**Final Project: Group 21 //** ST 566 // Winter 2022

[Ruslan Mamedov](mailto:mamedovr@oregonstate.edu), [Alex Vincent Kretschmer](mailto:kretscal@oregonstate.edu)

Introduction

Using the salmon price dataset showing farm-bred Norwegian salmon export price from the ASTSA package (*Salmon: Monthly Export Price of Salmon in Astsa: Applied Statistical Time Series Analysis*, n.d.), we examine various methodologies and test different time series models to provide a forecast for the price going forward 2 years. The original dataset shows the price per kg in US dollars from September 2003 to June 2017. We used a training set of the first 12 years of data and the test data included the last two from 2015 to 2017 for comparison to our model forecasts

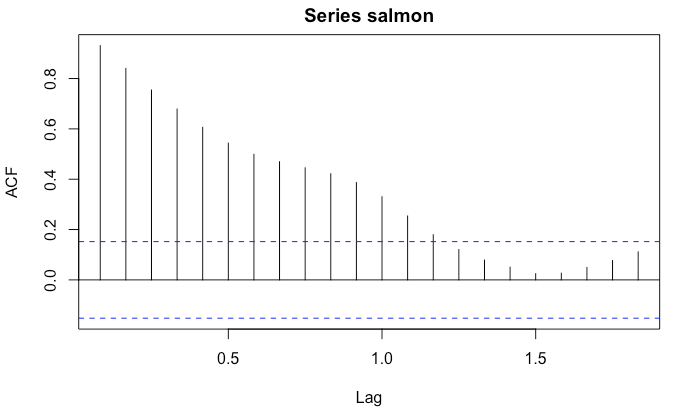
Methods

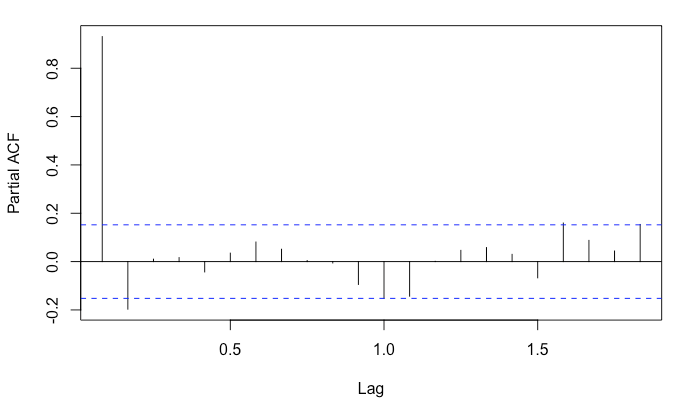
As with any time series analysis, it is helpful to first plot the training data:

Chart, line chart

Description automatically generated

There is a positive long-term slope over time, showing an upward trend. There also seems to be increasing variance as time goes on. Seasonality is not obvious at first glance, though price swings may be on a larger time scale (between 3-4 years potentially).

ACF and PACF plots suggest a lack of stationarity due to the geometric decay shown in the ACF plots and the cut off after lag 1 in the PACF plot. This results in eliminating an ARMA model from our selection.:



A Dickey-Fuller test can help determine the stationarity of the first order difference time series shown:

Chart, line chart

Description automatically generated

The data does appear to be stationary after first order differencing.

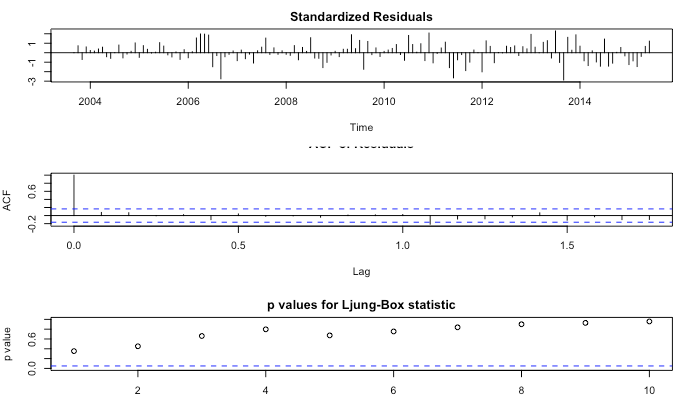
In determining an appropriate model for time series, we perform diagnostics from various models including:

* ARIMA
* SARIMA
* Holt-Winters exponential smoothing
* ML models: XGBoost

Results

* ARIMA

A multitude of various ARIMA models with first order differencing were constructed and then compared using their AIC. The lowest AIC resulted with a model of ARIMA(5,1,5) model. The diagnostics for the model all seemed appropriate to continue:



Residuals did not show a pattern and ACF plot was shown not to have any statistically significant lags. The p-values were all well above .05 for the Ljung-Box statistic.

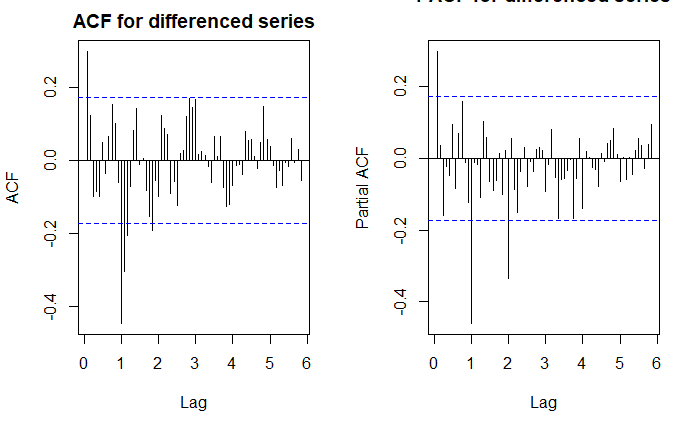
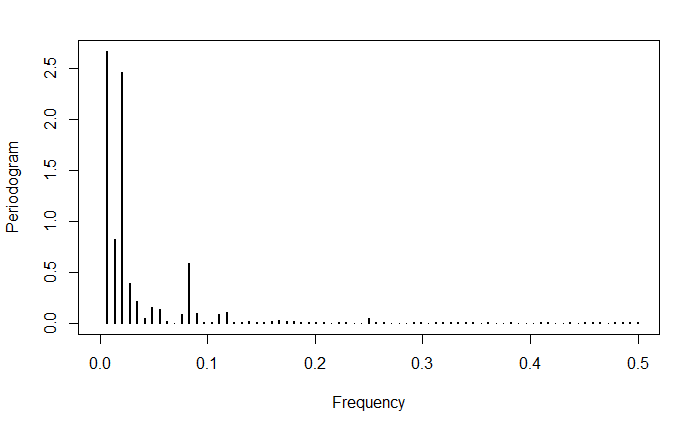
Using the salmon training data, the plot of the forecast (and 95% confidence interval) is shown below:

Chart, line chart

Description automatically generated

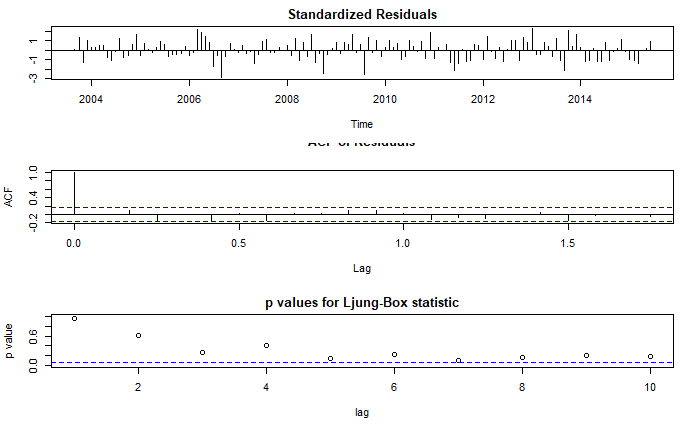
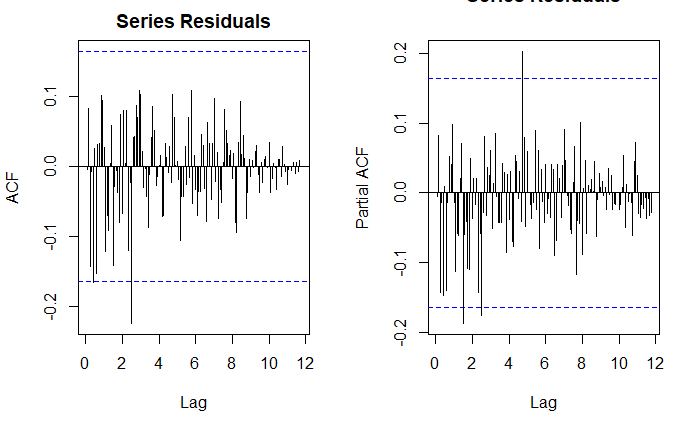
* SARIMA

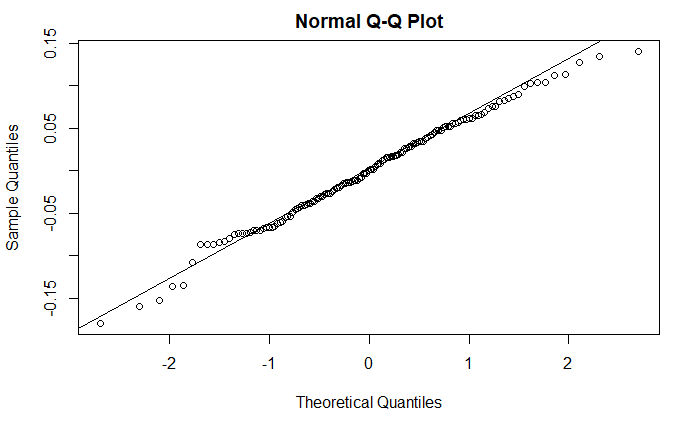
Although the best-fit ARIMA model above does capture the main patterns in predicted series, it might be quite challenging for interpretation due to high order parameters (p=5, q=5). Hence, we try applying the SARIMA method to see if capturing the trend and seasonality patterns can provide a simpler model with comparable accuracy. ACF and PACF plots on the differenced series along with a periodogram are used to search for the potential parameter values of SARIMA(p,d,q)x(PDQ)S model.



There's a definitive frequency peak around 0.081 (period = 1/0.083 = 12 months = 1 year) as well as at 0.020833333 (period = 1/0.020833333 = 48 months = 4 years). We will not work with larger periods due to the lack of data. Since we are differencing the trend and possibly the seasonality, d = 1 and D=0 or D = 1 with periods = 12 or 48 based on periodogram. 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 appears to be non-zero at seasonal lag 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 non-seasonal components, 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 we fit multiplicative seasonal ARIMA models to the logged series with periods 12 and 48, and compare them by AIC. The model with pdq/PDQ coefficients of 1 1 0 and 1 0 1 (period=12) gives minimal AIC value of -371.3271 and hence is used for the prediction. The diagnostics for this model, including autocorrelation and normality of the residuals, are presented below:





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

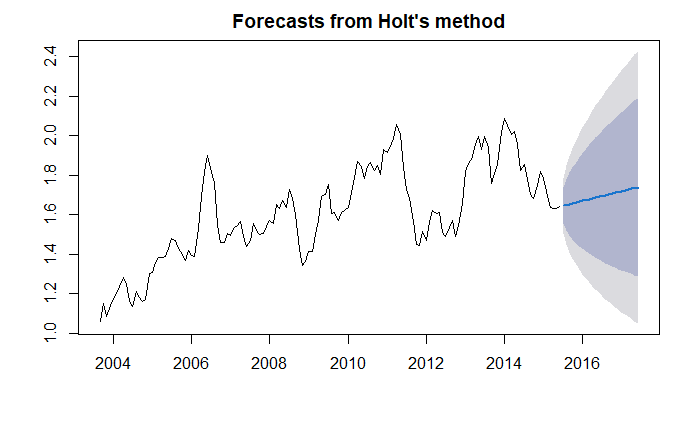
Below is the prediction on the hold out test set using our SARIMA model. Although the predicted values do run below the actual ones, they do resemble the patterns in the test dataset. It’s also interesting to note that the simpler SARIMA model provides a very similar prediction to ARIMA with high order parameters:

Chart, line chart, histogram

Description automatically generated

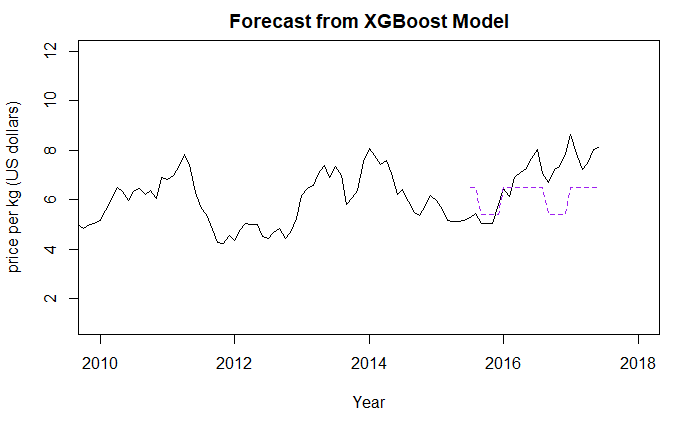
* Holt-Winters exponential smoothing with multiplicative seasonality

We also apply Holt-Winters exponential smoothing, a method based on simple updating equations rather than any time series models. Although it does not help with understanding the data generating mechanism, it is still useful for forecasting and accuracy comparison against time series based models.



* Machine Learning model: XGBoost

XGBoost Machine Learning modeling is used to compare it with the classical methods of time series forecasting in terms of accuracy.



* Model Comparison

Table

Description automatically generated

Chart, line chart

Description automatically generated

We use the following two metrics for the model comparison:

1. Mean squared prediction error (MSPE) to compare the accuracy of the models
2. R-squared to assess how well each prediction depicts the patterns in the original data.

It is interesting to note that although exponential smoothing provides the best accuracy among non-ML based models, SARIMA did a better job depicting patterns in the original data (R-squared twice less than for Holt-Winters method). The accuracy of the SARIMA model is practically identical to higher order ARIMA one. That being said, the ARIMA model has twice less R-squared value which given the same accuracy of the models could be an indication of overfitting for ARIMA.

The machine learning method demonstrates a better accuracy than the rest of the techniques but doesn’t provide much insight on the data generating process or model interpretation. It is also of interest the similarities in plot patterns between XGBoost and the ARIMA/SARIMA models.

Discussion

The dataset did not seem to follow the typical pattern for seasonality with certain months/quarters showing consistently higher or lower prices. This added a bit of complexity when comparing the various models and knowing when to adjust for seasonality or the lack thereof. Based on the results shown above, there’s no clear winner when forecasting on a given series: each model has its pros and cons and the final choice would depends on goals of the analysis and user requirements.

Among non-ML based models, SARIMA did a better job depicting patterns in the original data (R-squared twice less than for Holt-Winters method). The accuracy of the SARIMA model is practically identical to higher order ARIMA one. On the other hand, Holt-Winters model is definitely easier for the explanation and provides the best accuracy among non-ML based models, although it doesn’t depict the patterns in the original series.

The machine learning method accuracy is at least twice as good as the rest of the models but provides less transparency with the model building process and interpretation. Interestingly enough, the patterns depicted by XGBoost are very similar to the ones from ARIMA/SARIMA while they are obtained using different methodologies.

Also to note, we did not hear back from Jack regarding the group project, so this work was done by Ruslan and Alex exclusively. Alex provided the introduction and initial plots as well as the ARMA and ARIMA models and diagnostics, while Ruslan provided SARIMA, Holt-Winters, and Machine learning models, diagnostics, and plots. Ruslan also provided the model comparison tables and discussion.

Appendix:

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)")+ggtitle("Farm-Bred Norwegian Salmon Export Price")+theme\_economist(base\_size = 8)

#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

47

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)

48

(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 log(price per kg (US dollars))",

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 log(price per kg (US dollars))",

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(2,1,0), c(1,1,0)), list(c(1,1,1), c(1,0,0)), list(c(2,1,1), c(1,0,0)), list(c(2,1,1), c(1,1,0)), list(c(1,1,0), c(1,0,1)), list(c(1,1,0), c(1,1,1)), list(c(2,1,0), c(1,0,1)), list(c(2,1,0), c(1,1,1)), list(c(1,1,1), c(1,0,1)), list(c(2,1,1), c(1,0,1)),list(c(2,1,1), c(1,1,1))

)

min.aic<-0

pq.values<-c(0,0,0)

PQ.values<-c(0,0,0)

period = 0

for (j in c(12, 24, 48)){

49

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 sigmaˆ2", round(fit.ARMA$sigma2,7),"and aic",fit.ARMA$aic,"\n")

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 of",fit.ARMA$aic,"\n")

## 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 dollars)",

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)

50

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 dollars)",

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, nthread = 2, nrounds = 2)

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 dollars)",

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 (US dollars)", ylim = c(4.5,8.6))

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)ˆ2)

rsq <- function (x, y) cor(x, y) ˆ 2

models.names<-c("ARIMA","SARIMA","Holt-Winters multiplicative method", "XGBoost")

mspe.models<-c( mspe(ARIMA\_forecast, salmon\_test), mspe(exp(SARIMA\_forecast$pred), salmon\_test), mspe(exp(HW\_forecast$mean), salmon\_test), mspe(XG\_forecast, salmon\_test))

rsq.models<-c(rsq(ARIMA\_forecast, salmon\_test),rsq(exp(SARIMA\_forecast$pred), salmon\_test),rsq(exp(HW\_forecast$mean), salmon\_test), rsq(XG\_forecast, salmon\_test))

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")