9.3 AR(1)

April 19, 2020

1 Lecture 9 (Final): Forecasting Using ARIMA Models

Will firstly use bread price data to forecast AR(1) process

And will then use air passenger data to forecast ARIMA(0,1,1)x(0,1,1)$_{12}$

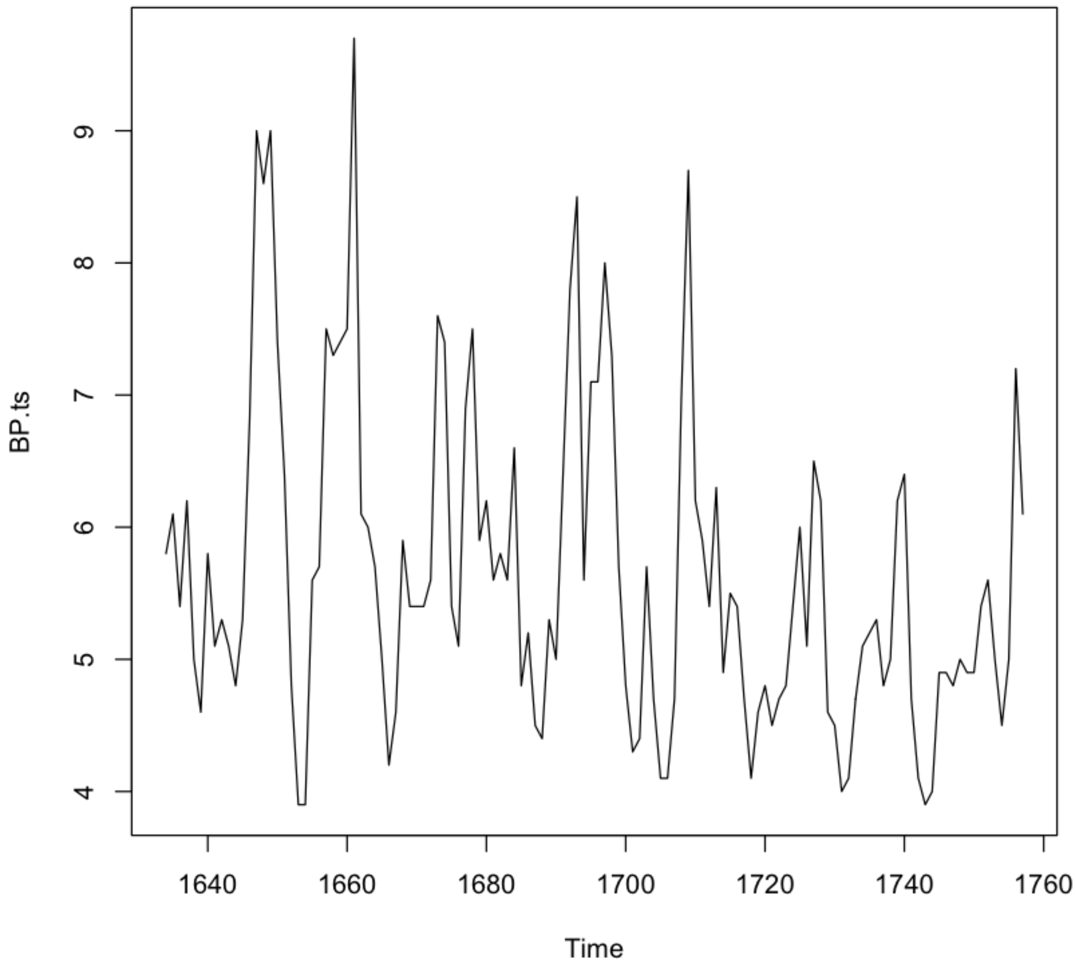
[1]:
```
price <- c(5.8, 6.1, 5.4, 6.2, 5.0, 4.6, 5.8, 5.1, 5.3, 5.1, 4.8, 5.3, 6.8, 9.
 ↪0, 8.6,
9.0, 7.4, 6.4, 4.8, 3.9, 3.9, 5.6, 5.7, 7.5, 7.3, 7.4, 7.5, 9.7, 6.1, 6.0, 5.7,␣
 ↪5.0,
4.2, 4.6, 5.9, 5.4, 5.4, 5.4, 5.6, 7.6, 7.4, 5.4, 5.1, 6.9, 7.5, 5.9, 6.2, 5.6,␣
 ↪5.8,
5.6, 6.6, 4.8, 5.2, 4.5, 4.4, 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)

BP.ts <- ts(price, start=1634, frequency = 1)
```

1.1 Quickly recall the fitting and diagnostics of these models, starting with the simple bread prices

Visual inspection, no major trend / predictable seasonality / increase in variance

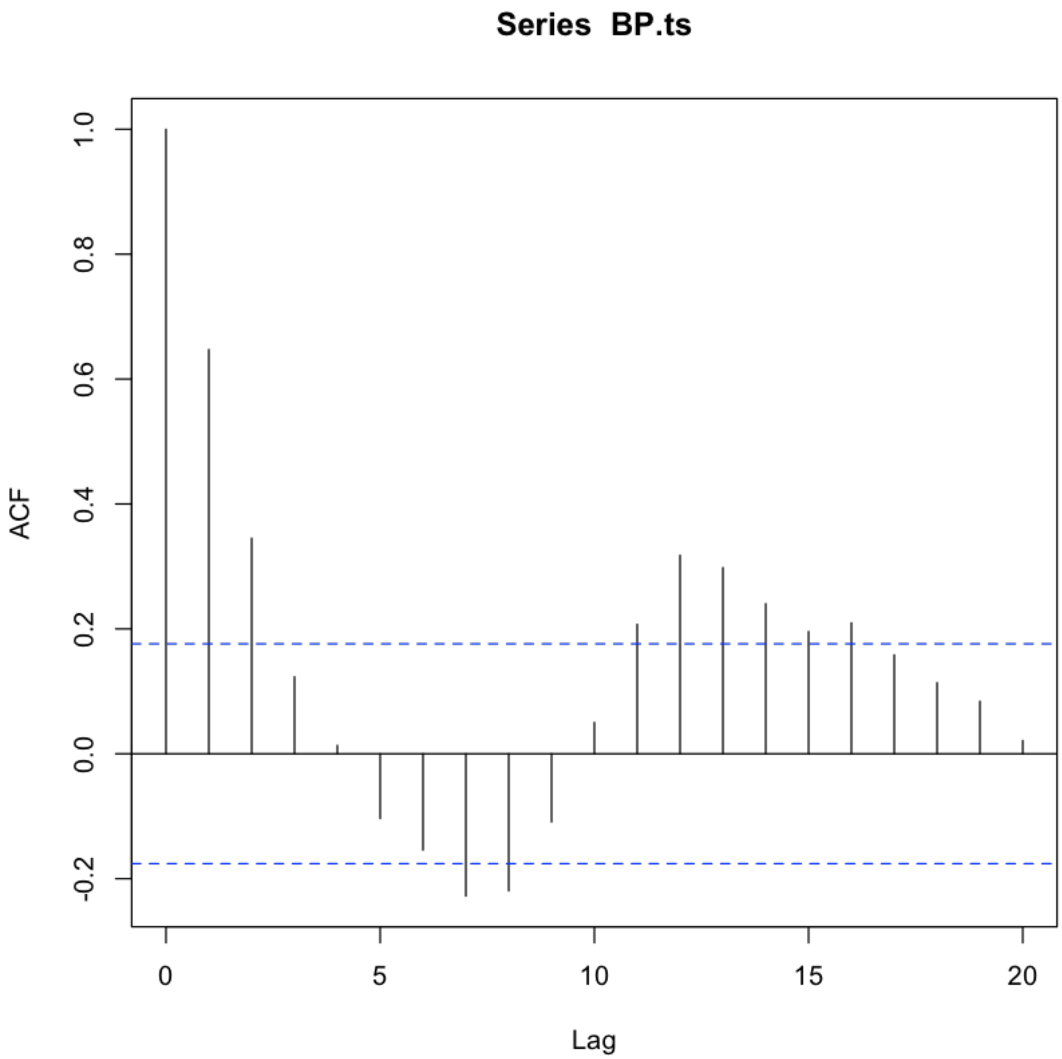
[2]:
```
plot(BP.ts)
```



1.2 ACF

Geometrically decreasing function, with negative ACF values. Suggests a AR process with a AR parameter that's negative.

[3]:
```
acf(BP.ts)
```

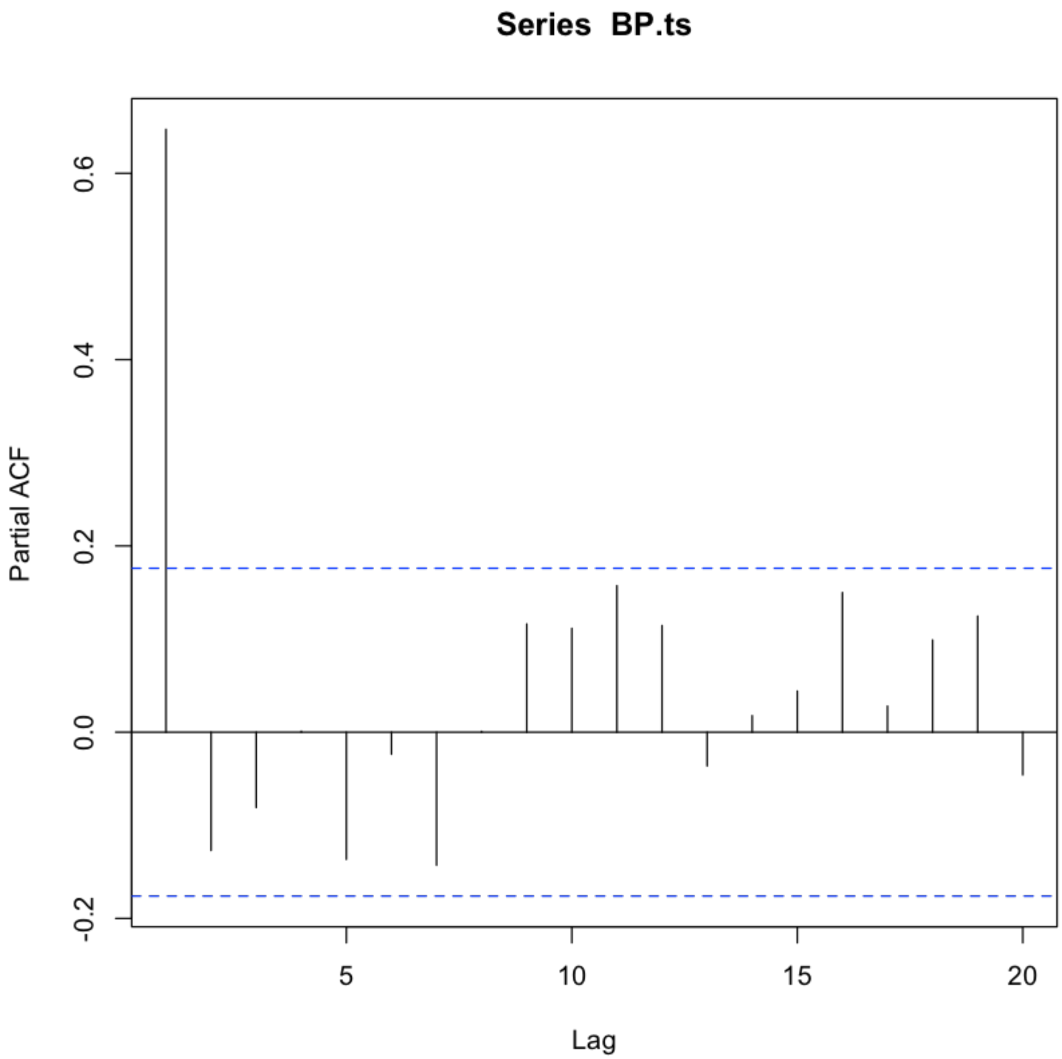


1.3 PACF

Nothing significant above the white noise 95% probability limits at ~0.18, this indicating that ideal p of AR(p) process is likely to be 1.

Let's fit both AR(1) and AR(2) and compare the AIC statistics, as well as the significance of the additional parameter.

[4]:
```
pacf(BP.ts)
```



1.4 AR(1)

[5]:
```
BP.ar.1 <- arima(BP.ts, order = c(1,0,0))

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
```

1.4.1 t-statistic of parameter

[6]:
```
ar.1.t <- 0.6429 / 0.0678
ar.1.t
```

9.48230088495575

1.4.2 degrees of freedom

[7]:
```
ar.1.df <- length(BP.ts) - 3
```

1.5 p-value associated with t-statistic

highly significant at the 1% significance level

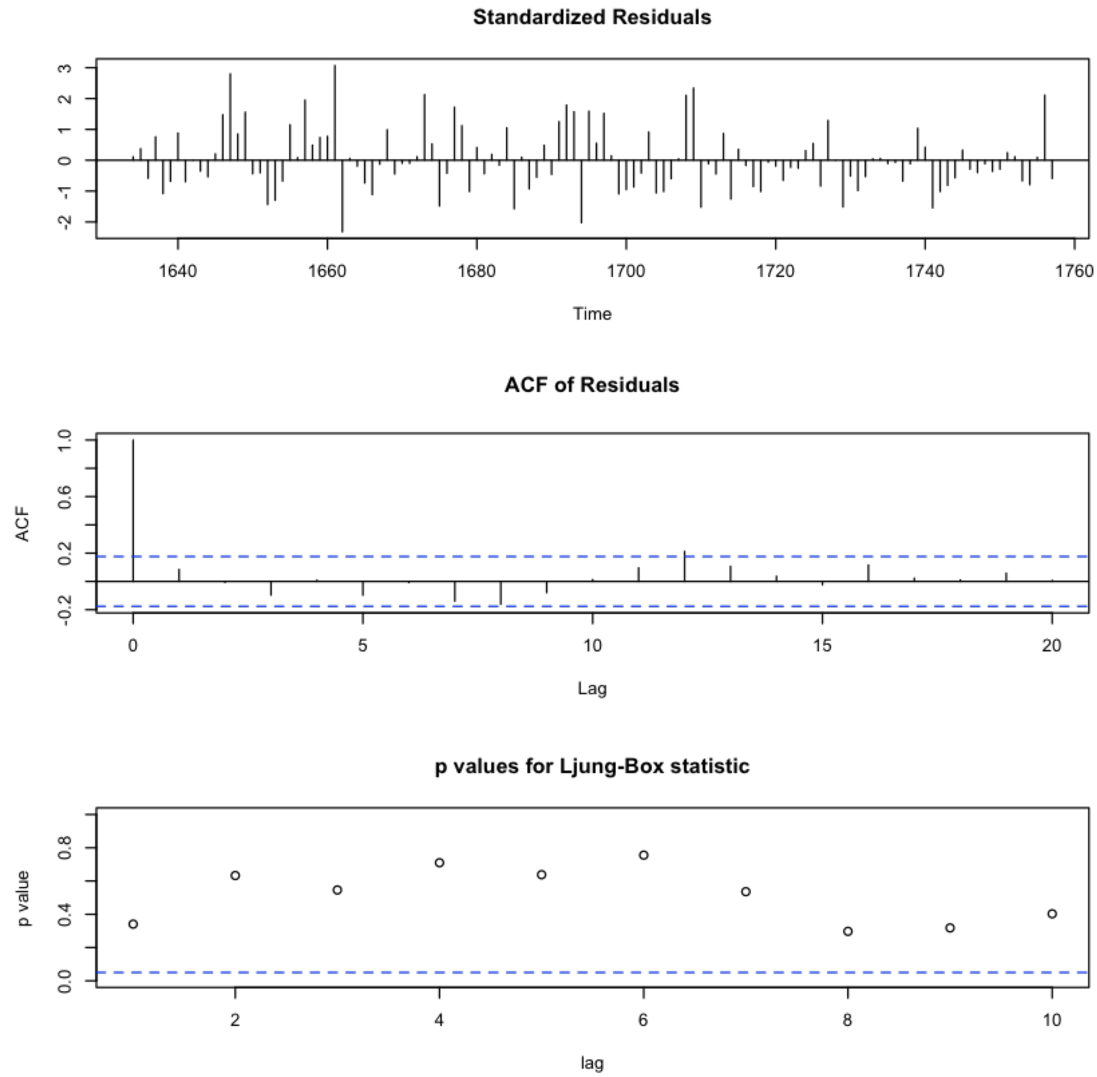
[8]:
```
# lower.tail = FALSE so that P(T > t) since our observed t is positive
# and due to the symmetry of the t-distribution, we do a two tailed test as␣
↪we're
# interested in equal or more extremeness at either end
2*pt(q = ar.1.t, ar.1.df, lower.tail = FALSE)
```

2.7560859751886e-16

and finally taking a quick look at the diagnostic plots

standardised residuals - the residuals divided by the estimated sigma^2 - the estimated variance of the underlying white noise terms

[9]:
```
tsdiag(BP.ar.1)
```



AR(2)

taking a look at AR(2) and checking whether the additional parameter is significant, and whether the AIC statistic is lower than AR(1).

[10]:
```
BP.ar.2 <- arima(BP.ts, order = c(2,0,0))

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
```

t-statistic of additional parameter

[11]:
```
ar.2.t <- -0.1235 / 0.0892

ar.2.t
```

-1.38452914798206

degrees of freedom of AR(2)

[12]:
```
ar.2.df <- length(BP.ts) - 4
```

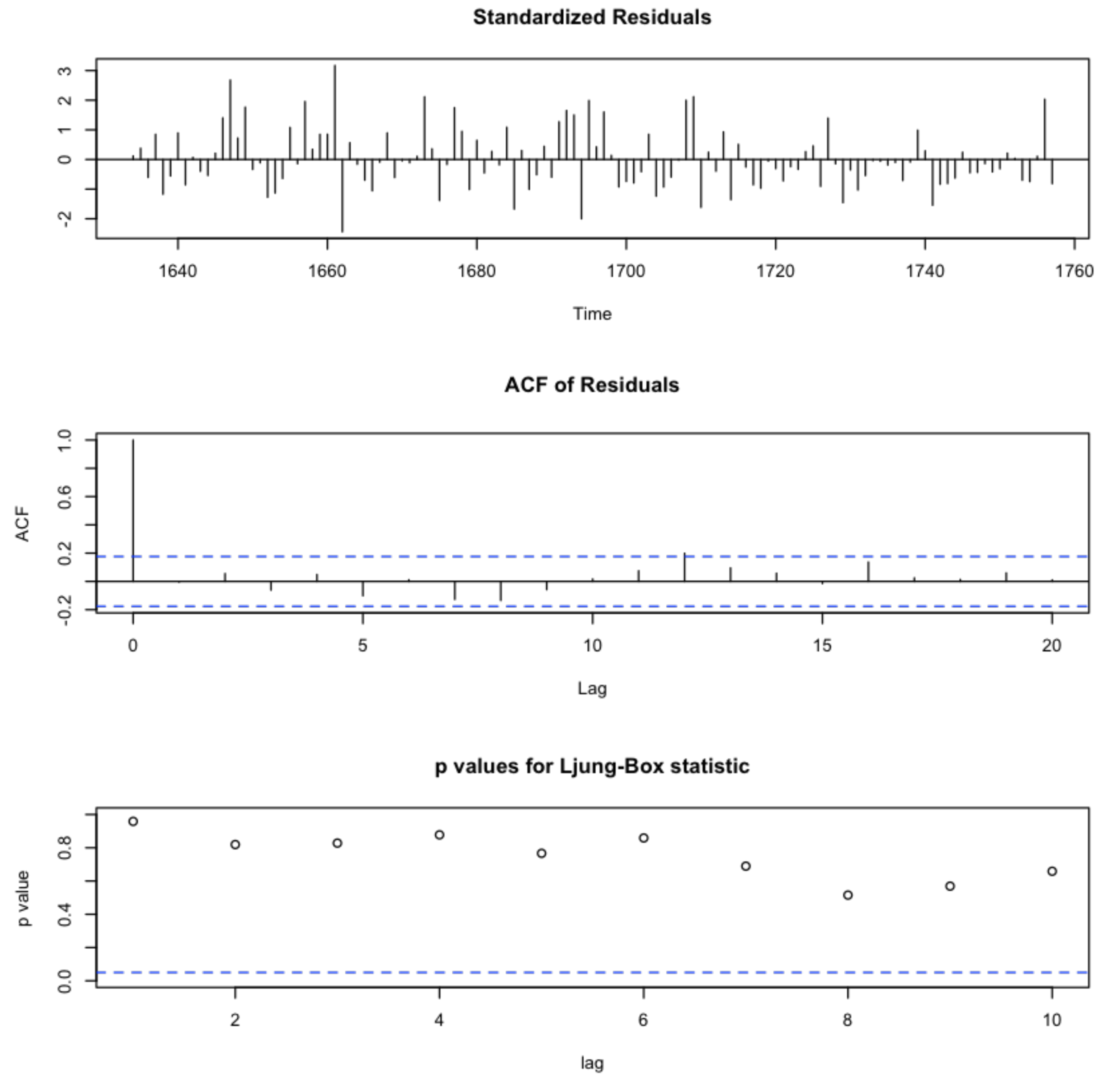
p-value associated with t-statistic for the second parameter of AR(2) model

not significant at the 10% level

[13]:
```
2*pt(ar.2.t, ar.2.df, lower.tail = TRUE)
```

0.168765584886517

[14]:
```
tsdiag(BP.ar.2)
```



1.5.1 so we accept the AR(1) model for modelling bread prices

---

2 9.3 Forecasting bread prices

2.1 AR(1)

let's quickly get the 95% probability limit t-statistic from a t-distribution with 121 degrees of freedom

[15]:
```
t <- qt(0.025, 121, lower.tail = FALSE)
t
```

1.97976376250537

[16]:
```
h <- 10
years <- 1958:(1958+h-1)
```

[17]:
```
BP.fore <- predict(BP.ar.1, 10)

BP.fore$pred
```

A Time Series:
1. 5.94318407878558 2. 5.84236141228889 3. 5.77753884835035 4. 5.73586206135817 5. 5.70906653726199 6. 5.69183871901093 7. 5.68076232727379 8. 5.67364091165826 9. 5.66906229421924 10. 5.66611853432868

[18]:
```
BP.fore$se
```

A Time Series:
1. 0.930329162677166 2. 1.10602296508448 3. 1.17097354119759 4. 1.19679272999601 5. 1.20730424927878 6. 1.21162272783268 7. 1.21340335568425 8. 1.21413864604819 9. 1.21444246094625 10. 1.21456802587075

[19]:
```
L95 <- BP.fore$pred - t*BP.fore$se
U95 <- BP.fore$pred + t*BP.fore$se
```

[20]:
```
BP.PL95 <- data.frame(years, L95, BP.fore$pred, U95)

BP.PL95
```

A data.frame: 10 × 4

years	L95	BP.fore.pred	U95
<int>	<ts>	<ts>	<ts>
1958	4.101352	5.943184	7.785016
1959	3.652697	5.842361	8.032026
1960	3.459288	5.777539	8.095790
1961	3.366495	5.735862	8.105229
1962	3.318889	5.709067	8.099244
1963	3.293112	5.691839	8.090565
1964	3.278510	5.680762	8.083014
1965	3.269933	5.673641	8.077349
1966	3.264753	5.669062	8.073371
1967	3.261561	5.666119	8.070676

2.2 Taking another look at the AR(1) summary

Note the intercept is the **estimate of the mean**, $\hat{\mu}$.

$\hat{\mu} = 5.6608$

$\hat{\phi} = 0.6429$

$\hat{\sigma}^2 = 0.8655$

[21]:
```
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
```

Last value of the time series in year 1757:

[22]:
```
BP.ts[length(BP.ts)]
```

6.1

Let's feed these numbers into the forecast equation we've produced in the notes:

$$\hat{y}(h) = \mu + \phi^h (y_T - \mu)$$

[23]:
```
mu.hat <- 5.6608

phi.hat <- 0.6429

sigma.2.hat <- 0.8655

y.t <- BP.ts[length(BP.ts)]

y.hat.1 <- mu.hat + phi.hat * (y.t - mu.hat)
y.hat.1
```

5.94316168

[24]:
```
# check that against the AR(1) R model forecast
BP.fore$pred[1]
```

5.94318407878558

2.2.1 And $V(1) = \mathbf{Var}(e_T(h)) = \sigma^2$

Therefore our estimate of $V(1) = \hat{\sigma}^2 = 0.8655$

And we'd expect the 95% confidence limit to be $\sim t_{0.025} \times \hat{\sigma}$

[25]:
```
y.hat.1 + c(-t*sqrt(sigma.2.hat),+t*sqrt(sigma.2.hat))
```

1. 4.10134285811832 2. 7.78498050188168

Comparing with R output

[26]:
```
BP.fore$pred[1] + c(-t*BP.fore$se[1],t*BP.fore$se[1])
```

1. 4.10135211531536 2. 7.78501604225579
