HW3: Model Selection

Angel Sarmiento

2/17/2020

Introduction

Model selection involves the choosing of ideal models that will be useful for either inference, prediction, or both. Predictive models normally involve correlation as a primary relation between predictor and response variables. Inference models can be used to develop and understand causal relationships between variables. This investigation will involve several criterion for model selection in the interest of deciding on the best model for forecasting. The decision criterion specifically will involve numerous AIC and BIC metrics, Cross-validation techniques (LOOCV and k-folds), standardized tests, and contextual knowledge.

Information Criteria and Cross-validation

For the first set of model selection tests, an integrated of order three Autoregressive model, AR(3), will be tested with 30 observations and normalized residuals. This model has the form:

$$y = 0.5 + 0.5y_{t-1} - 0.1y_{t-2} + 0.25y_{t-3} + r$$

As an AR(3) model, there are 7 possible different lag structures that can be tested to ensure a good model is assumed by our random variables. Ideally, the best model will be a model that predicts a lag structure of 3 lags since that is what the true model is. The seven models will be validated using Leave-One-Out Cross Validation and their respective lag structures. These models are then compared using the corresponding AIC, BIC, and Root Mean Squared Errors. All of these metrics can be used to infer that a model is fit better by looking for lower values compared to others in a table.

After this, kfolds cross-validation is done with 10 folds to see if a model done with k-folds performs better than our LOOCV model. The k-folds cross-validation is also evaluated on a training-testing set split where 20% of the data is set aside for testing the parameter estimation of our models. All of these models are shown in the table below.

	RMSE	Rsquared	MAE	AIC	BIC	k_RMSE
lag_123	0.9629017	0.3053385	0.8292584	77.22099	83.70018	0.8550203
lag_12	0.9286814	0.2133180	0.8013109	77.57986	82.90868	0.9732725
lag_13	0.9361855	0.5618618	0.8169802	75.76747	80.95082	0.8617535
lag_23	0.9344129	0.2060845	0.8003244	75.23032	80.41366	0.8564205
lag_1	0.8933342	0.7944672	0.7774979	78.15567	82.25755	0.7616703
lag_2	0.9034111	0.1099990	0.7760394	75.58005	79.57667	0.8059060
lag_3	0.9083645	0.5059412	0.7887204	73.77595	77.66346	0.8541914

Table 1. 7 Autoregressive models comparison.

According to the performance of the model, there is a lot to unpack. The LOOCV models with lower lag amounts seem to perform better than their counterparts with more lags included. These models have lower

AIC and BIC, as well as lower RMSE values. Specifically, it looks as though models that include the first lag have lower RMSE values. The model with 1 lag also has a lower RMSE in the k-folds model. Assuming nothing were known about the approximate lag structure of the data, the fifth model (the model with only the first lag) would be the model of choice. This is especially due to the ambiguous comparison given by the AIC and BIC when taken into account alone. Seeing the k-folds performance on the first and fifth models shows a bit of a better picture when compared to LOOCV. The first model performs better with k-folds but still false short in comparison to model 5. The fit will be further demonstrated with the plot in figure 1 below, where four plots of the best fitting models are shown.

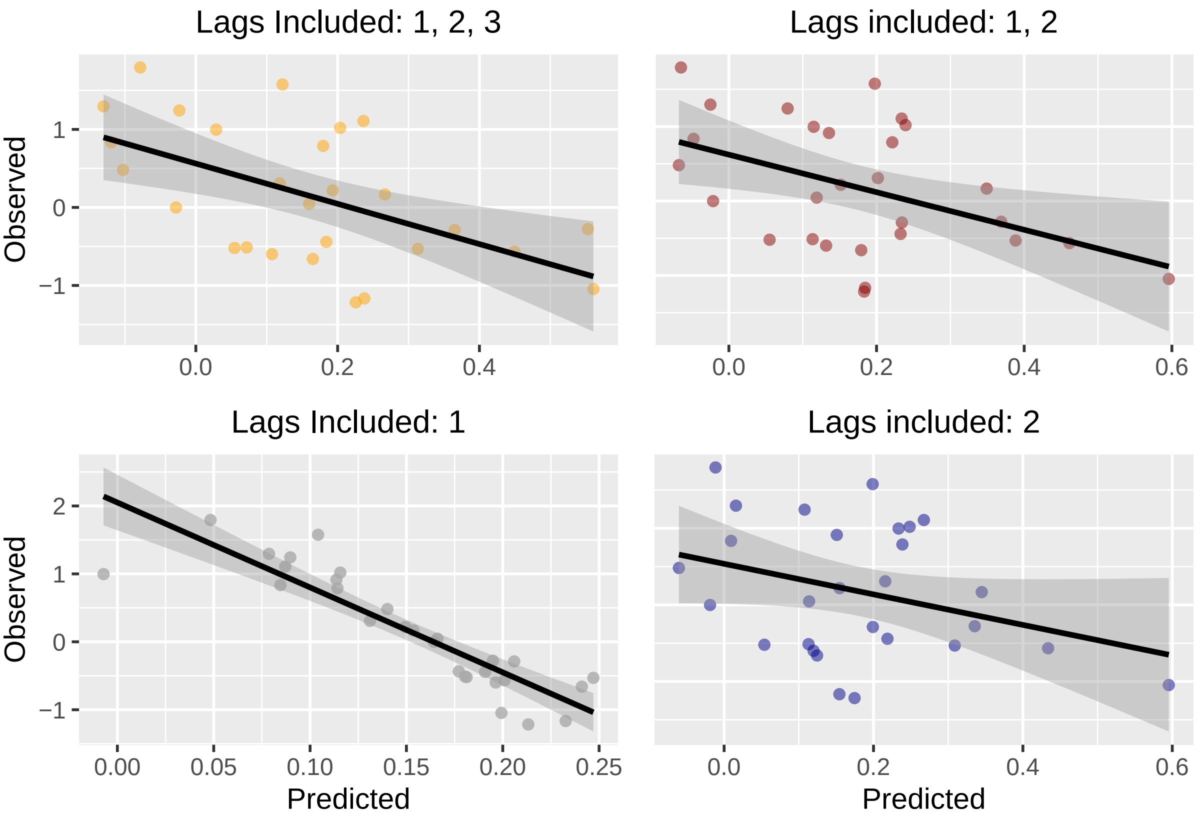

Figure 1. LOOCV model selection for top 4 best lag structures

These models are capable of representing the data fairly well, but they might do better if there were more data to fit to. Next, the observation count will increase to 300 and the same process as before will be done again. This time the k-folds cross-validation results will be graphed as well to show how well they fit. Developing these models as before:

	RMSE	Rsquared	MAE	AIC	BIC	k_RMSE
model123	1.004916	0.0020868	0.8065024	847.9117	866.3803	1.0134128
model12	1.007868	0.0047858	0.8082413	852.4037	867.1920	0.9533517
model13	1.003349	0.0022251	0.8041624	847.1763	861.9512	0.9935075
model23	1.002882	0.0031913	0.8056776	846.4457	861.2206	1.0140926
model1	1.004297	0.0158576	0.8028300	853.3115	864.4129	1.0685084
model2	1.005526	0.0027379	0.8076101	850.7809	861.8722	0.9526619
model3	1.001418	0.0034349	0.8024487	845.7798	856.8610	0.9942850

Table 2. Model results for 300 observations using LOOCV and k-folds CV

From these results it seems that most of the LOOCV models performed similarly in terms of RMSE, but differed in their AIC and BIC results. Looking at these information criterion shows that *model3*, *model23*, and *model13* are the top performers here. Compared to the k-folds, most of the LOOCV models get outperformed. The K-folds examples are the only models to yield less than 1.0 RMSE. Figures 2 and 3 show the plots of the top LOOCV models and the top K-folds models respectively.

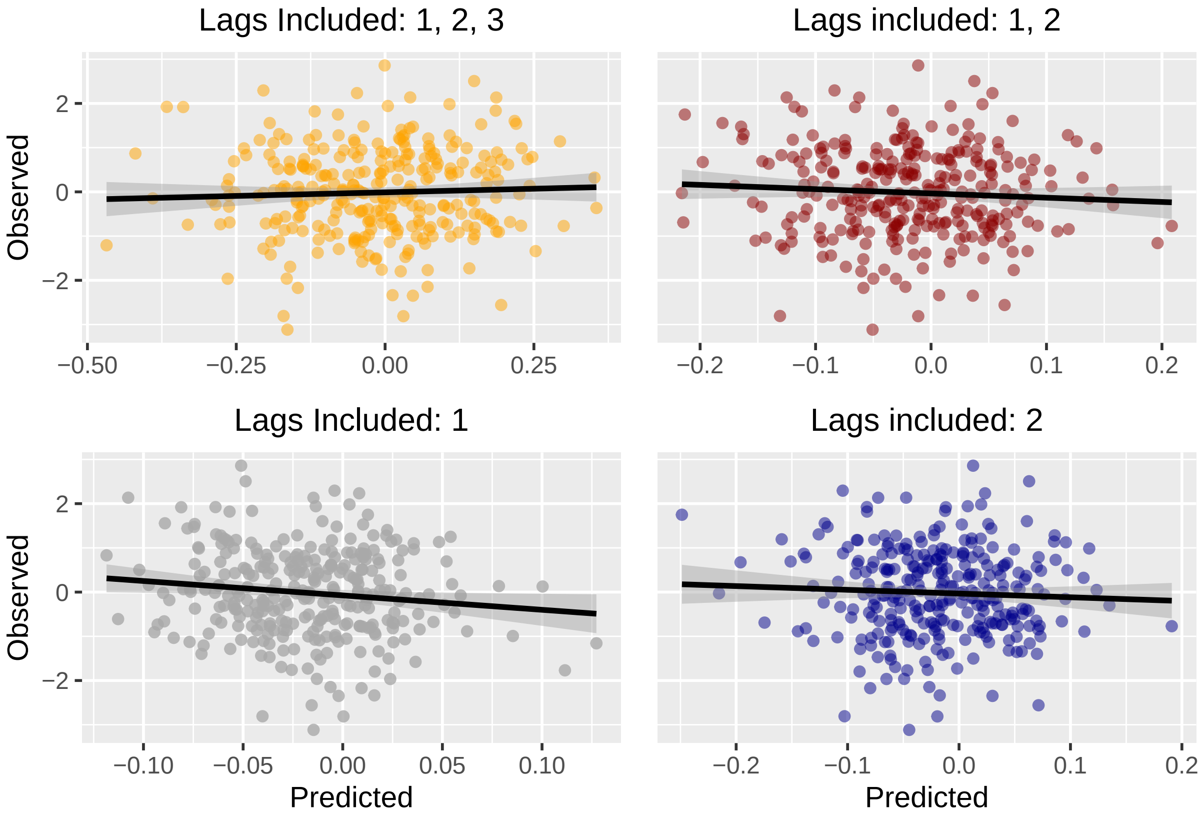

Figure 2. Model fits for the top performing models. LOOCV.

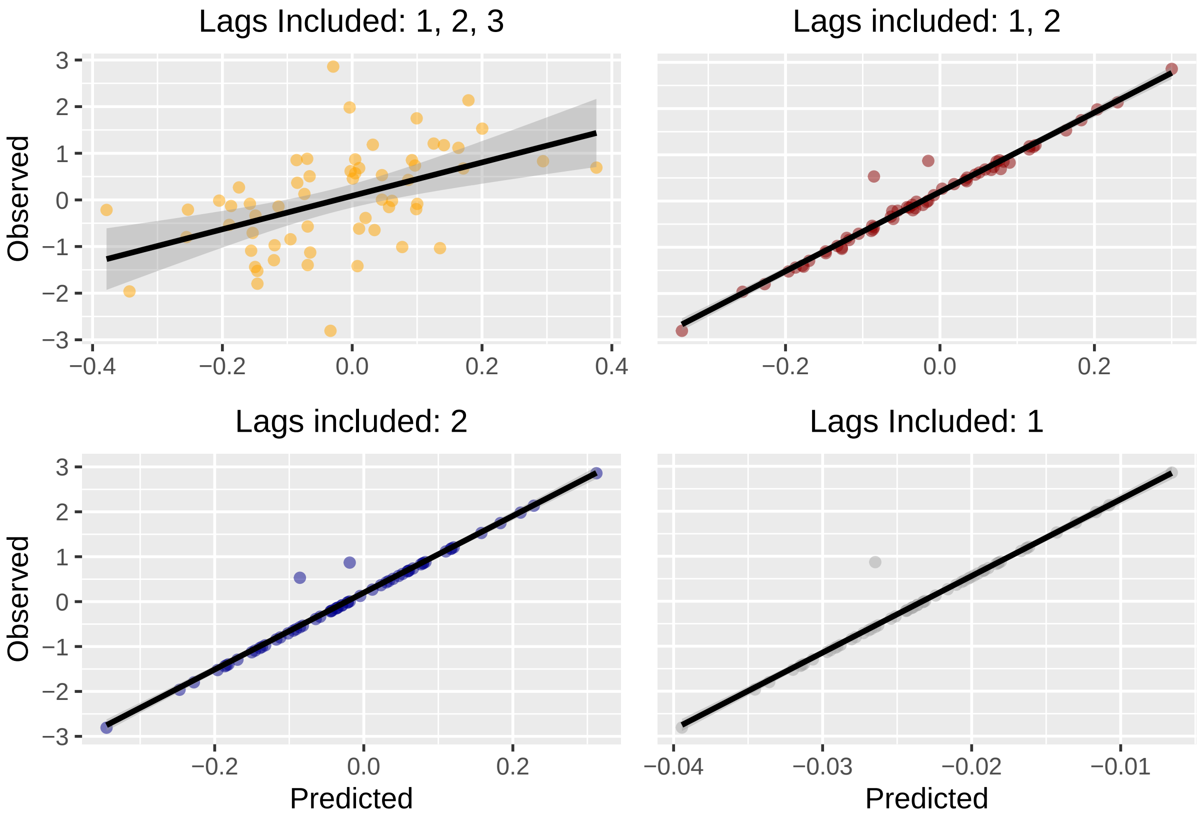

Figure 3. Model fits for the top performing models. K-folds.

These plots show that, in this simulation, k-folds generally yields a better fit on the data and is a fairly good predictor of data outside the training data. Since both selection processes ($n = 30$ and $n = 300$) showed similar results, it would be fair to conclude that k-fold will perform better on the data for prediction and should thus be chosen.

Model Selection for Nonfarm Employment

In previous investigations, data collected from FRED was used to create and explore numerous different models and model types. This section will be devoted to developing an idea of the best model to select for the prediction of nonfarm employment. The data will be imported below and cross validation will be used on several model to see which one is the best choice. It is also helpful to know that the data will be logs of each of the variables and monthly indicators will be added for the models.

DATE	ln_fl_nonfarm	ln_fl_lf	ln_us_epr	ln_fl_bp
1988-01-01	8.506900	15.59221	4.110874	9.374668
1988-02-01	8.518872	15.59531	4.112512	9.412710
1988-03-01	8.529398	15.59868	4.115780	9.662753
1988-04-01	8.526074	15.60442	4.123903	9.542159
1988-05-01	8.526965	15.61056	4.127134	9.610324
1988-06-01	8.528588	15.62448	4.143135	9.865526

Table 3. Data

The four different models are as follows:

$$\Delta y_t = \beta_0 + \sum_{(a,l)=0}^{12} \beta_a L_l \Delta y_{t-1} + \sum_{b,k}^{12} \beta_b L_k \Delta X_{lf,t} + \sum_{c,k}^{12} \beta_c L_k \Delta X_{bp,t} + \sum_{d,k}^{12} \beta_d L_k \Delta X_{epr,t} + \beta_e X_m + DATE + \varepsilon_t \quad (1)$$

Where $DATE$ is a time trend, $k = 0, 1, 2, 3, ...12$, $l = 1, 2, 3, ...12$, m is the month from $1, 2, 3, ...12$, and L is the lag.

$$\Delta y_t = \beta_0 + \sum_{(a,l)=0}^{12} \beta_a L_l \Delta y_{t-1} + \sum_{b,k}^{2} \beta_b L_k \Delta X_{lf,t} + \sum_{c,k}^{2} \beta_c L_k \Delta X_{bp,t} + \sum_{d,k}^{2} \beta_d L_k \Delta X_{epr,t} + \beta_e X_m + DATE + \varepsilon_t \quad (2)$$

Where $DATE$ is a time trend, $k = 0, 1, 2$, $l = 1, 2, 3, ...12$, m is the month from $1, 2, 3, ...12$, and L_k is the lag at value k or l.

$$\Delta y_t = \beta_0 + \sum_{(a,l)=0}^{12} \beta_a L_l \Delta y_{t-1} + \sum_{b,k}^{2,12} \beta_b L_k \Delta X_{lf,t} + \sum_{c,k}^{2,12} \beta_c L_k \Delta X_{bp,t} + \sum_{d,k}^{2,12} \beta_d L_k \Delta X_{epr,t} + \beta_e X_m + DATE + \varepsilon_t \quad (3)$$

Where $DATE$ is a time trend, $k = 0, 1, 2 \ or \ 12$, $l = 1, 2, 3, ...12$, m is the month from $1, 2, 3, ...12$, and L_k is the lag at value k or l.

$$\Delta y_t = \beta_0 + \sum_{(a,l)=0}^{12,24} \beta_a L_l \Delta y_{t-1} + \sum_{b,k}^{2,12,24} \beta_b L_k \Delta X_{lf,t} + \sum_{c,k}^{2,12,24} \beta_c L_k \Delta X_{bp,t} + \sum_{d,k}^{2,12,24} \beta_d L_k \Delta X_{epr,t} + \beta_e X_m + \varepsilon_t \quad (4)$$

	RMSE	Rsquared	MAE	AIC	BIC
Model 1	0.0047814	0.7733409	0.0035833	-2910.616	-2840.124
Model 2	0.0047430	0.7769618	0.0035699	-2917.164	-2846.672
Model 3	0.0047580	0.7755934	0.0035853	-2915.047	-2832.807
Model 4	0.0047489	0.7758198	0.0035472	-2821.398	-2728.198

Table 3. Model Comparison for Nonfarm employment using LOOCV

Given the results of the different model training, the best model would be model number 2. Model number 2 has the lowest RMSE, the lowest AIC, explains the most variance and is the most parsimonious of the models. Prediction with a model like model 2 would likely yield promising results especially when compared to the other models. The least parsimonious model, model 4, has the highest AIC and BIC, as well as a comparable RMSE to model 2. However this model is very complex and could most likely have inaccurate paramater estimates due to the sheer amount of lags included.

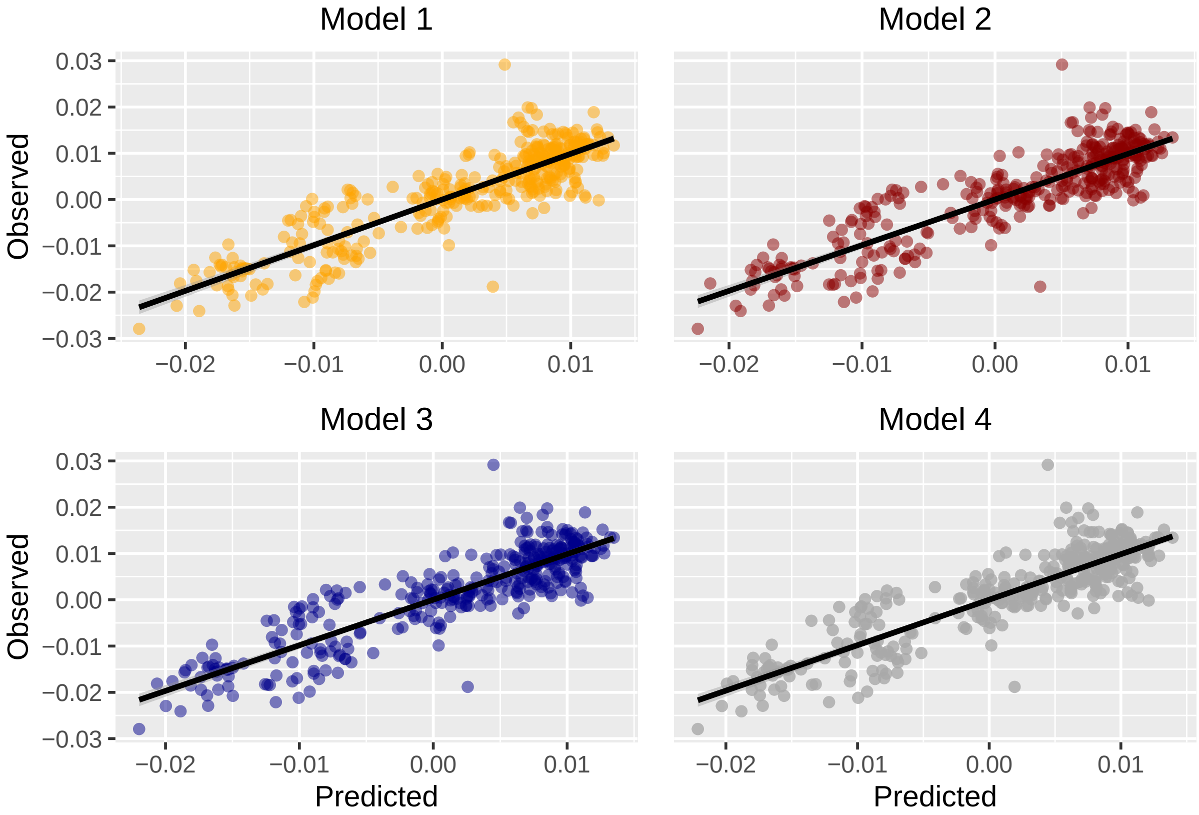

Figure 4. All four models plotted in comparison with one another.

Conclusion

From intuition, it makes sense why a model that includes a full cycle of lags (12 for 12 months) might represent a dynamically complete model. Upon further thought however, it seems clear that variables like prime age employment rate and labor force might only affect nonfarm employment in shorter timespans. It is commonly heard that a majority of people enter the job market and get hired within six months. Including a full cycle in spite of this most likely just results in more error in estimating model parameters. For prediction purposes, this does not matter so much since accurate predictions do not require the most accurate coefficients. However, the inferential capability of a model is severely impacted by the presence of unneeded terms that explain little variance. This is why I believe model 2, the most parsimonious and well-performing of the models is the best choice.

Appendix A: Script

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

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

#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,
```

```r
                   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$finalMode
#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$finalMode

#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, ]
```

```r
# 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")

#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
```

```r
lag_123 <- postResample(predictions1, test_set2$y)
lag_12 <- postResample(predictions2, test_set2$y)
lag_13 <- postResample(predictions3, test_set2$y)
lag_23 <- postResample(predictions4, test_set2$y)
lag_1 <- postResample(predictions5, test_set2$y)
lag_2 <- postResample(predictions6, test_set2$y)
lag_3 <- postResample(predictions7, test_set2$y)


#creating a new matrix and binding the RMSE values
results2 <- rbind(lag_123, lag_12, lag_13, lag_23, lag_1, lag_2, lag_3 ) %>%
  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"

kable(results_matrix, format = "latex") %>%
  kable_styling(position = "center", latex_options = c("striped", "HOLD_position"))

require(ggplot2)

results_mods <- data.frame(cbind(predictions = lm_model1$pred[,1], observations = lm_model1$pred[,2]))

model_plot1 <- ggplot(results_mods, aes(predictions, observations)) +
  geom_point(color = "orange", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Lags Included: 1, 2, 3") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))


results_mods2 <- data.frame(cbind(predictions = lm_model2$pred[,1], observations = lm_model2$pred[,2]))

model_plot2 <- ggplot(results_mods2, aes(predictions, observations)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Lags included: 1, 2") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))

results_mods6 <- data.frame(cbind(predictions = lm_model6$pred[,1], observations = lm_model6$pred[,2]))

model_plot6 <- ggplot(results_mods6, aes(predictions, observations)) +
  geom_point(color = "darkblue", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Lags included: 2") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))

results_mods5 <- data.frame(cbind(predictions = lm_model5$pred[,1], observations = lm_model5$pred[,2]))

model_plot5 <- ggplot(results_mods5, aes(predictions, observations)) +
  geom_point(color = "darkgrey", alpha = 0.8) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 5') +
  ggtitle("Lags Included: 1") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black", size=12, hjust = 0.5))



require(patchwork)
patchwork = model_plot1 + model_plot2 + model_plot5 + model_plot6

patchwork[[1]] = patchwork[[1]] + theme(axis.title.x = element_blank())

patchwork[[2]] = patchwork[[2]] + theme(axis.title.x = element_blank())

patchwork[[2]] = patchwork[[2]] + theme(axis.text.y = element_blank(),
                                        axis.ticks.y = element_blank(),
                                        axis.title.y = element_blank() )

patchwork[[4]] = patchwork[[4]] + theme(axis.text.y = element_blank(),
                                        axis.ticks.y = element_blank(),
                                        axis.title.y = element_blank() )

patchwork


#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
#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
```

```
#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,
```

```r
                      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")

#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$y)
model12 <- postResample(predictions2, test_set2$y)
model13 <- postResample(predictions3, test_set2$y)
model23 <- postResample(predictions4, test_set2$y)
model1 <- postResample(predictions5, test_set2$y)
model2 <- postResample(predictions6, test_set2$y)
model3 <- postResample(predictions7, test_set2$y)


#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"

kable(results_matrix2, format = "latex") %>%
  kable_styling(position = "center", latex_options = c("striped", "HOLD_position"))


#PLOTTING LOOCV FOR 300 OBS
require(ggplot2)

results_mods <- data.frame(cbind(predictions = lm_model1$pred[,1], observations = lm_model1$pred[,2]))

model_plot1 <- ggplot(results_mods, aes(predictions, observations)) +
  geom_point(color = "orange", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Lags Included: 1, 2, 3") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))


results_mods2 <- data.frame(cbind(predictions = lm_model2$pred[,1], observations = lm_model2$pred[,2]))

model_plot2 <- ggplot(results_mods2, aes(predictions, observations)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Lags included: 1, 2") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))

results_mods6 <- data.frame(cbind(predictions = lm_model6$pred[,1], observations = lm_model6$pred[,2]))

model_plot6 <- ggplot(results_mods6, aes(predictions, observations)) +
  geom_point(color = "darkblue", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Lags included: 2") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))

results_mods5 <- data.frame(cbind(predictions = lm_model5$pred[,1], observations = lm_model5$pred[,2]))

model_plot5 <- ggplot(results_mods5, aes(predictions, observations)) +
  geom_point(color = "darkgrey", alpha = 0.8) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 5') +
  ggtitle("Lags Included: 1") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black", size=12, hjust = 0.5))


#setting up a grid of all of the plots
require(patchwork)
patchwork = model_plot1 + model_plot2 + model_plot5 + model_plot6

patchwork[[1]] = patchwork[[1]] + theme(axis.title.x = element_blank())

patchwork[[2]] = patchwork[[2]] + theme(axis.title.x = element_blank())

patchwork[[2]] = patchwork[[2]] + theme(axis.text.y = element_blank(),
                                        axis.ticks.y = element_blank(),
                                        axis.title.y = element_blank() )

patchwork[[4]] = patchwork[[4]] + theme(axis.text.y = element_blank(),
                                        axis.ticks.y = element_blank(),
                                        axis.title.y = element_blank() )

patchwork


#PLOTTING KFOLDS FOR 300 OBS

results_modscv1 <- data.frame(cbind(predictions = predictions1, observations = test_set2$y))

model_plotcv1 <- ggplot(results_modscv1, aes(predictions, observations)) +
  geom_point(color = "orange", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Lags Included: 1, 2, 3") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))


results_modscv2 <- data.frame(cbind(predictions = predictions2, observations = test_set2$y))

model_plotcv2 <- ggplot(results_modscv2, aes(predictions, observations)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Lags included: 1, 2") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))

results_modscv6 <- data.frame(cbind(predictions = predictions6, observations = test_set2$y))

model_plotcv6 <- ggplot(results_modscv6, aes(predictions, observations)) +
  geom_point(color = "darkblue", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Lags included: 2") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))

results_modscv5 <- data.frame(cbind(predictions = predictions5, observations = test_set2$y))

model_plotcv5 <- ggplot(results_modscv5, aes(predictions, observations)) +
  geom_point(color = "darkgrey", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 5') +
  ggtitle("Lags Included: 1") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))



#setting up a grid of all of the plots
patchwork2 = model_plotcv1 + model_plotcv2 + model_plotcv6 + model_plotcv5
```

```r
patchwork2[[1]] = patchwork2[[1]] + theme(axis.title.x = element_blank())

patchwork2[[2]] = patchwork2[[2]] + theme(axis.title.x = element_blank())

patchwork2[[2]] = patchwork2[[2]] + theme(axis.text.y = element_blank(),
                                          axis.ticks.y = element_blank(),
                                          axis.title.y = element_blank() )

patchwork2[[4]] = patchwork2[[4]] + theme(axis.text.y = element_blank(),
                                          axis.ticks.y = element_blank(),
                                          axis.title.y = element_blank() )

patchwork2



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

data[3:6] <- log(data[3:6])
data[,1] <- NULL

colnames(data)[2:5] <- c("ln_fl_nonfarm", "ln_fl_lf", "ln_us_epr", "ln_fl_bp")
kable(head(data), format = "latex") %>%
  kable_styling(position = "center", latex_options = "striped")


#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)
```

```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
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.usep
                 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
data['d.nonfarm_lag_24'] <- difference(data$ln_fl_nonfarm, lag = 24, difference = 1)
data['d.lf_lag'] <- difference(data$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.nonfarm_lag_24 + 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)
#Removing the first column
AIC_final[,1] <- NULL

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

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

#appending them to the results matrix
final_results['AIC'] <- AIC_final
final_results['BIC'] <- BIC_final

#adding rownames
row.names(final_results) <- c("Model 1", "Model 2", "Model 3", "Model 4")

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


results_final <- data.frame(cbind(predictions = model_1$pred[,1], observations = model_1$pred[,2]))

model_final_1 <- ggplot(results_final, aes(predictions, observations)) +
  geom_point(color = "orange", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Model 1") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))


results_final_2 <- data.frame(cbind(predictions = model_2$pred[,1], observations = model_2$pred[,2]))

model_final_2 <- ggplot(results_final_2, aes(predictions, observations)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Model 2") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))

results_final_3 <- data.frame(cbind(predictions = model_3$pred[,1], observations = model_3$pred[,2]))

model_final_3 <- ggplot(results_final_3, aes(predictions, observations)) +
  geom_point(color = "darkblue", alpha = 0.5) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 1') +
  ggtitle("Model 3") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black",size=12,hjust = 0.5))

results_final_4 <- data.frame(cbind(predictions = model_4$pred[,1], observations = model_4$pred[,2]))

model_final_4 <- ggplot(results_final_4, aes(predictions, observations)) +
  geom_point(color = "darkgrey", alpha = 0.8) +
  geom_smooth(method = "lm", colour = "black")+ ggtitle('Model 5') +
  ggtitle("Model 4") +
  xlab("Predicted") +
  ylab("Observed") +
  theme(plot.title = element_text(color="black", size=12, hjust = 0.5))



require(patchwork)
patchwork = model_final_1 + model_final_2 + model_final_3 + model_final_4

patchwork[[1]] = patchwork[[1]] + theme(axis.title.x = element_blank())

patchwork[[2]] = patchwork[[2]] + theme(axis.title.x = element_blank())

patchwork[[2]] = patchwork[[2]] + theme(axis.text.y = element_blank(),
                                        axis.ticks.y = element_blank(),
                                        axis.title.y = element_blank() )

patchwork[[4]] = patchwork[[4]] + theme(axis.text.y = element_blank(),
                                        axis.ticks.y = element_blank(),
                                        axis.title.y = element_blank() )

patchwork

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