Lab 3 and 4: Autoregressive Model Forecasting and RMSE

Griffin Shelor

2024-02-06

Since we used the same dataset and the models created in Lab 3, I used the same qmd document for Lab 4 that I used for Lab 3, and added the plots and RMSE code at the end.

```
## loading packages
library(pacman)
pacman::p_load(tidyverse, here, tseries, astsa, forecast, stlplus, fpp)
## reading in data
portal <- read.csv(here("UNR-EcoForecast-main", "data",
 ↪  "portal_timeseries.csv"))
```

Question 1

```
## setting number of rows and subsetting data to use for prediction later
rows <- nrow(portal)
observed <- portal[1:(rows - 10),]
observed_rows <- nrow(observed)

## fitting linear model
set.seed(802)
model <- glm(NDVI[-1] ~ NDVI[-(observed_rows)], data = observed)
model
summary(model)
plot(model)
```

Question 2

```
## creating function to manually predict future NDVI
forecast_func <- function(B0, B1, data, t) {
  NDVI_predictions <- B0 + B1 * data[t-1,2]
  return(NDVI_predictions)
}
NDVI_ten <- NULL
## calling forecast_func to predict next 10 months NDVI values
for (i in 263:273) {
  ndvi_pred <- forecast_func(model$coefficients[1], model$coefficients[2],
 ↪  portal,  i)
  NDVI_ten <- rbind(NDVI_ten, ndvi_pred)
}

portal$NDVI[264:273]
NDVI_ten
```

Question 3

```
## repeating questions 1 and 2 but with rain as a covariate
## fitting linear model
set.seed(802)
model_rain <- glm(NDVI[-1] ~ NDVI[-(observed_rows)] + rain[-(observed_rows)],
 ↪  data = observed)
model_rain
summary(model_rain)
plot(model_rain)
```

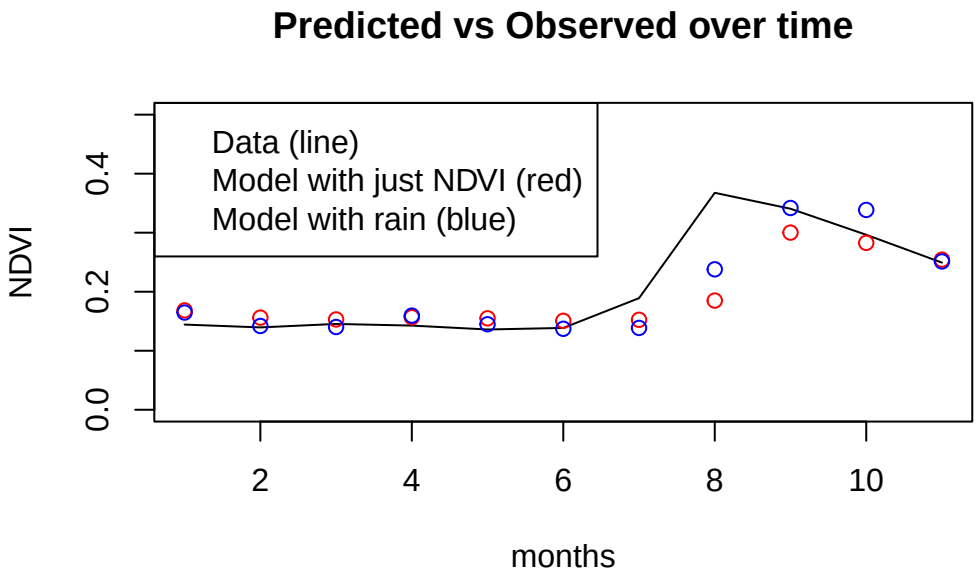
```
## creating function to manually predict future NDVI but this time including
 ↪  rain
forecast_func_rain <- function(B0, B1, B2, data, t) {
  NDVI_predictions_rain <- B0 + (B1 * data[t-1,2]) + (B2 * data[t-1,3])
  return(NDVI_predictions_rain)
}
NDVI_ten_rain <- NULL
## calling forecast_func to predict next 10 months NDVI values
for (i in 263:273) {
```

```
  ndvi_pred_rain <- forecast_func_rain(model_rain$coefficients[1],
 ↪  model_rain$coefficients[2], model_rain$coefficients[3], portal,  i)
  NDVI_ten_rain <- rbind(NDVI_ten_rain, ndvi_pred_rain)
}

portal$NDVI[264:273]
NDVI_ten_rain <- c(NDVI_ten_rain)
```

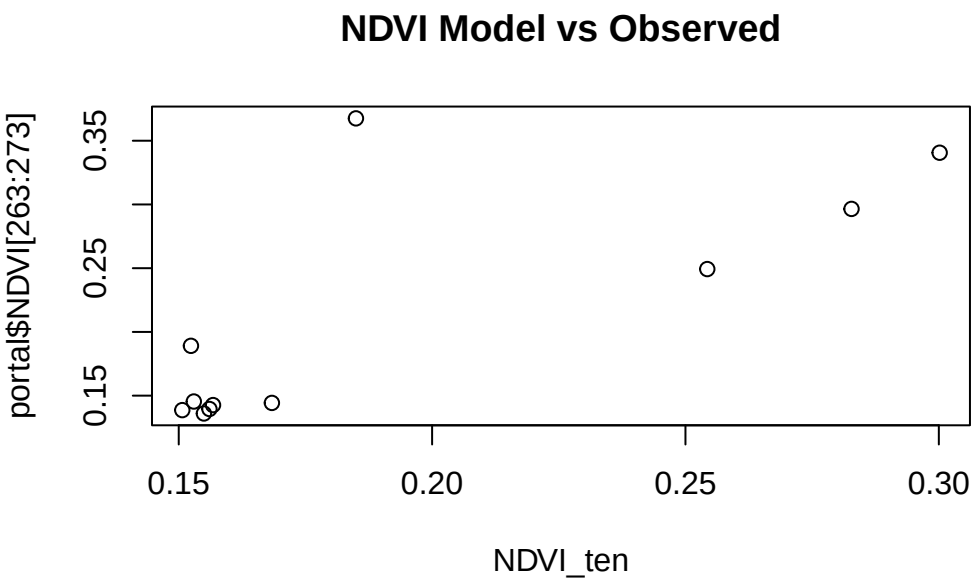
RMSE function

```
## plotting forecasts
plot(1:11, portal$NDVI[263:273], type = "l",col = "black", ylim = c(0,0.5),
 ↪  xlab = "months", ylab = "NDVI", main = "Predicted vs Observed over time")
points(NDVI_ten, col = 'red')
points(NDVI_ten_rain, col = 'blue')
legend("topleft", legend = c("Data (line)", "Model with just NDVI (red)",
 ↪  "Model with rain (blue)"), col = 1:3)
```

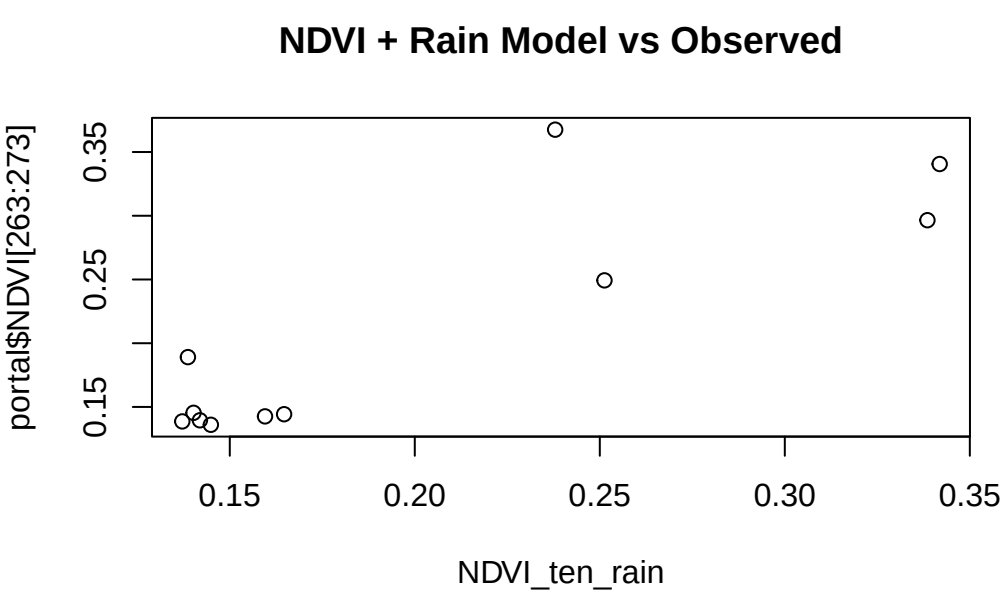



```
## plotting forecasts
# plot(1:11, NDVI_ten_rain, type = "l", ylim = c(0,0.5))
# points(portal$NDVI[263:273], col = 'red')
```

```r
plot(NDVI_ten, portal$NDVI[263:273], main = "NDVI Model vs Observed")
```




```r
plot(NDVI_ten_rain, portal$NDVI[263:273], main = "NDVI + Rain Model vs
↪  Observed")
```

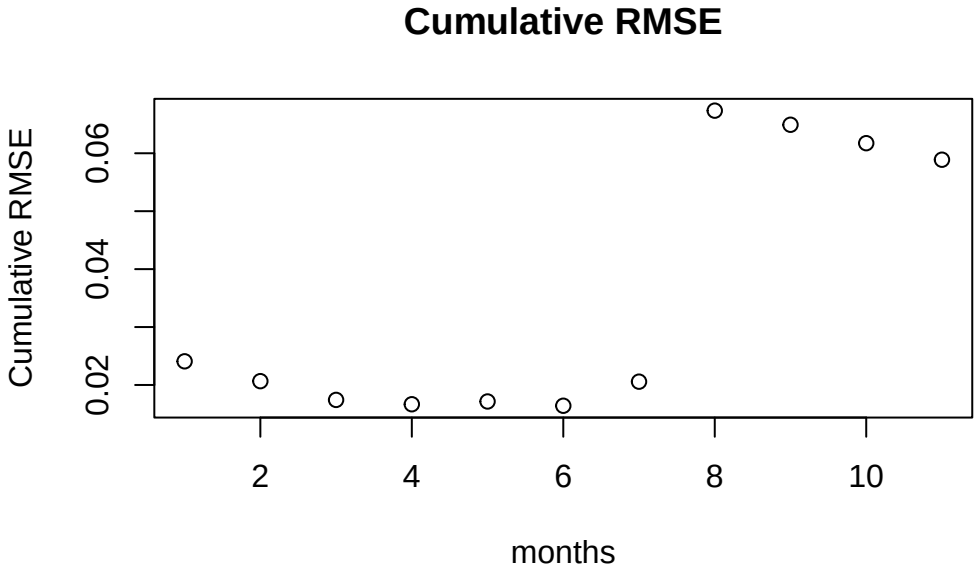


```r
## creating RMSE function
RMSE_func <- function(ypred, yobs) {
  SS <- sum((ypred - yobs)^2)
  RMSE_out <- sqrt(SS / length(yobs))
  return(RMSE_out)
}

obs <- portal$NDVI[(nrow(portal)-10):nrow(portal)]
## converting NDVI_ten to a vector
NDVI_ten <- c(NDVI_ten)
## creating empty RMSE_vals vector
RMSE_vals <- NULL
for (i in 1:11) {
  RMSE_vals <- append(RMSE_vals, RMSE_func(NDVI_ten[1:i], obs[1:i]))
}
RMSE_vals
```

```
 [1] 0.02408487 0.02065801 0.01742499 0.01667065 0.01715327 0.01641342
 [7] 0.02057096 0.06735679 0.06491925 0.06174075 0.05888684
```

```r
plot(1:11, RMSE_vals, xlab = "months", ylab = "Cumulative RMSE", main =
 ↪  "Cumulative RMSE")
```



Based on my plot of RMSE, the forecasts diverge at approximately month 8. This also fits with my plot of both models compared to the observed data, where the widest gap between

predicted NDVI and observed NDVI occurs. This could be because the NDVI that month was an outlier, or maybe there was an outlier in one of the predictor variables such as rain which could not properly account for the variation in NDVI present around this time. It is also possible that my model could be overfitted to the dataset used to "train" it and thus is not as suitable as I would like for test datasets.