ST 595

Project 3

Alex Kretschmer, Ruslan Mamedov

2022-05-15

Introduction

Methods and Results

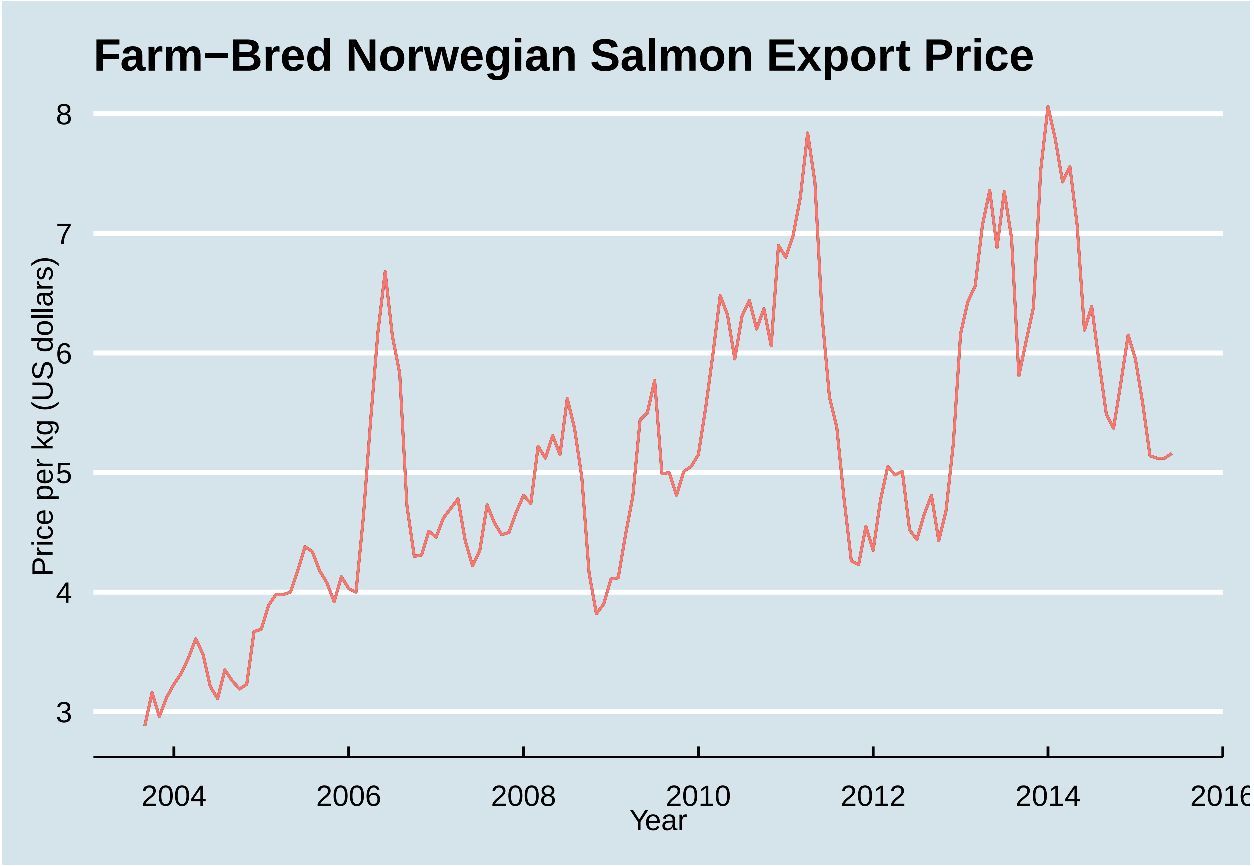
1. Exploratory analysis, simple plots showing overall dataset, trend, seasonality

```
data(salmon)
salmon_test<-ts(salmon[143:166], start = c(2015,7), frequency = 12)
salmon_train<-ts(salmon[1:142], start = c(2003,9), frequency = 12)
```

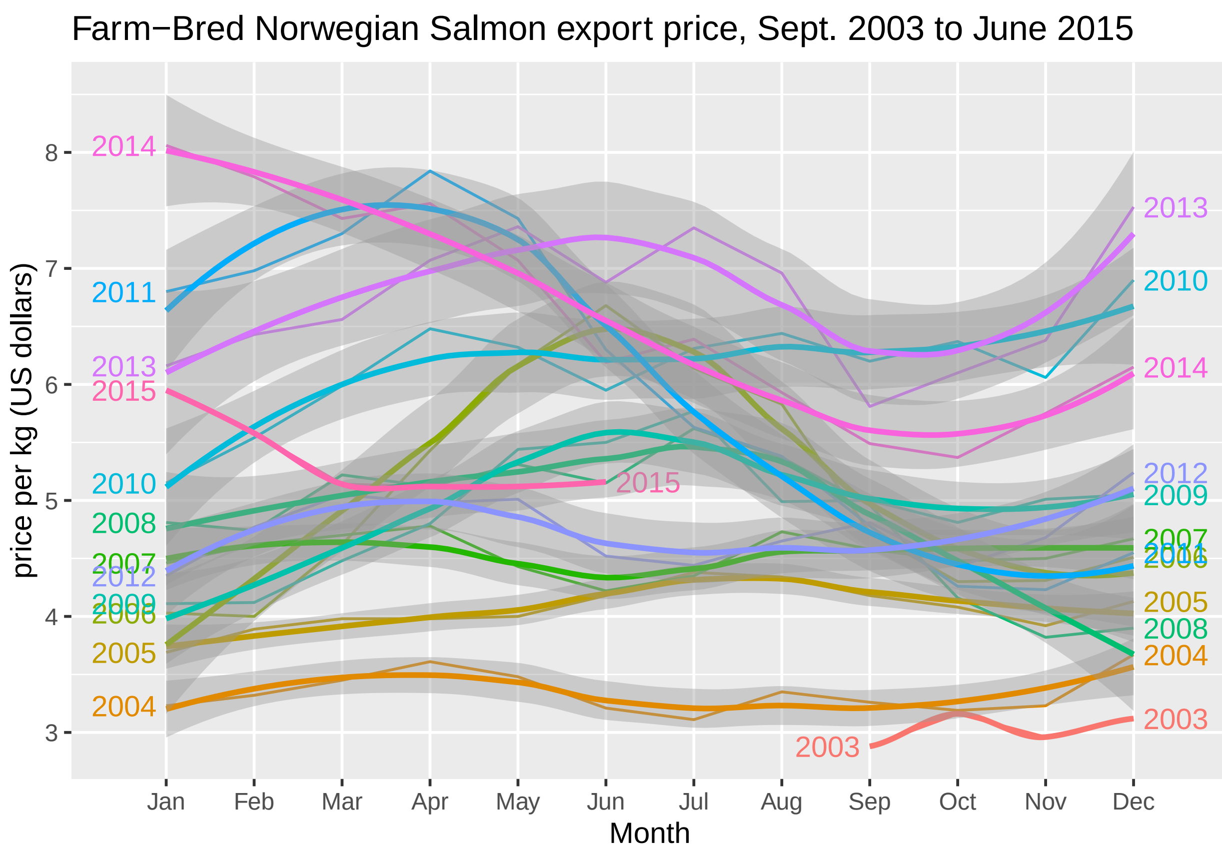
```
head(salmon_train)
```

```
##       Jan  Feb Mar Apr May Jun Jul Aug  Sep  Oct  Nov  Dec
## 2003                                   2.88 3.16 2.96 3.12
## 2004 3.23 3.32
```

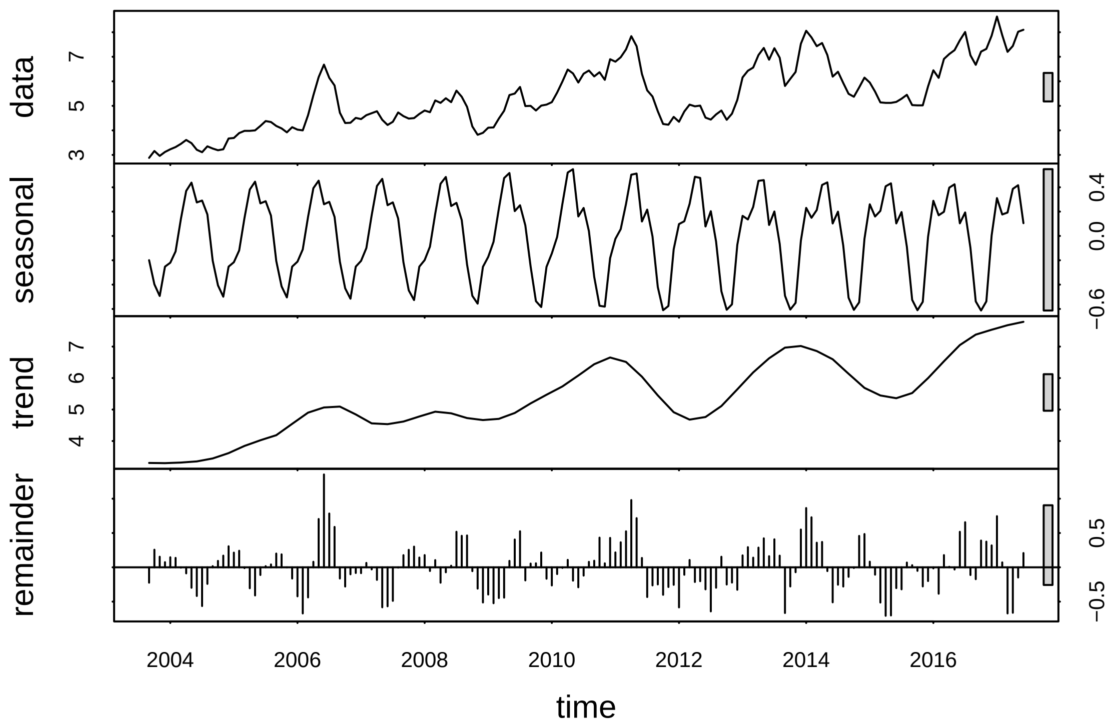
```
#plot
autoplot(salmon_train)+ geom_line( color="#F8766D")+xlab("Year")+ ylab("Price per kg (US dollars)")+ggt
```



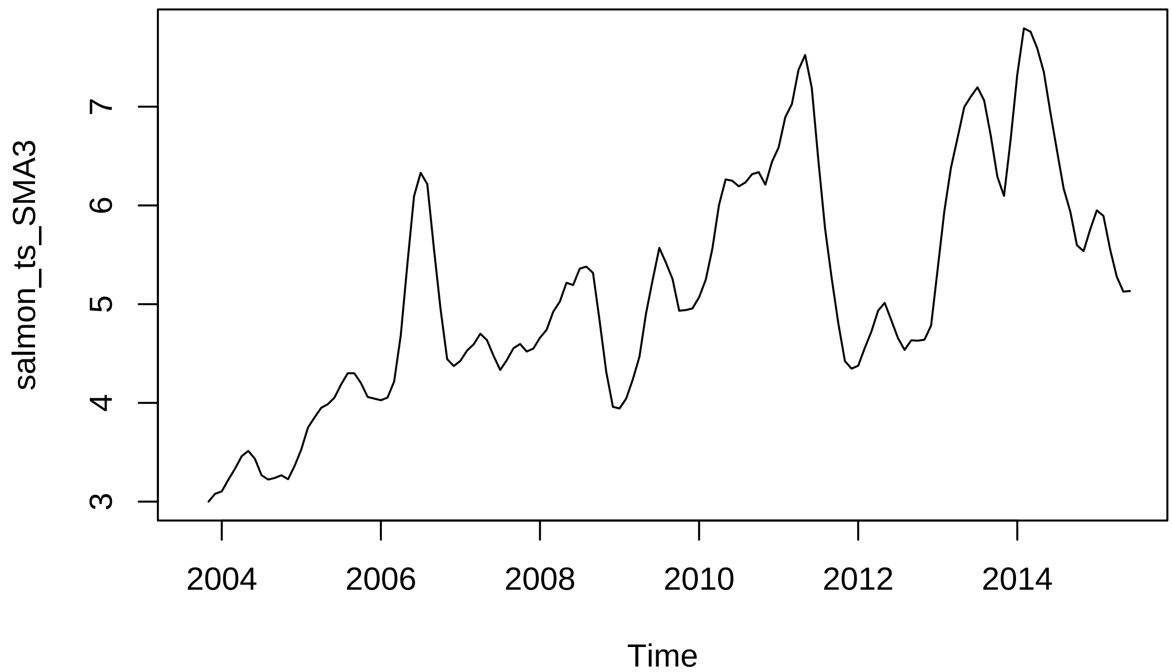

```r
#install.packages("fpp2")
library(fpp2)
ggseasonplot(salmon_train, year.labels=TRUE, year.labels.left=TRUE) +
  ylab("price per kg (US dollars)") + geom_smooth() +
  ggtitle("Farm-Bred Norwegian Salmon export price, Sept. 2003 to June 2015")
```



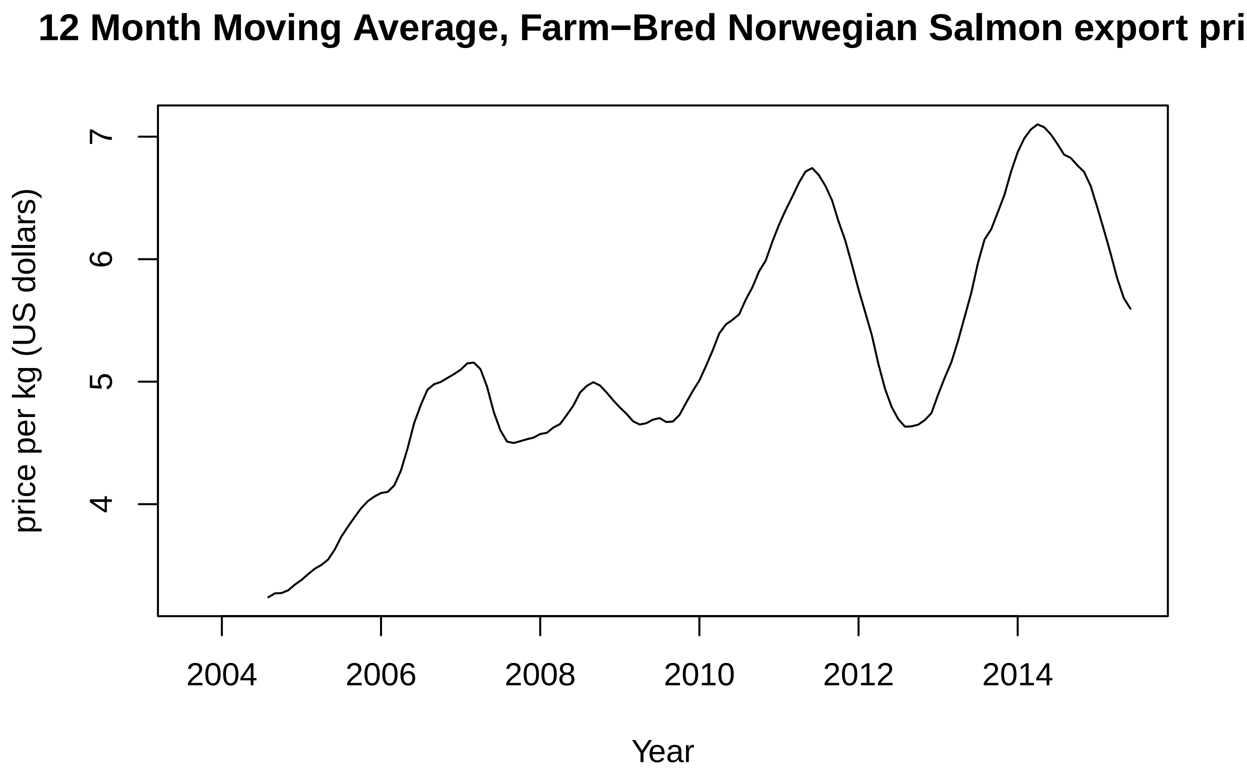

```
fit = stl(salmon, s.window=12)
plot(fit)
```



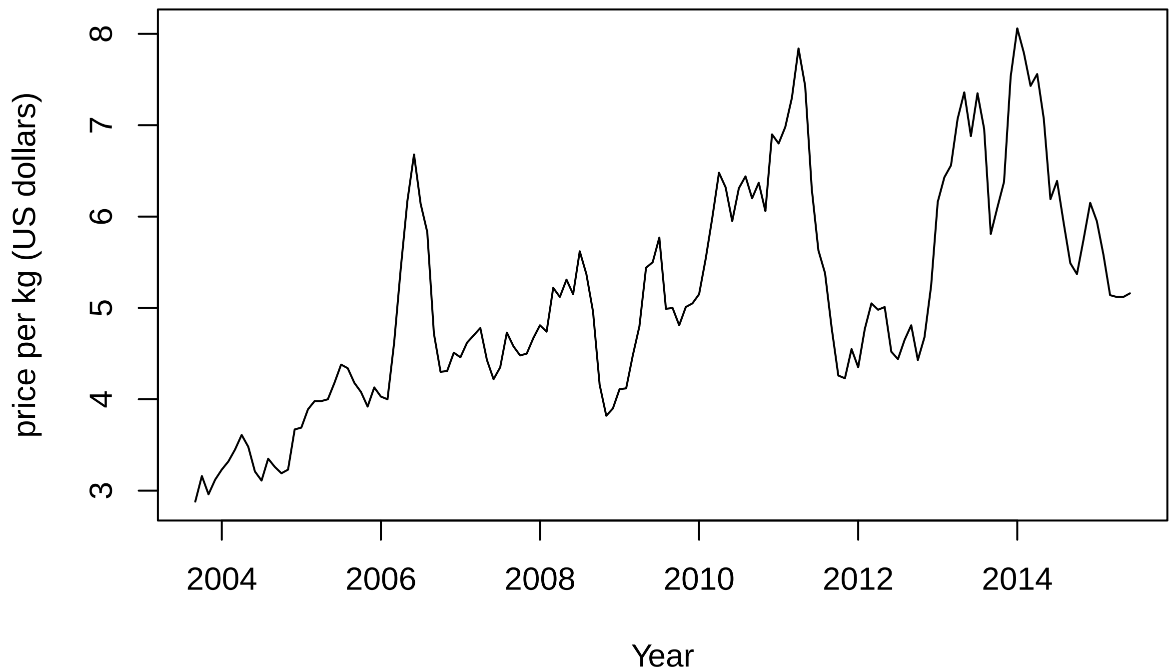

```r
library(TTR)
salmon_ts_SMA3 <- SMA(salmon_train,n=3)
plot.ts(salmon_ts_SMA3)
```



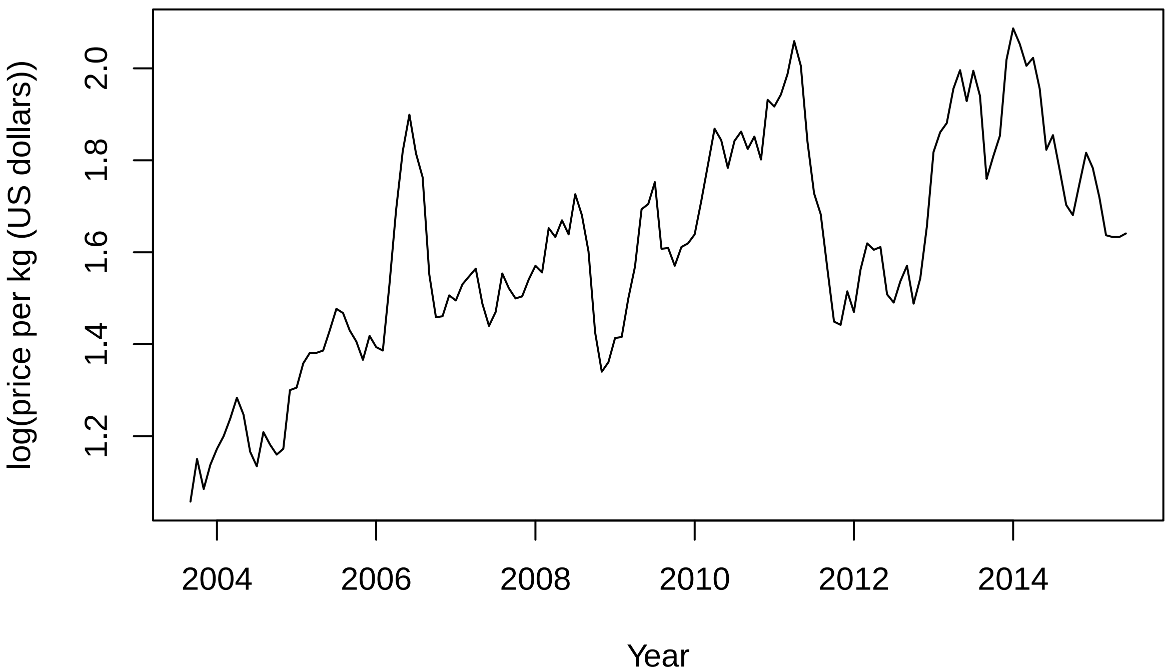

```
salmon_ts_SMA12 <- SMA(salmon_train,n=12)
plot.ts(salmon_ts_SMA12, xlab = "Year", ylab = "price per kg (US dollars)",
main = "12 Month Moving Average, Farm-Bred Norwegian Salmon export price")
```



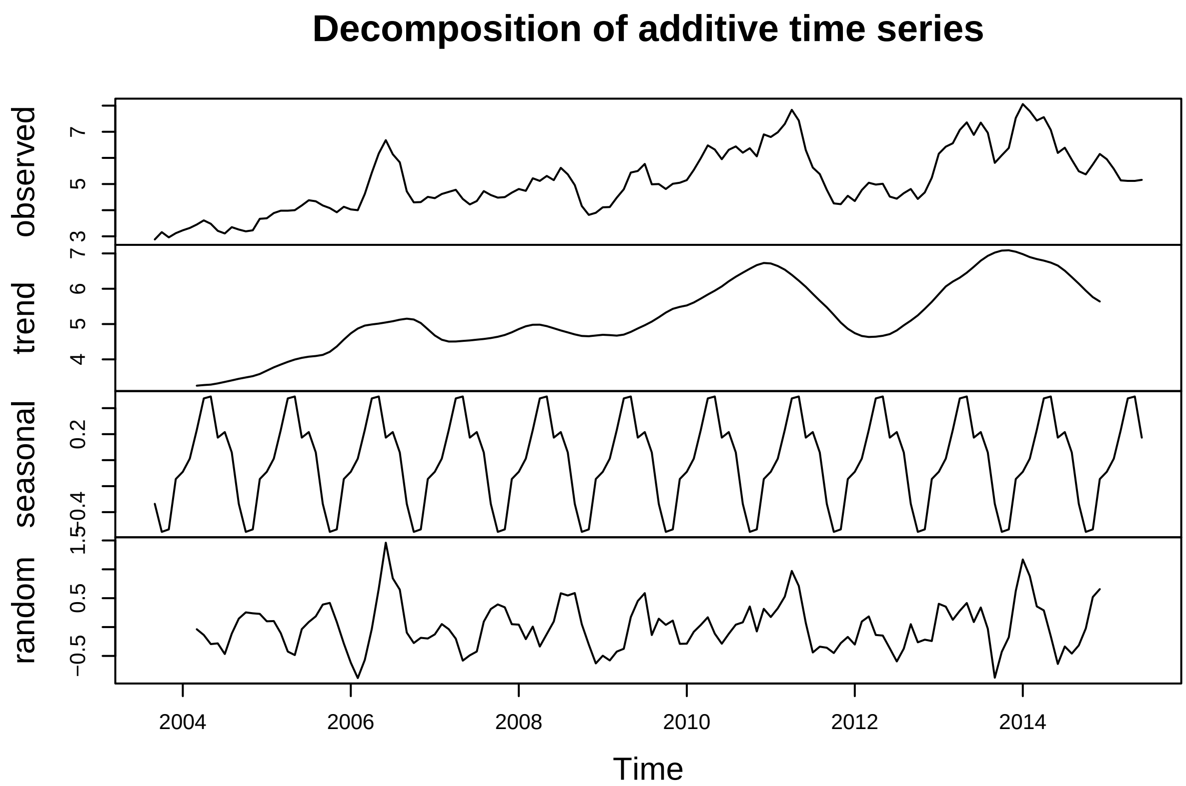

```r
plot(salmon_train, xlab = "Year", ylab = "price per kg (US dollars)")
```



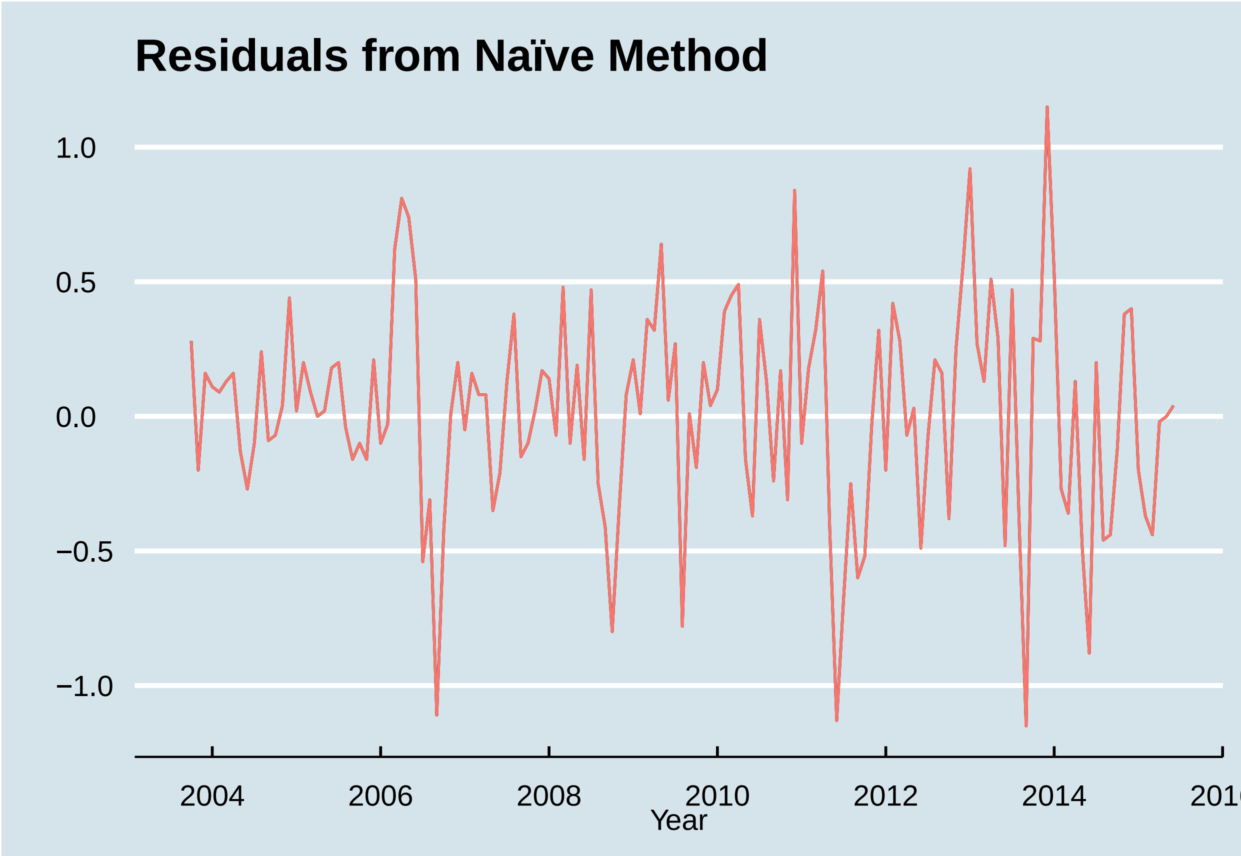

```r
salmon_ts_log <- log(salmon_train)
plot(salmon_ts_log, xlab = "Year", ylab = "log(price per kg (US dollars))")
```



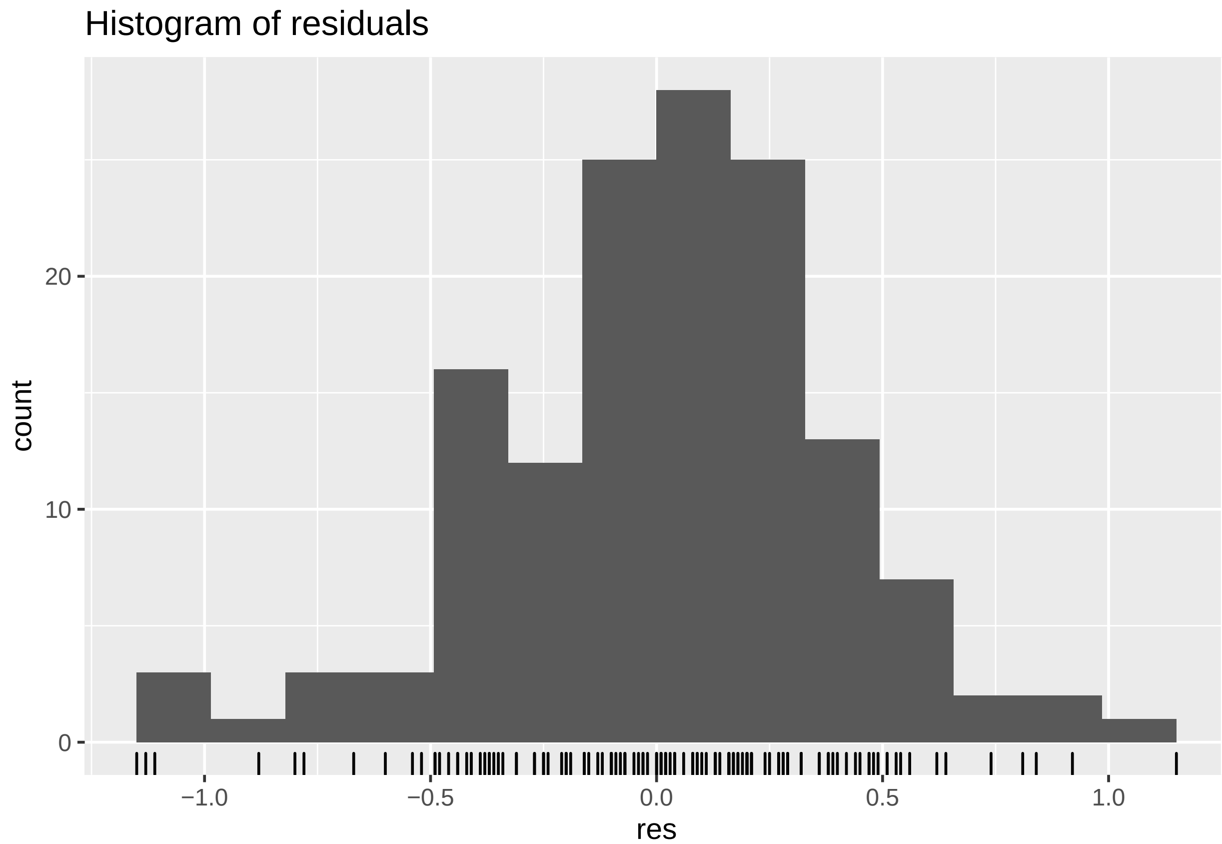

```r
salmon_ts_components <- decompose(salmon_train)
plot(salmon_ts_components)
```



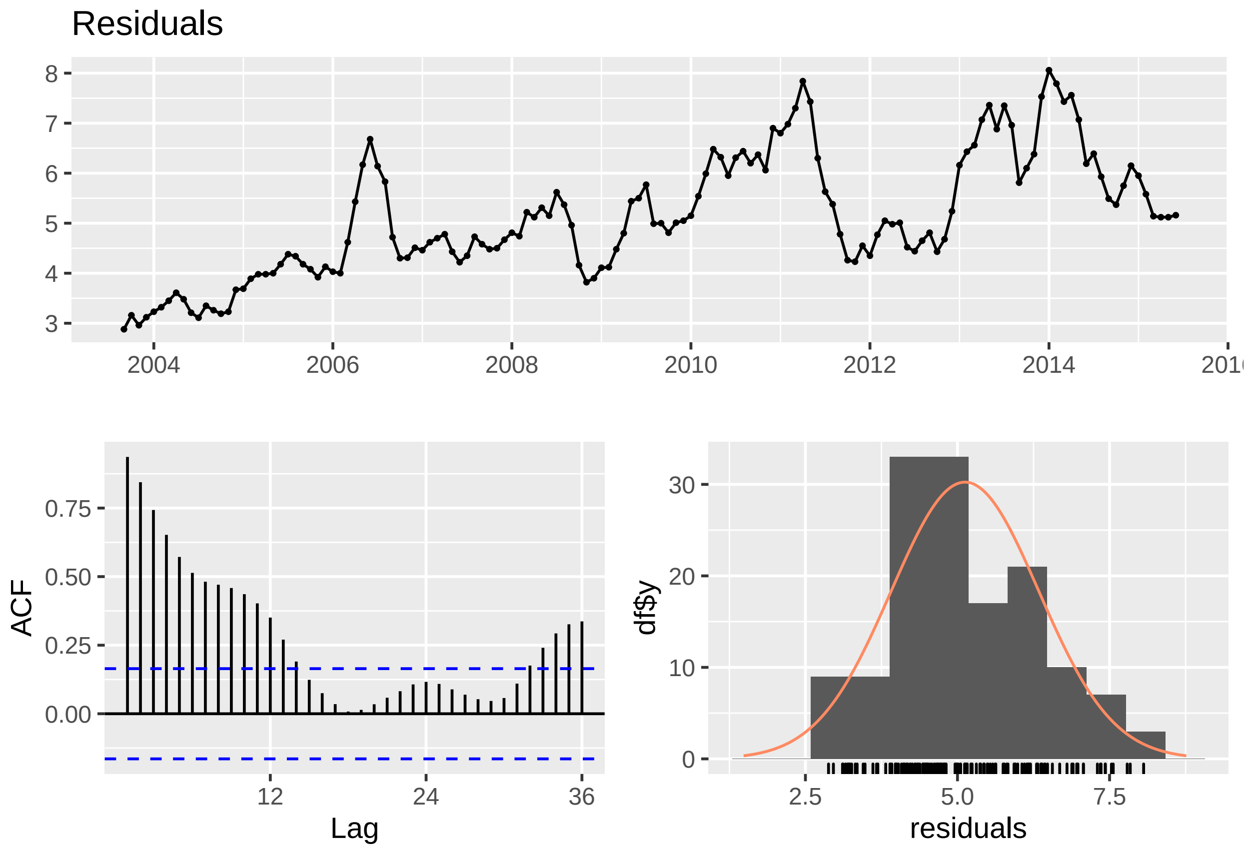

```
res <- residuals(naive(salmon_train))
autoplot(res) + xlab("Year") + ylab("") +
  ggtitle("Residuals from Naïve Method")+ geom_line( color="#F8766D")+theme_economist(base_size = 8)
```




```
gghistogram(res) + ggtitle("Histogram of residuals")
```



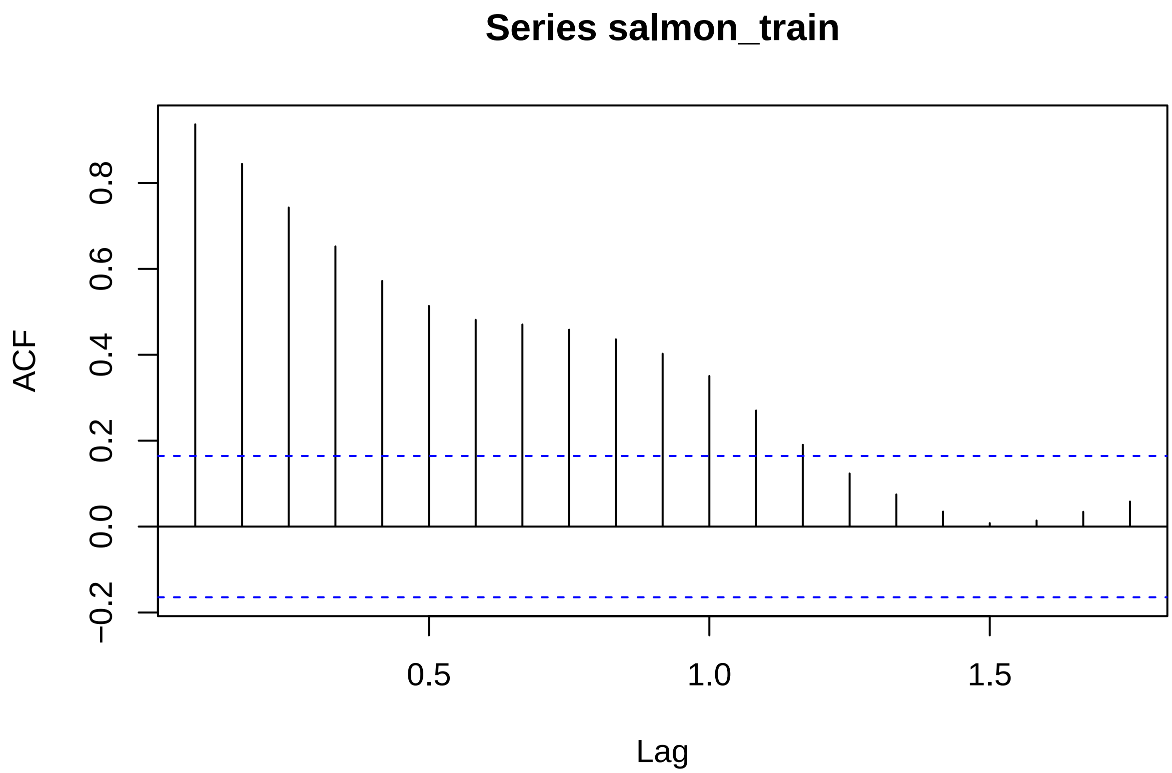

```
checkresiduals(salmon_train, lag, df = NULL, plot = TRUE)
```



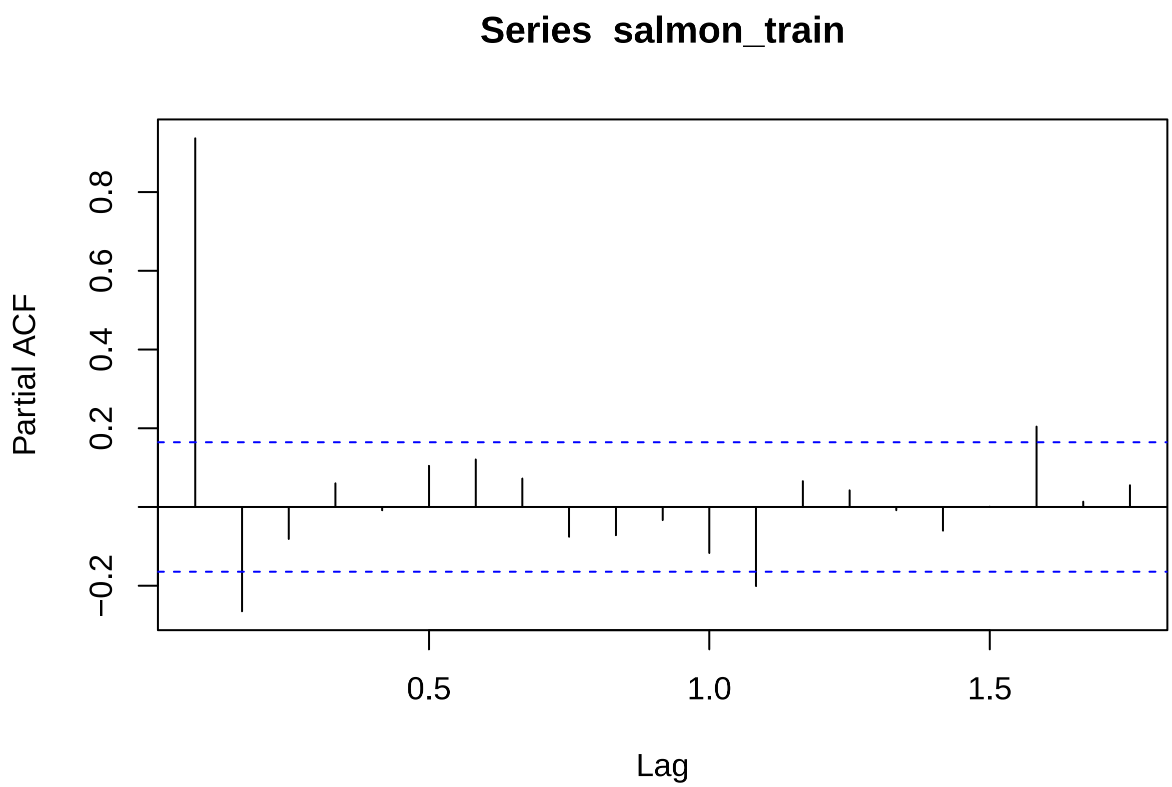
The residuals from the naive method show stationarity though with an increase in variance as time progresses.

2. Testing ARMA models and showing ACF and PACF plots to determine which ARMA models may be appropriate.

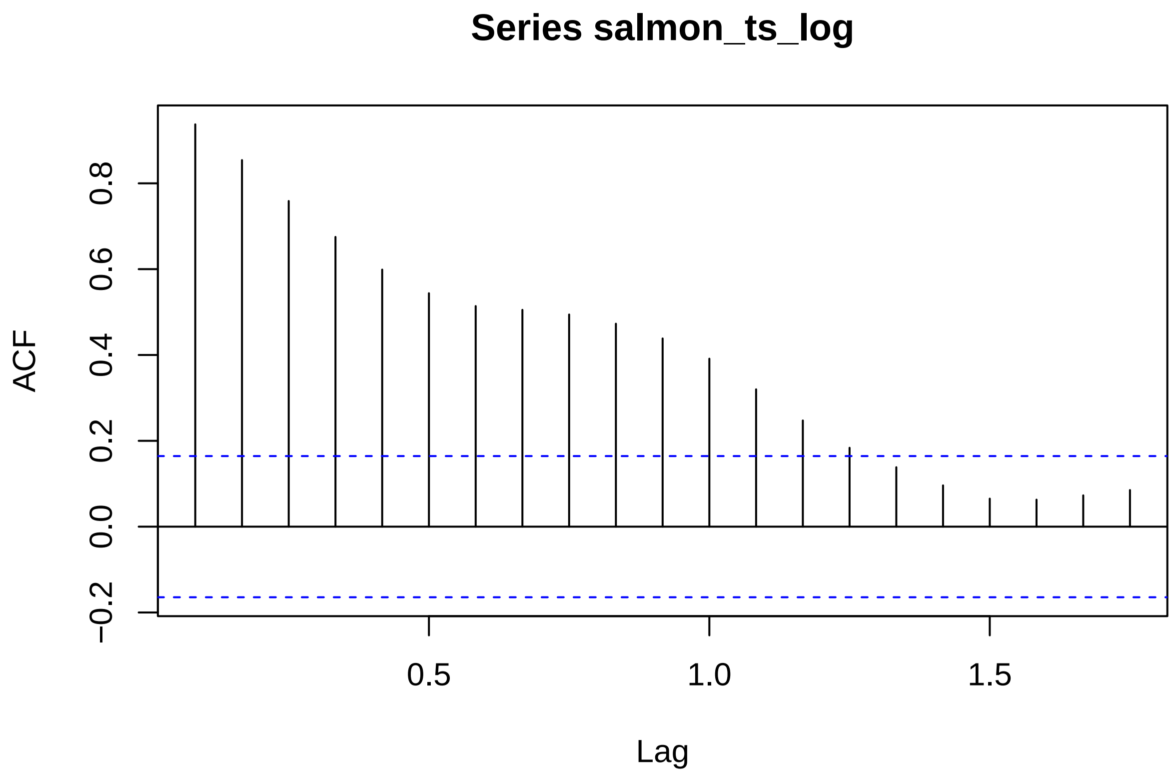
```
#acf and pacf plots
acf(salmon_train)
```



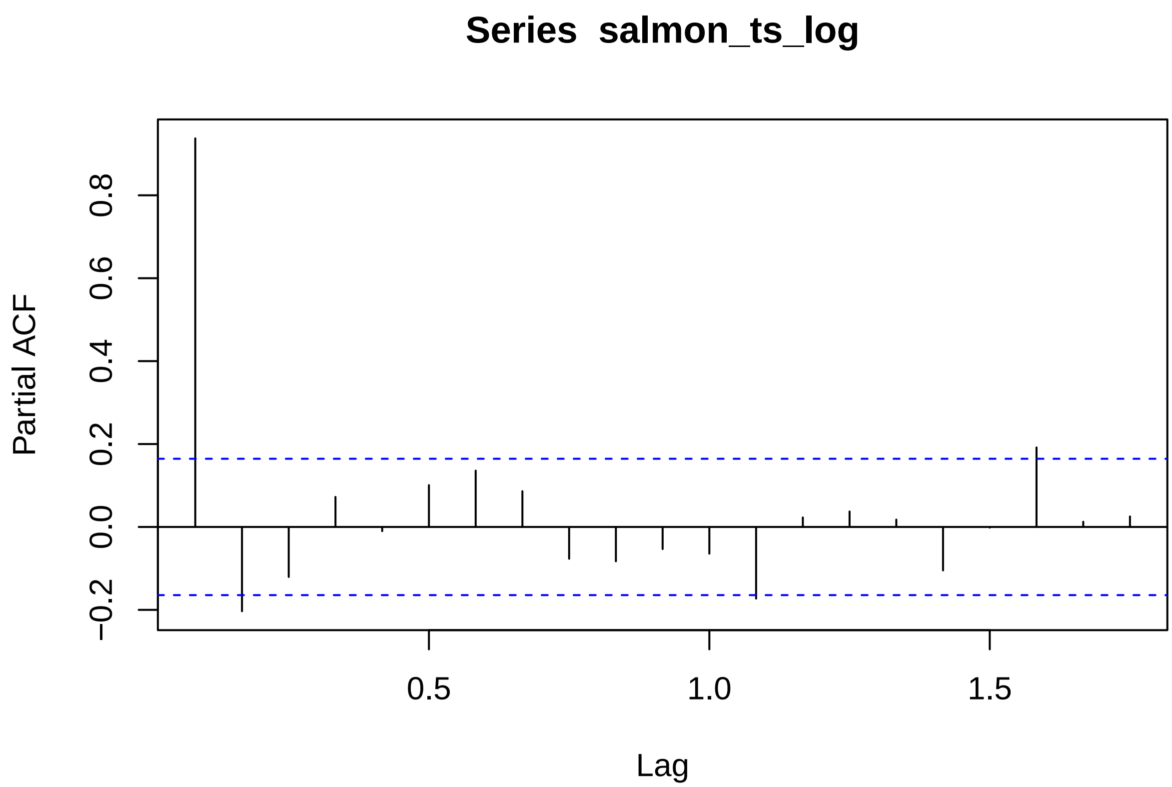

```
pacf(salmon_train)
```




```r
#log acf and pacf plots
acf(salmon_ts_log)
```




```
pacf(salmon_ts_log)
```

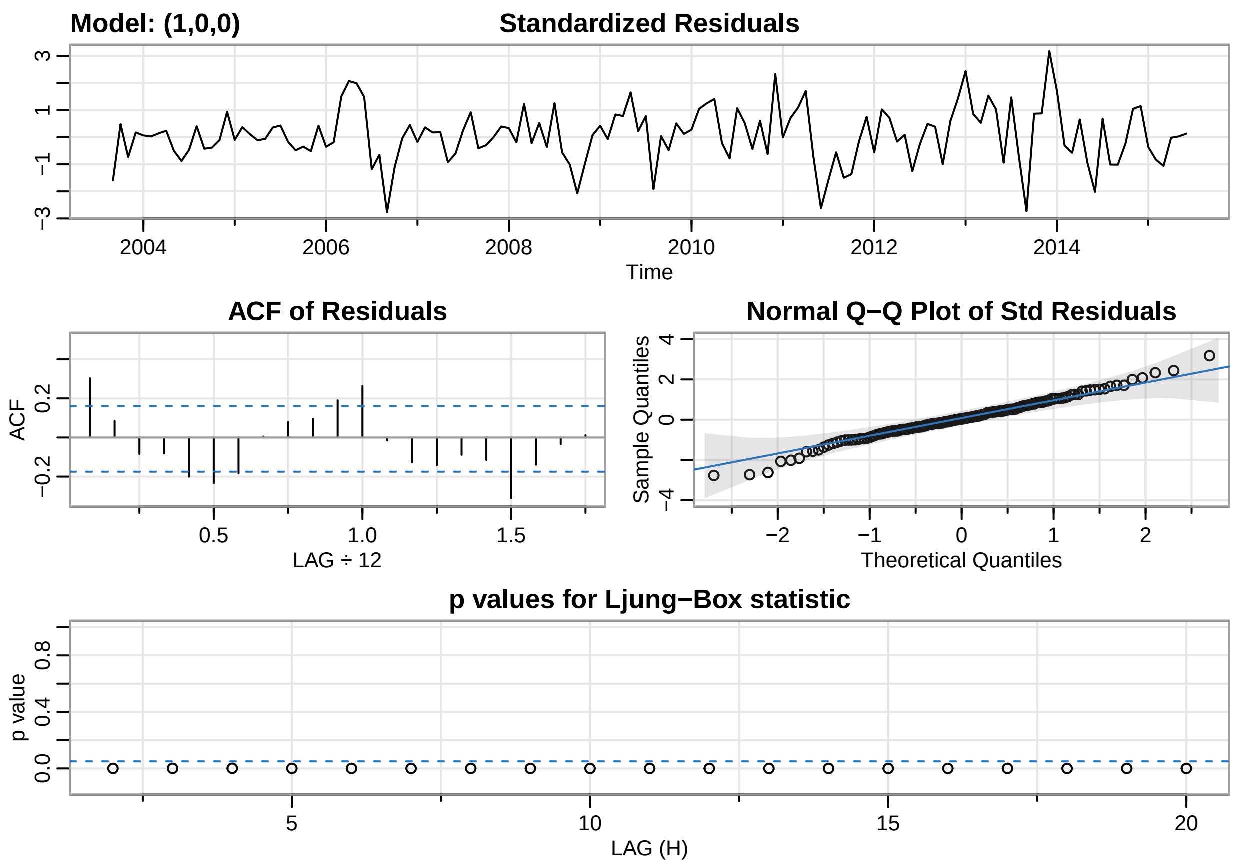


The ACF plots show geometric decay as they move slowly toward zero. The PACF plots show a significant lag at lag 1. An appropriate ARMA model may be to use an AR(1) model.

```
sarima(salmon_train, 1, 0, 0)
```

```
## initial  value 0.180705
## iter   2 value -0.961817
## iter   3 value -0.963532
## iter   4 value -0.964629
## iter   5 value -0.964675
## iter   6 value -0.965392
## iter   7 value -0.965392
## iter   8 value -0.965393
## iter   9 value -0.965394
## iter  10 value -0.965396
## iter  11 value -0.965398
## iter  12 value -0.965399
## iter  13 value -0.965399
## iter  14 value -0.965399
## iter  15 value -0.965400
## iter  16 value -0.965400
## iter  16 value -0.965400
## final  value -0.965400
## converged
## initial  value -0.943154
## iter   2 value -0.944016
## iter   3 value -0.946031
```

```
## iter   4 value -0.946334
## iter   5 value -0.946465
## iter   6 value -0.946577
## iter   7 value -0.946636
## iter   8 value -0.947037
## iter   9 value -0.947287
## iter  10 value -0.947337
## iter  11 value -0.947399
## iter  12 value -0.947428
## iter  13 value -0.947583
## iter  14 value -0.947594
## iter  15 value -0.947612
## iter  16 value -0.947641
## iter  17 value -0.947649
## iter  18 value -0.947692
## iter  19 value -0.947693
## iter  20 value -0.947695
## iter  21 value -0.947700
## iter  22 value -0.947701
## iter  23 value -0.947706
## iter  24 value -0.947707
## iter  25 value -0.947710
## iter  25 value -0.947710
## iter  25 value -0.947710
## final  value -0.947710
## converged
```

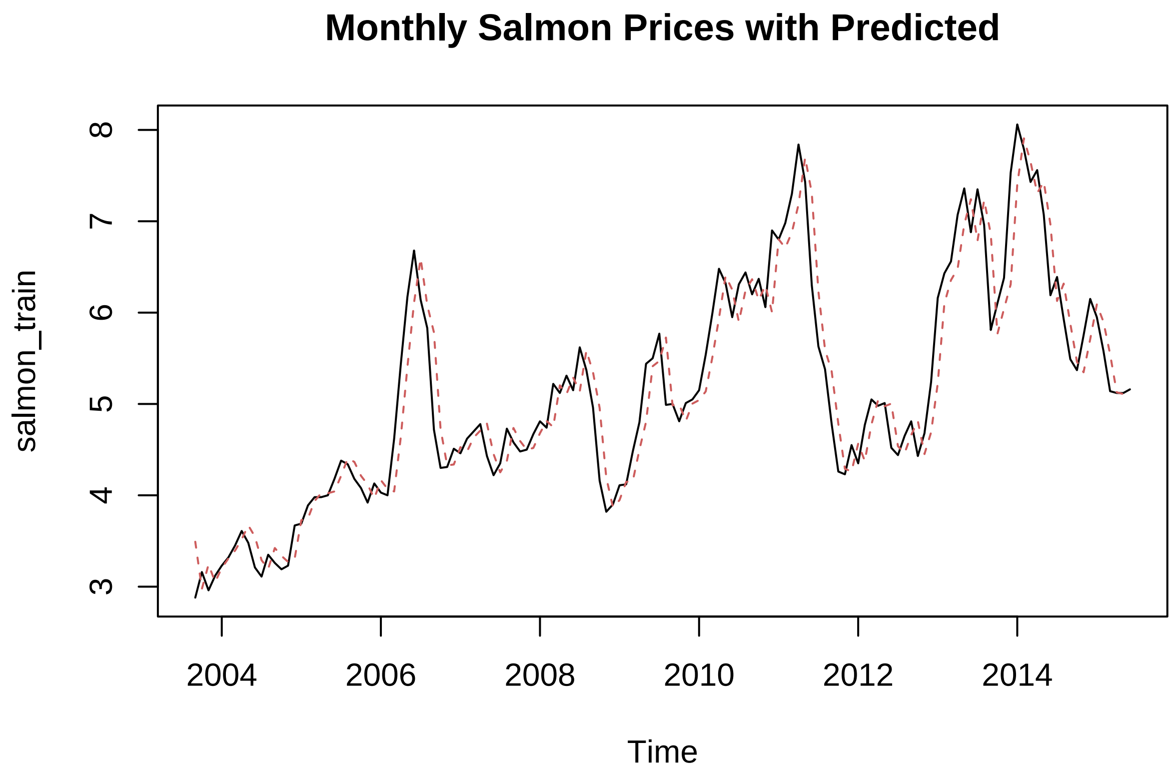


```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1   xmean
##       0.9521  4.8815
## s.e.  0.0250  0.6047
##
## sigma^2 estimated as 0.1478:  log likelihood = -66.91,  aic = 139.83
##
## $degrees_of_freedom
## [1] 140
##
## $ttable
##       Estimate     SE t.value p.value
## ar1     0.9521 0.0250 38.1319       0
## xmean   4.8815 0.6047  8.0726       0
##
## $AIC
## [1] 0.9847107
##
## $AICc
## [1] 0.9853187
##
## $BIC
## [1] 1.047158
```

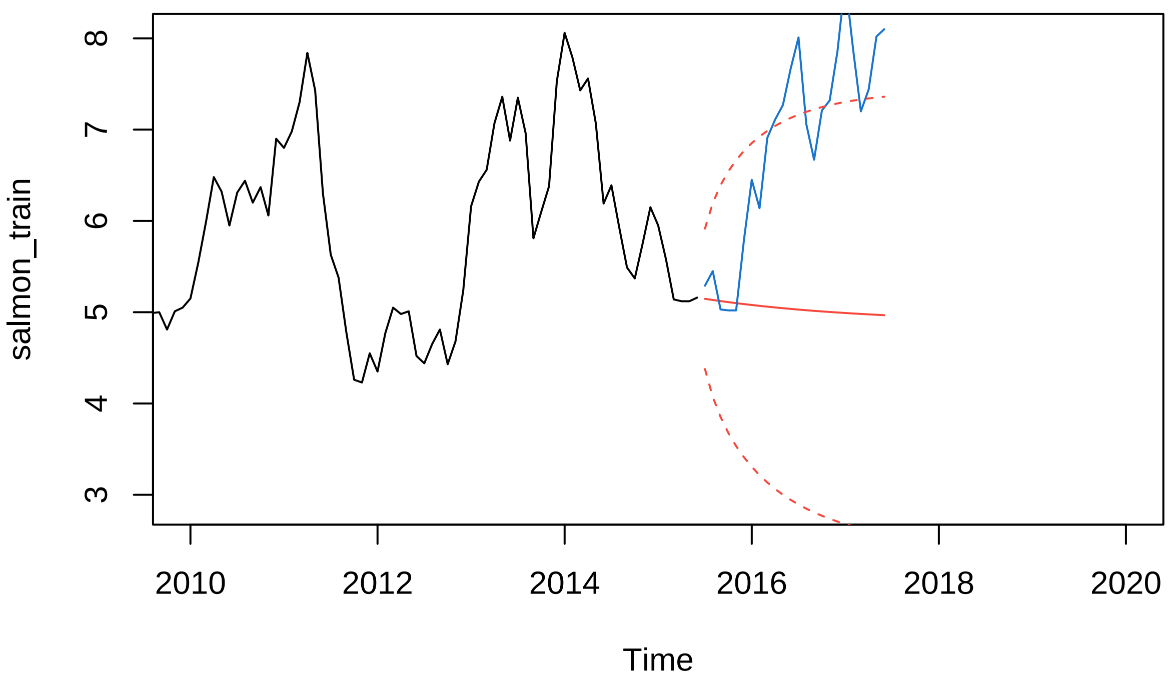
```
AR1 <- arima(salmon_train, order = c(1,0,0))
print(AR1)
```

```
##
## Call:
## arima(x = salmon_train, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.9521     4.8815
## s.e.  0.0250     0.6047
##
## sigma^2 estimated as 0.1478:  log likelihood = -66.91,  aic = 137.83
```

```
ts.plot(salmon_train, main = 'Monthly Salmon Prices with Predicted')
AR_fit = salmon_train - residuals(AR1)
points(AR_fit, type = 'l', col = "indianred", lty = 2)
```

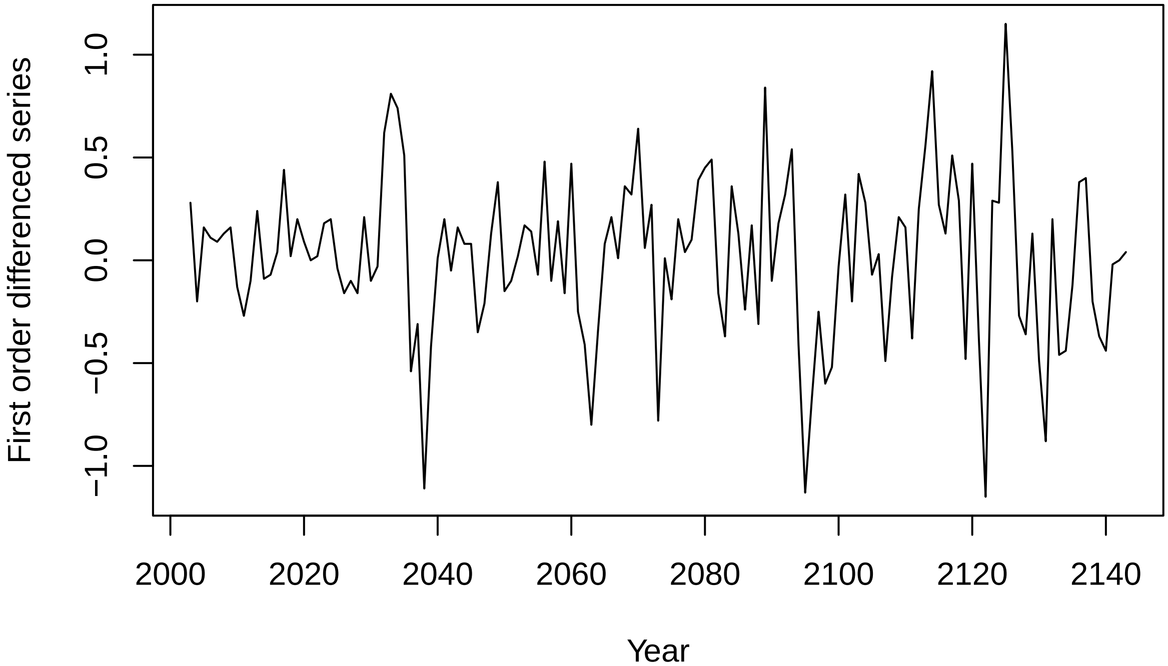



```
period_predict <- 24

ts.plot(salmon_train, xlim = c(2010,2020))
AR_forecast <- predict(AR1, n.ahead = period_predict)$pred
AR_forecast_se <- predict(AR1, n.ahead = period_predict)$se
points(AR_forecast, type = 'l', col = 2)
points(salmon_test, type = 'l', col = 4)
points(AR_forecast - 2*AR_forecast_se, type = 'l', col = 2, lty = 2)
points(AR_forecast + 2*AR_forecast_se, type = 'l', col = 2, lty = 2)
```

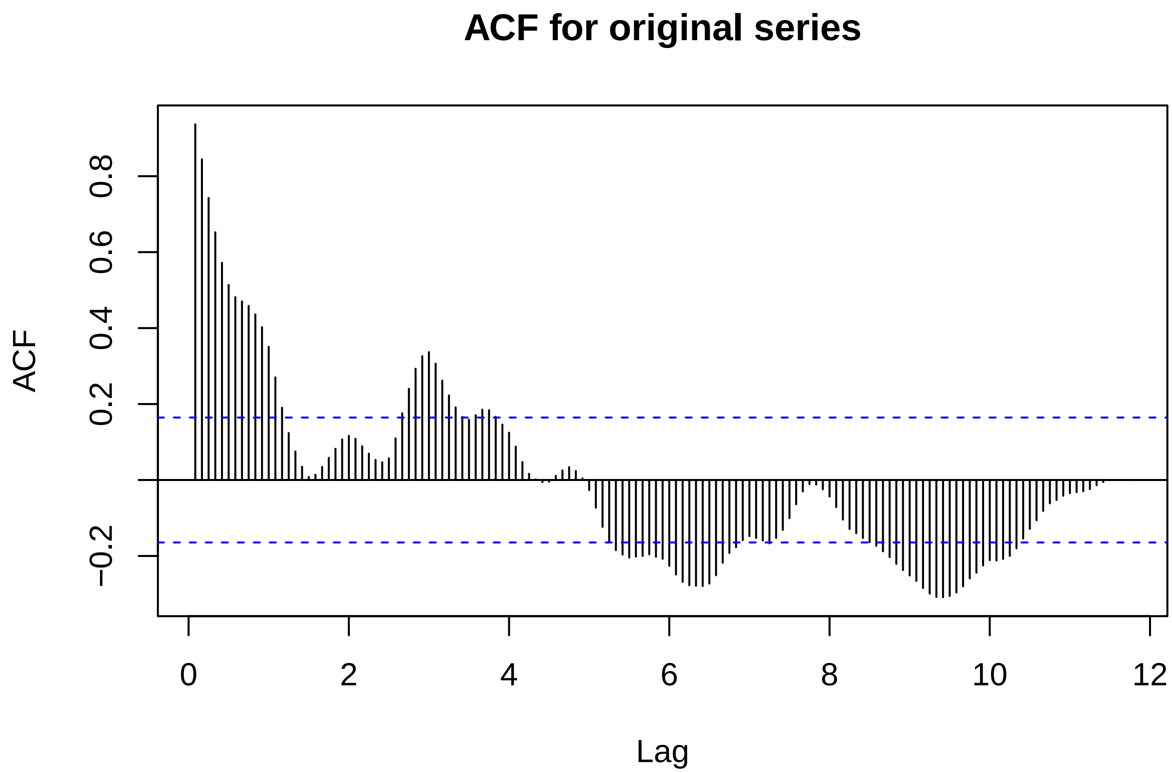


3. Testing various ARIMA models and comparing AIC and examining residuals/doing diagnostics

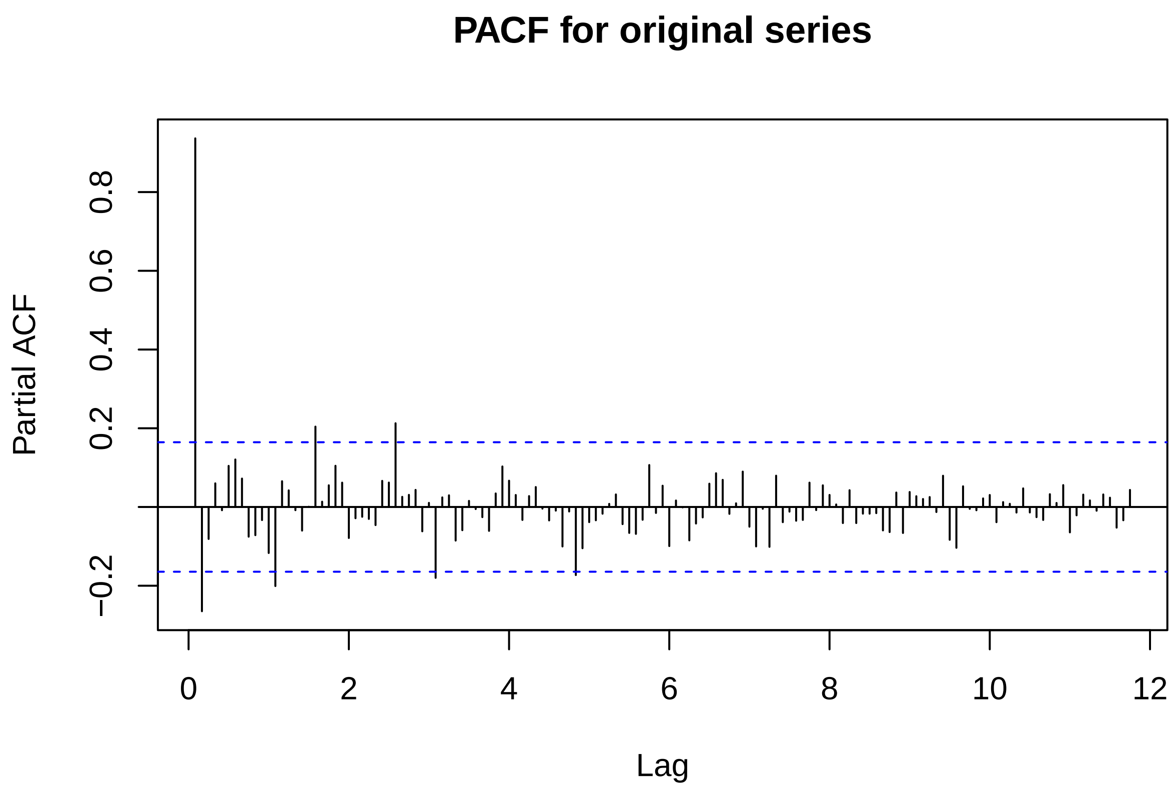
```
period_predict <- 24

#salmon_train_sub <- subset(salmon_train, salmon_train[["Year"]] > 2003)
#first difference of original data
diff_salmon <- diff(salmon_train)
diff_salmon <- ts(diff_salmon, start = c(2003), deltat = 1)
plot(diff_salmon, xlab = "Year", ylab = "First order differenced series")
```



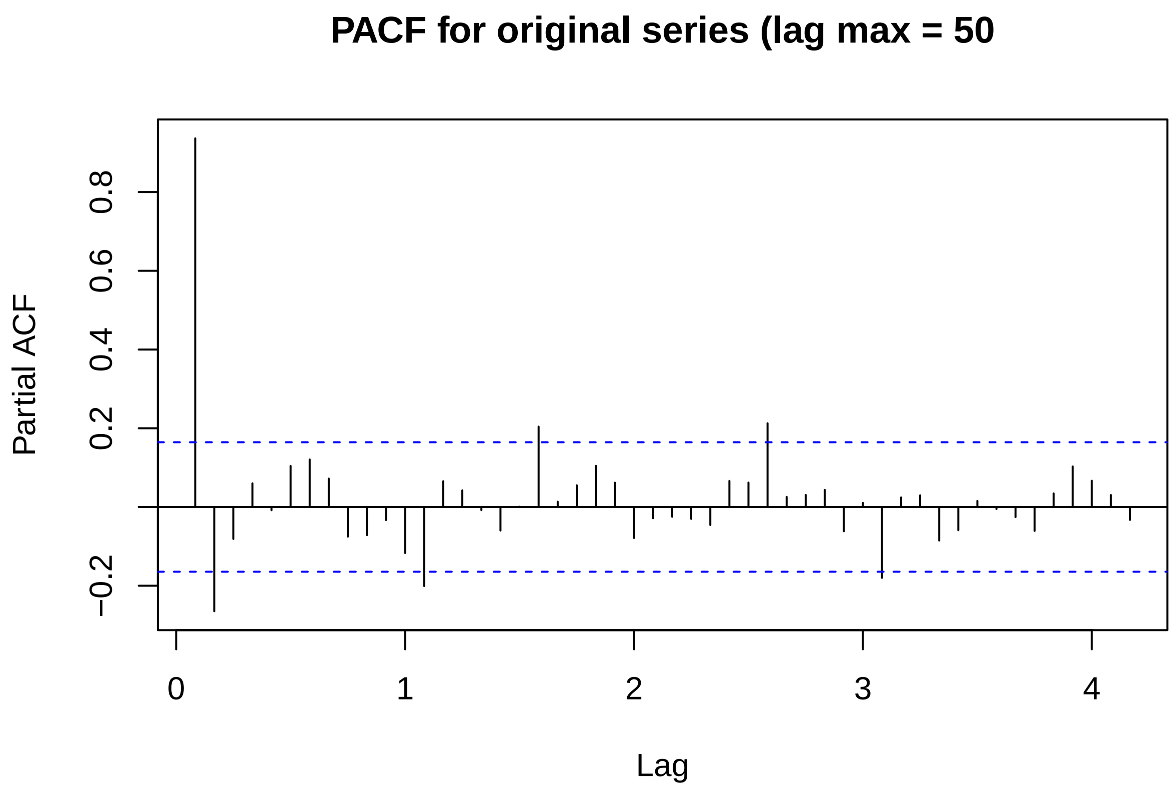

```
#sample ACF and PACF plots of the original series
acf(salmon_train, lag.max = 360, na.action = na.pass,
main = "ACF for original series")
```



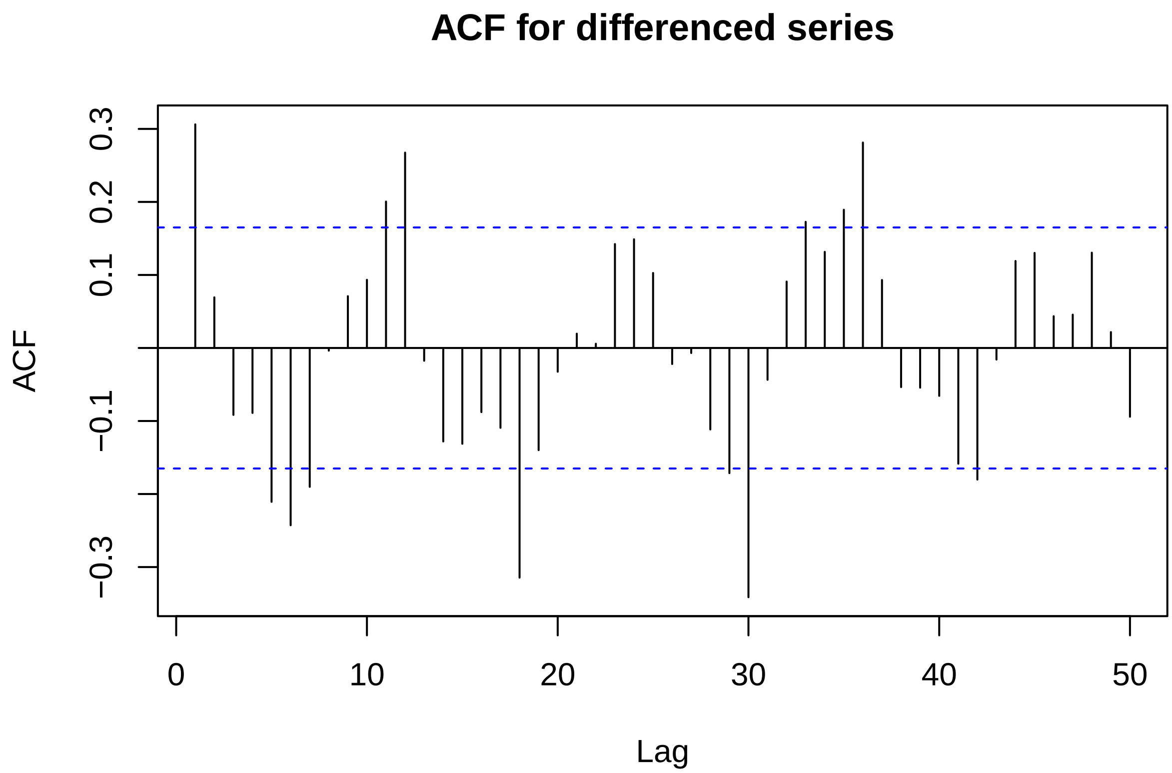

```
pacf(salmon_train, lag.max = 360, na.action = na.pass,
main = "PACF for original series")
```



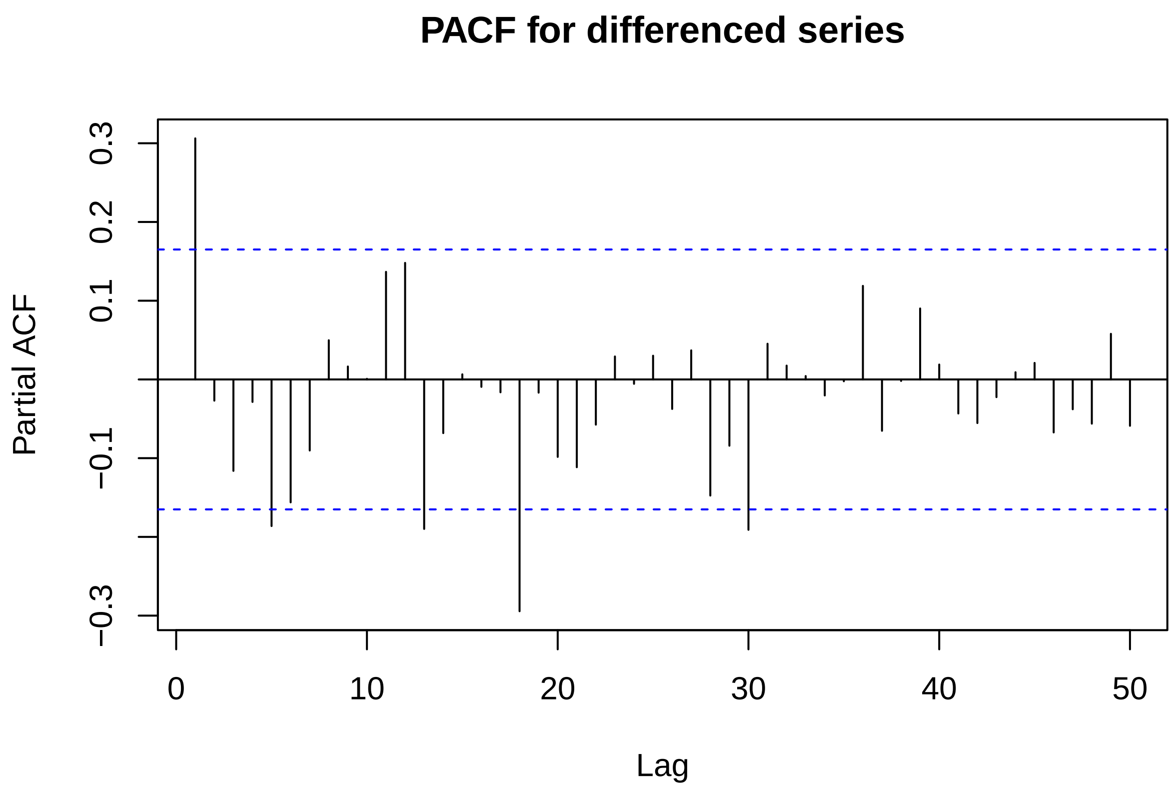

```
#max lag at 50 to discern where the p significant lag is
pacf(salmon_train, lag.max = 50, na.action = na.pass,
main = "PACF for original series (lag max = 50")
```




```
#sample ACF and PACF plots of the first difference original series
acf(diff_salmon, lag.max = 50, na.action = na.pass,
main = "ACF for differenced series")
```




```
pacf(diff_salmon, lag.max = 50, na.action = na.pass,
main = "PACF for differenced series")
```




```r
#various Arima models
n <- length(diff_salmon)

(fit.2_1 <- arima(salmon_train, order=c(2,1,1)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(2, 1, 1))
##
## Coefficients:
##           ar1     ar2     ma1
##       -0.5124  0.2876  0.8096
## s.e.   0.2074  0.0903  0.1979
##
## sigma^2 estimated as 0.1362:  log likelihood = -59.59,  aic = 125.17
```

```r
(fit.1_2 <- arima(salmon_train, order=c(1,1,2)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(1, 1, 2))
##
## Coefficients:
##           ar1     ma1     ma2
##       -0.5414  0.8598  0.2760
## s.e.   0.4471  0.4347  0.1191
##
```

```
## sigma^2 estimated as 0.1352:  log likelihood = -59.07,  aic = 124.15
```

```r
(fit.2_2 <- arima(salmon_train, order=c(2,1,2)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(2, 1, 2))
##
## Coefficients:
##           ar1     ar2     ma1     ma2
##       -0.5462  0.0143  0.8652  0.2657
## s.e.   0.4748  0.3313  0.4679  0.2534
##
## sigma^2 estimated as 0.1352:  log likelihood = -59.07,  aic = 126.15
```

```r
(fit.3_2 <- arima(salmon_train, order=c(3,1,2)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(3, 1, 2))
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2
##       0.4515  0.6498  -0.3357  -0.1797  -0.7164
## s.e.  0.2455  0.2755   0.0981   0.2468   0.2363
##
## sigma^2 estimated as 0.1293:  log likelihood = -56.12,  aic = 122.23
```

```r
(fit.2_3 <- arima(salmon_train, order=c(2,1,3)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(2, 1, 3))
##
## Coefficients:
##          ar1     ar2      ma1      ma2      ma3
##       0.7799  0.0353  -0.5054  -0.2165  -0.1976
## s.e.  0.9367  0.8436   0.9329   0.6220   0.2851
##
## sigma^2 estimated as 0.1302:  log likelihood = -56.62,  aic = 123.23
```

```r
(fit.3_3 <- arima(salmon_train, order=c(3,1,3)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(3, 1, 3))
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2      ma3
##       0.4466  0.6450  -0.3264  -0.1736  -0.7120  -0.0105
## s.e.  0.2916  0.2955   0.2329   0.3076   0.2553   0.2443
##
## sigma^2 estimated as 0.1293:  log likelihood = -56.11,  aic = 124.23
```

```r
(fit.4_3 <- arima(salmon_train, order=c(4,1,3)))
```

```
##
## Call:
```

```
## arima(x = salmon_train, order = c(4, 1, 3))
##
## Coefficients:
##          ar1     ar2      ar3     ar4      ma1      ma2     ma3
##       0.7059  0.5389  -0.5043  0.0843  -0.4348  -0.6736  0.1862
## s.e.  3.0337  1.4227   1.9637  1.0235   3.0313   0.6298  2.1573
##
## sigma^2 estimated as 0.1293:  log likelihood = -56.12,  aic = 126.23
```

```r
(fit.3_4 <- arima(salmon_train, order=c(3,1,4)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(3, 1, 4))
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2     ma3     ma4
##       0.5547  0.6154  -0.3777  -0.2805  -0.7177  0.0471  0.0435
## s.e.  0.4556  0.4075   0.2568   0.4601   0.3357  0.2573  0.1395
##
## sigma^2 estimated as 0.1292:  log likelihood = -56.07,  aic = 126.14
```

```r
(fit.4_4 <- arima(salmon_train, order=c(4,1,4)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(4, 1, 4))
##
## Coefficients:
##          ar1     ar2      ar3      ar4      ma1      ma2      ma3     ma4
##       0.5199  0.4199  -0.0828  -0.6444  -0.3879  -0.5493  -0.0567  0.8099
## s.e.  0.1998  0.2581   0.2099   0.1686   0.1619   0.1988   0.1557  0.1339
##
## sigma^2 estimated as 0.1149:  log likelihood = -50.38,  aic = 116.76
```

```r
(fit.5_4 <- arima(salmon_train, order=c(5,1,4)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(5, 1, 4))
##
## Coefficients:
##          ar1     ar2      ar3      ar4     ar5      ma1      ma2      ma3
##       0.3834  0.6688  -0.0856  -0.8956  0.1553  -0.1850  -0.7480  -0.1488
## s.e.  0.0945  0.0867   0.1321   0.1041  0.0908   0.0681   0.0521   0.1057
##          ma4
##       0.9792
## s.e.  0.0938
##
## sigma^2 estimated as 0.1122:  log likelihood = -49.08,  aic = 116.15
```

```r
(fit.4_5 <- arima(salmon_train, order=c(4,1,5)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(4, 1, 5))
##
```

```
## Coefficients:
##           ar1     ar2     ar3      ar4     ma1      ma2      ma3     ma4
##       -0.1758  0.6628  0.5121  -0.4204  0.4846  -0.5614  -0.8543  0.1377
## s.e.   0.3106  0.2198  0.1573   0.2485  0.3448   0.3109   0.2071  0.3532
##           ma5
##       -0.0136
## s.e.   0.1354
##
## sigma^2 estimated as 0.1233:  log likelihood = -54.32,  aic = 126.64
```

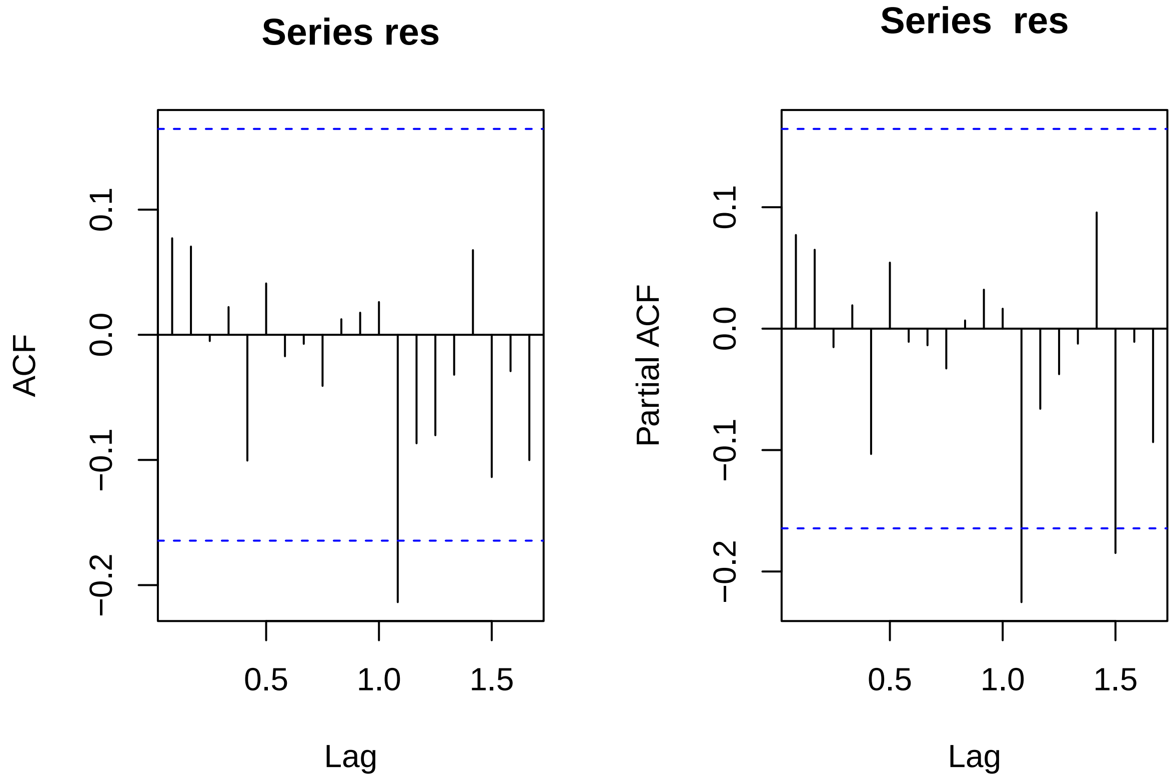
```
(fit.5_5 <- arima(salmon_train, order=c(5,1,5)))
```

```
##
## Call:
## arima(x = salmon_train, order = c(5, 1, 5))
##
## Coefficients:
##          ar1      ar2      ar3     ar4      ar5      ma1     ma2     ma3
##       0.6709  -0.2113  -0.1947  0.5358  -0.8995  -0.5423  0.1123  0.1123
## s.e.  0.0449   0.0644   0.0611  0.0651   0.0443   0.0478  0.0498  0.0488
##           ma4     ma5
##       -0.5423  1.0000
## s.e.   0.0471  0.0568
##
## sigma^2 estimated as 0.1043:  log likelihood = -46.39,  aic = 112.78
```

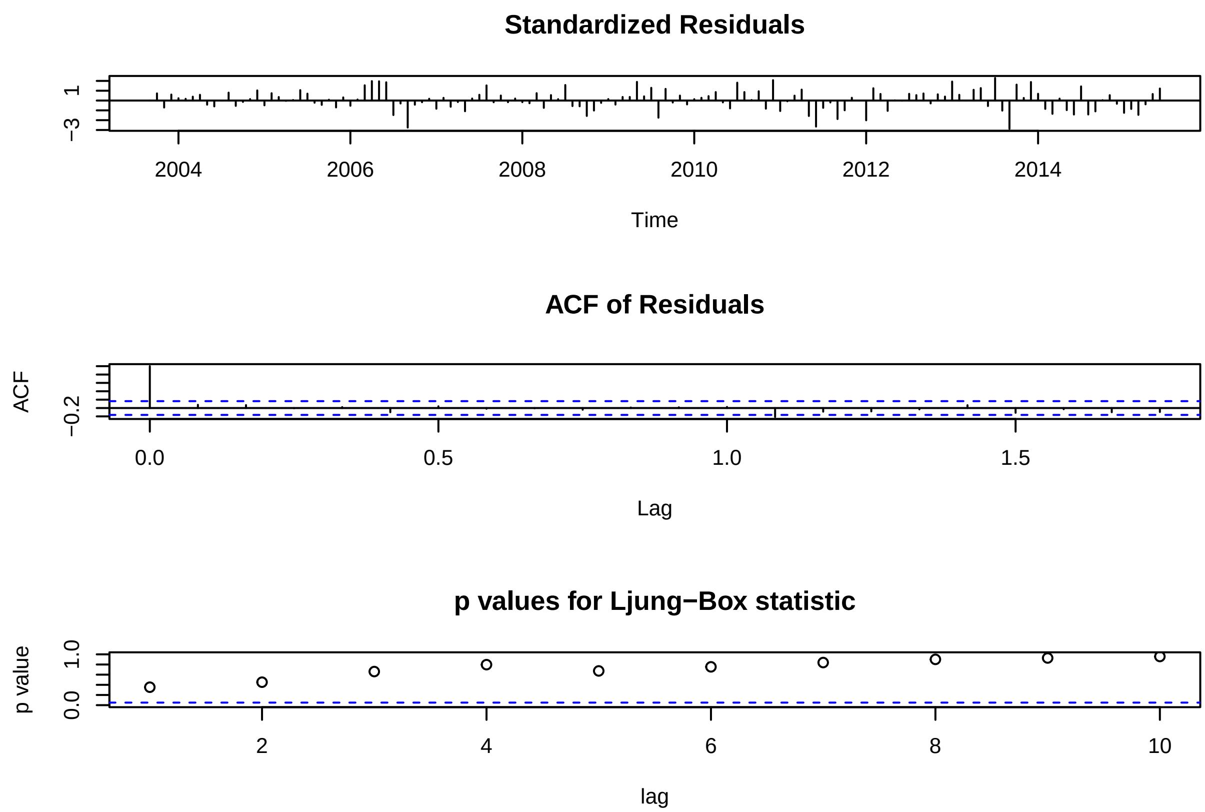
```
par(mfrow = c(1, 2))
res <- fit.5_5$residuals
acf(res, lag.max = 20)+ geom_line( color="#F8766D")+theme_economist(base_size = 8)
```

```
## NULL
```

```
pacf(res, lag.max = 20)
```



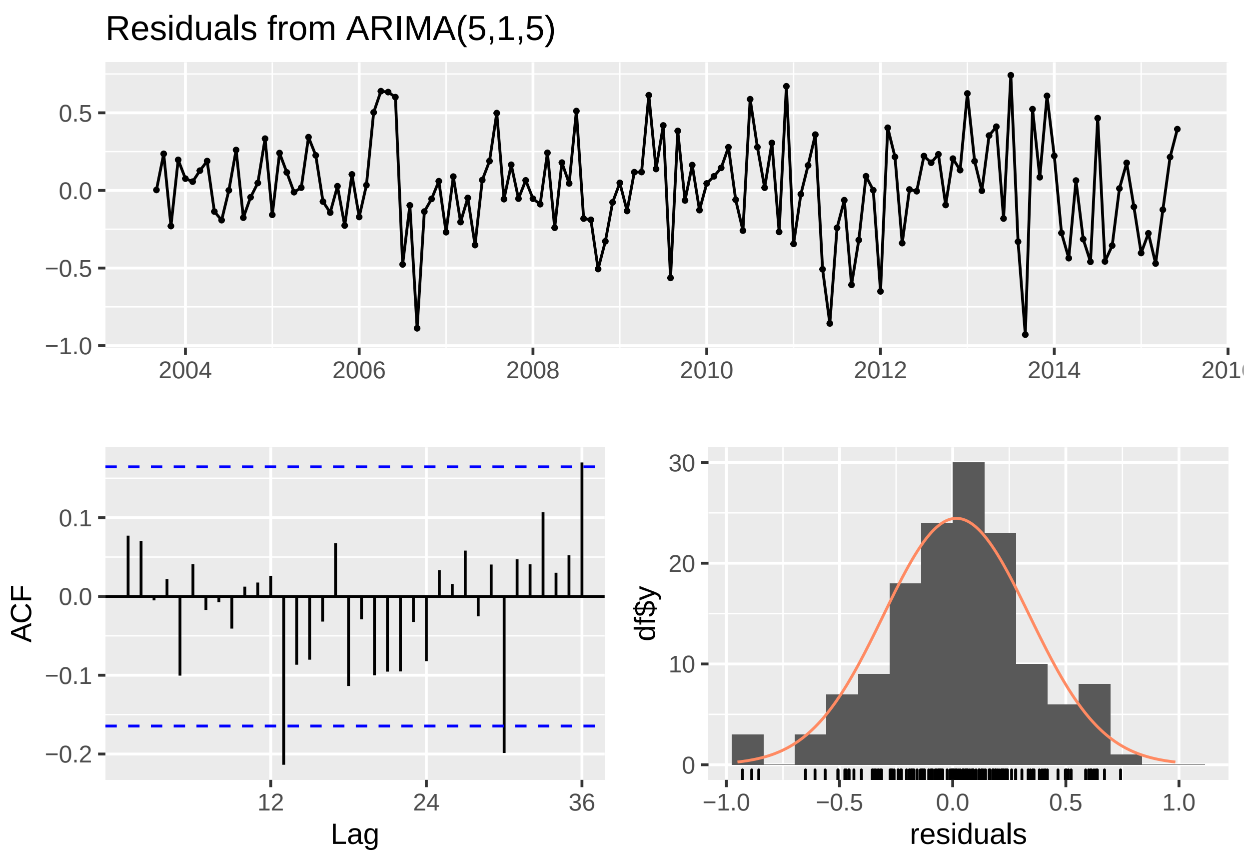

```
tsdiag(fit.5_5)
```



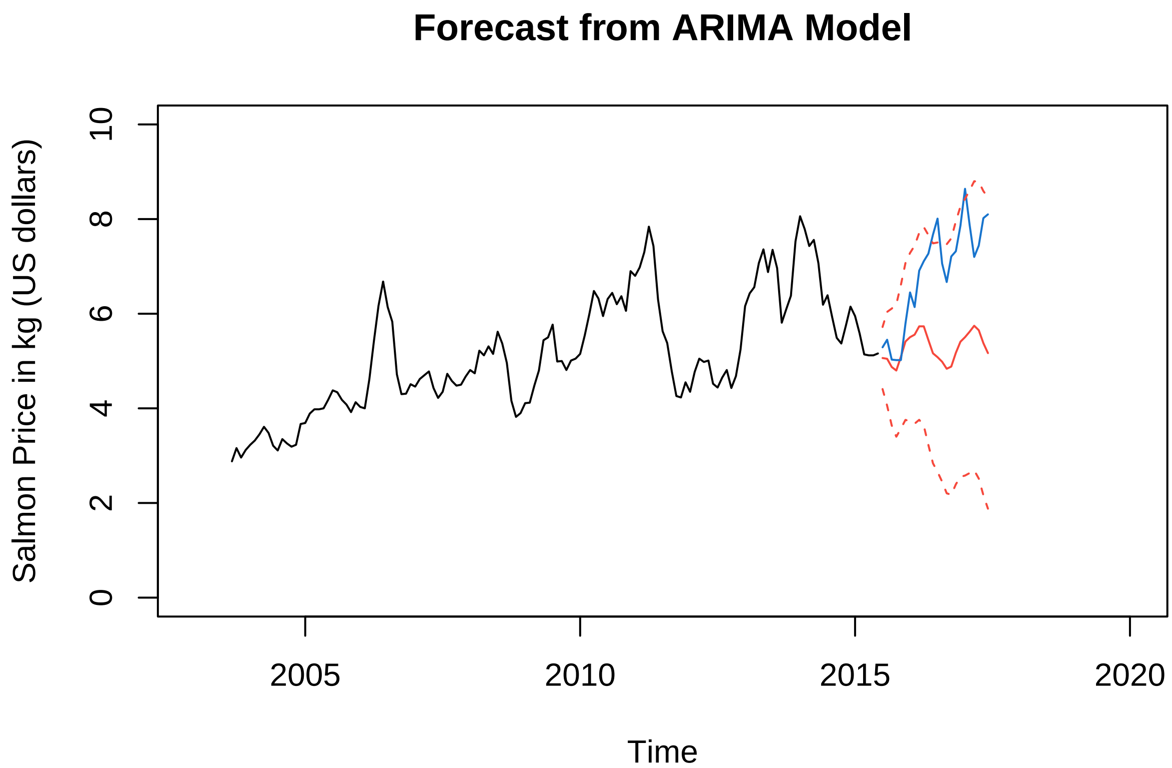

```
qqnorm(res)
qqline(res)
checkresiduals(fit.5_5)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(5,1,5)
## Q* = 22.68, df = 14, p-value = 0.06569
##
## Model df: 10.   Total lags used: 24
```

```
#autoplot(forecast(fit.5_5))
```




```r
pred <- predict(fit.5_5, n.ahead = 24)

ts.plot(salmon_train, xlim = c(2003, 2020), ylim = c(0,10),
ylab = "Salmon Price in kg (US dollars)", main = "Forecast from ARIMA Model")
ARIMA_forecast <- predict(fit.5_5, n.ahead = period_predict)$pred
ARIMA_forecast_se <- predict(fit.5_5, n.ahead = period_predict)$se
points(ARIMA_forecast, type = 'l', col = 2)
points(salmon_test, type = 'l', col = 4)
points(ARIMA_forecast - 2*ARIMA_forecast_se, type = 'l', col = 2, lty = 2)
points(ARIMA_forecast + 2*ARIMA_forecast_se, type = 'l', col = 2, lty = 2)
```

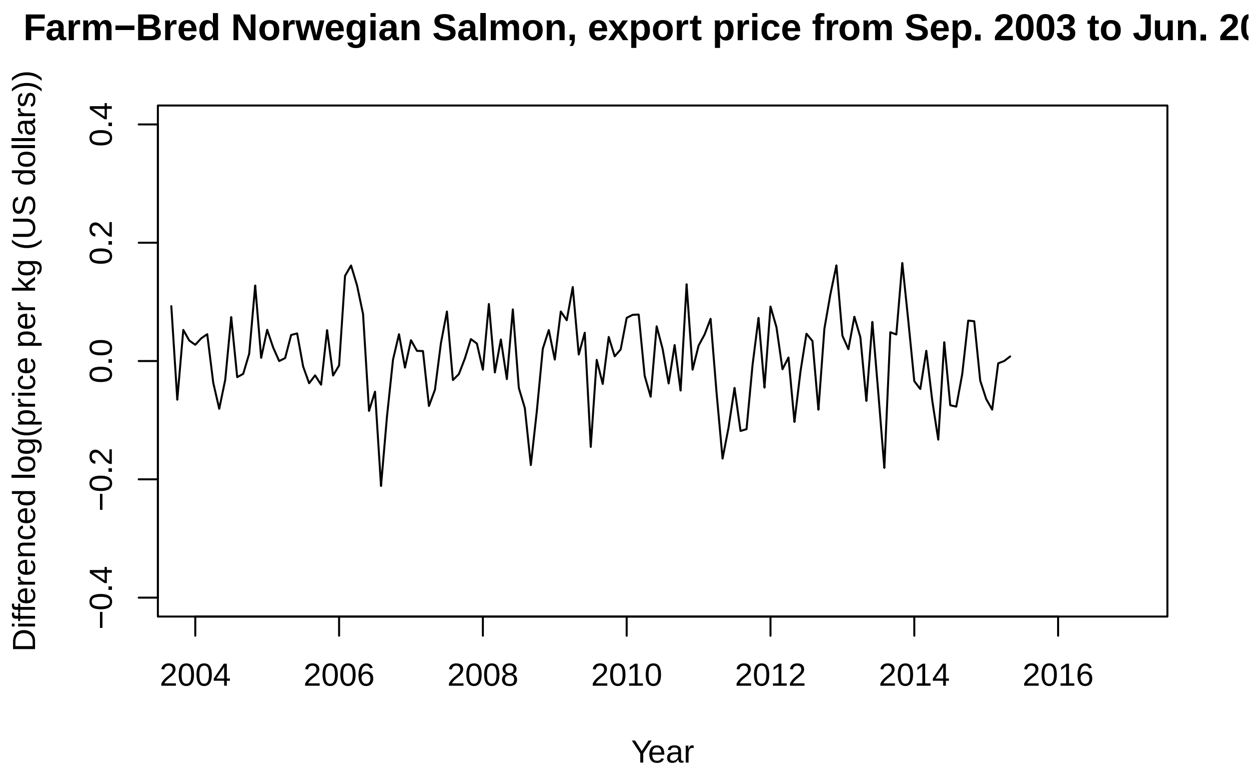


3. Testing various SARIMA models and comparing AIC and examining residuals/doing diagnostics.

Although the best-fit ARIMA model above does capture the main patters in predicted series, it might me quite challenging for interpretation due to high order parameters (p=5, q=5). Hence we try applying SARIMA method to see if capturing the trend and seasonality patterns can provide a simpler model having a comparable accuracy.

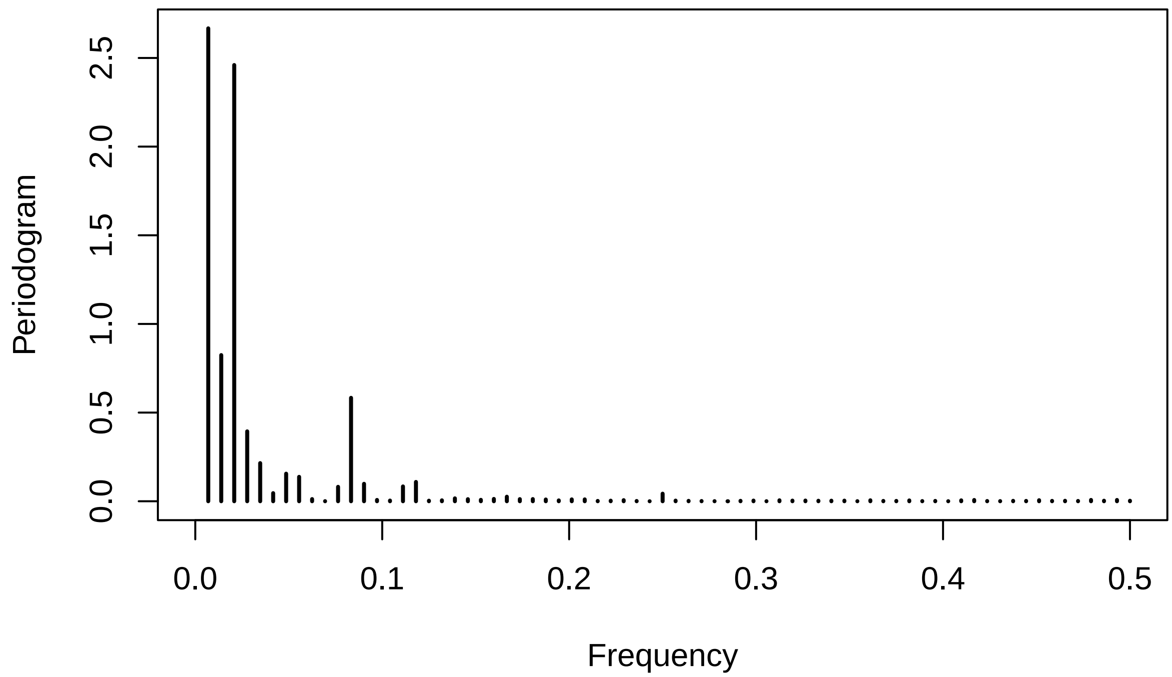
First, let's difference the trend.

```
salmon_ts_log.diff<-diff(salmon_ts_log, lag = 1)
salmon_ts_log.diff <- ts(salmon_ts_log.diff, start = c(2003,9), deltat = 1/12)
plot(salmon_ts_log.diff, xlim = c(2004, 2017), ylim = c(-0.4, 0.4), xlab = "Year", ylab = "Differenced 
main  =  "Farm-Bred Norwegian Salmon, export price from Sep. 2003 to Jun. 2015")
```



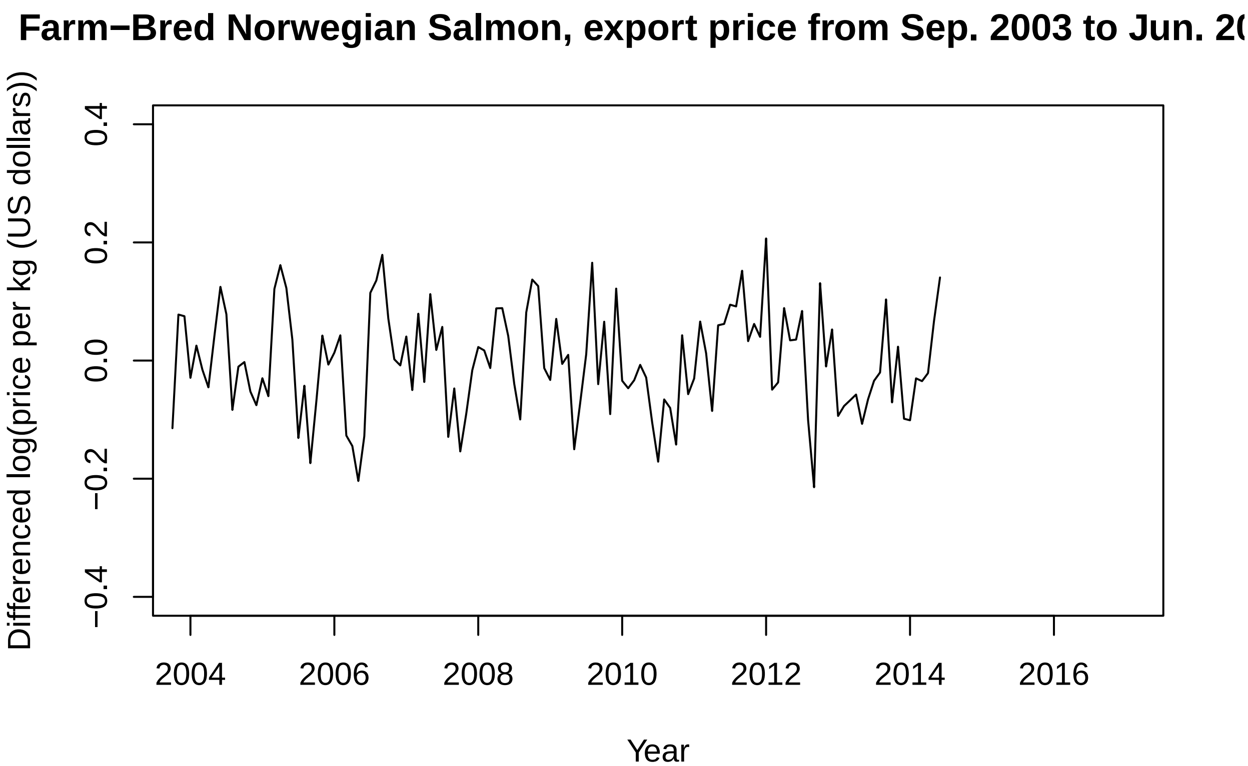
Although the series looks pretty stationary, let's do some frequency domain analysis for the seasonality patterns:

```
per.mod1<-periodogram(salmon_ts_log, log = 'no', xlim = c(0, 0.5))
```



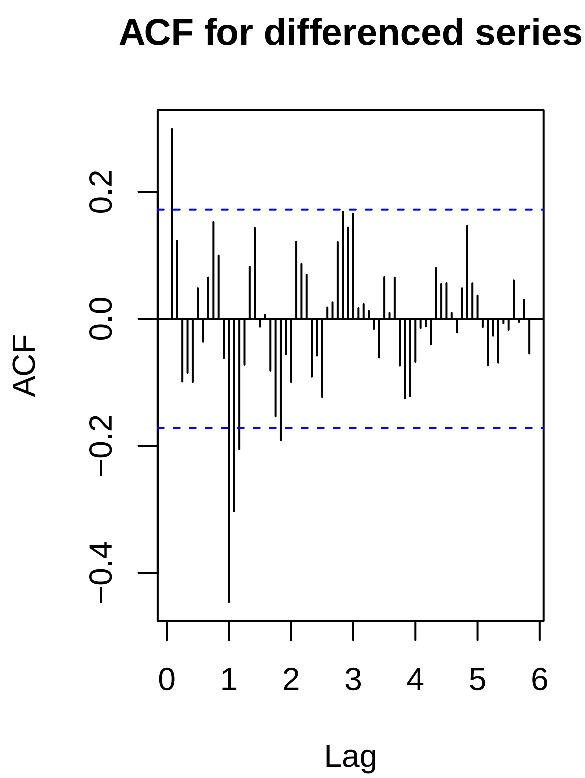
There's a definitive frequencies peak around 0.081 (period = 1/0.083 = 12 months = 1 year) as well as 0.020833333 (period = 1/0.020833333 = 48 months = 4 years). We will not work with larger periods as there's not enough data to deal with it.

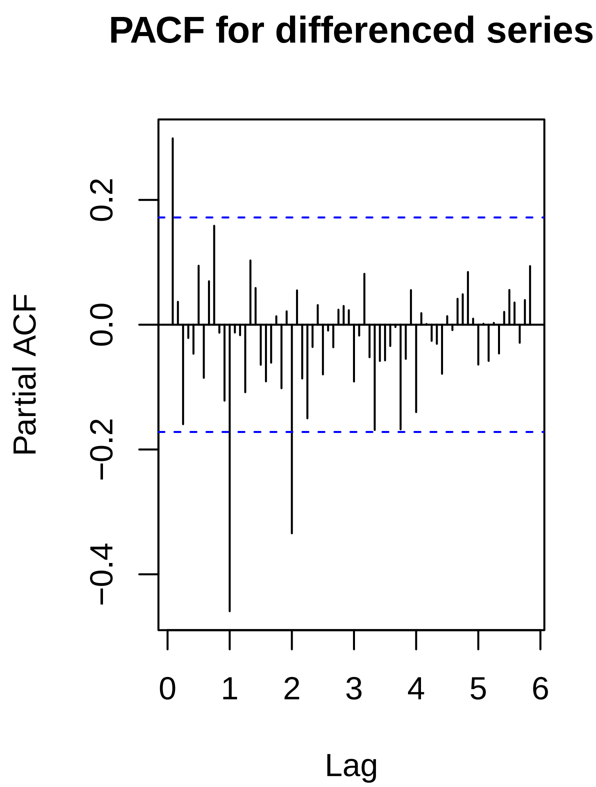
```
salmon_ts_log.diff<-c(NA, diff(salmon_ts_log.diff, lag = 12))
salmon_ts_log.diff <- ts(salmon_ts_log.diff, start = c(2003,9), deltat = 1/12)
plot(salmon_ts_log.diff, xlim = c(2004, 2017), ylim = c(-0.4, 0.4), xlab = "Year", ylab = "Differenced l
main  =  "Farm-Bred Norwegian Salmon, export price from Sep. 2003 to Jun. 2015")
```



Now the series look much more stationary. Let's calculate ACF/PACF of the differenced series:

```
par(mfrow = c(1, 2))
acf(salmon_ts_log.diff, main = "ACF for differenced series", na.action = na.pass,lag.max=70)
pacf(salmon_ts_log.diff, main = "PACF for differenced series", na.action = na.pass, lag.max=70)
```





We'd like to difference the trend and possibly the seasonality so d = 1 and D=0 or D = 1 with periods = 12 or 48.

To determine the AR and MA orders, we first decide on P and Q for the seasonal part, then p and q for the non-seasonal part.

The PACF seems to be non-zero at seasonal lags 12 months, suggesting P = 1 with period 12. It's not quite clear whether there is an analogous seasonal lag for ACF (e.g. Q=0 or Q=1).

For the non-seasonal component, both ACF and PACF could be non-zero at lag 1, p is either 1 or 2 and q is either 0 or 1.

Now let's fit multiplicative seasonal ARIMA models to the logged series with periods 12 and 48, and compare them by AIC to see how well the metric would correlate with our ACF/PACF plot interpretations.

```
pq.list <- list(list(c(1,1,0), c(1,0,0)), list(c(1,1,0), c(1,1,0)), list(c(2,1,0), c(1,0,0)),list(c(2,1
)
min.aic<-0
pq.values<-c(0,0,0)
PQ.values<-c(0,0,0)
period = 0
for (j in c(12, 24, 48)){
for (i in pq.list){
  fit.ARMA<-arima(salmon_ts_log, order = i[[1]],seasonal = list(order = i[[2]] , period = j))
  cat("pq/PQ coefficients",i[[1]],"and ",i[[2]], " with period", j,"gives sigma^2", round(fit.ARMA$sigma
  if (fit.ARMA$aic<min.aic){
    pq.values<-i[[1]]
    PQ.values<-i[[2]]
    period = j
```

```r
    min.aic=fit.ARMA$aic
  }
}
}
```

```
## pq/PQ coefficients 1 1 0 and  1 0 0  with period 12 gives sigma^2 0.0042346 and aic -365.4578
## pq/PQ coefficients 1 1 0 and  1 1 0  with period 12 gives sigma^2 0.0055377 and aic -297.5885
## pq/PQ coefficients 2 1 0 and  1 0 0  with period 12 gives sigma^2 0.0042345 and aic -363.4582
## pq/PQ coefficients 2 1 0 and  1 1 0  with period 12 gives sigma^2 0.0055072 and aic -296.2245
## pq/PQ coefficients 1 1 1 and  1 0 0  with period 12 gives sigma^2 0.0042345 and aic -363.458
## pq/PQ coefficients 2 1 1 and  1 0 0  with period 12 gives sigma^2 0.0040892 and aic -366.2196
## pq/PQ coefficients 2 1 1 and  1 1 0  with period 12 gives sigma^2 0.0053219 and aic -298.3961
## pq/PQ coefficients 1 1 0 and  1 0 1  with period 12 gives sigma^2 0.0038662 and aic -371.3271
## pq/PQ coefficients 1 1 0 and  1 1 1  with period 12 gives sigma^2 0.0037806 and aic -325.3892
## pq/PQ coefficients 2 1 0 and  1 0 1  with period 12 gives sigma^2 0.003846 and aic -369.5757
## pq/PQ coefficients 2 1 0 and  1 1 1  with period 12 gives sigma^2 0.0037707 and aic -323.9476
## pq/PQ coefficients 1 1 1 and  1 0 1  with period 12 gives sigma^2 0.0038578 and aic -369.4349
## pq/PQ coefficients 2 1 1 and  1 0 1  with period 12 gives sigma^2 0.00373 and aic -371.2874
## pq/PQ coefficients 2 1 1 and  1 1 1  with period 12 gives sigma^2 0.0037526 and aic -326.4631
## pq/PQ coefficients 1 1 0 and  1 0 0  with period 24 gives sigma^2 0.0044387 and aic -359.0803
## pq/PQ coefficients 1 1 0 and  1 1 0  with period 24 gives sigma^2 0.0055022 and aic -264.3026
## pq/PQ coefficients 2 1 0 and  1 0 0  with period 24 gives sigma^2 0.0044379 and aic -357.0924
## pq/PQ coefficients 2 1 0 and  1 1 0  with period 24 gives sigma^2 0.0054313 and aic -263.6104
## pq/PQ coefficients 1 1 1 and  1 0 0  with period 24 gives sigma^2 0.0044383 and aic -357.0864
## pq/PQ coefficients 2 1 1 and  1 0 0  with period 24 gives sigma^2 0.0043421 and aic -357.8439
## pq/PQ coefficients 2 1 1 and  1 1 0  with period 24 gives sigma^2 0.0052066 and aic -266.3389
## pq/PQ coefficients 1 1 0 and  1 0 1  with period 24 gives sigma^2 0.0039796 and aic -361.5361
## pq/PQ coefficients 1 1 0 and  1 1 1  with period 24 gives sigma^2 0.003738 and aic -272.3608
## pq/PQ coefficients 2 1 0 and  1 0 1  with period 24 gives sigma^2 0.003999 and aic -359.691
## pq/PQ coefficients 2 1 0 and  1 1 1  with period 24 gives sigma^2 0.0037039 and aic -271.3589
## pq/PQ coefficients 1 1 1 and  1 0 1  with period 24 gives sigma^2 0.0040498 and aic -359.5787
## pq/PQ coefficients 2 1 1 and  1 0 1  with period 24 gives sigma^2 0.003881 and aic -361.0296
## pq/PQ coefficients 2 1 1 and  1 1 1  with period 24 gives sigma^2 0.0035947 and aic -273.4228
## pq/PQ coefficients 1 1 0 and  1 0 0  with period 48 gives sigma^2 0.0043469 and aic -360.4626
## pq/PQ coefficients 1 1 0 and  1 1 0  with period 48 gives sigma^2 0.0058412 and aic -200.2045
## pq/PQ coefficients 2 1 0 and  1 0 0  with period 48 gives sigma^2 0.0043426 and aic -358.5217
## pq/PQ coefficients 2 1 0 and  1 1 0  with period 48 gives sigma^2 0.005766 and aic -199.9858
## pq/PQ coefficients 1 1 1 and  1 0 0  with period 48 gives sigma^2 0.0043458 and aic -358.4796
## pq/PQ coefficients 2 1 1 and  1 0 0  with period 48 gives sigma^2 0.0042281 and aic -359.4435
## pq/PQ coefficients 2 1 1 and  1 1 0  with period 48 gives sigma^2 0.0056486 and aic -201.2158
## pq/PQ coefficients 1 1 0 and  1 0 1  with period 48 gives sigma^2 0.0043466 and aic -358.5458
## pq/PQ coefficients 1 1 0 and  1 1 1  with period 48 gives sigma^2 0.0058378 and aic -198.2045
## pq/PQ coefficients 2 1 0 and  1 0 1  with period 48 gives sigma^2 0.0043437 and aic -356.6403
## pq/PQ coefficients 2 1 0 and  1 1 1  with period 48 gives sigma^2 0.0046132 and aic -198.0621
## pq/PQ coefficients 1 1 1 and  1 0 1  with period 48 gives sigma^2 0.0043295 and aic -356.6974
## pq/PQ coefficients 2 1 1 and  1 0 1  with period 48 gives sigma^2 0.0042047 and aic -357.6926
## pq/PQ coefficients 2 1 1 and  1 1 1  with period 48 gives sigma^2 0.0055385 and aic -199.2334
```

```r
(fit.ARMA<-arima(salmon_ts_log, order = pq.values, seasonal = list(order = PQ.values , period = period))
```

```
##
## Call:
## arima(x = salmon_ts_log, order = pq.values, seasonal = list(order = PQ.values,
##     period = period))
```

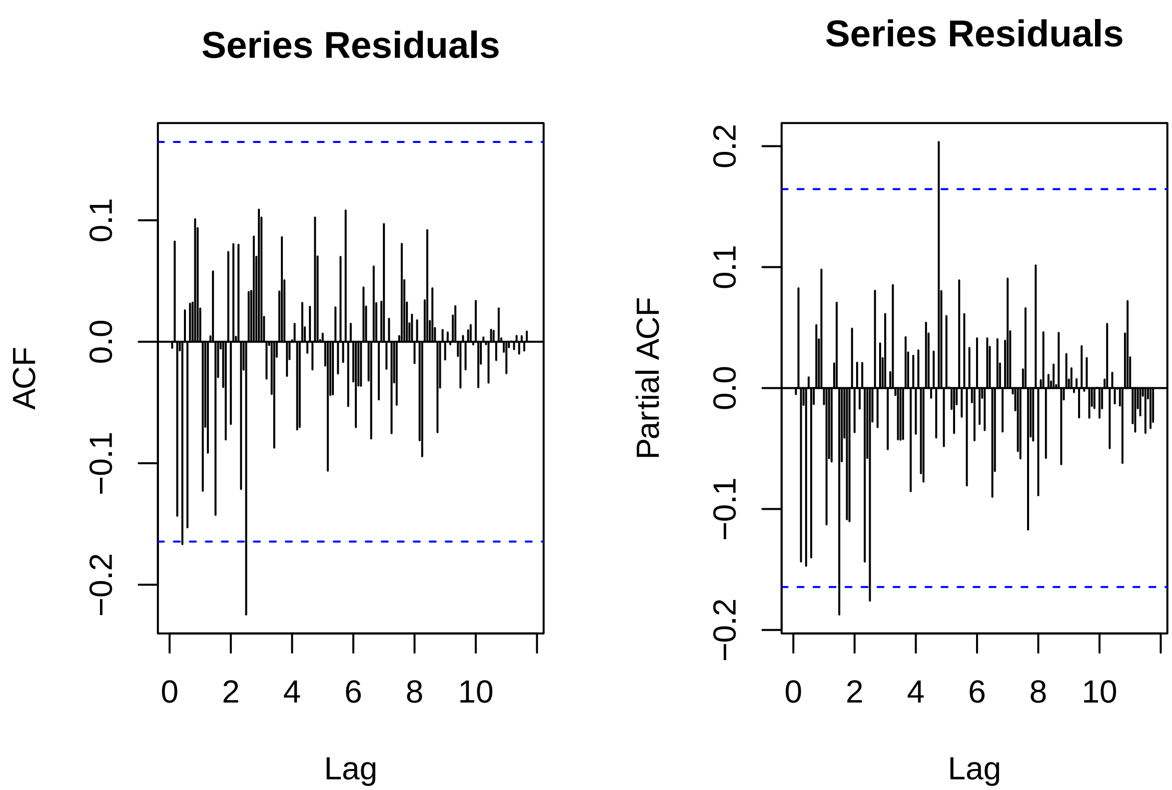
```
##
## Coefficients:
##          ar1     sar1     sma1
##       0.2776   0.9640  -0.8537
## s.e.  0.0821   0.0719   0.1607
##
## sigma^2 estimated as 0.003866:  log likelihood = 188.66,  aic = -371.33
```

```
cat("pq/PQ coefficients of", pq.values,"and",PQ.values, "with period", period, "gave minimal aic value
```

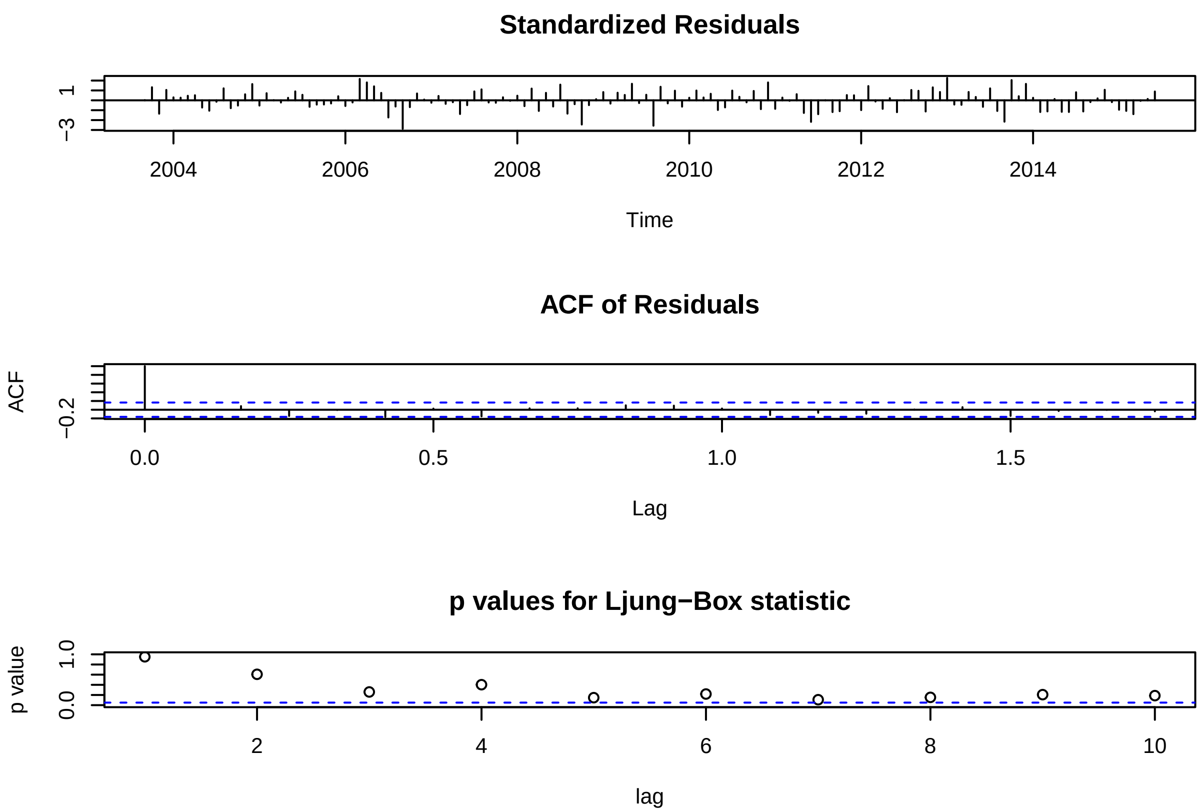
```
## pq/PQ coefficients of 1 1 0 and 1 0 1 with period 12 gave minimal aic value of -371.3271
```

The model with lowest AIC value correlates well with what was observed on ACF/PACF plots. It appears that models without seasonal differencing (D=0) give much better aic value (which could be due to the limited amount of data we have, especially for large periods like 4 years).
Now let's investigate diagnostics for this model, including autocorrelation and normality of the residuals.

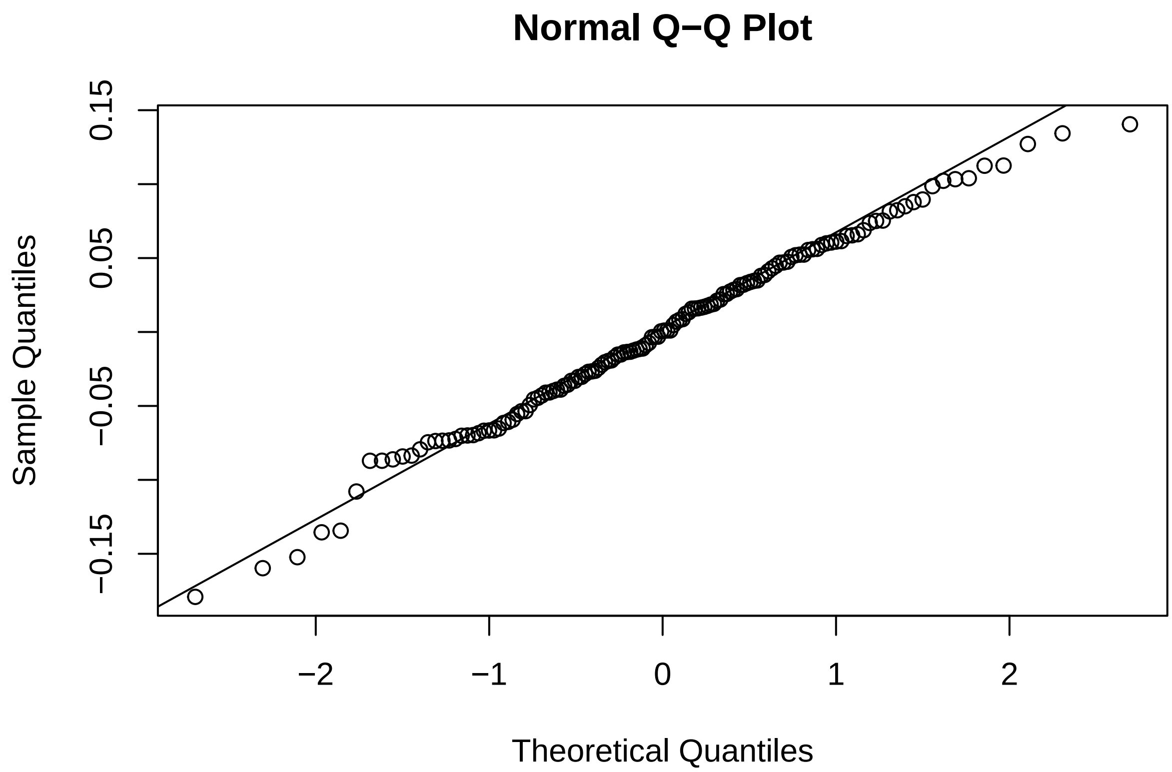
```
## Fitting residuals
par(mfrow = c(1, 2))
res <- fit.ARMA$residuals
acf(res, lag.max = 200, main = "Series Residuals")
pacf(res, lag.max = 200, main = "Series Residuals")
```




```
tsdiag(fit.ARMA)
```



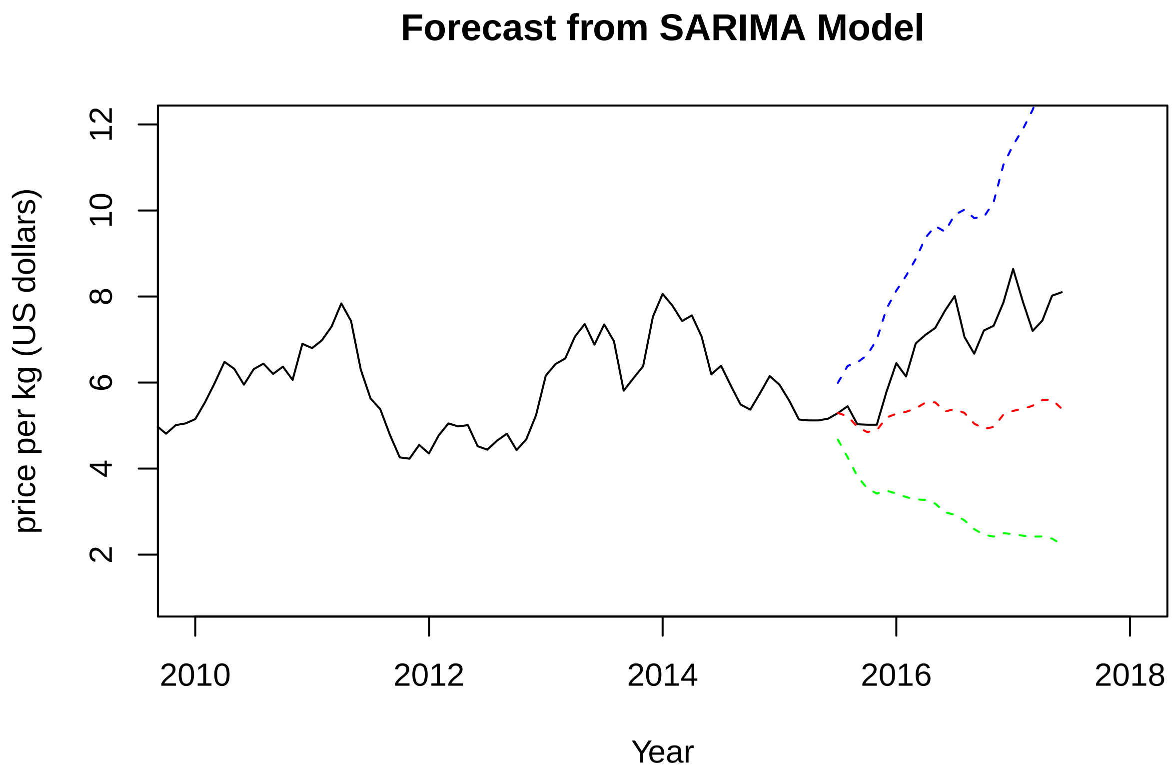

```
qqnorm(res)
qqline(res)
```



Overall, the distribution of residual seems to be independent of each other, at least for the middle quantiles. Autocorrelation p-values according to Ljung–Box are mostly above 0.05 (the data points are independently distributed) although some of them are questionable, especially at lag 7.

Now let's produce the forecasts for this series with a lead time of two years:

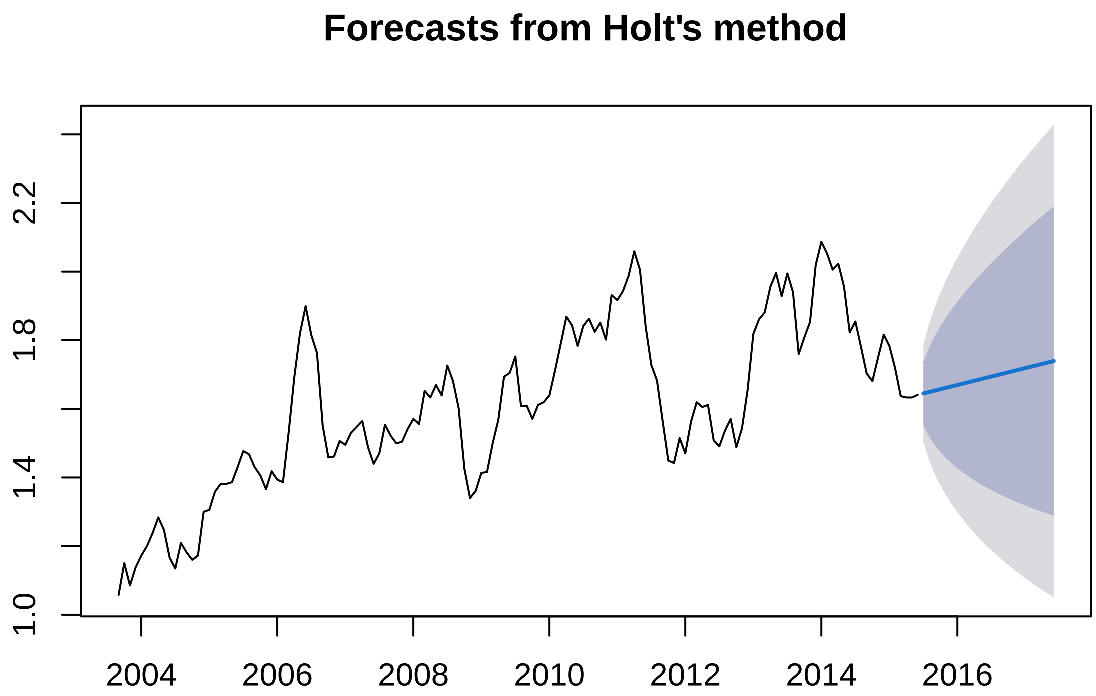
```
ts.plot(salmon, xlim = c(2010,2018), ylim = c(1,12), xlab  =  "Year",  ylab  =  "price per kg (US dolla
main  =  "Forecast from SARIMA Model")
SARIMA_forecast <- predict(fit.ARMA, n.ahead = 24)
####forecasted values
points(exp(SARIMA_forecast$pred), col = "red", type = 'l', lty = 2)
####2 sd forecasting limits
points(exp(SARIMA_forecast$pred-2*SARIMA_forecast$se),col='green', type = 'l', lty = 2)
points(exp(SARIMA_forecast$pred+2*SARIMA_forecast$se),col='blue', type = 'l', lty = 2)
```



Interestingly enough, the simpler SARIMA model provides a very similar prediction to ARIMA with high order parameters.

4. Forecasts from Holt-Winters multiplicative method

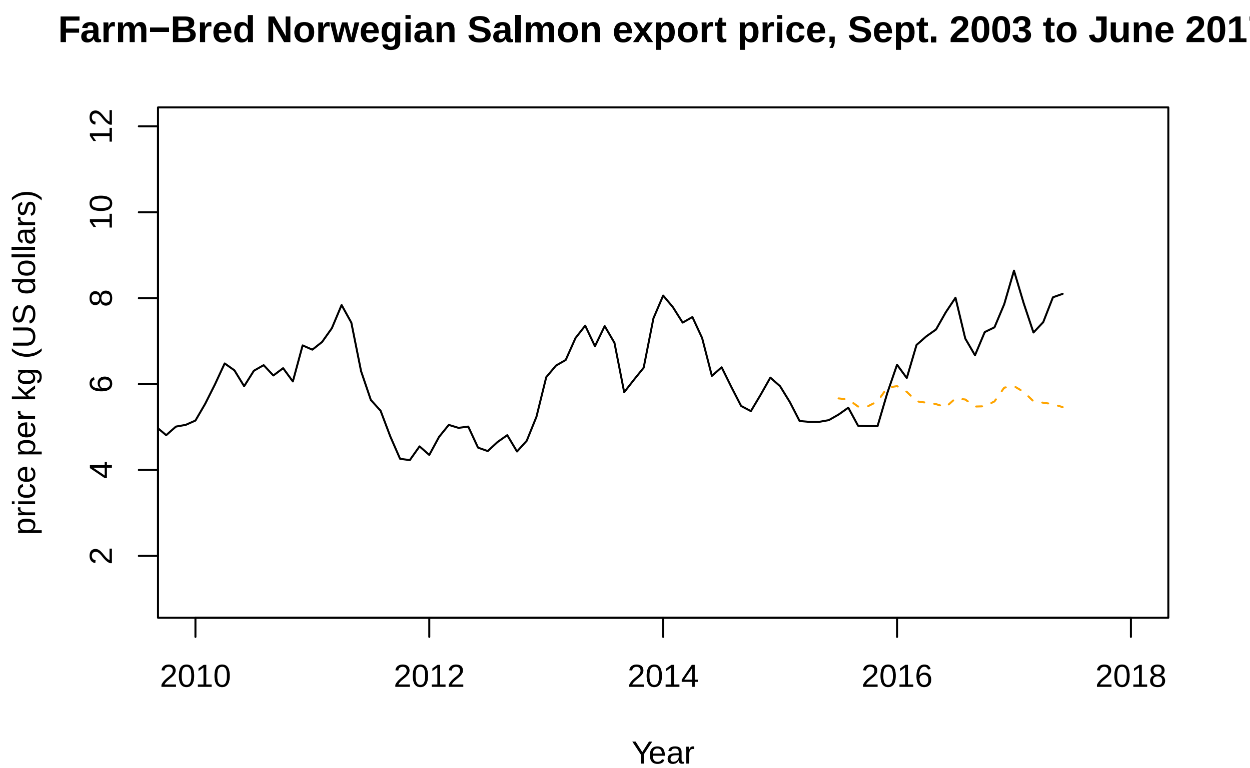
```
#The forecasting with linear trend:
HW_forecast <- holt(salmon_ts_log, seasonal = "multiplicative", h = 24)
plot(HW_forecast)
```



5. Forecasting with ML methods

```
library(randomForest)
library(zoo)
library(lubridate)
train_month<-month(as.yearmon(time(salmon_train)))
train_year<-year(as.yearmon(time(salmon_train)))
test_month<-month(as.yearmon(time(salmon_test)))
test_year<-year(as.yearmon(time(salmon_test)))

df_train <-as.data.frame(matrix(nrow=length(train_year),ncol=3))
df_train[1]<-train_month
df_train[2]<-train_year
df_train[3]<-salmon_train
colnames(df_train)<-c("Month", "Year","Price")

df_test <-as.data.frame(matrix(nrow=length(test_year),ncol=2))
df_test[1]<-test_month
df_test[2]<-test_year
colnames(df_test)<-c("Month", "Year")

rf = randomForest(Price ~ Month+Year, data = df_train)
RF_forecast = predict(rf, newdata = df_test)
RF_forecast<-ts(RF_forecast, start = c(2015,7), frequency = 12)

ts.plot(salmon, xlim = c(2010,2018), ylim = c(1,12), xlab  =  "Year",  ylab  =  "price per kg (US dolla
main  =  "Farm-Bred Norwegian Salmon export price, Sept. 2003 to June 2017")
```

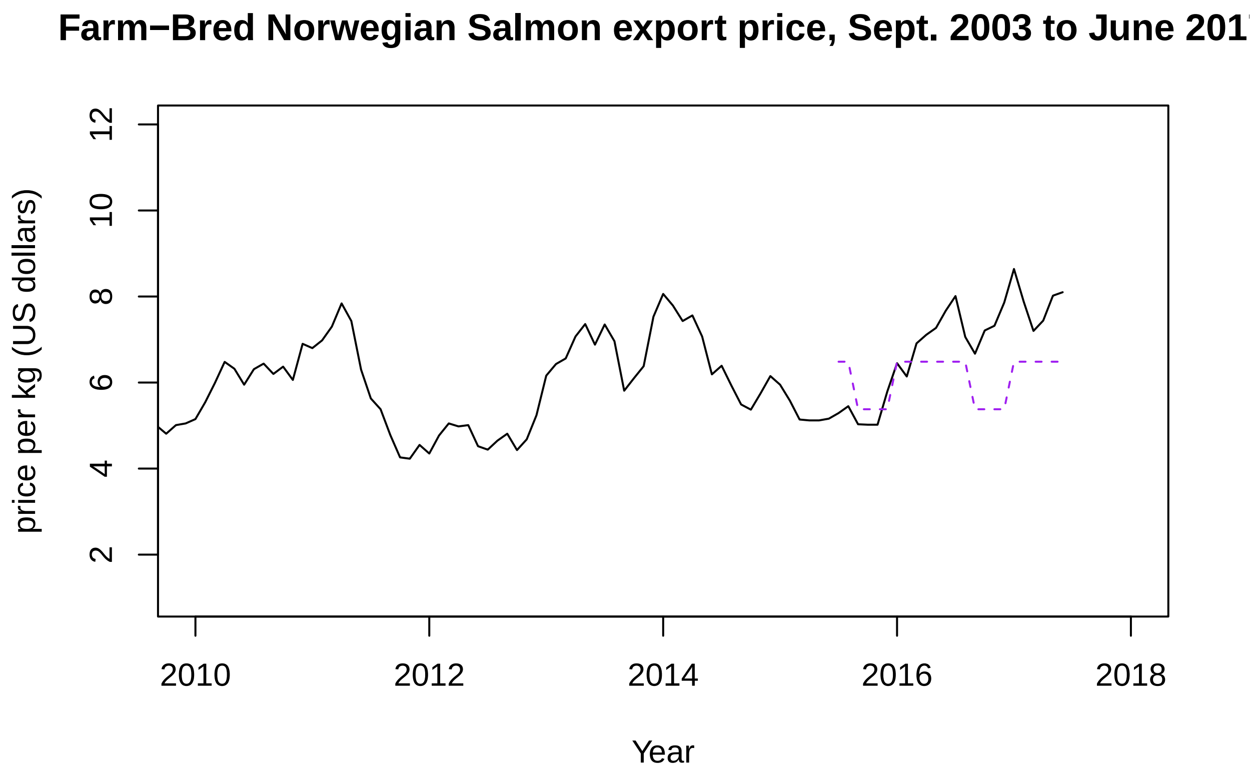
```r
points(RF_forecast, col = "orange", type = 'l', lty = 2)
```




```r
library(xgboost)

df_labels <-as.data.frame(matrix(nrow=length(train_year),ncol=1))
df_labels[1]<-salmon_train

xg = xgboost(data = as.matrix(df_train[1:2]), label = as.matrix(df_train[3]), max.depth = 2, eta = 1, n
```

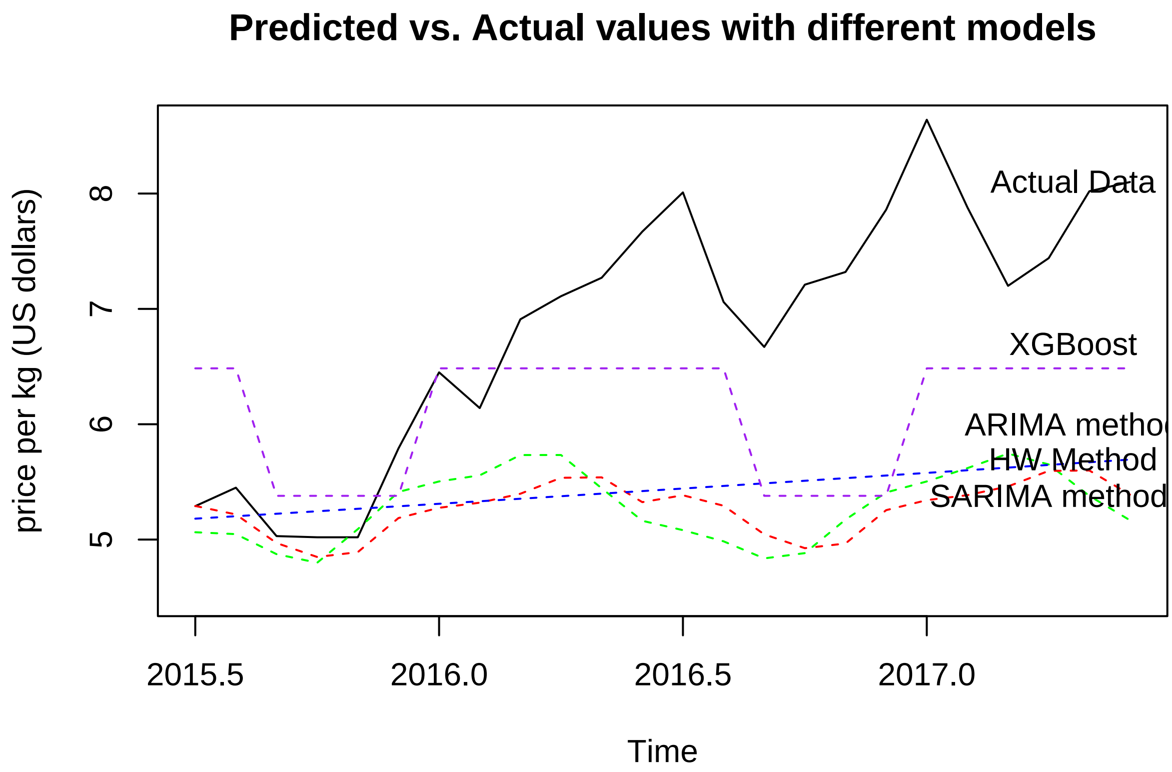
```
## [1]  train-rmse:0.810038
## [2]  train-rmse:0.744680
```

```r
XG_forecast = predict(xg, newdata = as.matrix(df_test))
XG_forecast<-ts(XG_forecast, start = c(2015,7), frequency = 12)

ts.plot(salmon, xlim = c(2010,2018), ylim = c(1,12), xlab  =  "Year",  ylab  =  "price per kg (US dolla
main  =  "Farm-Bred Norwegian Salmon export price, Sept. 2003 to June 2017")
points(XG_forecast, col = "purple", type = 'l', lty = 2)
```



5. Comparing the models.

```
ts.plot(salmon_test, main = "Predicted vs. Actual values with different models", ylab = "price per kg (U
text(x =2017.3, y = 8.105195, labels = c("Actual Data"))

points(ARIMA_forecast, col = "green", type = 'l', lty = 2)
text(x =2017.3, y = 6, labels = c("ARIMA method"))

points(exp(SARIMA_forecast$pred), col = "red", type = 'l', lty = 2)
text(x =2017.25, y = 5.382934, labels = c("SARIMA method"))

points(exp(HW_forecast$mean), col = "blue", type = 'l', lty = 2)
text(x =2017.3, y = 5.698073, labels = c("HW Method"))

#points(RF_forecast, col = "orange", type = 'l', lty = 2)
#text(x =2017.3, y = 6, labels = c("Random Forest"))

points(XG_forecast, col = "purple", type = 'l', lty = 2)
text(x =2017.3, y = 6.7, labels = c("XGBoost"))
```




```
library(kableExtra)
models<-c( exp(SARIMA_forecast$pred), exp(HW_forecast$mean), RF_forecast)

mspe<-function (x, y) mean((y - x)^2)
rsq <- function (x, y) cor(x, y) ^ 2

models.names<-c("ARIMA","SARIMA","Holt-Winters multiplicative method", "XGBoost")
mspe.models<-c( mspe(ARIMA_forecast, salmon_test), mspe(exp(SARIMA_forecast$pred), salmon_test), mspe(e
rsq.models<-c(rsq(ARIMA_forecast, salmon_test),rsq(exp(SARIMA_forecast$pred), salmon_test),rsq(exp(HW_f

models.comparison <- data.frame(models.names, mspe.models, rsq.models)
colnames(models.comparison)<- c("Model Names","Mean squared prediction error", "R-squared")

(models.comparison)
```

```
##                         Model Names Mean squared prediction error R-squared
## 1                             ARIMA                      3.445080 0.1652143
## 2                            SARIMA                      3.446801 0.3390391
## 3 Holt-Winters multiplicative method                      2.913708 0.7380610
## 4                           XGBoost                      1.514137 0.1655173
```

```
kable(models.comparison, format="latex", booktabs=TRUE) %>%
kable_styling(latex_options="scale_down")
```

Model Names	Mean squared prediction error	R-squared
ARIMA	3.445080	0.1652143
SARIMA	3.446801	0.3390391
Holt-Winters multiplicative method	2.913708	0.7380610
XGBoost	1.514137	0.1655173

Discussion

```
library(astsa)
library(TSA)
library(ggplot2)
library(ggfortify)
library(forecast)
library(ggthemes)
knitr::opts_chunk$set(echo = T, warning = F, message = F, fig.align = "center")
data(salmon)
salmon_test<-ts(salmon[143:166], start = c(2015,7), frequency = 12)
salmon_train<-ts(salmon[1:142], start = c(2003,9), frequency = 12)
head(salmon_train)
#plot
autoplot(salmon_train)+ geom_line( color="#F8766D")+xlab("Year")+ ylab("Price per kg (US dollars)")+ggt
#install.packages("fpp2")
library(fpp2)
ggseasonplot(salmon_train, year.labels=TRUE, year.labels.left=TRUE) +
  ylab("price per kg (US dollars)") + geom_smooth() +
  ggtitle("Farm-Bred Norwegian Salmon export price, Sept. 2003 to June 2015")

fit = stl(salmon, s.window=12)
plot(fit)
library(TTR)
salmon_ts_SMA3 <- SMA(salmon_train,n=3)
plot.ts(salmon_ts_SMA3)

salmon_ts_SMA12 <- SMA(salmon_train,n=12)
plot.ts(salmon_ts_SMA12, xlab  =  "Year",  ylab  =  "price per kg (US dollars)",
main  =  "12 Month Moving Average, Farm-Bred Norwegian Salmon export price")
plot(salmon_train, xlab = "Year", ylab = "price per kg (US dollars)")

salmon_ts_log <- log(salmon_train)
plot(salmon_ts_log, xlab = "Year", ylab = "log(price per kg (US dollars))")
salmon_ts_components <- decompose(salmon_train)
plot(salmon_ts_components)
res <- residuals(naive(salmon_train))
autoplot(res) + xlab("Year") + ylab("") +
  ggtitle("Residuals from Naïve Method")+ geom_line( color="#F8766D")+theme_economist(base_size = 8)

gghistogram(res) + ggtitle("Histogram of residuals")

checkresiduals(salmon_train, lag, df = NULL, plot = TRUE)
#acf and pacf plots
```

```
acf(salmon_train)
pacf(salmon_train)

#log acf and pacf plots
acf(salmon_ts_log)
pacf(salmon_ts_log)

sarima(salmon_train, 1, 0, 0)

AR1 <- arima(salmon_train, order = c(1,0,0))
print(AR1)

ts.plot(salmon_train, main = 'Monthly Salmon Prices with Predicted')
AR_fit = salmon_train - residuals(AR1)
points(AR_fit, type = 'l', col = "indianred", lty = 2)
period_predict <- 24

ts.plot(salmon_train, xlim = c(2010,2020))
AR_forecast <- predict(AR1, n.ahead = period_predict)$pred
AR_forecast_se <- predict(AR1, n.ahead = period_predict)$se
points(AR_forecast, type = 'l', col = 2)
points(salmon_test, type = 'l', col = 4)
points(AR_forecast - 2*AR_forecast_se, type = 'l', col = 2, lty = 2)
points(AR_forecast + 2*AR_forecast_se, type = 'l', col = 2, lty = 2)

period_predict <- 24

#salmon_train_sub <- subset(salmon_train, salmon_train[["Year"]] > 2003)
#first difference of original data
diff_salmon <- diff(salmon_train)
diff_salmon <- ts(diff_salmon, start = c(2003), deltat = 1)
plot(diff_salmon, xlab = "Year", ylab = "First order differenced series")

#sample ACF and PACF plots of the original series
acf(salmon_train, lag.max = 360, na.action = na.pass,
main = "ACF for original series")

pacf(salmon_train, lag.max = 360, na.action = na.pass,
main = "PACF for original series")

#max lag at 50 to discern where the p significant lag is
pacf(salmon_train, lag.max = 50, na.action = na.pass,
main = "PACF for original series (lag max = 50")

#sample ACF and PACF plots of the first difference original series
acf(diff_salmon, lag.max = 50, na.action = na.pass,
main = "ACF for differenced series")

pacf(diff_salmon, lag.max = 50, na.action = na.pass,
main = "PACF for differenced series")

#various Arima models
n <- length(diff_salmon)
```

```
(fit.2_1 <- arima(salmon_train, order=c(2,1,1)))
(fit.1_2 <- arima(salmon_train, order=c(1,1,2)))
(fit.2_2 <- arima(salmon_train, order=c(2,1,2)))
(fit.3_2 <- arima(salmon_train, order=c(3,1,2)))
(fit.2_3 <- arima(salmon_train, order=c(2,1,3)))
(fit.3_3 <- arima(salmon_train, order=c(3,1,3)))
(fit.4_3 <- arima(salmon_train, order=c(4,1,3)))
(fit.3_4 <- arima(salmon_train, order=c(3,1,4)))
(fit.4_4 <- arima(salmon_train, order=c(4,1,4)))
(fit.5_4 <- arima(salmon_train, order=c(5,1,4)))
(fit.4_5 <- arima(salmon_train, order=c(4,1,5)))
(fit.5_5 <- arima(salmon_train, order=c(5,1,5)))




par(mfrow = c(1, 2))
res <- fit.5_5$residuals
acf(res, lag.max = 20)+ geom_line( color="#F8766D")+theme_economist(base_size = 8)
pacf(res, lag.max = 20)
tsdiag(fit.5_5)
qqnorm(res)
qqline(res)
checkresiduals(fit.5_5)
#autoplot(forecast(fit.5_5))
pred <- predict(fit.5_5, n.ahead = 24)

ts.plot(salmon_train, xlim = c(2003, 2020), ylim = c(0,10),
ylab = "Salmon Price in kg (US dollars)", main = "Forecast from ARIMA Model")
ARIMA_forecast <- predict(fit.5_5, n.ahead = period_predict)$pred
ARIMA_forecast_se <- predict(fit.5_5, n.ahead = period_predict)$se
points(ARIMA_forecast, type = 'l', col = 2)
points(salmon_test, type = 'l', col = 4)
points(ARIMA_forecast - 2*ARIMA_forecast_se, type = 'l', col = 2, lty = 2)
points(ARIMA_forecast + 2*ARIMA_forecast_se, type = 'l', col = 2, lty = 2)
salmon_ts_log.diff<-diff(salmon_ts_log, lag = 1)
salmon_ts_log.diff <- ts(salmon_ts_log.diff, start = c(2003,9), deltat = 1/12)
plot(salmon_ts_log.diff, xlim = c(2004, 2017), ylim = c(-0.4, 0.4), xlab = "Year", ylab = "Differenced 
main  =  "Farm-Bred Norwegian Salmon, export price from Sep. 2003 to Jun. 2015")
per.mod1<-periodogram(salmon_ts_log, log = 'no', xlim = c(0, 0.5))
salmon_ts_log.diff<-c(NA, diff(salmon_ts_log.diff, lag = 12))
salmon_ts_log.diff <- ts(salmon_ts_log.diff, start = c(2003,9), deltat = 1/12)
plot(salmon_ts_log.diff, xlim = c(2004, 2017), ylim = c(-0.4, 0.4), xlab = "Year", ylab = "Differenced 
main  =  "Farm-Bred Norwegian Salmon, export price from Sep. 2003 to Jun. 2015")
par(mfrow = c(1, 2))
acf(salmon_ts_log.diff, main = "ACF for differenced series", na.action = na.pass,lag.max=70)
pacf(salmon_ts_log.diff, main = "PACF for differenced series", na.action = na.pass, lag.max=70)
pq.list <- list(list(c(1,1,0), c(1,0,0)), list(c(1,1,0), c(1,1,0)), list(c(2,1,0), c(1,0,0)),list(c(2,1
)
min.aic<-0
pq.values<-c(0,0,0)
PQ.values<-c(0,0,0)
period = 0
for (j in c(12, 24, 48)){
```

```
for (i in pq.list){
  fit.ARMA<-arima(salmon_ts_log, order = i[[1]],seasonal = list(order = i[[2]] , period = j))
  cat("pq/PQ coefficients",i[[1]],"and ",i[[2]], " with period", j,"gives sigma^2", round(fit.ARMA$sigma
  if (fit.ARMA$aic<min.aic){
    pq.values<-i[[1]]
    PQ.values<-i[[2]]
    period = j
    min.aic=fit.ARMA$aic
  }
}
}
(fit.ARMA<-arima(salmon_ts_log, order = pq.values, seasonal = list(order = PQ.values , period = period)
 cat("pq/PQ coefficients of", pq.values,"and",PQ.values, "with period", period, "gave minimal aic value
## Fitting residuals
par(mfrow = c(1, 2))
res <- fit.ARMA$residuals
acf(res, lag.max = 200, main = "Series Residuals")
pacf(res, lag.max = 200, main = "Series Residuals")
tsdiag(fit.ARMA)
qqnorm(res)
qqline(res)
ts.plot(salmon, xlim = c(2010,2018), ylim = c(1,12), xlab  =  "Year",  ylab  =  "price per kg (US dolla
main  =  "Forecast from SARIMA Model")
SARIMA_forecast <- predict(fit.ARMA, n.ahead = 24)
####forecasted values
points(exp(SARIMA_forecast$pred), col = "red", type = 'l', lty = 2)
####2 sd forecasting limits
points(exp(SARIMA_forecast$pred-2*SARIMA_forecast$se),col='green', type = 'l', lty = 2)
points(exp(SARIMA_forecast$pred+2*SARIMA_forecast$se),col='blue', type = 'l', lty = 2)
#The forecasting with linear trend:
HW_forecast <- holt(salmon_ts_log, seasonal = "multiplicative", h = 24)
plot(HW_forecast)
library(randomForest)
library(zoo)
library(lubridate)
train_month<-month(as.yearmon(time(salmon_train)))
train_year<-year(as.yearmon(time(salmon_train)))
test_month<-month(as.yearmon(time(salmon_test)))
test_year<-year(as.yearmon(time(salmon_test)))

df_train <-as.data.frame(matrix(nrow=length(train_year),ncol=3))
df_train[1]<-train_month
df_train[2]<-train_year
df_train[3]<-salmon_train
colnames(df_train)<-c("Month", "Year","Price")

df_test <-as.data.frame(matrix(nrow=length(test_year),ncol=2))
df_test[1]<-test_month
df_test[2]<-test_year
colnames(df_test)<-c("Month", "Year")

rf = randomForest(Price ~ Month+Year, data = df_train)
RF_forecast = predict(rf, newdata = df_test)
```

```
RF_forecast<-ts(RF_forecast, start = c(2015,7), frequency = 12)

ts.plot(salmon, xlim = c(2010,2018), ylim = c(1,12), xlab  =  "Year",  ylab  =  "price per kg (US dolla
main  =  "Farm-Bred Norwegian Salmon export price, Sept. 2003 to June 2017")
points(RF_forecast, col = "orange", type = 'l', lty = 2)
library(xgboost)

df_labels <-as.data.frame(matrix(nrow=length(train_year),ncol=1))
df_labels[1]<-salmon_train

xg = xgboost(data = as.matrix(df_train[1:2]), label = as.matrix(df_train[3]), max.depth = 2, eta = 1, n
XG_forecast = predict(xg, newdata = as.matrix(df_test))
XG_forecast<-ts(XG_forecast, start = c(2015,7), frequency = 12)

ts.plot(salmon, xlim = c(2010,2018), ylim = c(1,12), xlab  =  "Year",  ylab  =  "price per kg (US dolla
main  =  "Farm-Bred Norwegian Salmon export price, Sept. 2003 to June 2017")
points(XG_forecast, col = "purple", type = 'l', lty = 2)
ts.plot(salmon_test, main = "Predicted vs. Actual values with different models", ylab = "price per kg (
text(x =2017.3, y = 8.105195, labels = c("Actual Data"))

points(ARIMA_forecast, col = "green", type = 'l', lty = 2)
text(x =2017.3, y = 6, labels = c("ARIMA method"))

points(exp(SARIMA_forecast$pred), col = "red", type = 'l', lty = 2)
text(x =2017.25, y = 5.382934, labels = c("SARIMA method"))

points(exp(HW_forecast$mean), col = "blue", type = 'l', lty = 2)
text(x =2017.3, y = 5.698073, labels = c("HW Method"))

#points(RF_forecast, col = "orange", type = 'l', lty = 2)
#text(x =2017.3, y = 6, labels = c("Random Forest"))

points(XG_forecast, col = "purple", type = 'l', lty = 2)
text(x =2017.3, y = 6.7, labels = c("XGBoost"))

library(kableExtra)
models<-c( exp(SARIMA_forecast$pred), exp(HW_forecast$mean), RF_forecast)

mspe<-function (x, y) mean((y - x)^2)
rsq <- function (x, y) cor(x, y) ^ 2

models.names<-c("ARIMA","SARIMA","Holt-Winters multiplicative method", "XGBoost")
mspe.models<-c( mspe(ARIMA_forecast, salmon_test), mspe(exp(SARIMA_forecast$pred), salmon_test), mspe(e
rsq.models<-c(rsq(ARIMA_forecast, salmon_test),rsq(exp(SARIMA_forecast$pred), salmon_test),rsq(exp(HW_f

models.comparison <- data.frame(models.names, mspe.models, rsq.models)
colnames(models.comparison)<- c("Model Names","Mean squared prediction error", "R-squared")

(models.comparison)

kable(models.comparison, format="latex", booktabs=TRUE) %>%
kable_styling(latex_options="scale_down")
```