

# ST565: Time Series HW5

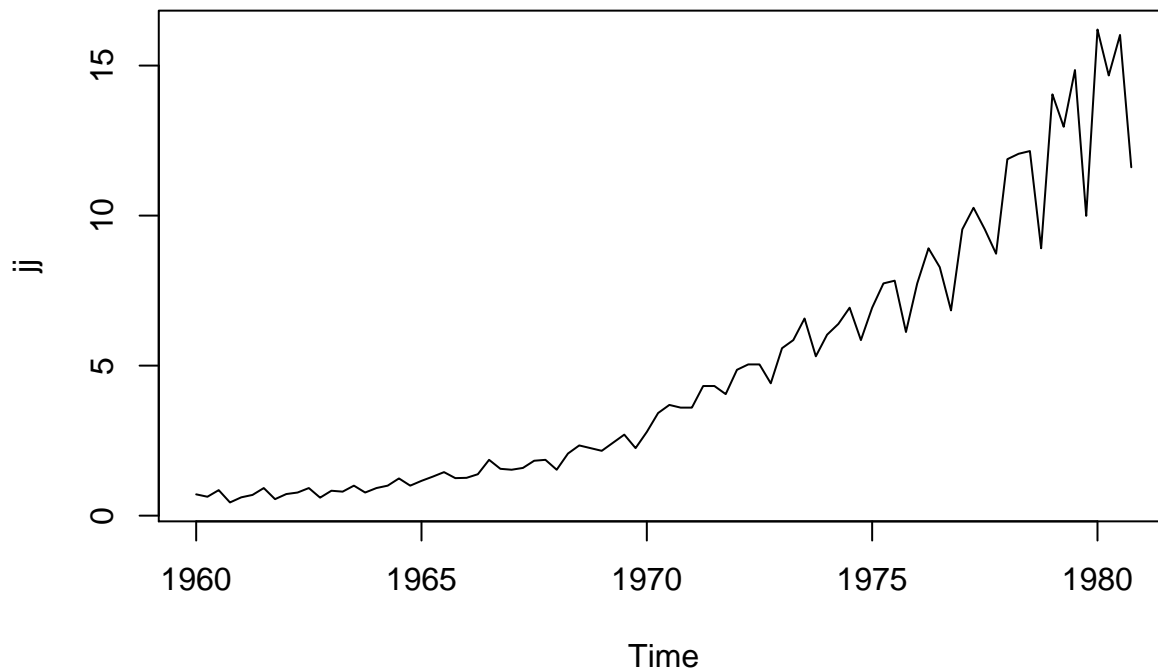
Amirhosein “Emerson” Azarbakht [azarbaka@oregonstate.edu](mailto:azarbaka@oregonstate.edu)

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).

```
# install.packages("astsa")
library(astsa)
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 0:maxord[2]) for(q in 0:maxord[3])
    for (P in 0:maxord[4]) for(D in 0: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 = "CSP")
        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(3,3,3,3,3,3))
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

```

```

## 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

```

```

## 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

```

```

## 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

```

```

## 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

## 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

## 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

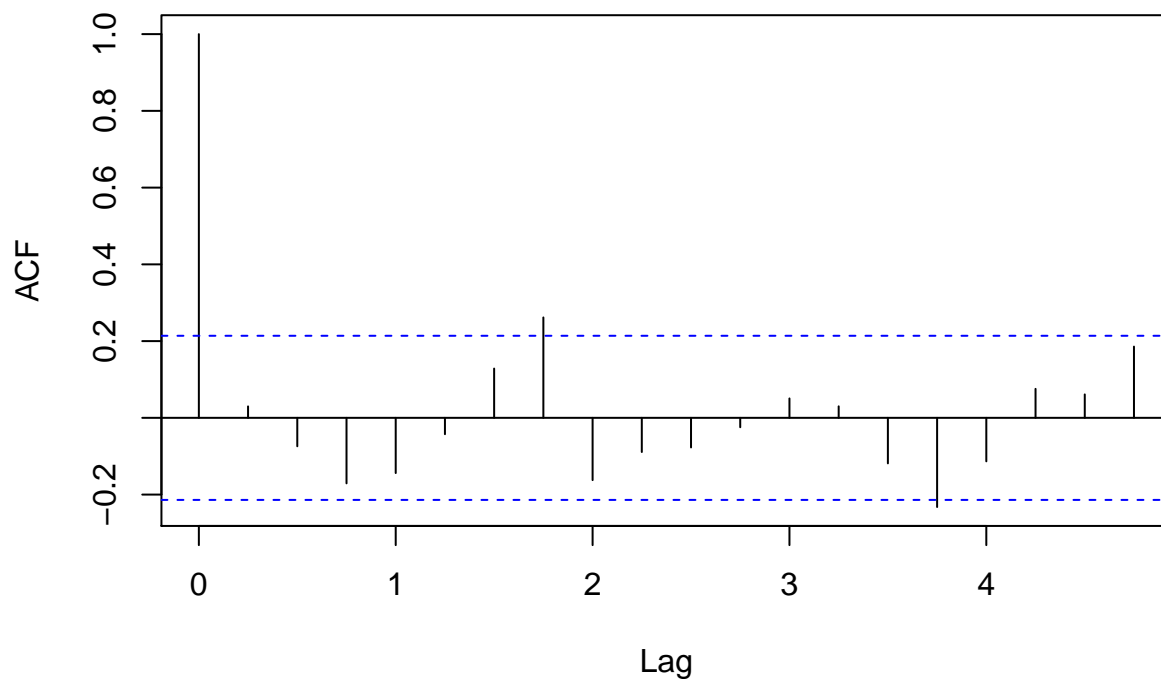
## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

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

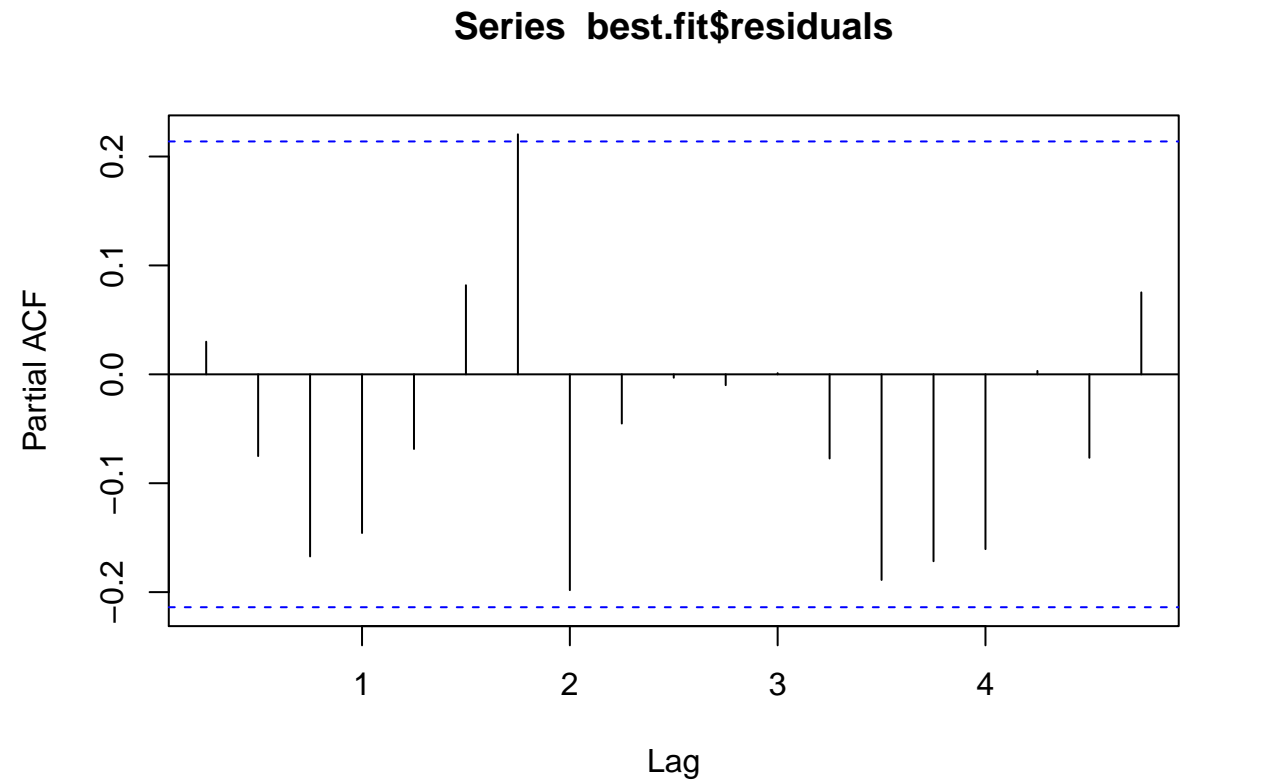
```
best.fit <- best.arima[[2]]
```

```
acf(best.fit$residuals)
```

### Series best.fit\$residuals



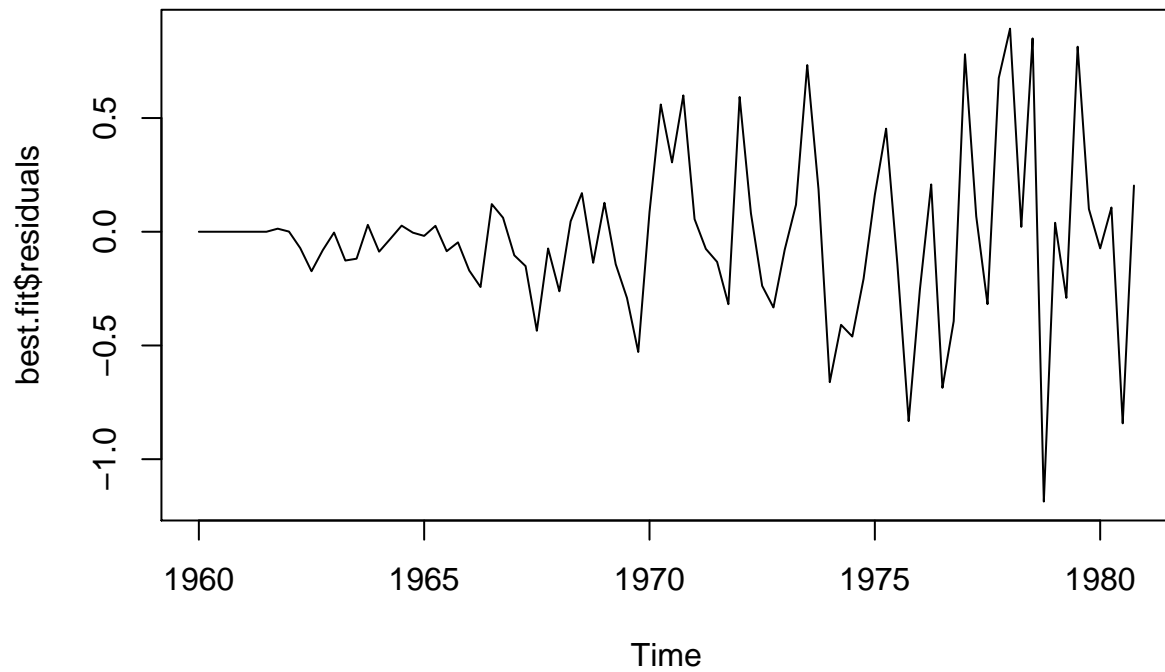
```
pacf(best.fit$residuals)
```



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

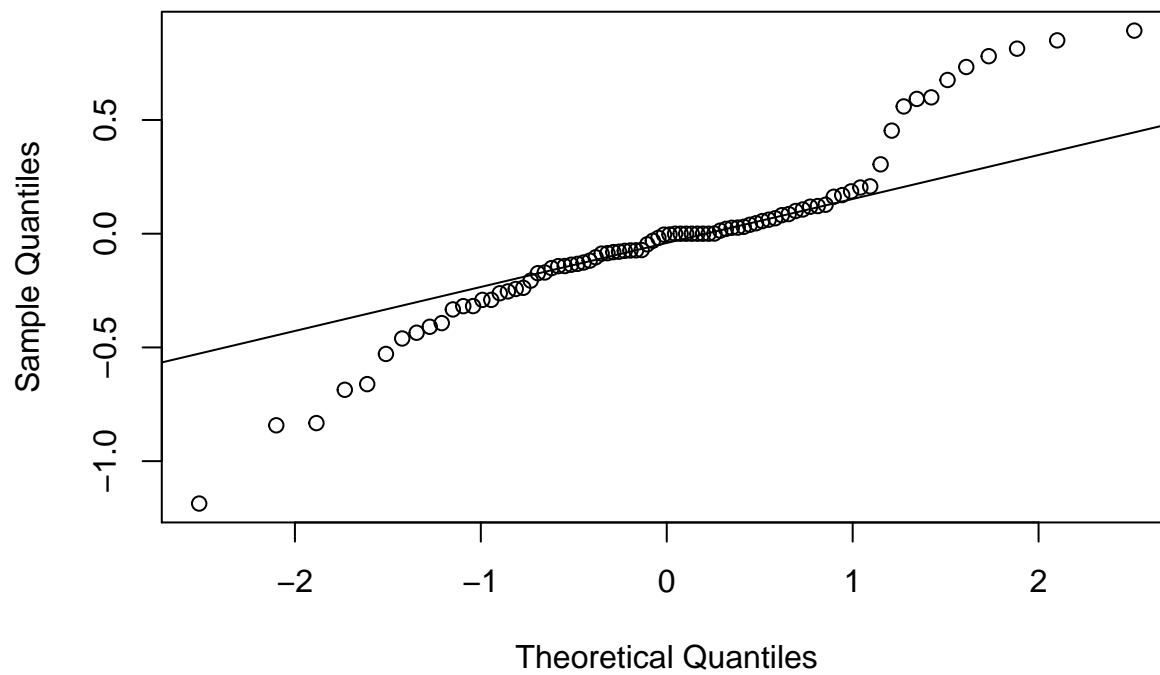
```
## [1] 2 1 1 1 0 0
```

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



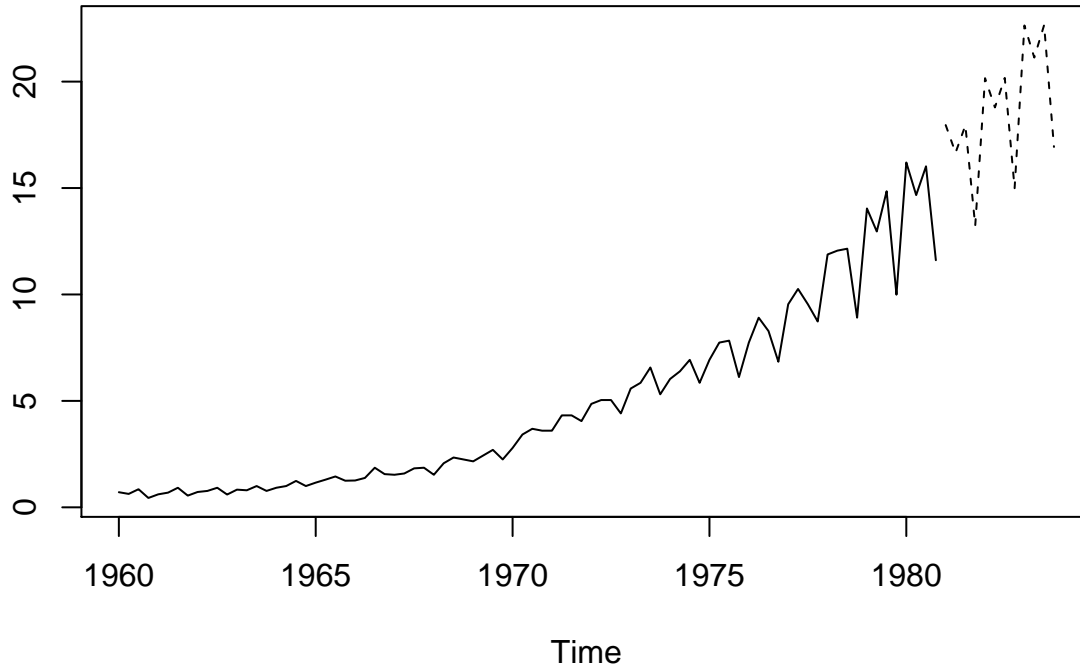
```
qqnorm(best.fit$residuals)  
qqline(best.fit$residuals)
```

### Normal Q-Q Plot



```
ts.plot(cbind(window(jj,start = 1960), predict(best.fit, 12)$pred), lty = 1:2)
```

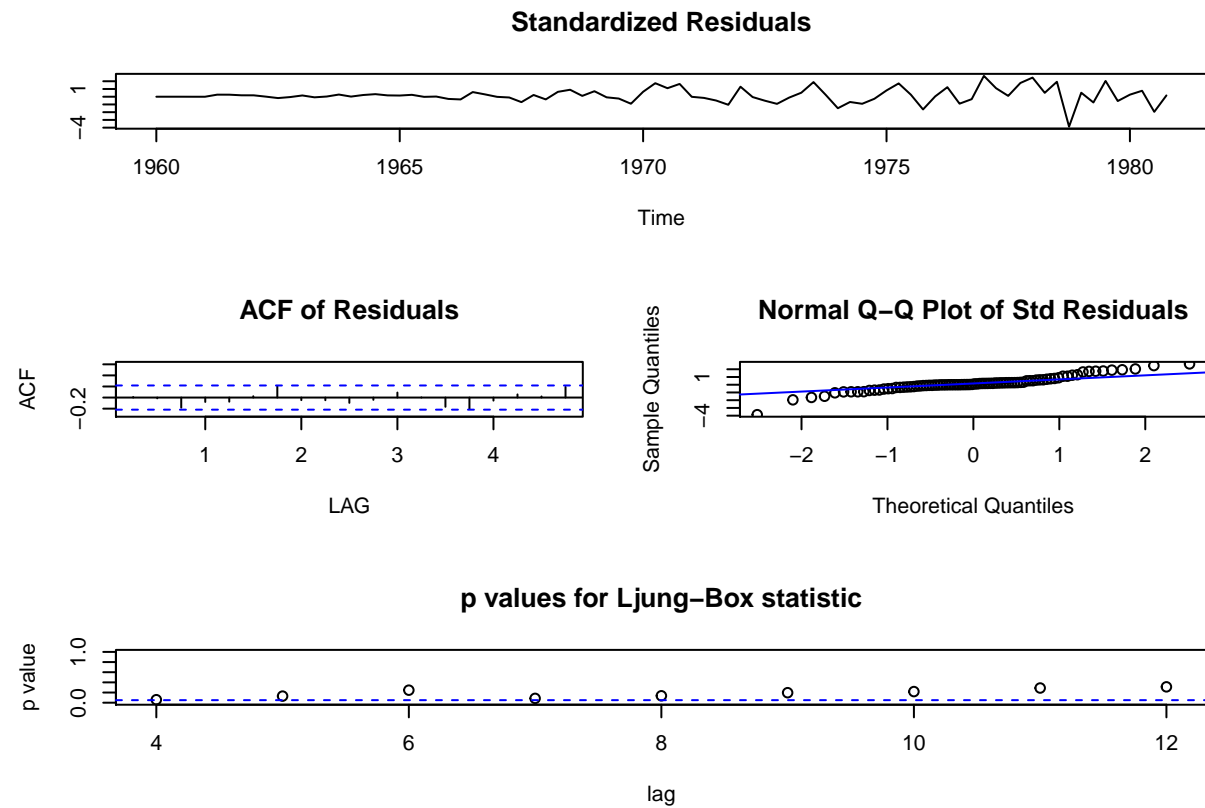
```
## Warning in predict.Arima(best.fit, 12): MA part of model is not invertible
```



```
as1 <- sarima(jj, 1,1,2,0,1,0,4)
```

```
## initial value -0.504277
## iter 2 value -0.550534
## iter 3 value -0.824365
## iter 4 value -0.837005
## iter 5 value -0.841083
## iter 6 value -0.842730
## iter 7 value -0.844963
## iter 8 value -0.859063
## iter 9 value -0.860614
## iter 10 value -0.860949
## iter 11 value -0.861072
## iter 12 value -0.861211
## iter 13 value -0.861238
## iter 14 value -0.861239
## iter 14 value -0.861239
## iter 14 value -0.861239
## final value -0.861239
## converged
## initial value -0.861022
## iter 2 value -0.861101
## iter 3 value -0.861117
## iter 4 value -0.861118
## iter 5 value -0.861118
```

```
## iter 6 value -0.861120
## iter 7 value -0.861120
## iter 8 value -0.861120
## iter 8 value -0.861120
## final value -0.861120
## converged
```

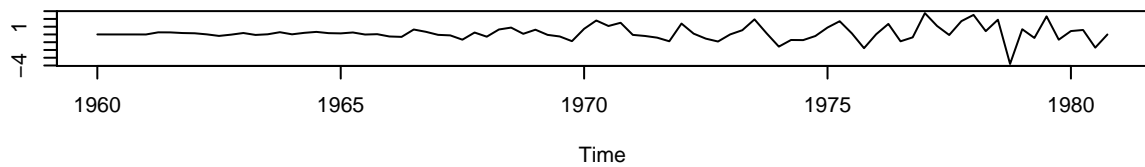


```
as2 <- sarima(jj, 2,1,1,1,1,0,4)
```

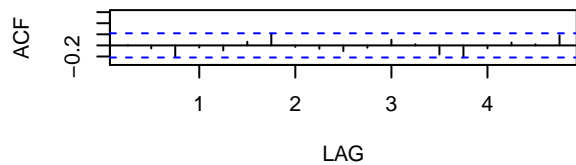
```
## initial value -0.471310
## iter 2 value -0.624500
## iter 3 value -0.783368
## iter 4 value -0.809675
## iter 5 value -0.819967
## iter 6 value -0.822661
## iter 7 value -0.823909
## iter 8 value -0.824921
## iter 9 value -0.825014
## iter 10 value -0.825028
## iter 11 value -0.825028
## iter 12 value -0.825028
## iter 13 value -0.825028
## iter 14 value -0.825028
## iter 14 value -0.825028
## iter 14 value -0.825028
## final value -0.825028
```

```
## converged
## initial value -0.858168
## iter 2 value -0.858205
## iter 3 value -0.858240
## iter 4 value -0.858264
## iter 5 value -0.858279
## iter 6 value -0.858282
## iter 7 value -0.858282
## iter 8 value -0.858282
## iter 8 value -0.858282
## iter 8 value -0.858282
## final value -0.858282
## converged
```

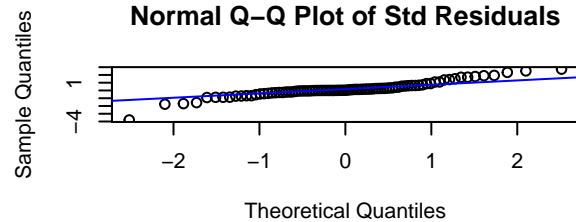
**Standardized Residuals**



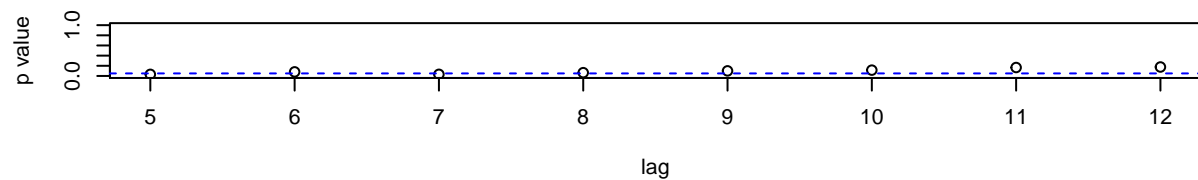
**ACF of Residuals**



**Normal Q-Q Plot of Std Residuals**

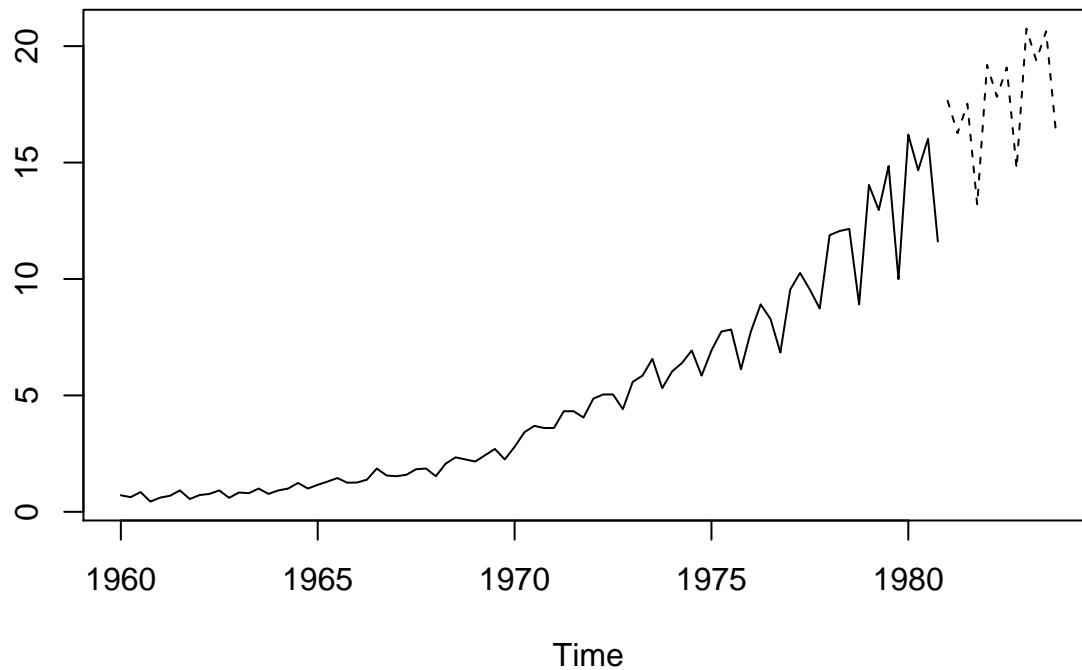


**p values for Ljung-Box statistic**



```
ts.plot(cbind(window(jj,start = 1960), predict(as2$fit, 12)$pred), lty = 1:2)
```





## Question 2: The Holt Winters Method

- Read 5.2.3. in Chatfield
- 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?
- Investigate the R function `HoltWinters`. How do you specify the decisions from above? How does the function choose starting values and estimate parameters?
- Use the function to produce forecasts (along with prediction intervals) for Johnson and Johnson returns in question 1.