Introductory Time Series with R: Selected solutions from odd numbered exercises

Paul Cowpertwait & Andrew Metcalfe

August 5, 2009

Chapter 1 Solutions

1. The data can be read into R and the plots obtained using the following code. In the plots it will be seen that the chocolate production series exhibits an increasing trend – this will be particular clear in the plot of the aggregated series. In addition, the boxplot will indicate that production tends to reach a minimum in January (possibly following post-Christmas sales).

```
> www = "http://www.massey.ac.nz/~pscowper/ts/cbe.dat"
> cbe = read.table(www, head=T)
> choc.ts <- ts(cbe[,1], st=1958, fr=12)
> plot(choc.ts)
> plot(aggregate(choc.ts))
> boxplot(choc.ts ~ cycle(choc.ts))
```

3. Below the data are entered into R and `LI` and `PI` found.

```
> q0 <- c(0.33, 2000, 40, 3, 2)
> p0 <- c(18000, 0.8, 40, 80, 200)
> qt <- c(0.5, 1500, 20, 2, 1)
> pt <- c(20000, 1.6, 60, 120, 360)
> LI <- sum(q0 * pt)/sum(q0 * p0)
> PI <- sum(qt * pt)/sum(qt * p0)
> c(LI, PI)

[1] 1.358 1.250
```

(a) From the R code we see that $PI_t = 1.250$.

(b) LI_t uses the quantities from the base year, which is earlier than quantities used in PI_t. PI is usually less than LI because people tend to move away from items that show sharp price increases to substitutes that have not shown such steep price increases. In this case the cost of new cars has increased by a factor of 1.11, whereas the cost of servicing has increased by a factor of 1.50 and the cost of petrol has doubled. People have tended to buy more new cars thus reducing the costs of servicing and petrol consumption.

(c) The code below calculates the Irving-Fisher index.

```
> sqrt(LI * PI)
[1] 1.303
```

Chapter 2 Solutions

1. (a) The code below reads the data in and then produces a scatter plot and calculates the correlation between the x and y variables. The plot indicates an almost quadratic relationship, which is not reflected in the value of the correlation (since correlation is a measure of linear relationship).

```
> www = "http://www.massey.ac.nz/~pscowper/ts/varnish.dat"
> varnish = read.table(www, head=T)
> plot(varnish)
> cor(varnish)
           x          y
x  1.0000000 -0.2528782
y -0.2528782  1.0000000
```

(b) There is a clear pattern but it is non-linear resulting in small correlation.

```
> www = "http://www.massey.ac.nz/~pscowper/ts/guesswhat.dat"
> guesswhat = read.table(www, head=T)
> plot(guesswhat)
> cor(guesswhat)
           x          y
x 1.00000000 0.06457764
y 0.06457764 1.00000000
```

3. (a) The data can be read into R as follows and a plot of the decomposed series obtained from `plot(decompose())`.

```
> www = "http://www.massey.ac.nz/~pscowper/ts/global.dat"
> global = scan(www)
> global.ts = ts(global, st=1856, fr=12)
> global.decom = decompose(global.ts)
> plot(global.decom)
```

Since the data are 'global' temperature we would not expect to observe substantial seasonal variation. This is supported by the close standard deviations and a boxplot (from the code below).

```
> sd(global.ts)
[1] 0.2735360
> sd(global.ts - global.decom$seas)
[1] 0.2715033
> boxplot(global.ts ~ cycle(global.ts))
```

A plot of the trend with seasonal effect added is given by:

```
> ts.plot(global.decom$trend,
          global.decom$trend+global.decom$seas, lty=1:2)
```

(b) The residual series will have the first and last six values missing due to estimation of the trend by a moving average. Hence, the correlogram of the residuals is given by:

```
> length(global.decom$rand)
[1] 1800
> acf(global.decom$rand[-c(1:6,1795:1800)])
```

In the plot there is evidence of short-term positive serial correlation (lag 1) and periodic correlation (negative serial correlation at lag 5 and positive serial correlation at lag 10). This suggests global temperatures persist from one month to the next and include some cyclical trends different from the seasonal period.

Chapter 3 Solutions

1. (a) x and y are linearly related since substraction yields $y_t = x_t + \epsilon_t$ ($t = 1, \ldots, 100$), where ϵ_t is an $\mathrm{N}(0, \sigma^2)$ random variable, with $\sigma^2 = 2k^2$. Hence, we find high crosscorrelation which decreases as k increases.

   (b) The variables x and y both follow sine waves of period 37 time units, with x lagging behind y by 4 time units. The relationship between the variables can be seen using `plot(x,y)` which shows x

and y have a non-linear relationship scattered about an elliptical path.

3. (a) Differentiating $F(t)$ gives:

$$f(t) = \frac{(p+q)^2}{p}\left[1 + \frac{q}{p}e^{-(p+q)t}\right]^{-2} e^{-(p+q)t}$$

Dividing this by $1 - F(t)$ gives:

$$\frac{f(t)}{1-F(t)} = \frac{p+q}{1+\frac{q}{p}e^{-(p+q)t}} = p + qF(t)$$

as required.

(b) The required plot can be obtained in R using:

```
> F.logis = function(t) 1+exp(-pi*t/sqrt(3))
> T = seq(-3,3, length=1000)
> plot(pnorm(T), F.logis(T), type='l')
```

(c) Differentiating $f(t)$ gives:

$$f'(t) = \frac{(p+q)^3 e^{-(p+q)t}(qe^{-(p+q)t} - p)}{p^2(1+\frac{q}{p}e^{-(p+q)t})^3}$$

At the peak (maximum) $f'(t) = 0$, which occurs when $qe^{-(p+q)t} - p = 0$, and the result follows.

Chapter 4 Solutions

1. Using `w <- rexp(1000) - 1`, the correlogram for `w`, obtained from `acf(w)` indicates white noise, and the histogram show a positively skewed distribution.

3. (a) $x_t - \mu = \alpha(x_{t-1} - \mu) + w_t \Rightarrow x_t = (1-\alpha)\mu + \alpha x_{t-1} + w_t$ from which we obtain: $\alpha_0 = (1-\alpha)\mu$ and $\alpha_1 = \alpha$.

(b) An AR(2) model with non-zero mean can be written as either:
(i) $x_t - \mu = \alpha(x_{t-1} - \mu) + \beta(x_{t-2} - \mu) + w_t$ or
(ii) $x_t = \alpha_0 + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + w_t$. Expanding and rearranging (i) gives: $x_t = (1-\alpha-\beta)\mu + \alpha x_{t-1} + \beta x_{t-2} + w_t \Rightarrow \alpha_0 = (1-\alpha-\beta)\mu$, $\alpha_1 = \alpha$ and $\alpha_2 = \beta$.

Chapter 5 Solutions

1. (a)
```
> Time = 1:100
> w = rnorm(100, sd=25)
> z = w
> for (i in 2:100) z[i] = 0.5*z[i-1] + w[i]
> x = z
> x = 70 + 2*Time - 3*Time^2 + z
> plot(x, type='l')
```

   (b)
```
> x.lm = lm(x ~ Time + I(Time^2))
> coef(x.lm)
(Intercept)        Time   I(Time^2)
  70.369283    2.019250   -3.001847
```

   (c)
```
> confint(x.lm)
                2.5 %    97.5 %
(Intercept) 51.456393 89.282172
Time         1.154880  2.883620
I(Time^2)   -3.010138 -2.993555
```

   The above confidence intervals contains the underlying population values of 70, 2, and -3. These confidence intervals are narrower than they should be because the errors are positively correlated – see solution to part (e).

   (d) The correlogram of the residuals (obtained from the code below) shows a significant value at lag 1, due to the autocorrelation introduced using a simulated AR process of residual errors.

```
> acf(resid(x.lm))
```

   (e) The GLS fit has wider standard errors for the parameter estimates. This is most apparent in a confidence interval (not asked for but also shown below).

```
> x.gls = gls(x ~ Time + I(Time^2), cor=corAR1(0.5))
> summary(x.gls)
 ...
Coefficients:
               Value Std.Error   t-value p-value
(Intercept) 73.24947 16.736346    4.3767  0.0000
Time         1.86129  0.765383    2.4318  0.0169
I(Time^2)   -3.00019  0.007332 -409.1942  0.0000
 ...

> confint(x.gls)
```

```
                2.5 %     97.5 %
(Intercept) 40.4468388 106.052109
Time         0.3611645   3.361409
I(Time^2)   -3.0145624  -2.985822
```

3. (a) The data have an increasing variance. If the standard deviation is approximately proportional to the mean (constant coefficient of variation) the logarithms of the data will have a constant standard deviation (and hence constant variance). This makes modelling easier. Furthermore simulating the logarithm of a variable ensures that all the simulated values of the variable itself are positive.

   (b) The code is shown below, from which it can be seen that the best model (with smallest AIC) includes all explanatory variables.

```
> www = "http://www.massey.ac.nz/~pscowper/ts/cbe.dat"
> cbe = read.table(www, head=T)
> cbe[1:2,]
  choc beer elec
1 1451 96.3 1497
2 2037 84.4 1463

> attach(cbe)
> length(elec)
[1] 396
> length(elec)/12
[1] 33
> imth = rep(1:12,33)
> T1 = 1:396
> T2 = T1^2
> elec.lm = lm(log(elec)~T1+T2+factor(imth))
> step(elec.lm)
Start:  AIC=-2717.56
log(elec) ~ T1 + T2 + factor(imth)

               Df Sum of Sq      RSS      AIC
<none>                          0.39 -2717.56
- factor(imth) 11      2.49     2.88 -1944.07
- T2            1      2.56     2.95 -1914.39
- T1            1     20.40    20.78 -1141.12
...
   (Intercept)              T1              T2   factor(imth)2
     7.271e+00       7.960e-03      -6.883e-06      -1.991e-02
 factor(imth)3   factor(imth)4   factor(imth)5   factor(imth)6
     6.598e-02       3.288e-02       1.462e-01       1.777e-01
```

```
 factor(imth)7   factor(imth)8   factor(imth)9  factor(imth)10
     2.375e-01       1.994e-01       1.074e-01       9.044e-02
factor(imth)11  factor(imth)12
     4.278e-02       2.350e-02
```

(c)
```
> c1=cos(2*pi*T1/12); s1=sin(2*pi*T1/12)
> c2=cos(2*2*pi*T1/12); s2=sin(2*2*pi*T1/12)
> c3=cos(3*2*pi*T1/12); s3=sin(3*2*pi*T1/12)
> c4=cos(4*2*pi*T1/12); s4=sin(4*2*pi*T1/12)
> c5=cos(5*2*pi*T1/12); s5=sin(5*2*pi*T1/12)
> c6=cos(6*2*pi*T1/12)
> elec.harm = lm(log(elec)~T1+T2+c1+s1+c2+s2+c3+s3+c4+s4+c5+s5+c6)
> step(elec.harm)
...

Coefficients:
(Intercept)           T1           T2           c1           s1
  7.363e+00    7.961e-03   -6.883e-06   -8.840e-02   -5.803e-02
         c2           s2           s3           c5           s5
  1.496e-02    1.510e-02   -1.544e-02    1.003e-02    2.190e-02
         c6
 -7.985e-03
```

(d) The best fitting model is (marginally) the harmonic model with an AIC of -2722. The correlogram indicates there is still some seasonal effects present. The partial correlogram is difficult to interpret in the presence of existing seasonal effects. However, the high lag 1 term followed by low values suggest that an AR(1) process with seasonal terms for the error series would be worth trying.

```
> elec.best = lm(log(elec)~T1+T2+c1+s1+c2+s2+
+ s3+c5+s5+c6)
```

(e)
```
> ar(resid(elec.best))

Coefficients:
      1        2        3        4        5        6        7
 0.3812   0.2187   0.0853   0.0274  -0.0004   0.0350  -0.0241
      8        9       10       11       12       13       14
 0.0071   0.1094   0.0317   0.1208   0.1715  -0.0889  -0.0664
     15       16       17       18       19       20       21
-0.0458  -0.0896   0.0211   0.0038  -0.1251   0.0162  -0.0539
     22       23
-0.0161   0.1609

Order selected 23  sigma^2 estimated as  0.0003933
```

(f) The correlogram indicates the residuals of the fitted AR model are approximately white noise. Note the first 23 values had to be removed, since an AR(23) model had been fitted so the first 23 residuals were NA.

```
> fit.ar <- ar(resid(elec.best))
> acf(fit.ar$res[-(1:23)])
```

(g) The equation of the best fitted model is:

$$\begin{aligned}
\log(x_t) &= 7.36 + 7.96 \times 10^{-3} t - 6.88 \times 10^{-6} t^2 \\
&\quad -8.84 \times 10^{-2} \cos(2\pi t/12) - 5.80 \times 10^{-2} \sin(2\pi t/12) \\
&\quad +1.50 \times 10^{-2} \cos(4\pi t/12) + 1.51 \times 10^{-2} \sin(4\pi t/12) \\
&\quad -1.54 \times 10^{-2} \sin(6\pi t/12) + 1.00 \times 10^{-2} \cos(10\pi t/12) \\
&\quad +2.19 \times 10^{-2} \sin(10\pi t/12) - 7.99 \times 10^{-3} \cos(\pi t) + z_t
\end{aligned}$$

where x_t is the electricity production at time t ($t = 1, \ldots, 396$), z_t is the residual series, which follows an AR(23) process, and t is the time in months ($t = 1, \ldots, 396$).

(h) The predicted values are in `elec.pred`, and these have been adjusted using a correction factor of $\frac{1}{2}\sigma^2$ to account for the bias due to taking logs. Finally, the predicted values are added to the time plot of the original series. When viewing this plot it is evident that the predictions are not particularly good, since they fail to follow the trends, and that better predictions would probably be obtained using a multiplicative Holt-Winters procedure.

```
> new.t = 397:(397+119)
> new.c1 = cos(2*pi*new.t/12); new.s1 = sin(2*pi*new.t/12)
> new.c2 = cos(2*2*pi*new.t/12); new.s2 = sin(2*2*pi*new.t/12)
> new.c3 = cos(3*2*pi*new.t/12); new.s3 = sin(3*2*pi*new.t/12)
> new.c5 = cos(5*2*pi*new.t/12); new.s5 = sin(5*2*pi*new.t/12)
> new.c6 = cos(6*2*pi*new.t/12);
> new.t2 = new.t^2
> new.dat = data.frame(T1=new.t, T2=new.t2, c1=new.c1, s1=new.s1,
+ c2=new.c2, s2=new.s2, s3=new.s3, c5=new.c5, s5=new.s5, c6=new.c6)
> ar.pred = predict(ar(resid(elec.best)), n.ahead=120)
> log.pred = predict(elec.best, new.dat)
> elec.pred = exp(log.pred + ar.pred$pred + 0.5*0.0003933)
> elec.ts = ts(elec, st=1958, fr=12)
> elec.pred.ts = ts(elec.pred, st=1991, fr=12)
> ts.plot(elec.ts, elec.pred.ts, lty=1:2)
```

Chapter 6 Solutions

1.

$$\begin{aligned} x_t = \sum_{i=0}^{q} \beta_i w_{t-i} \Rightarrow \gamma(k) &= \operatorname{cov}\left(\sum_{i=0}^{q} \beta_i w_{t-i}, \sum_{j=0}^{q} \beta_j w_{t+k-j}\right) \\ &= \sum_{i=0}^{q}\sum_{j=0}^{q} \beta_i \beta_j \operatorname{cov}\left(w_{t-i}, w_{t+k-j}\right) \\ &= \sum_{i=j-k}^{q} \beta_i \beta_j \sigma^2 \\ &= \sigma^2 \sum_{i=0}^{q-k} \beta_i \beta_{i+k} \end{aligned}$$

from which the result follows using $\rho(k) = \gamma(k)/\gamma(0)$.

3. (a) Rearranging and expressing in terms of the backward shift operator B gives:

$$\begin{aligned} (1 - B + \frac{1}{4}B^2)x_t &= (1 + \frac{1}{2}B)w_t \\ \Rightarrow (1 - \frac{1}{2}B)^2 x_t &= (1 + \frac{1}{2}B)w_t \end{aligned}$$

which is ARMA(2, 1) and is both stationary and invertible since the roots of the equations in B all exceed unity in absolute value.

(b)

$$\begin{aligned} x_t &= 2x_{t-1} - x_{t-2} + w_t \\ \Rightarrow (1 - 2B + B^2)x_t &= w_t \\ \Rightarrow (1 - B)^2 x_t &= w_t \end{aligned}$$

This is ARMA(2,0) and is non-stationary since $B = 1$ is a solution to the characteristic equation taken from the left-hand-side of the equation. The model is invertible, since the MA part on the right-hand-side is just white noise. The model would usually be expressed as ARIMA(0,2,0).

(c) Rearranging and expressing in terms of B gives:

$$\begin{aligned}(1-\frac{3}{2}B+\frac{1}{2}B^2)x_t &= (1-\frac{1}{2}B+\frac{1}{4}B^2)w_t \\ \Rightarrow (1-B)(1-\frac{1}{2}B)x_t &= (1-\frac{1}{2}B+\frac{1}{4}B^2)w_t\end{aligned}$$

This is ARMA(2,2) and is non-stationary since the characteristic equation taken from the left-hand-side has a unit root $B = 1$. The model would usually be expressed as ARIMA(1,1,2). The model is invertible, since the complex roots of the right-hand-side equation exceed unity in absolute value:

```
> polyroot(c(1,-1/2,1/4))
[1] 1+1.732051i 1-1.732051i
> Mod(polyroot(c(1,-1/2,1/4)))
[1] 2 2
```

Chapter 7 Solutions

1. (a) Rearranging in terms of B gives:

$$\begin{aligned}(1-B+0.25B^2)z_t &= (1+0.5B)w_t \\ \Rightarrow (1-0.5B)^2 z_t &= (1+0.5B)w_t\end{aligned}$$

which is an ARIMA(2,0,1) or ARMA(2,1) model, and is stationary because the roots of the polynomial on the left hand side, $B = 2$, exceed unity.

3. (a) Using the equation $x_t = a + bt + w_t$ gives:

$$\begin{aligned}y_t &= \nabla x_t \\ &= x_t - x_{t-1} \\ &= a + bt + w_t - \{a + b(t-1) + w_{t-1}\} \\ &= b + w_t - w_{t-1} \\ \Rightarrow x_0 + \sum_{i=1}^{t} y_i &= x_0 + \sum_{i=1}^{t}\{b + w_i - w_{i-1}\} \\ &= x_0 + bt + w_t = x_t\end{aligned}$$

as required (taking $w_0 = 0$). Note $x_0 = a$.

(b) Back-substituting for x_{t-1}:

$$\begin{aligned} x_t &= x_{t-1} + b + w_t + \beta w_{t-1} \\ &= x_{t-2} + b + w_{t-1} + \beta w_{t-2} + b + w_t + \beta w_{t-1} \\ &= x_{t-2} + 2b + w_t + (1+\beta) w_{t-1} + \beta w_{t-2} \\ &= x_{t-3} + 3b + w_t + (1+\beta) w_{t-1} + (1+\beta) w_{t-2} + \beta w_{t-3} \\ &\vdots \\ &= x_0 + bt + w_t + (1+\beta) \sum_{i=1}^{t-1} w_i \end{aligned}$$

This has variance:

$$\text{Var}(x_t) = \sigma_w^2 \left\{1 + (1+\beta)^2 (t-1)\right\}$$

which increases as t increases, unless $\beta = -1$ in which case the variance is σ_w^2.

Chapter 8 Solutions

1. (a)
```
> www <- "http://www.massey.ac.nz/~pscowper/ts/LAN.txt"
> x <- read.table(www, head=T)
> attach(x)
> hist(bits)
> boxplot(bits)
```

(b)
```
> mean((bits-mean(bits))^3)/sd(bits)^3
[1] 2.888421
> mean((bits-mean(bits))^4)/sd(bits)^4
[1] 11.27845
```

(c)
```
> logbit <- log(1+bits)
> boxplot(logbit)
> hist(logbit)
> mean((logbit-mean(logbit))^3)/sd(logbit)^3
[1] -1.063624
> mean((logbit-mean(logbit))^4)/sd(logbit)^4
[1] 3.413031
```

(d) The fractionally differenced series can be found using the code on p162, which can be pasted into R from via the book website (www.massey.ac.nz/ pscowper/ts/scripts.R).

```
> x <- bits
> fds.fit <- fracdiff(x, nar=0, nma=0)
> n <- length(x)
> L <- 30
> d <- fds.fit$d
> fdc <- d
> fdc[1] <- fdc
> for (k in 2:L) fdc[k] <- fdc[k-1] * (d+1-k) / k
> y <- rep(0, L)
> for (i in (L+1):n) {
+     csm <- x[i]
+     for (j in 1:L) csm <- csm + ((-1)^j) * fdc[j] * x[i-j]
+     y[i] <- csm
+   }
> y <- y[(L+1):n]
> y.ar <- ar(y)
> y.res <- y.ar$res[-c(1:y.ar$order)]
> boxplot(y.res)
> hist(y.res)
```

(e) A range of models can be fitted and tested using the `arima` function nested within `AIC` as shown below.

```
> AIC(arima(y, order=c(1,0,1)))
[1] 70358.06
> AIC(arima(y, order=c(2,0,2)))
[1] 70243.97
> AIC(arima(y, order=c(5,0,5)))
[1] 70198.84
> AIC(arima(y, order=c(6,0,6)))
[1] 70201.07
> AIC(arima(y, order=c(6,0,5)))
[1] 70199.11
> AIC(arima(y, order=c(4,0,5)))
[1] 70207.49
> AIC(arima(y, order=c(5,0,4)))
[1] 70199.53
```

A best fitting model is ARMA(5, 5). This is better than the fitted AR model because the residual variance (adjusted by degrees of freedom) is smaller.

```
> y.ar
...
Order selected 22  sigma^2 estimated as  2781676
```

```
> y.arma
...
sigma^2 estimated as 2780911  ...
```

(f)

```
> x <- logbits
> fds.fit <- fracdiff(x, nar=2, nma=0)
> n <- length(x)
> L <- 30
> d <- fds.fit$d
> fdc <- d
> fdc[1] <- fdc
> for (k in 2:L) fdc[k] <- fdc[k-1] * (d+1-k) / k
> y <- rep(0, L)
> for (i in (L+1):n) {
+     csm <- x[i]
+     for (j in 1:L) csm <- csm + ((-1)^j) * fdc[j] * x[i-j]
+     y[i] <- csm
+   }
> y <- y[(L+1):n]
>
> y.ar <- ar(y)
> y.ar$order
[1] 26
> plot(y, type='l')
> acf(y)
> acf(y.ar$res[-c(1:26)])
> acf(y.ar$res[-c(1:26)]^2)
```

Chapter 9 Solutions

1. (a)

```
> TIME <- 1:128
> c1 <- cos(2*pi*TIME/128)
> s1 <- sin(2*pi*TIME/128)
> c2 <- cos(4*pi*TIME/128)
> var(c1)
[1] 0.503937
> var(s1)
[1] 0.503937
> var(c2)
[1] 0.503937
> cor(c1,s1)
[1] 3.020275e-18
> cor(c1,c2)
[1] -9.62365e-17
```

```
> cor(s1,c2)
[1] -3.823912e-17
```

(b) From the above code, we can see that the harmonic terms are uncorrelated and have variance approximately 1/2. This approximation improves for higher n, e.g. for $n = 10000$ we have:

```
> var(cos(2*pi*(1:10000)/10000))
[1] 0.50005
> var(cos(pi*(1:10000)/10000))
[1] 0.50005
```

Also, note that $\mathrm{Var}\{\cos(\pi t)\}$ is approximately 1:

```
> var(cos(pi*(1:10000)))
[1] 1.0001
```

Hence, from p173, and using $a_m^2 + b_m^2 = A_m^2$ and $b_{n/2} = 0$, we have:

$$\begin{aligned}\mathrm{Var}(x_t) &= \frac{1}{2}a_1^2 + \frac{1}{2}b_1^2 + \ldots + \frac{1}{2}a_{n/2-1}^2 + \frac{1}{2}b_{n/2-1}^2 + a_{n/2}^2 \\ &= \frac{1}{2}A_1^2 + \ldots + \frac{1}{2}A_{n/2-1}^2 + A_{n/2}^2\end{aligned}$$

which is equivalent to Parseval's Theorem.

(c) This follows from the fact that $b_{n/2} = 0$ and $\mathrm{Var}\{\cos(\pi t)\} = 1$ (as mentioned above).

Chapter 12 Solutions

1. (a) Increasing the variance of w_t means the filter follows the series more closely, because a higher variance enables greater adaptation of the parameter θ at each time step.

   (b) Increasing both variances – the parameter variance to 10 and the observation variance to 200 – produces an almost identical result to the initial result shown in the text (Fig. 12.1) which had variances 0.1 and 2 for w_t and v_t (respectively). Note the *ratios* of the observation and parameter variances are identical in both cases which is the reason for the results being so similar. The slight discrepancy is due to initial conditions.

3. (a) After the data are entered into R (into a vector `morg.dat`), the variances can be found as follows.

```
> length(morg.dat)
[1] 64
> morg.var <- vector(len=13)
> morg.var[1] <- var(morg.dat[1:5])
> morg.var[2] <- var(morg.dat[6:10])
> morg.var[3] <- var(morg.dat[11:14]) # 3rd week has 4 trading days
> k = 10 # an initial first index value
> for (i in 4:13) {
     k = k + 5
     morg.var[i] <- var(morg.dat[k:(k+4)])
   }
> morg.var
 [1]  1.090530  1.414130  0.560625  5.231030 18.970800  2.206530  1.722170
 [8] 28.039030  2.530550  2.224150  1.996750  2.056080  1.407570
 > mean(morg.var)
[1] 5.342303
```

(b)

```
> morg.mean <- vector(len=13)
> morg.mean[1] <- mean(morg.dat[1:5])
> morg.mean[2] <- mean(morg.dat[6:10])
> morg.mean[3] <- mean(morg.dat[11:14])
> k = 10
> for (i in 4:13) {
     k <- k + 5
     morg.mean[i] <- mean(morg.dat[k:(k+4)])
   }
> morg.mean
 [1] 38.2040 39.4560 41.2925 39.7060 26.4800 26.3460 23.1080 16.0160
 [9] 19.2200 18.7800 15.4500 17.0660 13.1680
```

(c)

```
> morg.mean.var <- var(morg.mean)
> morg.var.mean <- mean(morg.var)
> morg.mean.var - morg.var.mean / 5
[1] 108.3423
```

(d)

```
> week <- vector(len=64)
> week[1:5] <- 1
> week[6:10] <- 2
> week[11:14] <- 3
> k <- 10
> wk <- 3
> for (i in 4:13) {
     k <- k + 5
     wk <- wk + 1
     week[k:(k+4)] <- wk
   }
> anova(aov(morg.dat ~ factor(week)))
```

```
Analysis of Variance Table

Response: morg.dat
             Df Sum Sq Mean Sq F value    Pr(>F)
factor(week) 12 6318.2   526.5  96.856 < 2.2e-16 ***
Residuals    51  277.2     5.4
---
```

Allowing for the Labor Day holiday gives a within-week variance of 5.4.