Homework 3

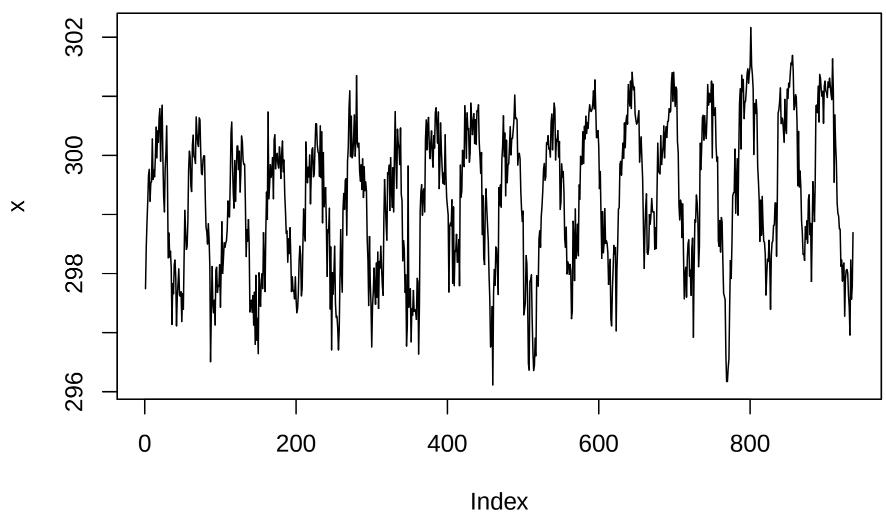
R Sangole

Jun 4, 2018

Contents

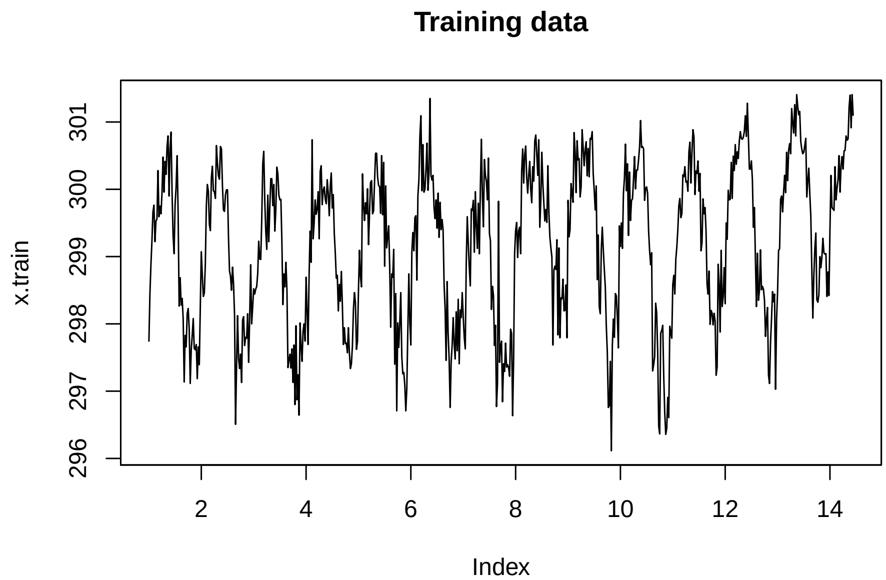
I've decided to use the time series in the final itself for the last homework. Specifically, I'm using the `reanalysis_avg_temp_k` for San Juan. This is what the data looks like:

```
x <- ts(dengueTS.sj$reanalysis_avg_temp_k)
plot.zoo(x)
```

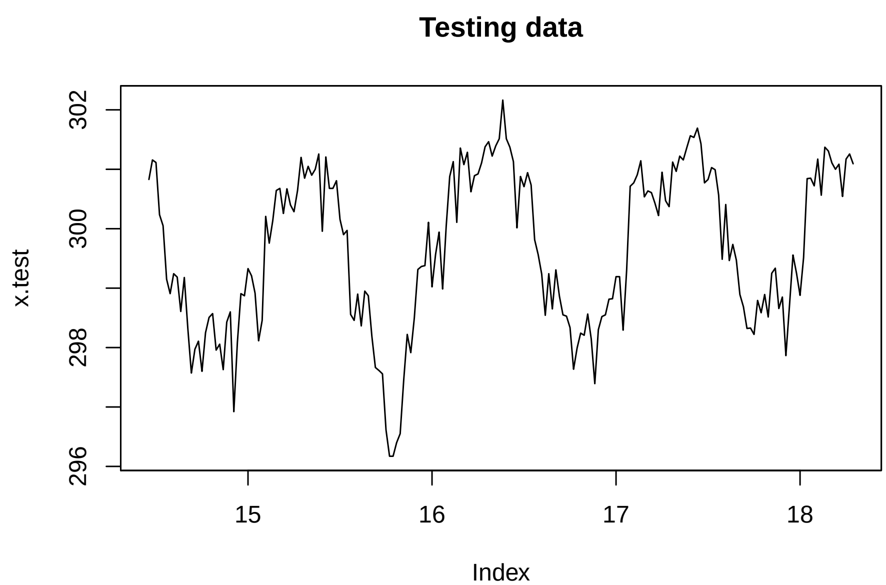


First, splitting the data into training & testing:

```
x.train <- ts(x[1:700], frequency = 52)
plot.zoo(x.train, main = 'Training data')
```




```
x.test <- ts(x[701:900], frequency = 52, start = c(14,25))
plot.zoo(x.test, main = 'Testing data')
```




```
xtest = ts(x.test, start = 14.46154, frequency = 52)
```

I'm running three models:

- 1 - NNETAR
- 2 - ARIMA
- 3 - STLF

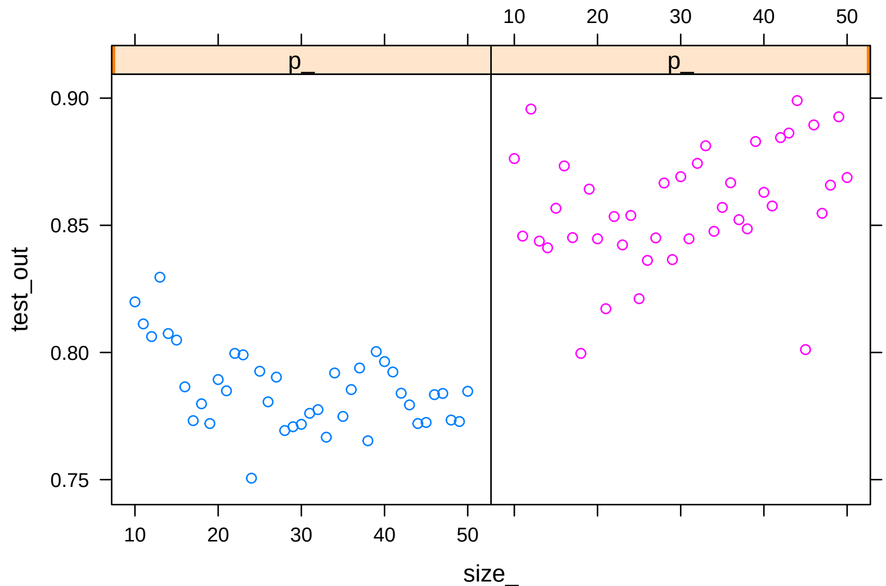
Model 1 - `NNETAR`

The `nnetar` has a few parameters which can be tuned to minimized. I'm minimizing RMSE, while searching the grid for `p` (non-seasonal differences) and `size` (number of hidden nodes in the 1st layer).

This code finds the smallest value of RMSE. The smallest value is found for `size` = 41 and `p` = 1.

```
# Model 1
wk_num = dengueTS.sj$weekofyear
wk_num.train = wk_num[1:700]
wk_num.test = wk_num[701:900]

size_ <- 10:50
p_ <- 1:2
grid <- expand.grid(size_,p_, NA)
names(grid) <- c('size_','p_','test_out')
for(i in 1:dim(grid)[1]){
    X = grid[i, ]
    nnetFit <- x.train %>%
        forecast::nnetar(repeats = 10,size = X[[1]], p = X[[2]], P = 1, xreg = wk_num.train) %
        forecast::forecast(h=200, xreg = wk_num.test)
    test_RMSE <- accuracy(nnetFit, x = x.test)[2,2]
    grid[as.numeric(rownames(X)),3] <- test_RMSE
}
xyplot(test_out~size_|p_, groups=p_, grid)
```

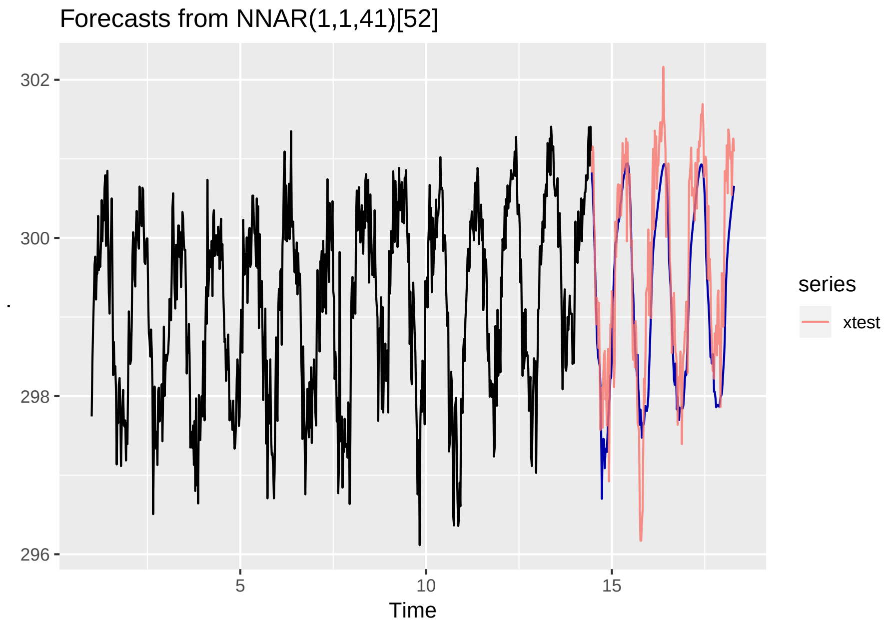



```
nnetFit <- x.train %>%
    forecast::nnetar(repeats = 10,size = 41, p = 1, P = 1, xreg = wk_num.train) %>%
    forecast::forecast(h=200, xreg = wk_num.test)
accuracy(nnetFit, x = ts(x.test,frequency = 52, start = c(14,25)))
```

```
##                        ME      RMSE       MAE           MPE      MAPE
## Training set -0.002177355 0.4510197 0.3546691 -0.0009625405 0.1186139
## Test set      0.410742972 0.7632390 0.6243170  0.1366586073 0.2083576
##                   MASE       ACF1 Theil's U
## Training set 0.5905077 -0.1009886        NA
```

```
## Test set     1.0394589  0.6006436  1.350123
```

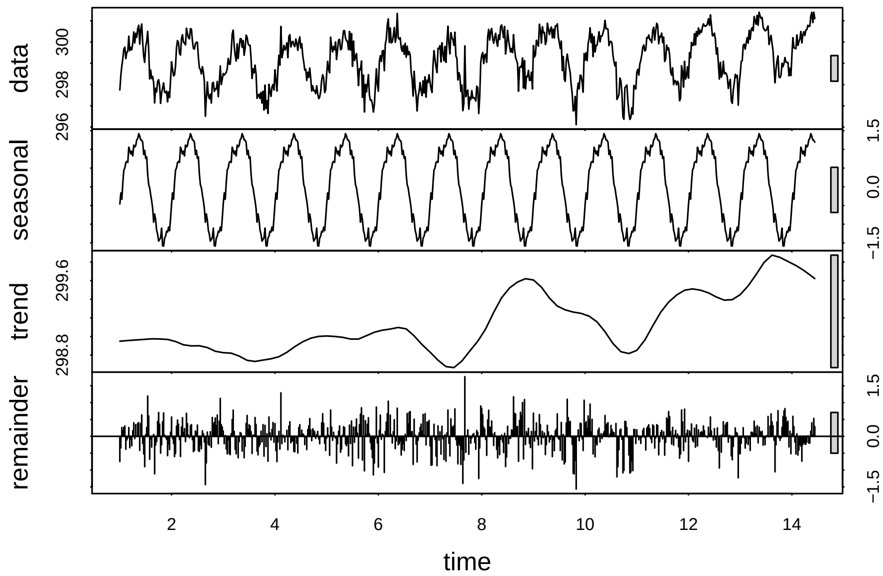
```r
nnetFit %>%
    autoplot() +
    forecast::autolayer(xtest, alpha = 0.8)
```



Model 2 - `Auto.Arima`

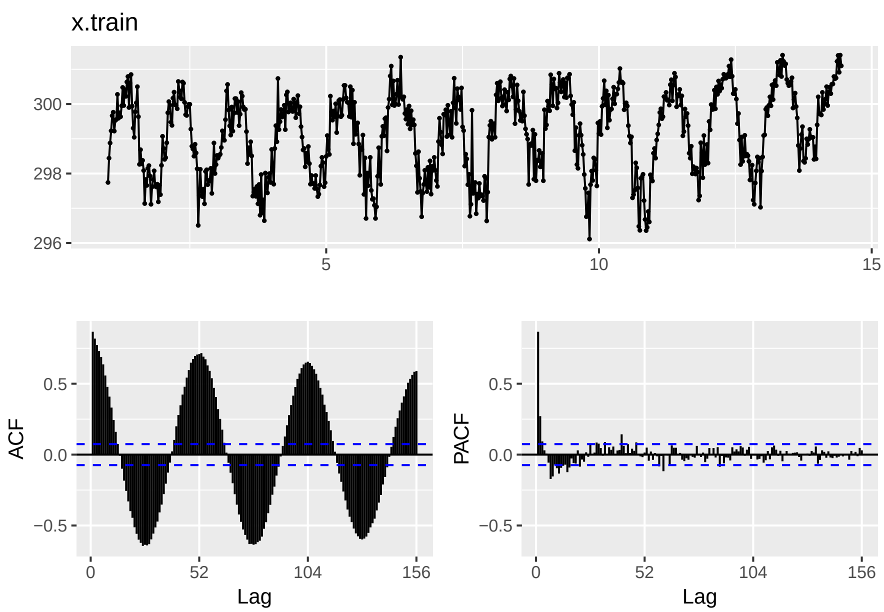
This model is a standard arima model, with seasonal differences. The seasonality is 52 weeks. The decomposition shows a good fit for the seasonal extraction, and acceptable trend.

```r
# Model 2
xtest = ts(x.test, start = 14.46154, frequency = 52)
par('mar'=c(1.2,3,1,1))
stl(x.train, s.window = 'p') %>% plot
```

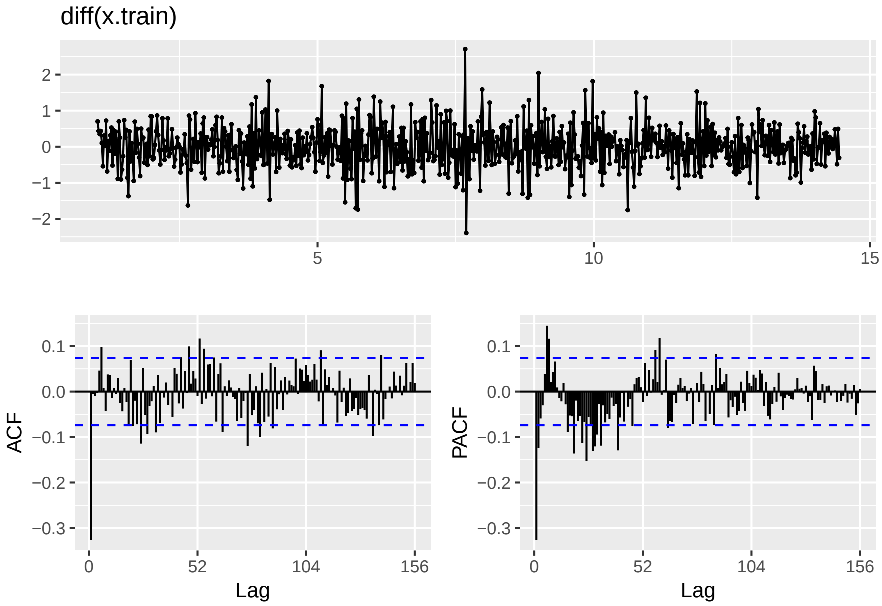


Non-differenced P/ACF show strong seasonality. Upon differencing, we can see that there is a remnant 52-week lag structure. PACF shows that p=2 seems to be required for a stationary series.

```
forecast::ggtsdisplay(x.train)
```




```
forecast::ggtsdisplay(diff(x.train))
```



Fitting an auto arima model, we can see that a (2,0,1)(1,1,0) fits well with a 52-week lag structure for the seasonality.

```
arimaFit <- auto.arima(x.train)
arimaFit
```

```
## Series: x.train
## ARIMA(2,0,1)(1,1,0)[52]
##
## Coefficients:
##          ar1      ar2      ma1     sar1
##       1.1736  -0.1959  -0.8762  -0.5396
## s.e.  0.0498   0.0454   0.0284   0.0329
##
## sigma^2 estimated as 0.3481:  log likelihood=-577.93
## AIC=1165.87   AICc=1165.96   BIC=1188.24
```

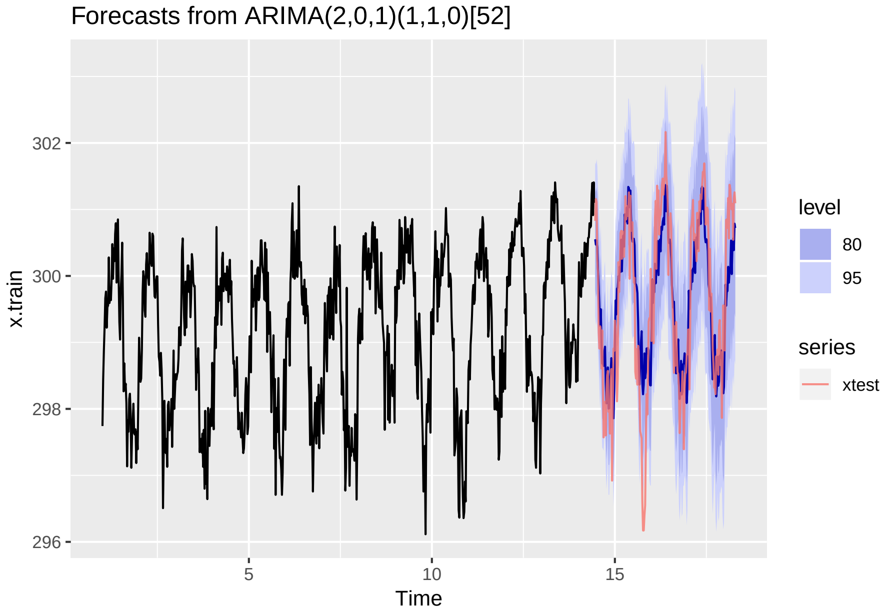
Forecasts fit quite well!

```
arimaFit %>%
    forecast(h=200) %>%
    accuracy(x = ts(x.test,frequency = 52, start = c(14,25)))
```

```
##                        ME      RMSE       MAE          MPE      MAPE
## Training set  0.037506835 0.5659209 0.4422658  0.012308431 0.1479212
## Test set     -0.003053442 0.7309725 0.5630547 -0.001774609 0.1881292
##                   MASE       ACF1 Theil's U
## Training set 0.7363522 0.01571643        NA
## Test set     0.9374601 0.64511930  1.295214
```

```
arimaFit %>%
    forecast(h=200) %>%
    autoplot()+
```

```r
    forecast::autolayer(xtest,alpha=0.8)
```



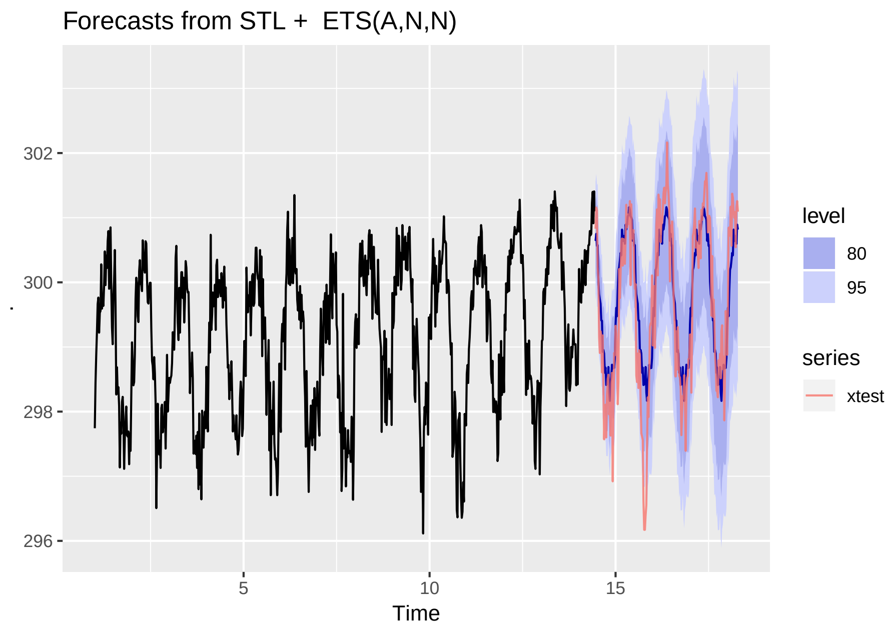
Model 3 - STLF

The STLF is so easy to use…

```r
# Model 3
stlfFit <- x.train %>%
    forecast::stlf(y = ., h = 200, s.window = 'periodic', robust = T)
```

And the model fits quite well!

```r
stlfFit %>%
    autoplot()+
    forecast::autolayer(xtest,alpha=0.8)
```




```r
accuracy(stlfFit, xtest)
```

```
##                        ME      RMSE       MAE          MPE      MAPE
## Training set  0.007698103 0.4652713 0.3604522  0.002345785 0.1205678
## Test set     -0.094119478 0.6627146 0.4889362 -0.032103212 0.1634252
##                   MASE      ACF1 Theil's U
## Training set 0.6001362 0.1600782        NA
## Test set     0.8140562 0.5959591  1.172407
```

Comparison of the models

The models rank in this order:

1. STLF (RMSE = 0.66)
2. ARIMA (RMSE = 0.73)
3. NNAR (RMSE = 0.77)