STAT33600 Homework 2

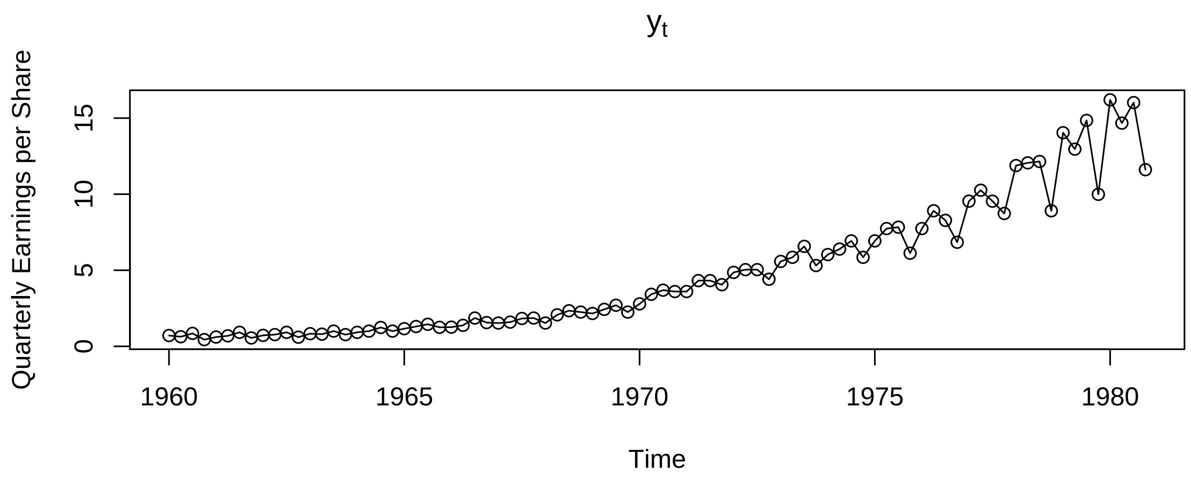
Sarah Adilijiang

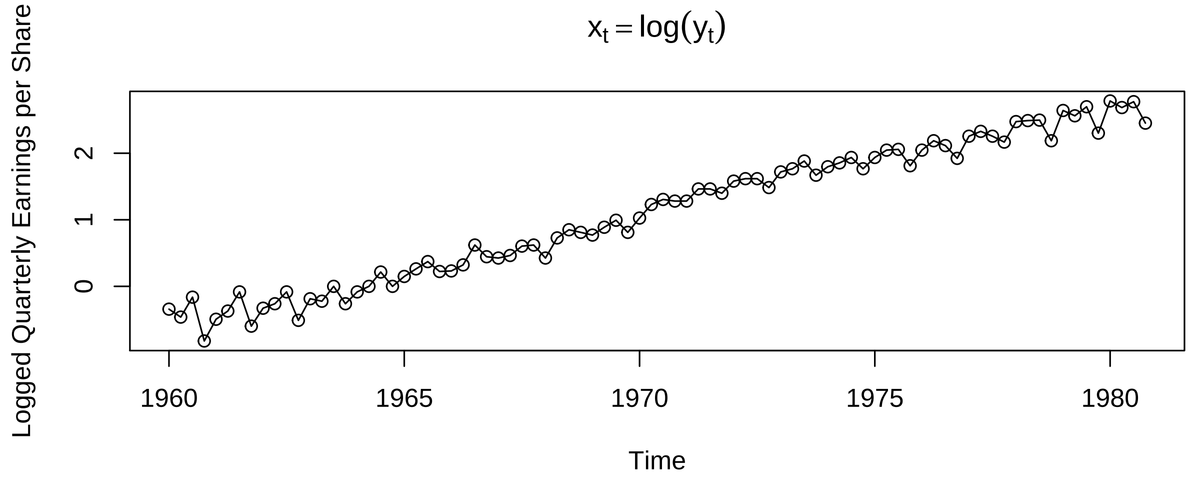
2.1

```r
library(astsa)
```

```
## Warning: package 'astsa' was built under R version 3.5.3
```

```r
par(mfrow=c(2,1))
# data = data.frame(jj)
plot(jj, type='o', ylab='Quarterly Earnings per Share', main = expression(y[t]))
plot(log(jj), type='o', ylab='Logged Quarterly Earnings per Share',
                         main=expression(x[t]==log(y[t])))
```





$x_t = \log(y_t)$, where y_t is the Johnson & Johnson quarterly data shown above.

$x_t = T_t + S_t + N_t$, where T_t is a trend component, S_t is a seasonal component, N_t is noise.

We can see that there is an obvious upward trend in the series x_t, so it is a good choice to use simple linear regression for the trend.

(a)

Regression model:

$$x_t = \beta t + \alpha_1 Q_1(t) + \alpha_2 Q_2(t) + \alpha_3 Q_3(t) + \alpha_4 Q_4(t) + w_t$$

where $T_t = \beta t$ is the simple linear trend component (no intercept), $S_t = \alpha_1 Q_1(t) + \alpha_2 Q_2(t) + \alpha_3 Q_3(t) + \alpha_4 Q_4(t)$ is the seasonal component, $N_t = w_t$ is the assumed Gaussian white noise sequence.

Since indicator variables $Q_i(t) = 1$ if time t corresponds to quarter $i = 1, 2, 3, 4$, and zero otherwise. So $Q_i(t) = 1$ can be treated as factor variables.

```
trend = time(jj) - 1970    # helps to 'center' time    # mean(t) = 1970.375
Q = factor(cycle(jj)) # make (Q)uarter factors  # Q = 1,2,3,4,1,2,3,4,... (4 levels)
model = lm(log(jj) ~ 0 + trend + Q, na.action = NULL)  # no intercept
#model.matrix(model)  # view the model design matrix
summary(model)   # view the results
```

```
##
## Call:
## lm(formula = log(jj) ~ 0 + trend + Q, na.action = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29318 -0.09062 -0.01180  0.08460  0.27644
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## trend 0.167172   0.002259   74.00   <2e-16 ***
## Q1    1.052793   0.027359   38.48   <2e-16 ***
## Q2    1.080916   0.027365   39.50   <2e-16 ***
## Q3    1.151024   0.027383   42.03   <2e-16 ***
## Q4    0.882266   0.027412   32.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1254 on 79 degrees of freedom
## Multiple R-squared:  0.9935, Adjusted R-squared:  0.9931
## F-statistic:  2407 on 5 and 79 DF,  p-value: < 2.2e-16
```

Result:

$R^2 = 0.9931$ and all the variables are very significant, so the model fits the series data well.

Therefore, the fitted regression model is:

$$\hat{x}_t = 0.167172t + 1.052793Q_1(t) + 1.080916Q_2(t) + 1.151024Q_3(t) + 0.882266Q_4(t)$$

(b)

```
as.numeric(4 * coef(model)['trend'])
```

```
## [1] 0.6686887
```

Suppose the annual logged earnings per share for year T is:

$$X_T = x_T + x_{T+0.25} + x_{T+0.50} + x_{T+0.75} = (4T + 1.5)\hat{\beta} + \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3 + \hat{\alpha}_4$$

Then the annual incease for year $T+1$ is: $X_{T+1} - X_T = 4\hat{\beta}$, which is a constant. So the average of the annual increase is also $4\hat{\beta}$.

As a result, if the model is correct, the estimated average annual increase in the logged earnings per share is about $4\hat{\beta} = 0.6686887$.

(c)

```r
# actual difference for each year
as.numeric(0.25*coef(model)['trend'] + coef(model)['Q4'] - coef(model)['Q3'])
```

```
## [1] -0.2269646
```

```r
# decreasing percentage of the average difference
x = matrix(as.numeric(fitted(model)), ncol=4, byrow=TRUE)
(mean(x[,4])-mean(x[,3])) / mean(x[,3])
```

```
## [1] -0.1838351
```

For year T, the difference of logged earnings per share between the fourth quarter and the third quarter is:

$$x_{T+0.75} - x_{T+0.50} = \hat{\beta}(T+0.75) + \hat{\alpha}_4 - \hat{\beta}(T+0.5) - \hat{\alpha}_3 = 0.25\hat{\beta} + \hat{\alpha}_4 - \hat{\alpha}_3 = -0.2269646$$

which is a constant that is independent of year T. So the the average logged earnings will decrease from the third quarter to the fourth quarter.

But the decrease percentage of does depends on year T. We can compute it and averaging through the observed years and get that:

$$\frac{\bar{x}_{T+0.75} - \bar{x}_{T+0.5}}{\bar{x}_{T+0.5}} = -0.1838351$$

As a result, if the model is correct, the average logged earnings rate will decrease from the third quarter to the fourth quarter by about 18.38% percentage.

(d)

```r
trend = time(jj) - 1970
Q = factor(cycle(jj))
model = lm(log(jj) ~ trend + Q, na.action = NULL)  # include an intercept
summary(model)
```

```
##
## Call:
## lm(formula = log(jj) ~ trend + Q, na.action = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29318 -0.09062 -0.01180  0.08460  0.27644
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.052793   0.027359  38.480  < 2e-16 ***
## trend        0.167172   0.002259  73.999  < 2e-16 ***
## Q2           0.028123   0.038696   0.727   0.4695
## Q3           0.098231   0.038708   2.538   0.0131 *
## Q4          -0.170527   0.038729  -4.403 3.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1254 on 79 degrees of freedom
## Multiple R-squared:  0.9859, Adjusted R-squared:  0.9852
## F-statistic:  1379 on 4 and 79 DF,  p-value: < 2.2e-16
```

```
# view the model design matrix
model.matrix(model)[1:20,]
```

```
##    (Intercept)  trend Q2 Q3 Q4
## 1            1 -10.00  0  0  0
## 2            1  -9.75  1  0  0
## 3            1  -9.50  0  1  0
## 4            1  -9.25  0  0  1
## 5            1  -9.00  0  0  0
## 6            1  -8.75  1  0  0
## 7            1  -8.50  0  1  0
## 8            1  -8.25  0  0  1
## 9            1  -8.00  0  0  0
## 10           1  -7.75  1  0  0
## 11           1  -7.50  0  1  0
## 12           1  -7.25  0  0  1
## 13           1  -7.00  0  0  0
## 14           1  -6.75  1  0  0
## 15           1  -6.50  0  1  0
## 16           1  -6.25  0  0  1
## 17           1  -6.00  0  0  0
## 18           1  -5.75  1  0  0
## 19           1  -5.50  0  1  0
## 20           1  -5.25  0  0  1
```

Result:

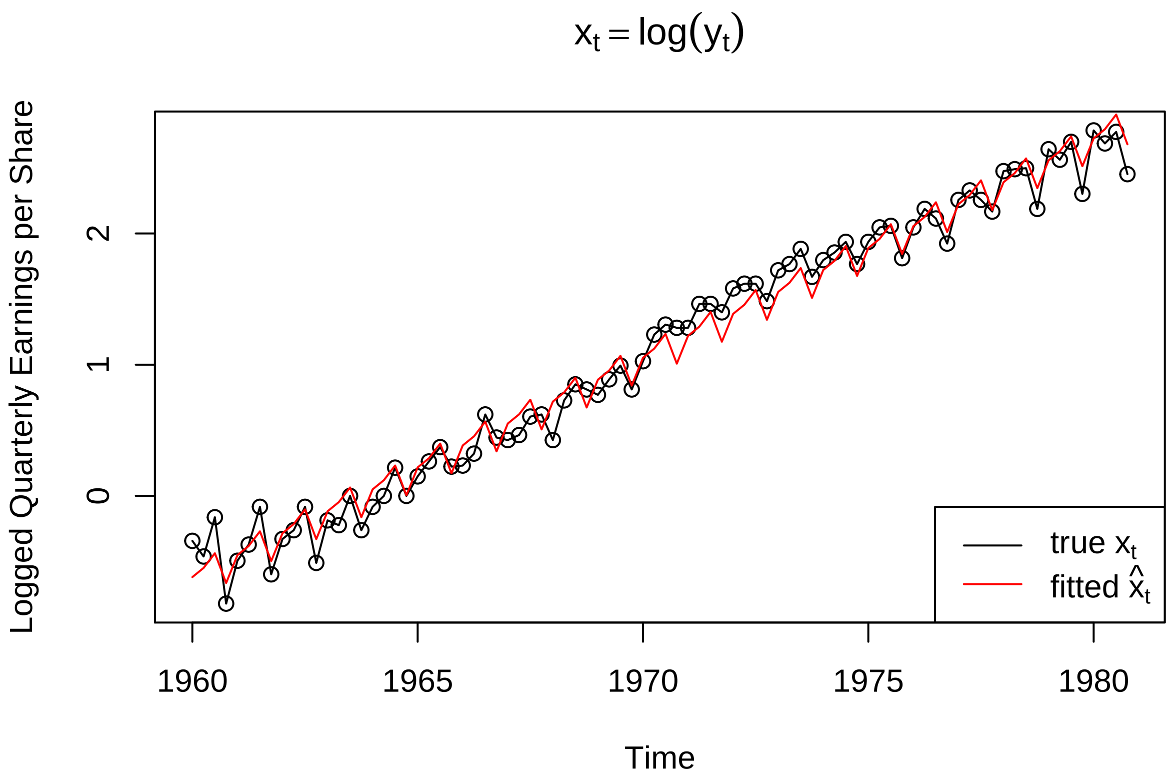
If we include an intercept in the model, the factor Q_1 gets kicked out from the model, and the variable Q_2 becomes not significant, also the variable Q_3 is less significant than it in the previous model. As a result, the $R^2 = 0.9859$, which is also smaller than the previous model.

If we have a look at the design matrix, we can easily find that, when adding the intercept vector $\mathbf{1} = (1, ..., 1)^T$, it will be exactly the sum of the four quarter factor vectors, i.e. $\mathbf{1} = \mathbf{Q_1} + \mathbf{Q_2} + \mathbf{Q_3} + \mathbf{Q_4}$, thus having redundant dummy variables and colinearity problem where a variable is the linear combination of some other variables. In this case, R will drop one of the dummy variable. However, here the Q_1 is actually needed for the model and removing it may cause the model performence gets much worse than before.

(e)

The fitted $\hat{x}_t$ is shown as below:

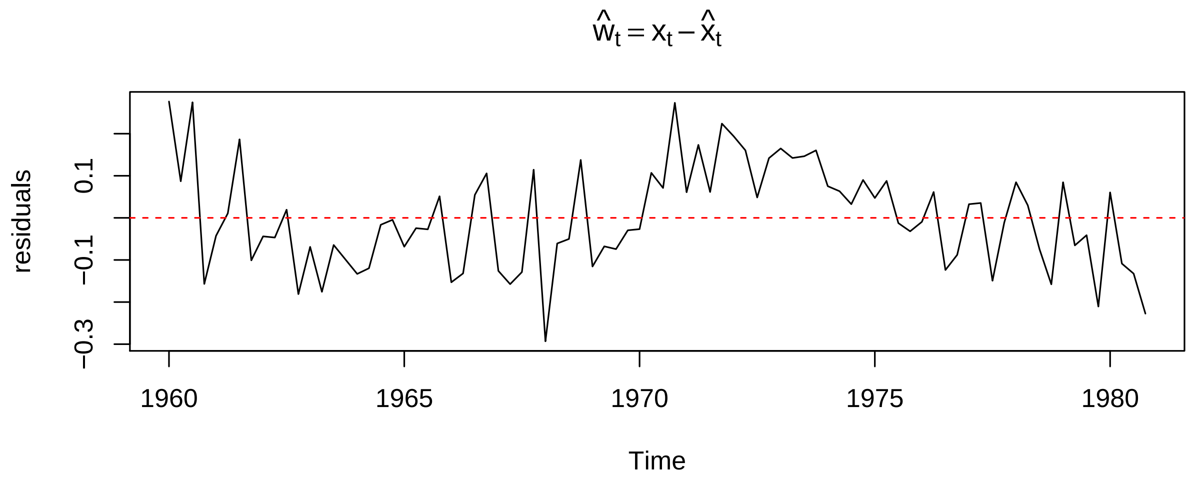
```
plot(log(jj), type='o', ylab='Logged Quarterly Earnings per Share',
                        main=expression(x[t]==log(y[t])))
lines(as.numeric(trend+1970), fitted(model), col=2, lty=1)
legend('bottomright', legend = c(expression(paste('true ',x[t])), expression(paste('fitted ',hat(x)[t]))
```

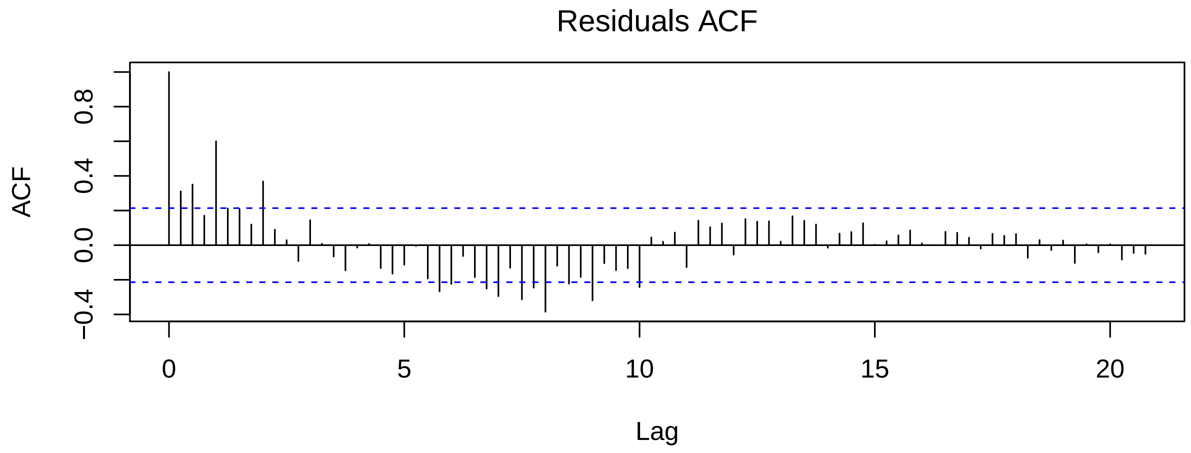


```
#lines(as.numeric(trend+1970), coef(model)['trend']*trend, col=3, lty=1)
#lines(as.numeric(trend+1970), coef(model)['trend']*trend + mean(coef(model)[2:5]), col=4, lty=3)
```

The residuals $x_t - \hat{x}_t$ are shown as below:

```
# residual plot & residual ACF plot
par(mfrow=c(2,1))
plot(resid(model), ylab='residuals', main=expression(hat(w)[t] == x[t] - hat(x)[t]))
abline(h=0, col=2, lty=2)
acf(resid(model), lag.max=100, plot=TRUE, main='')  # max lag = 84-1 = 83
title('Residuals ACF', line=1, font.main=1)
```





```r
# residual ACF value at lag=0
acf(resid(model), lag.max=100, plot=FALSE)[0]
```

```
##
## Autocorrelations of series 'resid(model)', by lag
##
## 0
## 1
```

Comments:

(1) Fitted value plot

From the fitted plot, we can see that the fitted $\hat{x}_t$ can generally match the basic trend of the series data x_t, especially in the middle part of the sequence. However, at the two sides of the sequence, the model does not fit the data well.

(2) Residual plot

The residuals $\hat{w}_t = x_t - \hat{x}_t$. Since w_t is Gaussian white noise sequence, so theoretically, w_t is stationary with constant mean 0, and its actual ACF of is:

$$\rho_w(h) = \frac{\gamma_w(h)}{\gamma_w(0)} = \begin{cases} 1 & if\ h = 0 \\ 0 & if\ h \neq 0 \end{cases}$$

However, from the residual plot, we can see that the residuals fluactuates around 0, but there seems to be some trend, not randomly normal distributed. On the other hand, the ACF plot shows that the sample ACF does equal to 1 when $h = 0$, but is not significantly close to 0 when $h \neq 0$, especially when the time lag is smaller than 10 years, the sample ACF can be quite large for some lags. Therefore, the residual does not looks like Gaussian white noise here.

As a result, in terms of residuals, the model does not fit the data well.

Combining (1) and (2), the model does not fit the data well. So in general, we should not only look at whether the fitted line have described the general trend of the data well. Instead, we should examine the residuals and check if they are stationary sequence and no siginifanct trend in its distribution.

2.3

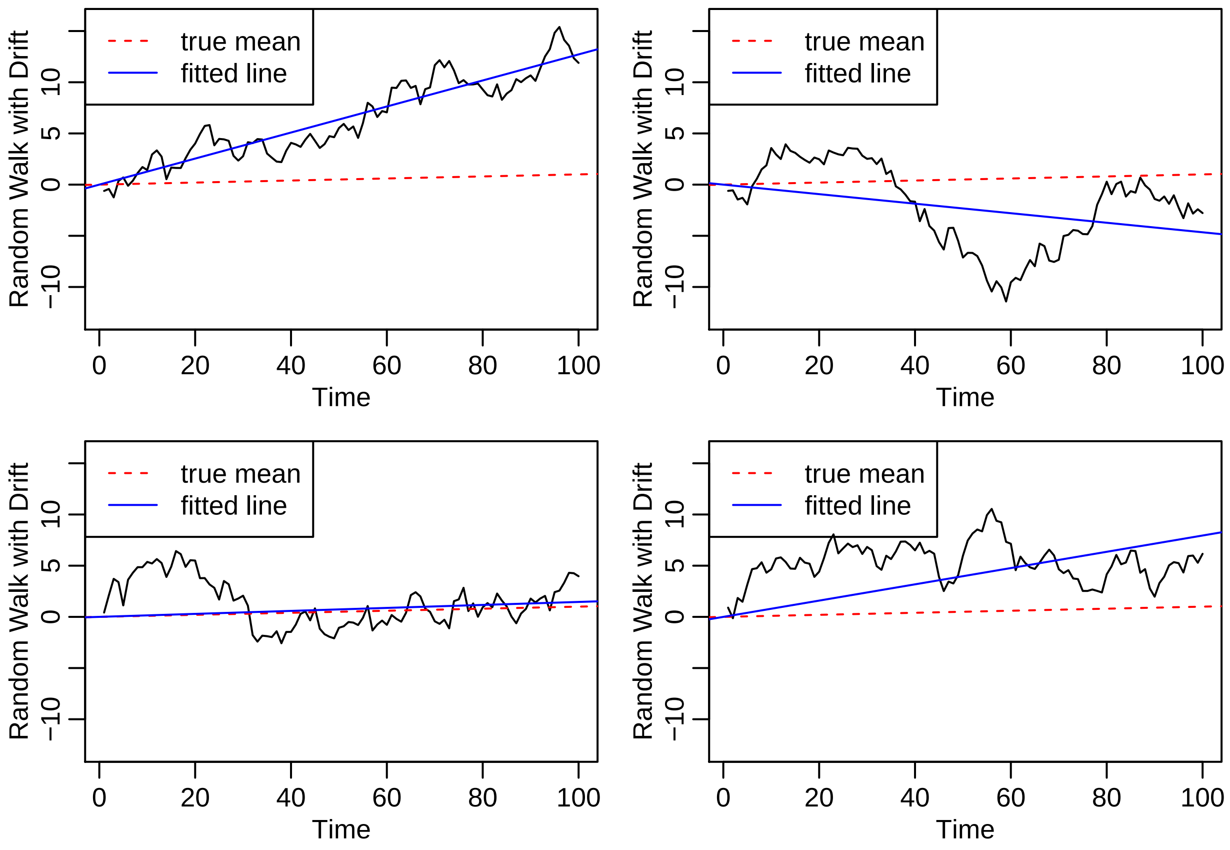
(a)

Random Walk with drift:

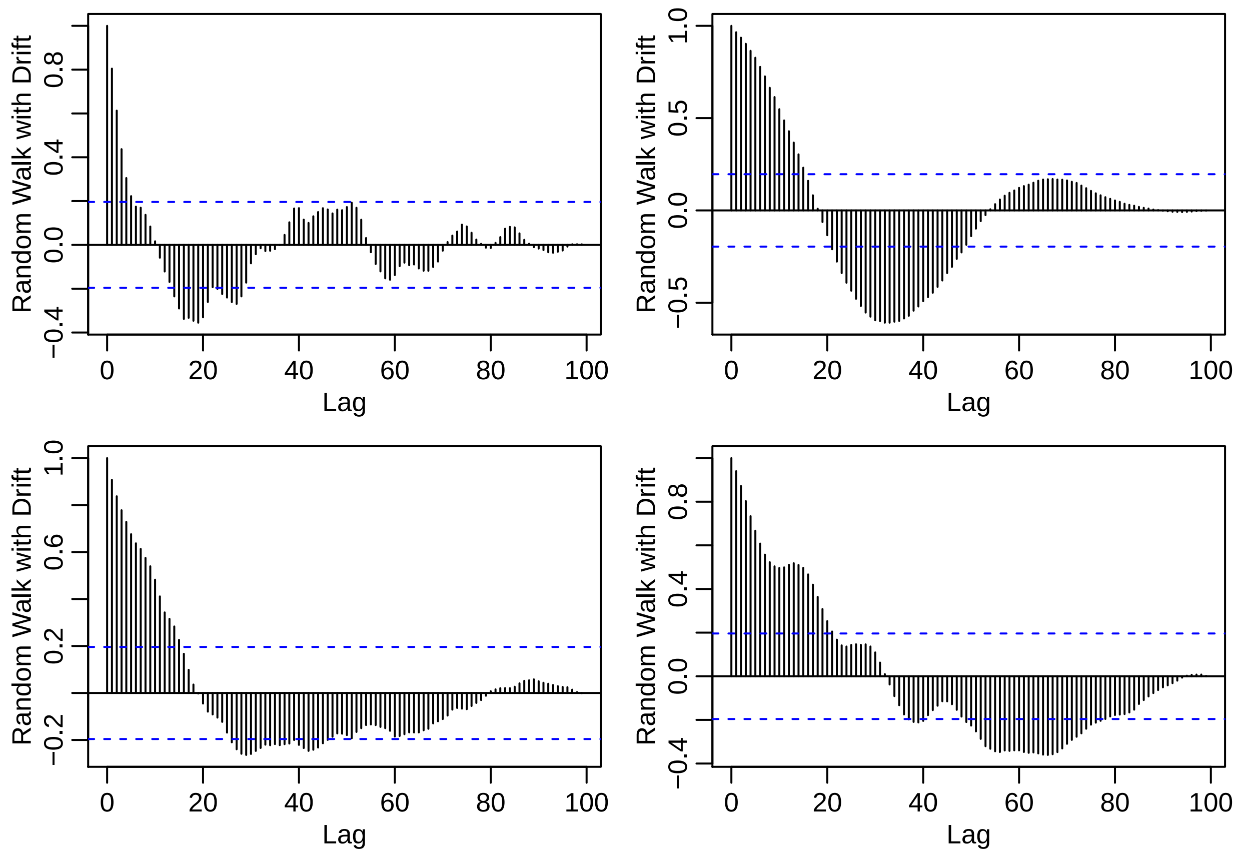
$x_t = \delta + x_{t-1} + w_t$, where initial condition $x_0 = 0$, and w_t is white noise.

So we can rewrite it as: $x_t = \delta t + \sum_{j=1}^{t} w_j$, here the true $\delta = 0.01$.

```r
set.seed(1)
par(mfrow=c(2,2), mar=c(2.5,2.5,0,0)+0.5, mgp=c(1.6,0.6,0)) # set up
for (i in 1:4) {
    x = ts(cumsum(rnorm(100,0.01,1)))       # data x_t (t=1,2,...,100)
    reg = lm(x~0+time(x), na.action=NULL)  # regression
    plot(x, ylab='Random Walk with Drift', ylim=c(-13,16))   # plots
    abline(a=0, b=0.01, col=2, lty=2)       # true mean \delta * t (red-dashed)
    abline(reg, col=4, lty=1)              # fitted line (blue-solid)
    legend('topleft', legend=c('true mean','fitted line'), col=c(2,4), lty=c(2,1))
}
```



```
# residual ACF plot
set.seed(1)
par(mfrow=c(2,2), mar=c(2.5,2.5,0,0)+0.5, mgp=c(1.6,0.6,0))
for (i in 1:4) {
    x = ts(cumsum(rnorm(100,0.01,1)))
    reg = lm(x~0+time(x), na.action=NULL)
    acf(resid(reg), lag.max=100, plot=TRUE, ylab='Random Walk with Drift', main='')
}
```

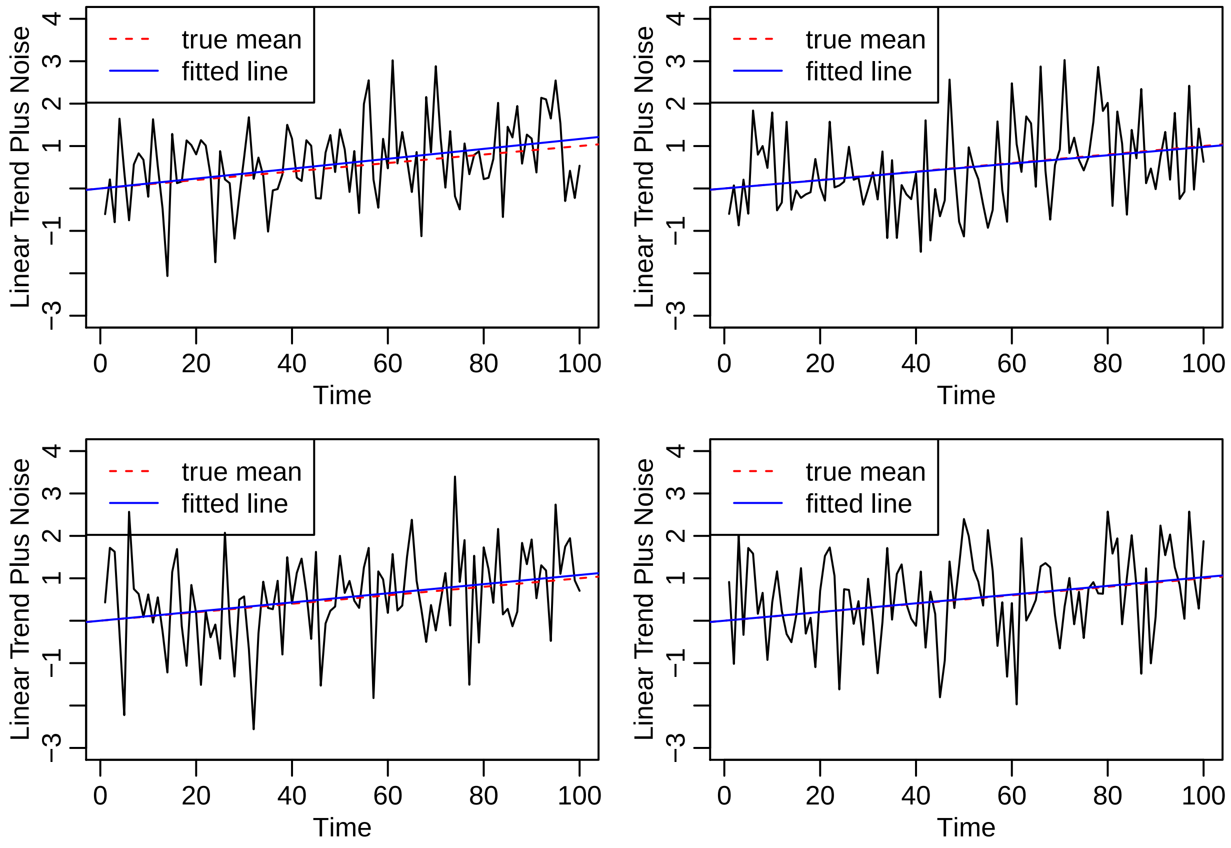


(b)

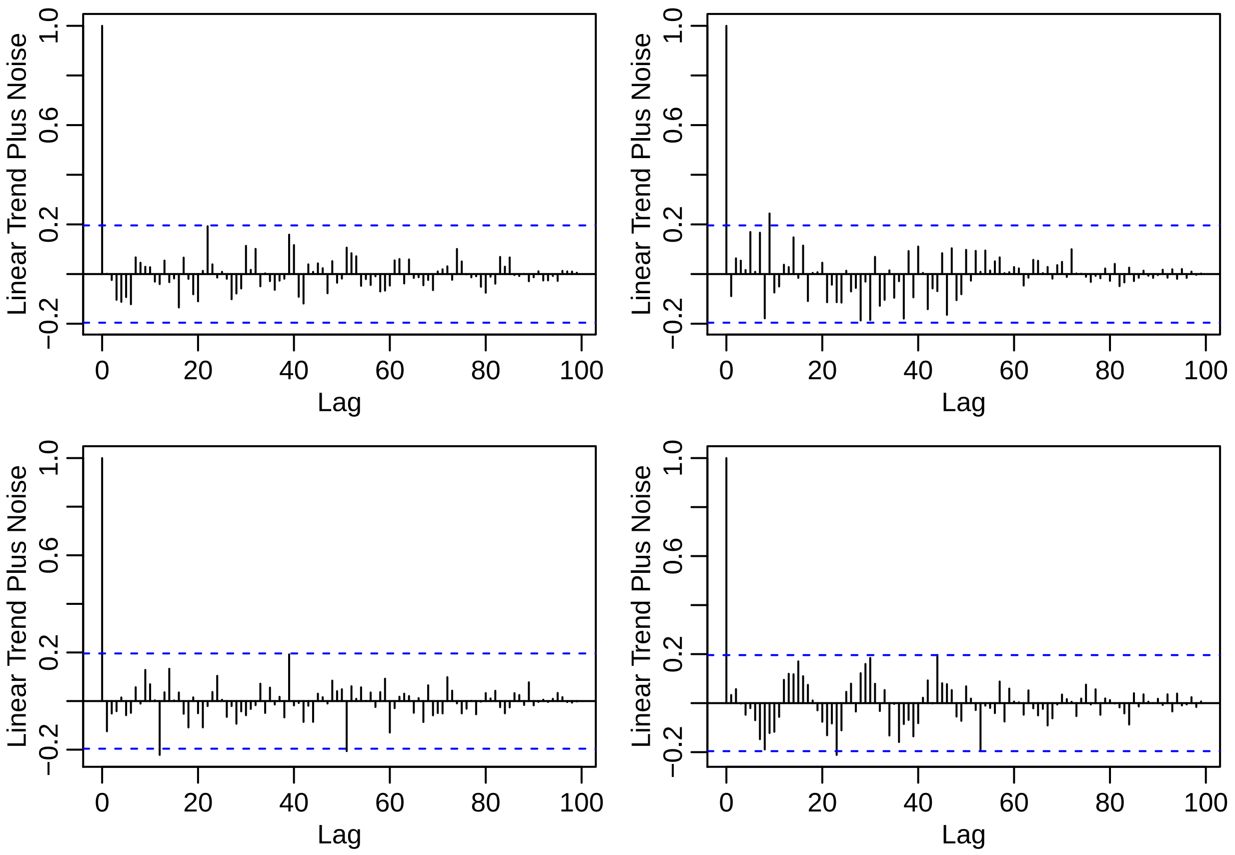
Linear trend plus noise:

$y_t = \beta t + w_t$, where the true $\beta = 0.01$, and w_t is white noise (same as in (a)).

```
set.seed(1)
par(mfrow=c(2,2), mar=c(2.5,2.5,0,0)+0.5, mgp=c(1.6,0.6,0)) # set up
for (i in 1:4) {
    y = ts(0.01*(1:100) + rnorm(100,0.01,1)) # data y_t (t=1,2,...,100)
    reg = lm(y~0+time(y), na.action=NULL)   # regression
    plot(y, ylab='Linear Trend Plus Noise', ylim=c(-3,4))  # plots
    abline(a=0, b=0.01, col=2, lty=2)         # true mean \delta * t (red-dashed)
    abline(reg, col=4, lty=1)                # fitted line (blue-solid)
    legend('topleft', legend=c('true mean','fitted line'), col=c(2,4), lty=c(2,1))
}
```



```r
# residual ACF plot
set.seed(1)
par(mfrow=c(2,2), mar=c(2.5,2.5,0,0)+0.5, mgp=c(1.6,0.6,0))
for (i in 1:4) {
    y = ts(0.01*(1:100) + rnorm(100,0.01,1))
    reg = lm(y~0+time(y), na.action=NULL)
    acf(resid(reg), lag.max=100, plot=TRUE, ylab='Linear Trend Plus Noise', main='')
}
```



(c)

Comments:

In both examples, we are using the linear regression model $x_t = \beta t + w_t$ to fit the data, which assumes that w_t should be a Gaussian white noise. So the true mean function is: $\mu_t = \beta t = 0.01t$

In (b), the linear trend plus noise model is: $y_t = \beta t + w_t$, where w_t is indeed a Gaussian white noise. So we can see that the fitted line matches the true mean fucntion $\mu_t = \beta t = 0.01t$ pretty well. And the residuals ACF plot also shows that the autocorrelation between the white noises are almost within the blue line which means they are not siginificantly different from 0, so the residual does look like white noise here.

However, in (a), the random walk with drift is $x_t = \delta t + \sum_{j=1}^{t} w_j$, where the noise term $\sum_{j=1}^{t} w_j$ is the accumulated sum of the Gaussian white noise, so the noise term $\sum_{j=1}^{t} w_j$ is not iid for different t any more but are autocorrelated. In this case, the linear model assumption is violated. As a result, we can see that the fitted line can be far away from the true mean function: $\mu_t = \delta t = 0.01t$. And the residuals ACF plot can prove that there is high autocorrelation between the residuals, which are significantly different from 0, so the residuals are not assumed Gaussian white noise in this model.

2.4 Kullback-Leibler Information

For Gaussian regression model: $y = Z\beta + \varepsilon$, where ε is iid $N(0, \sigma^2 I_n)$, so vector $y \sim N(Z\beta,\ \Sigma = \sigma^2 I_n)$, and its density function is:

$$f(y;\theta) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \exp\left\{-\frac{1}{2}(y - Z\beta)^T \Sigma^{-1}(y - Z\beta)\right\}$$

$$= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left\{-\frac{1}{2\sigma^2}(y - Z\beta)^T(y - Z\beta)\right\}$$

where $\theta = (\beta^T, \sigma^2)^T$

So the Kullback-Leibler Information is:

$$I(\theta_1, \theta_2) = n^{-1}E_1\left[\log\frac{f(y;\theta_1)}{f(y;\theta_1)}\right] = n^{-1}E_1\left[-\frac{n}{2}\log\frac{\sigma_1^2}{\sigma_2^2} - \frac{1}{2\sigma_1^2}(y - Z\beta_1)^T(y - Z\beta_1) + \frac{1}{2\sigma_2^2}(y - Z\beta_2)^T(y - Z\beta_2)\right]$$

$$= -\frac{1}{2}\log\frac{\sigma_1^2}{\sigma_2^2} - \frac{1}{2n\sigma_1^2}E_1[(y - Z\beta_1)^T(y - Z\beta_1)] + \frac{1}{2n\sigma_2^2}E_1[(y - Z\beta_2)^T(y - Z\beta_2)]$$

Since:

$$E_1[(y - Z\beta_1)^T(y - Z\beta_1)] = E_1(\varepsilon_1^T\varepsilon_1) = n\sigma_1^2$$

and:

$$E_1[(y - Z\beta_2)^T(y - Z\beta_2)] = E_1[(y - Z\beta_1 + Z\beta_1 - Z\beta_2)^T(y - Z\beta_1 + Z\beta_1 - Z\beta_2)]$$

$$= E_1[(y - Z\beta_1)^T(y - Z\beta_1) + (Z\beta_1 - Z\beta_2)^T(y - Z\beta_1) + (y - Z\beta_1)^T(Z\beta_1 - Z\beta_2) + (Z\beta_1 - Z\beta_2)^T(Z\beta_1 - Z\beta_2)]$$

$$= n\sigma_1^2 + 2(Z\beta_1 - Z\beta_2)^T E_1[(y - Z\beta_1)] + (Z\beta_1 - Z\beta_2)^T(Z\beta_1 - Z\beta_2)$$

$$= n\sigma_1^2 + 2(Z\beta_1 - Z\beta_2)^T E_1(\varepsilon_1) + (Z\beta_1 - Z\beta_2)^T(Z\beta_1 - Z\beta_2)$$

$$= n\sigma_1^2 + (Z\beta_1 - Z\beta_2)^T(Z\beta_1 - Z\beta_2)$$

Thus, we have:

$$I(\theta_1, \theta_2) = -\frac{1}{2}\log\frac{\sigma_1^2}{\sigma_2^2} - \frac{1}{2} + \frac{1}{2n\sigma_2^2}[n\sigma_1^2 + (Z\beta_1 - Z\beta_2)^T(Z\beta_1 - Z\beta_2)]$$

$$= \frac{1}{2}\left(\frac{\sigma_1^2}{\sigma_2^2} - \log\frac{\sigma_1^2}{\sigma_2^2} - 1\right) + \frac{(\beta_1 - \beta_2)^T Z^T Z(\beta_1 - \beta_2)}{2n\sigma_2^2}$$

2.6

$$x_t = \beta_0 + \beta_1 t + w_t$$

(a)

The mean function is: $\mu_t = E(x_t) = \beta_0 + \beta_1 t$, which is not a constant and does depend on time t. So the series x_t is nonstationary.

(b)

The first difference series is:

$$\nabla x_t = x_t - x_{t-1} = (\beta_0 + \beta_1 t + w_t) - (\beta_0 + \beta_1(t-1) + w_{t-1}) = \beta_1 + w_t - w_{t-1}$$

Its mean function is: $\mu_t = E(\nabla x_t) = E(\beta_1 + w_t - w_{t-1}) = \beta_1$, which is a constant that is independent of time t.

And its autocorrelation fuction is:

$$\gamma(h) = Cov(\nabla x_{t+h}, \nabla x_t) = Cov(\beta_1 + w_{t+h} - w_{t+h-1},\ \beta_1 + w_t - w_{t-1})$$

$$= 2\gamma_w(h) - \gamma_w(h-1) - \gamma_w(h+1) = \begin{cases} 2\sigma_w^2, & if\ h = 0 \\ -\sigma_w^2, & if\ |h| = 1 \\ 0, & if\ |h| \geq 2 \end{cases}$$

which is only a function of the time lag h and independent of time t.

Therefore, based on these two conditions, the series ∇x_t is stationary.

(c)

If series $x_t = \beta_0 + \beta_1 t + y_t$, where y_t is a stationary process with mean fucntion μ_y and autocovariance function $\gamma_y(h)$.

Then the mean function of the first difference series is:

$$\nabla x_t = x_t - x_{t-1} = (\beta_0 + \beta_1 t + y_t) - (\beta_0 + \beta_1(t-1) + y_{t-1}) = \beta_1 + y_t - y_{t-1}$$

Its mean function is: $\mu_t = E(\nabla x_t) = E(\beta_1 + y_t - y_{t-1}) = \beta_1 + \mu_y - \mu_y = \beta_1$, which is a constant that is independent of time t.

And its autocorrelation fuction is:

$$\gamma(h) = Cov(\nabla x_{t+h}, \nabla x_t) = Cov(\beta_1 + y_{t+h} - y_{t+h-1},\ \beta_1 + y_t - y_{t-1})$$

$$= 2\gamma_y(h) - \gamma_y(h-1) - \gamma_y(h+1)$$

which is only a function of the time lag h and independent of time t.

Therefore, based on these two conditions, the series ∇x_t is also stationary in this case.

2.8

(a)

```r
n = length(varve) # 634
m = ceiling(n/2)  # 317
# diffence in variance
before = c(var(varve[1:m]), var(varve[(m+1):n])); before
```

```
## [1] 133.4574 594.4904
```

```r
(before[2]-before[1])/before[1]
```

```
## [1] 3.454533
```

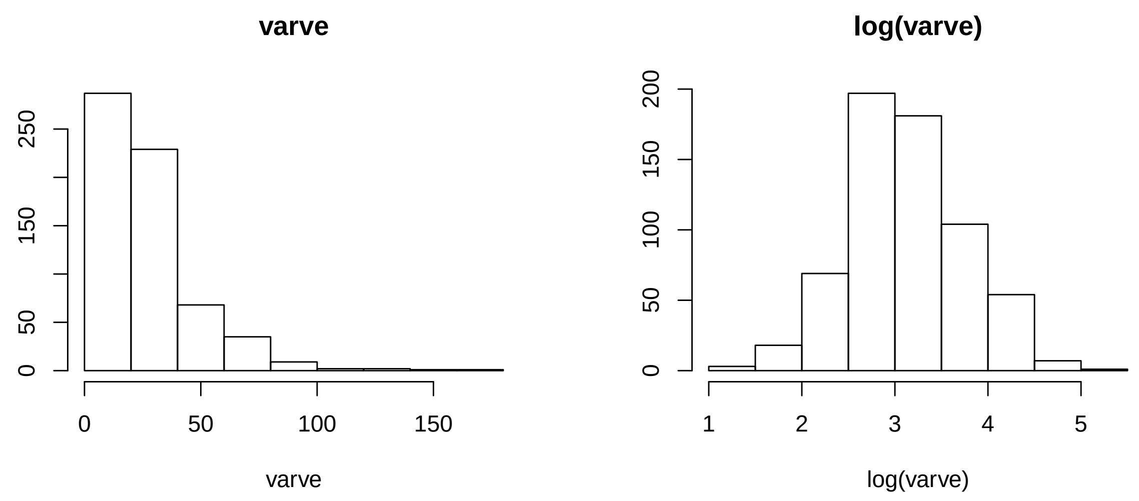
```r
after = c(var(log(varve[1:m])), var(log(varve[(m+1):n]))); after
```

```
## [1] 0.2707217 0.4513710
```

```r
(after[2]-after[1])/after[1]
```

```
## [1] 0.6672882
```

```r
# histogram
par(mfrow=c(1,2))
hist(varve, main="varve", ylab="")
hist(log(varve), main="log(varve)", ylab="")
```



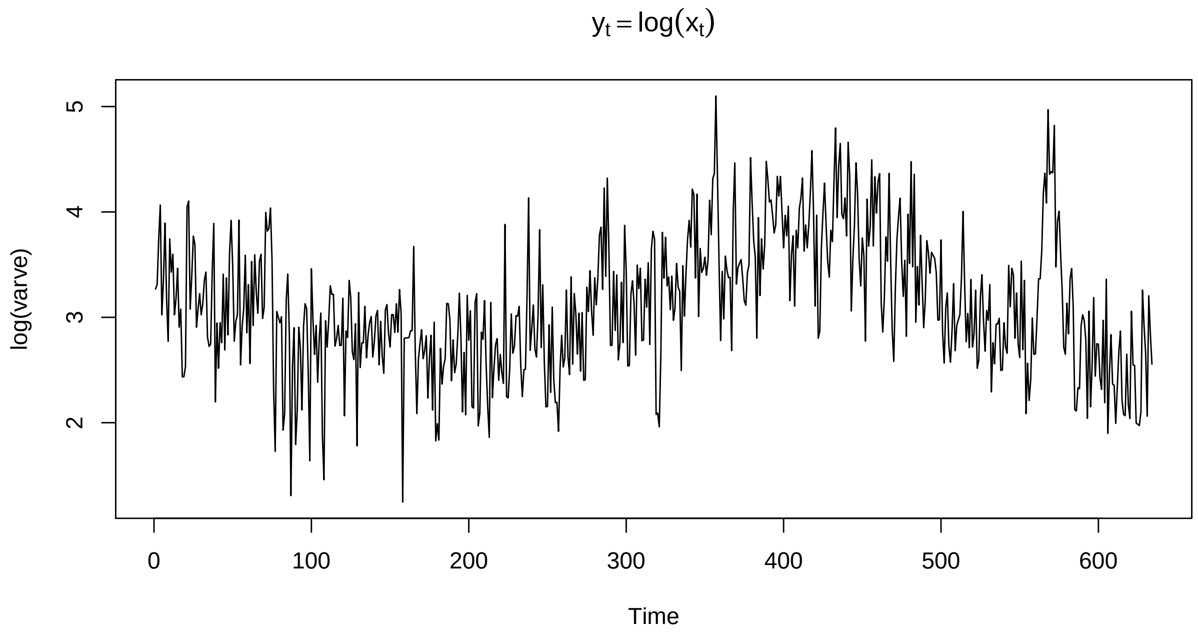
Comments:

Before the transformation, the variance of the second half increases about 345.45% than the first half, which obviously indicate the problem of heteroscedasticity. However, after the log transformation, both half of the sample variance get much smaller, and the difference of the sample variance between the first half and second half gets much smaller as well, only increases about 66.73%. Therefore, the transformation has stabilized the varaince over the series.

The histogram plot also shows that before transformation, the series x_t is very left skewed and not normally distributed at all. However, after the transformation, the series $y_t = \log(x_t)$ is generally normal distributed. Therefore, the transformation has improved the data approximation to normality.

(b)

```r
plot(log(varve), main=expression(y[t] == log(x[t])), ylab="log(varve)")
```



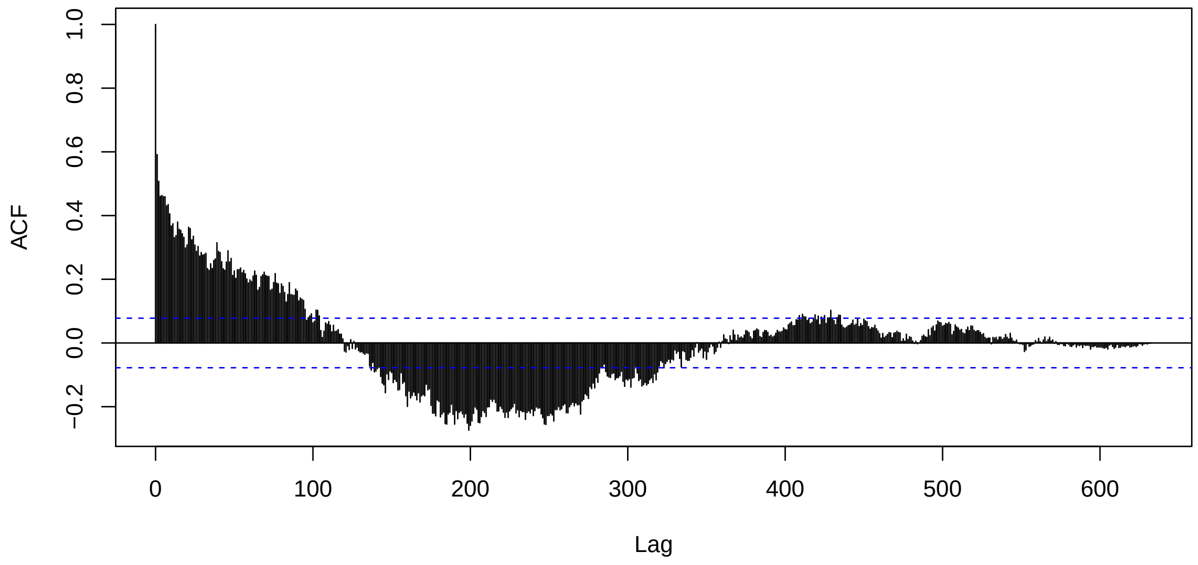
Comments:

In Fig1.2, there is an apparent upward trend in the series during the latter part of the twentieth century that has been used as an argument for the global warming hypothesis.

In our plot, the series y_t also shows an upward trend from about year 200 to year 450, which is similar with the global temperature records.

(c)

```
acf(log(varve), lag.max=650, plot=TRUE, main='')  # max lag = 634-1 = 633
```

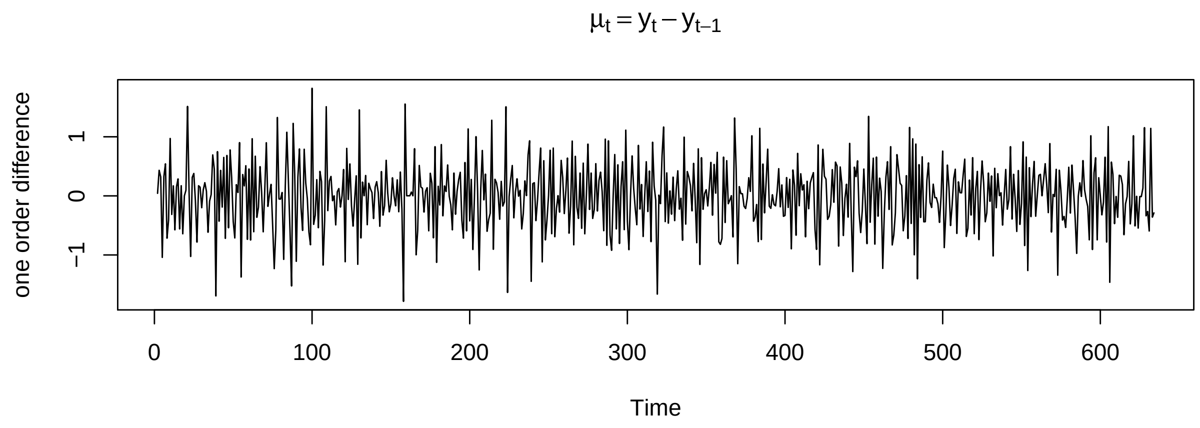


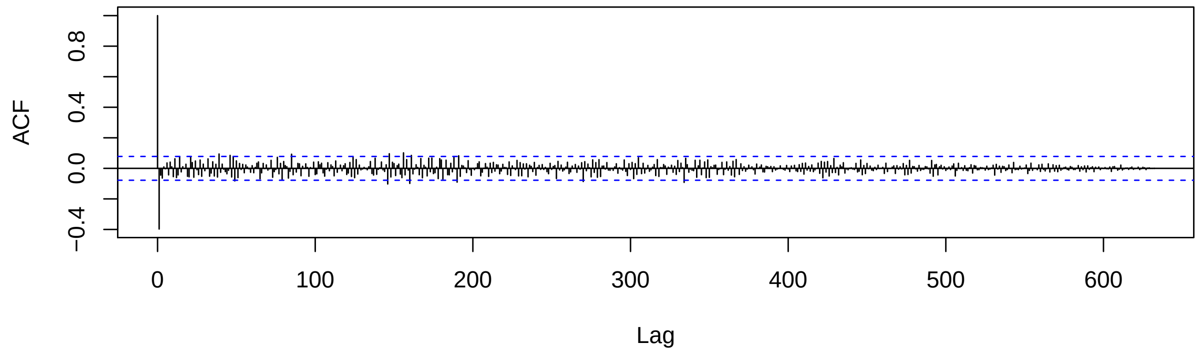
Comments:

When $h = 0$, the sample ACF $= 1$. When h is between about (0, 120), the sample ACF is significantly positive and have a decreasing trend. When h is between about (120, 350), the sample ACF is significantly negative and it first increases from 120 to 200 then decreases from 200 to 350. After the year 350, the sample ACF is not significantly different from 0.

(d)

```
par(mfrow=c(2,1))
# time plot
plot(diff(log(varve), lag=1, differences=1), ylab='one order difference',
     main=expression(mu[t]==y[t]-y[t-1]))
# acf plot and values
acf(diff(log(varve), lag=1, differences=1), lag.max=650, plot=TRUE, main='')  # max lag=633-1=632
```





```
acf(diff(log(varve), lag=1, differences=1), lag.max=650, plot=FALSE)[0:5]
```

```
##
## Autocorrelations of series 'diff(log(varve), lag = 1, differences = 1)', by lag
##
##      0      1      2      3      4      5
##  1.000 -0.397 -0.044 -0.064  0.009 -0.003
```

Comments:

The time plot shows that the mean function of the difference series has a generally constant mean function which does not depend on time t. And the sample ACF plot shows that the autocorrelation function is almost in below format:

$$\rho_\mu(h) = \begin{cases} 1 & if\ h = 0 \\ -0.397 & if\ h = 1 \\ 0 & if\ h \geq 2 \end{cases}$$

Note that acf of difference series is only a function of lag h, especially it has value 0 for everywhere when $h \geq 2$ (uncorrelated).

Therefore, based on above two points, we may say that the order one difference series $\mu_t = y_t - y_{t-1}$ is a reasonably stationary series.

From Footnote 2: $\log(1+p) = p - \frac{p^2}{2} + \frac{p^3}{3} - \cdots$ for $-1 < p < 1$. If p is near zero, the higher-order terms in the expansion are negligible.

So in this question, we have:

$$\mu_t = y_t - y_{t-1} = \log(x_t) - \log(x_{t-1}) = \log\left(\frac{x_t}{x_{t-1}}\right) = \log\left(1 + \frac{x_t - x_{t-1}}{x_{t-1}}\right) \approx \frac{x_t - x_{t-1}}{x_{t-1}}$$

Thus the difference series μ_t can be interpreted as the annual glacial varve thicknesses increasing rate for year t.

(e)

$u_t = \mu + w_t + \theta w_{t-1}$, where w_t are assumed independent with mean 0 and varaince σ^2.

So:

$$\begin{aligned}
\gamma_u(h) = Cov(u_{t+h}, u_t) &= Cov(\mu + w_{t+h} + \theta w_{t+h-1},\ \mu + w_t + \theta w_{t-1}) \\
&= \gamma_w(h) + \theta\gamma_w(h-1) + \theta\gamma_w(h+1) + \theta^2\gamma_w(h) \\
&= (\theta^2+1)\gamma_w(h) + \theta\ (\gamma_w(h-1) + \gamma_w(h+1)) \\
&= \begin{cases} (\theta^2+1)\sigma_w^2 & if\ h = 0 \\ \theta\sigma_w^2 & if\ |h| = 1 \\ 0 & if\ |h| > 1 \end{cases}
\end{aligned}$$

(f)

The autocorrelation function is:

$$\rho_u(h) = \frac{\gamma_u(h)}{\gamma_u(0)} = \begin{cases} 1 & if\ h = 0 \\ \frac{\theta}{\theta^2+1} & if\ |h| = 1 \\ 0 & if\ |h| > 1 \end{cases}$$

So we have:

$$\rho_u(1) = \frac{\theta}{\theta^2+1} \quad \text{and} \quad \gamma_u(0) = (\theta^2+1)\sigma_w^2$$

Therefore, we can use the sample $\hat{\rho}_u(1)$ and $\hat{\gamma}_u(0)$ to estimate $\hat{\theta}$ and $\hat{\sigma}_w^2$:

$$\hat{\rho}_u(1) = \frac{\hat{\theta}}{\hat{\theta}^2+1} \quad \text{and} \quad \hat{\gamma}_u(0) = (\hat{\theta}^2+1)\hat{\sigma}_w^2$$

Sovling these two equations, we can get that:

$$\hat{\theta} = \frac{1 \pm \sqrt{1 - 4\hat{\rho}_u^2(1)}}{2\hat{\rho}_u(1)} \quad \text{and} \quad \hat{\sigma}_w^2 = \frac{2\hat{\rho}_u^2(1)\hat{\gamma}_u(0)}{1 \pm \sqrt{1 - 4\hat{\rho}_u^2(1)}} = \frac{\hat{\rho}_u(1)\hat{\gamma}_u(0)}{\hat{\theta}}$$

```
# compute
u = diff(log(varve), lag=1, differences=1)
rho_1 = acf(u, lag.max=650, plot=FALSE)$acf[2]; rho_1
```

```
## [1] -0.3974306
```

```
gamma_0 = var(u); gamma_0
```

```
## [1] 0.3322131
```

```r
# estimate
theta1 = (1 - sqrt(1-4*rho_1^2)) / (2*rho_1)
theta2 = (1 + sqrt(1-4*rho_1^2)) / (2*rho_1)
sigmas1 = rho_1*gamma_0 / theta1
sigmas2 = rho_1*gamma_0 / theta2
c(theta1, sigmas1)
```

```
## [1] -0.4946886  0.2668986
```

```r
c(theta2, sigmas2)
```

```
## [1] -2.02147378  0.06531456
```

Therefore, the estimated values are: $\hat{\theta} = -0.4946886$ and $\hat{\sigma}_w^2 = 0.2668986$, or $\hat{\theta} = -2.02147378$ and $\hat{\sigma}_w^2 = 0.06531456$.

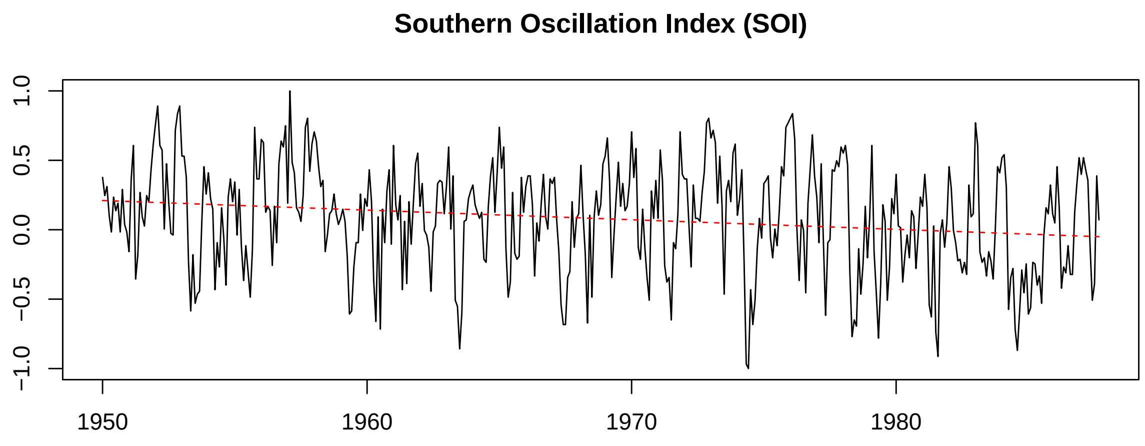
2.9

(a) Linear Regression Detrending

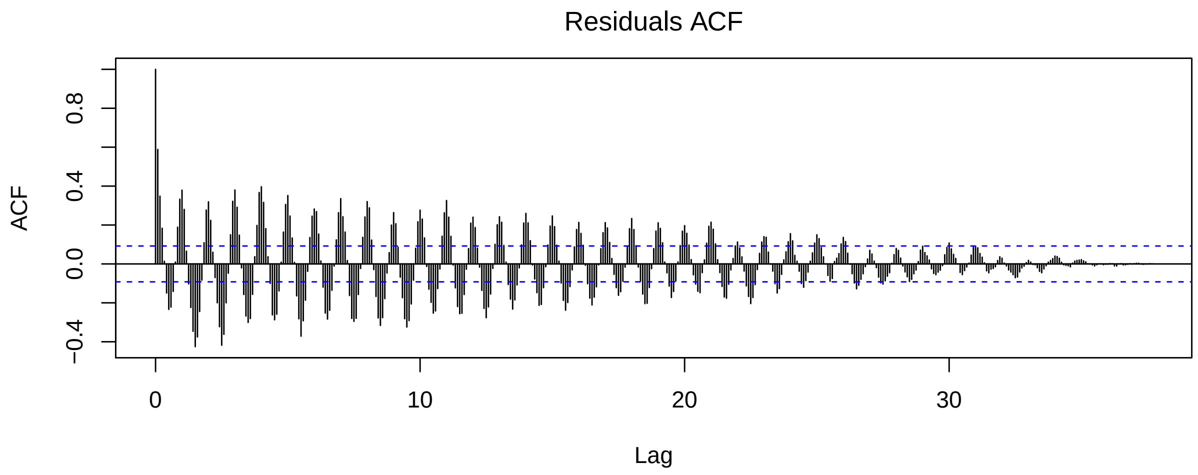
```r
# model
model = lm(soi ~ time(soi), na.action=NULL)
summary(model)
```

```
##
## Call:
## lm(formula = soi ~ time(soi), na.action = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04140 -0.24183  0.01935  0.27727  0.83866
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.70367    3.18873   4.298 2.12e-05 ***
## time(soi)   -0.00692    0.00162  -4.272 2.36e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3756 on 451 degrees of freedom
## Multiple R-squared:  0.0389, Adjusted R-squared:  0.03677
## F-statistic: 18.25 on 1 and 451 DF,  p-value: 2.359e-05
```

```r
# plot
# 453 months over years 1950-1987
plot(soi, ylab="", xlab="", main="Southern Oscillation Index (SOI)")
lines(fitted(model), col=2, lty=2)
```



```
# residucal ACF
acf(resid(model), lag.max=500, plot=TRUE, main='')  # max lag = 453-1 = 452
title('Residuals ACF', line=1, font.main=1)
```



Comments:

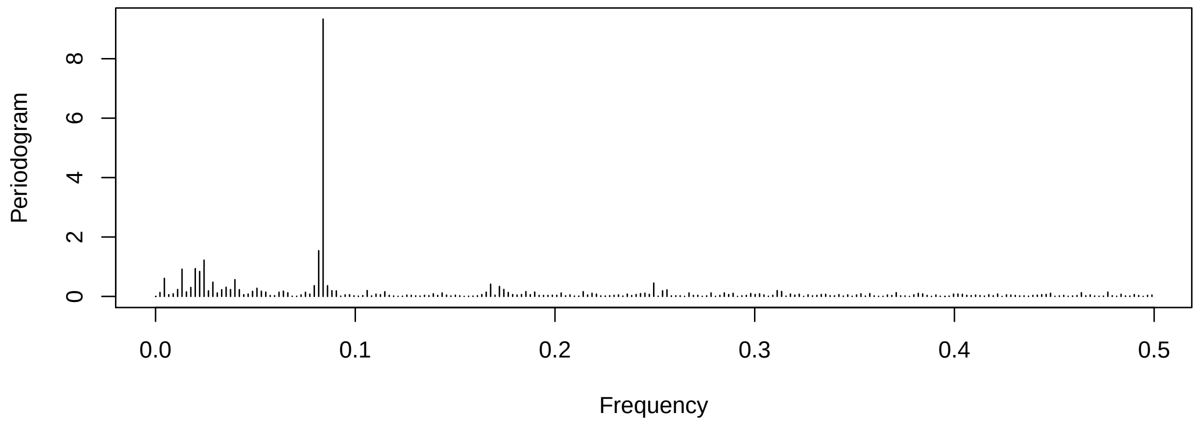
Fit the regression model: $S_t = \beta_0 + \beta_1 t + w_t$, we get the fitted model is: $\hat{S}_t = 13.70367 - 0.00692t$, and the coefficient for time t is very significant. Thus we can say that there is a significant decreasing trend in the sea surface temperature.

However, by examing the residuals ACF, the residuals are significantly different from 0 and have a obvious periodic cycle. From this perspective, the regression model have not described the data substantially, so after regression detrending, we need to further detrend the cyclic component as well.

(b) Periodogram

```
x = resid(model)  # the residuals that are detrended from part (a)
n = length(x)   # 453
per = Mod(fft(x))^2 / n
```

```
m = ceiling(n/2)       # mirror effect at the folding frequency of 1/2
freq = (1:n - 1)/n
plot(freq[1:m], per[1:m], type='h', xlab='Frequency', ylab='Periodogram')
```



```
# two main peaks in periodogram
y = cbind(1:m, per[1:m], freq[1:m], 1/freq[1:m])
colnames(y) = c('n', 'per', 'freq', 'months/cycle')
y[order(y[,'per'], decreasing=TRUE)[1:10], ]
```

```
##        n       per        freq months/cycle
## 39    39 9.3342138 0.083885210     11.92105
## 38    38 1.5403252 0.081677704     12.24324
## 12    12 1.2221961 0.024282561     41.18182
## 10    10 0.9352039 0.019867550     50.33333
## 7      7 0.9176151 0.013245033     75.50000
## 11    11 0.8401284 0.022075055     45.30000
## 3      3 0.6088403 0.004415011    226.50000
## 19    19 0.5645641 0.039735099     25.16667
## 14    14 0.4788602 0.028697572     34.84615
## 114  114 0.4489265 0.249448124      4.00885
```

Result:

The two main peaks are:

(1) Highest: the periodogram is about 9.3342138, its frequency is at about 0.083885210, which means about one cycle every $1/0.083885210 \approx 11.92105 \approx 12$ months. This is the annual cycle.

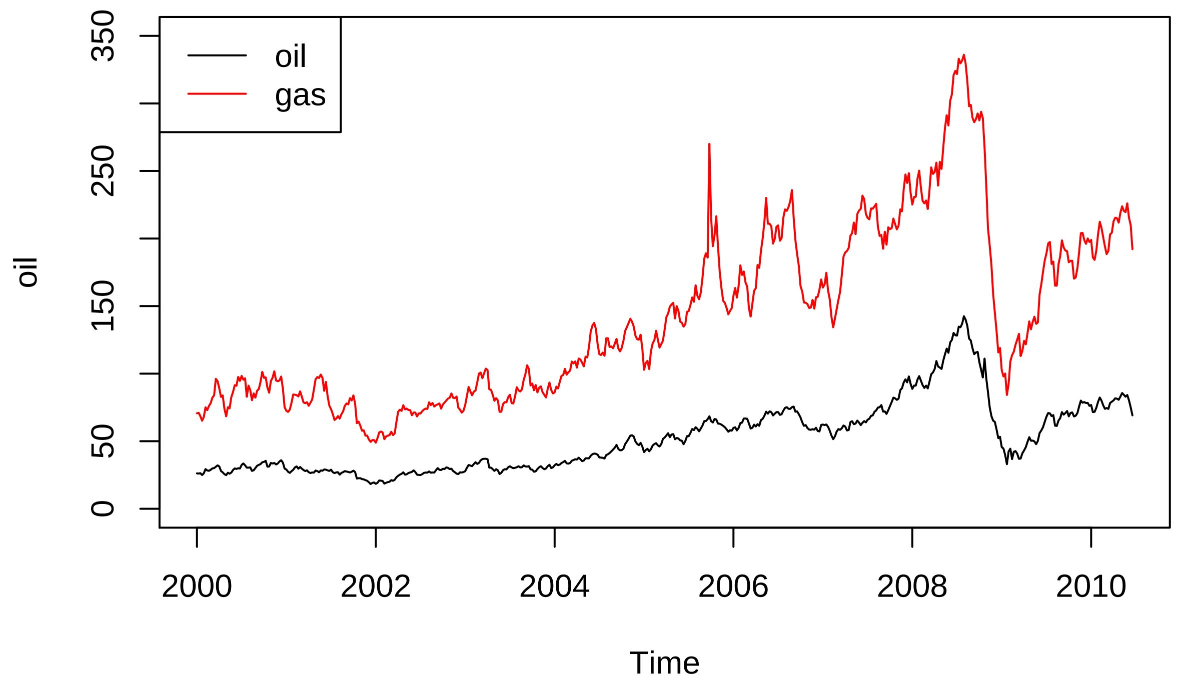
(2) Second highest: the periodogram is about 1.2221961, frequency is at about 0.024282561, which means about one cycle every $1/0.024282561 \approx 41.18182 \approx 41$ months. So the EL Nino cycle is about 41 months/cycle.

2.10

(a)

```
plot(oil, col=1, lty=1, ylim=c(0,350))
lines(gas, col=2, lty=1)
```

```r
legend("topleft", legend=c("oil","gas"), col=c(1,2), lty=1)
```



Comments:

These two series both looks like Random Walk with Drift model.

The two series are not seem to be stationary because there seems to have the heteroscedasticity problem, where the variance of the second half is larger than the variance of the first half. Also, there is an increasing trend in both series which indicates nonstationary series.

(b)

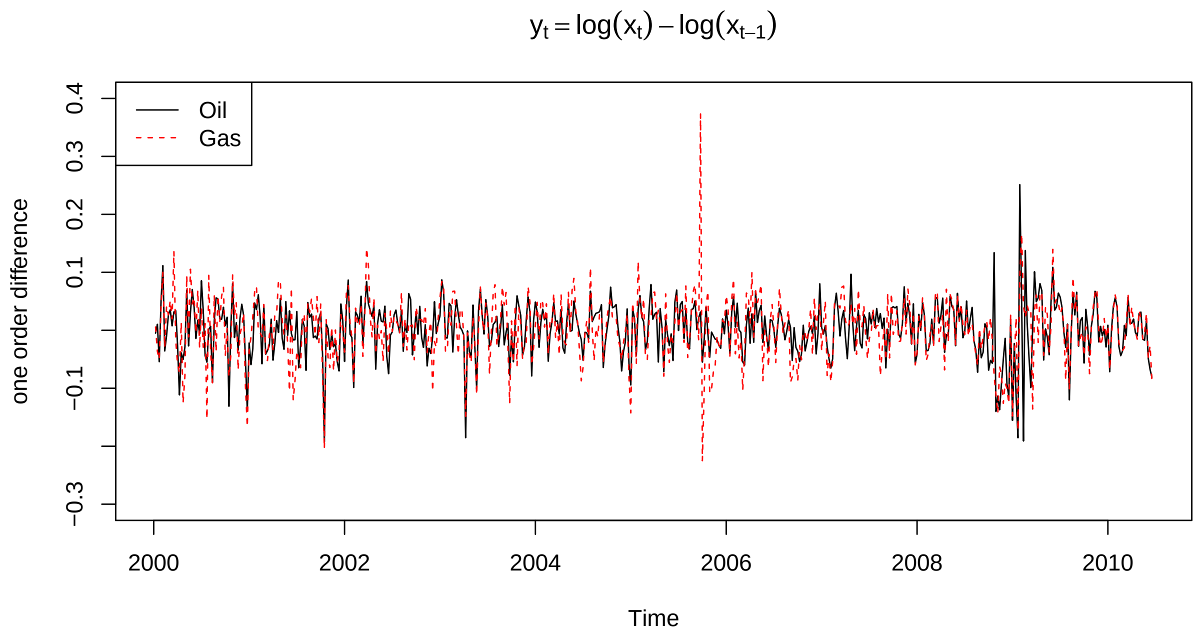
As what is discussed in question 2.8(d), here we have that:

$$y_t = \nabla \log(x_t) = \log(x_t) - \log(x_{t-1}) = \log\left(\frac{x_t}{x_{t-1}}\right) = \log\left(1 + \frac{x_t - x_{t-1}}{x_{t-1}}\right) \approx \frac{x_t - x_{t-1}}{x_{t-1}}$$

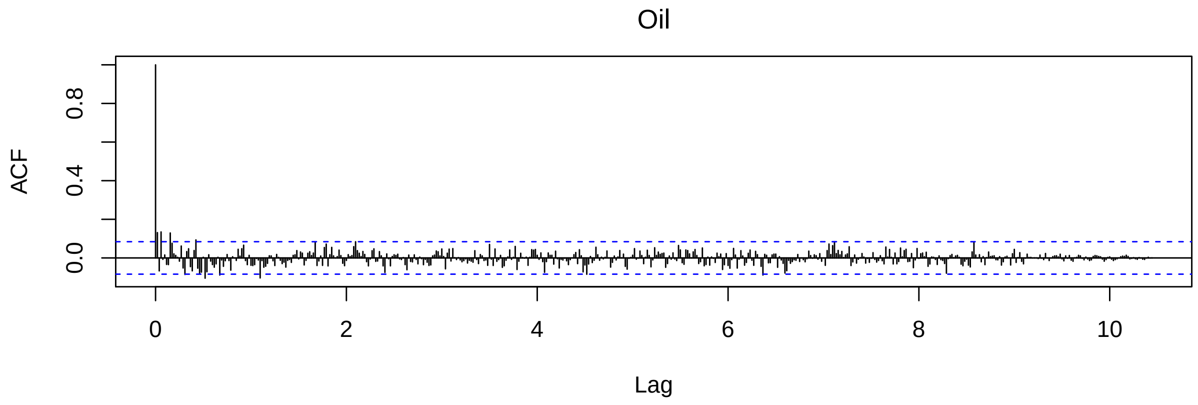
So the series y_t can be interpreted as the percentage change in price for time t, and it can be applied to the data to convert the original nonstationary series x_t into a stationary series.

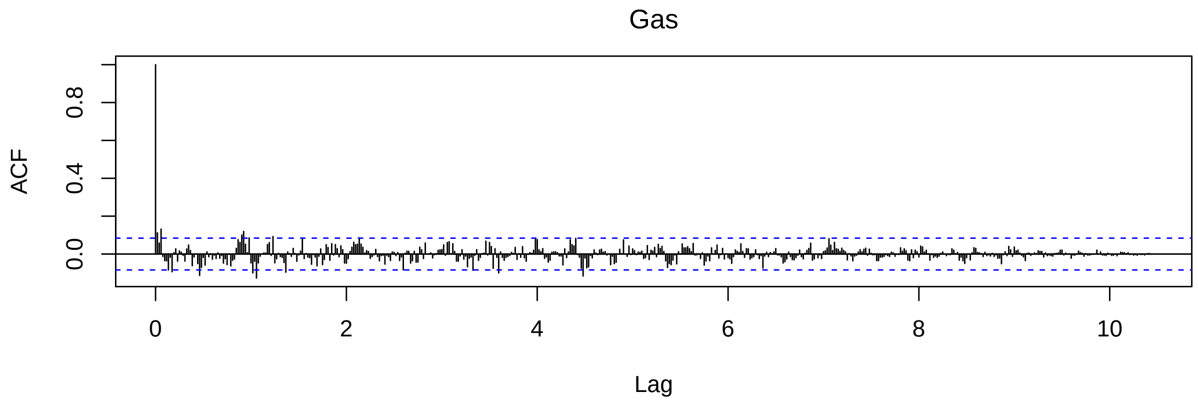
(c)

```r
# time plot
plot(diff(log(oil), lag=1, differences=1), ylab='one order difference',
     main=expression(y[t]==log(x[t])-log(x[t-1])), col=1, lty=1, ylim=c(-0.3,0.4))
lines(diff(log(gas), lag=1, differences=1), col=2, lty=2)
legend("topleft", legend=c("Oil","Gas"), col=c(1,2), lty=c(1,2))
```



```
# acf plots
par(mfrow=c(2,1))
acf(diff(log(oil), lag=1, differences=1), lag.max=length(oil), plot=TRUE, main='')
title('Oil', line=1, font.main=1)
acf(diff(log(gas), lag=1, differences=1), lag.max=length(gas), plot=TRUE, main='')
title('Gas', line=1, font.main=1)
```





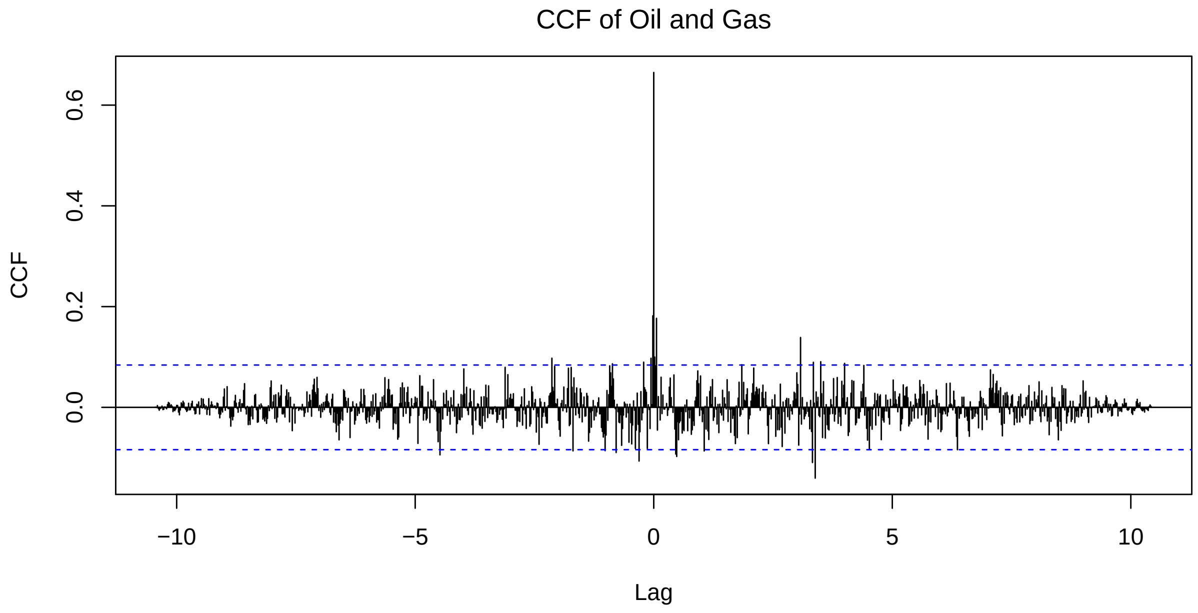
Comments:

The time plot shows that the mean function of the difference series has a generally constant mean function which does not depend on time t. And the sample ACF plot shows that the autocorrelation function is almost zero except for at $h = 0$.

Therefore, based on above two points, we may say that the order one difference series $y_t = \nabla \log(x_t)$ is a reasonably stationary series.

(d)

```
ccf(diff(log(oil)), diff(log(gas)), lag.max=max(length(oil),length(gas)), main='', ylab='CCF')
title('CCF of Oil and Gas', line=1, font.main=1)
```

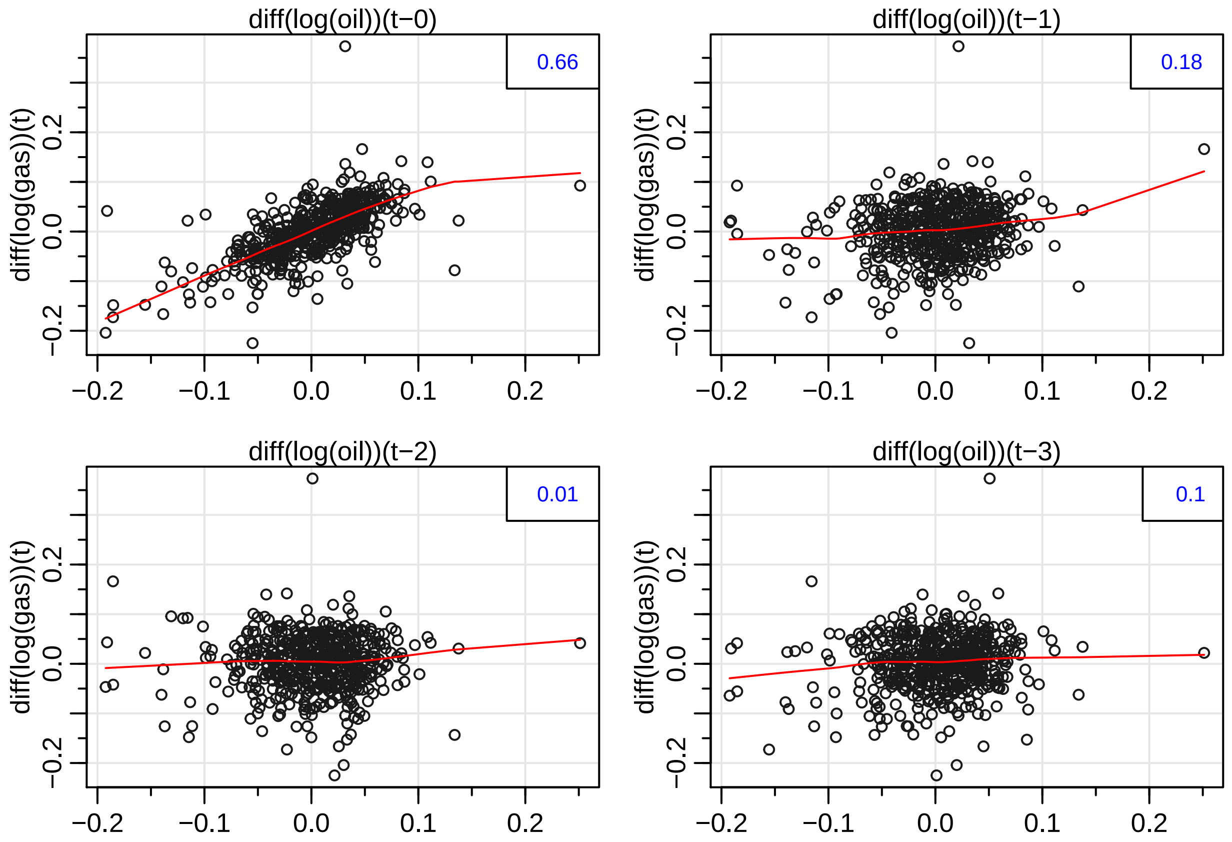


Comments:

The cross-correlation function (CCF) $\rho_{xy}(h)$ here for gas and oil, we have a large $\rho_{xy}(0)$ at lag $h = 0$, and positive cross-correlation values around $h = 0$. So we can get that the cross-covarance $\gamma_{xy}(0) = Cov(x_{t+h}, y_t)_{h=0} = Cov(x_t, y_t)$ is large for all time t. This indicates that the price of gas and oil has a strong correlation at any time t, and this correlation may expand to time lag in a short period of time.

(e)

```
lag2.plot(diff(log(oil)), diff(log(gas)), max.lag=3)
```



Comments:

We can see that when $h = 0$, there is a obvious linear relationship between y_t for oil and gas. when $h = 1$, there is only a small but negligible linear relationship left. when $h > 1$, there is no correlation any more.

On the other hand, from all the fout plots, we can observe an outlier which always affects the regression fitting results.

(f)

(i)

```
poil = diff(log(oil))
pgas = diff(log(gas))
indi = ifelse(poil<0, 0, 1)
mess = ts.intersect(pgas, poil, poilL=lag(poil,-1), indi)  # bind two time series
model = lm(pgas ~ indi + poil + poilL, data=mess)
summary(model)
```

```
##
## Call:
## lm(formula = pgas ~ indi + poil + poilL, data = mess)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18451 -0.02161 -0.00038  0.02176  0.34342
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006445   0.003464  -1.860  0.06338 .
## indi         0.012368   0.005516   2.242  0.02534 *
## poil         0.683127   0.058369  11.704  < 2e-16 ***
## poilL        0.111927   0.038554   2.903  0.00385 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04169 on 539 degrees of freedom
## Multiple R-squared:  0.4563, Adjusted R-squared:  0.4532
## F-statistic: 150.8 on 3 and 539 DF,  p-value: < 2.2e-16
```

```
coef(model)[1] + coef(model)[2]
```

```
## (Intercept)
##  0.00592331
```

Comments:

All the variables are significant. However, $R^2 = 0.4563$ indicates that the model does not fit the data very well.

The fitted model is:

$$\hat{G}_t = -0.006445 + 0.012368 I_t + 0.683127 Q_t + 0.111927 Q_{t-1}$$

(ii)

When there is negative growth in oil price at time t, the indicator $I_t = 0$, so the fitted model is:

$$\hat{G}_t = -0.006445 + 0.683127 Q_t + 0.111927 Q_{t-1}$$

When there is no or positive growth in oil price at time t, the indicator $I_t = 1$, so the fitted model is:

$$\hat{G}_t = -0.006445 + 0.012368 + 0.683127 Q_t + 0.111927 Q_{t-1}$$

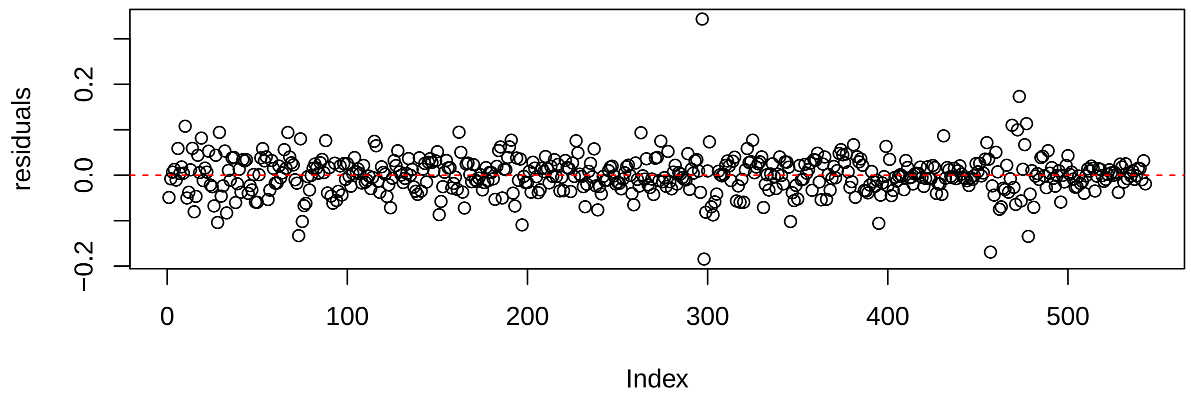
$$= 0.00592331 + 0.683127 Q_t + 0.111927 Q_{t-1}$$

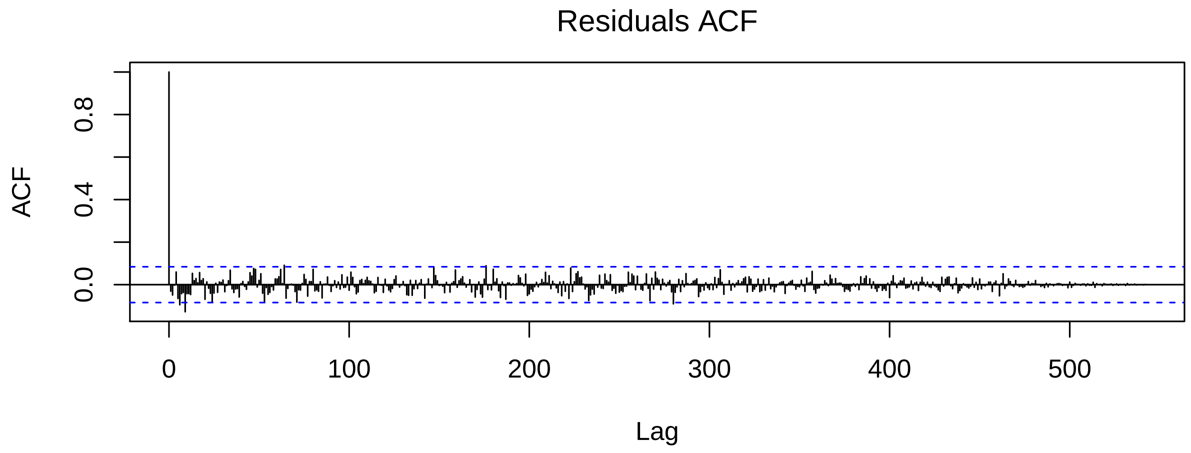
We can see that the two models have the same slopes for both Q_t and Q_{t-1}, the only difference is the intercept. And there is also no big difference in the absolute value of the intercepts. In this case, I think these results seems not to support the asymmetry hypothesis.

(iii)

The residuals $\hat{w}_t = G_t - \hat{G}_t$ are shown as below:

```
# residual plot & residual ACF plot
par(mfrow=c(2,1))
plot(resid(model), ylab='residuals')
abline(h=0, col=2, lty=2)
acf(resid(model), lag.max=length(oil), plot=TRUE, main='')  # max lag = 84-1 = 83
title('Residuals ACF', line=1, font.main=1)
```





```r
# residual ACF value at lag=0
acf(resid(model), lag.max=length(oil), plot=FALSE)[0:6]
```

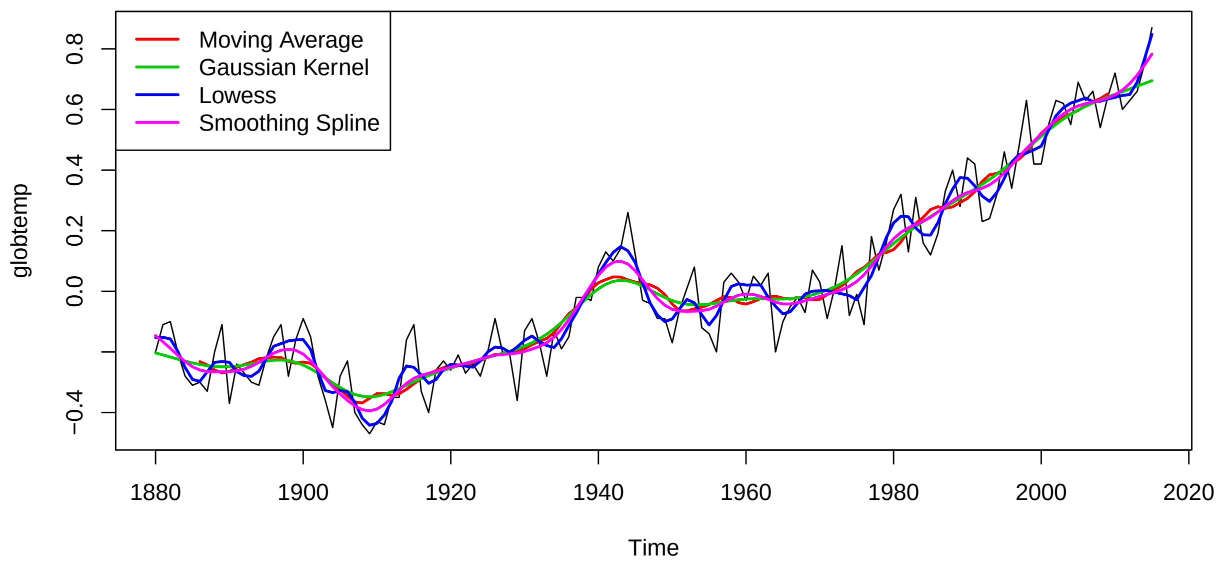
```
##
## Autocorrelations of series 'resid(model)', by lag
##
##      0      1      2      3      4      5      6
##  1.000 -0.031 -0.049 -0.003  0.059 -0.065 -0.095
```

Comments:

The residuals seems to randomly distribute around 0 without significant treand. And the ACF of residuals are close to 0 execpt for at $h = 0$. These together indicates that the residuals does look like a white noise.

2.11

```r
# plot data
plot(globtemp)

# Moving Average Smoother
# this particular weights help removes the obvious annual temperature cycle
# and helps emphasize the El Nino cycle
wgts = c(0.5, rep(1,11), 0.5)/12
temp_f = filter(globtemp, sides=2, filter=wgts)
lines(temp_f, lwd=2, col=2)

# Kernel Smoothing
lines(ksmooth(time(globtemp), globtemp, "normal", bandwidth=12), lwd=2, col=3)

# Lowess
lines(lowess(globtemp, f=0.05), lwd=2, col=4)  # emphasize El Nino cycle

# Smoothing Splines
lines(smooth.spline(globtemp, spar=0.5), lwd=2, col=6) # emphasize El Nino cycle

legend('topleft', legend=c('Moving Average','Gaussian Kernel','Lowess','Smoothing Spline'), col=c(2,3,4
```



Comments:

In this plot, the Moving Average Smoother with a particular weight over 12 two-side data points, and the Gaussian Kernel Smoothing with bandwidth 12 are performing in a similar way, which makes the series much more smoother.

On the other hand, the Lowess method using the smoother span of f=0.05 of the data, and the Smoothing Splines method using the smoothing parameter of spar=0.5, which both emphasize the El Nino cycle, are both not as smooth as the first two methods, the smoothing line is more irregular. However, they catch more data details than the first two methods. Plus, the Smoothing Splines method here is a little more smoother than the Lowess method.

From this question, we can learn that different methods have their own cons and pros, it depends on your specific need to determine which method to use. And for each method, there are some smoothing parameters that you can tune to describe the data better, being more smoother or catching more details.