

Homework #3

Dustin Leatherman

2/9/2020

3.6

Let c_t be the cardiovascular mortality series discussed in Chapter 2, Ex 2.2 and let $x_t = \Delta c_t$ be the differenced data

a

Plot x_t and compare it to the actual data plotted in Figure 2.2. Why does differencing seem to be reasonable in this case?

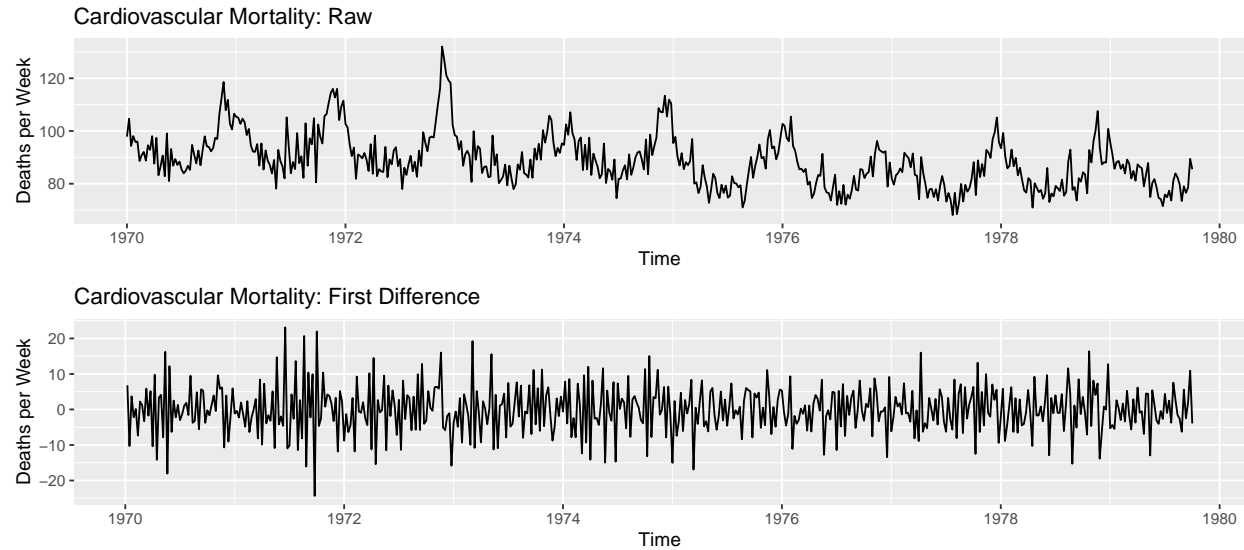
Differencing is reasonable in this case because this time series appears to trend downwards indicating that it is *not* stationary. The differenced plot shows that the mean of the time series to be constant, ≈ 0 .

```
plus2 <- function(x) x + 2
minus2 <- function(x) x - 2
cmort.diff <- diff(cmort)

plot.data <-
  autoplot(cmort, ylab = "Deaths per Week") +
  # default scale used non-whole numbers so set the scale nicer
  scale_x_continuous("Time", breaks = trans_breaks(plus2, minus2)) +
  ggtitle("Cardiovascular Mortality: Raw")

plot.diff <-
  autoplot(cmort.diff, ylab = "Deaths per Week") +
  # default scale used non-whole numbers so set the scale nicer
  scale_x_continuous("Time", breaks = trans_breaks(plus2, minus2)) +
  ggtitle("Cardiovascular Mortality: First Difference")

grid.arrange(
  plot.data,
  plot.diff,
  ncol = 1
)
```



b

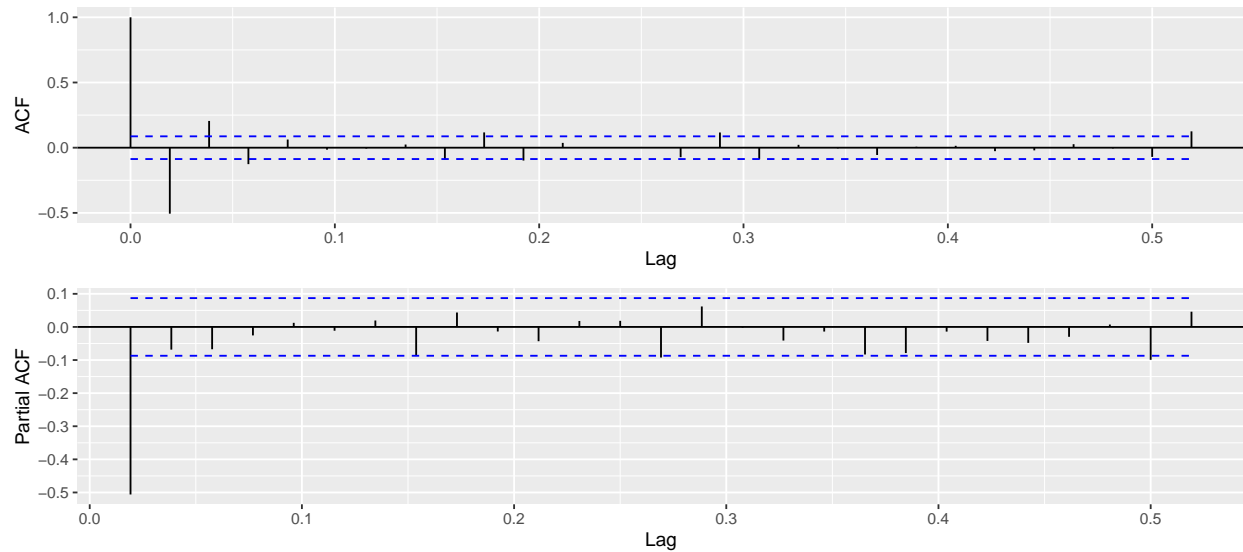
Calculate and plot the sample ACF and PACF of x_t and using Table 3.1, argue that an AR(1) is appropriate for x_t .

An AR(1) model is appropriate for this data since the PACF plot cuts off after Lag 1 and the ACF plot tailing off to zero after Lag 1.

```
cmort.acf <-
  autoplot(acf(cmort.diff, plot=FALSE)) +
  geom_hline(yintercept = 0)

cmort.pacf <-
  autoplot(pacf(cmort.diff, plot = FALSE), ylab = "Partial ACF") +
  geom_hline(yintercept = 0)

grid.arrange(
  cmort.acf,
  cmort.pacf,
  ncol = 1
)
```



c

Fit an AR(1) to x_t using maximum likelihood (basically unconditional least squares) as in Section 3.6. The easiest way to do this is to use `sarima` from `astsa`. Comment on the significance of the regression parameter estimates of the model. What is the estimate of the white noise variance?

There is convincing evidence that there is a correlation between the current timepoint and the previous timepoint (two-tailed t-test. p-value = 0). There is no evidence that the estimated mean is non-zero (two-tailed t-test. p-value = 0.8782). This is fine though since we are only interested in whether the mean is constant.

The estimated variance of the white noise parameter is **33.81**.

```
sarima(cmort, p = 1, d = 1, q = 0, details = FALSE)
```

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##      Q), period = S), xreg = constant, transform.pars = trans, fixed = fixed,
##      optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1  constant
##        -0.5064  -0.0263
## s.e.    0.0383   0.1715
##
## sigma^2 estimated as 33.81:  log likelihood = -1612.05,  aic = 3230.11
##
## $degrees_of_freedom
## [1] 505
##
## $ttable
##      Estimate      SE  t.value p.value
## ar1      -0.5064 0.0383 -13.2233  0.0000
## constant -0.0263 0.1715  -0.1533  0.8782
```

```
##
## $AIC
## [1] 6.371023
##
## $AICc
## [1] 6.37107
##
## $BIC
## [1] 6.396044
```

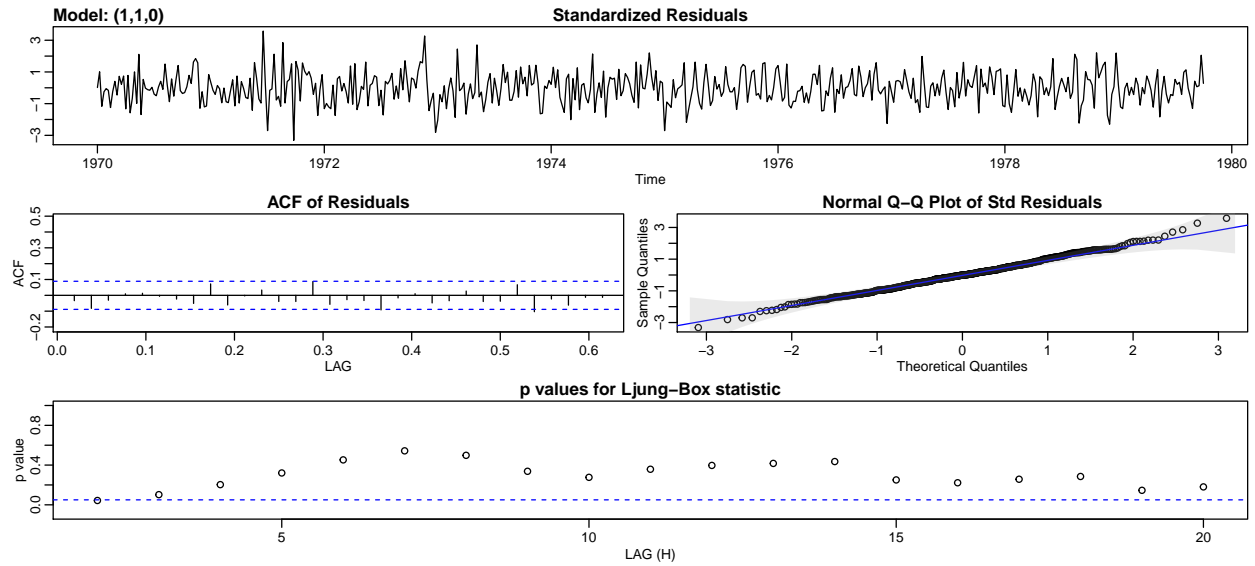
d

Examine the residuals and comment on whether or not you think the residuals are white.

The standardized residuals appear to have, in general, constant variance throughout the time series. There are a couple points between 1970 and 1972 where the residuals exceed $|3|$ and while unusual, is not an indicator of whether or not the series is white. The ACF Plot of the Residuals shows that they are uncorrelated with previous values. The normality assumption is also met per the QQ Plot. Thus, the residuals appear to have the characteristics of white noise.

```
a <- sarima(cmort, p = 1, d = 1, q = 0)
```

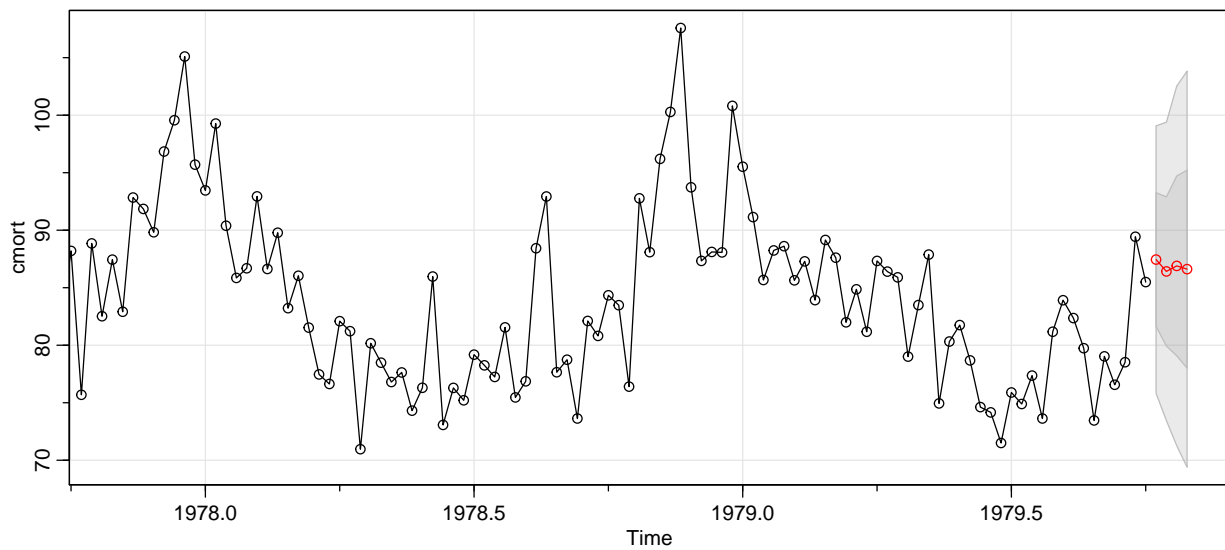
```
## initial value 1.908706
## iter 2 value 1.760342
## iter 3 value 1.760341
## iter 4 value 1.760341
## iter 4 value 1.760341
## iter 4 value 1.760341
## final value 1.760341
## converged
## initial value 1.760656
## iter 2 value 1.760656
## iter 3 value 1.760656
## iter 3 value 1.760656
## iter 3 value 1.760656
## final value 1.760656
## converged
```



e

Assuming the fitted model is the true model, find the forecasts over a four-week horizon, x_{n+m}^n for $m = 1, 2, 3, 4$ and the corresponding 95% prediction intervals; $n = 508$ here. The easiest way to do this is to use `sarima.for` from `astsa`.

```
cmort.pred <- sarima.for(cmort, p = 1, d = 1, q = 0, n.ahead = 4)
```



| m | estimate | standard err | 95% prediction interval |
|---|----------|--------------|-------------------------|
| 1 | 87.44554 | 5.814551 | [76.04902, 98.85206] |
| 2 | 86.41566 | 6.484342 | [73.70635, 99.12497] |
| 3 | 86.89756 | 7.814494 | [71.58115, 102.214] |
| 4 | 86.61391 | 8.606418 | [69.74533, 103.4825] |

Note: $Z_{0.95}$ is used to calculate the prediction interval.

3.9

Generate 10 realizations of length $n = 200$ each of an ARMA(1,1) process with $\phi = 0.9$, $\theta = 0.5$, $\sigma^2 = 1$. Find the MLEs of the three parameters in each case and compare the estimators to the true values.

```
set.seed(123)
# simulateARIMA(c(1,0,1), ar = 0.9, ma = 0.5, n = 200)
simulateARIMA <- function(order, ar, ma, n, sd = 1, realizations = 10) {
  lapply(1:realizations, function(i) {
    # get simulated data
    sim <- arima.sim(model=list(order=order, ar=ar, ma=ma), n=n, rand.gen=rnorm, sd=sd)

    # fit an ARMA model
    fit <- arima(sim, order = order)

    # output a Tibble with all estimated parameters
    fit %>%
      tidy %>%
      # get parameter estimates and pivot to columns
      select(term, estimate) %>%
      filter(term != "intercept") %>%
      pivot_wider(names_from = term, values_from = estimate) %>%
      # get estimated variance
      bind_cols(
        glance(fit) %>%
          select(sigma) %>%
          mutate(sigma_sq = sigma^2)
      ) %>%
      # add realization identifier
      mutate(
        realization = i,
        ar = ar,
        ar_distance = abs(ar1 - ar),
        ma = ma,
        ma_distance = abs(ma1 - ma),
        sigma = sd,
        sigma_sq_distance = abs(sigma - sigma_sq)
      ) %>%
      # output final format
      select(
        realization,
        estimate_ar = ar1,
        ar,
        ar_distance,
        estimate_ma = ma1,
        ma,
        ma_distance,
        estimate_sigma_sq = sigma_sq,
        sigma,
        sigma_sq_distance
      )
  }) %>%
  # bind list(rows) to single tibble
```

```

    bind_rows
  }

simulateARIMA(order = c(1,0,1), ar = 0.9, ma = 0.5, n = 200) %>%
  kable(
    digits = 4,
    caption = "Estimated values for 10 Realizations of an ARMA(1,1) Process",
    col.names = c("Realization", "Estimate", "Actual", "Distance", "Estimate", "Actual", "Distance", "E
  ) %>%
  add_header_above(c(" ", "AR" = 3, "MA" = 3, "Variance" = 3)) %>%
  kable_styling(full_width = FALSE, protect_latex = TRUE, latex_options = "hold_position")

```

Table 2: Estimated values for 10 Realizations of an ARMA(1,1) Process

| Realization | AR | | | MA | | | Variance | | |
|-------------|----------|--------|----------|----------|--------|----------|----------|--------|----------|
| | Estimate | Actual | Distance | Estimate | Actual | Distance | Estimate | Actual | Distance |
| 1 | 0.8126 | 0.9 | 0.0874 | 0.5800 | 0.5 | 0.0800 | 0.8677 | 1 | 0.1323 |
| 2 | 0.8734 | 0.9 | 0.0266 | 0.4249 | 0.5 | 0.0751 | 0.9667 | 1 | 0.0333 |
| 3 | 0.9165 | 0.9 | 0.0165 | 0.4879 | 0.5 | 0.0121 | 1.0904 | 1 | 0.0904 |
| 4 | 0.8949 | 0.9 | 0.0051 | 0.4689 | 0.5 | 0.0311 | 1.1487 | 1 | 0.1487 |
| 5 | 0.8937 | 0.9 | 0.0063 | 0.5491 | 0.5 | 0.0491 | 0.9173 | 1 | 0.0827 |
| 6 | 0.8828 | 0.9 | 0.0172 | 0.5558 | 0.5 | 0.0558 | 0.8903 | 1 | 0.1097 |
| 7 | 0.8513 | 0.9 | 0.0487 | 0.5302 | 0.5 | 0.0302 | 1.0798 | 1 | 0.0798 |
| 8 | 0.7647 | 0.9 | 0.1353 | 0.4126 | 0.5 | 0.0874 | 0.9474 | 1 | 0.0526 |
| 9 | 0.8993 | 0.9 | 0.0007 | 0.5954 | 0.5 | 0.0954 | 0.9002 | 1 | 0.0998 |
| 10 | 0.9019 | 0.9 | 0.0019 | 0.4781 | 0.5 | 0.0219 | 0.9284 | 1 | 0.0716 |

3.9 ext

Continue with 3.9, generate 10,000 realizations and compare the mean of the estimates to the true values

The mean of the estimates are closer to the true parameter values which is proof of the Central Limit Theorem at work.

```

simulateARIMA(order = c(1,0,1), ar = 0.9, ma = 0.5, n = 200, realizations = 10000) %>%
  summarise_at(c("estimate_ar", "estimate_ma", "estimate_sigma_sq"), list(mean)) %>%
  kable(
    digits = 4,
    caption = "Mean of estimated values over 10,000 Realizations of an ARMA(1,1) Process"
  ) %>%
  kable_styling(full_width = FALSE, protect_latex = TRUE, latex_options = "hold_position")

```

Table 3: Mean of estimated values over 10,000 Realizations of an ARMA(1,1) Process

| estimate_ar | estimate_ma | estimate_sigma_sq |
|-------------|-------------|-------------------|
| 0.8798 | 0.5085 | 0.9851 |