1, 2, 1) with no regressors

#### ARIMA(1, 2, 1) with retail as a regressor

#### ARIMA(1, 2, 1) with retail and ipi as regressors

Personally, I like the ARIMA(1, 2, 1) with no regressors. These are my reasons.

1. It's a valid model with great diagnostics.
2. It's the most parsimonious.
3. While retail is statistically significant, I argue that it isn't practially significant, because its coefficient is so small.
4. The forecasted values for unem from the basic ARIMA(1, 2, 1) are very close to the actual ones observed in 2016. (See below).

#### Here is the forecasting for all three ARIMA(1, 2, 1) models discussed before.

##### Read in 2016 data

econ <- read.xlsx("Unemployment.xls", sheetName = "UNRATENSA")

##### Convert retail to billions

**I am getting errors here too** Quitting from lines 92-93 (ForcastingwPlots.Rmd) Error in $<-.data.frame(\*tmp\*, "retail\_sales", value = numeric(0)) : replacement has 0 rows, data has 332 Calls: ... withVisible -> eval -> eval -> $<- -> $<-.data.frame Execution halted

#econ$retail\_sales <- econ$retail\_sales/1000

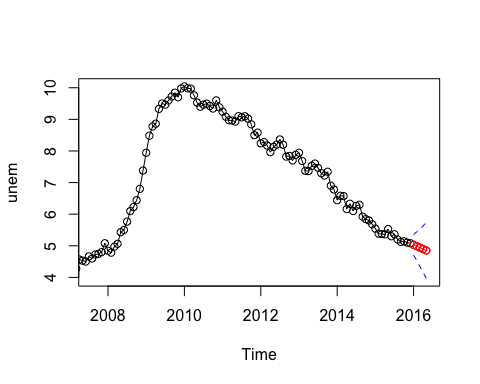
##### Note about seasonal adjusting:

Using Joseph's code, the decompose( ) function doesn't like to do seasonal decomposition unless you have complete cycles of data. We only have 3 retail numbers for 2016 (not the whole year), and only 4 ipi numbers for 2016, so the decompose( ) function did strange things. I don't know how to seasonally adjust only the 2016 values for retail and ipi. If someone could figure out how to seasonally adjust them that would help. I don't think it's a huge deal though.

##### Here is the forecasting for the ARIMA(1, 2, 1) with no regressors

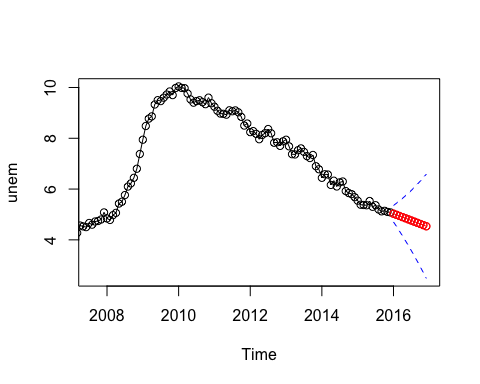
I did five forecasts ahead, because we have those actual unem numbers. Then I did forecasts for one year, and two years. I think going beyond two years might be too far.

sarima.for(unem, p = 1, d = 2, q = 1, n.ahead = 5)



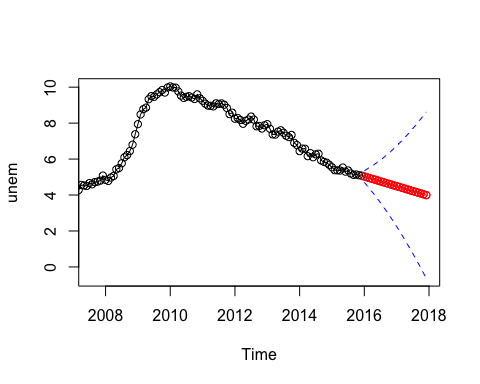
## $pred  
## Jan Feb Mar Apr May  
## 2016 5.030273 4.986233 4.940948 4.895915 4.850831  
##   
## $se  
## Jan Feb Mar Apr May  
## 2016 0.1620509 0.2280503 0.2980669 0.3685263 0.4413540

sarima.for(unem, p = 1, d = 2, q = 1, n.ahead = 12)



## $pred  
## Jan Feb Mar Apr May Jun Jul  
## 2016 5.030273 4.986233 4.940948 4.895915 4.850831 4.805757 4.760681  
## Aug Sep Oct Nov Dec  
## 2016 4.715605 4.670530 4.625454 4.580378 4.535303  
##   
## $se  
## Jan Feb Mar Apr May Jun Jul  
## 2016 0.1620509 0.2280503 0.2980669 0.3685263 0.4413540 0.5167292 0.5948357  
## Aug Sep Oct Nov Dec  
## 2016 0.6757157 0.7593681 0.8457642 0.9348617 1.0266113

sarima.for(unem, p = 1, d = 2, q = 1, n.ahead = 24)



## $pred  
## Jan Feb Mar Apr May Jun Jul  
## 2016 5.030273 4.986233 4.940948 4.895915 4.850831 4.805757 4.760681  
## 2017 4.490227 4.445151 4.400076 4.355000 4.309924 4.264849 4.219773  
## Aug Sep Oct Nov Dec  
## 2016 4.715605 4.670530 4.625454 4.580378 4.535303  
## 2017 4.174697 4.129622 4.084546 4.039470 3.994395  
##   
## $se  
## Jan Feb Mar Apr May Jun Jul  
## 2016 0.1620509 0.2280503 0.2980669 0.3685263 0.4413540 0.5167292 0.5948357  
## 2017 1.1209610 1.2178577 1.3172493 1.4190848 1.5233148 1.6298920 1.7387710  
## Aug Sep Oct Nov Dec  
## 2016 0.6757157 0.7593681 0.8457642 0.9348617 1.0266113  
## 2017 1.8499084 1.9632627 2.0787943 2.1964653 2.3162397

##### Compare the predictions with the actual values.

Jan 2016: actual 5.3 , predicted = 5.0 Feb 2016: actual 5.2 , predicted = 5.0  
Mar 2016: actual 5.1 , predicted = 4.9 Apr 2016: actual 4.7 , predicted = 4.9 May 2016: actual 4.5 , predicted = 5.9

I would say, overall, that is a very good model.

##### Next I did forecasting with the models that had predictors.

Since sarima.for( ) doesn't take a xreg argument, I had to build the model with Arima( ) and then use the forecast( ) function from the {forecast} package.

##### Here is forecast for ARIMA(1, 2, 1) with retail as predictor

fit.for\_retail <- Arima(unem, order = c(1, 2, 1), xreg = retail)  
summary(fit.for\_retail)

## Series: unem   
## ARIMA(1,2,1)   
##   
## Coefficients:  
## ar1 ma1 retail  
## -0.1911 -0.7975 0.0030  
## s.e. 0.0700 0.0451 0.0012  
##   
## sigma^2 estimated as 0.02598: log likelihood=112.16  
## AIC=-216.32 AICc=-216.17 BIC=-201.87  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE  
## Training set 0.0006961878 0.1597189 0.1254584 0.06415485 2.134112  
## MASE ACF1  
## Training set 0.1704633 -0.006400981

The forecast( ) function can only make forecasts when you have values for all the predictors. Thus, with this model, we can only predict the first three months of 2016. I think this is another reason to go with regular ARIMA(1, 2, 1). We could potentially come up with a model for the predictors and then forecast their own times, but this seems like a lot of work and would introduce a lot more variability in our modeling.

**This probably isn't working for me because of the other code I couldn't run earlier. That is why I added in the summary above.** Error in forecast.Arima(fit.for\_retail, h = 3, xreg = econ$retail\_sales[313:315]) : No regressors provided Calls: ... withVisible -> eval -> eval -> forecast -> forecast.Arima Execution halted

#forecast(fit.for\_retail, h = 3, xreg = econ$retail\_sales[313:315])  
#plot(forecast(fit.for\_retail, h = 3, xreg = econ$retail\_sales[313:315]))

##### Here are the actual and predicted unem values:

* Jan 2016: actual 5.3 , predicted = 4.8
* Feb 2016: actual 5.2 , predicted = 4.8
* Mar 2016: actual 5.1 , predicted = 4.9

I think the predictors from the regular ARIMA(1, 2, 1) are better. Again, how are we going to forecast more than three months if we don't know the retail values for more than three months?

##### Here is the forecasting for ARIMA(1, 2, 1) with both retail and ipi as regressors.

**Same problem here for me too. All because I couldn't get the original regressors in.** Error in forecast.Arima(fit.for\_retail\_ipi, h = 5, xreg = cbind(econ$retail\_sales[313:315], : No regressors provided

#fit.for\_retail\_ipi <- Arima(unem, order = c(1, 2, 1), xreg = cbind(retail, ipi))  
#forecast(fit.for\_retail\_ipi, h = 5, xreg = cbind(econ$retail\_sales[313:315], econ$industrial\_production\_index[313:315]))  
#plot(forecast(fit.for\_retail\_ipi, h = 5, xreg = cbind(econ$retail\_sales[313:315], econ$industrial\_production\_index[313:315])))

##### Here are the actual and predicted unem values for this model:

* Jan 2016: actual 5.3 , predicted = 4.8
* Feb 2016: actual 5.2 , predicted = 4.8
* Mar 2016: actual 5.1 , predicted = 4.9

##### This model doesn't really do any better than the one with only retail as a predictor.

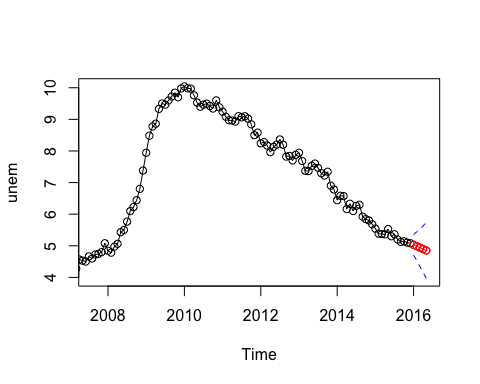
**Overall, I think that we should use the basic ARIMA(1, 2, 1) without any regressors.** Dr. P mentions parsimony constantly, so I think that this a relatively simple, but very useful and valid model.

### All these plots are forecasts with the ARIMA(1, 2, 1) model with seasonally adjusted data and no predictors

Also, if we want to include it, the FRED website has posted the unemployment rate for June 2016 as 5.1%

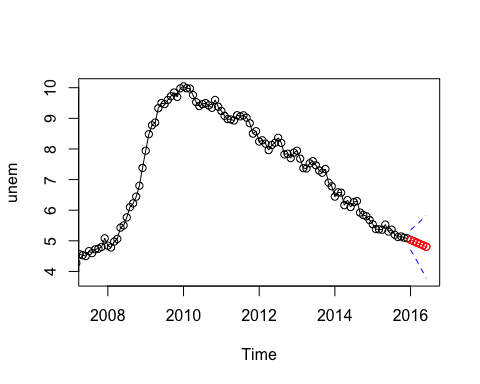
##### Here are the basic sarima( ) plots for the forecasts, h = 5, 6, 12, and 24

sarima.for(unem, p = 1, d = 2, q = 1, n.ahead = 5)



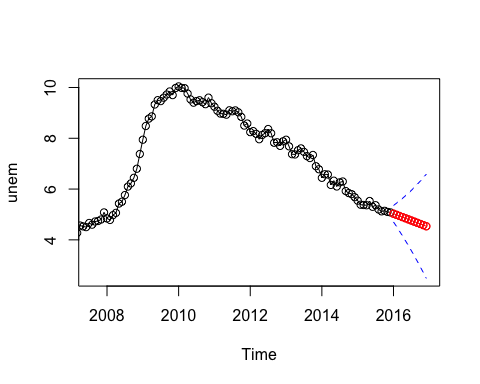
## $pred  
## Jan Feb Mar Apr May  
## 2016 5.030273 4.986233 4.940948 4.895915 4.850831  
##   
## $se  
## Jan Feb Mar Apr May  
## 2016 0.1620509 0.2280503 0.2980669 0.3685263 0.4413540

sarima.for(unem, p = 1, d = 2, q = 1, n.ahead = 6)



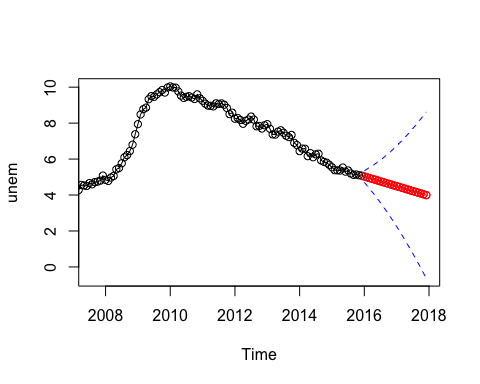
## $pred  
## Jan Feb Mar Apr May Jun  
## 2016 5.030273 4.986233 4.940948 4.895915 4.850831 4.805757  
##   
## $se  
## Jan Feb Mar Apr May Jun  
## 2016 0.1620509 0.2280503 0.2980669 0.3685263 0.4413540 0.5167292

sarima.for(unem, p = 1, d = 2, q = 1, n.ahead = 12)



## $pred  
## Jan Feb Mar Apr May Jun Jul  
## 2016 5.030273 4.986233 4.940948 4.895915 4.850831 4.805757 4.760681  
## Aug Sep Oct Nov Dec  
## 2016 4.715605 4.670530 4.625454 4.580378 4.535303  
##   
## $se  
## Jan Feb Mar Apr May Jun Jul  
## 2016 0.1620509 0.2280503 0.2980669 0.3685263 0.4413540 0.5167292 0.5948357  
## Aug Sep Oct Nov Dec  
## 2016 0.6757157 0.7593681 0.8457642 0.9348617 1.0266113

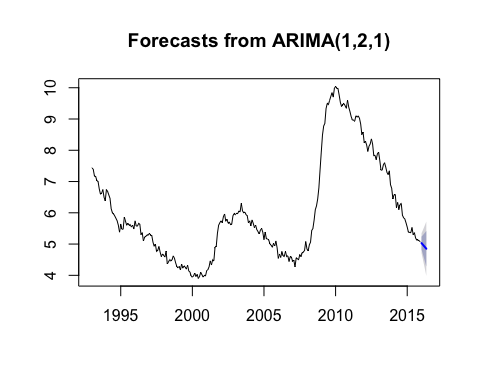
sarima.for(unem, p = 1, d = 2, q = 1, n.ahead = 24)



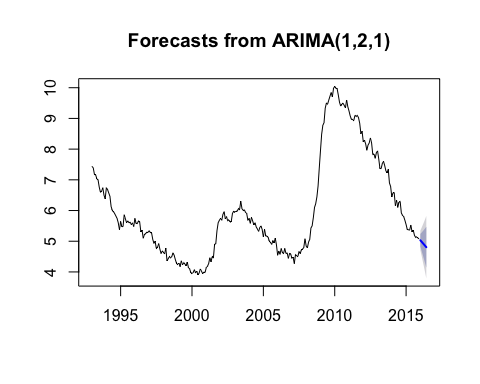
## $pred  
## Jan Feb Mar Apr May Jun Jul  
## 2016 5.030273 4.986233 4.940948 4.895915 4.850831 4.805757 4.760681  
## 2017 4.490227 4.445151 4.400076 4.355000 4.309924 4.264849 4.219773  
## Aug Sep Oct Nov Dec  
## 2016 4.715605 4.670530 4.625454 4.580378 4.535303  
## 2017 4.174697 4.129622 4.084546 4.039470 3.994395  
##   
## $se  
## Jan Feb Mar Apr May Jun Jul  
## 2016 0.1620509 0.2280503 0.2980669 0.3685263 0.4413540 0.5167292 0.5948357  
## 2017 1.1209610 1.2178577 1.3172493 1.4190848 1.5233148 1.6298920 1.7387710  
## Aug Sep Oct Nov Dec  
## 2016 0.6757157 0.7593681 0.8457642 0.9348617 1.0266113  
## 2017 1.8499084 1.9632627 2.0787943 2.1964653 2.3162397

##### Here are the basic Arima( ) plots for the forecasts, h = 5, 6, 12, and 24

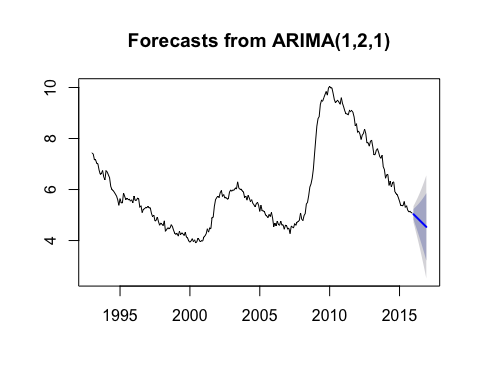
fit <- Arima(unem, order = c(1, 2, 1))  
plot(forecast(fit, h = 5))



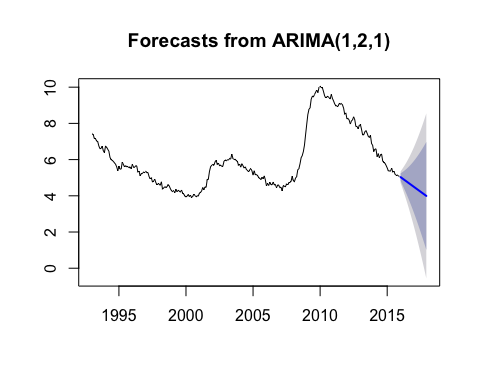
plot(forecast(fit, h = 6))



plot(forecast(fit, h = 12))



plot(forecast(fit, h = 24))

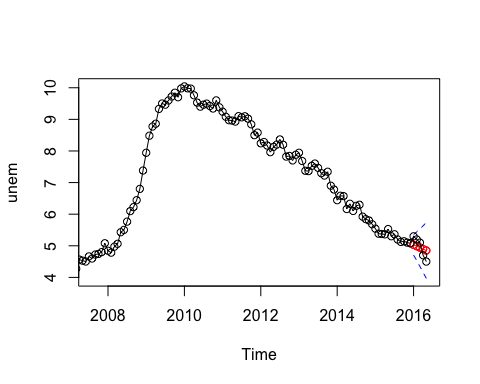


###### Here are the sarima( ) plots with the forecasts and the predicted values for h = 5, and h = 6

sarima.for(unem, p = 1, d = 2, q = 1, n.ahead = 5)

## $pred  
## Jan Feb Mar Apr May  
## 2016 5.030273 4.986233 4.940948 4.895915 4.850831  
##   
## $se  
## Jan Feb Mar Apr May  
## 2016 0.1620509 0.2280503 0.2980669 0.3685263 0.4413540

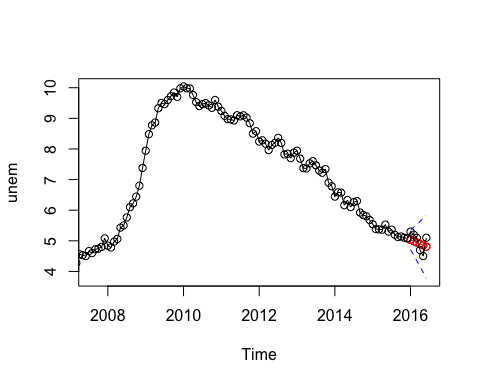
points(seq(2015+11/12, 2016+4/12, by = 1/12), c(5.080477, 5.3, 5.2, 5.1, 4.7, 4.5))  
lines(seq(2015+11/12, 2016+4/12, by = 1/12), c(5.080477, 5.3, 5.2, 5.1, 4.7, 4.5))



sarima.for(unem, p = 1, d = 2, q = 1, n.ahead = 6)

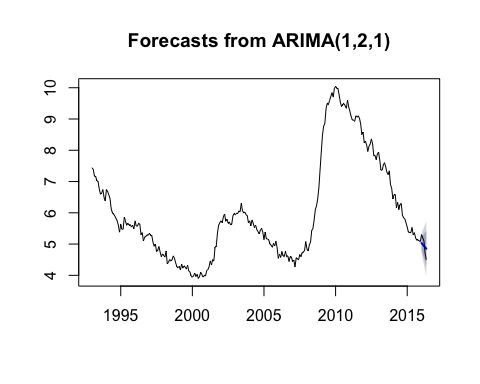
## $pred  
## Jan Feb Mar Apr May Jun  
## 2016 5.030273 4.986233 4.940948 4.895915 4.850831 4.805757  
##   
## $se  
## Jan Feb Mar Apr May Jun  
## 2016 0.1620509 0.2280503 0.2980669 0.3685263 0.4413540 0.5167292

points(seq(2015+11/12, 2016+5/12, by = 1/12), c(5.080477, 5.3, 5.2, 5.1, 4.7, 4.5, 5.1))  
lines(seq(2015+11/12, 2016+5/12, by = 1/12), c(5.080477, 5.3, 5.2, 5.1, 4.7, 4.5, 5.1))



##### Here are the Arima( ) plots with the forecasts and the predicted values for h = 5, and h = 6

plot(forecast(fit, h = 5))  
lines(seq(2015+11/12, 2016+4/12, by = 1/12), c(5.080477, 5.3, 5.2, 5.1, 4.7, 4.5))



plot(forecast(fit, h = 6))  
lines(seq(2015+11/12, 2016+5/12, by = 1/12), c(5.080477, 5.3, 5.2, 5.1, 4.7, 4.5, 5.1))

