Q3

we fit AR(2) model to dataset using least square & Yule-Walker methods, compare result.

3.1 Fit AR(2) Model

astsa package has *cmort* dataset. I use *forecast* & *stats* for modeling.

```
library(astsa)
```

```
## Warning: package 'astsa' was built under R version 4.3.2
```

```
library(forecast)
```

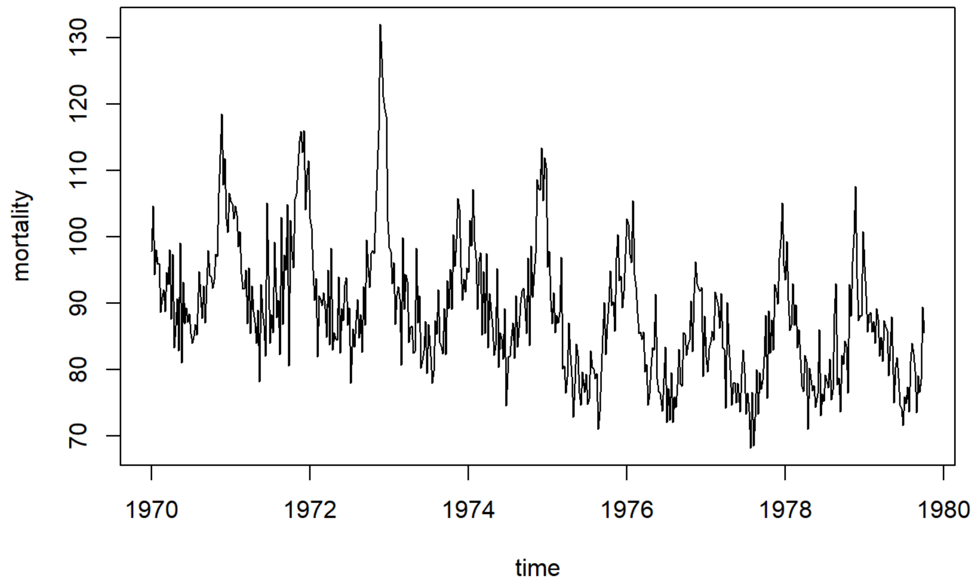
```
## Warning: package 'forecast' was built under R version 4.3.3
```

```
library(stats)
```

cmort Data

I load and visualize dataset to see some trend.

```
data(cmort)
plot(cmort,ylab="mortality",xlab="time")
```



3.2 Fitting AR(2) Model with Least Squares

From notes, AR(2) model is:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \epsilon_t$$

- X_t: value at time t
- ϕ_1, ϕ_2: params to estimate
- ϵ_t: error term (white noise, random fluctuation not captured by model).

Fitting Model

Using `ar.ols` function, I fit AR(2) model.

```
r1<-ar.ols(cmort,order=2)
c1<-r1$ar
s1<-r1$var.pred
se1<-r1$asy.se.coef
c1
```

```
## , , 1
##
##           [,1]
## [1,] 0.4285906
## [2,] 0.4417874
```

- *coefficients*: estimated values for ϕ_1 & ϕ_2 show how much previous observatoins affect current value.

```
s1
```

```
## [1] 32.31749
```

- *var*: value of σ^2 is estimated error in model prediction. smaller variance means model fits data and prediction are more consistent.

```
se1
```

```
## $x.mean
## [1] 0.2527231
##
## $ar
## [1] 0.03979433 0.03976163
```

- *std errors*: shows reliability of estimate. smaller std error => estimate are statistically significant.

also variables i used were:

- **r1**: AR(2) model fitted using least squares.
- **c1**: estimated coeffs ϕ_1 and ϕ_2 from least squares.
- **s1**: Variance of errors from least squares.
- **se1**: Std. errors for ϕ_1 & ϕ_2 from least squares.

3.3 Forecasting

i use fitted model to forecast next 8week.

Forecast calculation from classnotes:

$$X_{T+m} = \phi_1 \cdot X_T + \phi_2 \cdot X_{T-1}$$

equation is projecting future values based on fitted model params.

I will calculate 95% prediction interval to quantify uncertainty in forecast.

```
h<-8  # No. of period to forecast(8 weeks)
f_v<-predict(r1,n.ahead=h)  # to store predicted future values
p_v<-f_v$pred     # Point forecast for next 8 weeks
s_e<-f_v$se                  # Std errors with forecasts

u_b<-p_v+1.96*s_e      # uper limit of prediction interval
l_b<-p_v-1.96*s_e  #lower limit of prediction interval

f_r<-data.frame(
  W=1:h,
  F=p_v,
  L_95_CI=l_b,
  U_95_CI=u_b
)

f_r      #Table of forecasted values & CI.
```

```
##   W        F  L_95_CI   U_95_CI
## 1 1 87.59986 76.45756  98.74217
## 2 2 86.76349 74.64094  98.88604
## 3 3 87.33714 73.35405 101.32022
## 4 4 87.21350 72.33052 102.09648
## 5 5 87.41394 71.62769 103.20019
## 6 6 87.44522 71.02807 103.86238
## 7 7 87.54719 70.58705 104.50732
## 8 8 87.60471 70.22059 104.98882
```

plot with 95% prediction intervals

I plot forecast with 95% prediction intervals.

```
plot(1:h,p_v,type="l",col="blue",ylim=c(min(l_b),max(u_b)),
     ylab="Forecast",xlab="Weeks Ahead",main="AR(2) Forecast with 95% Prediction Intervals")
lines(1:h,u_b,col="red",lty=2)  # uperbound line in red
lines(1:h,l_b,col="pink",lty=2)  # lowerbound line in pink
```

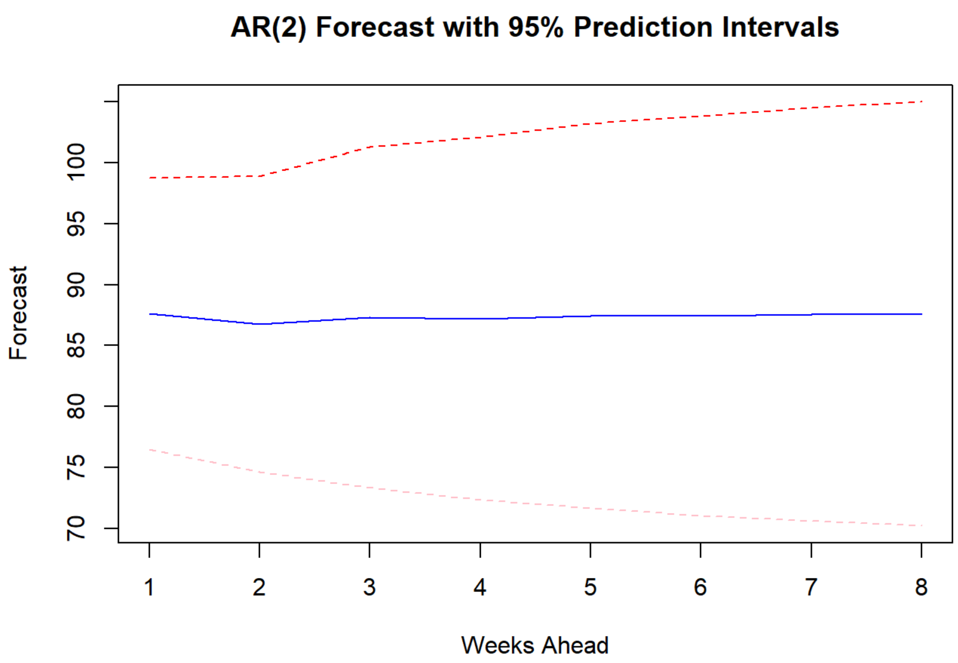


table of forecasts shows predicted mortality values. also bounds of 95%prediction interval. we can see uncertainty with predictions. closer bounds are to forecast values, more certain I am about predictions.

3.4 Fitting AR(2) Model with Yule-Walker Equations

Next, I use Yule-Walker equations to estimate parameters for AR(2) model:

$$\phi_1 = \frac{\gamma(1)}{\gamma(0)}$$
$$\phi_2 = \frac{\gamma(2) - \phi_1\gamma(1)}{\gamma(0)}$$

here $\gamma(k)$ is autocovariance function at lag k.

model fitusing Yule-Walker

Using `ar.yw` , I will fit AR(2) model.

```
r2<-ar.yw(cmort,order=2)  # fit AR(2)
c2<-r2$ar            # estimated coeffs (phi_1 & phi_2)
s2<-r2$var.pred     # Var of prediction error
se2<-sqrt(diag(r2$asy.var.coef))  # Std errors for phi_1, phi_2
c2
```

```
## [1] 0.4339481 0.4375768
```

- *Coeffs*: ϕ_1 and ϕ_2 derived from ACF to see past values influence on current value, similar to least squares method.

```
s2
```

```
## [1] 32.84056
```

- *var*: noise var estimated using Yule-Walker to see uncertainty in prediction.

```
se2
```

```
## [1] 0.04001303 0.04001303
```

- *std errors*:shows reliability of estimates.

variables i used :

- **r2**: AR(2) model fitted using Yule-Walker equations.
- **c2**: Estimated coefficients ϕ_1 and ϕ_2 from Yule-Walker.
- **s2**: Var of errors from Yule-Walker.
- **se2**: std errors for ϕ_1 ,ϕ_2 from Yule-Walker.

3.5 Compare Results

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.3.3
```

```
c<-data.frame(
  M=c("Least Squares","Yule-Walker"),  # Method names for comparison
  C_1=c(c1[1],c2[1]),  # First coefficient (phi_1)
  C_2=c(c1[2],c2[2]),  # Second coefficient (phi_2)
  S=c(s1,s2),          # Estimated variances
  SE_C_1=c(se1[1],se2[1]),  # Standard errors for first coefficient
  SE_C_2=c(se1[2],se2[2])   # Standard errors for second coefficient
)

c_t<-kable(c,digits=10,caption="Comparison of AR(2) Estimates") %>%
  kable_styling("striped",full_width=F) %>%
  column_spec(1:6,border_left=TRUE,border_right=TRUE)

c_t
```

Comparison of AR(2) Estimates

M	C_1	C_2	S	SE_C_1.x.mean	SE_C_1.0.0400130313449369	SE_C_2.ar	SE_C_2.0.0400130313449369
Least Squares	0.4285906	0.4417874	32.31749	0.2527231	0.04001303	0.03979433	0.04001303
Yule-Walker	0.4339481	0.4375768	32.84056	0.2527231	0.04001303	0.03976163	0.04001303

AR(2) to cmort dataset using LS and YW methods provides similar estimate. small differenc in coeffs and std error => both methods are consistent.