HW3

Angel Sarmiento

2/17/2020

```
#library Import
library(caret)
library(tidyverse)

library(fable)
library(feasts)
library(fredr)
library(tsibble)
library(kableExtra)
library(plyr)
set.seed(23)
```

Problem 1

```
#creating the tibble/dataframe with 30 observations
tsdf <- tibble(ts_index = c(1:30), r = rnorm(30))

tsdf$y <- 0

#logic that doesnt really work, but sets the y values according to a funciton with 3 lags
tsdf <- tsdf %>%
  mutate(y = ifelse(ts_index < 4, r,
            y = 0.5 + 0.5*lag(y,1) - 0.1*lag(y, 2) + 0.25*lag(y, 3) + r))

#replacing rows 1 through 3 with the associated r values
tsdf[1:3,3] <- tsdf[1:3,2]
```

```
#making 7 different possible Autoregressive models (1, 2, 3) using loocv in caret
train_ct1 <- trainControl(method = "LOOCV")

# 1,2,3 lag structure
lm_model1 <- train(y ~ lag(y) + lag(y, 2) + lag(y, 3),  data = tsdf,
                   na.action = na.pass,
                   trControl = train_ct1,
                   method = "lm")
# 1 2 lag structure
lm_model2 <- train(y ~ lag(y) +  lag(y, 2),  data = tsdf,
                   na.action = na.pass,
                   trControl = train_ct1,
```

```
                  method = "lm")
# 1 3 lag structure
lm_model3 <- train(y ~ lag(y) + lag(y, 3),  data = tsdf,
                  na.action = na.pass,
                  trControl = train_ct1,
                  method = "lm")

lm_model4 <- train(y ~ lag(y, 2) + lag(y, 3),  data = tsdf,
                  na.action = na.pass,
                  trControl = train_ct1,
                  method = "lm")

lm_model5 <- train(y ~ lag(y),  data = tsdf,
                  na.action = na.pass,
                  trControl = train_ct1,
                  method = "lm")

lm_model6 <- train(y ~ lag(y, 2),  data = tsdf,
                  na.action = na.pass,
                  trControl = train_ct1,
                  method = "lm")

lm_model7 <- train(y ~ lag(y, 3),  data = tsdf,
                  na.action = na.pass,
                  trControl = train_ct1,
                  method = "lm")
```

```
#binding all of the results by rows into one matrix
results_matrix <- as_tibble(bind_rows(lm_model1$results,
                              lm_model2$results,
                              lm_model3$results,
                              lm_model4$results,
                              lm_model5$results,
                              lm_model6$results,
                              lm_model7$results))
results_matrix[,1] <- NULL

#Matrix of the AIC values
AIC_matrix <- AIC(lm_model1$finalModel, lm_model2$finalModel, lm_model3$finalModel,
                  lm_model4$finalModel, lm_model5$finalModel, lm_model6$finalModel, lm_model7$finalModel
```

```
Warning in AIC.default(lm_model1$finalModel, lm_model2$finalModel,
lm_model3$finalModel, : models are not all fitted to the same number of
observations
```

```
#Removing the first column
AIC_matrix[,1] <- NULL

BIC_matrix <- BIC(lm_model1$finalModel, lm_model2$finalModel, lm_model3$finalModel,
                  lm_model4$finalModel, lm_model5$finalModel, lm_model6$finalModel, lm_model7$finalModel
```

```
Warning in BIC.default(lm_model1$finalModel, lm_model2$finalModel,
```

```
lm_model3$finalModel, : models are not all fitted to the same number of
observations
```

```
#removing the first column
BIC_matrix[,1] <- NULL

#appending them to the results matrix
results_matrix['AIC'] <- AIC_matrix
results_matrix['BIC'] <- BIC_matrix
```

```
#similar cross-validation to above, except using k-folds


#making 7 different possible Autoregressive models (1, 2, 3) using loocv in caret
train_ct2 <- trainControl(method = "cv", number = 10)

# Creating a training set using 80% of the data
inTrain2 <- createDataPartition(y = tsdf$y, p = 0.8, list = FALSE)

#training data
train_set2 <- tsdf[inTrain2, ]
#test data (with the other 20 percent)
test_set2 <- tsdf[-inTrain2, ]




# 1,2,3 lag structure
lm_model_cv1 <- train(y ~ lag(y) + lag(y, 2) + lag(y, 3),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")
# 1 2 lag structure
lm_model_cv2 <- train(y ~ lag(y) +  lag(y, 2),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")
```

```
Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
There were missing values in resampled performance measures.
```

```
# 1 3 lag structure
lm_model_cv3 <- train(y ~ lag(y) + lag(y, 3),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")

lm_model_cv4 <- train(y ~ lag(y, 2) + lag(y, 3),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")
```

```
Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
There were missing values in resampled performance measures.
```

```
lm_model_cv5 <- train(y ~ lag(y),  data = train_set2,
                   na.action = na.pass,
                   trControl = train_ct2,
                   method = "lm")
```

```
Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
There were missing values in resampled performance measures.
```

```
lm_model_cv6 <- train(y ~ lag(y, 2),  data = train_set2,
                   na.action = na.pass,
                   trControl = train_ct2,
                   method = "lm")
```

```
Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
There were missing values in resampled performance measures.
```

```
lm_model_cv7 <- train(y ~ lag(y, 3),  data = train_set2,
                   na.action = na.pass,
                   trControl = train_ct2,
                   method = "lm")

beepr::beep("coin")
```

```
#Getting the results and appending them to the same matrix as before

models <- c(lm_model_cv1, lm_model_cv2, lm_model_cv3,
            lm_model_cv4, lm_model_cv5, lm_model_cv6, lm_model_cv7)

predictions1 <- predict(lm_model_cv1, test_set2)
predictions2 <- predict(lm_model_cv2, test_set2)
predictions3 <- predict(lm_model_cv3, test_set2)
predictions4 <- predict(lm_model_cv4, test_set2)
predictions5 <- predict(lm_model_cv5, test_set2)
predictions6 <- predict(lm_model_cv6, test_set2)
predictions7 <- predict(lm_model_cv7, test_set2)

#post prediction resampling
newdata1 <- postResample(predictions1, test_set2$y)
newdata2 <- postResample(predictions2, test_set2$y)
newdata3 <- postResample(predictions3, test_set2$y)
newdata4 <- postResample(predictions4, test_set2$y)
newdata5 <- postResample(predictions5, test_set2$y)
```

```
Warning in pred - obs: longer object length is not a multiple of shorter object
length

Warning in pred - obs: longer object length is not a multiple of shorter object
length
```

```
newdata6 <- postResample(predictions6, test_set2$y)
newdata7 <- postResample(predictions7, test_set2$y)


#creating a new matrix and binding the RMSE values
results2 <- rbind(newdata1, newdata2, newdata3, newdata4, newdata5, newdata6, newdata7) %>%
  as.data.frame() %>%
  subset(select = RMSE)


#Binding it to the original matrix
results_matrix <- cbind(results_matrix, results2)

#renaming the columns
colnames(results_matrix)[6] <- "k_RMSE"
```

Here is the section with 300 observations instead

```
#creating the tibble/dataframe with 300 observations
tsdf <- tibble(ts_index = c(1:300), r = rnorm(300))

tsdf$y <- 0

#logic that doesnt really work, but sets the y values according to a funciton with 3 lags
tsdf <- tsdf %>%
  mutate(y = ifelse(ts_index < 4, r,
            y = 0.5 + 0.5*lag(y,1) - 0.1*lag(y, 2) + 0.25*lag(y, 3) + r))

#replacing rows 1 through 3 with the associated r values
tsdf[1:3,3] <- tsdf[1:3,2]
```

```
#making 7 different possible Autoregressive models (1, 2, 3) using loocv in caret
train_ct1 <- trainControl(method = "LOOCV")

# 1,2,3 lag structure
lm_model1 <- train(y ~ lag(y) + lag(y, 2) + lag(y, 3),  data = tsdf,
                   na.action = na.pass,
                   trControl = train_ct1,
                   method = "lm")
# 1 2 lag structure
lm_model2 <- train(y ~ lag(y) +  lag(y, 2),  data = tsdf,
                   na.action = na.pass,
                   trControl = train_ct1,
                   method = "lm")
# 1 3 lag structure
lm_model3 <- train(y ~ lag(y) + lag(y, 3),  data = tsdf,
                   na.action = na.pass,
                   trControl = train_ct1,
                   method = "lm")
```

```
lm_model4 <- train(y ~ lag(y, 2) + lag(y, 3),  data = tsdf,
                   na.action = na.pass,
                   trControl = train_ct1,
                   method = "lm")

lm_model5 <- train(y ~ lag(y),  data = tsdf,
                   na.action = na.pass,
                   trControl = train_ct1,
                   method = "lm")

lm_model6 <- train(y ~ lag(y, 2),  data = tsdf,
                   na.action = na.pass,
                   trControl = train_ct1,
                   method = "lm")

lm_model7 <- train(y ~ lag(y, 3),  data = tsdf,
                   na.action = na.pass,
                   trControl = train_ct1,
                   method = "lm")
```

```
#binding all of the results by rows into one matrix
results_matrix2 <- as_tibble(bind_rows(lm_model1$results,
                               lm_model2$results,
                               lm_model3$results,
                               lm_model4$results,
                               lm_model5$results,
                               lm_model6$results,
                               lm_model7$results))
results_matrix2[,1] <- NULL

#Matrix of the AIC values
AIC_matrix <- AIC(lm_model1$finalModel, lm_model2$finalModel, lm_model3$finalModel,
                  lm_model4$finalModel, lm_model5$finalModel, lm_model6$finalModel, lm_model7$finalModel
```

```
Warning in AIC.default(lm_model1$finalModel, lm_model2$finalModel,
lm_model3$finalModel, : models are not all fitted to the same number of
observations
```

```
#Removing the first column
AIC_matrix[,1] <- NULL

BIC_matrix <- BIC(lm_model1$finalModel, lm_model2$finalModel, lm_model3$finalModel,
                  lm_model4$finalModel, lm_model5$finalModel, lm_model6$finalModel, lm_model7$finalModel
```

```
Warning in BIC.default(lm_model1$finalModel, lm_model2$finalModel,
lm_model3$finalModel, : models are not all fitted to the same number of
observations
```

```
#removing the first column
BIC_matrix[,1] <- NULL

#appending them to the results matrix
```

```
results_matrix2['AIC'] <- AIC_matrix
results_matrix2['BIC'] <- BIC_matrix
```

```
#similar cross-validation to above, except using k-folds


#making 7 different possible Autoregressive models (1, 2, 3) using loocv in caret
train_ct2 <- trainControl(method = "cv", number = 10)

# Creating a training set using 80% of the data
inTrain2 <- createDataPartition(y = tsdf$y, p = 0.8, list = FALSE)

#training data
train_set2 <- tsdf[inTrain2, ]
#test data (with the other 20 percent)
test_set2 <- tsdf[-inTrain2, ]




# 1,2,3 lag structure
lm_model_cv1 <- train(y ~ lag(y) + lag(y, 2) + lag(y, 3),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")
# 1 2 lag structure
lm_model_cv2 <- train(y ~ lag(y) +  lag(y, 2),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")
# 1 3 lag structure
lm_model_cv3 <- train(y ~ lag(y) + lag(y, 3),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")

lm_model_cv4 <- train(y ~ lag(y, 2) + lag(y, 3),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")

lm_model_cv5 <- train(y ~ lag(y),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")

lm_model_cv6 <- train(y ~ lag(y, 2),  data = train_set2,
                  na.action = na.pass,
                  trControl = train_ct2,
                  method = "lm")

lm_model_cv7 <- train(y ~ lag(y, 3),  data = train_set2,
                  na.action = na.pass,
```

```
                  trControl = train_ct2,
                  method = "lm")

beepr::beep("coin")
```

```
#Getting the results and appending them to the same matrix as before

models <- c(lm_model_cv1, lm_model_cv2, lm_model_cv3,
            lm_model_cv4, lm_model_cv5, lm_model_cv6, lm_model_cv7)

predictions1 <- predict(lm_model_cv1, test_set2)
predictions2 <- predict(lm_model_cv2, test_set2)
predictions3 <- predict(lm_model_cv3, test_set2)
predictions4 <- predict(lm_model_cv4, test_set2)
predictions5 <- predict(lm_model_cv5, test_set2)
predictions6 <- predict(lm_model_cv6, test_set2)
predictions7 <- predict(lm_model_cv7, test_set2)

#post prediction resampling
model123 <- postResample(predictions1, test_set2$y)
```

```
Warning in pred - obs: longer object length is not a multiple of shorter object
length

Warning in pred - obs: longer object length is not a multiple of shorter object
length
```

```
model12 <- postResample(predictions2, test_set2$y)
```

```
Warning in pred - obs: longer object length is not a multiple of shorter object
length

Warning in pred - obs: longer object length is not a multiple of shorter object
length
```

```
model13 <- postResample(predictions3, test_set2$y)
```

```
Warning in pred - obs: longer object length is not a multiple of shorter object
length

Warning in pred - obs: longer object length is not a multiple of shorter object
length
```

```
model23 <- postResample(predictions4, test_set2$y)
```

```
Warning in pred - obs: longer object length is not a multiple of shorter object
length

Warning in pred - obs: longer object length is not a multiple of shorter object
length
```

```
model1 <- postResample(predictions5, test_set2$y)
```

```
Warning in pred - obs: longer object length is not a multiple of shorter object
length

Warning in pred - obs: longer object length is not a multiple of shorter object
length
```

```
model2 <- postResample(predictions6, test_set2$y)
```

```
Warning in pred - obs: longer object length is not a multiple of shorter object
length

Warning in pred - obs: longer object length is not a multiple of shorter object
length
```

```
model3 <- postResample(predictions7, test_set2$y)
```

```
Warning in pred - obs: longer object length is not a multiple of shorter object
length

Warning in pred - obs: longer object length is not a multiple of shorter object
length
```

```
#creating a new matrix and binding the RMSE values
results3 <- rbind(model123, model12, model13, model23, model1, model2, model3 ) %>%
  as.data.frame() %>%
  subset(select = RMSE)


#Binding it to the original matrix
results_matrix2 <- cbind(results_matrix2, results3)

#renaming the columns
colnames(results_matrix2)[6] <- "k_RMSE"
```

Here is the section with the Model Selection using the four models

```
#importing the data
data <- read_csv("data.csv") %>% na.omit()
```

```
Warning: Missing column names filled in: 'X1' [1]

Parsed with column specification:
cols(
  X1 = col_double(),
  DATE = col_date(format = ""),
  fl_nonfarm = col_double(),
```

```
  fl_lf = col_double(),
  us_epr_25to54 = col_double(),
  fl_bp = col_double()
)
```

```
data[3:6] <- log(data[3:6])

colnames(data)[3:6] <- c("ln_fl_nonfarm", "ln_fl_lf", "ln_us_epr", "ln_fl_bp")
head(data)
```

```
# A tibble: 6 x 6
     X1 DATE       ln_fl_nonfarm ln_fl_lf ln_us_epr ln_fl_bp
  <dbl> <date>             <dbl>    <dbl>     <dbl>    <dbl>
1   589 1988-01-01          8.51     15.6      4.11     9.37
2   590 1988-02-01          8.52     15.6      4.11     9.41
3   591 1988-03-01          8.53     15.6      4.12     9.66
4   592 1988-04-01          8.53     15.6      4.12     9.54
5   593 1988-05-01          8.53     15.6      4.13     9.61
6   594 1988-06-01          8.53     15.6      4.14     9.87
```

First Model

```
#Making the four different models
#creating a new dataframe
#FIRST MODEL
data['d.nonfarm'] <- difference(data$ln_fl_nonfarm, differences = 1)
data['d.nonfarm_lag'] <- difference(data$ln_fl_nonfarm, lag = 12, difference = 1)
data['d.lf_lag'] <- difference(data$ln_fl_lf, lag = 12, differences = 1)
data['d.fl_bp_lag'] <- difference(data$ln_fl_bp, lag = 12, differences = 1)
data['d.usepr'] <- difference(data$ln_us_epr, lag = 12, differences = 1)

months <- yearmonth(data$DATE) %>%
  format(format = "%m") %>%
  as.factor()
data['months'] <- months
```

```
#LOOCV
model_1 <- train(d.nonfarm ~ d.nonfarm_lag + d.lf_lag + d.fl_bp_lag + d.usepr + months + DATE,
                 na.action = na.exclude,
                 data = data,
                 trControl = trainControl(method = "LOOCV"),
                 method = "lm")

#writing results to final table
final_results <- rbind(model_1$results)
```

Second Model

```r
#SECOND MODEL
#changing the lag structure
data['d.lf_lag'] <- difference(data$ln_fl_lf, lag = 2, differences = 1)
data['d.fl_bp_lag'] <- difference(data$ln_fl_bp, lag = 2, differences = 1)
data['d.usepr'] <- difference(data$ln_us_epr, lag = 2, differences = 1)


model_2 <- train(d.nonfarm ~ d.nonfarm_lag + d.lf_lag + d.fl_bp_lag + d.usepr + months + DATE,
                 na.action = na.exclude,
                 data = data,
                 trControl = trainControl(method = "LOOCV"),
                 method = "lm")

final_results <- rbind(final_results, model_2$results)
```

Third Model

```r
#THIRD MODEL
data['d.lf_lag'] <- difference(data$ln_fl_lf, lag = 2, differences = 1)
data['d.lf_lag_12'] <- difference(data$ln_fl_lf, lag = 12, differences = 1)
data['d.fl_bp_lag'] <- difference(data$ln_fl_bp, lag = 2, differences = 1)
data['d.fl_bp_12'] <- difference(data$ln_fl_bp, lag = 12, differences = 1)
data['d.usepr'] <- difference(data$ln_us_epr, lag = 2, differences = 1)
data['d.usepr_12'] <- difference(data$ln_us_epr, lag = 12, differences = 1)

#LOOCV
model_3 <- train(d.nonfarm ~ d.nonfarm_lag + d.lf_lag + d.lf_lag_12 + d.fl_bp_lag + d.fl_bp_12 + d.usepr
                 na.action = na.exclude,
                 data = data,
                 trControl = trainControl(method = "LOOCV"),
                 method = "lm")

#writing results to final table

final_results <- rbind(final_results, model_3$results)
```

Fourth Model

```r
#FOURTH MODEL
data['d.lf_lag'] <- difference(data$ln_fl_lf, lag = 2, differences = 1)
data['d.lf_lag_12'] <- difference(data$ln_fl_lf, lag = 12, differences = 1)
data['d.lf_lag_24'] <- difference(data$ln_fl_lf, lag = 24, differences = 1)
data['d.fl_bp_lag'] <- difference(data$ln_fl_bp, lag = 2, differences = 1)
data['d.fl_bp_12'] <- difference(data$ln_fl_bp, lag = 12, differences = 1)
data['d.fl_bp_24'] <- difference(data$ln_fl_bp, lag = 24, differences = 1)
data['d.usepr'] <- difference(data$ln_us_epr, lag = 2, differences = 1)
data['d.usepr_12'] <- difference(data$ln_us_epr, lag = 12, differences = 1)
```

```
data['d.usepr_24'] <- difference(data$ln_us_epr, lag = 24, differences = 1)

#LOOCV
model_4 <- train(d.nonfarm ~ d.nonfarm_lag + d.lf_lag + d.lf_lag_12 + d.lf_lag_24 + d.fl_bp_lag +
                   d.fl_bp_12 + d.fl_bp_24 + d.usepr + d.usepr_12 + d.usepr_24 + months,
                 na.action = na.exclude,
                 data = data,
                 trControl = trainControl(method = "LOOCV"),
                 method = "lm")

#writing results to final table

final_results <- rbind(final_results, model_4$results)
```

```
final_results[,1] <- NULL

#Matrix of the AIC values
AIC_final <- AIC(model_1$finalModel, model_2$finalModel, model_3$finalModel,
                 model_4$finalModel)
```

```
Warning in AIC.default(model_1$finalModel, model_2$finalModel,
model_3$finalModel, : models are not all fitted to the same number of
observations
```

```
#Removing the first column
AIC_final[,1] <- NULL

BIC_final <- BIC(model_1$finalModel, model_2$finalModel, model_3$finalModel,
                 model_4$finalModel)
```

```
Warning in BIC.default(model_1$finalModel, model_2$finalModel,
model_3$finalModel, : models are not all fitted to the same number of
observations
```

```
#removing the first column
BIC_final[,1] <- NULL

#appending them to the results matrix
final_results['AIC'] <- AIC_final
final_results['BIC'] <- BIC_final
beepr::beep("coin")

kable(final_results, format = "latex") %>%
  kable_styling(position = "center", latex_options = "striped")
```

RMSE	Rsquared	MAE	AIC	BIC
0.0047814	0.7733409	0.0035833	-2910.616	-2840.124
0.0047430	0.7769618	0.0035699	-2917.164	-2846.672
0.0047580	0.7755934	0.0035853	-2915.047	-2832.807
0.0047374	0.7768743	0.0035409	-2822.839	-2733.522