Chapter 3 HW

Orly Olbum

3.1

(Structural Regression Model). For the Johnson & Johnson data, say yt, shown in Figure 1.1, let xt = log(yt). In this problem, we are going to fit a special type of structural model, xt = Tt + St + Nt where Tt is a trend component, St is a seasonal component, and Nt is noise. In our case, time t is in quarters (1960.00, 1960.25, . . .) so one unit of time is a year.

(a) Fit the regression model xt = Bt + a1Q1(t) + a2Q2(t) + a3Q3(t) + a4Q4(t) + wt xt = trend + seasonal + noise where Qi(t) = 1 if time t corresponds to quarter i = 1, 2, 3, 4, and zero otherwise. The Qi(t)'s are called indicator variables. We will assume for now that wt is a Gaussian white noise sequence. Hint: Detailed code is given in Appendix A, near the end of Section A.5.

```
yt = jj
xt = log(yt)
trend = time(yt) - 1970
Q = factor(cycle(yt))
reg = lm(xt ~ 0 + trend + Q, na.action = NULL)
summary(reg)
```

```
##
## Call:
## lm(formula = xt ~ 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
```

(b) If the model is correct, what is the estimated average annual increase in the logged earnings per share?

Using the model diagnostics, the average annual increase in logged earnings per share is $ 0.16.

(c) If the model is correct, does the average logged earnings rate increase or decrease from the third quarter to the fourth quarter? And, by what percentage does it increase or decrease?

Assuming a correct model, the logged earnings rate decreases from the third to the fourth quarter, by 23.5%.

(d) What happens if you include an intercept term in the model in (a)? Explain why there was a problem.

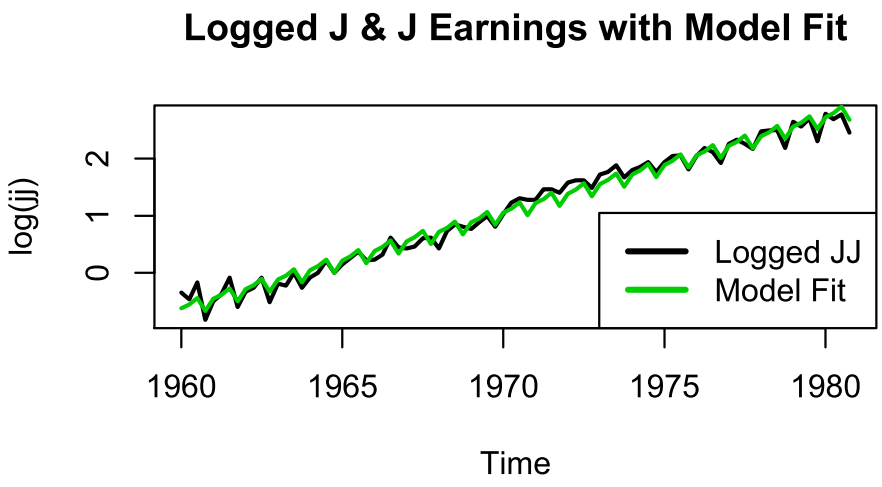
```
reg2 = lm(xt ~ trend + Q, na.action = NULL)
summary(reg2)
```

```
##
## Call:
## lm(formula = xt ~ 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
```

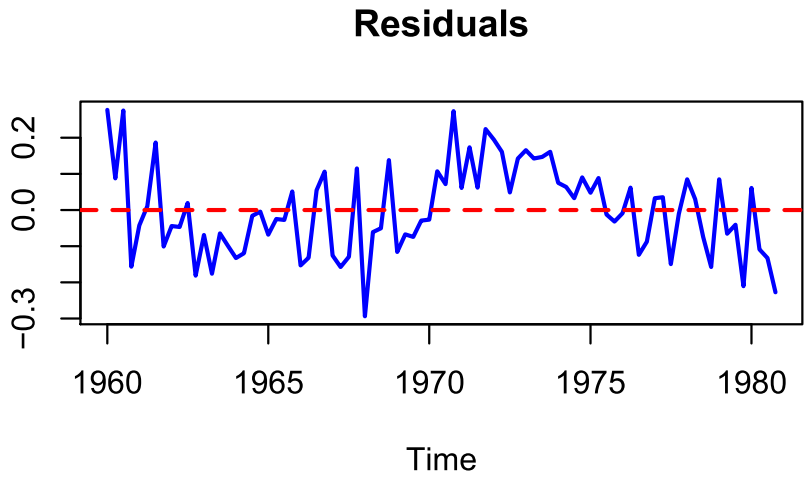
If we include an intercept in the model, we lose the Q1 trend and the estimates do not align correctly.

(e) Graph the data, xt, and superimpose the fitted values, say xt-hat, on the graph. Examine the residuals, xt - xt-hat, and state your conclusions. Does it appear that the model fits the data well (do the residuals look white)?

```
par(mfrow = c(1, 1))
plot(xt, lwd = 2, main = "Logged J & J Earnings with Model Fit", ylab = "log(jj)")
lines(fitted(reg), col = 3, lwd = 2)
legend("bottomright", legend = c("Logged JJ", "Model Fit"), lty = 1, lwd = 3,
       col = c(1, 3), bg = "white")
```




```
plot(resid(reg), col = 4, lwd = 2, main = "Residuals", ylab = "")
abline(h = 0, lty = "dotted", col = 2, lwd = 2)
```



The fitted values look very close to the actual (logged) data, and the residuals all lie around 0 with no obvious trend (maybe slightly cyclical...), indicating a good model fit.

3.2

For the mortality data examined in Example 3.5:

(a) Add another component to the regression in (3.17) that accounts for the particulate count four weeks prior; that is, add Pt-4 to the regression in (3.17). State your conclusion.

```
temp = tempr - mean(tempr)
temp2 = temp^2
```

```
trend = time(cmort)
partL4 = lag(part, -4)

ded = ts.intersect(cmort, trend, temp, temp2, part, partL4)

fit = lm(cmort ~ trend + temp + temp2 + part + partL4, data = ded, na.action = NULL)
summary(fit)
```

```
##
## Call:
## lm(formula = cmort ~ trend + temp + temp2 + part + partL4, data = ded,
##     na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.228  -4.314  -0.614   3.713  27.800
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.808e+03  1.989e+02  14.123  < 2e-16 ***
## trend       -1.385e+00  1.006e-01 -13.765  < 2e-16 ***
## temp        -4.058e-01  3.528e-02 -11.503  < 2e-16 ***
## temp2        2.155e-02  2.803e-03   7.688 8.02e-14 ***
## part         2.029e-01  2.266e-02   8.954  < 2e-16 ***
## partL4       1.030e-01  2.485e-02   4.147 3.96e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.287 on 498 degrees of freedom
## Multiple R-squared:  0.608,  Adjusted R-squared:  0.6041
## F-statistic: 154.5 on 5 and 498 DF,  p-value: < 2.2e-16
```

```
summary(aov(fit))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## trend         1  10280   10280  260.05  < 2e-16 ***
## temp          1   8610    8610  217.81  < 2e-16 ***
## temp2         1   3476    3476   87.92  < 2e-16 ***
## part          1   7493    7493  189.53  < 2e-16 ***
## partL4        1    680     680   17.20 3.96e-05 ***
## Residuals   498  19687      40
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the regression with the lagged part term added, the p-value is significant and all terms show to be significant predictors for cmort.

(b) Using AIC and BIC, is the model in (a) an improvement over the final model in Example 3.5?

```
dedprior = ts.intersect(cmort, trend, temp, temp2, part)
fitprior = lm(cmort ~ trend + temp + temp2 + part, data = dedprior, na.action = NULL)
num = length(cmort)
AIC(fitprior)/num - log(2*pi); BIC(fitprior)/num - log(2*pi)
```

```
## [1] 4.721732
```

```
## [1] 4.771699
```

```
AIC(fit)/num - log(2*pi); BIC(fit)/num - log(2*pi)
```

```
## [1] 4.641492
```

```
## [1] 4.699677
```

The new AIC and BIC are both lower than the prior model, indicating a better model fit for the data.

3.4 Consider linear trend, additive noise, independent random variables w_t $(E(w_t)=0,\ var(w_t)=\sigma_w^2)$

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

β_0 & β_1 are fixed constants.

ⓐ Prove that x_t is nonstationary

$E(x_t) = E(\beta_0 + \beta_1 t + w_t) = \beta_0 + \beta_1 t$

Since the mean of x_t relies on t, we have already violated the first requirement of stationarity, so x_t is nonstationary.

ⓑ Prove that first difference series $\nabla x_t = x_t - x_{t-1}$ is stationary by finding mean and autocovariance function

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

$E(\nabla x_t) = E(\beta_1 + w_t - w_{t-1}) = \beta_1$ (not dependent on t!)

$\gamma(t+h, t) = cov(\nabla x_{t+h}, \nabla x_t) = E[(\nabla x_{t+h} - E(\nabla x_{t+h}))(\nabla x_t - \beta_1)]$

$= E[(\nabla x_{t+h} - \beta_1)(\nabla x_t - \beta_1)]$

$= E(\nabla x_{t+h} \nabla x_t - \beta_1 \nabla x_t - \beta_1 \nabla x_{t+h} + \beta_1^2)$

$= E(\nabla x_{t+h} \nabla x_t) - \beta_1 E(\nabla x_t) - \beta_1 E(x_{t+h}) + \beta_1^2$

$= E(\nabla x_{t+h} \nabla x_t) - \beta_1^2 - \beta_1^2 + \beta_1^2 = E(\nabla x_{t+h} \nabla x_t) - \beta_1^2$

And if $\nabla x_{t+h} \perp\!\!\!\perp \nabla x_t$, $E(\nabla x_{t+h} \nabla x_t) = E(\nabla x_{t+h}) E(\nabla x_t)$

$= \beta_1^2 - \beta_1^2 = 0$

which satisfies both requirements of stationarity!

ⓒ Repeat ⓑ, with $w_t \leftrightarrow y_t$ (stationary), $E(y_t) = \mu_y$, $acv(y_t) = \gamma_y(h)$

For $x_t = \beta_0 + \beta_1 t + y_t$

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

→ next page

3.4 ⓒ, Continued

$\nabla x_t = \beta_1 + y_t - y_{t-1}$ $\quad E(y_t) = \mu_y$

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

$= \beta_1 + \mu_y - \mu_y = \beta_1$ (not dependent on t!)

$\gamma(h) = \text{cov}(\nabla x_{t+h}, \nabla x_t) = E[(\nabla x_{t+h} - E(\nabla x_{t+h}))(\nabla x_t - \beta_1)]$

$E(\nabla x_{t+h}) = E(\beta_1 + y_{t+h} - y_{(t+h)-1}) = \beta_1$

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

$= 2\text{cov}(y_{t+h}) - \text{cov}(y_{t+1}) - \text{cov}(y_{t-1})$

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

$= 2\gamma_y(h) - 2\gamma_y(1) \quad = 0$ for $h = 1, -1$

making $\nabla x_t = \beta_1 + y_t - y_{t-1}$ stationary

3.6

The glacial varve record plotted in Figure 3.9 exhibits some nonstationarity that can be improved by transforming to logarithms and some additional nonstationarity that can be corrected by differencing the logarithms.

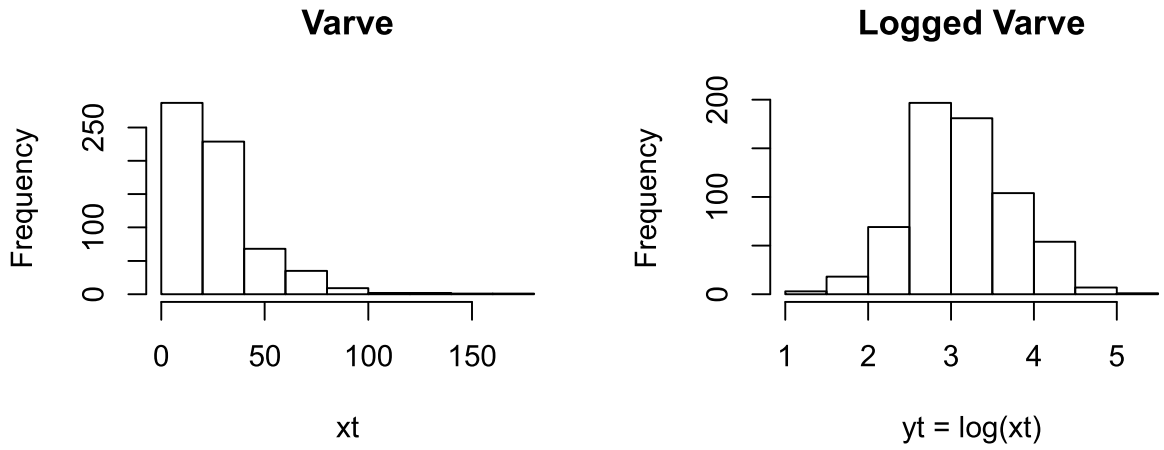
(a) Argue that the glacial varves series, say xt, exhibits heteroscedasticity by computing the sample variance over the first half and the second half of the data. Argue that the transformation yt = log(xt) stabilizes the variance over the series. Plot the histograms of xt and yt to see whether the approximation to normality is improved by transforming the data.

```
xt = varve
yt = log(varve)

n = length(varve)
varve1 = varve[1:n/2]
varve2 = varve[(n/2 + 1):n]
firstvar = var(varve1)
secondvar = var(varve2)
firstvar; secondvar
```

```
## [1] 132.501
```

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

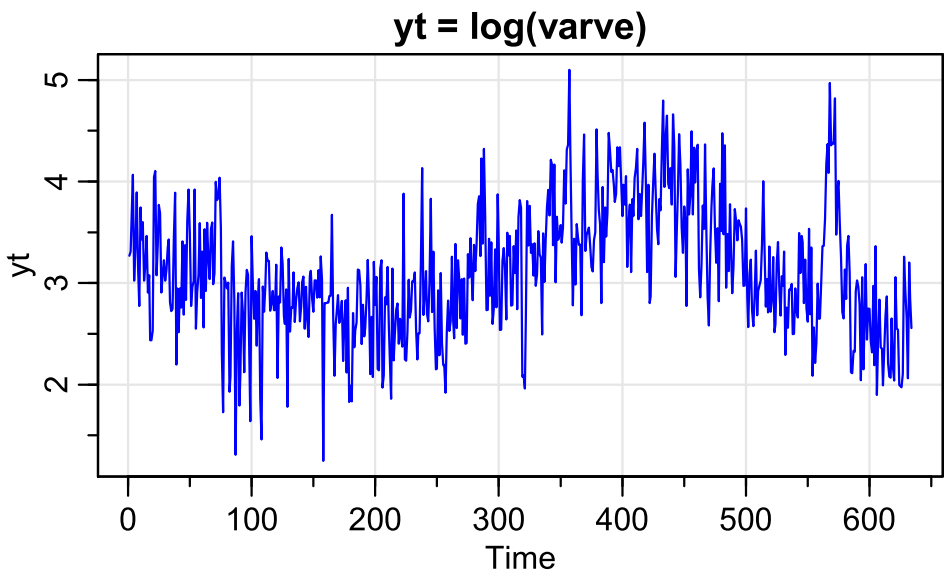
```
par(mfrow = c(1, 2))
hist(xt, main = "Varve")
hist(yt, main = "Logged Varve", xlab = "yt = log(xt)")
```



The variance in the second half of the varve data is much larger than the variance exhibited in the first half, indicating non-homogeneous variance. We need to transform the data to smooth it by taking the natural log. The plot of xt shows a trend between thickness and amount deposited, and logging the data removes this as seen in the plot of yt = log(xt). The histograms show evidence that the log(varve) data has been normalized.

(b) Plot the series yt. Do any time intervals, of the order 100 years, exist where one can observe behavior comparable to that observed in the global temperature records in Figure 1.2?

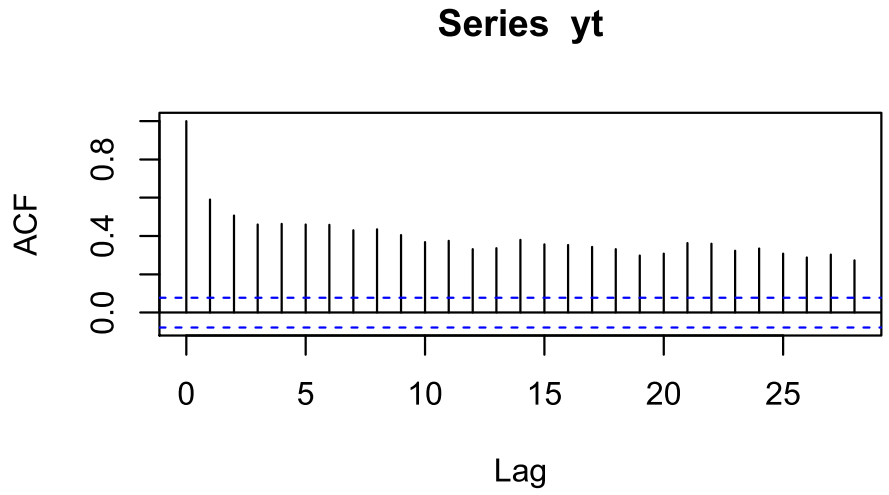
```r
par(mfrow = c(1, 1))
tsplot(yt, main = "yt = log(varve)", col = 4, margin = 0)
```



From time about 150 to 400, the data looks like Figure 1.2 - a steady increase with some variation from the overall trend.

(c) Examine the sample ACF of yt and comment.

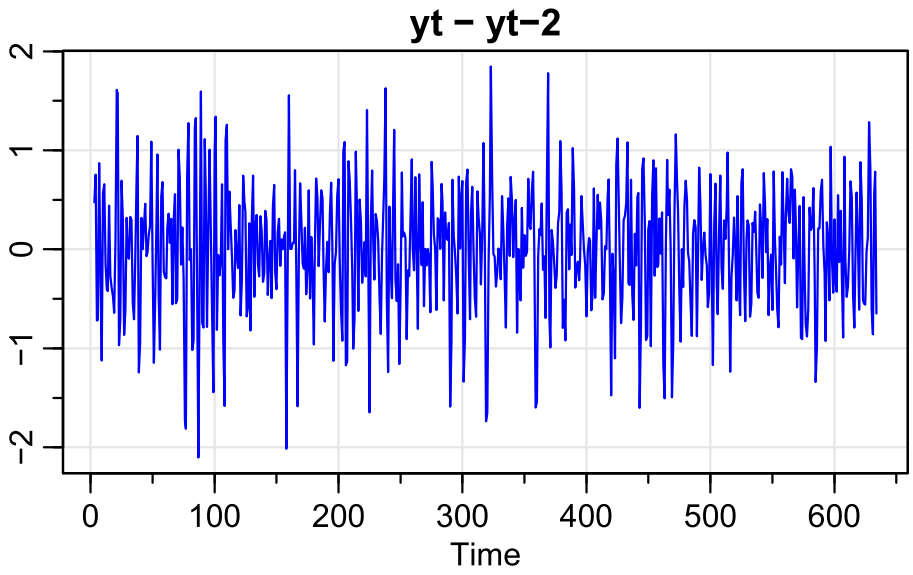
```r
acf(yt, plot = TRUE)
```



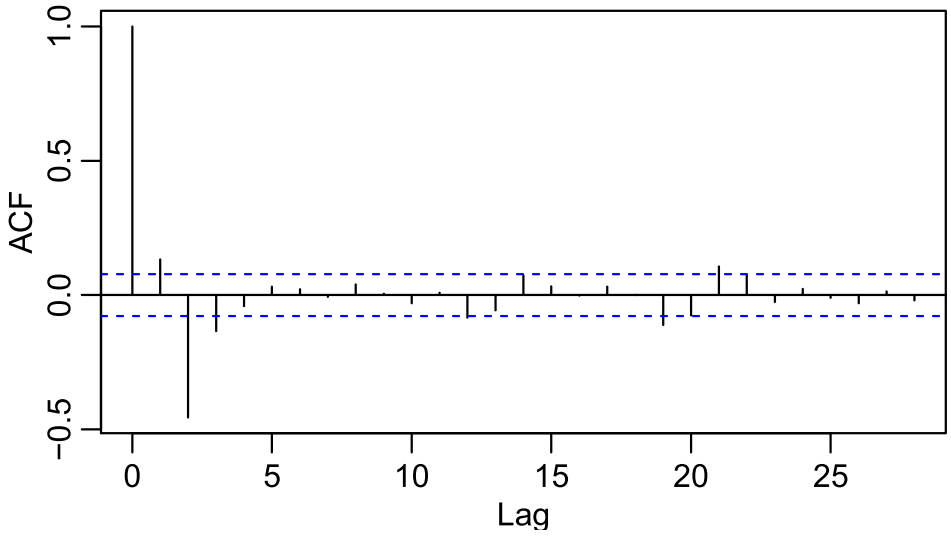
The ACF of the logged varve data shows a somewhat decreasing trend after 0, but the correlations stay above the boundary and behave somewhat cyclically, indicating consistent autocorrelation beyond lag 1.

(d) Compute the difference ut = yt - yt-2, examine its time plot and sample ACF, and argue that differencing the logged varve data produces a reasonably stationary series. Can you think of a practical interpretation for ut?

```
ut = yt - lag(yt, -2)
tsplot(ut, ylab = "", main = "yt - yt-2", col = 4, margin = 0)
```




```
acf(ut, plot = TRUE)
```

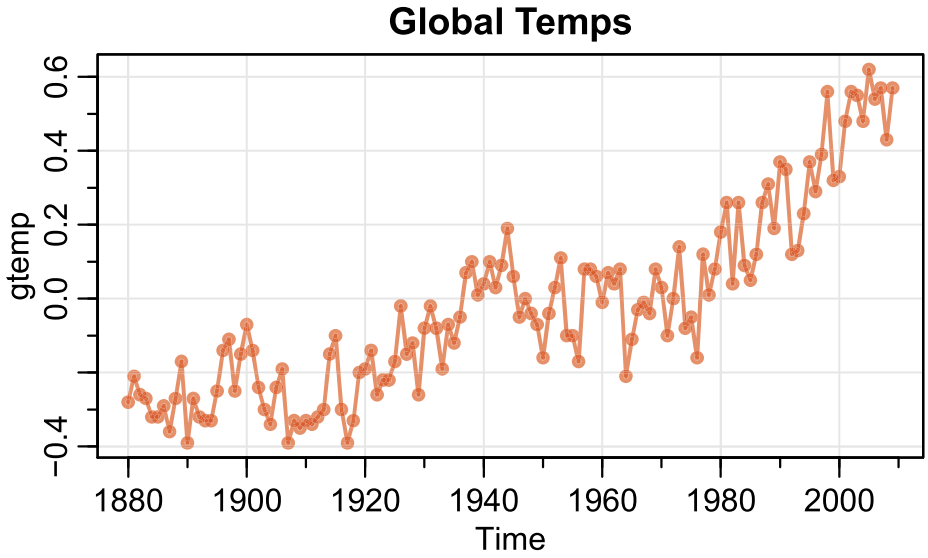


Just with the time plot we can see that the differenced logged data looks stationary, and the ACF tells us this as well - beyond lag 2 there is no autocorrelation. Differencing logged data provides symmetry along with stationarity, and a practice application for this type of transformation could be to any data that is exponential in behavior, such as anything to do with population growth. Perhaps compound interest or pandemic data can be analyzed in this way.

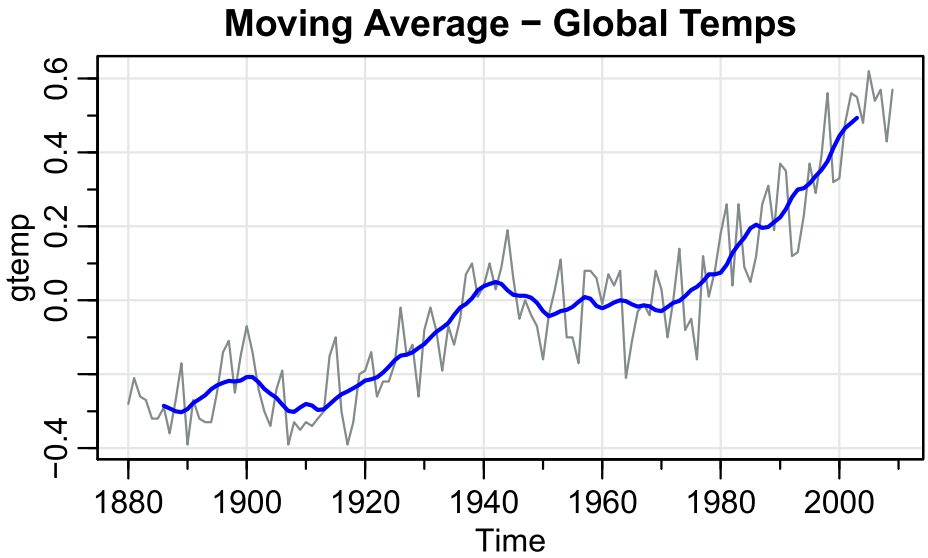
3.7

Use the three different smoothing techniques described in Example 3.16, Example 3.17, and Example 3.18, to estimate the trend in the global temperature series displayed in Figure 1.2. Comment.

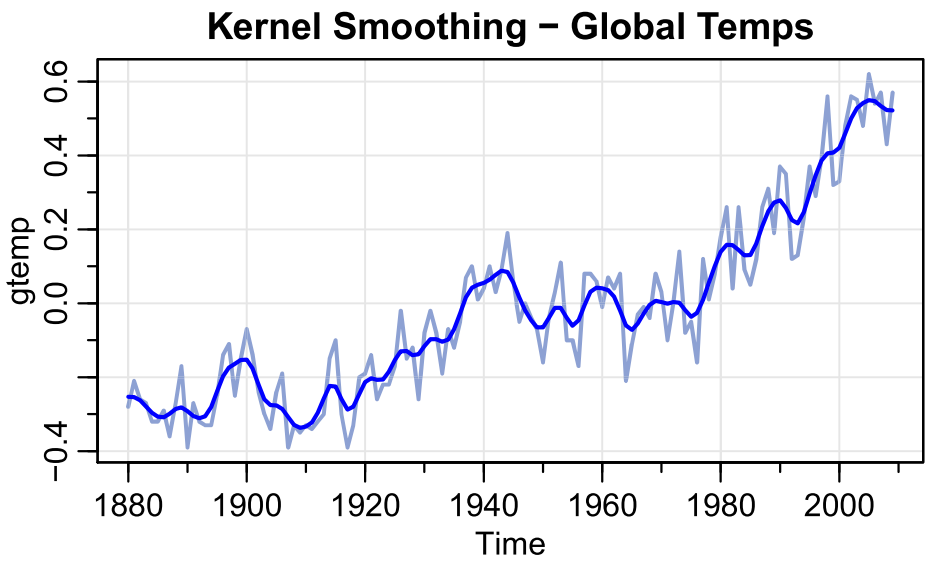
```
culer = c(rgb(.85, .30, .12, .6), rgb(.12, .65, .85, .60))

tsplot(gtemp, col = culer[1], lwd = 2, type = "o", pch = 20, main = "Global Temps")
```



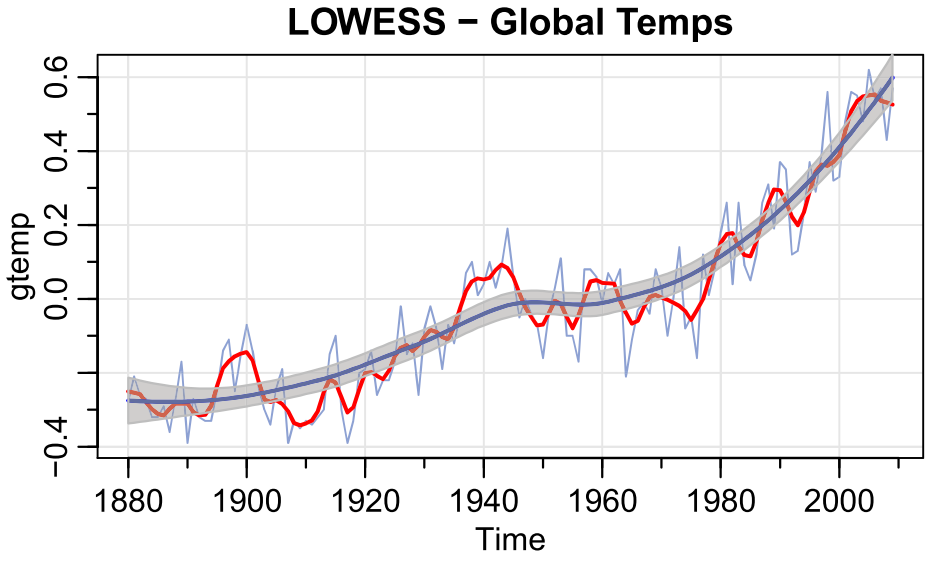

```
par(mfrow = c(1, 1))

# moving average smoother
w = c(.5, rep(1, 11), .5)/12
gtempf = filter(gtemp, sides = 2, filter = w)
tsplot(gtemp, col = "azure4", main = "Moving Average - Global Temps")
lines(gtempf, lwd = 2, col = 4)
```




```
# kernel smoothing
tsplot(gtemp, col = rgb(0.5, 0.6, 0.85, 0.9), lwd = 2, main = "Kernel Smoothing - Global Temps")
lines(ksmooth(time(gtemp), gtemp, "normal", bandwidth = 4), lwd = 2, col = 4)
```




```
# lowess
tsplot(gtemp, col = rgb(0.5, 0.6, 0.85, 0.9), main = "LOWESS - Global Temps")
lines(lowess(gtemp, f = 0.05), lwd = 2, col = 2)
lo = predict(loess(gtemp ~ time(gtemp)), se = TRUE)
trnd = ts(lo$fit, start = 1880, freq = 1)
lines(trnd, col = 4, lwd = 2)
L = trnd - qt(0.975, lo$df)*lo$se
U = trnd + qt(0.975, lo$df)*lo$se
xx = c(time(gtemp), rev(time(gtemp)))
yy = c(L, rev(U))
polygon(xx, yy, border = 8, col = gray(0.6, alpha = 0.4))
```



1. Moving Average Smoother The moving average smoothing method removes the obvious cycles and emphasizes any stand-out points, in this case extreme temperatures exhibited in the global warming data.

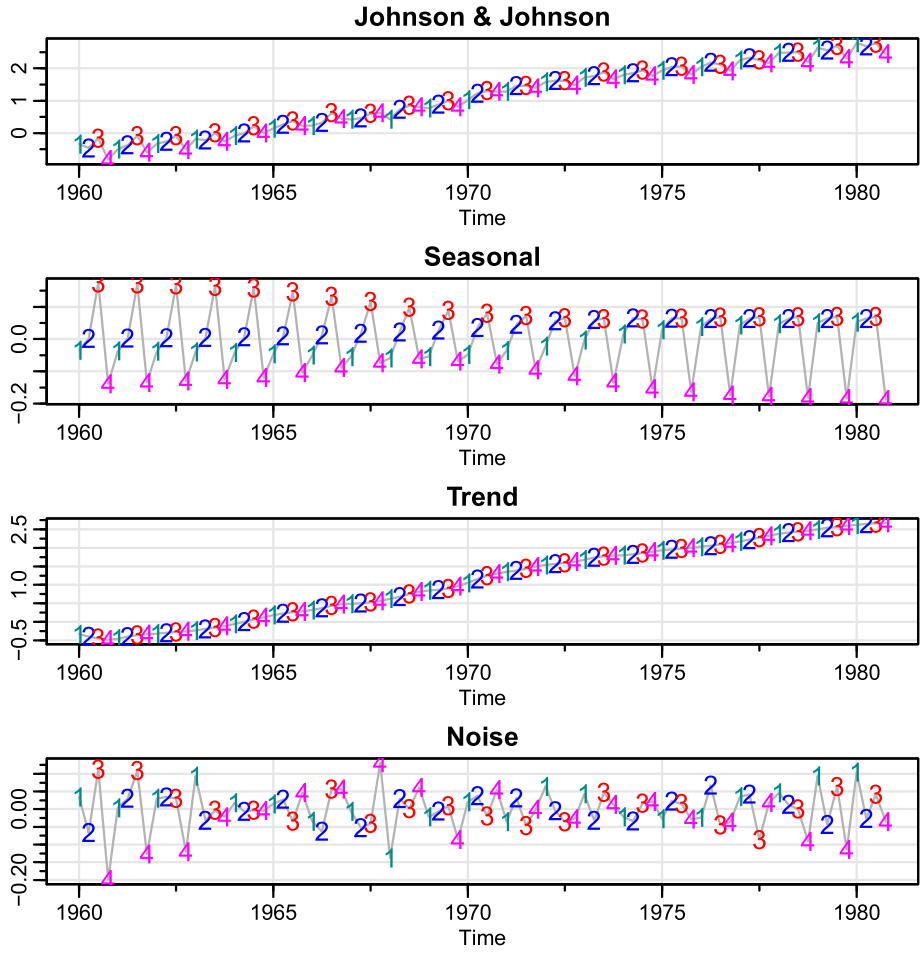
2. Kernel Smoothing To obtain an even smoother fit, kernel smoothing uses a weight to average observations of a dataset. Use b = 4 to smooth over each year.

3. Lowess Lowess (locally weighted scatterplot smoothing) uses k-nearest neighbor regression to smooth data. Weights are calculated based on a proportion of neighbors to each data point.

3.9

As in Problem 3.1, let yt be the raw Johnson & Johnson series shown in Figure 1.1, and let xt = log(yt). Use each of the techniques mentioned in Example 3.20 to decompose the logged data as xt = Tt + St + Nt and describe the results. If you did Problem 3.1, compare the results of that problem with those found in this problem.

```
culer = c("cyan4", 4, 2, 6)
par(mfrow = c(4, 1), cex.main = 1)
x = window(log(jj), start = 1960)
out = stl(x, s.window = 15)$time.series

tsplot(x, main = "Johnson & Johnson", ylab = "", col = gray(0.7))
text(x, labels = 1:4, col = culer, cex = 1.25)
tsplot(out[,1], main = "Seasonal", ylab = "", col = gray(0.7))
text(out[,1], labels = 1:4, col = culer, cex = 1.25)
tsplot(out[,2], main = "Trend", ylab = "", col = gray(0.7))
text(out[,2], labels = 1:4, col = culer, cex = 1.25)
tsplot(out[,3], main = "Noise", ylab = "", col = gray(0.7))
text(out[,3], labels = 1:4, col = culer, cex = 1.25)
```



The Seasonal plot shows an increase from Q1 to Q2, increase from Q2 to Q3, sharp decrease from Q3 to 4, and an increase back from Q4 to the next Q1. The Trend plot shows a steady increase over the 20 years of the data from quarter to quarter, relatively smooth. The Noise plot shows higher deviations from Q3 and Q4 from the data than Q1 and Q2.

In 3.1 we had a good model fit (residuals around 0 and a trend line that fit the data well). Here, we see a Noise plot that would indicate a good fit, all around 0, and a seasonal trend that is somewhat regular (in terms of direction but not always in magnitude). This generally agrees with the model in 3.1 being a good fit.