ST565: Time Series HW5

Amirhosein "Emerson" Azarbakht azarbaka@oregonstate.edu

Required reading 5.4 & 5.5 in Chatfield

Question 1

Fit a seasonal ARIMA model to the Johnson and Johnson quarterly returns data in the package astsa and forecast the next 12 quarters (including prediction intervals).

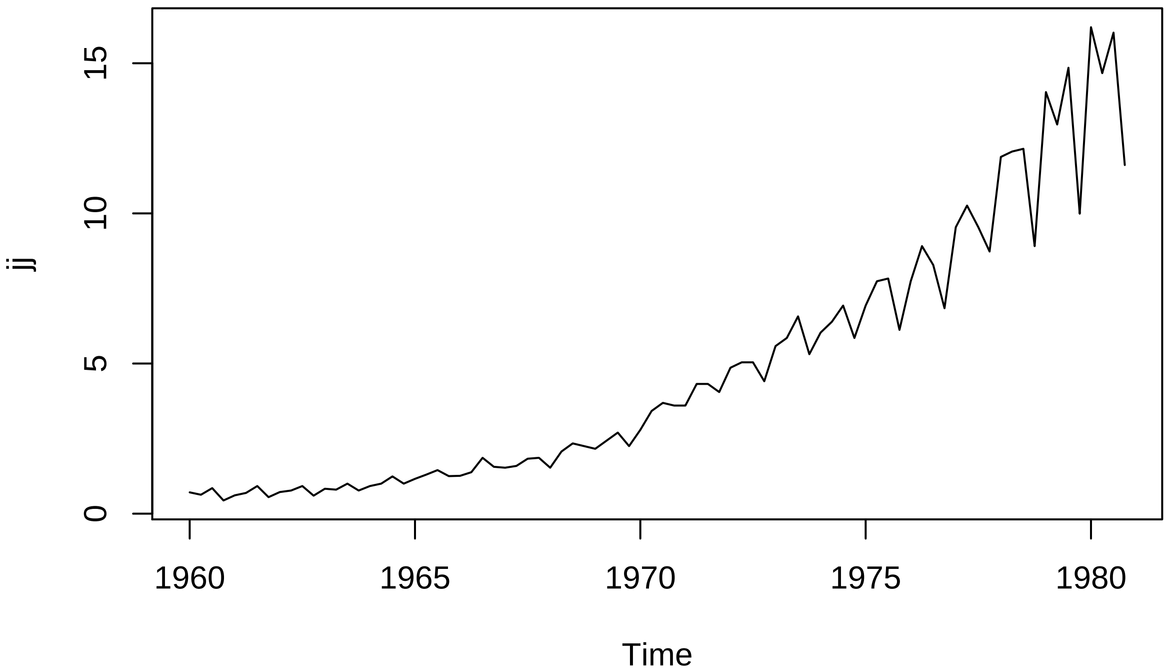
```r
# install.packages("astsa")
library(forecast)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: timeDate

## This is forecast 6.2
```

```r
library(astsa)
```

```
##
## Attaching package: 'astsa'

## The following object is masked from 'package:forecast':
##
##     gas
```

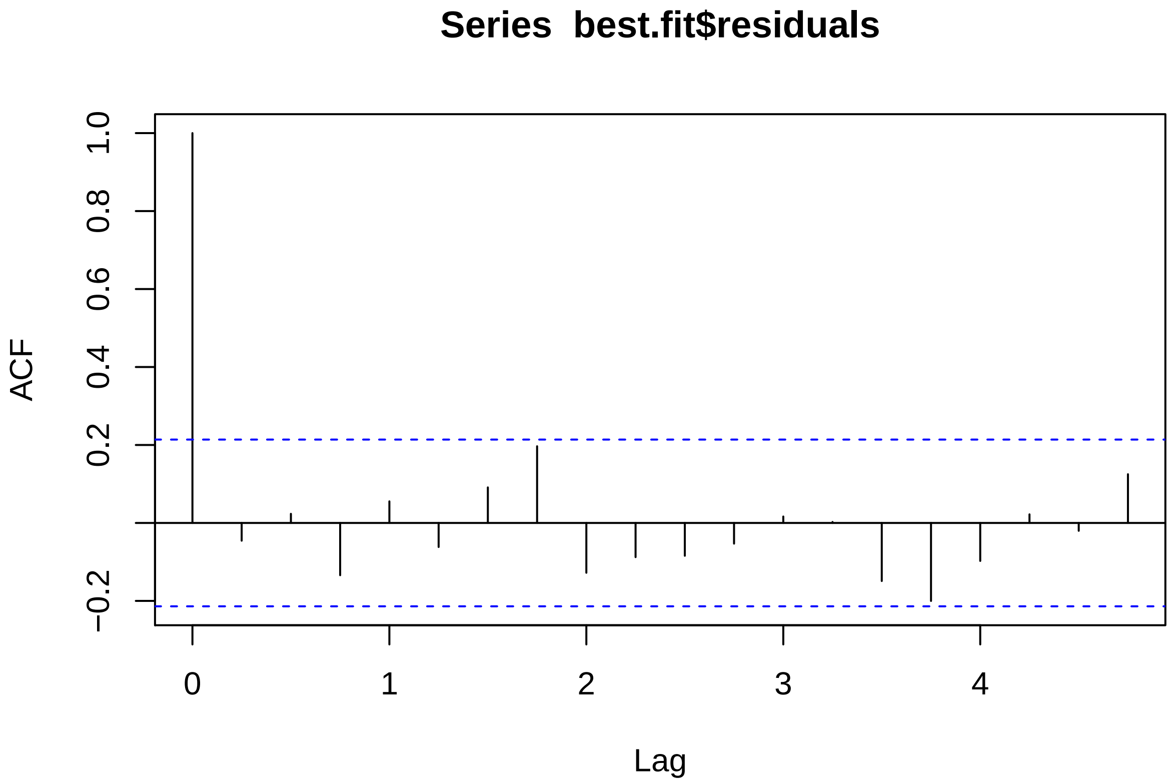
```r
data(jj)
ts.plot(jj)
```



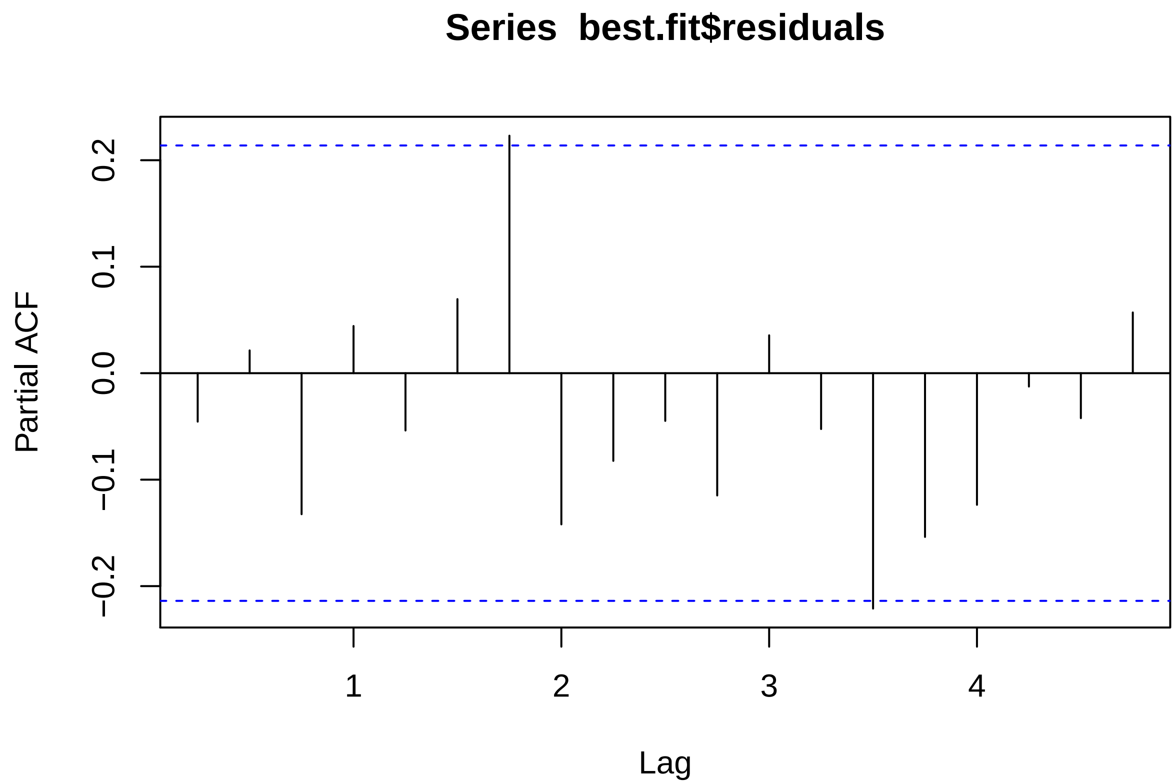

```
get.best.arima <- function(x.ts, maxord = c(1,1,1,1,1,1)) {
  best.aic <- 1e8
  n <- length(x.ts)
  for (p in 0:maxord[1]) for(d in 1:maxord[2]) for(q in 0:maxord[3])
    for (P in 0:maxord[4]) for(D in 1:maxord[5]) for(Q in 0:maxord[6])
      {
      fit <- arima(x.ts, order = c(p,d,q), seas = list(order = c(P,D,Q), frequency(x.ts)), method = "CS
      fit.aic <- -2 * fit$loglik + (log(n) + 1) * length(fit$coef)
      if (fit.aic < best.aic)
        {
        best.aic <- fit.aic
        best.fit <- fit
        best.model <- c(p,d,q,P,D,Q)
        }
      }
  list(best.aic, best.fit, best.model)
}


best.arima <- get.best.arima(jj, maxord = c(2,2,2,2,2,2))
```

```
## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1
```

```r
best.fit <- best.arima[[2]]

acf(best.fit$residuals)
```



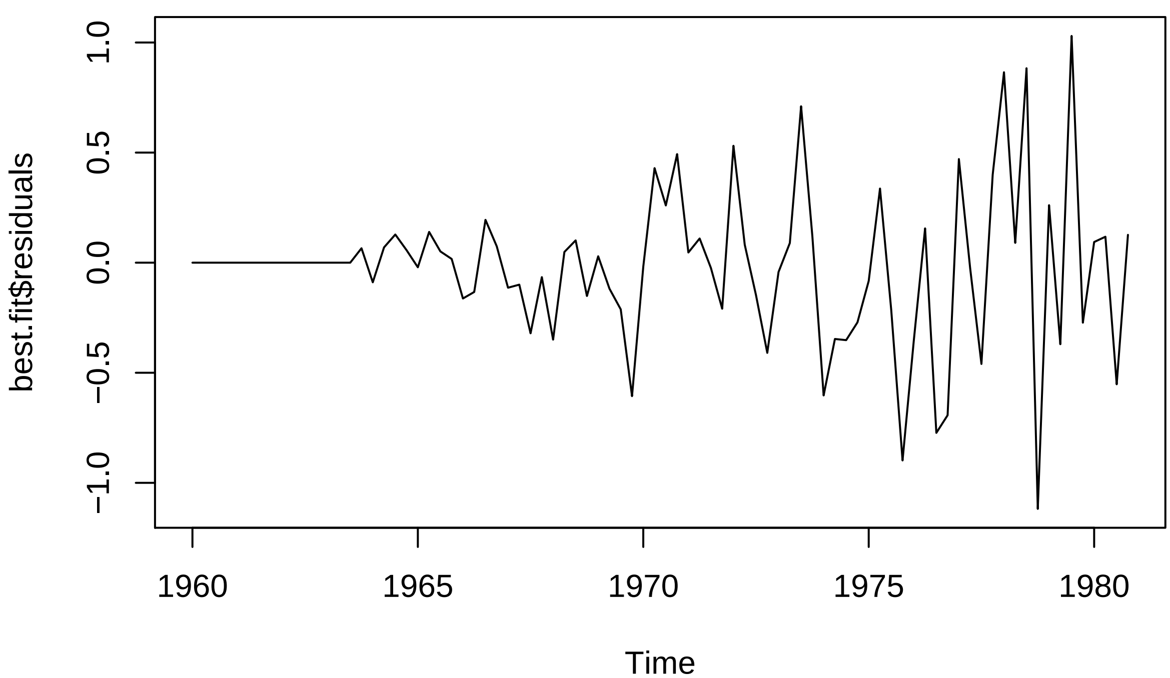

```r
pacf(best.fit$residuals)
```



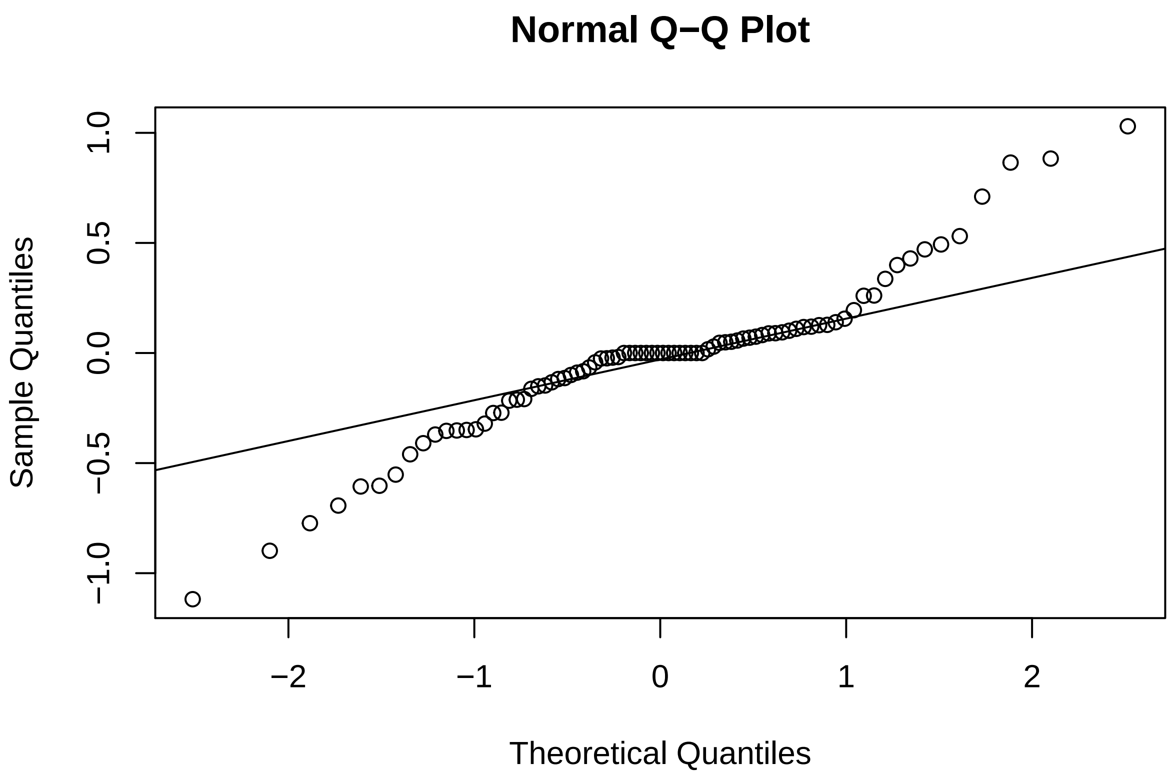

```r
best.arima [[3]]
```

```
## [1] 2 1 1 1 2 1
```

```r
plot(best.fit$residuals)
```



```r
qqnorm(best.fit$residuals)
qqline(best.fit$residuals)
```



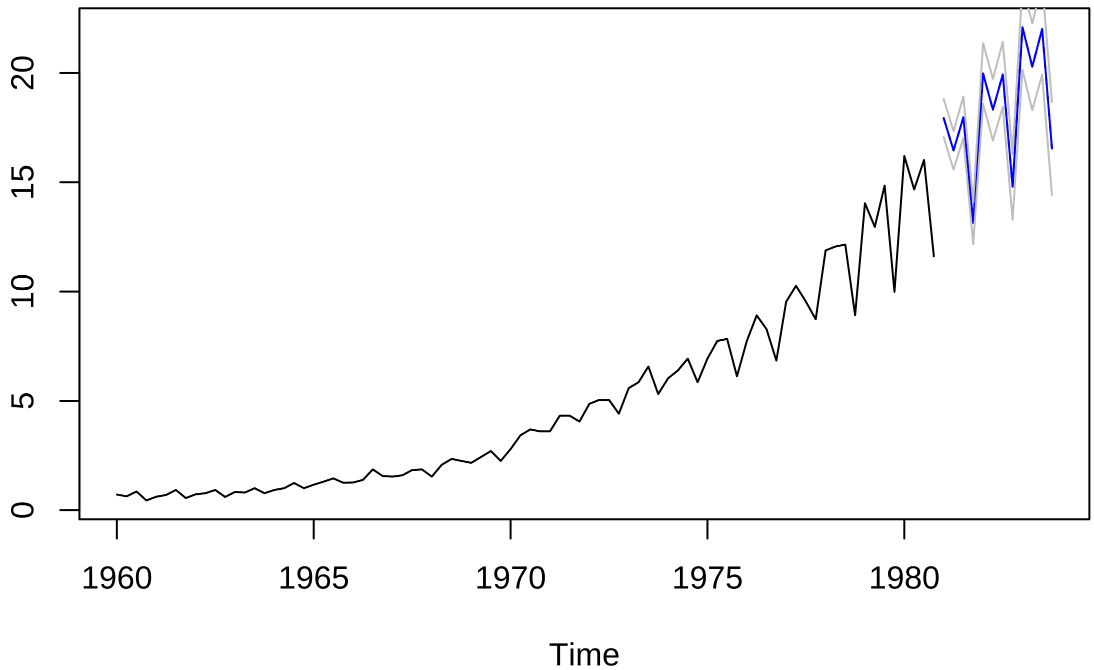

```r
prediction <- predict(best.fit, 12)
```

```
## Warning in predict.Arima(best.fit, 12): MA part of model is not invertible
```

```r
prediction
```

```
## $pred
##          Qtr1     Qtr2     Qtr3     Qtr4
## 1981 17.93643 16.45906 17.97239 13.13971
## 1982 19.97997 18.32015 19.93535 14.79452
## 1983 22.09596 20.28372 22.01715 16.54413
##
## $se
##           Qtr1      Qtr2      Qtr3      Qtr4
## 1981 0.4364227 0.4394069 0.4734084 0.4777317
## 1982 0.6944537 0.7038458 0.7439318 0.7547596
## 1983 0.9765810 0.9937175 1.0456149 1.0642018
```

```r
ts.plot(cbind(window(jj,start = 1960), prediction$pred), lty = 1:2)

lines(prediction$pred, lty = 1:2, col = "blue")
lines(prediction$pred - (2 * prediction$se), lty = 1:2, col = "grey")
lines(prediction$pred + (2 * prediction$se), lty = 1:2, col = "grey")
```



Question 2: The Holt Winters Method

What decisions need to be made to use a Holt Winters forecasting approach? What starting values do you need to specify? What parameters need estimating?

- If data contains trend and seasonality, we can use Holt Winters method, rather than Simple Exponential Smoothing method.
- Plot the data, examine whether the variance is constant or varying through time. If former case, choose an additive method, if latter case, choose a multiplicative method.
- We need to specify starting values for L_1, T_1, and $I_{1\ldots s}$ for the first year, using the first few observations in the time series. For example, we can use $L_1 = \sum x_i/s$.
- Values for α, γ, δ need to be estimated by minimizing the square one-step prediction error $\sum e_t^2$.
- Decide whether to normalize the seasonal indices at reguilar intervals by making them sum to zero for the additive case, and sum to 1 for the multiplicative case.
- Choose an automatic approach or non-automatic.

Investigate the R function HoltWinters. How do you specify the decisions from above? How does the function choose starting values and estimate parameters?

Using the function options, e.g. seasonal = c("additive", "multiplicative"), etc.

```
# HoltWinters(x, alpha = NULL, beta = NULL, gamma = NULL, seasonal = c("additive", "multiplicative"), s
```

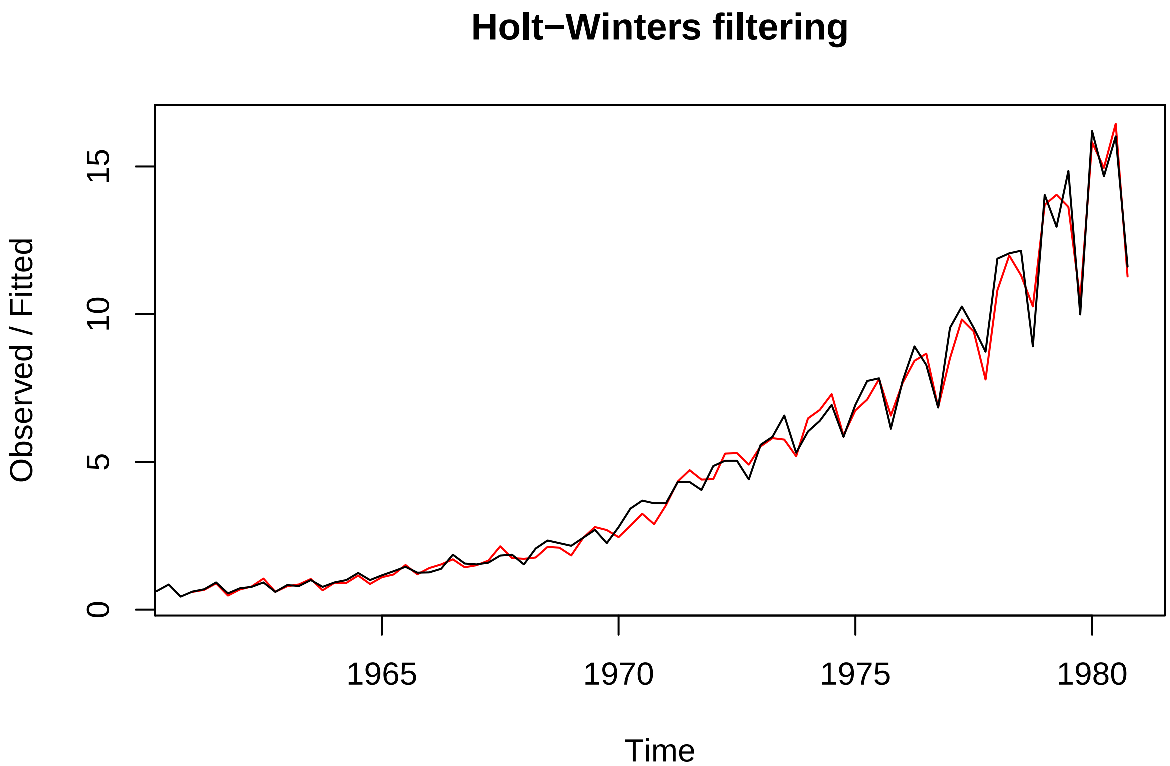
According to R documentation: "For seasonal models, start values for a, b and s are inferred by performing a simple decomposition in trend and seasonal component using moving averages (see function decompose) on the start.periods first periods (a simple linear regression on the trend component is used for starting level and trend). For level/trend-models (no seasonal component), start values for a and b are x[2] and x[2] - x[1], respectively. For level-only models (ordinary exponential smoothing), the start value for a is x[1]."

Use the function to produce forecasts (along with prediction intervals) for Johnson and Johnson returns in question 1.

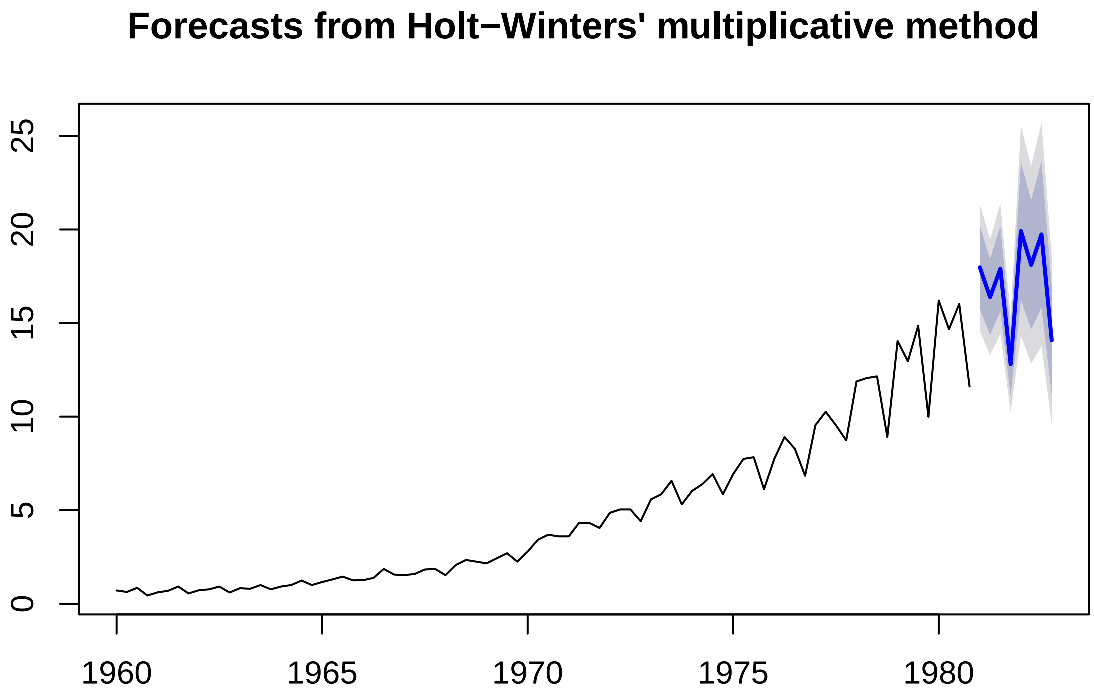
```
hwt <- HoltWinters(jj, seasonal = "multiplicative", start.periods = 2)
hwt$SSE
```

```
## [1] 14.35317
```

```
plot(hwt)
```




```
predictHW <- predict(hwt,n.ahead = 12, prediction.interval = TRUE, level = 0.95)
hwtfit <- hw(jj, seasonal="multiplicative")
plot(hwtfit)
```


Forecasts from Holt−Winters' multiplicative method
0
5
10
15
20
25
1960
1965
1970
1975
1980