Workshop 2: the PEN

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

Setup

Quarto renders from a blank slate: it runs code chunks *in order,* and based on an *empty environment.* That means you'll have to load any packages you'll be using, even if they're already loaded in your R session. You'll also have to load any data you plan to work with. Do all this at the beginning of the document, so the rest of your chunks are ready to run.

Packages

`itsmr` (from the textbook) If you need to install it, run the code

```r
install.packages("itsmr")
```

in the console. The CONSOLE. Not the SCRIPT.

```r
library(itsmr) # Load ITSMR. This also loads the wine dataset
```

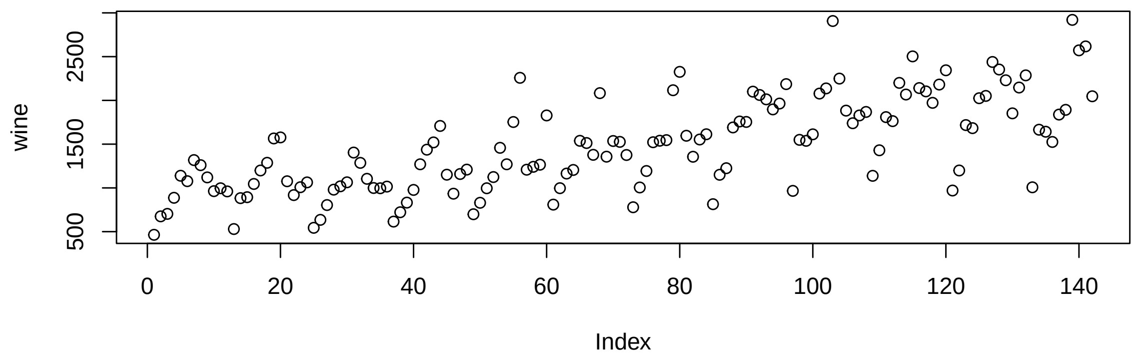
Data

Info	Description
Dataset	Happy Australian Red Wine Sales (unit = kilolitres)
Times Sampled	(Monthly) Jan, 1980 – Oct, 1991 (142 total obs.)
Source	ITSM Time Series Package

It's included with the ITSMR package, so we don't need to load any external files.

Plotting

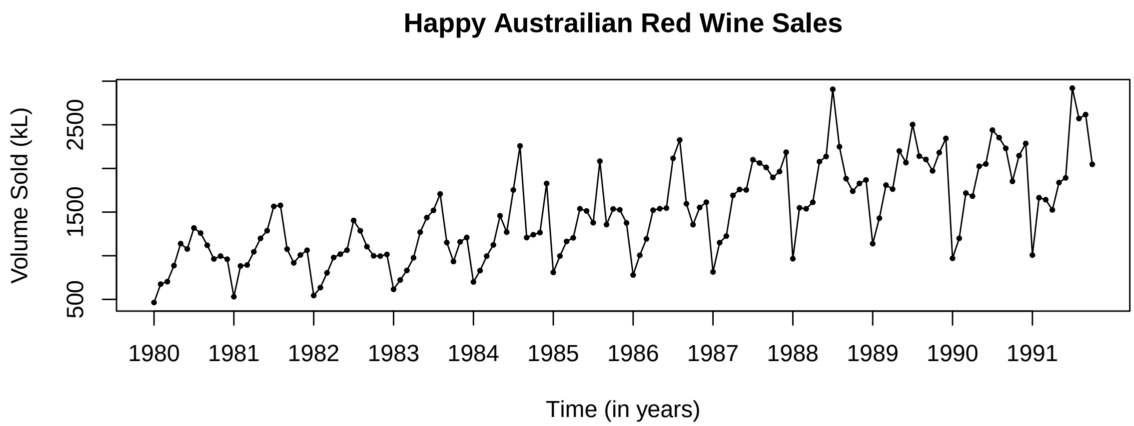
```
# --- Plotting
# make the world's most basic plot of the data
plot(wine)
```



Okay… this doesn't really tell us what's going on. It doesn't demonstrate the data's most interesting patterns, and it doesn't give the viewer any context. Let's now create a scientifically meaningful plot of the data, with an appropriate x (time) axis.

```
# --- Time Axis: create a function [axis.wine()] to add this axis on the fly
year.ticks <- 12*(0:11) + 1 # indicate 1 tick every January (total = 12)

axis.wine <- function(){
  axis(side = 1,              # bottom edge of plot
       at    = year.ticks,    # tick placement
       labels = 1980:1991)    # tick labels
}

# --- Base plot: create a function [plot.wine()] to speed-plot the data
plot.wine <- function(){
  plot.ts(wine,
          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 (trust me)
```

```
            xaxt = 'n')                                   # NO X-AXIS TICKS! (yet)
  axis.wine() # add x-axis
}

# --- Test wine-plotting function
plot.wine()
```



Analysis

The Plan

We want to decompose the data according to the classical model

$$X_t = m_t + s_t + Y_t \qquad (\star)$$

Think of it like this: X_t is a pen, and x_t are the lines drawn by the pen. We've run out of ink. So now we have to get a tube of the same coloured ink that will fit the pen's model.

1. **Eliminate** m_t :
   This is the shell — the *general shape* of the pen. Remove it.
   What your left with is $\hat{r}_m = X_t - \hat{m}_t$.
2. **Extract** s_t :
   Remove the spring + clicky components that are responsible for the pen's *repetitive pattern* of being open-closed-open-closed.
   Now you have $\hat{r} = X_t - \hat{m}_t - \hat{s}_t$ (the ink tube).
3. **Examine** Y_t :
   Look at this tube of *residual* ink. How long/wide is it? What colour is the ink?

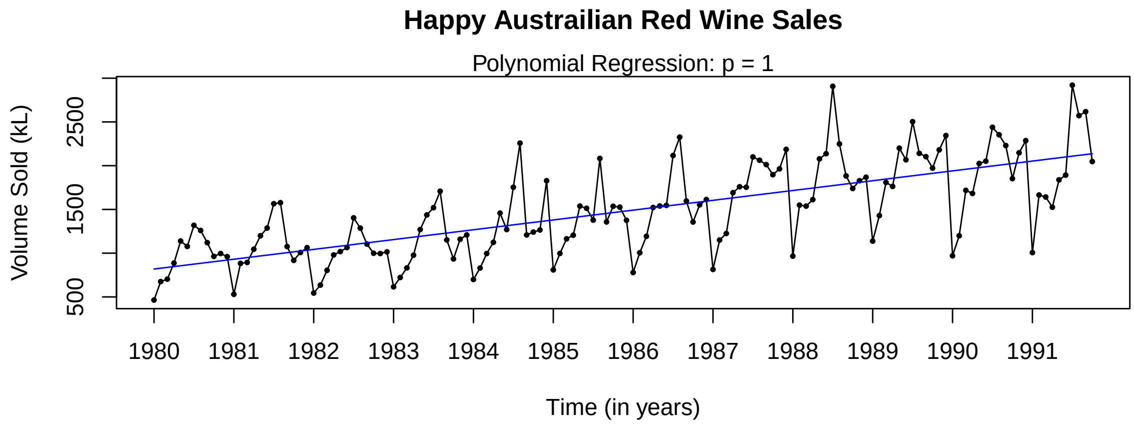
Later in this course, we'll learn where to get more ink and how to reconstruct the pen.

Eliminate $\mathrm{m_t}$: The Body of the pen

Polynomial Regression

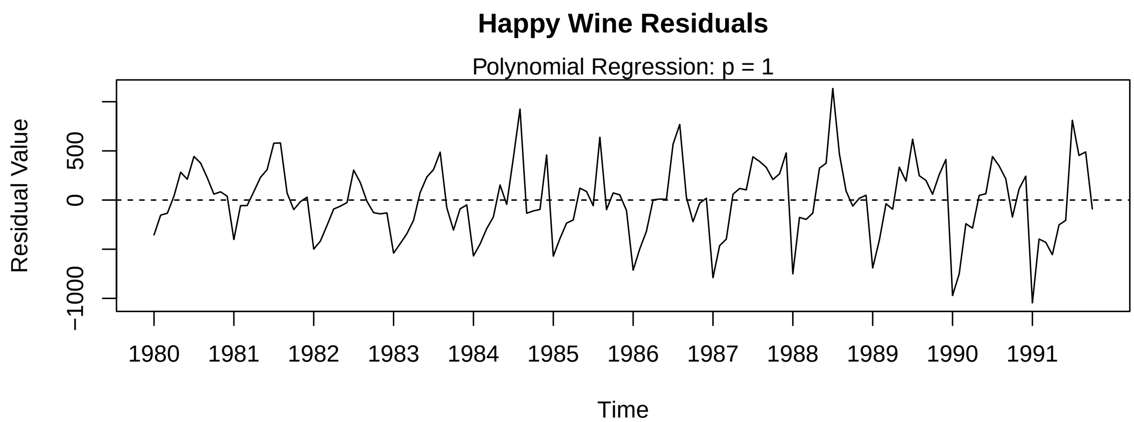
We suspect there may be a linear trend. Or, it's possible that a shallow quadratic may fit the data. I'm going to go with linear, that is, $p = 1$.

```
# --- Estimation
m.pr <- trend(wine, p = 1)

# --- Plotting
plot.wine()                           # Initial plot, using our function
mtext("Polynomial Regression: p = 1") # Subtitle indicating model used
lines(m.pr, col = "blue")             # Adds our m_t estimate
```



Now we need to compute and plot the residuals: $(x_t - \hat{m}_t)$.
This is like the ink tube with the spring still attached.

```
# --- Residuals
rm.pr <- wine - m.pr

# --- Plot
plot.ts(rm.pr,
        main = "Happy Wine Residuals",
        ylab = "Residual Value",
        xaxt = 'n')  # <- makes room for our special x-axis
axis.wine()                               # add special x-axis
mtext("Polynomial Regression: p = 1")  # indicate model
abline(h = 0, lty = 2)                    # (draws a dashed line at y = 0)
```

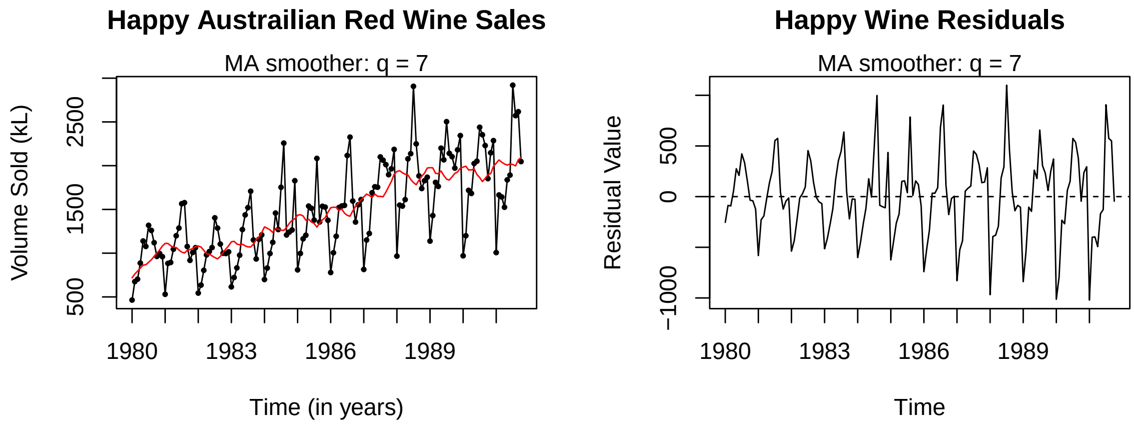


What we want to see is some kind of noisy but regular wave-like structure, since we plan to model seasonality, next. These residuals look ready to go. If there was a linear/curved trend remaining, we would want to consider a different model.

MA Smoothing

Let's repeat the same process, using a moving average smoother ($q = 7$) instead of polynomial regression. We'll plot the estimate and residuals side-by-side.

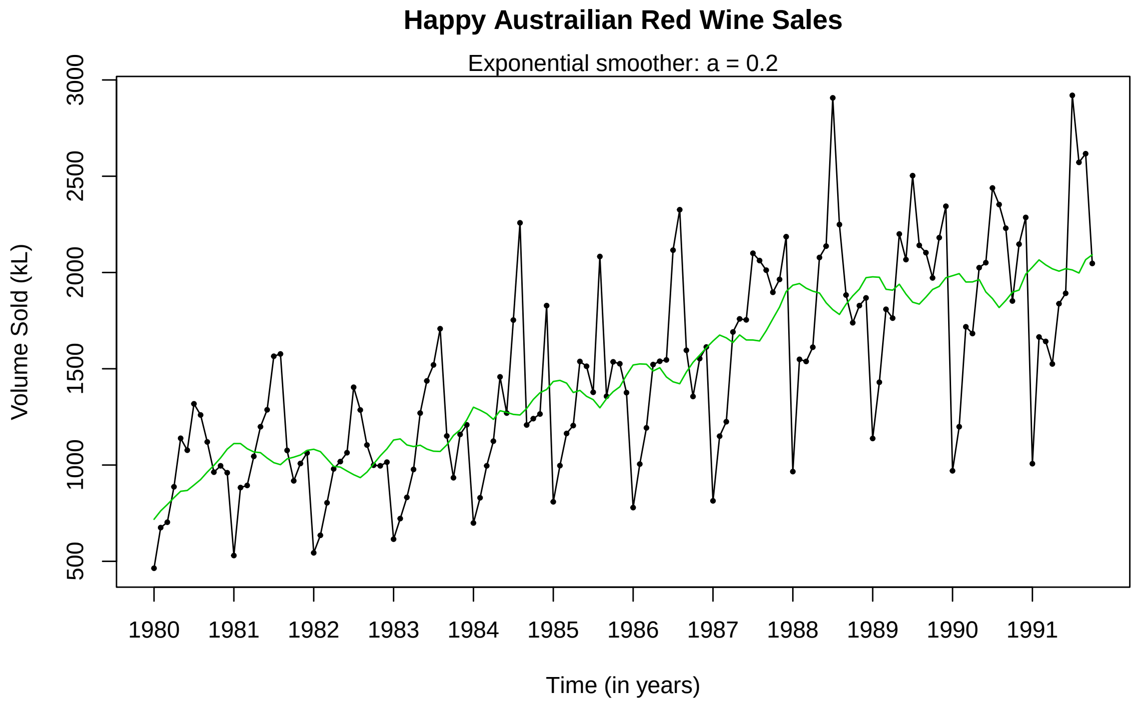
```
# --- Trend Estimation & Residuals
m.ma <- smooth.ma(wine, q = 7) # estimated m_t
rm.ma <- wine - m.ma           # calculated residuals x_t - \hat{m}_t

# --- Plotting
par(mfrow = c(1,2)) # OPTIONAL matrix of plots: 1 row, 2 columns
                    # (it puts 2 plots side-by-side in the PDF)
                    # (not extremely important but good to know how to do)

# Estimated m_t
plot.wine()                    # Initial plot, using our function
mtext("MA smoother: q = 7")    # Subtitle indicating model used
lines(m.ma, col = "red")       # Adds our NEW (MA-smoother) m_t estimate

# Residuals from m_t
plot.ts(rm.ma,                         # MA-smoother residuals!
        main = "Happy Wine Residuals",
        ylab = "Residual Value",
        xaxt = 'n')  # <- makes room for our special x-axis
axis.wine()                            # add special x-axis
mtext("MA smoother: q = 7")            # indicate model
abline(h = 0, lty = 2)                 # (draws a dashed line at y = 0)
```

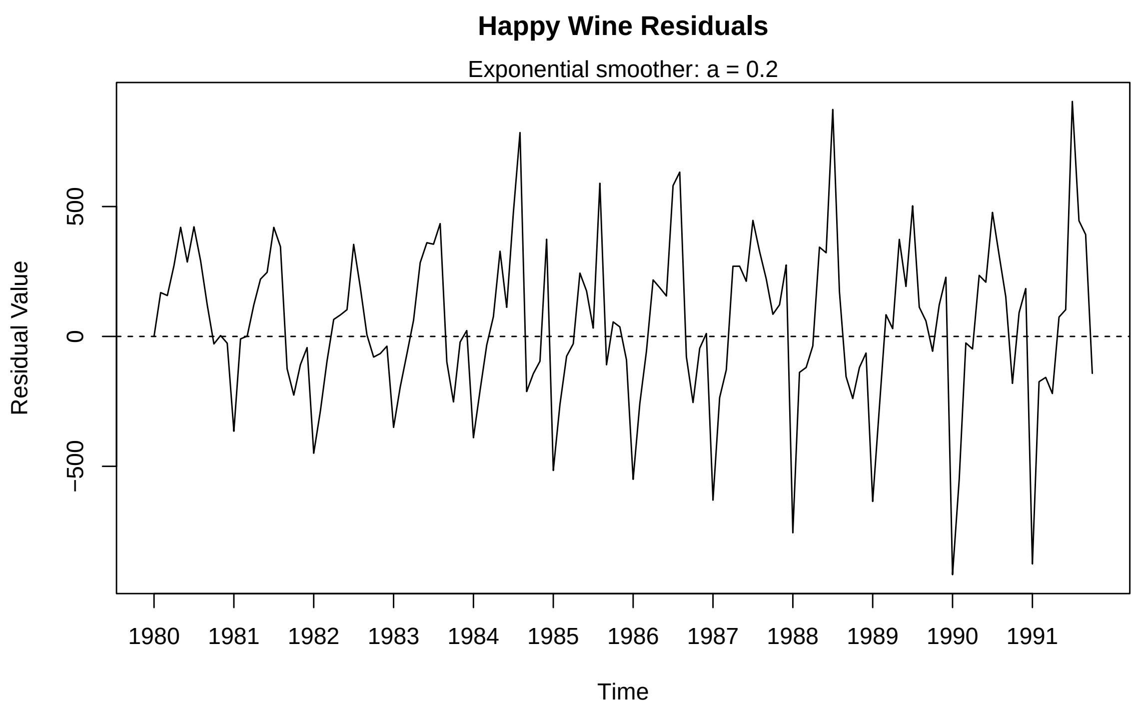


Exponential Smoothing

We now repeat the whole thing using an exponential smoother with parameter $\alpha = 0.2$

```
# --- Trend Estimation & Residuals
m.exp <- smooth.exp(wine, alpha = 0.2) # estimated m_t
rm.exp <- wine - m.exp                 # calculated residuals

# --- Plotting
par(mfrow = c(2,1)) # OPTIONAL matrix of plots: 2 rows, 1 column
                    # (2 plots stacked on top of each other in the PDF)
                    # (again, not important, but good to see as an example)

# Estimated m_t
plot.wine()                            # Initial plot, using our function
mtext("Exponential smoother: a = 0.2") # Subtitle indicating model used
lines(m.ma, col = "green3")            # Adds expo-smoother's m_t estimate

# Residuals from m_t
plot.ts(rm.exp,                        # Expo-smoother residuals!
        main = "Happy Wine Residuals",
        ylab = "Residual Value",
        xaxt = 'n')  # <- makes room for our special x-axis
axis.wine()                            # add special x-axis
mtext("Exponential smoother: a = 0.2") # indicate model
abline(h = 0, lty = 2)                 # (draws a dashed line at y = 0)
```


Happy Austrailian Red Wine Sales
Exponential smoother: a = 0.2
Volume Sold (kL)
500
1000
1500
2000
2500
3000
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
Time (in years)


Happy Wine Residuals
Exponential smoother: a = 0.2
Residual Value
−500
0
500
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
Time

Extract s_t : The Spring

We apply this not to the original data, but to the residuals we got when we removed the trend. Just like how we can't remove the spring from a pen without opening up the pen. Let's use the residuals from our polynomial regression estimate.

Harmonic Regression

We suspect there is a seasonal component of period $d = 12$. Let's model this, and plot it over our polynomial regression residuals. Are there other periods that might be relevant?

```
# --- Seasonal Component & Residuals
```

The S1 Method

Let's do the same thing using the `season()` function from ITSMR. Again, we choose $d = 12$.

```
# --- Seasonal Component & Residuals
```

Plotting

```
par(mfrow = c(2,1)) # Optional matrix of plots: 2 rows, 1 column

# --- Plotting
# Estimated s_t: Harmonic Regression
# plot.ts(rm.exp,
#         main = "Happy Wine | Exponential Smoother Residuals",
#         ylab = "Residual Value",
#         xaxt = 'n')
# axis.wine()
# mtext("Harmonic Regression:  d = ????")
# # add s_t!
#
# # Estimated s_t: S1 Method
# plot.ts(rm.exp,
#         main = "Happy Wine | Exponential Smoother Residuals",
#         ylab = "Residual Value",
#         xaxt = 'n')
# axis.wine()
# mtext("S1 method:  q = 12")
# # add s_t!
```

Examine Y_t : The Residual Ink

```
# # Residuals from s_t: harmonic regression
# plot.ts(????,
#         main = "Happy Wine Residuals | deseasonalized",
#         ylab = "Residual Value",
#         xaxt = 'n')
# axis.wine()
# mtext("Harmonic Regression:  d = ????")
# abline(h = 0, lty = 2)
#
# # Residuals from s_t: S1 method
# plot.ts(????,
#         main = "Happy Wine Residuals | deseasonalized",
#         ylab = "Residual Value",
#         xaxt = 'n')
# axis.wine()
# mtext("S1 method:  q = 12")
# abline(h = 0, lty = 2)
```

Autocorrelation

We will learn what this is on Wednesday. Basically, it describes the correlation between any two points in the series, as a function of the time-distance between those two points. Noise, by definition, has no such correlation across time.

We want our residuals $\hat{Y}_t$ to be *noisy.* We want our pen's ink to be *inky.*

```
# --- Plotting ACFs
# Residuals from s_t: harmonic regression
# Residuals from s_t: S1 method
```

Values exceeding the dashed blue lines indicate significant autocorrelation.

These ACFs suggest that the residuals left by the season fit are less correlated across time than those we obtained via harmonic regression. Thus the season fit is preferred.

Putting the pen back together

```
# Put the de-noised series back together, plot it on top of the original data
```