Lecture 7 (Week 8) Bread Prices

April 13, 2020

0.1 Bread Price notebook

0.1.1 Highlighting the following functions and diagnostic tools:

ACF

PACF

0.1.2 The three stages of iterative modelling:

1. Model selection / identification:
   - AR(p) / MA(q) / ARMA(p,q) / ARIMA(p,d,q)
   - What are the p's and q's?
2. Parameter Estimation:
   - If AR model, what are the ϕ_k's?
   - If MA model, what are the θ_i's?
   - If ARMA model, what are the ϕ_k's & θ_i's?
   - If ARIMA model, what differencing needs to be applied? What's d?

0.1.3 Loading in the data:

```
In [14]: price <- c(5.8, 6.1, 5.4, 6.2, 5.0, 4.6, 5.8, 5.1, 5.3, 5.1, 4.8, 5.3, 6.8, 9.0, 8.6,
         9.0, 7.4, 6.4, 4.8, 3.9, 3.9, 5.6, 5.7, 7.5, 7.3, 7.4, 7.5, 9.7, 6.1, 6.0, 5.7, 5.0,
         4.2, 4.6, 5.9, 5.4, 5.4, 5.4, 5.6, 7.6, 7.4, 5.4, 5.1, 6.9, 7.5, 5.9, 6.2, 5.6, 5.8,
         5.6, 6.6, 4.8, 5.2, 4.5, 4.4, 5.3, 5.0, 6.4, 7.8, 8.5, 5.6, 7.1, 7.1, 8.0, 7.3, 5.7,
         4.8, 4.3, 4.4, 5.7, 4.7, 4.1, 4.1, 4.7, 7.0, 8.7, 6.2, 5.9, 5.4, 6.3, 4.9, 5.5, 5.4,
         4.7, 4.1, 4.6, 4.8, 4.5, 4.7, 4.8, 5.4, 6.0, 5.1, 6.5, 6.2, 4.6, 4.5, 4.0, 4.1, 4.7,
         5.1, 5.2, 5.3, 4.8, 5.0, 6.2, 6.4, 4.7, 4.1, 3.9, 4.0, 4.9, 4.9, 4.8, 5.0, 4.9, 4.9,
         5.4, 5.6, 5.0, 4.5, 5.0, 7.2, 6.1)
```

```
In [15]: length(price)
```

124

0.1.4 Creating a timeseries (object) variable called BP.ts using the ts R function

```
In [16]: BP.ts <- ts(price, start=1634, frequency = 1)
```

0.1.5 Creating a shorter bread price timeseries

```
In [20]: BP.short.ts <- ts(price[1:(1+(1690-1634))],start=1634)
```

0.1.6 Summarising the timeseries data

```
In [22]: summary(BP.ts)
```

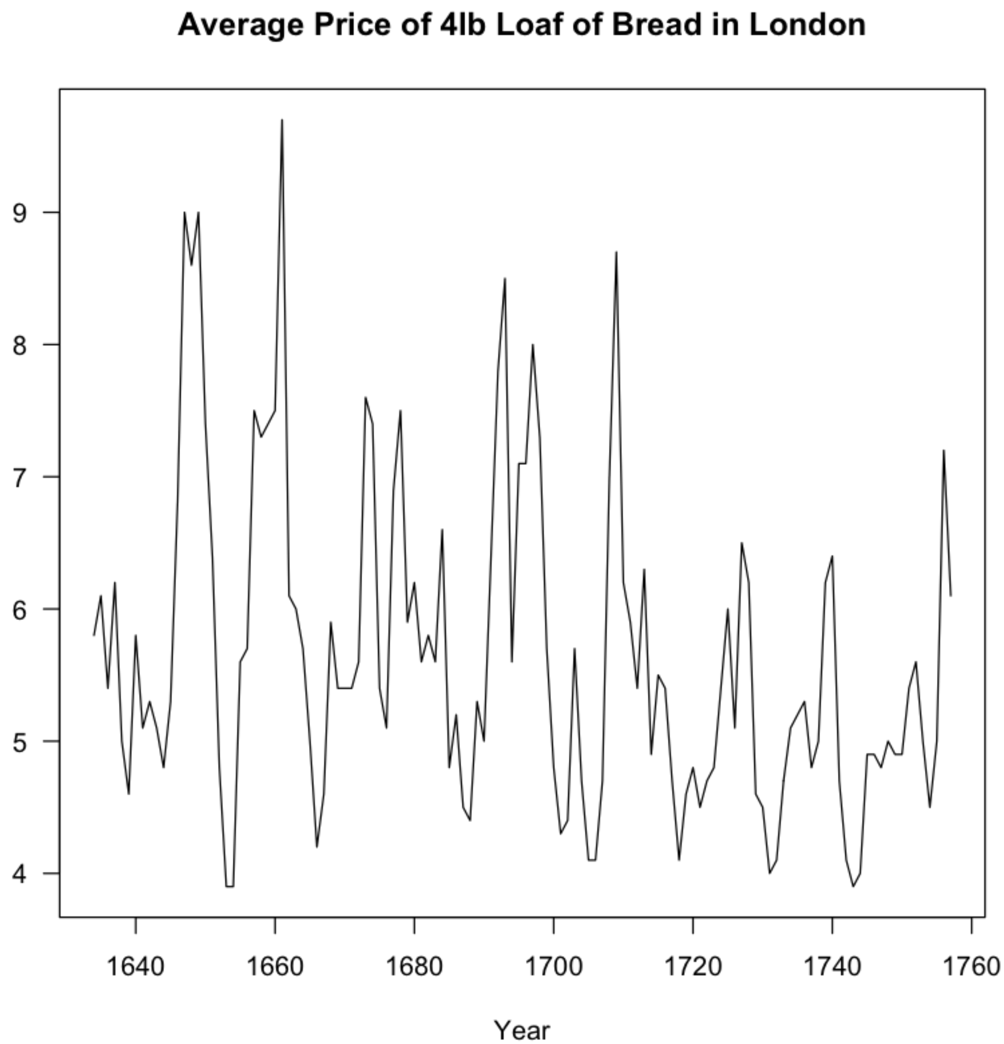
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.900   4.800   5.400   5.652   6.200   9.700
```

```
In [23]: summary(BP.short.ts)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.900   5.100   5.600   5.949   6.600   9.700
```

0.1.7 Plotting the data

```
In [24]: plot(BP.ts,xlab="Year",ylab="",
              main="Average Price of 4lb Loaf of Bread in London",las=1)
```



0.2 Plotting the ACF

Notice that we have a geometrically decreasing function and it's sinusoidal so we expect one of the model parameters to be negative

It looks like it's either an AR(p) process or an ARMA(p,q) process where p>0.

We're not really seeing any MA type behaviour though in the ACF - would expect to see more MA-esque signatures early doors, and then some kind of discontinuous cut off / step behaviour at q.

```
In [32]: T = length(BP.ts)
         T
```

124
There are 124 data points in the timeseries, therefore **T = 124**

Expect white noise process correlations, r_τ ~ (0, 1/T)$

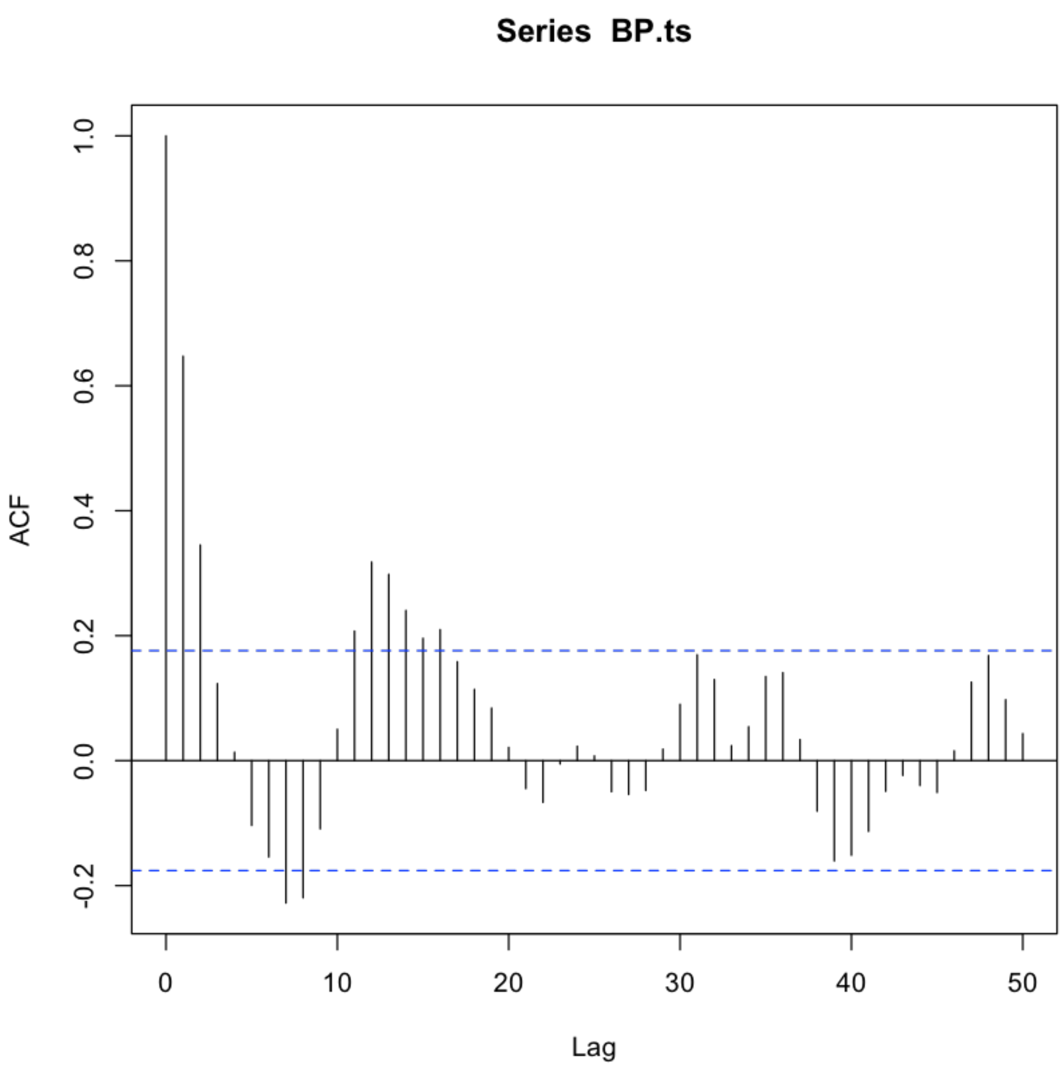
Therefore expect the white noise correlation sample standard deviation, $s = 1/\sqrt{T}$

And this the 95% probability limits to be roughly +/- 2s

In [37]:
```
s = 1/(T**0.5)
2*s
```

0.179605302026775
Therefore you roughly see the 95% confidence bands in the chart below being at **~0.18**

In [39]:
```
BP.acf <- acf(BP.ts, lag.max = 50)
```



```
In [42]: BP.acf[1]
```

```
Autocorrelations of series BP.ts, by lag

    1
0.647
```

0.3 Plotting the PACF

Notice that essentially we're in the noise after lag 1

Therefore we can suggest an **AR(1)** model.

We can read off what r_1 is from the chart or from the above series: where $r_1 = 0.647$

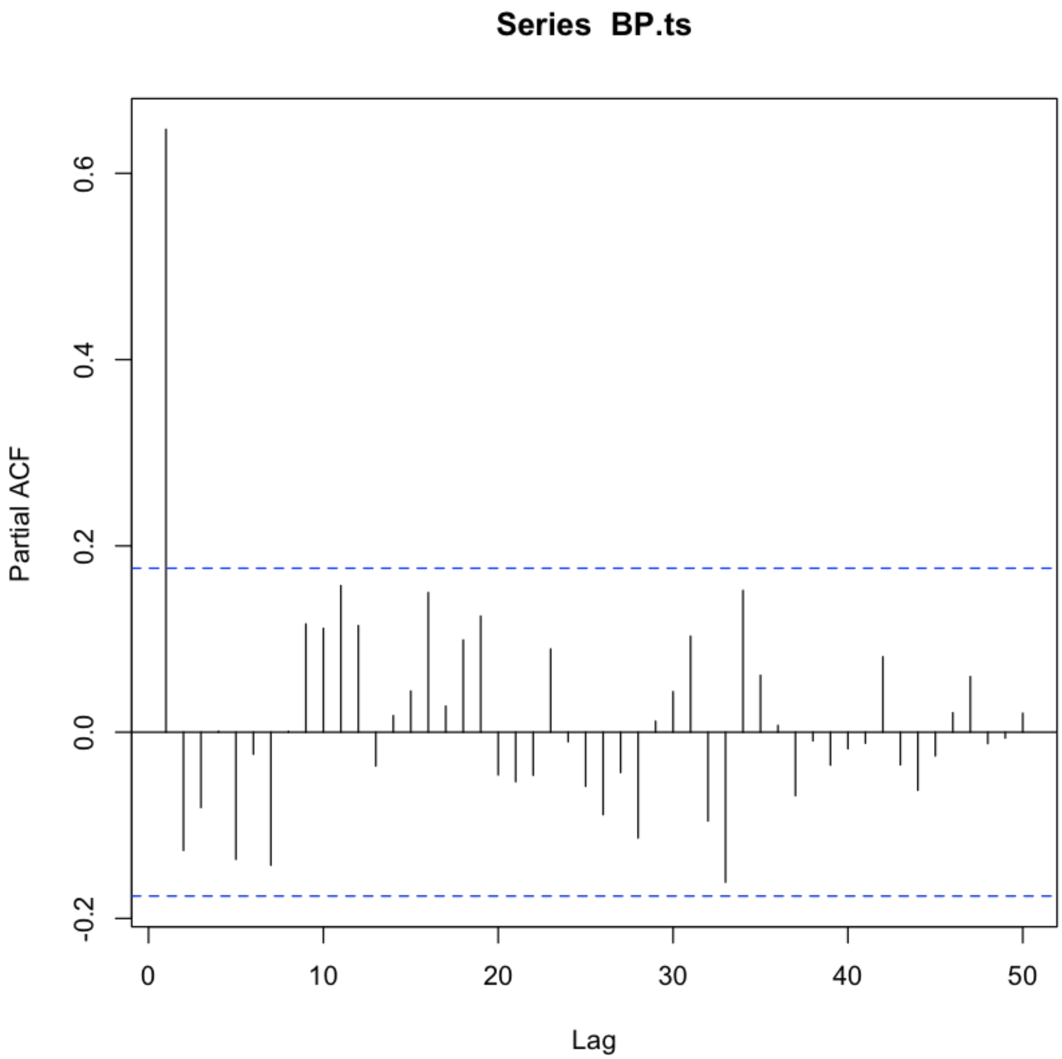
And we know from **AR(1)** that $\rho_\tau = \phi^\tau$, therefore $\rho_0 = 1$ and $\rho_1 = \phi_1 = \phi$

So $r_1 = \hat{\phi} = 0.647$

And our guess for the AR(1) model: $\hat{Y}_t = 0.647 Y_{t-1}$

Therefore our estimate for $\hat{\phi}_{11} = 0.647$

```
In [38]: BP.pacf <- pacf(BP.ts, lag.max = 50)
```



0.4 Fitting the model

Using R's ARIMA modelling function

Will fit **AR(1)** (expecting this to be the "best" interms of the combination of parsimony and fit, as well as, AR(2), MA(1), MA(2), ARMA(1,1), and ARMA(2,1).

0.4.1 AR(1)

```
In [43]: # fitting an ARIMA(1,0,0) model
         ##ăwhich is  the same as fitting an AR(1) model
         BP.ar.1 <- arima(BP.ts, order=c(1,0,0))
```

```
In [49]: #ăcan see we have a single coefficient - the phi_1 parameter - and it's standard erro
         ##ănote also the similarity between the above estimate of phi_1 = 0.647 and R's MLE e
         BP.ar.1
```

```
Call:
arima(x = BP.ts, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.6429     5.6608
s.e.  0.0678     0.2307

sigma^2 estimated as 0.8655:  log likelihood = -167.26,  aic = 340.52
```

0.4.2 AR(2)

```
In [66]: #ăfitting arima(2,0,0) model
         ## which is the same as an AR(2) model
         BP.ar.2 <- arima(BP.ts, order=c(2,0,0))

In [67]: # interesting, so it's saying that
         BP.ar.2
```

```
Call:
arima(x = BP.ts, order = c(2, 0, 0))

Coefficients:
         ar1      ar2  intercept
      0.7231  -0.1235     5.6546
s.e.  0.0888   0.0892     0.2051

sigma^2 estimated as 0.8521:  log likelihood = -166.31,  aic = 340.62
```

0.4.3 MA(1)

```
In [68]: BP.ma.1 <- arima(BP.ts, order=c(0,0,1))

In [70]: BP.ma.1
```

```
Call:
arima(x = BP.ts, order = c(0, 0, 1))

Coefficients:
         ma1  intercept
```

```
      0.5312     5.6495
s.e.  0.0577     0.1365

sigma^2 estimated as 0.991:  log likelihood = -175.55,  aic = 357.11
```

0.4.4 MA(2)

```
In [72]: BP.ma.2 <- arima(BP.ts, order=c(0,0,2))

In [73]: BP.ma.2
```

```
Call:
arima(x = BP.ts, order = c(0, 0, 2))

Coefficients:
         ma1     ma2  intercept
      0.7058  0.3379     5.6518
s.e.  0.0871  0.0737     0.1693

sigma^2 estimated as 0.8601:  log likelihood = -166.89,  aic = 341.78
```

0.4.5 ARMA(1,0,1)

```
In [74]: BP.arma.1.0.1 <- arima(BP.ts, order=c(1,0,1))

In [75]: BP.arma.1.0.1
```

```
Call:
arima(x = BP.ts, order = c(1, 0, 1))

Coefficients:
         ar1     ma1  intercept
      0.5598  0.1465     5.6560
s.e.  0.1046  0.1182     0.2139

sigma^2 estimated as 0.8552:  log likelihood = -166.53,  aic = 341.05
```

0.4.6 ARMA(2,0,1)

```
In [76]: BP.arma.2.0.1 <- arima(BP.ts, order=c(2,0,1))

In [77]: BP.arma.2.0.1
```

```
Call:
arima(x = BP.ts, order = c(2, 0, 1))

Coefficients:
         ar1      ar2      ma1  intercept
      1.3499  -0.5292  -0.6414     5.6558
s.e.  0.1916   0.1231   0.1994     0.1649

sigma^2 estimated as 0.8407:  log likelihood = -165.49,  aic = 340.99
```

---

1 Choosing the model

So have just fitted a bunch of models.

1. What do the different diagnostics mean?
2. What diagnostics should you use to select a model?
3. How to properly figure out the confidence limits?

   Let's have a look at the AR(1) summary, and discuss:

```
In [78]: BP.ar.1
```

```
Call:
arima(x = BP.ts, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.6429     5.6608
s.e.  0.0678     0.2307

sigma^2 estimated as 0.8655:  log likelihood = -167.26,  aic = 340.52
```

- There's a single coefficient - the ϕ coefficient of the AR(1) model: $Y_t = \phi Y_{t-1} + \epsilon_t$
- There's the standard error associated with the parameter estimate of ϕ
- sigma^2 is the estimate of the white noise variance
  - recall that $\epsilon_t \sim \text{NID}(0, \sigma^2)$
- where the mean of ϵ_t is **zero**

- and then you have the **AIC**!
  - **Before getting into the thick of it, the goal is to minimise the AIC when comparing models to select the best model**
  - But this doesn't take into account the parsimony that you may be after - you may be willing to give up a bit of AIC to get a more parsimonious model with greater transparency and fewer sources of (parameter) error.

2 AIC: the Akaike information criterion

- AIC assumes data are normally distributed -> **MEANING THE WHITE NOISE TERMS ARE NORMALLY DISTRIBUTED AS** NID(0, σ^2)
- AIC defined as:

  AIC = -2 * maximised log likelihood + 2n

  AIC $\approx T \ln \hat{\sigma}^2 + 2n + \text{constant}$

2.1 Recall: the goal is to MINIMIZE the AIC score for the model

n is the number of parameters fitted - so this is a penalty for adding more parameters

T is the size of the sample dataset

$\hat{\sigma}^2$ is the estimated white noise variance - if this is very low, then there isn't much variance around mean zero white noise, so you're systematic part of the equation - the $Y_t = \phi Y_{t-1}$ bit - isn't being impacted too much by the white noise.

So you want the white noise to be as low as possible, and the parameters to be as few as possible

They also scale the white noise variance by the number of data points in the model.

2.2 Choosing a model:

- **IF** competing models have the same number of parameters, then the one with the lowest estimated white noise variance will be deemed the best model, and have the lowest AIC score.

- AIC does penalise higher order models... But in practice AIC tends to favour higher order models (i.e. not penalising them high enough - as the parsimony factor is subjective rather than objective - so it's probably cautious on the subjective element)

- This means that AIC often selects higher order models, effectively overestimating the true order of the underlying ARIMA process.

```
In [ ]:
```