

Chapter 4 Solutions

4.2.1. Exercises

Exercise 4.1

- (a) $x_t^* = \nabla x_t$ gives 0.5, 0.5, 0.5, 0.5, 0.1, -0.9, -1.2, 0.2
- (b) $x_t^{**} = \nabla_4 x_t = (1 - B^4)x_t = 2, 1.6, 0.2, -1.5, -1.8$ starting in line 5.
- (c) $x_t^* = \nabla^2 x_t = (1 - B)^2 x_t$ gives 0, 0, -0.4, -1, -0.3, 1.4 starting in line 4.
- (d) $y_t = \nabla^4 x_t = (1 - B)^4 x_t = (1 - B)^2(1 - B)^2 x_t$ gives 0, -0.4, -0.2, 1.3, 1
- (e) $w_t = \nabla_4 x_9 = (1 - B^4)x_9$ gives $y_9 - y_5 = -2.2$
- (f) $z_t = \nabla(\nabla^4 x_t) = \nabla y_t$ gives -0.4, 0.2, 1.5, -0.3

□

4.2.6. Exercises

Exercise 4.2

Let x_t be a monthly time series and $y_t = \log(x_t)$. Equation 4.1 says:

$$y^* = \nabla_k^D \nabla^d y_t = (1 - B^k)^D (1 - B)^d y_t$$

where y_t^* is the differenced series.

According to the problem, in backshift polynomial form, if the seasonality was monthly, that expression would give us a time series y_t^* equal to the following difference operator and backshift operator operations, respectively:

$$y_t^* = \nabla_{12}^2 \nabla^2 y_t = (1 - B^{12})^2 (1 - B)^2 y_t,$$

If x_t had been quarterly, and $y_t = \log(x_t)$, with quarterly seasonality, we would have an expression for y_t^* in difference operator and backshift operator operations, respectively, equal to:

$$y_t^* = \nabla_4^2 \nabla^2 y_t = (1 - B^4)^2 (1 - B)^2 y_t$$

□

Exercise 4.3

The following R code with comments indicates what we did in R. After that code, we write the final differencing and backshift expression.

```
x=AirPassengers ## already a ts() object
plot.ts(x, main="Upward linear-like trend and variance increasing\n
proportional to the trend (at least in appearance)")

## We try to see if log will stabilize the variance.
plot.ts(log(x))
## We check square root transformation to see if it is better.
plot.ts(sqrt(x)) # not so good
plot.ts(x^(3/4))
# best transformation is log.
y=log(x)

### We check the ACF to see if differencing is needed
acf(y, lag=50, main="Long memory. Needs differencing.
Autocorrelations \n are significant up to 42 months past")
# Try different differencing operations
par(mfrow=c(3,1)) # will put in one graph to compare
y.regdiff=diff(y, lag=1, diff=1); acf(y.regdiff, lag=50) # could be
y.seasdiff=diff(y, lag=12, diff=1); acf(y.seasdiff, lag=50)
y.seas.reg=diff(y.regdiff, lag=12, diff=1); acf(y.seas.reg, lag=50)
# It appears that seasonal difference of the regular difference is the best.
```

Based on what we see after we run the R code, if $x_t = \text{AirPassengers}$, then we must take log to stabilize the variance, obtaining $y = \log(x)$.

To make y stationary, we first take regular difference, and then we do seasonal differencing to obtain y_t^* . Since AirPassengers is a monthly time series, the expression in terms of Equation 4.1 is

$$y^* = \nabla_{12}^1 \nabla y_t = (1 - B^{12})(1 - B)y_t.$$

□

4.5. Problems

Problem 4.1

Column 4: $\nabla y_t^* = (1 - B)y_t = (14.4064 - 15, 14.9383 - 14.4064, 16.0374 - 14.9383, 15.6320 - 16.0374, 14.3975 - 15.6320)$

Column 5: $y_t^{**} = \nabla^2 y_t = \nabla y_t^* = (1 - B)^2 y_t = (0.5319 + 0.5936, 1.0991 - 0.5319, -0.4054 - 1.0991, -1.2345 + 0.4054)$

Column 7: $y_t^{***} = (14.9383 - 15, 16.0374 - 14.4064, 15.6320 - 14.9383, 14.3975 - 16.0374)$

The following R program will give all the columns of the table. Just need to pay attention to respect the lag and put the numbers in the right row, as in the table.

Completed Table 4.2

Time	y_t	$y_{t-1} = By_t$	$y_t^* = \nabla y_t$	$y_t^{**} = \nabla^2 y_t = \nabla y_t^*$	$y_{t-2} = B^2 y_t$	$y_t^{***} = y_t - y_{t-2} = \nabla_2 y_t$
2001	15					
2002	14.4064	15	-0.5936			
2003	14.9383	14.4064	0.5319	1.1255	15	-0.0617
2004	16.0374	14.9383	1.0991	0.5672	14.4064	1.631
2005	15.6320	16.0374	-0.4054	-1.5045	14.9383	0.6937
2006	14.3975	15.6320	-1.2345	-0.8291	16.0374	-1.6399

```

y.t=ts(c(15, 14.4064, 14.9383,16.0374, 15.6320,14.3975), start=c(2001),freq=1)
y.t

# column 3 in the Table

By.t = lag(y.t,k=-1) # lag function-be careful.
By.t
By.t.4.table=window(By.t, end=c(2006))
By.t.4.table

# Column 4
diff.1= diff(y.t,lag=1,differences=1)
diff.1

## Column 5
diff.2=diff(y.t, lag=1, differences=2)
diff.2

## Column 6
B2y.t=lag(y.t,k=-2)
B2y.t
B2y.t.4.table=window(B2y.t, end=c(2006))
B2y.t.4.table

## Column 7
diff.3=diff(y.t, lag=2, differences=1)
diff.3
    
```

□

Problem 4.2

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + W_t - \beta W_{t-2}$$

Completed table 4.3

DATE	HOUSTNSA (y_t)	y_{t-12}	$\nabla_{12}y_t$
1959-01-01	96.2		
1959-02-01	99.0		
1959-03-01	127.7		
1959-04-01	150.8		
1959-05-01	152.5		
1959-06-01	147.8		
1959-07-01	148.1		
1959-08-01	138.2		
1959-09-01	136.4		
1959-10-01	120.0		
1959-11-01	104.7		
1959-12-01	95.6		
1960-01-01	86.0	96.2	-10.2
1960-02-01	90.7	99.0	-8.3
1960-03-01	90.5	127.7	-37.2
1960-04-01	123.0	150.8	-27.8
1960-05-01	130.2	152.5	-22.3
1960-06-01	122.8	147.8	-25
1960-07-01	114.3	148.1	-33.8
1960-08-01	130.3	138.2	-7.9

□

Problem 4.3

HOUSTNSA is an interesting data set. We can see in the time plot (after you run the code given next) that there is a change in patterns of the time series. Whereas before 1985 there is cyclical behavior, business cycles, after 1985 the recurrent cycles disappear. We only observe a large drop in 2010. We observe seasonality all across the time interval considered. This observation is more obviously seen when we seasonally differenced.

After seasonally differencing, the business cycles before 1985 are still there.

Since there are those business cycles before 1985, we difference the data as well.

(a) The results can be seen in Completed Table 4.3. $\nabla_{12}y_t = y_t - y_{t-12}$. Subtract nonempty rows of column 4 from column 3.

(b) Updated code supporting our conclusions follows:

```
houstnsa=read.csv("HOUSTNSA.csv",header=T)
houstnsa[1:20,]
y=ts(houstnsa[,2],start=c(1959,1),
end=c(2019, 11),frequency=12)

## plot of the time series and differenced
par(mfrow=c(3,1))
plot.ts(y, main="Raw time series (ts) shows change
in pattern around 1990")
```

```
y.1star=diff(y,lag=12,differences=1) # seasonal diff of ts
plot.ts(y.1star, main="Seasonally
differenced ts preserves cycles before 1990")
# conclude that perhaps regular difference makes sense
y.1stardiff=diff(y.1star,lag=1,diff=1) #reg diff of seasonal diff of ts
plot.ts(y.1stardiff, main="Reg diff of seasonal diff confirms
changing variability")
```

HOUSTNSA is an interesting data set. We can see in the time plot (after you run the code given next) that there is a change in patterns of the time series. Whereas before 1990 there is cyclical behavior, business cycles, after 1985 the recurrent cycles disappear. We only observe a large drop in 2010. We observe seasonality all across the time interval considered. This observation is more obviously seen when we seasonally differenced.

After seasonally differencing, the business cycles before 1990 are still there.

Since there are those business cycles before 1985, we difference the data as well.

- (c) After seeing the regular difference of the seasonal difference of the time series, we notice that there is more variability before 1990. So we take the log. The stand alone program to do that is given next.

```
houstnsa=read.csv("HOUSTNSA.csv",header=T)
houstnsa[1:20,]
y=ts(houstnsa[,2],start=c(1959,1),
end=c(2019, 11),frequency=12)

y.log=log(y); class(y.log)
## plot of the logged time series and differenced
par(mfrow=c(3,1))
plot.ts(y.log, main="Raw logged time series shows change
in pattern around 1990")
y.1star=diff(y.log,lag=12,differences=1) # seas diff of log ts
plot.ts(y.1star, main="Seasonally
differenced logged ts preserves cycles before 1990")
# conclude that perhaps regular difference makes sense
y.1stardiff=diff(y.1star,lag=1,diff=1) #reg diff of seasonal diff of log ts
plot.ts(y.1stardiff, main="Reg diff of seasonal diff of log ts \n confirms changing variability")
# this code below is for the next question
dev.off() # type this before doing the next line.

### The remaining code is for the next two questions.
## part (d)
par(mfrow=c(2,1)) # will compare acf after diff of log and acf
# of mult decomp of raw.
## WE do first the acf of the random term obtained after differencing
acf(y.1stardiff, main="ACF of the differenced data, lag=50)
## part (e)
## we now do the acf of the random term of multiplicative decomposition
plot(decompose(y, type="mult"))
random=decompose(y, type="mult")$random
acf(random, main="ACF of random term of mult decomposition",
na.action=na.exclude, lag=50)
```

We notice in the plots of the regular difference of the seasonal difference of the logged time series that the variability has become more about the same across the whole time interval considered, albeit with a burst of volatility in 1980. We conclude that the logged data has similar variability across the time interval.

- (d) We do the correlogram of the regular difference of the seasonal difference of the logged time series. The time series after differencing is the random term. We notice that the ACF reveals that the random term is stationary.

See code posted in the question below this one.

- (e) For comparison, we also do multiplicative decomposition of the time series and look at the random component of the decomposition and look at the acf of the random term of the multiplicative decomposition, comparing both. Notice that we do multiplicative decomposition of the raw (without log) time series, because by doing multiplicative decomposition we are implying that there is changing variability, and hopefully the multiplicative decomposition will account for that. The random term of the multiplicative decomposition looks different than the one after differencing. It seems that differencing has stabilized the variable more than the multiplicative decomposition.

In addition to that, we notice that the ACF of the random term of the multiplicative decomposition indicates some nonstationarity in the data still.

The reader is recommended to try the following and compare. Do the additive decomposition of the logged data and compare with the two acf that we already obtained. Draw conclusions.

□

4.6 Quiz

Question 4.1

$$\nabla^3 y_t = (1 + 3B^2 - 3B - B^3)y_t$$

□

Question 4.3

Yes, because the ACF of the ddAP shows that ddAP is stationary. (It does not show significant autocorrelations up to very large k) We obtain the acf as follows, after entering the above code in R and executing it.

```
acf(ddAP, lag=50, cognellcogcmain="ACF of seasonal difference of regular \n
difference of AirPassengers reveals stationary")
```

However, notice that although the ACF shows that the ddAP is stationary, we observe in a time plot that the ddAP time series has much more variability at the end of the time series, which suggests that we should have logged the time series to make it variance stationary (logged or some other transformation). The time plot can be seen as follows:

```
plot.ts(ddAP, main="Time plot of seasonal difference of regular difference \n
of AirPassengers reveals variance nonstationarity")
```

□

Question 4.5

None of the above. Although it appears as if the first difference was done, it is not done with the y_t and it is not done as $(1 - B)y_t = y_t - y_{t-1}$.

□

Question 4.7

The new time series is

$$(1 - B^2)^2 y_t$$

□

Question 4.9

```
data=scan("viscosity.txt")
head(data)
y=ts(data)
acf(y, lag=50, main="ACF of viscosity reveals stationarity")
```

No, because the ACF shows that the time series is stationary in mean (which is equivalent to saying that it has constant mean across time). There are significant autocorrelations only at lags $k=1$ and $k=3$, and the autocorrelations die down quickly.

Notice that if we do not use the argument `freq=` in the `ts()` function, the lags in the horizontal axis of the ACF indicate exactly lag=1, 2, ...etc. If we had put used that argument, the lag=1 would have indicated the first cycle, and within that cycle we would have several lags not labeled. See the difference with the acf in Question 4.10.

□