Homework 3 Solutions

Due: Tuesday 2/11/20 by 10:00am

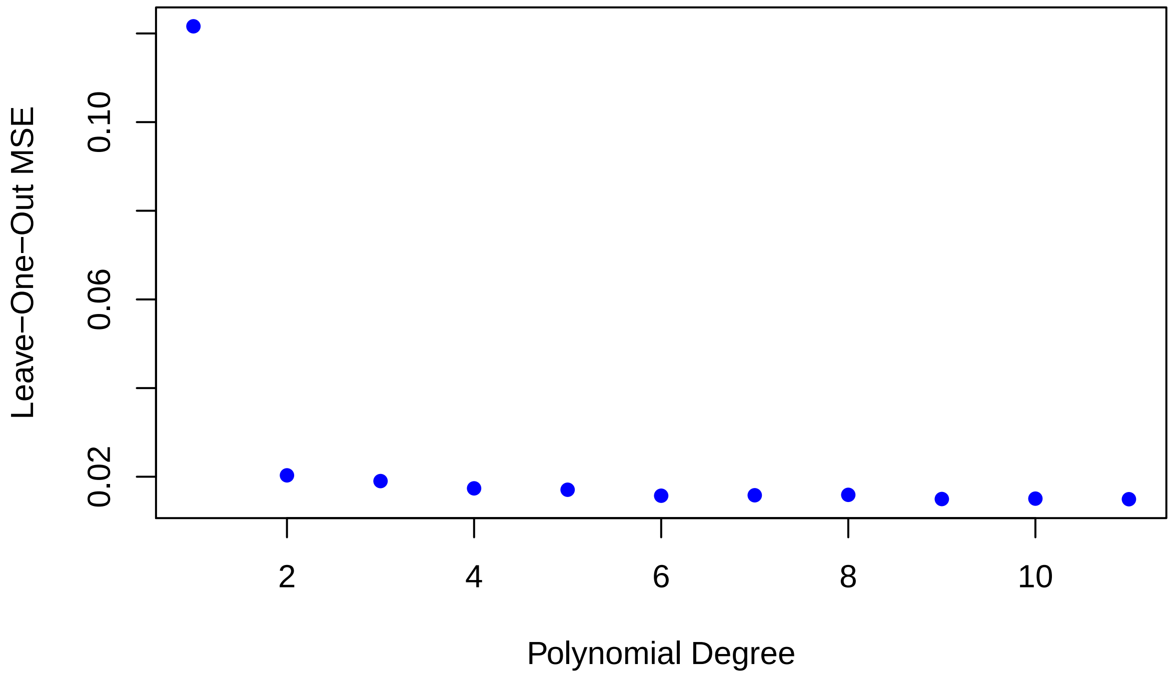
Both problems will require that you continue to work with `broc` data again. It is posted on the course website, which contains the average price of one pound of broccoli in urban areas each month, from July 1995 through December 2019.

Fitting Versus Forecasting

1. Consider the model $\texttt{price}_i = \mu + \sum_{j=1}^{d} \phi_j z_{ji} + \sum_{k=2}^{12} \alpha_k (\texttt{month}_i = k) + \epsilon_i$, $\epsilon_i \overset{i.i.d.}{\sim} \text{normal}(0, \sigma^2)$, where $\boldsymbol{z}_j$ corresponds to the orthogonal polynomials of degree j, respectively, over the set of points given by **`days since start`**. You can use the `poly` function to construct the orthogonal polynomials.

(a) Using leave-one-out cross validation on the all but the last 12 months of data. Plot the average MSE on the test data as a function of d. Which value of d produces the lowest test MSE?

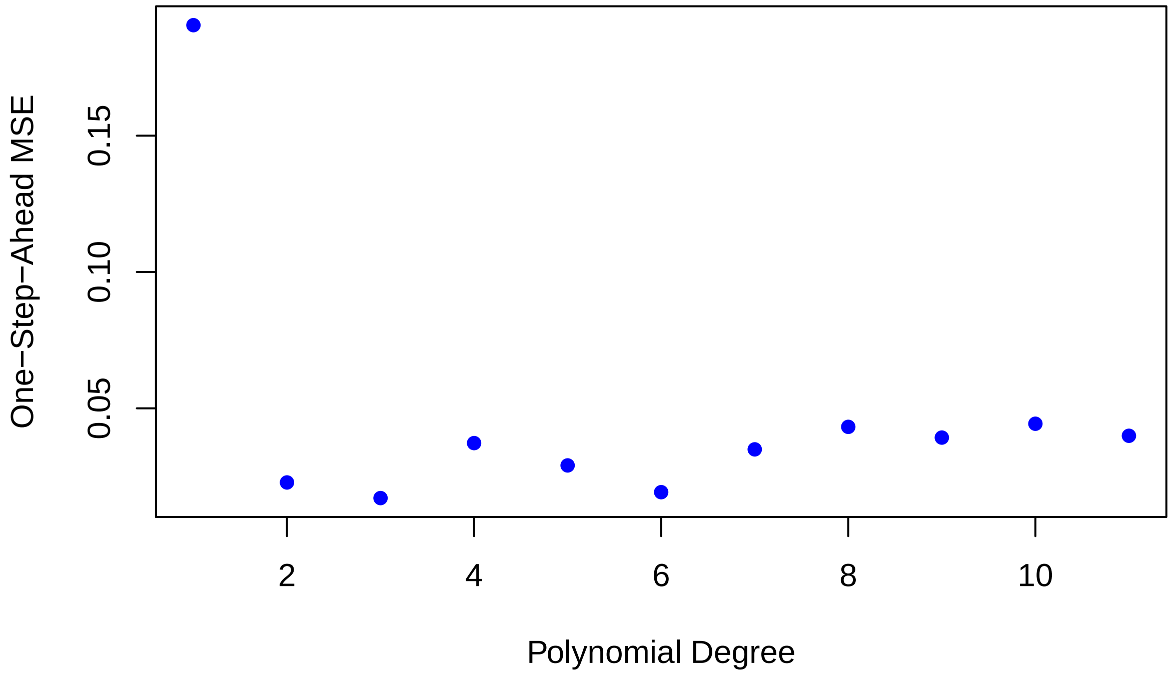
```r
load("~/Dropbox/Teaching/TimeSeries2020/stat697/content/data/broc.RData")
broc$fdate <- as.Date(broc$date, "%Y-%m-%d")
broc$month <- format(broc$fdate, "%m")
broc$dayssincestart <- as.numeric(broc$fdate) - min(as.numeric(broc$fdate))

# Let's use leave-one-out to pick the "best" model for the first 23.5 years of data
sub <- 1:(nrow(broc) - 12)

broc.poly.mses.iid <- matrix(nrow = length(sub), ncol = 11)
broc.poly.mses.dep <- matrix(nrow = 24, ncol = 11)
broc.bics <- rep(NA, 11)

for (p in 0:(ncol(broc.poly.mses.iid) - 1)) {
  # IID Style Cross Validation
  for (i in sub) {
    if (p == 0) {
      linmod.broc <- lm(price~factor(month), data = broc,
                        subset = sub[-i])
    } else {
      linmod.broc <- lm(price~poly(dayssincestart, p)+factor(month), data = broc,
                        subset = sub[-i])

    }
    broc.poly.mses.iid[i, p + 1] <- (broc$price[i] - predict(linmod.broc, broc)[i])^2
  }
  for (j in 1:nrow(broc.poly.mses.dep)) {

    if (p == 0) {
      linmod.broc <- lm(price~factor(month), data = broc,
                        subset = j - 1 + 1:(length(sub) - nrow(broc.poly.mses.dep)))
    } else {
      linmod.broc <- lm(price~poly(dayssincestart, p)+factor(month), data = broc,
                         subset = j - 1 + 1:(length(sub) - nrow(broc.poly.mses.dep)))

    }
    broc.poly.mses.dep[j, p + 1] <- (broc$price[j + length(sub) - nrow(broc.poly.mses.dep)] -
                                       predict(linmod.broc, broc)[j + length(sub) - nrow(broc.poly.mses

  }
  if (p == 0) {
      linmod.broc <- lm(price~factor(month), data = broc,
                         subset = sub)
    } else {
      linmod.broc <- lm(price~poly(dayssincestart, p)+factor(month), data = broc,
                         subset = sub)

  }
  msr <- mean(linmod.broc$residuals^2)
  broc.bics[p + 1] <- log(msr) + log(length(sub))*(p + 12)/length(sub)
}
plot(colMeans(broc.poly.mses.iid), ylab = "Leave-One-Out MSE",
     xlab = "Polynomial Degree", col = "blue", pch = 16)
```



The value $d = 10$ produces the lowest test MSE.

(b) Perform one-step-ahead cross validation on the all but the last 12 months of data, using time series of length 100 for each training subset. Plot the average MSE of the one-step-ahead forecast as a measure of model performance as a function of d. Which value of d minimizes one-step-ahead forecast error?

```
plot(colMeans(broc.poly.mses.dep), ylab = "One-Step-Ahead MSE",
     xlab = "Polynomial Degree", col = "blue", pch = 16)
```

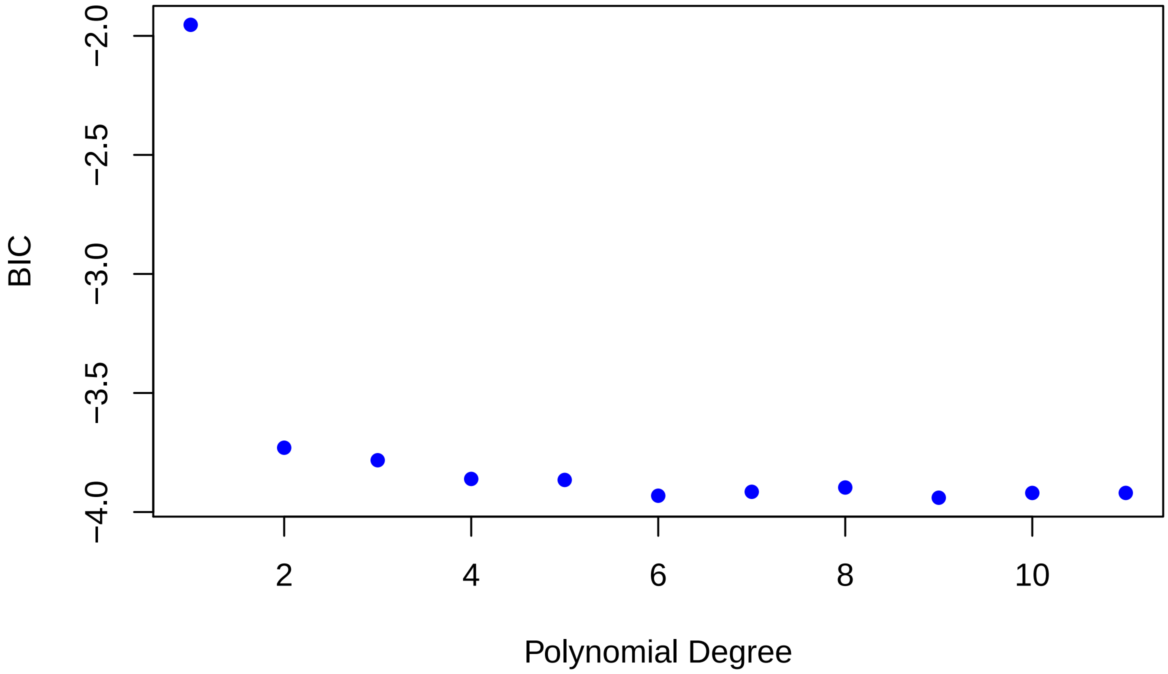


The value $d = 2$ minimizes one-step-ahead forecast error.

(c) Plot AIC, AICc, and BIC/SIC for the model fit to the training data as a function of d. Pick a criterion (AIC, AICc, or BIC), and state which value of d you would choose based on it. In at most one sentence, justify your choice of criterion with reference to the data.

I would choose BIC/SIC because it penalizes large models most aggressively, and my personal preference is for simple models especially when this much data is available.

```r
plot(broc.bics, ylab = "BIC",
     xlab = "Polynomial Degree", col = "blue", pch = 16)
```



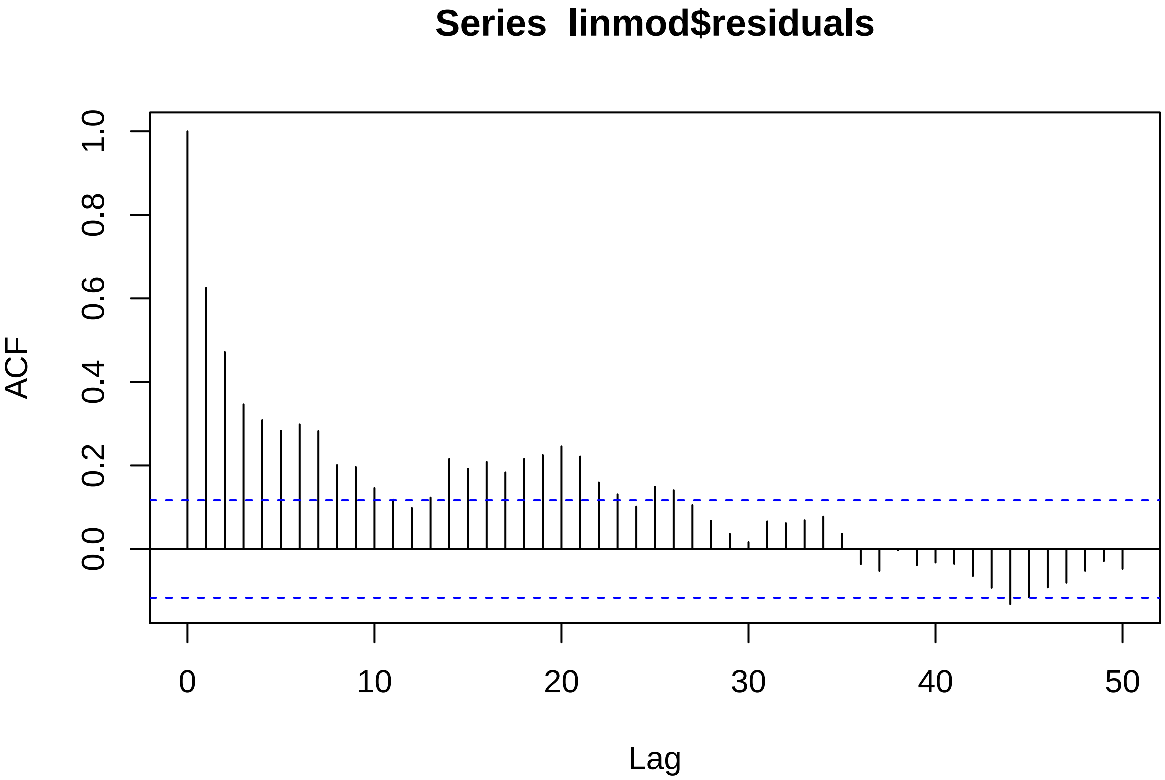
The value $d = 8$ minimizes BIC. Despite using BIC (which aggresively penalizes large models) this is still a large model!

Autocorrelation

For this problem, we will examine the residuals r_i from fitting the model to all but the last 12 months of the data: $\texttt{price}_i = \mu + \beta_1 \texttt{days since start}_i + \sum_{k=2}^{12} \alpha_k (\texttt{month}_i = k) + \epsilon_i$, $\epsilon_i \overset{i.i.d.}{\sim} \text{normal}(0, \sigma^2)$.

(a) Plot the autocorrelation function of the residuals for lags $0, 1, \dots, 50$.

```r
load("~/Dropbox/Teaching/TimeSeries2020/stat697/content/data/broc.RData")
set.seed(1)
broc$fdate <- as.Date(broc$date, "%Y-%m-%d")
broc$month <- format(broc$fdate, "%m")
broc$dayssincestart <- as.numeric(broc$fdate) - min(as.numeric(broc$fdate))
n.sub <- nrow(broc) - 12
linmod <- lm(price~dayssincestart+factor(month), data = broc,
             subset = 1:(n.sub))
lag.max <- 50
acf(linmod$residuals, lag.max = lag.max)
```

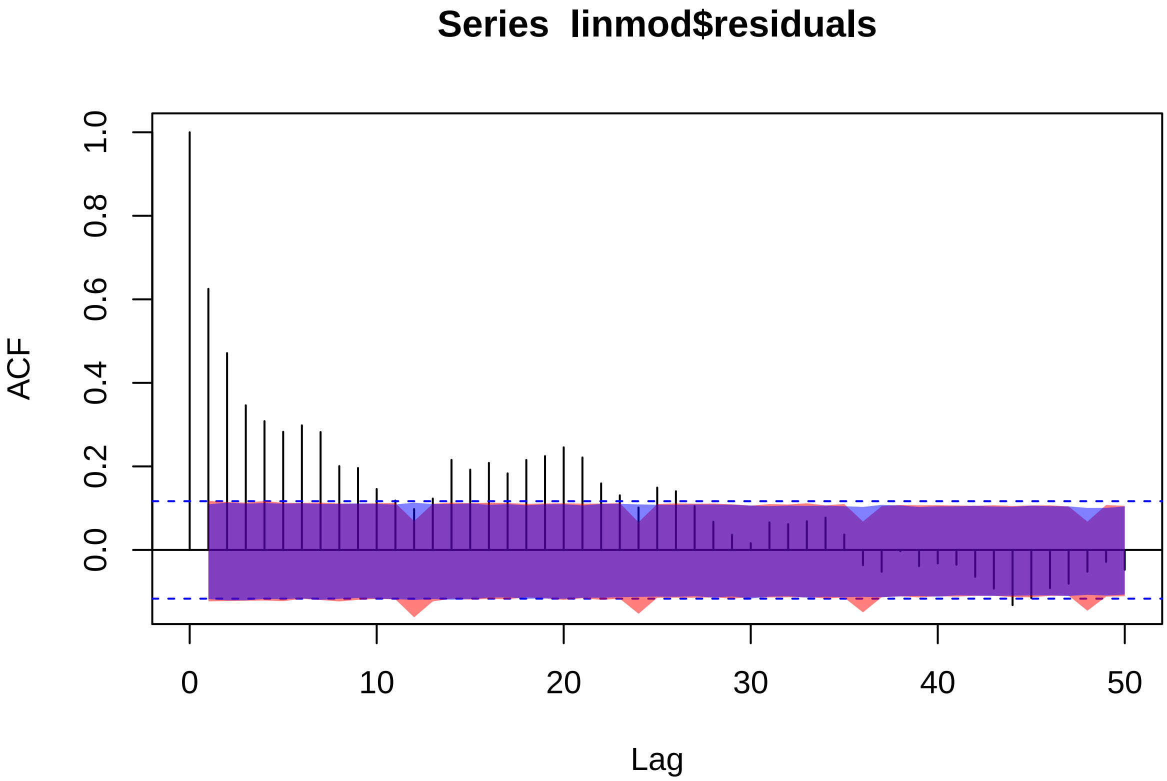


(b) Using the parametric bootstrap, simulate 10,000 bootstrap samples of the data from the model, using the least squares estimates of $\boldsymbol{\beta}$ and σ^2, according to the following two procedures:

- Procedure 1: Simulate bootstrap samples of the data $\boldsymbol{y}^{(k)}$ according to the model;
- Procedure 2: Simulate bootstrap samples of the residuals $\boldsymbol{\epsilon}^{(k)} \sim \text{normal}\left(\mathbf{0}, \sigma^2 \boldsymbol{I}\right)$, using the estimate of σ^2 from the least squares regression fit. For each simulated dataset, compute the autocorrelation function for lags $0, 1, \dots, 50$. Save each autocorrelation function value. Add two 95% intervals for each

autocorrelation function value to your plot - for each autocorrelation value you will have one 95% interval for Procedure 1, and one 95% interval for Procedure 2.

```
set.seed(1)
nboot <- 10000
acfs <- array(NA, dim = c(nboot, lag.max + 1, 2))
for (i in 1:nboot) {
  by <- rnorm(n.sub, mean = linmod$fitted.values,
              sd = summary(linmod)$sigma)
  br <- rnorm(n.sub, sd = summary(linmod)$sigma)
  blm <- lm(by~broc$dayssincestart[1:n.sub]+factor(broc$month[1:n.sub]))
  acfs[i, , 1] <- acf(blm$residuals, plot = FALSE, lag.max = lag.max)$acf
  acfs[i, , 2] <- acf(br, plot = FALSE, lag.max = lag.max)$acf
}
acf.low <- apply(acfs, c(2, 3), quantile, prob = 0.025)
acf.hig <- apply(acfs, c(2, 3), quantile, prob = 0.975)
acf(linmod$residuals, lag.max = lag.max)
polygon(c(1:lag.max, lag.max:1),
        c(acf.hig[-1, 1],
          rev(acf.low[-1, 1])), pch = 16, col = rgb(1, 0, 0, 0.5),
        border = FALSE)
polygon(c(1:lag.max, lag.max:1),
        c(acf.hig[-1, 2],
          rev(acf.low[-1, 2])), pch = 16, col = rgb(0, 0, 1, 0.5),
        border = FALSE)
```



(c) In at most one sentence, what feature(s) of the residuals does Procedure 2 account for, whereas Procedure 1 does not, and does this appear to matter for this data?

The residuals from Procedure 1 account for the fact that the residuals are are estimators of the errors as opposed to the true values of the errors $\boldsymbol{\epsilon}$. However, this does not appear to matter much for this data - both give approximately the same intervals, indicating the autocorrelation functions of the true and estimated residuals in this setting. The slight differences we observe are likely a consequence of fact that the estimated residuals are slightly correlated, i.e. $\boldsymbol{r} \sim \text{normal}\left(\boldsymbol{0}, \sigma^2\left(\boldsymbol{I} - \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\right)\right)$ where $\boldsymbol{X}$ is the design matrix from the model $\texttt{price}_i = \mu + \beta_1\texttt{days since start}_i + \sum_{k=2}^{12}\alpha_k(\texttt{month}_i = k) + \epsilon_i$ with $\epsilon_i \overset{i.i.d.}{\sim} \text{normal}(0, \sigma^2)$, whereas the actual (unobserved) residuals are not, i.e. $\boldsymbol{\epsilon} \sim \text{normal}\left(\boldsymbol{0}, \sigma^2\boldsymbol{I}\right)$.

I tried to draw your attention to this because in practice, we'll often be applying time series models to residuals from linear regression models, as if they were the (unobserved) residuals themselves. In practice, this is generally not a big deal as long as our sample is large enough and we have a reasonable number of covariates that are not too strongly correlated, but it is worth keeping in mind.

(d) Based on the Figure you made in (b), is there evidence for residual correlation across time in the broccoli data after subtracting off a linear time trend? Answer in at most one sentence.

Yes - several of the observed sample autocorrelations fall far outside of the intervals, regardless of how the intervals are calculated.

(e) Recall the approximate distribution of each sample autocorrelation value $\hat{\rho}_y(h)$ as $n \to \infty$, for fixed h, for a Gaussian white noise process $\boldsymbol{y}$. If we assume that $\hat{\rho}_y(h)$ and $\hat{\rho}_y(l)$ are independent if $h \neq l$, what is the approximate distribution of $n\sum_{l=1}^{h}\hat{\rho}_y(h)^2$ for a Gaussian white noise process $\boldsymbol{y}$ as $n \to \infty$, for fixed h?

The approximate distribution of $n\sum_{l=1}^{h}\hat{\rho}_y(h)^2$ will be χ^2_h or equivalently $\text{gamma}\left(\text{shape} = \frac{h}{2}, \text{rate} = \frac{1}{2}\right)$, because $n\sum_{l=1}^{h}\hat{\rho}_y(h)^2$ can be rewritten as the sum of h squared independent squared normal random variables.

(f) Using your results from (e), test the null hypothesis that the first $h = 50$ autocorrelations sum to exactly zero at level $\alpha = 0.05$. Give the value of the test statistic, the corresponding quantile of the test statistic under the null.

```
n <- length(linmod$residuals)
test.stat <- n*sum(acf(linmod$residuals, lag.max = lag.max, plot = FALSE)$acf[-1]^2)
qu <- qchisq(c(0.95), df = lag.max)
```

The value of the test statistic is 503.99, which greatly exceeds 67.5, the 0.95 quantile of a chi square distribution with 50 degrees of freedom. Based on this test we would reject the null hypothesis that the first $h = 50$ autocorrelations sum to zero at level $\alpha = 0.05$.

(g) Using the two parametric bootstrap procedures described in (b), test the null hypothesis that the first $h = 50$ autocorrelations sum to exactly zero at level $\alpha = 0.05$. Give the value of the test statistic, the corresponding quantiles of the test statistic under the null for each procedure.

```
qus <- apply(apply(acfs, c(1, 3), function(x) {n*sum(x[-1]^2)}), 2, quantile, prob = 0.95)
```

The 0.95 quantiles from the two procedures are 69.03 and 64.51, respectively. Both are still much smaller than the test statistic 503.99, so we would reject the null hypothesis that the first $h = 50$ autocorrelations sum to zero at level $\alpha = 0.05$ using either procedure.

(h) Based on (f) and (g), does your answer to (d) change? Answer in at most one sentence.

No - regardless of whether or not we assess the sample autocorrelations individually or jointly, we observe strong evidence for residual autocorrelation across time in the broccoli data after subtracting off a linear time trend.