ARMA-X Analysis

Contents

Data

```r
# 1. Load Political Social Media

#contains posts from Twitter & TruthSocial
social <- read.csv(here("data/mothership", "social.csv"))

social_hourly <- read.csv(here("data/mothership", "socialhourly.csv"))


# 2. Load Financial

#S&P500
SPY <- read.csv(here("data/mothership", "SPY.csv"))

#STOXX50
VGK <- read.csv(here("data/mothership", "VGK.csv"))

#CSI 300 (China)
ASHR <- read.csv(here("data/mothership", "ASHR.CSV"))
```

```r
#make posixct
SPY$timestamp = as.POSIXct(SPY$timestamp,format = "%Y-%m-%d %H:%M:%S")
VGK$timestamp = as.POSIXct(VGK$timestamp,format = "%Y-%m-%d %H:%M:%S")
ASHR$timestamp = as.POSIXct(ASHR$timestamp,format = "%Y-%m-%d %H:%M:%S")
social$timestamp = as.POSIXct(social$timestamp,format = "%Y-%m-%d %H:%M:%S")
social_hourly$timestamp = as.POSIXct(social_hourly$timestamp,format = "%Y-%m-%d %H:%M:%S")
```

Selecting Relevant Data

Volatility

```r
#find hourly volatility
#NOTE: this ignores tweets made outside trading hours!!
SPY_volatility_alltime = dplyr::select(SPY,timestamp,r_vol_h)

#aggregating per hour
SPY_volatility_alltime = SPY_volatility_alltime %>%
          mutate(timestamp = floor_date(timestamp, unit = "hour")) %>%
          distinct(timestamp, .keep_all = TRUE)

#select time period
SPY_volatility = filter(SPY_volatility_alltime,
                  between(timestamp,
                          as.Date('2019-01-01'),
                          as.Date('2025-04-10')))
```

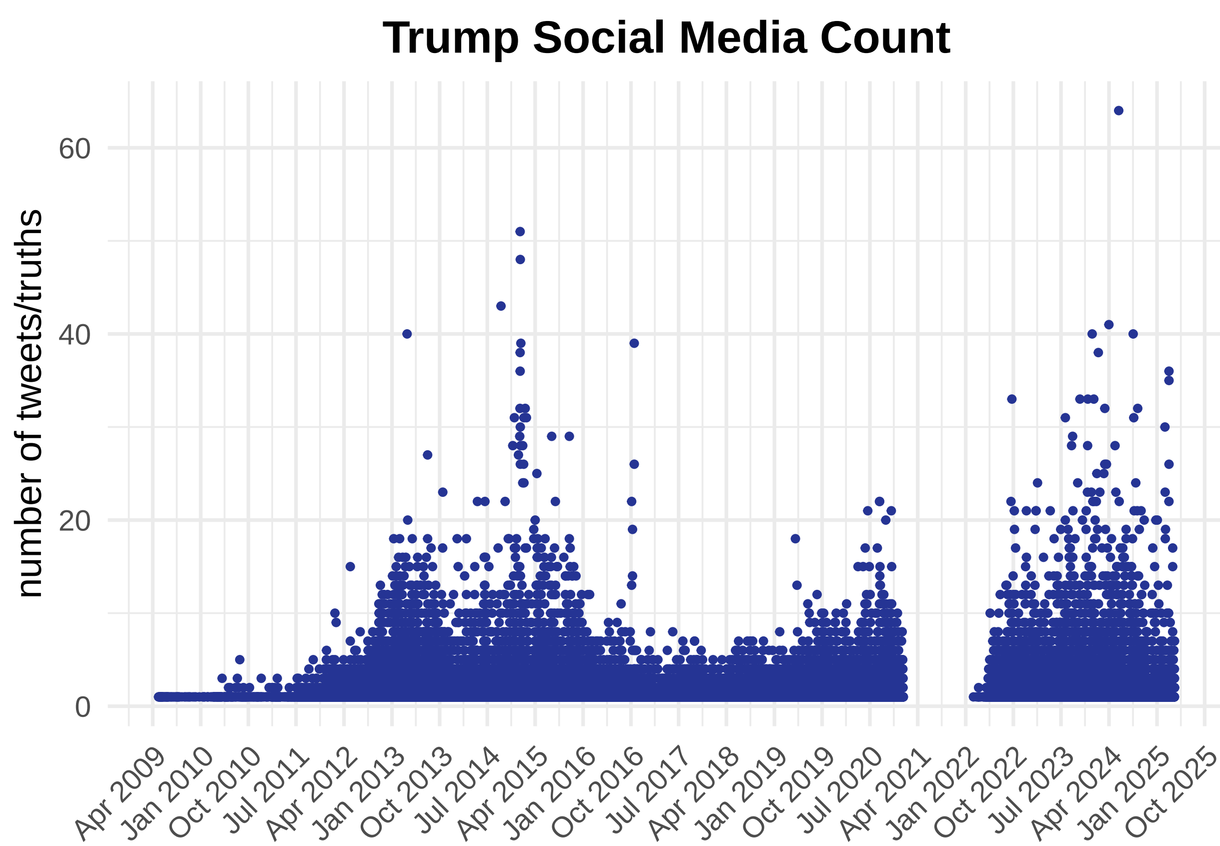
Number of Posts

```r
#find count
tweetcount_alltime = dplyr::select(social_hourly,timestamp,N)

#select time period
tweetcount = filter(tweetcount_alltime,
                    between(timestamp,
                            as.Date('2019-01-01'),
                            as.Date('2025-04-10')))

#plot
ggplot(tweetcount_alltime, aes(x = timestamp, y = N)) +
    geom_point(color = "#253494", size = 1) +
    scale_x_datetime(date_labels = "%b %Y", date_breaks = "9 month") +
    labs(title = "Trump Social Media Count",
         x = NULL,
         y = "number of tweets/truths") +
    theme_minimal(base_size = 14) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1),
          plot.title = element_text(face = "bold", hjust = 0.5))
```

```
## Warning: Removed 1172 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Dummy for Social Media Post

```r
#find count
tweedummy_alltime = dplyr::select(social_hourly,timestamp,dummy)

#select time period
tweetdummy = filter(tweedummy_alltime,
                   between(timestamp,
                           as.Date('2019-01-01'),
                           as.Date('2025-04-10')))
```

Merge

```r
#merge our dependant and independant vars
armax_data = left_join(SPY_volatility, tweetcount, by="timestamp")
armax_data = left_join(armax_data, tweetdummy, by="timestamp")

#convert NA to zeroes
armax_data$N[is.na(armax_data$N)] = 0
armax_data$dummy[is.na(armax_data$dummy)] = 0
```

ARMA-X Models

Find Number of Lags

```r
nb.lags <- 3 #r
count_lags <- embed(armax_data$N, nb.lags + 1)
dummy_lags <- embed(armax_data$dummy, nb.lags + 1)
colnames(count_lags) <- paste0("Lag_", 0:nb.lags)

#align volatility to match count rows (for lag)
vol_aligned <- tail(armax_data$r_vol_h, nrow(count_lags))

#choosing how many lags
# fit an ARMA(0,0,0) model with lm (with r set above)
eq <- lm(vol_aligned ~ count_lags)
eq2 <- lm(vol_aligned ~ dummy_lags)

#compute Newey-West HAC standard errors
var.cov.mat <- NeweyWest(eq, lag = 7, prewhite = FALSE)
robust_se <- sqrt(diag(var.cov.mat))
#for both
var.cov.matD <- NeweyWest(eq2, lag = 7, prewhite = FALSE)
robust_seD <- sqrt(diag(var.cov.matD))

#output table; significant lags are how many we choose
stargazer(eq, eq, type = "latex",
```

```r
          column.labels = c("(no HAC)", "(HAC)"), keep.stat = "n",
          se = list(NULL, robust_se), no.space = TRUE)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Wed, Apr 30, 2025 - 19:28:11

Table 1:

	Dependent variable:	
	vol_aligned	
	(no HAC)	(HAC)
	(1)	(2)
count_lagsLag_0	−0.001*	−0.001***
	(0.001)	(0.0002)
count_lagsLag_1	−0.001	−0.001*
	(0.001)	(0.0003)
count_lagsLag_2	0.0002	0.0002
	(0.001)	(0.0004)
count_lagsLag_3	−0.0002	−0.0002
	(0.001)	(0.0002)
Constant	0.036***	0.036***
	(0.001)	(0.002)
Observations	11,036	11,036
Note:	*p<0.1; **p<0.05; ***p<0.01	

```r
#output table; significant lags are how many we choose
stargazer(eq2, eq2, type = "latex",
          column.labels = c("(no HAC)", "(HAC)"), keep.stat = "n",
          se = list(NULL, robust_seD), no.space = TRUE)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Wed, Apr 30, 2025 - 19:28:11

Tweet Count on Volatility by hour

```r
#find best armax model and fit
armax_fit <- select_armax(armax_data$r_vol_h, armax_data$N,
                          max_p = 5, max_q = 5, max_r = 5, criterion = "AIC")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

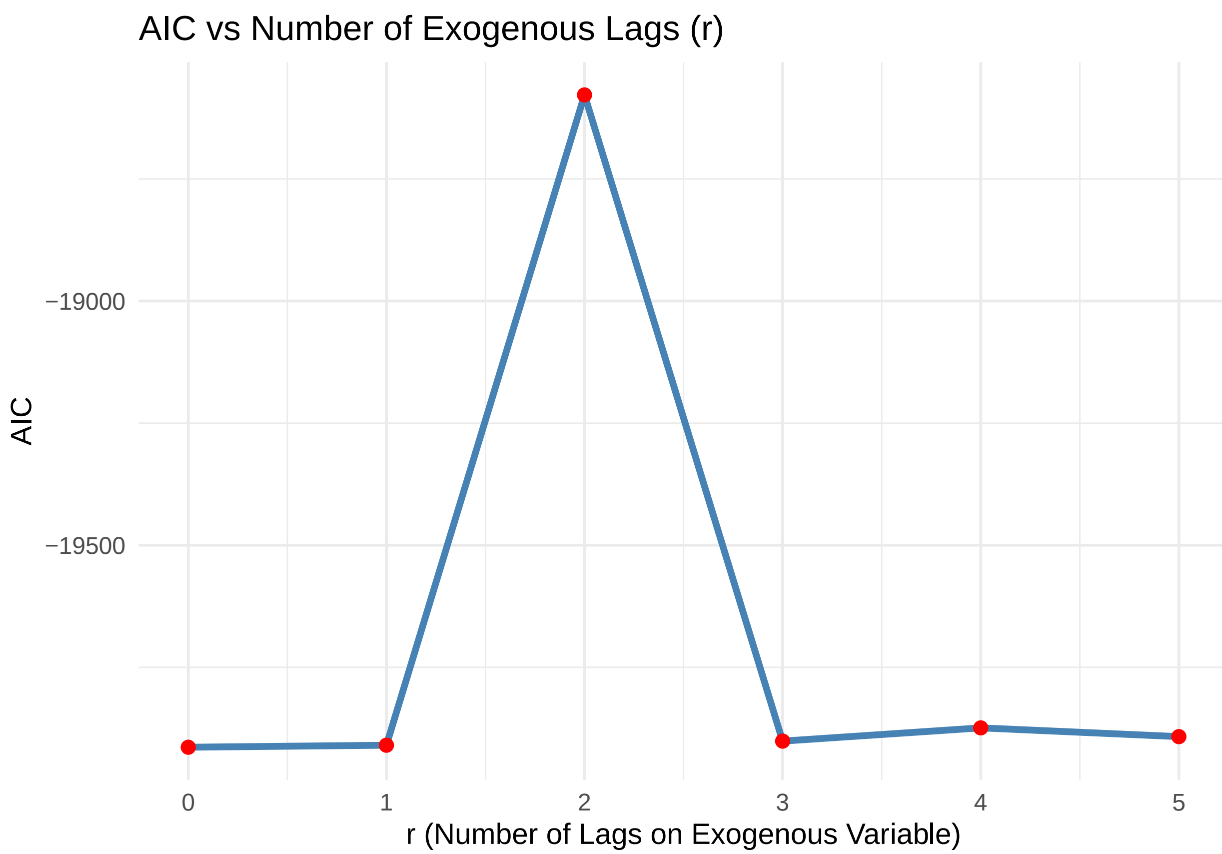
```r
summary(armax_fit$model)
```

Table 2:

	Dependent variable:	
	vol_aligned	
	(no HAC)	(HAC)
	(1)	(2)
dummy_lags1	−0.001	−0.001
	(0.003)	(0.003)
dummy_lags2	−0.003	−0.003
	(0.003)	(0.002)
dummy_lags3	0.006**	0.006*
	(0.003)	(0.003)
dummy_lags4	0.004	0.004
	(0.003)	(0.003)
Constant	0.033***	0.033***
	(0.002)	(0.001)
Observations	11,036	11,036
Note:	*p<0.1; **p<0.05; ***p<0.01	

```
## Series: y_trimmed
## Regression with ARIMA(4,0,5) errors
##
## Coefficients:
##          ar1     ar2     ar3      ar4     ma1      ma2      ma3     ma4     ma5
##       0.0838  1.7090  0.0159  -0.8111  0.2502  -1.7016  -0.5867  0.8504  0.2422
## s.e.  0.0182  0.0137  0.0173   0.0138  0.0213   0.0113   0.0297  0.0108  0.0152
##       intercept  X1_Lag_0
##          0.0335    -4e-04
## s.e.     0.0240     4e-04
##
## sigma^2 = 0.009625:  log likelihood = 9968.81
## AIC=-19913.63   AICc=-19913.6   BIC=-19825.92
##
## Training set error measures:
##                        ME       RMSE        MAE       MPE     MAPE     MASE
## Training set 0.0008280272 0.09805725 0.02189906 -74.14026 116.5155 1.095958
##                      ACF1
## Training set -0.004526612
```

```
armax_fit$ICplot
```




```
armax_fit$params
```

```
## $p
## [1] 4
##
## $q
## [1] 5
##
## $r
## [1] 0
```

Tweet Dummy on Volatility by hour

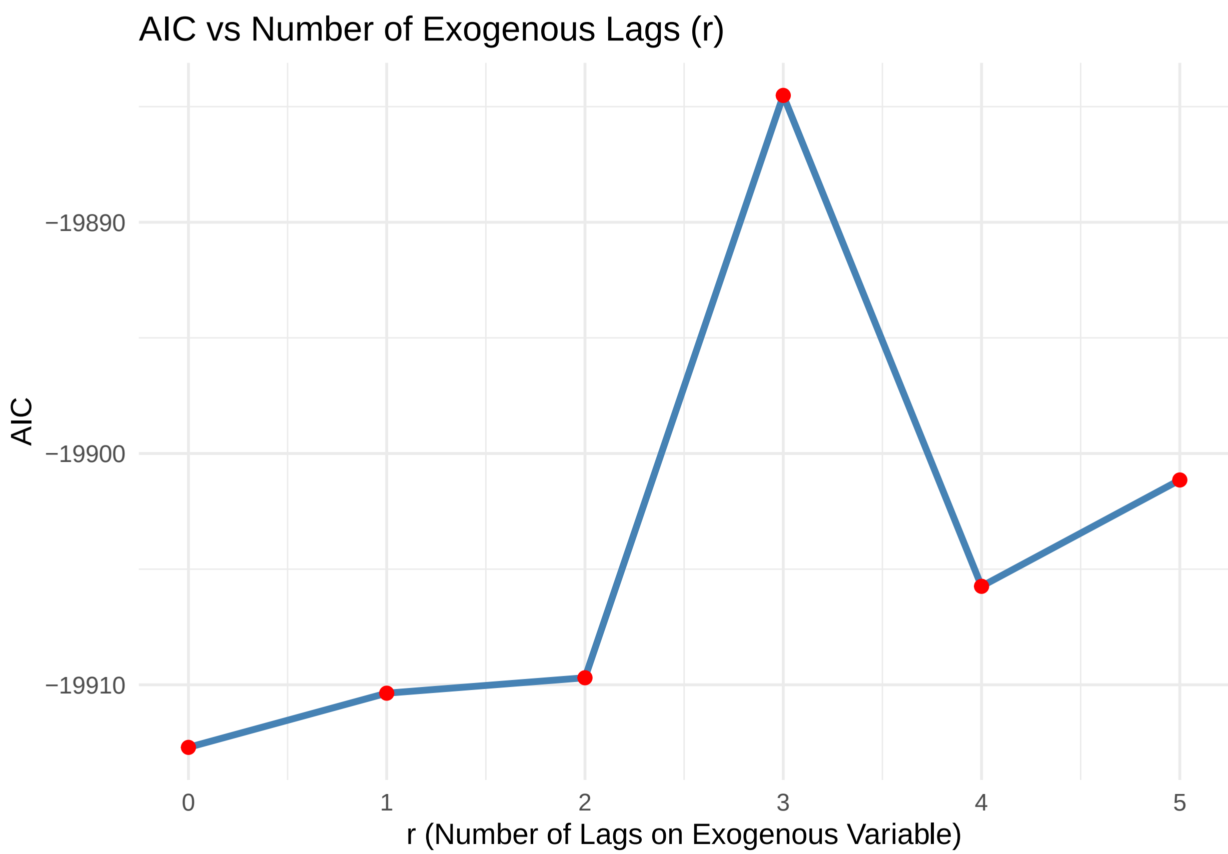
```
#find best armax model and fit
armax_fit <- select_armax(armax_data$r_vol_h, armax_data$dummy,
                          max_p = 5, max_q = 5, max_r = 5, criterion = "AIC")

summary(armax_fit$model)
```

```
## Series: y_trimmed
## Regression with ARIMA(4,0,5) errors
##
## Coefficients:
```

```
##          ar1     ar2     ar3      ar4     ma1      ma2      ma3     ma4     ma5
##       0.0836  1.7077  0.0161  -0.8097  0.2505  -1.7007  -0.5867  0.8497  0.2419
## s.e.  0.0185  0.0138  0.0175   0.0138  0.0215   0.0115   0.0299  0.0110  0.0152
##       intercept  X1_Lag_0
##          0.0332   -0.0003
## s.e.     0.0243    0.0018
##
## sigma^2 = 0.009626:  log likelihood = 9968.35
## AIC=-19912.71   AICc=-19912.68   BIC=-19825
##
## Training set error measures:
##                       ME       RMSE        MAE       MPE     MAPE     MASE
## Training set 0.000833075 0.09806137 0.02187804 -74.22759 115.9798 1.094907
##                      ACF1
## Training set -0.004629921
```

```
armax_fit$ICplot
```




```
armax_fit$params
```

```
## $p
## [1] 4
##
## $q
## [1] 5
```

```
##
## $r
## [1] 0
```