Homework 8 Solutions

Due: Wednesday 4/15/20 by 5:00pm

This week, we're going to work with ARIMA models and the broccoli data. We'll try modeling dependence over time using an autoregressive integrate moving average model of orders p and q.

We're going to keep working with the `broc` data for a little while longer. It is boring to keep doing so, but it's nice to have a constant baseline as we learn. 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. In this problem, we'll continue to consider two different ways of modeling the `broc` data. Letting z_t refer to the observed broccoli prices, the two approaches we will consider are:

i. First regressing out month effects obtained from fitting a linear regression model to all but the last 12 months of data. Letting y_t refer to the corresponding residuals, we will model the residuals as an ARMA(p, q) process:

$$y_t = \mu + \sum_{i=1}^{p} \phi_i \left(y_{t-i} - \mu\right) + \sum_{j=1}^{q} \theta_j w_{t-j} + w_t, \quad w_t \overset{i.i.d.}{\sim} \text{normal}\left(0, \sigma^2_w\right), \tag{1}$$

where $y_t - \mu$ is a stationary process. We will treat the estimated regression coefficients from the initial linear regression model as fixed and known throughout.

ii. Directly incorporating month effects:

$$z_t = \boldsymbol{x}_t'\boldsymbol{\beta} + \sum_{i=1}^{p} \phi_i \left(z_{t-i} - \boldsymbol{x}_{t-i}'\boldsymbol{\beta}\right) + \sum_{j=1}^{q} \theta_j w_{t-j} + w_t, \quad w_t \overset{i.i.d.}{\sim} \text{normal}\left(0, \sigma^2_w\right), \tag{2}$$

where $z_t - \boldsymbol{x}_t'\boldsymbol{\beta}$ is a stationary process.

```r
load("~/Dropbox/Teaching/TimeSeries2020/stat697/content/data/broc.RData")
source("~/Dropbox/Teaching/TimeSeries2020/stat697/content/code/arma_autocovariance.R")
source("~/Dropbox/Teaching/TimeSeries2020/stat697/content/code/diy_prediction.R")
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$fdate))
n <- nrow(broc)
m <- nrow(broc) - 12
linmod <- lm(price~factor(month), data = broc,
             subset = 1:m)
pred <- predict(linmod, broc)
y <- broc$price - pred
z <- broc$price
X <- model.matrix(~factor(month), data = broc)
```

Differencing

First, let's implement the augemented Dickey-Fuller and Phillips-Perron tests to perform a level $\alpha = 0.05$ test the null hypothesis that the undifferenced data is non-stationary. Implement each test with and without a linear time trend using the `ndiffs` function for from the `forecast` package for `R`. Make sure that you specify the type of test correctly.

(a) Make a 2 × 2 table that provides the results of each test, with and without a linear time trend applied to residuals y_t.

	Adjusted Dickey-Fuller	Phillips-Perron
No Time Trend	$d = 1$	$d = 1$
Linear Time Trend	$d = 0$	$d = 0$

```r
library(forecast)
ndiffs(y[1:m], test = "adf", type = "level")
ndiffs(y[1:m], test = "adf", type = "trend")
ndiffs(y[1:m], test = "pp", type = "level")
ndiffs(y[1:m], test = "pp", type = "trend")
```

(b) Based on (a), is it necessary to difference the residuals y_t if a linear time trend is not included?

Both tests lead to the same conclusion - if a linear time trend is not included, it is necessary to difference the residuals y_t.

(c) Based on (a), is it necessary to difference the residuals y_t if a linear time trend is included?

Both tests lead to the same conclusion - if a linear time trend is included, it is not necessary to difference the residuals y_t.

(d) Make a 2 × 2 table that provides the results of each test, with and without a linear time trend applied to observed data z_t.

	Adjusted Dickey-Fuller	Phillips-Perron
No Time Trend	$d = 1$	$d = 1$
Linear Time Trend	$d = 0$	$d = 0$

(d) Does it make sense to apply either test directly to the observed time series data, z_t, given that we believe month effects may be present? Why or why not?

It does not - both tests have a very simple null that does *not* allow for the inclusion of additional covariates, e.g. month effects. It makes more sense to perform either test after regressing month effects out, as in (a)-(c).

(e) Based on (a) and your answer in (d), is it necessary to difference the observed data z_t if a linear time trend is not included?

It makes the most sense to use the results from (a) to answer this question. Therefore, our conclusions are unchanged from (b). If a linear time trend is not included, it is necessary to difference the observed data z_t.

(f) Based on (a) and your answer in (d), is it necessary to difference the observed data z_t if a linear time trend is included?

Again, it makes the most sense to use the results from (a) to answer this question. Therefore, our conclusions are unchanged from (c). If a linear time trend is included, it is not necessary to difference the observed data z_t.

Order Selection for Differenced Data

In this part, our goal is to choose the order of the ARIMA$(p, 1, q)$ model for approaches (i) and (ii). Consider $p = 0, 1, 2, 3$ and $q = 0, 1, 2$, and the following measures of model performance:

- One-step-ahead moving window style cross-validation, using the mean squared error of the forecasts
- Twelve-steps-ahead moving window style cross-validation, using the mean squared error of the forecasts
- AIC
- BIC
- Box-Pierce Test Result
  - This is the test you implemented on Homework 3, see parts (e) and (f) of the second problem If there are any choices you need to make when computing these measures of model performance, e.g. number of moving windows, make the same choices that you made on HW 7.

(b) Summarize your results in a table which shows the best choice of p and q for each measure of of model performance.

	i.	ii.
CV-1	$p = 2$, $q = 2$	$p = 2$, $q = 2$
CV-12	$p = 2$, $q = 2$	$p = 2$, $q = 2$
AIC	$p = 1$, $q = 1$	$p = 1$, $q = 1$
BIC	$p = 1$, $q = 1$	$p = 1$, $q = 1$
BP	Either $p = 1$, $q = 1$ or $p = 0$, $q = 2$	Either $p = 1$, $q = 1$ or $p = 0$, $q = 2$

```r
ps <- 0:3
qs <- 0:2
results <- array(dim = c(length(ps), length(qs), 5))
num.cv <- 50
for (p in ps) {
  cat("p = ", p, "\n")
  for (q in qs) {
    l1mse <- rep(NA, num.cv)
    l12mse <- rep(NA, num.cv)
    arima.fit <- arima(y[1:m], order = c(p, 1, q),
                       method = "ML")
    acf.fit <- acf(arima.fit$residuals, lag.max = 20,
                   plot = FALSE)$acf[-1, 1, 1]
    cr <- qchisq(0.95, df = 20)
    bpt <- m*sum(acf.fit^2)

    for (i in 1:num.cv) {
      cat("i = ", i, "\n")
      l1arima.fit <- arima(y[i:(m - num.cv + i - 1)], order = c(p, 1, q),
                          method = "ML")
      l1mse[i] <- (y[m - num.cv + i] - c(predict(l1arima.fit, n.ahead = 1)$pred))^2
      l12arima.fit <- arima(y[i:(m - num.cv + i - 12)], order = c(p, 1, q),
                           method = "ML")
      l12mse[i] <- sum((y[m - num.cv + i - 12 +  1:12] -
                          c(predict(l12arima.fit, n.ahead = 12)$pred))^2)
    }


    results[which(p == ps), which(q == qs), ] <- c(mean(l1mse, na.rm = TRUE),
                                                   mean(l12mse, na.rm = TRUE),
                                                   AIC(arima.fit),
                                                   BIC(arima.fit),
                                                   bpt > cr)
  }
}
which(results[, , 1] == min(results[, , 1]), arr.ind = TRUE)
which(results[, , 2] == min(results[, , 2]), arr.ind = TRUE)
which(results[, , 3] == min(results[, , 3]), arr.ind = TRUE)
which(results[, , 4] == min(results[, , 4]), arr.ind = TRUE)
results[, , 5]
# Repeat for ii.
for (p in ps) {
  for (q in qs) {
    l1mse <- rep(NA, num.cv)
    l12mse <- rep(NA, num.cv)
    arima.fit <- arima(z[1:m], order = c(p, 1, q),
                          method = "ML",
                      xreg = X[1:m, -1], include.mean = FALSE)
    acf.fit <- acf(arima.fit$residuals, lag.max = 20,
                   plot = FALSE)$acf[-1, 1, 1]
    cr <- qchisq(0.95, df = 20)
    bpt <- m*sum(acf.fit^2)

    for (i in 1:num.cv) {
      l1arima.fit <- arima(z[i:(m - num.cv + i - 1)], order = c(p, 1, q),
                          method = "ML",
                          xreg = X[i:(m - num.cv + i - 1), -1],
                          include.mean = FALSE)
      l1mse[i] <- (z[m - num.cv + i] -
                     c(predict(l1arima.fit, n.ahead = 1,
                               newxreg = X[m - num.cv + i, -1,
                                           drop = FALSE])$pred))^2
      l12arima.fit <- arima(z[i:(m - num.cv + i - 12)], order = c(p, 1, q),
                           method = "ML",
                           xreg = X[i:(m - num.cv + i - 12), -1],
                           include.mean = FALSE)
      l12mse[i] <- sum((z[m - num.cv + i - 12 +  1:12] -
                          c(predict(l12arima.fit,
                                    newxreg = X[m - num.cv + i - 12 +  1:12, -1,
                                                drop = FALSE],
                                    n.ahead = 12)$pred))^2)
    }


    results[which(p == ps), which(q == qs), ] <- c(mean(l1mse, na.rm = TRUE),
                                                   mean(l12mse, na.rm = TRUE),
                                                   AIC(arima.fit),
                                                   BIC(arima.fit),
                                                   bpt > cr)
  }
}
which(results[, , 1] == min(results[, , 1]), arr.ind = TRUE)
which(results[, , 2] == min(results[, , 2]), arr.ind = TRUE)
which(results[, , 3] == min(results[, , 3]), arr.ind = TRUE)
which(results[, , 4] == min(results[, , 4]), arr.ind = TRUE)
results[, , 5]
```

(c) For this problem, does the best choice of p and q depend much on the approach used?

In this case, the best choice of p and q does not depend on the approach used at all.

(d) Choose one measure of model performance to base your choice of p and q on, and justify your choice.

I tend to prefer the simplest plausible model, so I would usualy use the Box-Pierce test to assess model fit. However, when we use the Box-Pierce test by choosing the simplest model that has residuals that a Box-Pierce test indicates are uncorrelated, the Box-Pierce test doesn't give a single best model. Because BIC also tends to prefer simpler models, I would use the model that both BIC and the Box-Pierce test agree on - $p = 1$ and $q = 1$.

Forecasting

Using the best ARIMA$(p, 1, q)$ model based on your choice in part (d) of the previous problem, compute predicted values of the rest of the time series (just the last 12 observations) using the observed data using both of the two approaches. You can use the `get.preds` function I have provided to get these predictions, with `h` set to the number of remaining observations.

(a) Make a table that shows estimates of the month effects, the autoregressive and/or moving average parameters, and noise variance fit using both approaches. Does the approach used affect the estimates much?

	i.	ii.
$\hat{\alpha}_2$	-0.044	-0.049
$\hat{\alpha}_3$	-0.084	-0.094
$\hat{\alpha}_4$	-0.091	-0.104
$\hat{\alpha}_5$	-0.122	-0.139
$\hat{\alpha}_6$	-0.101	-0.122
$\hat{\alpha}_7$	-0.124	-0.127
$\hat{\alpha}_8$	-0.155	-0.162
$\hat{\alpha}_9$	-0.119	-0.129
$\hat{\alpha}_{10}$	-0.089	-0.102
$\hat{\alpha}_{11}$	-0.083	-0.100
$\hat{\alpha}_{12}$	-0.067	-0.087
$\hat{\phi}_1$	0.445	0.446
$\hat{\theta}_1$	-0.862	-0.862
$\hat{\sigma}^2_w$	0.011	0.011

The approach used does not affect the estimates very much at all.

```
p <- 1
q <- 1
arima.fit.full <- arima(z[1:m], order = c(p, 1, q),
                          method = "ML",
                      xreg = X[1:m, -1], include.mean = FALSE)
phi.full <- arima.fit.full$coef[paste("ar", 1:p, sep = "")]
theta.full <- arima.fit.full$coef[paste("ma", 1:q, sep = "")]
```

```
sig.sq.full <- arima.fit.full$sigma2
arima.fit.part <- arima(y[1:m], order = c(p, 1, q),
                          method = "ML", include.mean = FALSE)
phi.part <- arima.fit.part$coef[paste("ar", 1:p, sep = "")]
theta.part <- arima.fit.part$coef[paste("ma", 1:q, sep = "")]
sig.sq.part <- arima.fit.part$sigma2

beta.full <- arima.fit.full$coef[(p + q + 1):length(arima.fit.full$coef)]
beta.part <- linmod$coefficients
```

(b) Compare the predicted values for the last 12 observations obtained using each approach by plotting them on the same plot as the last 24 observations. Does the approach used affect the predicted values much?

See the plot after (e). The approach used does not affect the predicted at all.

```
pred.full <- predict(arima.fit.full, n.ahead = length(y) - m,
                     newxreg = X[(m + 1):length(y), -1])$pred
pred.part <- predict(arima.fit.part, n.ahead = length(y) - m)$pred +
  pred[(m + 1):length(y)]
```

(c) Use `predict` to obtain the variances of the predictions of the last 12 observations, and add 95% prediction intervals to your plot for each method. Does the approach used affect variances of the predictions much?

See the plot after (e). The approach used does not affect the variances of the predictions at all.

```
pred.full.se <- predict(arima.fit.full, n.ahead = length(y) - m,
                     newxreg = X[(m + 1):length(y), -1])$se
pred.part.se <- predict(arima.fit.part, n.ahead = length(y) - m)$se
```

(d) Using the parametric bootstrap, obtain approximate standard errors of the predicted values of the last 12 observations. Remember - in approach i. we are treating the fitted values from the regression as fixed, so just simulate new residuals and continue to use the same fitted values. Add 95% confidence intervals to your plot for each method. Does the approach used affect the confidence intervals much? How do the confidence intervals compare to the prediction intervals?

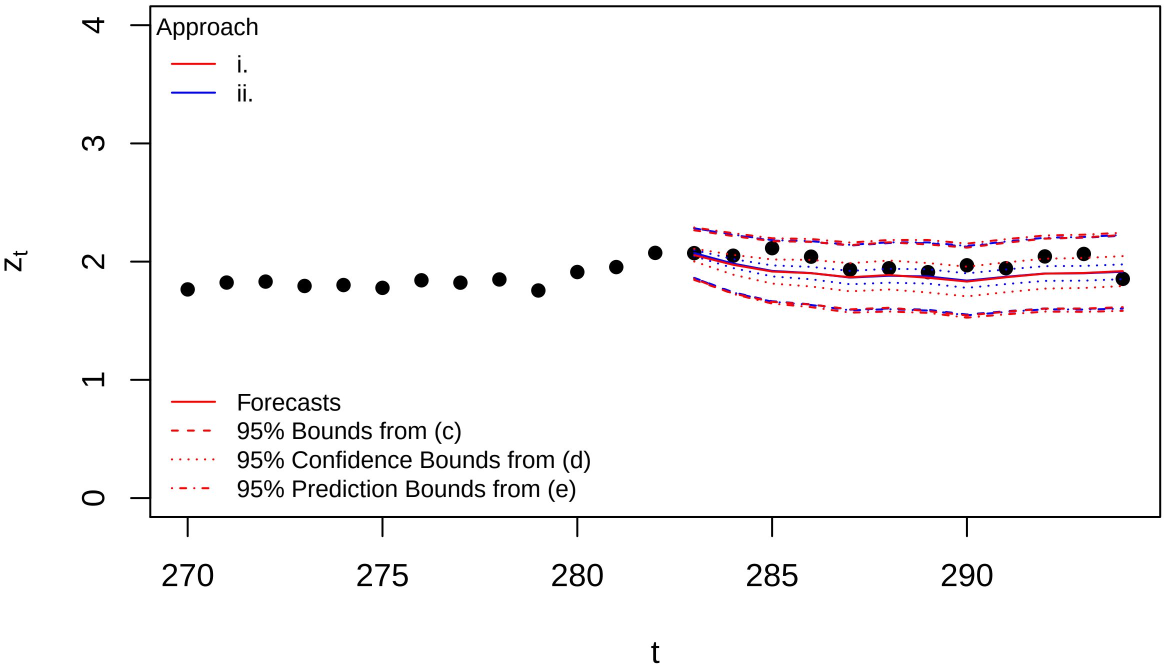
See the plot after (e). Again, the approach used does affect the approximate standard errors of the predicted values slightly. Surprisingly, the approximate standard errors for approach ii. tend to be slightly smaller even though they account for uncertainty about $\boldsymbol{\beta}$, whereas the standard errors for approach i. do not. Regardless of the approach used, the confidence intervals tend to be much smaller than than the prediction intervals from (c).

```
nboot <- 100

pred.full.boot <- matrix(NA, nrow = nboot, ncol = length(y) - m)
pred.part.boot <- matrix(NA, nrow = nboot, ncol = length(y) - m)

for (i in 1:nboot) {

  # Simulate data using the full model
  z.sim.full <- arima.sim(model = list("ar" = phi.full,
                                       "ma" = theta.full,
                                       "order" = c(p, 1, q)),
                          n = m - 1,
                          sd = sqrt(sig.sq.full)) + X[1:m, -1]%*%beta.full
  z.sim.part <- arima.sim(model = list("ar" = phi.part,
                                       "ma" = theta.part,
                                       "order" = c(p, 1, q)),
                          n = m - 1,
                          sd = sqrt(sig.sq.part)) + X[1:m, ]%*%beta.part

  y.sim <- z.sim.part - X[1:m, ]%*%beta.part

  arima.fit.full.sim <- arima(z.sim.full[1:m], order = c(p, 1, q),
                          method = "ML",
                         xreg = X[1:m, -1], include.mean = FALSE)
  beta.full.sim <- beta.full <- arima.fit.full$coef[(p + q + 1):length(arima.fit.full$coef)]
  phi.full.sim <- arima.fit.full.sim$coef[paste("ar", 1:p, sep = "")]
  theta.full.sim <- arima.fit.full.sim$coef[paste("ma", 1:q, sep = "")]
  sig.sq.full.sim <- arima.fit.full.sim$sigma2
  z.tmp <- z[1:m] - X[1:m, -1]%*%beta.full.sim
  h <- length(z) - m
  pred.diff <- get.preds(y = z.tmp[-1] - z.tmp[-length(z.tmp)],
                   phi = phi.full.sim,
                 theta = theta.full.sim, sig.sq.w = sig.sq.full.sim,
                 h = h)$pred
  pred.full.sim <- c(z.tmp[m] + cumsum(pred.diff[m + 0:(h - 1)]) +
    X[(m + 1):length(z), -1]%*%beta.full.sim)

  arima.fit.part.sim <- arima(y.sim[1:m], order = c(p, 1, q),
                         method = "ML", include.mean = FALSE)
  phi.part.sim <- arima.fit.part.sim$coef[paste("ar", 1:p, sep = "")]
  theta.part.sim <- arima.fit.part.sim$coef[paste("ma", 1:q, sep = "")]
  sig.sq.part.sim <- arima.fit.part.sim$sigma2
  pred.diff <- get.preds(y = y[2:m] - y[1:(m - 1)],
                         phi = phi.part.sim,
                         theta = theta.part.sim,
                         sig.sq.w = sig.sq.part.sim,
                         h = h)$pred
  pred.part.sim <- c(y[m] + cumsum(pred.diff[m + 0:(h - 1)]) +
    pred[(m + 1):length(y)])

  pred.full.boot[i, ] <- pred.full.sim
  pred.part.boot[i, ] <- pred.part.sim

}
pred.full.boot.se <- apply(pred.full.boot, 2, sd, na.rm = TRUE)
pred.part.boot.se <- apply(pred.part.boot, 2, sd, na.rm = TRUE)
```

(e) The prediction intervals for the last 12 observations obtained in (c) account for variability due to uncertainty regarding future values of the time series but do not account for uncertainty regarding the values of the parameters of the model. We can obtain a variance estimate that accounts for both sources of uncertainty by taking the sum of the variances obtained in (c) and (d). Prediction intervals that account for both sources of uncertainty can be obtained using this variance new estimate. Add these prediction intervals for the last 12 observations to your plot for each approach. Does the approach used affect variances of the confidence intervals much? Which source of variability appears to dominate?

See the plot after (e). The approach used does affect the approximate standard errors of the predicted values, which makes sense because the approximate standard errors for approach ii. account for uncertainty about $\boldsymbol{\beta}$ whereas the standard errors for approach i. do not. The variability due to future values being unobserved appears to dominate the variability due to uncertainty about the parameter estimates.

```r
plot(c((m - 12):length(y)), z[c((m - 12):length(y))],
     ylim = c(0, 4), pch = 16,
     xlab = "t", ylab = expression(z[t]))
lines((m + 1):length(y), pred.full, col = "blue")
lines((m + 1):length(y), pred.full + qnorm(0.975)*pred.full.se,
      col = "blue", lty = 2)
lines((m + 1):length(y), pred.full + qnorm(0.025)*pred.full.se,
      col = "blue", lty = 2)
lines((m + 1):length(y), pred.full + qnorm(0.975)*pred.full.boot.se,
      col = "blue", lty = 3)
lines((m + 1):length(y), pred.full + qnorm(0.025)*pred.full.boot.se,
      col = "blue", lty = 3)
lines((m + 1):length(y), pred.full + qnorm(0.975)*sqrt(pred.full.boot.se^2 + pred.full.se^2),
      col = "blue", lty = 4)
lines((m + 1):length(y), pred.full + qnorm(0.025)*sqrt(pred.full.boot.se^2 + pred.full.se^2),
      col = "blue", lty = 4)

lines((m + 1):length(y), pred.part + qnorm(0.975)*pred.part.se,
      col = "red", lty = 2)
lines((m + 1):length(y), pred.part + qnorm(0.025)*pred.part.se,
      col = "red", lty = 2)
lines((m + 1):length(y), pred.part + qnorm(0.975)*pred.part.boot.se,
      col = "red", lty = 3)
lines((m + 1):length(y), pred.part + qnorm(0.025)*pred.part.boot.se,
      col = "red", lty = 3)
lines((m + 1):length(y), pred.part, col = "red")
lines((m + 1):length(y), pred.full + qnorm(0.975)*sqrt(pred.part.boot.se^2 + pred.part.se^2),
      col = "red", lty = 4)
lines((m + 1):length(y), pred.full + qnorm(0.025)*sqrt(pred.part.boot.se^2 + pred.part.se^2),
      col = "red", lty = 4)
legend("topleft", lty = c(1, 1), col = c("red", "blue"),
       legend = c("i.", "ii."), title = "Approach", cex = 0.75,
       bty = "n")
legend("bottomleft", lty = c(1, 2, 3, 4), col = c("red"),
       legend = c("Forecasts",
                  "95% Bounds from (c)",
                  "95% Confidence Bounds from (d)",
                  "95% Prediction Bounds from (e)"), cex = 0.75,
       bty = "n")
```



Putting It All Together

In this problem, let's just consider the approach (ii). Recall the following measures of model performance:

- One-step-ahead moving window style cross-validation, using the mean squared error of the forecasts
- Twelve-steps-ahead moving window style cross-validation, using the mean squared error of the forecasts
- AIC
- BIC
- Box-Pierce Test Result

(a) Which measure(s) of model performance would be appropriate to use if you wanted to compare the performance of undifferenced and differenced ARIMA$(p, 0, q)$ and ARIMA$(p, 1, q)$ models?

The cross-validation approachs and the Box-Pierce test approaches are appropriate to use to compare the performance of differenced and undifferenced models, because they are not based on the likelihoods. The likelihoods of the differenced and undifferenced models are not comparable, we can think of them as using different data.

(b) Pick one measure from the ones you identified in (a). You will use this measure to compare the performance of best ARIMA$(p, 1, q)$ model identified in this homework with the best ARIMA$(p, 0, q)$ identified in Homework 7. Justify your choice.

I will use twelve-step-ahead cross-validation. This is easily comparable across ARIMA$(p, 0, q)$ and ARIMA$(p, 1, q)$ models, and unlike the way we have used the Box-Pierce test, it provides a continuous measure of model fit. I will use twelve-step-ahead cross-validation over one-step-ahead cross validation because I am interested in finding a model that performs well for forecasting more than one point into the future.

(c) Provide the values of the measure you chose in (b) for the best ARIMA$(p, 1, q)$ model identified in this homework and the best ARIMA$(p, 0, q)$ identified in Homework 7.

	$d = 0$	$d = 1$
i.	0.269 ($p = 2$, $q = 1$)	0.266 ($p = 2$, $q = 2$)

	$d = 0$	$d = 1$
ii.	0.346 ($p = 3$, $q = 2$)	0.282 ($p = 2$, $q = 2$)

(d) Based on (c), indicate whether or not you think it is best to difference the data in this particular application.

Regardless of whether I chose approach i. or ii., I would choose to difference the data. Note that the squared error losses are not comparable across approaches here because of how I computed them, the squared error losses for i. are on the residual scale whereas the squared error losses for ii. are on the observed data scale.