Homework 4

STATS 207 (Time Series Analysis)

Fall 2020

Name :
Chih-Hsuan "Carolyn" Kao
Collaborated with
Xiaoye Yuan and Jiachen Ge

1. (5 points) Exercise 4.5. A time series was generated by first drawing the white noise series w_t from a normal distribution with mean zero and variance one. The observed series x_t was generated from

$$x_t = w_t - \theta w_{t-1}, \quad t = 0, \pm 1, \pm 2, \ldots,$$

where θ is a parameter. Derive a formula for the power spectrum of x_t, expressed in terms of θ and ω.

$$\gamma(0) = Var(W_t - \theta W_{t-1}) = 1 + \theta^2$$

$$\gamma(1) = cov(X_{t+1}, X_t) = -\theta$$

$$\gamma(-1) = cov(X_{t-1}, X_t) = -\theta$$

$$\gamma(h) = 0 \qquad \text{for } |h| > 1$$

Hence,

$$f(\omega) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \omega h}$$

$$= -\theta e^{2\pi i \omega} + 1 + \theta^2 - \theta e^{-2\pi i \omega}$$

$$= 1 + \theta^2 - \theta(\cos 2\pi\omega + \cos(-2\pi\omega) + \cancel{i\sin 2\pi\omega} + \cancel{i\sin(-2\pi\omega)})$$

$$= \underbrace{1 - 2\theta\cos 2\pi\omega + \theta^2}_{\text{This is the power spectrum}}$$

2. (5 points) Exercise 4.28. Determine the theoretical power spectrum of the series formed by combining the white noise series w_t (mean zero variance one) to form

$$y_t = w_{t-2} + 4w_{t-1} + 6w_t + 4w_{t+1} + w_{t+2}.$$

$$\gamma(h) = \text{cov}(y_{t+h}, y_t)$$
$$= \text{cov}[w_{t+h-2} + 4w_{t+h-1} + 6w_{t+h} + 4w_{t+h+1} + w_{t+h+2},\ w_{t-2} + 4w_{t-1} + 6w_t + 4w_{t+1} + w_{t+2}]$$

$$= \begin{cases} 70 & \text{when } h=0 \\ 56 & \text{when } h=\pm 1 \\ 28 & \text{when } h=\pm 2 \\ 8 & \text{when } h=\pm 3 \\ 1 & \text{when } h=\pm 4 \\ 0 & \text{otherwise} \end{cases}$$

To express the power spectrum in theoretical form,

$$f(\omega) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \omega h}$$
$$= (e^{8\pi i\omega} + e^{-8\pi i \omega}) + 8(e^{6\pi i\omega} + e^{-6\pi i\omega}) + 28(e^{4\pi i\omega} + e^{-4\pi i\omega}) + 56(e^{2\pi i\omega} + e^{-2\pi i\omega}) + 70$$

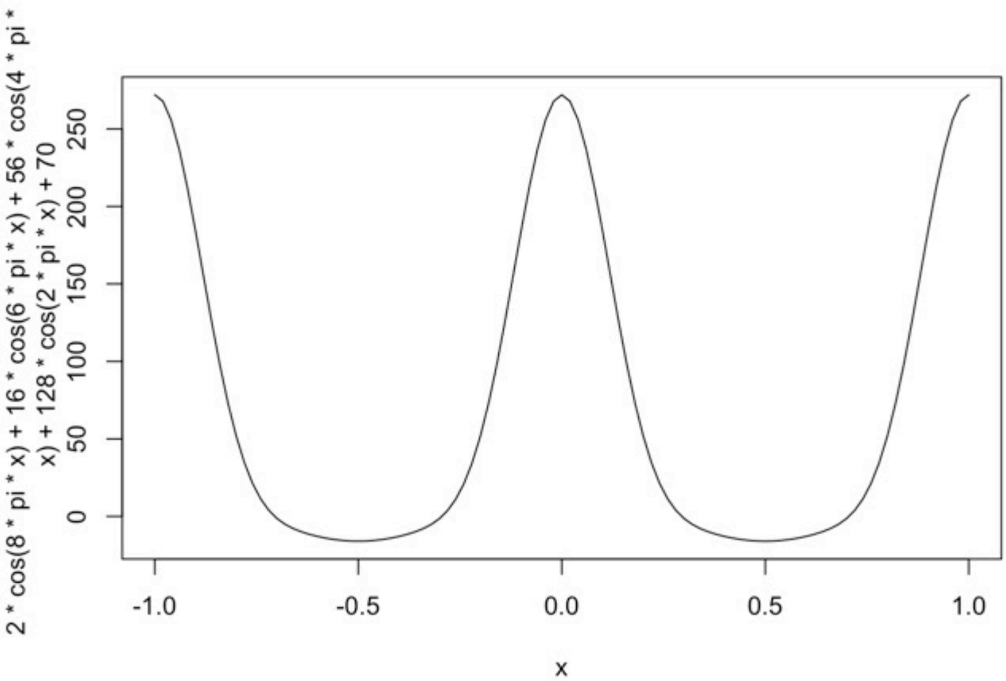
$$= 2\cos 8\pi\omega + 16\cos 6\pi\omega + 56\cos 4\pi\omega + 128\cos 2\pi\omega + 70$$

Plot the theoretical power spectrum. What is the 'color' of (y_t)?

Q2

Plot the theoretical power spectrum. What is the color of (y_t)?

```
curve(2*cos(8*pi*x) + 16*cos(6*pi*x) + 56*cos(4*pi*x) + 128*cos(2*pi*x) + 70, from=-1, to=1)
```



According to our lecture slides L11 page 26, some images that illustrate pink noise reaches peak at omega = 0 and descends symmetrically on both side, and that what we got from the image above. Hence, the color of (y_t) is pink.

3. (20 points) Exercise 4.25 and 4.26. Consider two processes

$$x_t = w_t \quad \text{and} \quad y_t = \phi x_{t-D} + v_t,$$

where w_t and v_t are independent white noise processes with common variance σ^2, ϕ is a constant, and D is a fixed integer delay.

(a) (5 points) Compute the coherency between x_t and y_t.

$$\gamma_{yx}(h) = \operatorname{cov}(y_{t+h}, x_t)$$
$$= \operatorname{cov}(\phi x_{t+h-D} + v_{t+h}, x_t)$$
$$= \phi \operatorname{cov}(x_{t+h-D}, x_t) + \operatorname{cov}(v_{t+h}, x_t)$$
$$= \phi \operatorname{cov}(w_{t+h-D}, w_t) + \operatorname{cov}(v_{t+h}, w_t)$$

As w_t and v_t are independent white noise

$$= \phi \operatorname{cov}(w_{t+h-D}, w_t)$$
$$= \begin{cases} \phi\sigma^2 & \text{when } h = D \\ 0 & \text{when } h \neq D \end{cases}$$

$$\gamma_x(h) = \operatorname{cov}(x_{t+h}, x_t)$$
$$= \operatorname{cov}(w_{t+h}, w_t) = \begin{cases} \sigma^2 & \text{when } h = 0 \\ 0 & \text{when } h \neq 0 \end{cases}$$

$$\gamma_y(h) = \operatorname{cov}(y_{t+h}, y_t)$$
$$= \operatorname{cov}(\phi x_{t+h-D} + v_{t+h}, \phi x_{t-D} + v_t)$$
$$= \operatorname{cov}(\phi w_{t+h-D} + v_{t+h}, \phi w_{t-D} + v_t)$$
$$= \phi^2 \operatorname{cov}(w_{t+h-D}, w_{t-D}) + \operatorname{cov}(v_{t+h}, v_t)$$
$$= \begin{cases} \phi^2\sigma^2 + \sigma^2 & \text{when } h = 0 \\ 0 & \text{when } h \neq 0 \end{cases}$$

Then, we can formulate

$$f_{yx}(w) = \sum_{h=-\infty}^{\infty} \gamma_{yx}(h) e^{-2\pi i w h}$$
$$= \phi \sigma^2 e^{-2\pi i w D}$$

$$f_x(\omega) = \sum_{h=-\infty}^{\infty} \gamma_x(h) e^{-2\pi i \omega h}$$

$$= \sigma^2$$

$$f_y(\omega) = \sum_{h=-\infty}^{\infty} \gamma_y(h) e^{-2\pi i \omega h}$$

$$= \phi^2 \sigma^2 + \sigma^2$$

$$= \sigma^2(\phi^2 + 1)$$

$$|f_{yx}(\omega)|^2 = \phi^2 \sigma^4 \left| \cos(-2\pi \omega D) + i \sin(-2\pi \omega D) \right|^2$$

$$= \phi^2 \sigma^4$$

Finally,

$$\rho_{xy}(\omega) = \sqrt{\frac{|f_{yx}(\omega)|^2}{f_y(\omega) f_x(\omega)}} = \frac{|\phi| \sigma^2}{\sigma^2 \sqrt{1+\phi^2}}$$

$$= \frac{|\phi|}{\sqrt{1+\phi^2}} \quad \#$$

(b) (5 points) Simulate $n = 1024$ normal observations from x_t and y_t for $\phi = 0.9$, $\sigma^2 = 1$, and $D = 0$. Then estimate and plot the coherency between the simulated series for the following values of L and comment: (i) $L = 1$, (ii) $L = 3$, (iii) $L = 41$, (iv) $L = 101$.

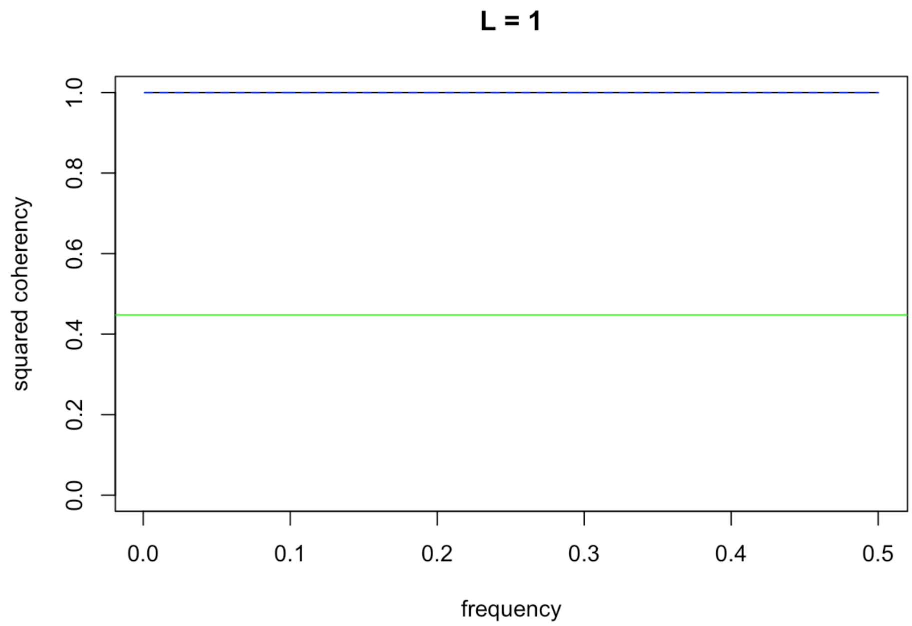
Q3(b)

```
set.seed(2020)
phi = 0.9
x = rnorm(n = 1024, 0, 1)
v = rnorm(n = 1024, 0, 1)
y = phi * x + v
s1 = spec.pgram(ts(cbind(x,y)), taper = 0, plot = F)
s2 = spectrum(ts(cbind(x,y)), span = 3, taper = 0, plot = F)
s3 = spectrum(ts(cbind(x,y)), span = 41, taper = 0, plot = F)
s4 = spectrum(ts(cbind(x,y)), span = 101, taper = 0, plot = F)
```

```
rho = function(phi){
  phi**2/(1+phi**2)
}
```

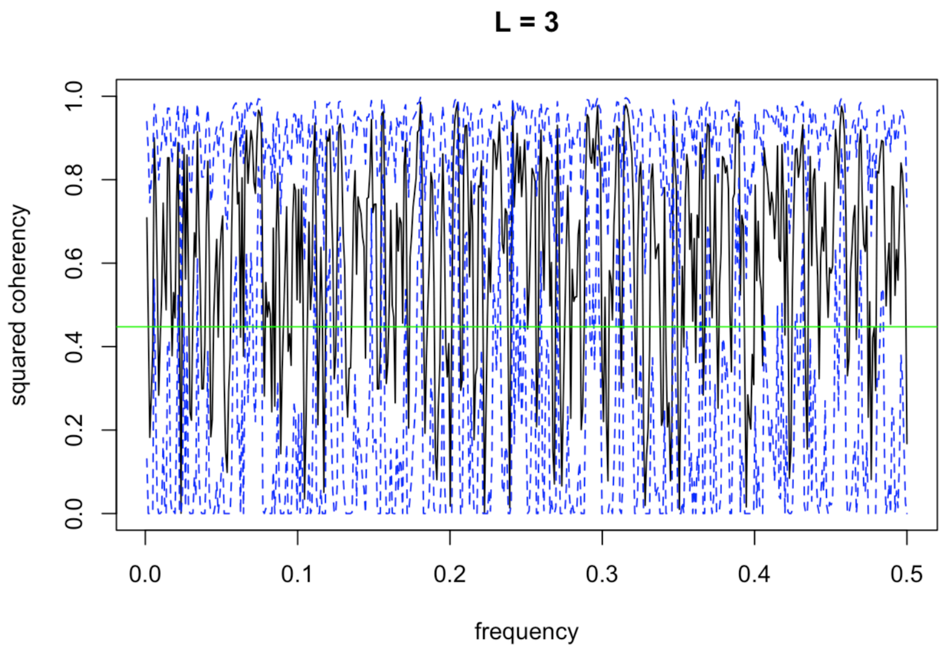
(i) L = 1

```
plot(s1, plot.type = 'coh', ci.lty = 2, main = "L = 1")
abline(h = rho(phi), col = 'green')
```



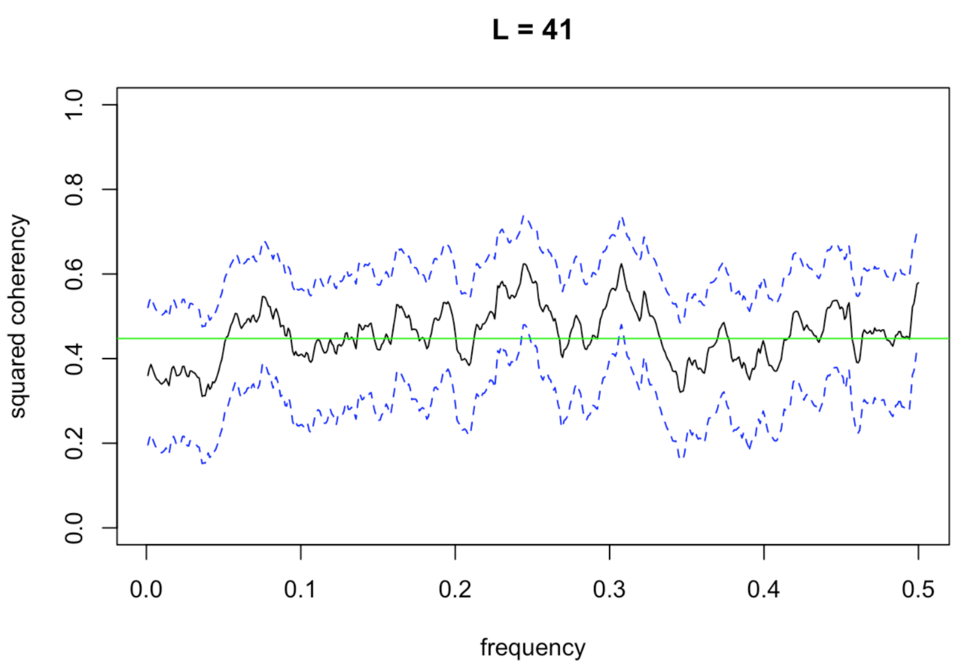
(ii) L = 3

```
plot(s2, plot.type = 'coh', ci.lty = 2, main = "L = 3")
abline(h = rho(phi), col = 'green')
```



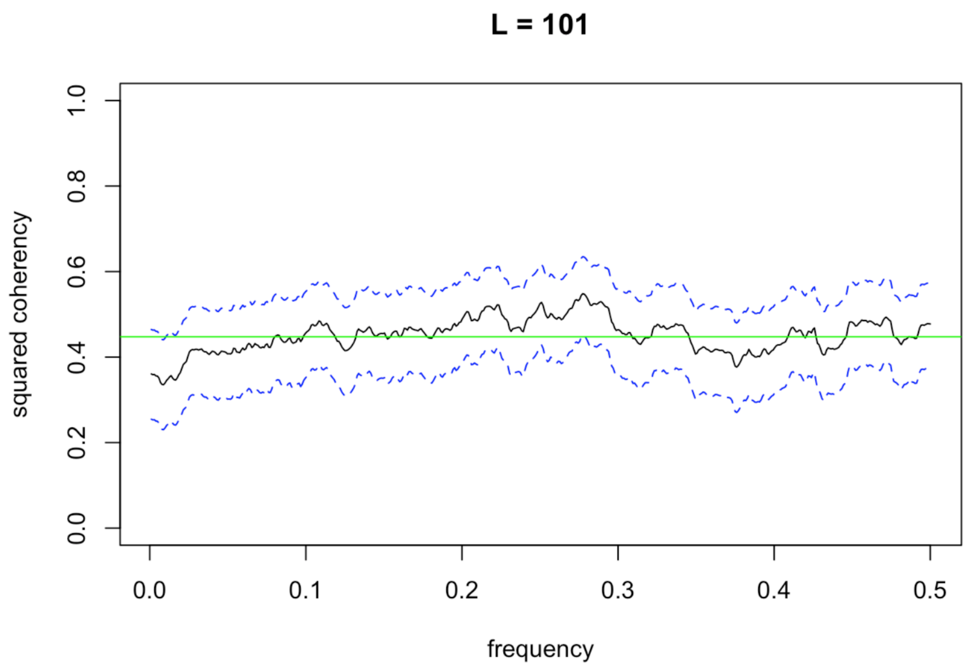
(iii) L = 41

```
plot(s3, plot.type = 'coh', ci.lty = 2, main = "L = 41")
abline(h = rho(phi), col = 'green')
```



(iv) L = 101

```
plot(s4, plot.type = 'coh', ci.lty = 2, main = "L = 101")
abline(h = rho(phi), col = 'green')
```



Sequentially, going from L = 1, L = 3 to L = 41, L = 101, we could gradually tell from the coherency between the simulation that as L goes up, the estimates become more accurate.

(c) (5 points) Compute the phase between x_t and y_t.

$$f_{yx}(\omega) = \phi\sigma^2 (\cos 2\pi\omega D - i \sin 2\pi\omega D)$$

$$\phi_{yx}(\omega) = \tan^{-1}\left(\frac{\phi\sigma^2 \sin 2\pi\omega D}{\phi\sigma^2 \cos 2\pi\omega D}\right)$$

$$= 2\pi\omega D$$

(d) (5 points) Simulate $n = 1024$ normal observations from x_t and y_t for $\phi = 0.9$, $\sigma^2 = 1$, and $D = 1$. Then estimate and plot the phase between the simulated series for the following values of L and comment: (i) $L = 1$, (ii) $L = 3$, (iii) $L = 41$, (iv) $L = 101$.

Q3(d)

```
set.seed(2020)
phi = 0.9
x = ts(rnorm(n = 1025, 0, 1)) #since D = 1 (lag exists)
v = rnorm(n = 1025, 0, 1) #since D = 1 (lag exists)
y = phi * lag(x,-1) + v
```

```
s1 = spec.pgram(ts(cbind(x,y))[2:1025,], taper = 0, plot = F)
s2 = spectrum(ts(cbind(x,y))[2:1025,], span = 3, taper = 0, plot = F)
s3 = spectrum(ts(cbind(x,y))[2:1025,], span = 41, taper = 0, plot = F)
s4 = spectrum(ts(cbind(x,y))[2:1025,], span = 101, taper = 0, plot = F)
```

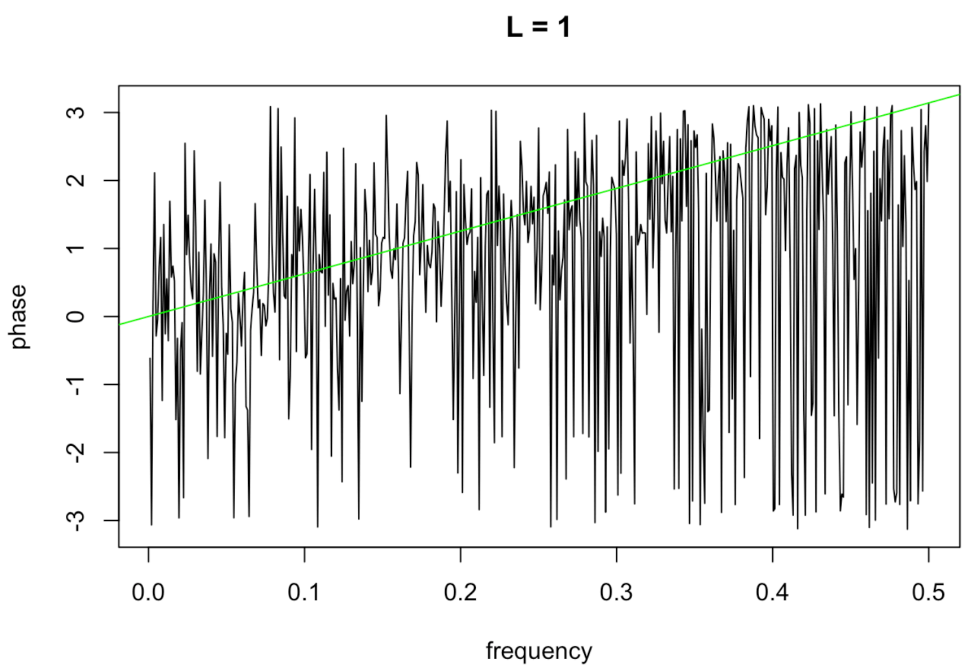
(i) L = 1

```
plot(s1, plot.type = 'phase', ci.lty = 2, main = "L = 1")
```

```
## Warning in qt(ci, 2/gg - 2): NaNs produced
```

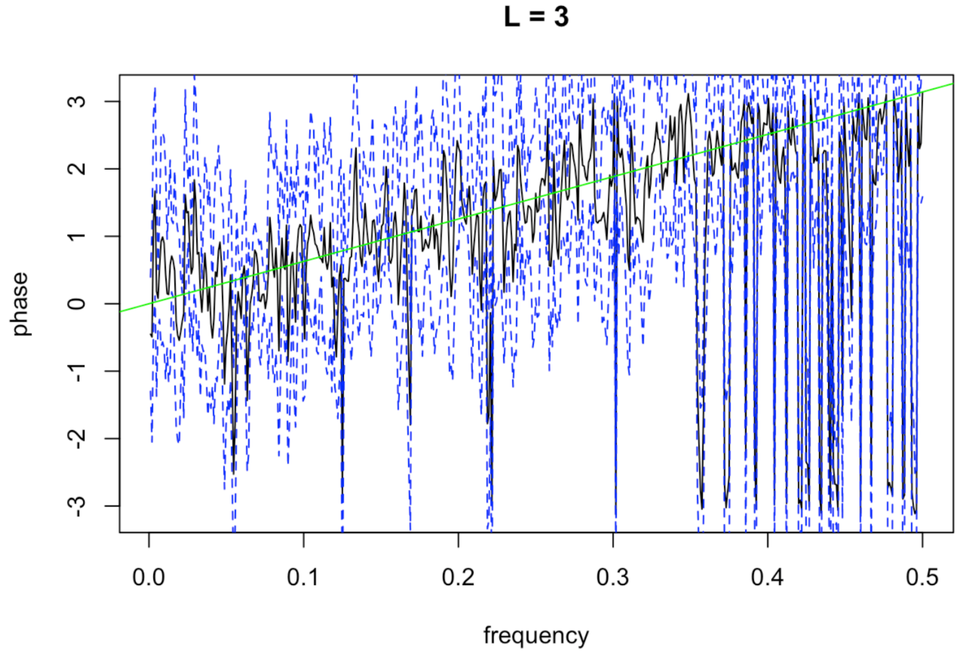
```
## Warning in sqrt(gg * (coh^{: NaNs produced
```

```
abline(a = 0, b = 2*pi, col = 'green')
```



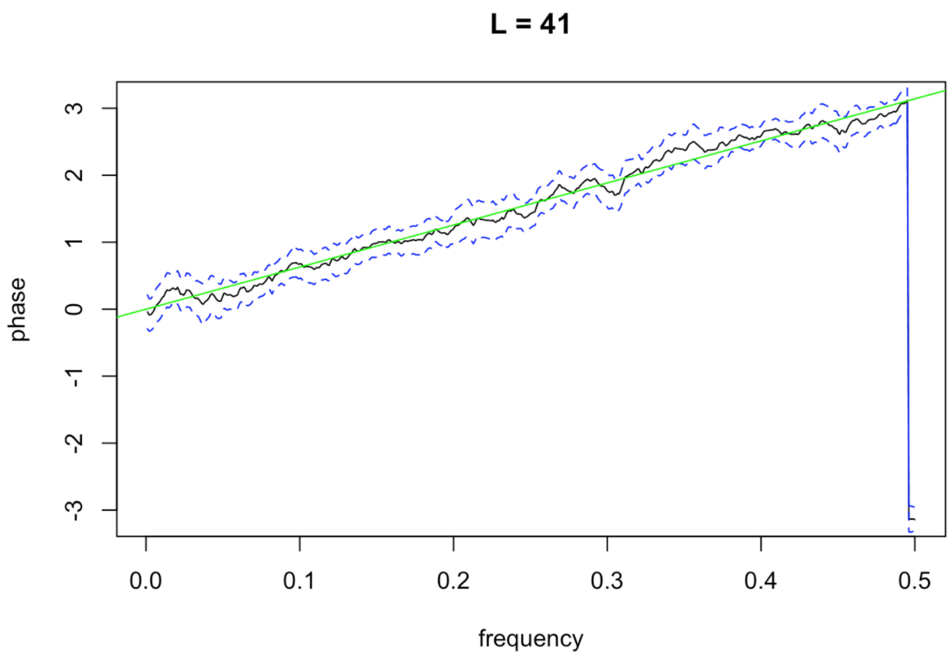
(ii) L = 3

```
plot(s2, plot.type = 'phase', ci.lty = 2, main = "L = 3")
abline(a = 0, b = 2*pi, col = 'green')
```



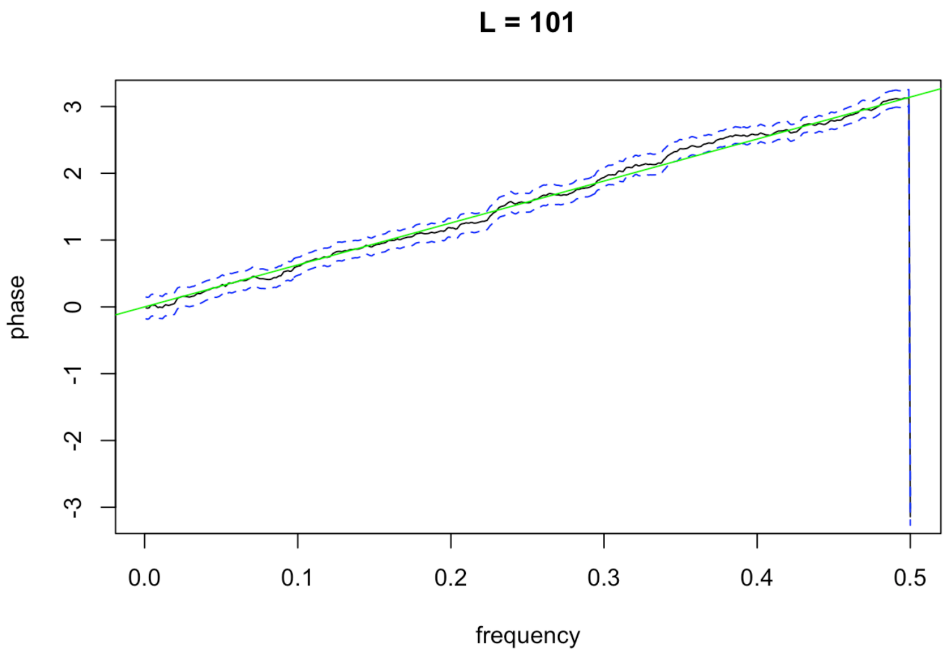
(iii) L = 41

```
plot(s3, plot.type = 'phase', ci.lty = 2, main = "L = 41")
abline(a = 0, b = 2*pi, col = 'green')
```



(iv) L = 101

```r
plot(s4, plot.type = 'phase', ci.lty = 2, main = "L = 101")
abline(a = 0, b = 2*pi, col = 'green')
```



Sequentially, going from L = 1, L = 3 to L = 41, L = 101, we could gradually tell from the phase plots that as L goes up, the estimates become more accurate, as seen from the black solid lines along with corresponding blue bandwidths (becoming more and more narrow).

4. (20 points) Exercise 4.34. The data set **climhyd**, contains 454 months of measured values for six climatic variables: (i) air temperature [Temp], (ii) dew point [DewPt], (iii) cloud cover [CldCvr], (iv) wind speed [WndSpd], (v) precipitation [Precip], and (vi) inflow [Inflow], at Lake Shasta in California. We would like to look at possible relations among the weather factors and between the weather factors and the inflow to Lake Shasta.

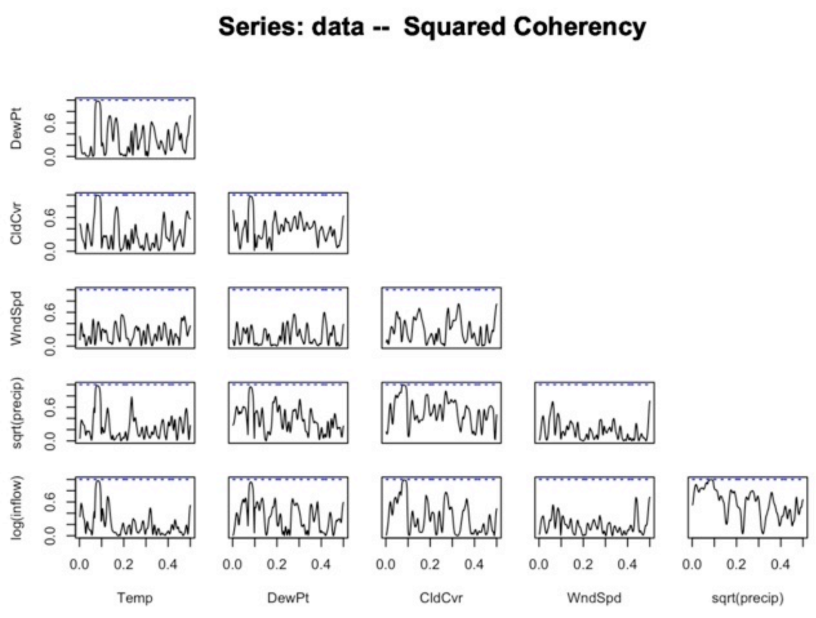
(a) (10 points) First transform the inflow and precipitation series as follows: $I_t = \log(i_t)$, where i_t is inflow, and $P_t = \sqrt{p_t}$, where p_t is precipitation. Then, compute the squared coherencies between all the weather variables and transformed inflow and argue that the strongest determinant of the inflow series is (transformed) precipitation. [Tip: If x contains multiple time series, then the easiest way to display all the squared coherencies is to plot the coherencies suppressing the confidence intervals, e.g., `mvspec(x, spans=c(7,7), taper=0.5, plot.type="coh", ci=-1)`.]

Q4(a)

```
library(astsa)
```

```
## Warning: package 'astsa' was built under R version 3.6.2
```

```
data = climhyd
data[,5] = sqrt(climhyd[,5])
data[,6] = log(climhyd[,6])

#transfrom inflow and t
colnames(data)[5:6] = c('sqrt(precip)', 'log(inflow)')
s = mvspec(data, spans = c(7,7), taper = 0.5, plot = F)
plot(s, plot.type = 'coh', ci = -1)
```



(b) (10 points) Fit a lagged regression model of the form

$$I_t = \beta_0 + \sum_{j=0}^{\infty} \beta_j P_{t-j} + w_t,$$

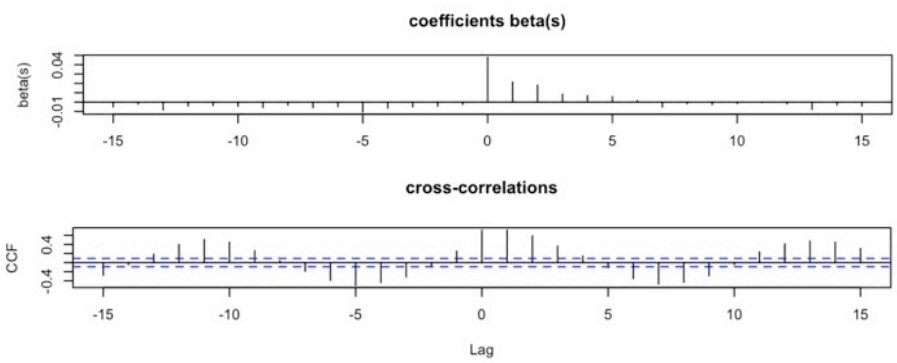
using thresholding, and then comment of the predictive ability of precipitation for inflow. (For model selection, i.e. thresholding the coefficients of P_{t-j} for j larger than a certain value to zero, any plausible approach will be accepted.)

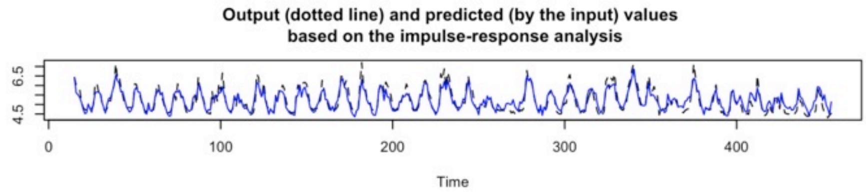
Q4(b)

```
fitLaggedReg = LagReg(data[,5], data[,6], L = 15, M = 32, threshold = 0.005)
```

```
## INPUT: data[, 5] OUTPUT: data[, 6]   L = 15    M = 32
##
## The coefficients beta(0), beta(1), beta(2) ... beta(M/2-1) are
##
## 0.04793956 0.02139648 0.01815036 0.008445174 0.007025733 0.005997653
## 0.00154858 -0.005373635 -0.001750188 -0.002687932 -0.001684001 -0.00060262
## -0.001971225 -0.00796091 -0.003235655 -0.004193099
##
##
## The coefficients beta(0), beta(-1), beta(-2) ... beta(-M/2+1) are
##
## 0.04793956 -0.002572632 -0.00435381 -0.005095025 -0.006422634 -0.01066165
## -0.005098131 -0.006017915 -0.003259976 -0.006007704 -0.004495352
## -0.003709271 -0.004155641 -0.00867441 -0.002252525 -0.005206633
```

```
## The positive lags, at which the coefficients are large
## in absolute value, and the coefficients themselves, are:
##      lag s       beta(s)
## [1,]     0  0.047939562
## [2,]     1  0.021396483
## [3,]     2  0.018150355
## [4,]     3  0.008445174
## [5,]     4  0.007025733
## [6,]     5  0.005997653
## [7,]     7 -0.005373635
## [8,]    13 -0.007960910
```






```
##
## The prediction equation is
## data[, 6](t) = alpha + sum_s[ beta(s)*data[, 5](t-s) ], where alpha = 4.368117
## MSE = 0.07900051
```

Here, I chose a threshold of 0.005 for thresholding out the beta above absolute value of 0.005. The reason behind it is mainly based on the judgment of MSE, as I got MSE of 0.08585649 with threshold of 0.002; MSE of 0.08735921 with threshold of 0.003; MSE of 0.08321044 with threshold of 0.004 and eventually the MSE of 0.07900051 with threshold 0.005 looks decent to me. The lags of beta in absolute value significantly decay. Additionally, from the bottom plot above, we could see that the predicted values (blue solid line) aligns with the actual precipitation for inflow output (black dotted line) very well; hence concluding that the predictive power is satisfying.

5. (15 points) Exercise 6.1. Consider a system process given by

$$x_t = -0.9x_{t-2} + w_t, \quad t = 1, \dots, n,$$

where $x_0 \sim \mathcal{N}(0, \sigma_0^2)$, $x_{-1} \sim \mathcal{N}(0, \sigma_1^2)$, and w_t is Gaussian white noise with variance σ_w^2. The system process is observed with noise, say,

$$y_t = x_t + v_t, \quad t \geq 1,$$

where v_t is Gaussian white noise with variance σ_v^2. Further, suppose x_0, x_{-1}, (w_t) and (v_t) are independent.

(a) (5 points) Write the system and observation equations in the form of a state space model, and find the values of σ_0^2 and σ_1^2 that make the observations y_t stationary.

Consider state equation $\vec{x}_t = \phi \vec{x}_{t-1} + \vec{w}_t$,

$$\begin{bmatrix} x_t \\ x_{t-1} \\ x_{t-2} \end{bmatrix} = \begin{bmatrix} 0 & -0.9 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ x_{t-3} \end{bmatrix} + \begin{bmatrix} w_t \\ 0 \\ 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} \sigma_w^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Observation equation $y_t = [1 \quad 0 \quad 0] \vec{x}_t + \vec{v}_t$

$$R = \sigma_v^2$$

Given a system process

$$\begin{aligned} x_t &= -0.9 x_{t-2} + w_t \\ &= -0.9(-0.9 x_{t-4} + w_{t-2}) + w_t \\ &= \begin{cases} (-0.9)^{\frac{t+1}{2}} x_{-1} + \sum_{i=0}^{\frac{t-1}{2}} (-0.9)^i w_{t-2i} & \text{when } t \text{ is odd} \\ (-0.9)^{\frac{t+1}{2}} x_0 + \sum_{i=0}^{t/2-1} (-0.9)^i w_{t-2i} & \text{when } t \text{ is even} \end{cases} \end{aligned}$$

$\text{cov}(y_{t+h}, y_t)$

(Since x_0, x_1, $\{v_t\}$, $\{w_t\}$ indep, $\{x_t\}$ is not dependent of $\{v_t\}$)

$$= \text{cov}(x_{t+h} + v_{t+h}, x_t + v_t)$$

$$= \text{cov}(x_{t+h}, x_t) + \text{cov}(v_{t+h}, v_t)$$

while we have $\gamma_v(h)$ indep of t

Hence, judging from $cov(x_{t+h}, x_t)$,
it can be separated into 2 cases by h,

When h is odd

Amongst all values of t,
one t is odd with remaining ones being even.

this means that x_t is linear combination of x_0 and w_t with even index

this means that x_t is linear combination of x_{-1} and w_t with odd index

From this, we see that
$cov(x_{t+h}, x_t) = 0$. independent of t.

When h is even

For $h=0$, $cov(x_{t+h}, x_t) = Var(x_t)$ $\Longrightarrow$ otherwise, $|h| > 0$

(even t)

$$
\begin{aligned}
&Var(x_t) \\
&= Var\left((-0.9)^{\frac{t}{2}} x_0 + \sum_{i=0}^{\frac{t}{2}-1} (-0.9)^i w_{t-2i}\right) \\
&= Var\left((-0.9)^{t/2} x_0\right) + Var\left(\sum_{i=0}^{t/2-1} (-0.9)^i w_{t-2i}\right) \\
&= 0.9^t \sigma_0^2 + \frac{100}{19}\sigma_w^2 - \frac{100}{19} 0.9^t \sigma_w^2 \\
&= 0.9^t \left(\sigma_0^2 - \frac{100}{19}\sigma_w^2\right) + \frac{100}{19}\sigma_w^2
\end{aligned}
$$

(odd t)

$$
\begin{aligned}
&Var(x_t) \\
&= Var\left((-0.9)^{\frac{t+1}{2}} x_{-1} + \sum_{i=0}^{\frac{t-1}{2}} (-0.9)^i w_{t-2i}\right) \\
&= Var\left((-0.9)^{\frac{t+1}{2}} x_{-1}\right) + Var\left(\sum_{i=0}^{\frac{t-1}{2}} (-0.9)^i w_{t-2i}\right) \\
&= 0.9^{t+1} \sigma_1^2 + \frac{100}{19}\sigma_w^2 - \frac{100}{19} 0.9^{t+1} \sigma_w^2 \\
&= 0.9^{t+1} \left(\sigma_1^2 - \frac{100}{19}\sigma_w^2\right) + \frac{100}{19}\sigma_w^2
\end{aligned}
$$

$$
\begin{aligned}
&cov(x_{t+h}, x_t) \\
&= cov(-0.9 x_{t+h-2} + w_{t+h}, x_t) \\
&= cov\left[(-0.9)^{h/2} x_t + \sum_{i=0}^{h/2-1} (-0.9)^i w_{t+h-2i}, x_t\right] \\
&= (-0.9)^{h/2} Var(x_t) + \underbrace{cov\left(\sum_{i=0}^{h/2-1} (-0.9)^i w_{t+h-2i}, x_t\right)}_{0} \\
&= (-0.9)^{h/2} Var(x_t)
\end{aligned}
$$

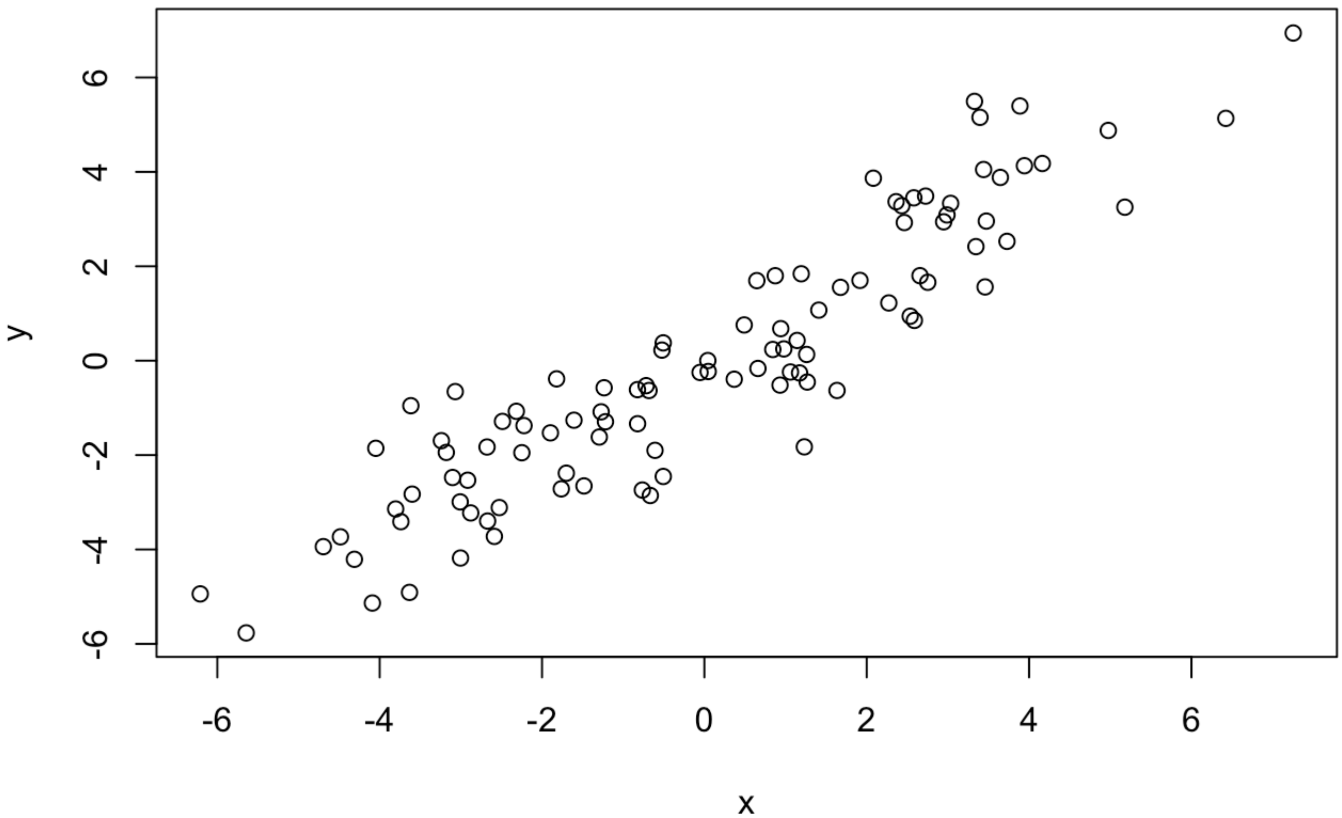
We need to have $Var(x_t)$ independent of t to make sure $\{y_t\}$ stationary

i.e. $\begin{cases} \sigma_0^2 - \frac{100}{19}\sigma_w^2 = 0 \\ \sigma_1^2 - \frac{100}{19}\sigma_w^2 = 0 \end{cases}$

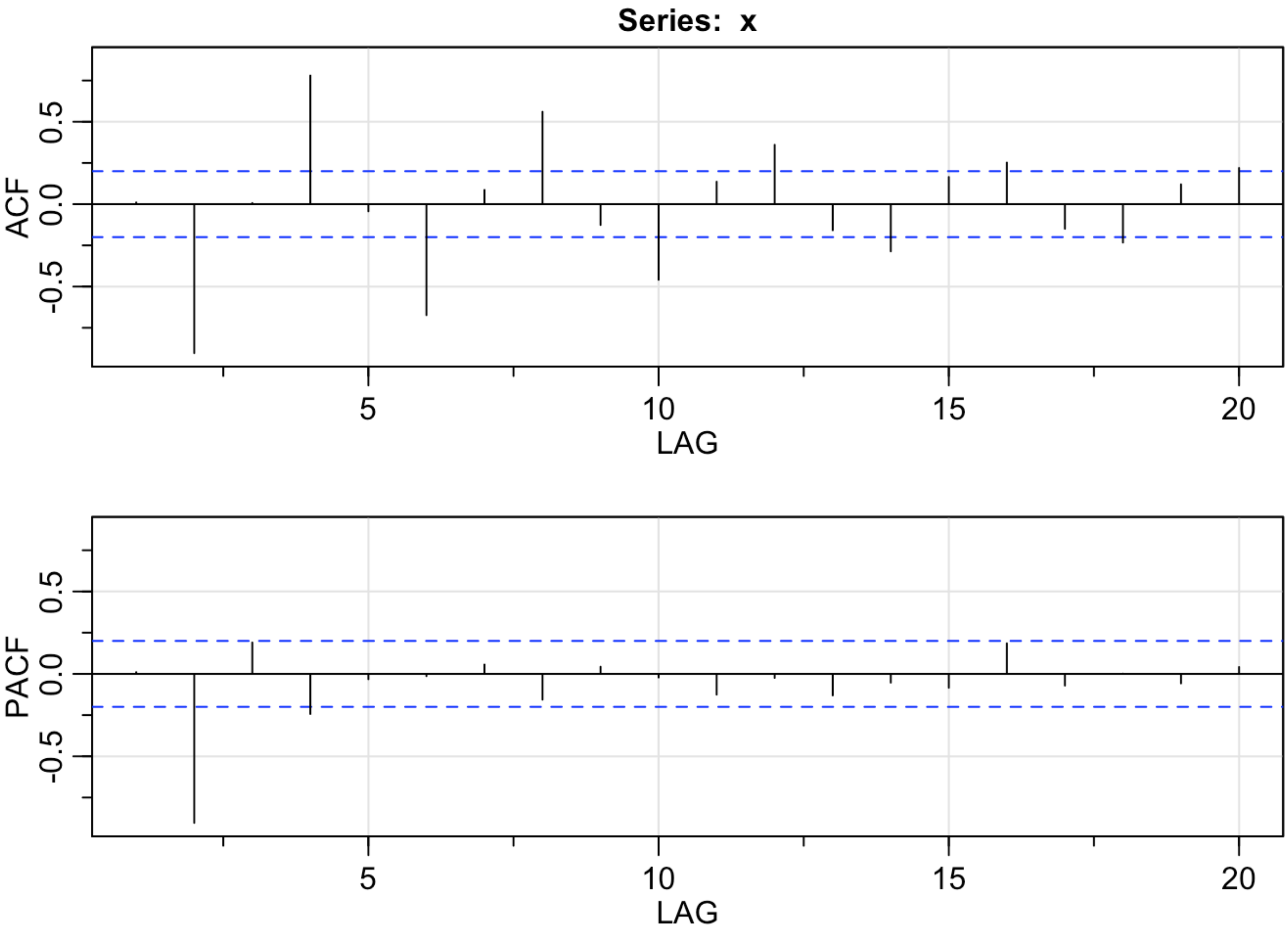
$\Rightarrow \sigma_0^2 = \sigma_1^2 = \frac{100}{19}\sigma_w^2$ #

(b) (10 points) Generate $n = 100$ observations with $\sigma_w = 1$, $\sigma_v = 1$ and using the values of σ_0^2, σ_1^2 in (a). Do a time plot of x_t and of y_t and compare the two processes. Also, compare the sample ACF and PACF of x_t and of y_t. Repeat this, but with $\sigma_v = 10$.

```
set.seed(2020)
n = 100

sigma_w = 1
sigma_v = 1
sigma_0 = 100/19 * sigma_w**2
sigma_minus_1 = 100/19 * sigma_w**2
x_0 = rnorm(1, 0, sigma_0)
x_minus_1 = rnorm(1, 0, sigma_minus_1)

w = rnorm(100, 0, sigma_w)
v = rnorm(100, 0, sigma_v)
x = rep(0, 100)
x[1] = -0.9*x_minus_1 + w[1]
x[2] = -0.9*x_0 + w[2]

for(i in 3:100){
  x[i] = -0.9*x[i-2] + w[i]
}

y=x+v

plot(x, y)
```

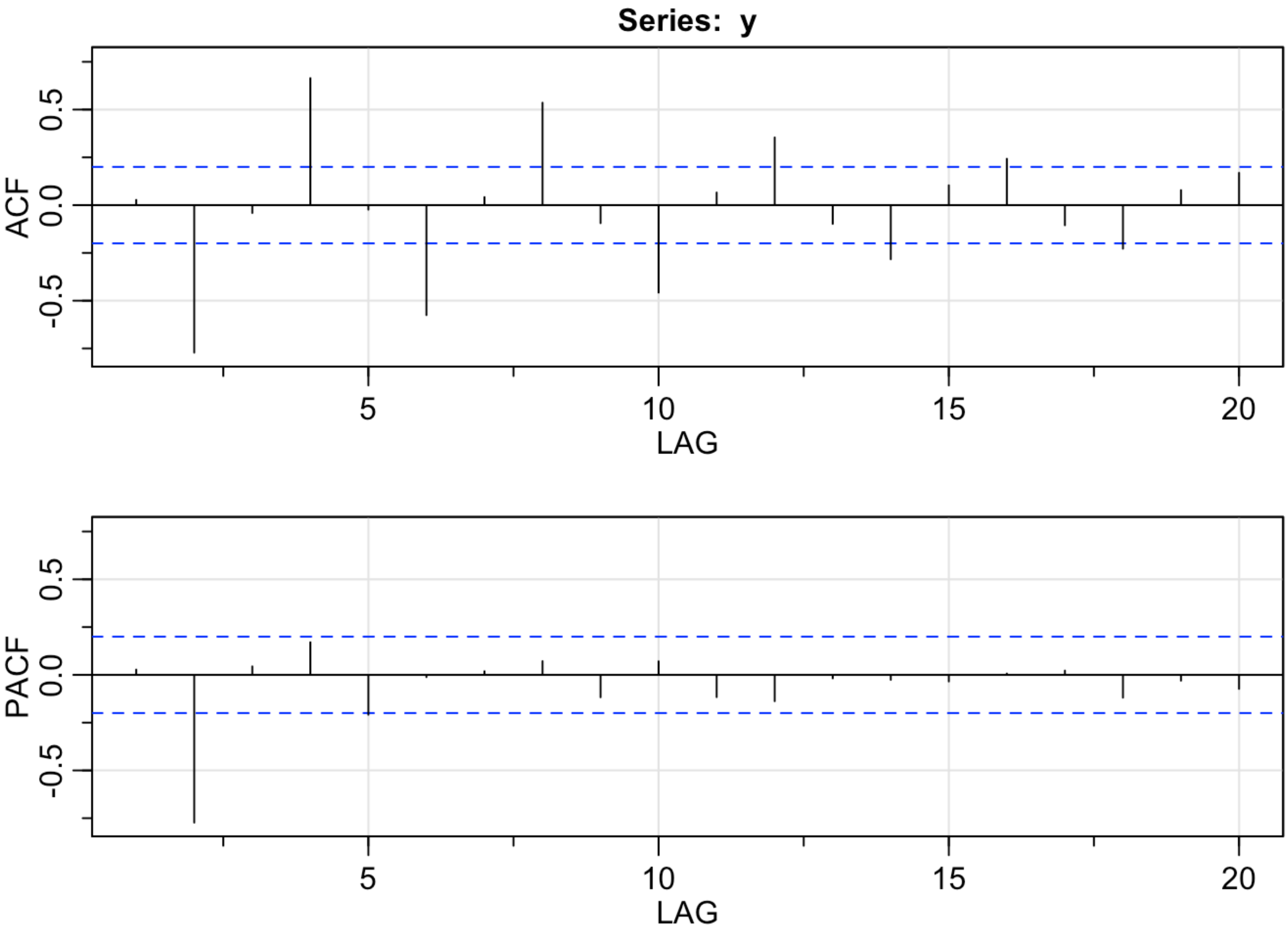



```
acf2(x)
```




```
##      [,1] [,2] [,3]  [,4]  [,5]  [,6] [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  0.01 -0.9 0.01  0.78 -0.04 -0.67 0.09  0.56 -0.13 -0.46  0.14  0.36
## PACF 0.01 -0.9 0.19 -0.24 -0.03 -0.01 0.06 -0.16  0.04 -0.02 -0.13 -0.02
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## ACF  -0.16 -0.29  0.16  0.25 -0.15 -0.23  0.12  0.22
## PACF -0.13 -0.05 -0.08  0.19 -0.07  0.00 -0.06  0.04
```

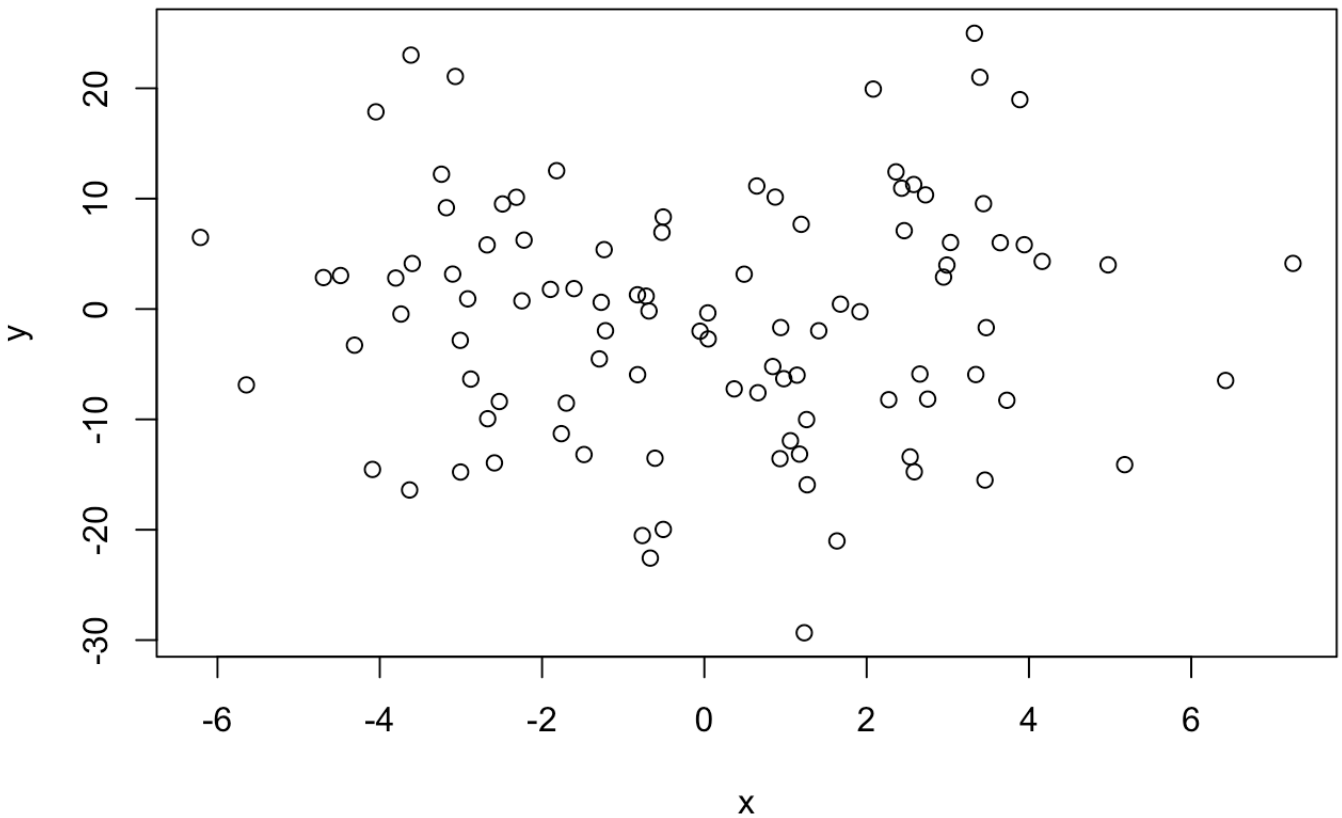
```
acf2(y)
```



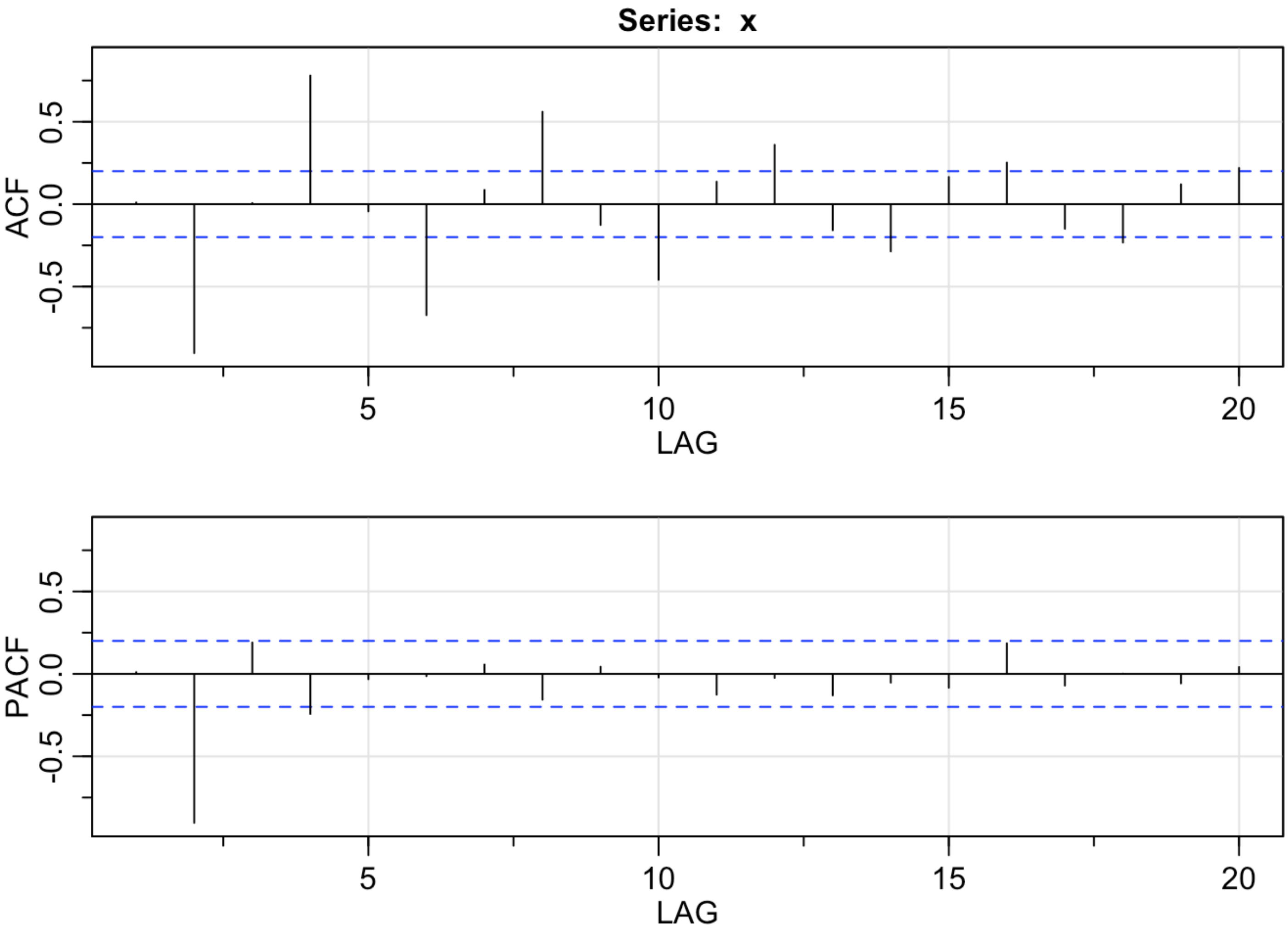

```
##       [,1]  [,2]  [,3] [,4]  [,5]  [,6] [,7] [,8]  [,9] [,10] [,11] [,12]
## ACF   0.03 -0.77 -0.04 0.66 -0.02 -0.58 0.04 0.54 -0.09 -0.46  0.07  0.35
## PACF  0.03 -0.77  0.04 0.17 -0.21 -0.01 0.02 0.07 -0.12  0.07 -0.12 -0.14
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## ACF  -0.10 -0.28  0.10  0.24 -0.11 -0.23  0.08  0.17
## PACF -0.02 -0.03 -0.03  0.01  0.02 -0.12 -0.03 -0.07
```

Comments & Comparison when $\sigma_v = 1$: From the scatter plot, we can see clear linear relationship between x and y. This aligns with the addition operation in the setup. Looking at the sample ACF, we see both x and y behave pretty much the same, as both show strong oscillation with slow decay along the lags, and the ACF hardly fall within bands for both x and y. For PACF, both decay within band at around lag of 4. Here, we see very close relationship in terms of behavior between x and y. It is noteworthy that from the ACF and PACF, both series look more like AR(2).

```r
set.seed(2020)
n = 100

sigma_w = 1
sigma_v = 10
sigma_0 = 100/19 * sigma_w**2
sigma_minus_1 = 100/19 * sigma_w**2
x_0 = rnorm(1, 0, sigma_0)
x_minus_1 = rnorm(1, 0, sigma_minus_1)

w = rnorm(100, 0, sigma_w)
v = rnorm(100, 0, sigma_v)
x = rep(0, 100)
x[1] = -0.9*x_minus_1 + w[1]
x[2] = -0.9*x_0 + w[2]

for(i in 3:100){
  x[i] = -0.9*x[i-2] + w[i]
}

y=x+v

plot(x, y)
```

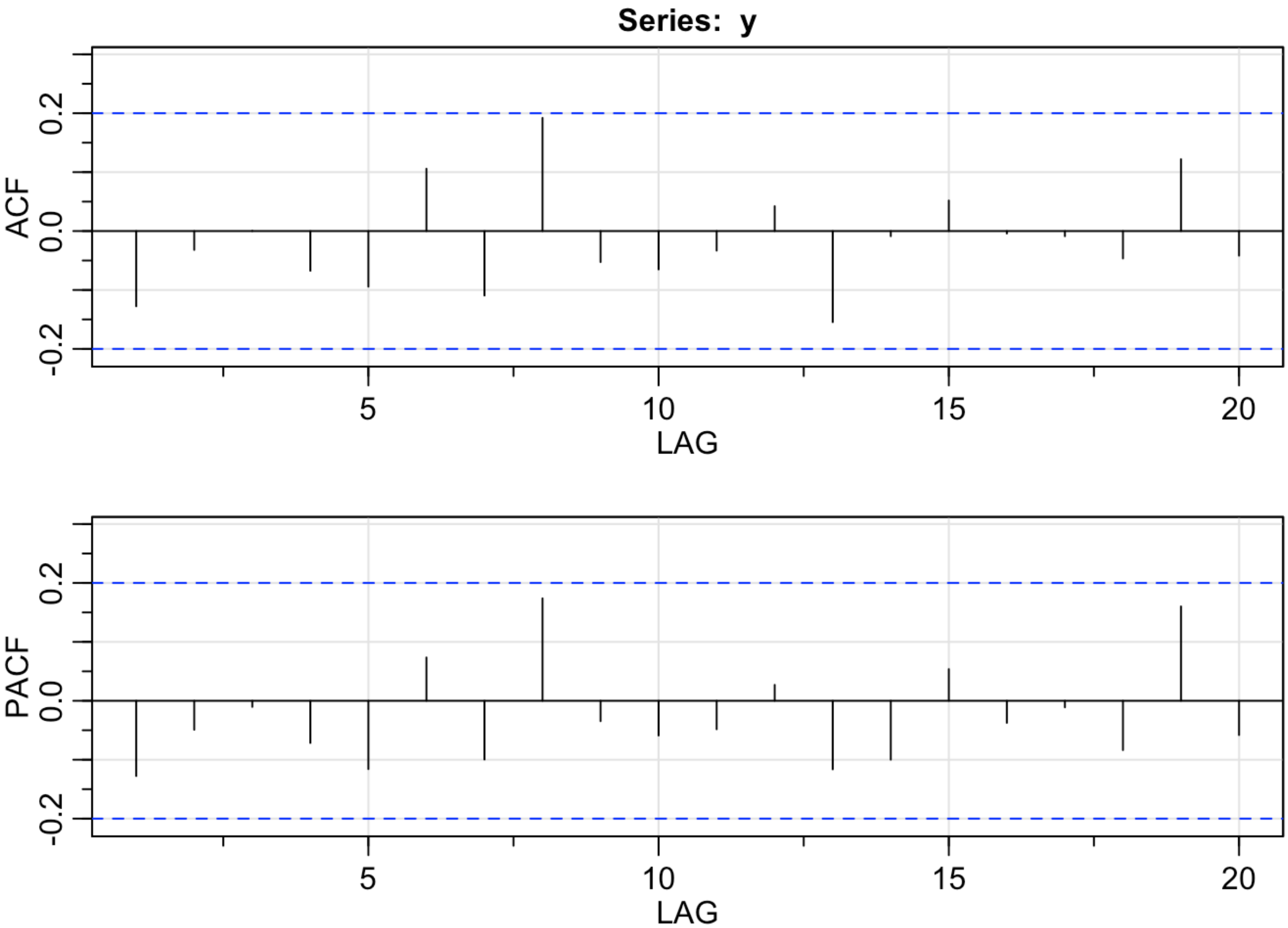



```
acf2(x)
```




```
##      [,1] [,2] [,3]  [,4]  [,5]  [,6] [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  0.01 -0.9 0.01  0.78 -0.04 -0.67 0.09  0.56 -0.13 -0.46  0.14  0.36
## PACF 0.01 -0.9 0.19 -0.24 -0.03 -0.01 0.06 -0.16  0.04 -0.02 -0.13 -0.02
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## ACF  -0.16 -0.29  0.16  0.25 -0.15 -0.23  0.12  0.22
## PACF -0.13 -0.05 -0.08  0.19 -0.07  0.00 -0.06  0.04
```

```
acf2(y)
```




```
##       [,1]  [,2]  [,3]  [,4]  [,5] [,6]  [,7] [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.13 -0.03  0.00 -0.07 -0.09 0.11 -0.11 0.19 -0.05 -0.07 -0.03  0.04
## PACF -0.13 -0.05 -0.01 -0.07 -0.12 0.07 -0.10 0.17 -0.03 -0.06 -0.05  0.03
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## ACF  -0.15 -0.01  0.05  0.00 -0.01 -0.05  0.12 -0.04
## PACF -0.12 -0.10  0.05 -0.04 -0.01 -0.08  0.16 -0.06
```

Comments & Comparison when $\sigma_v = 10$: From the scatter plot, there is no longer linear relationship between x and y. This aligns with the fact that we are adding in a even stronger variable with large variance when we do the addition operation. The influence is so big that the previous effect diminishes. Now looking at the series Y, it behaves more like ARMA now from the ACF and PACF. The ACF and PACF for x and y look significantly different now. For series x, we still have significant oscillation with slow decay across lags, while for series y, the ACF and PACF lie well within band all the way down with different lags. There is no close relationship in terms of ACF and PACF behavior, unlike the previous case.

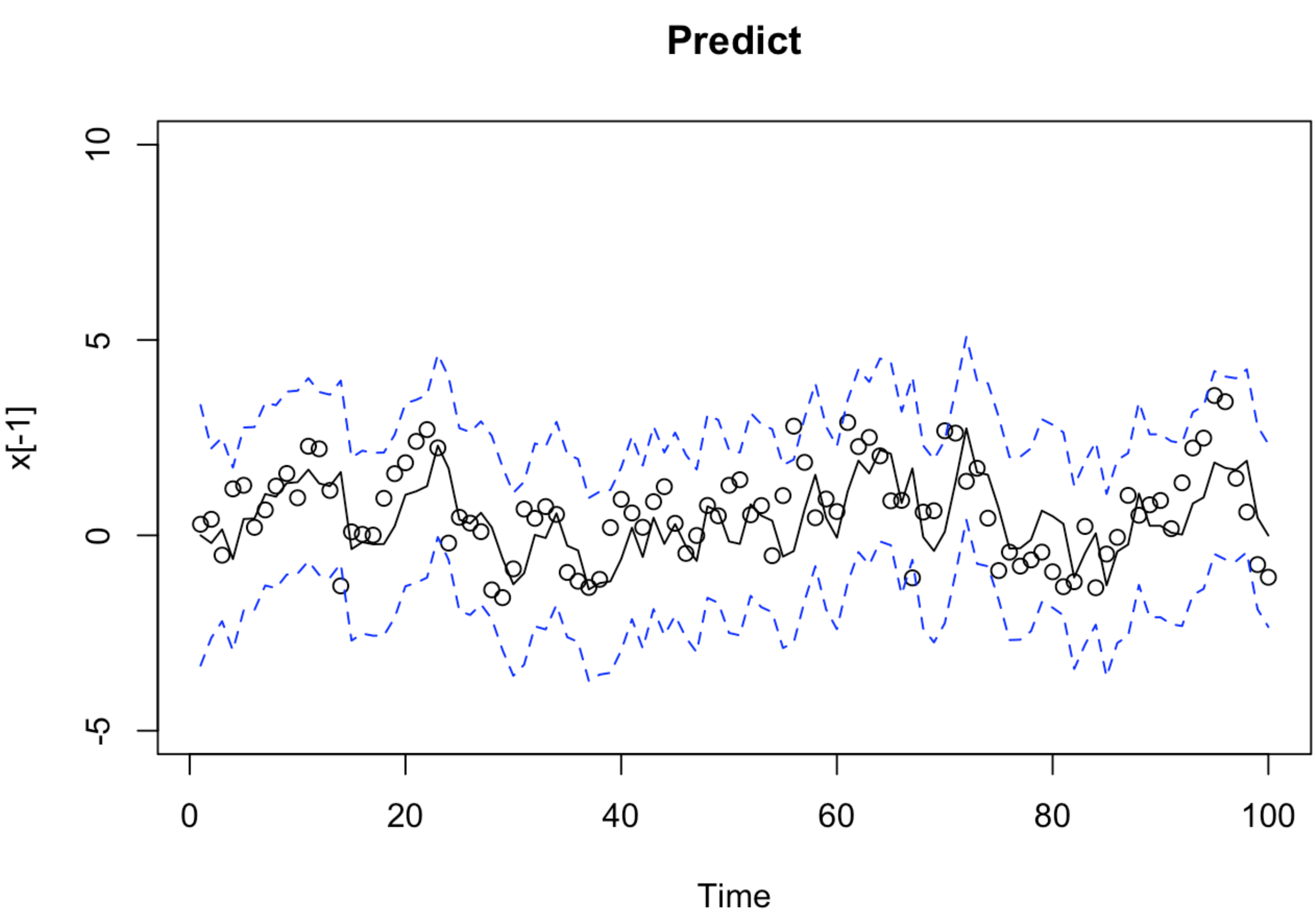
6. (10 points) Simulate $n = 100$ observations from the following state-space model:

$$x_t = 0.8x_{t-1} + w_t \quad \text{and} \quad y_t = x_t + v_t$$

where $x_0 \sim \mathcal{N}(0, 2.78)$, $w_t \sim$ iid $\mathcal{N}(0,1)$, and $v_t \sim$ iid $\mathcal{N}(0,1)$ are all mutually independent. Compute and plot the data, y_t, the one-step-ahead predictors, y_t^{t-1} along with ± 2 times of the root mean square prediction errors, $\sqrt{\mathbb{E}\left[(y_t - y_t^{t-1})^2\right]}$ using Example 6.5 as a guide.

Code starting from next page
is adopted from page 300 of
Time Series Analysis and its Applications (textbook)

```r
# generate data
set.seed(1);
num = 100
w = rnorm(num,0,1)
v = rnorm(num,0,1)
x = rep(0, num+1)
x[1] = rnorm(1,0,2.78)

# state: x[0], x[1],..., x[100]
for(i in 2:(num+1)){
  x[i] = 0.8*x[i-1] + w[i-1]
}

y = x[-1] + v # obs: y[1],..., y[100]

# filter and smooth (Ksmooth0 does both)
ks = Ksmooth0(num, y, A=1, mu0=0, Sigma0=2.78, Phi=0.8, cQ=1, cR=1) # start figure
Time = 1:num
plot(Time, x[-1], main='Predict', ylim=c(-5,10))
lines(ks$xp)
lines(ks$xp+2*sqrt(ks$Pp), lty=2, col=4)
lines(ks$xp-2*sqrt(ks$Pp), lty=2, col=4)
```



```
x[1];
```

```
## [1] 1.138137
```

```
ks$x0n;
```

```
##             [,1]
## [1,] -0.1261557
```

```
sqrt(ks$P0n) # initial value info
```

```
##          [,1]
## [1,] 1.17068
```

7. (15 points) Exercise 6.11. As an example of the way the state-space model handles the missing data problem, suppose the first-order autoregressive process

$$x_t = \phi x_{t-1} + w_t$$

has an observation missing at $t = m$, leading to the observations $y_t = A_t x_t$, where $A_t = 1$ for all t except $t = m$ wherein $A_t = 0$. Assume x_0 is mean zero with variance $\sigma_w^2/(1-\phi^2)$, where $\sigma_w^2 > 0$ is the variance of (w_t). Show the Kalman smoother estimators in this case are

$$x_t^n = \begin{cases} \phi y_1 & t = 0, \\ \frac{\phi}{1+\phi^2}(y_{m-1} + y_{m+1}) & t = m, \\ y_t & t \neq 0, m, \end{cases}$$

with mean squared covariances determined by

$$P_t^n = \begin{cases} \sigma_w^2 & t = 0, \\ \sigma_w^2/(1+\phi^2) & t = m, \\ 0 & t \neq 0, m. \end{cases}$$

Since we got $X_0 \sim N\left(0, \frac{\sigma_w^2}{1-\phi^2}\right)$

and $\{W_t\}$ as Gaussian white noise,

X_t is a linear combination of Gaussian RVs,

Also, from Piazza post@105, it is instructed that we could assume everything is Gaussian.

X_t is Gaussian

$\Rightarrow$ Any joint or conditional distributions of X_t is also Gaussian.

$$X_t^n = \mathbb{E}\{X_t \mid y_{1:n}\}$$

$$P_t^n = \mathbb{E}\{(X_t - X_t^n)(X_t - X_t^n)'\}$$

$\boxed{\text{For } t=0}$

since we have $\{w_t\}$ as Gaussian white noise, $x_0, (x_2, \dots, x_n)$ are conditionally independent given x_1,

$$x_0^n = \mathbb{E}(x_0 \mid y_1 \dots y_n)$$
$$= \mathbb{E}(x_0 \mid x_1 \dots x_{m-1}, 0, x_{m+1} \dots x_n)$$
$$= \mathbb{E}(x_0 \mid x_1)$$

$$x_0 \sim N\left(0, \frac{\sigma_w^2}{1-\phi^2}\right)$$

$$w_1 \sim N(0, \sigma_w^2)$$

$$x_1 = \phi x_0 + w_1 \sim N\left(0, \frac{\phi^2\sigma_w^2}{1-\phi^2} + \sigma_w^2\right)$$
$$= N\left(0, \frac{\sigma_w^2}{1-\phi^2}\right)$$

$$\mathrm{cor}(x_0, x_1) = \frac{\phi\, \mathrm{Var}(x_0)}{\sqrt{\mathrm{Var}(x_0)\mathrm{Var}(x_1)}} = \phi$$

$$\Rightarrow (x_0, x_1) \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_w^2/1-\phi^2 & (\phi/_{1-\phi^2})\,\sigma_w^2 \\ \frac{\phi}{1-\phi^2}\sigma_w^2 & \sigma_w^2/_{1-\phi^2} \end{pmatrix}\right)$$

$\Rightarrow$ $p(x_0 \mid x_1)$

$$= \frac{1}{\sqrt{2\pi\sigma^2_{x_0|x_1}}} \exp\left\{-\frac{1}{2}\left(\frac{x_0 - \alpha x_0(x_1 - \beta x_0|x_1\, x_1)}{\sigma_{x_0|x_1}}\right)^2\right\}$$

$$= \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left\{-\frac{1}{2}\left(\frac{x_0 - \phi x_1}{\sigma_w}\right)^2\right\}$$

As Gaussian pdf, $x_0 \mid x_1 \sim N(\phi x_1, \sigma_w^2)$

Hence, $\begin{cases} x_0^n = \mathbb{E}(x_0 \mid x_1 = y_1) = \phi y_1 \\ P_0^n = \mathbb{E}((x_0 - x_0^t)(x_0 - x_0^t)') = \sigma_w^2 \end{cases}$

$\boxed{\text{For } t=m}$ Assuming conditional Independence

$$\mathbb{E}(x_m \mid y_1 \cdots y_n)$$
$$= \mathbb{E}(x_m \mid x_1 \ldots x_{m-1}, 0, x_{m+1}, \ldots, x_n)$$
$$= \mathbb{E}(x_m \mid x_{m-1}, x_{m+1})$$

With $\{x_t\}$ being stationary,

$$P(x_m \mid x_{m-1}, x_{m+1})$$
$$= \frac{1}{\sqrt{2\pi \frac{\sigma_w^2}{1+\phi^2}}} \exp\left\{-\frac{1}{2} \frac{1+\phi^2}{\sigma_w^2} \left(x_m - \frac{\phi}{1+\phi^2}(x_{m-1} + x_{m+1})\right)^2\right\}$$

As a Gaussian pdf,

$$x_m \mid x_{m-1}, x_{m+1} \sim N\left(\frac{\phi}{1+\phi^2}(x_{m-1} + x_{m+1}), \frac{\sigma_w^2}{1+\phi^2}\right)$$

$$\Rightarrow \begin{cases} x_m^n = \dfrac{\phi}{1+\phi^2}(x_{m-1} + x_{m+1}) \\ P_m^n = \dfrac{\sigma_w^2}{1+\phi^2} \end{cases}$$

$\boxed{\text{For } t \neq 0 \text{ or } m}$ $\begin{cases} x_t^n = \mathbb{E}\{x_t \mid y_1 \cdots y_t \cdots y_n\} = y_t \\ P_t^n = \mathbb{E}\{(x_t - y_t)(x_t - y_t)'\} = 0 \end{cases}$

#

8. (10 points) Exercise 6.13. Redo Example 6.10 on the logged Johnson & Johnson quarterly earnings per share. Namely, apply the same modeling technique and code as in Example 6.10 to the logged earnings. Comment on the results of smoothing and forecasting.

```
jj = log(jj)
num = length(jj)
A = cbind(1,1,0,0)
# Function to Calculate Likelihood
Linn =function(para){
  Phi = diag(0,4); Phi[1,1] = para[1]
  Phi[2,]=c(0,-1,-1,-1); Phi[3,]=c(0,1,0,0); Phi[4,]=c(0,0,1,0)
  cQ1 = para[2]; cQ2 = para[3]
  cQ = diag(0,4); cQ[1,1]=cQ1; cQ[2,2]=cQ2
  cR = para[4] # sqrt r11
  kf = Kfilter0(num, jj, A, mu0, Sigma0, Phi, cQ, cR)
  return(kf$like)
  }

# Initial Parameters
mu0 = c(.7,0,0,0); Sigma0 = diag(.04,4)
init.par = c(1.03,.1,.1,.5)

# Estimation and Results
est = optim(init.par, Linn,NULL, method='BFGS', hessian=TRUE,control=list(trace=1,REPORT
=1))
```

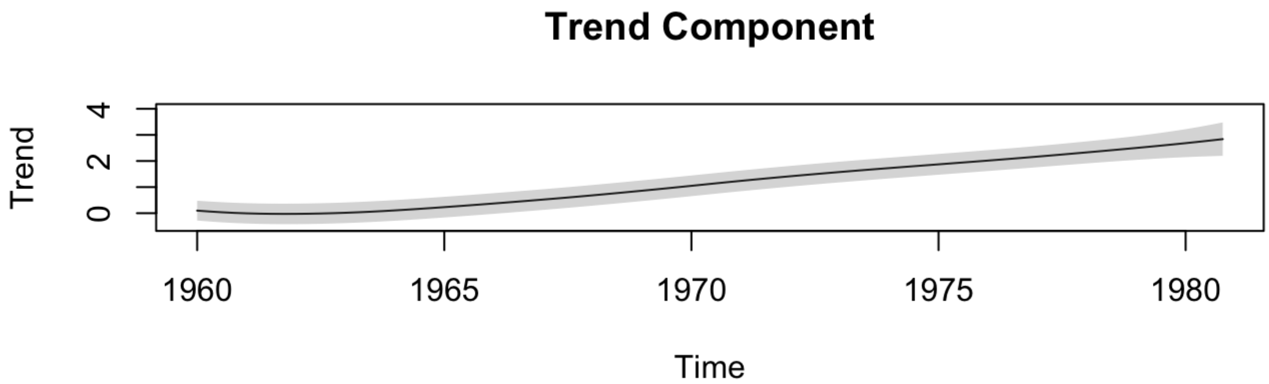
```
## initial  value -28.451785
## iter   2 value -30.744231
## iter   3 value -69.514381
## iter   4 value -78.377289
## iter   5 value -78.411537
## iter   6 value -103.836670
## iter   7 value -111.362949
## iter   8 value -126.168072
## iter   9 value -126.520431
## iter  10 value -127.240122
## iter  11 value -130.640237
## iter  12 value -132.930995
## iter  13 value -133.830379
## iter  14 value -135.678426
## iter  15 value -135.908167
## iter  16 value -135.927311
## iter  17 value -135.934767
## iter  18 value -135.935161
## iter  19 value -135.935438
## iter  20 value -135.943654
## iter  21 value -135.943673
## iter  21 value -135.943673
## iter  22 value -135.943677
## iter  22 value -135.943677
## iter  22 value -135.943677
## final  value -135.943677
## converged
```

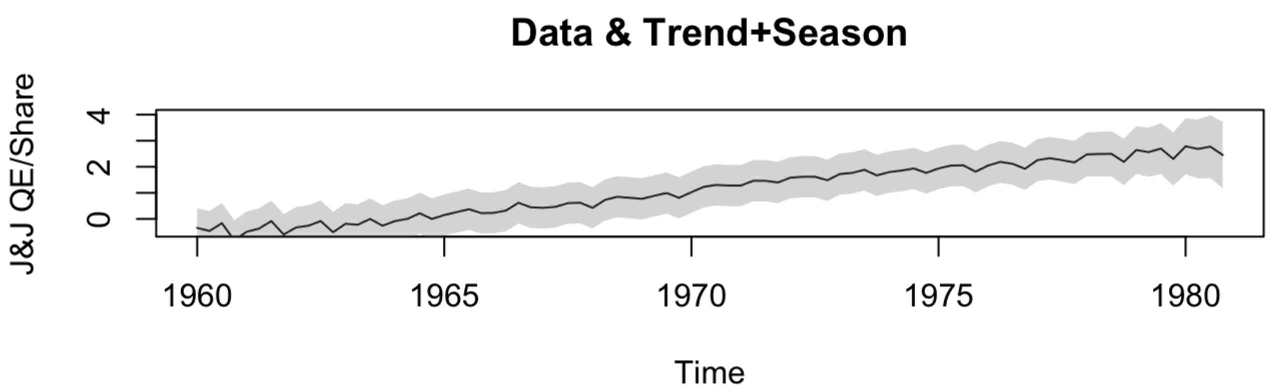
```
SE  = sqrt(diag(solve(est$hessian)))
u = cbind(estimate=est$par, SE)
rownames(u)=c('Phi11','sigw1','sigw2','sigv'); u
```

```
##             estimate          SE
## Phi11  1.019193e+00 0.004716694
## sigw1  6.326131e-02 0.007765171
## sigw2 -3.052053e-02 0.004740566
## sigv  -1.531284e-05 0.022302895
```

```
# Smooth
Phi = diag(0,4); Phi[1,1] = est$par[1]
Phi[2,]=c(0,-1,-1,-1); Phi[3,]=c(0,1,0,0); Phi[4,]=c(0,0,1,0)
cQ1 = est$par[2]; cQ2 = est$par[3]
cQ = diag(1,4); cQ[1,1]=cQ1; cQ[2,2]=cQ2
cR = est$par[4]
ks = Ksmooth0(num,jj,A,mu0,Sigma0,Phi,cQ,cR)
```

```
# Plots
Tsm = ts(ks$xs[1,,], start=1960, freq=4)
Ssm = ts(ks$xs[2,,], start=1960, freq=4)
p1 = 3*sqrt(ks$Ps[1,1,]); p2 = 3*sqrt(ks$Ps[2,2,])
par(mfrow=c(2,1))
plot(Tsm, main='Trend Component', ylab='Trend', ylim=c(-.5,4))
xx = c(time(jj), rev(time(jj)))
yy = c(Tsm-p1, rev(Tsm+p1))
polygon(xx, yy, border=NA, col=gray(.5, alpha = .3))
plot(jj, main='Data & Trend+Season', ylab='J&J QE/Share', ylim=c(-.5,4))
xx = c(time(jj), rev(time(jj)) )
yy = c((Tsm+Ssm)-(p1+p2), rev((Tsm+Ssm)+(p1+p2)) )
polygon(xx, yy, border=NA, col=gray(.5, alpha = .3))
```

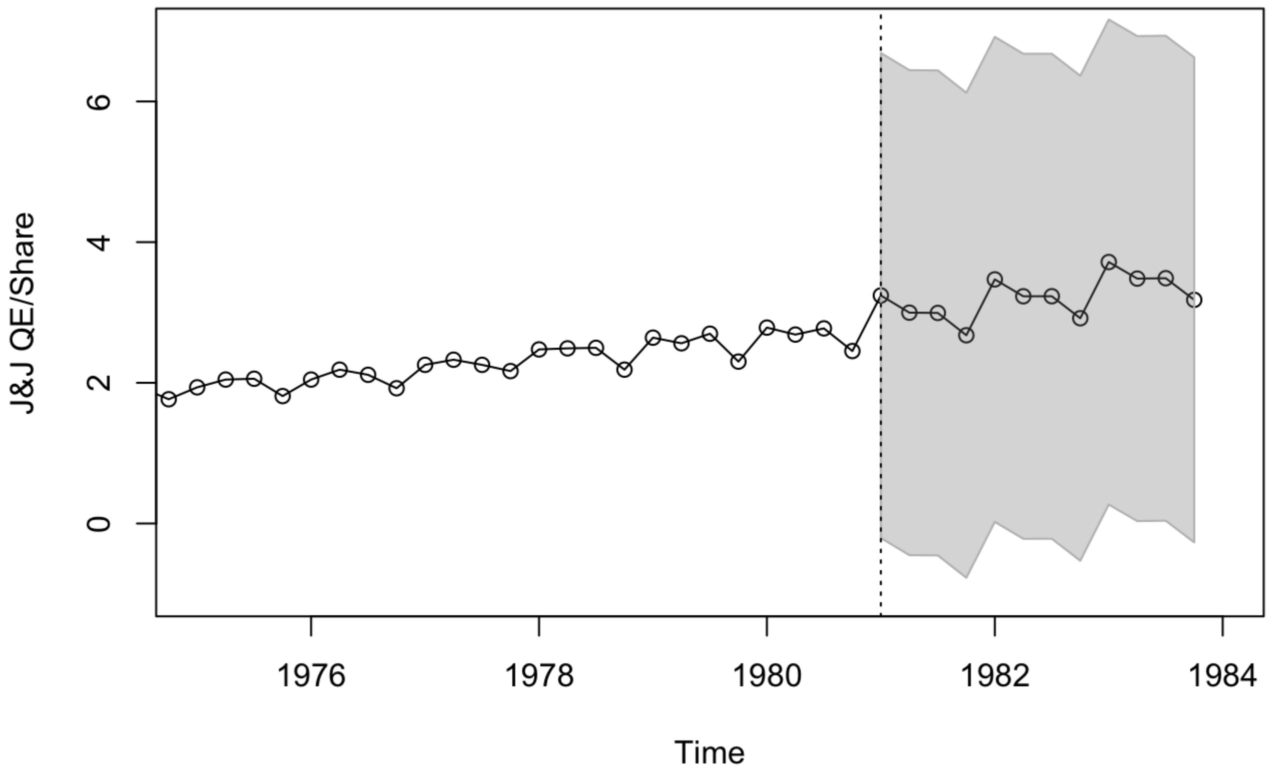




Comments:

From the plot above, we could see that the trend is linear over time. This makes sense because in the smoothing process, unlike what's shown on textbook p.318, from which we see the exponentially increasing seasonal components, we basically take log here to the exponential components, which results in such linearity. And as for the data with trend and season plot below, the trend is also linear with constant variance along time.

```
# Forecast
n.ahead = 12;
y = ts(append(jj, rep(0,n.ahead)), start=1960, freq=4)
rmspe = rep(0,n.ahead); x00 = ks$xf[,,num]; P00 = ks$Pf[,,num]
Q = t(cQ)%*%cQ; R = t(cR)%*%(cR)
for (m in 1:n.ahead){
   xp  = Phi%*%x00; Pp = Phi%*%P00%*%t(Phi)+Q
   sig = A%*%Pp%*%t(A)+R; K = Pp%*%t(A)%*%(1/sig)
   x00 = xp; P00 = Pp-K%*%A%*%Pp
   y[num+m] = A%*%xp; rmspe[m] = sqrt(sig)  }
plot(y, type='o', main='', ylab='J&J QE/Share', ylim=c(-1,7), xlim=c(1975,1984))
  upp = ts(y[(num+1):(num+n.ahead)]+2*rmspe, start=1981, freq=4)
  low = ts(y[(num+1):(num+n.ahead)]-2*rmspe, start=1981, freq=4)
   xx = c(time(low), rev(time(upp)))
   yy = c(low, rev(upp))
  polygon(xx, yy, border=8, col=gray(.5, alpha = .3))
  abline(v=1981, lty=3)
```



Comments:

For the plot above, we could see that the forecasted values are basically an extension of the latter part of the observed data, as the pattern is very well preserved. Here, we see that by transforming our data into logarithms, we are capable of stabilizing the underlying variance over time.