Workshop 4: Forecasting

(there's a "four-casting" pun in there somewhere)

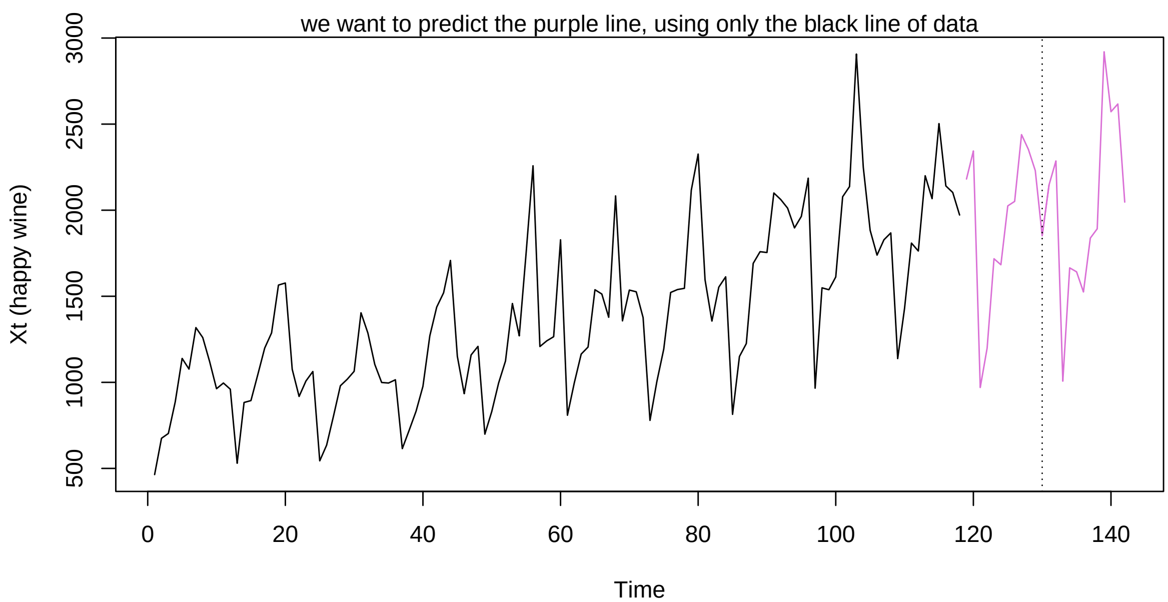
STAT 464/864 | Discrete Time Series Analysis
Skye P. Griffith, Queen's University - Fall 2024

Setup

```
library(itsmr); library(knitr)
```

Dataset	Happy Australian Red Wine Sales (unit = kilolitres)
Source	Included in the ITSMR Package (no external files)
Total Times Sampled	(Monthly) Jan, 1980 – Oct, 1991 (142 total obs.)
Truncated Series	Jan, 1980 – Oct, 1989 (118 obs.)
Times to Predict	Nov, 1989 – Oct, 1991 (24 obs.)

```
t.past   <- 1:118    ;  wine.past   <- wine[t.past  ]  # truncated data
t.future <- 119:142  ;  wine.future <- wine[t.future]  # desired prediction
```



Basic Forecasting using ITSMR

Prediction

1. Model the data's trend + seasonal components via classical time-series decomposition.

   Note that the available methods include differencing, the effect of which is equivalent to modelling step 3 as an ARIMA process. We will discuss this in W5.

2. Obtain the residuals from your classical model.
3. Model the residuals as an ARMA(p,q) process.
4. Predict some h steps into the future.

Forecasting Plot

1. Plot the "past data," leaving enough horizontal space for our prediction line, and enough vertical space for its 95% confidence interval.
2. Add our prediction and its 95% CI to the plot.

Comparison plot

1. Plot the true "future data" (last 12 months of the original dataset) by itself.
2. Add our forecast (+ its 95% CI) to the plot, on top of the true data.

Interpretation

1. Does the true data generally fall within the 95% CI? It should, 95% of the time.
2. How wide is that CI? The wider this interval, the worse our estimate's stardard error is.

Workshop 4 Models

Model	Trend Estimation	Seasonal Component Estimation	ARMA(p,q)
0	Linear Regression	S1 Method; $d = 12$	MA(3)
1	Linear Regression	Harmonic regression; $d = 4$	AR(2)
2	Linear Regression	Harmonic regression; $d = 12$	AR(2)
3	Linear Regression	Harmonic regression; $d = 4, 12$	AR(2)

Model 0

```r
# --- Classical Model
M.0 <- c("trend" , 1,              # model a linear trend
         "season", 12)             # S1 method with period 12
R.0 <- Resid(wine.past, M = M.0)  # get classical model's residuals

# --- ARMA
A.0 <- arma(x = R.0,  # input residuals
            p = 0,    # Order - AR(p)
            q = 3)    # Order - MA(q)

# --- Forecast
fc.0 <- forecast(wine.past,  # Predictors
                 M = M.0,    # classical model
                 a = A.0,    # ARMA(p,q) model
                 h = 24,     # Predict the next 24 months
                 opt = 0)    # don't display result: plz always set opt = 0
```

Models 1,2 & 3

```r
# --- Classical Models
M.1 <- c("trend", 1,    # linear regression
         "hr"   , 4)    # Harmonic regression with period 4

M.2 <- c("trend", 1, "hr", 12)
M.3 <- c("trend", 1, "hr", c(4,12))

# --- Residuals
R.1 <- Resid(wine.past, M = M.1)
R.2 <- Resid(wine.past, M = M.2)
R.3 <- Resid(wine.past, M = M.3)

# --- AR(2), equivalently ARMA(2,0)
A.1 <- arma(x = R.1, p = 2, q = 0)
A.2 <- arma(x = R.2, p = 2, q = 0)
A.3 <- arma(x = R.3, p = 2, q = 0)

# --- Forecast
fc.1 <- forecast(wine.past, M.1, A.1, 24, 0)
fc.2 <- forecast(wine.past, M.2, A.2, 24, 0)
fc.3 <- forecast(wine.past, M.3, A.3, 24, 0)
```

Plotting Prep

Plot Function

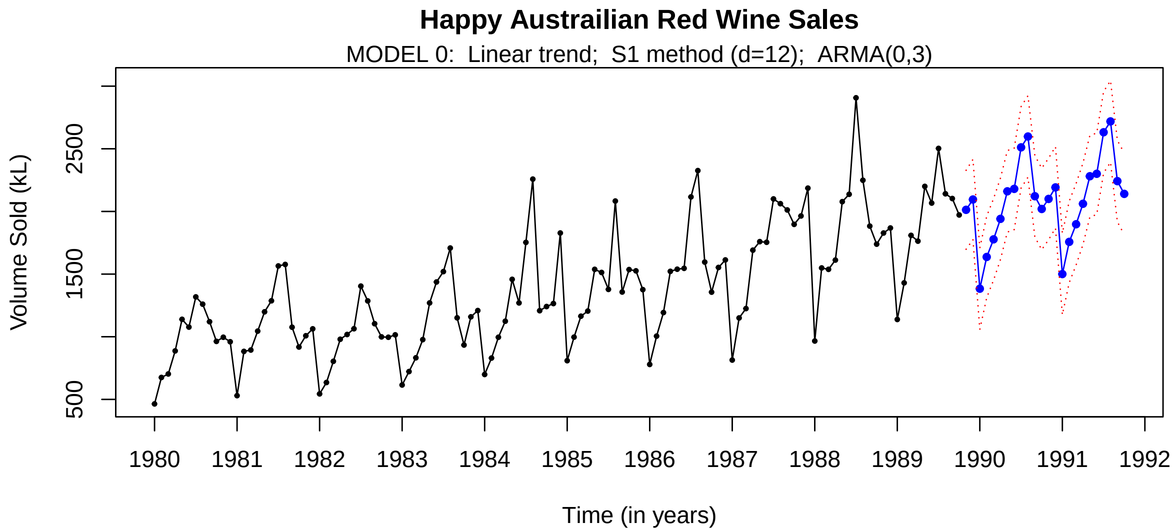
```
plot.wine <- function(x.range = range(t.past,t.future), # default x-axis lim
                      y.range = range(wine)){           # default y-axis lim

  plot.ts(wine.past,
          xlim = x.range,
          ylim = y.range,

          # ---- the rest is all from workshop 2
          main = "Happy Austrailian Red Wine Sales",  # main title
          xlab = "Time (in years)",                   # x-axis label
          ylab = "Volume Sold (kL)",                  # y-axis label
          type = "o",                                 # lines + points
          pch = 20, cex = 0.6,                        # bullets
          xaxt = 'n')                                 # NO X-AXIS TICKS

  axis(side = 1,              # bottom edge of plot
       at   = 12*(0:12) + 1,  # 1 tick every January (total = 13)
       labels = 1980:1992)    # tick labels
}
```

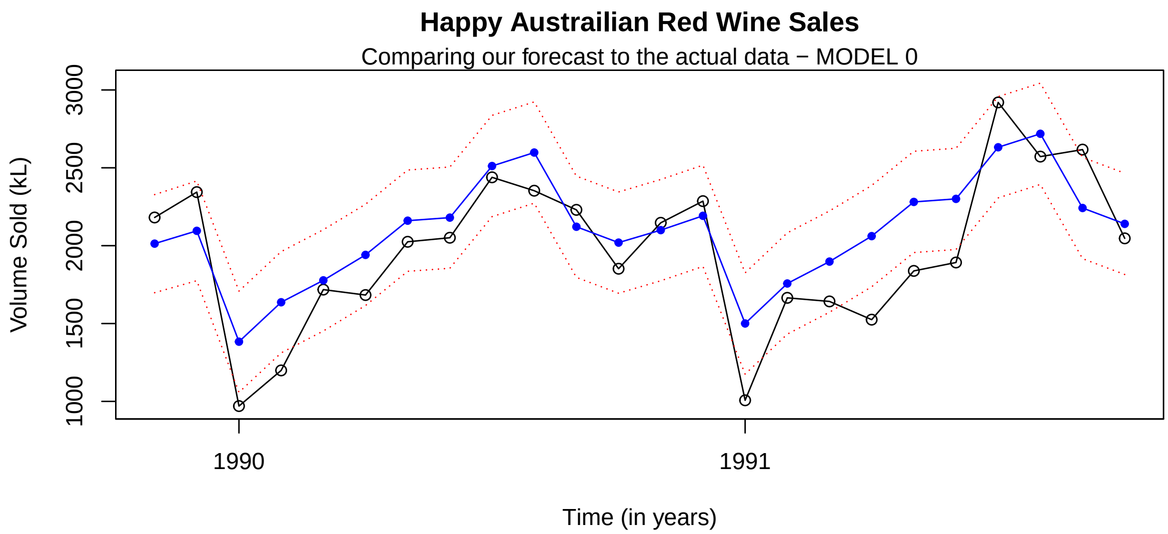
Forecasting Plot Function

```
plot.forecast <- function(fc){                       # input prediction
  points(x = t.future,                               # plot prediction onto future
         y = fc$pred,
         col = "blue", type = "o", pch = 20)         # blue lines with solid points

  lines(t.future, fc$l, col = "red", lty = 3)        # lower confidence limit (95%)
  lines(t.future, fc$u, col = "red", lty = 3)        # upper confidence limit (95%)

# axis(1, t.future[-3], month.name[-3],               # add month names (h=12 only)
# las = 2, tick = FALSE, font = 3, line = -0.6)      # no tick marks, italic
}
```

Results - Model 0

```
par(mfrow = c(2,1), mar = c(4,4,3.5,1))

# --- Forecasting Plot
plot.wine(y.range = range(wine, fc.0$l, fc.0$u)) # make room for 95% CI
mtext("MODEL 0:  Linear trend;  S1 method (d=12);  ARMA(0,3)")
plot.forecast(fc.0)

# --- Comparison Plot
plot.wine(x.range = range(t.future),
          y.range = range(wine.future, fc.0$l, fc.0$u))
mtext("Comparing our forecast to the actual data - MODEL 0")

points(x = t.future, y = wine.future, type = "o")  # TRUE future vals
plot.forecast(fc.0)                                # ESTIMATED future vals
```





Interpretation

1. The true data exits our forecast's confidence interval 7 times.

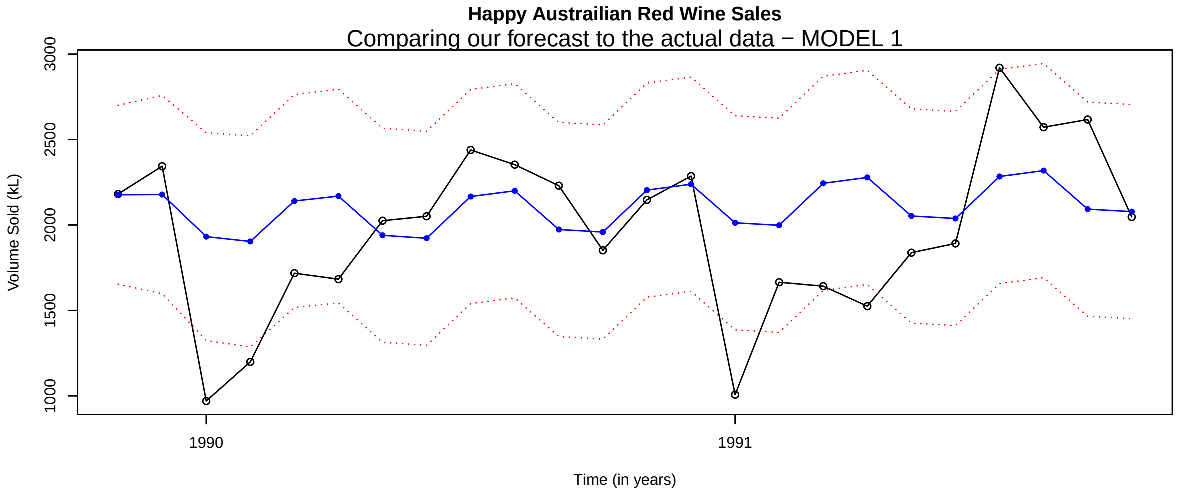
   This is more than $\alpha \times h = 0.05 \times 24 = 1.2$,
   which means our CI captured the true data less than the desired 95% of the time.

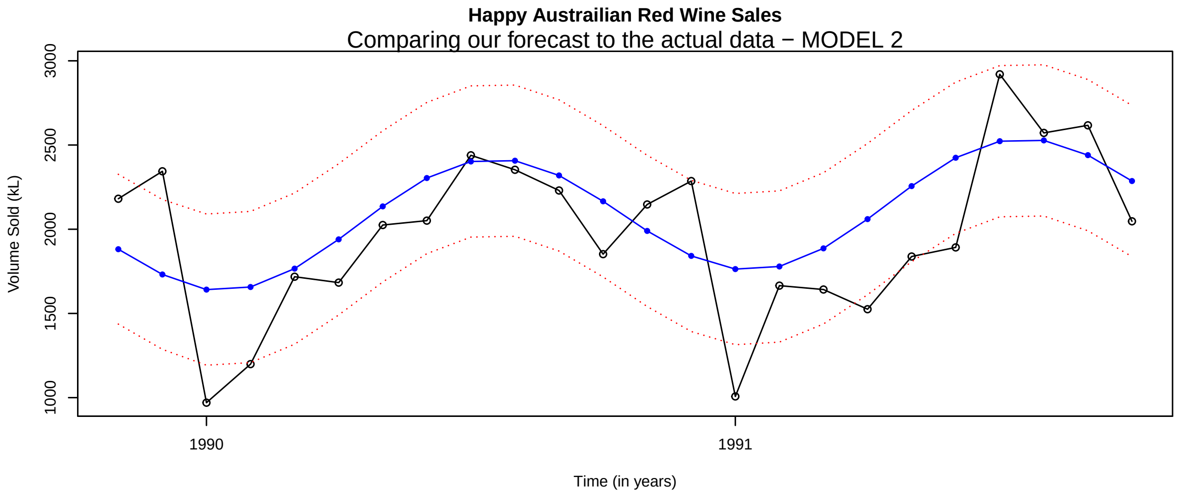
   Reminder: we are not technically performing any hypothesis tests here,
   we're just getting an intuition for how well our forecast fits the data.

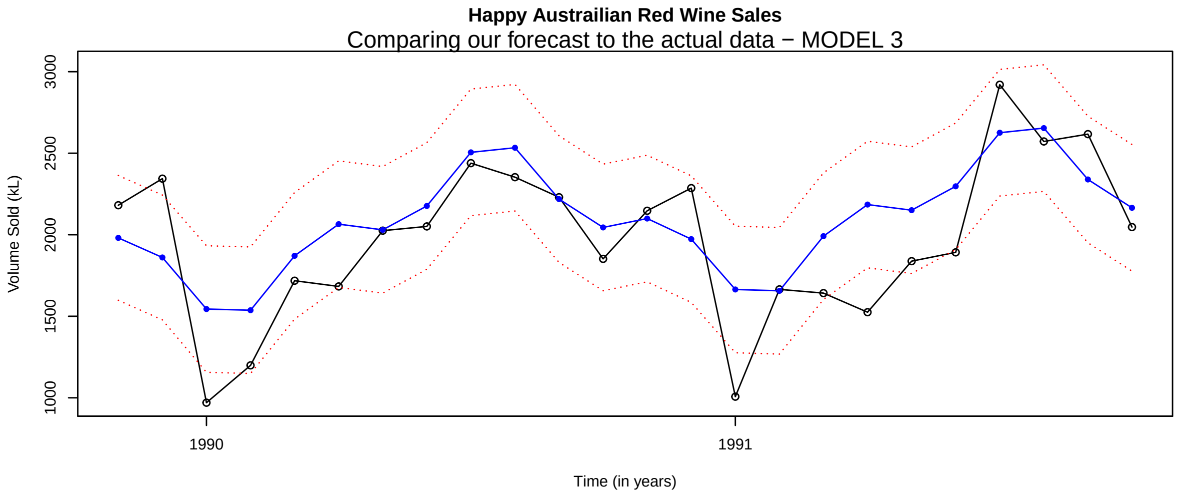
2. The width of the CI looks pretty good to me, but we haven't viewed the results of the other models yet, so we don't have much to compare it with.

Results - Models 1, 2 & 3

```
par(mfrow = c(3,1), mar = c(4,4,4,1))

# --- Model 1
plot.wine(x.range=range(t.future), y.range=range(wine.future,fc.1$l,fc.1$u))
mtext("Comparing our forecast to the actual data - MODEL 1")

points(x = t.future, y = wine.future, type = "o")  # TRUE future vals
plot.forecast(fc.1)                                 # ESTIMATED future vals

# --- Model 2
plot.wine(x.range=range(t.future), y.range=range(wine.future,fc.2$l,fc.2$u))
mtext("Comparing our forecast to the actual data - MODEL 2")

points(x = t.future, y = wine.future, type = "o")  # TRUE future vals
plot.forecast(fc.2)                                 # ESTIMATED future vals

# --- Model 3
plot.wine(x.range=range(t.future), y.range=range(wine.future,fc.3$l,fc.3$u))
mtext("Comparing our forecast to the actual data - MODEL 3")

points(x = t.future, y = wine.future, type = "o")  # TRUE future vals
plot.forecast(fc.3)                                 # ESTIMATED future vals
```


Happy Austrailian Red Wine Sales
Comparing our forecast to the actual data − MODEL 1
Volume Sold (kL)
1000
1500
2000
2500
3000
1990
1991
Time (in years)


Happy Austrailian Red Wine Sales
Comparing our forecast to the actual data − MODEL 2
Volume Sold (kL)
1000
1500
2000
2500
3000
1990
1991
Time (in years)


Happy Austrailian Red Wine Sales
Comparing our forecast to the actual data − MODEL 3
Volume Sold (kL)
1000
1500
2000
2500
3000
1990
1991
Time (in years)

Interpretation

1. The 95% CI's for Models 1 and 3 fail to capture the true data *5 times*, each, over the predicted time interval.

   Model 2's CI fails to capture said data *6 times*, implying a slightly less reliable fit, but still a slight improvement from Model 0. (This discrepancy is quite small).

2. The tighter CI on Model 3 indicates a reduced standard error from that of Model 1.

3. We conclude that our best forecast is either that obtained from Model 3, or perhaps Model 0 − in defense of the latter, I would say that the shape of Model 0's prediction is more accurate than Model 3, due to the S1 method not assuming a harmonic structure.