Sales Tax Forecasting

ECON436, Colorado State University

Abdulla Almansoori, Alexander Firth, Nathan Seto

22 April 2025

This document contains the R code that forecasts the sales tax revenue to be collected by the city of Fort Collins, Colorado in 2025. This is originally an assignment from ECON436 Economic Forecasting at Colorado State University and was completed by students Abdulla Almansoori, Alexander Firth, and Nathan Seto.

Data

```
# Import BLS and Fort Collins sales tax data set
FD_raw <- read_excel("~/ECON436_SalesTax/Data/FinalData.xlsx", sheet = 1)

# Traffic data (external)
TRAF_raw <- read_excel("~/ECON436_SalesTax/Data/TrafficVolume.xlsx")
```

Traffic data from CDOT.

Define Model Variables

```
FD_1 <- FD_raw[1:157, ]

# Prescribed Adjustments
FD_1$Lead_STF_Real[FD_1$year == 2024] <- FD_1$Lead_STF_Real[FD_1$year == 2024] *
    (3.85/4.35)
FD_1$SG[1:12] <- FD_1$SG[1:12] + 5
FD_1$G[1:12] <- FD_1$G[1:12] + 5
FD_1$EDUHS[1:12] <- FD_1$EDUHS[1:12] - 5

# Cleaning traffic data
TRAF_clean <- TRAF_raw[TRAF_raw$Year >= 2012 & TRAF_raw$Year <= 2024, ]
TRAF_clean <- TRAF_clean[, -c(1, 15)] |>
    arrange(Year)

TRAF_long <- TRAF_clean |>
    pivot_longer(cols = -Year, names_to = "Month", values_to = "Value") |>
    mutate(Month = match(Month, month.abb), Month = as.Date(paste(Year, Month, "01",
        sep = "-")), ) |>
    dplyr::select(Month, Year, Value, )
```

```r
TRAF_long <- TRAF_long |>
    mutate(Month = lubridate::month(TRAF_long$Month))

# Binding traffic data and FinalData
DF <- cbind(FD_1[1:nrow(TRAF_long), ], TRAF_long$Value)
DF <- DF |>
    mutate(Month = month(DF$Month))

# Selecting predictors + Lead_STF_Real
DF_NEW <- DF |>
    dplyr::select("LH", "INFO", "EDUHS", "MAN", "AFS", "RT", "TRAF_long$Value", "Lead_STF_Real")
DF_NEW <- DF_NEW |>
    rename(traffic_frequency = `TRAF_long$Value`)  # rename column

# Log all variables
LDF <- log(DF_NEW)
```

Conduct the Johansen Procedure

```r
# Johansen procedure
JOH <- ca.jo(LDF, ecdet = "none", type = "eigen", K = 12)
# K = number of periods in a year (4 if quarterly data, 12 if monthly data)
summary(JOH)
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: maximal eigenvalue statistic (lambda max) , with linear trend
##
## Eigenvalues (lambda):
## [1] 0.5543074037 0.3390995117 0.2882789243 0.2687118738 0.2075456524
## [6] 0.1398879556 0.0735274494 0.0008841772
##
## Values of teststatistic and critical values of test:
##
##            test 10pct  5pct  1pct
## r <= 7 |   0.13  6.50  8.18 11.65
## r <= 6 |  11.00 12.91 14.90 19.19
## r <= 5 |  21.70 18.90 21.07 25.75
## r <= 4 |  33.50 24.78 27.14 32.14
## r <= 3 |  45.06 30.84 33.32 38.78
## r <= 2 |  48.97 36.25 39.43 44.59
## r <= 1 |  59.64 42.06 44.91 51.30
## r = 0  | 116.37 48.43 51.07 57.07
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                             LH.l12    INFO.l12   EDUHS.l12     MAN.l12    AFS.l12
```

```
## LH.l12                1.00000000  1.0000000  1.00000000  1.00000000  1.0000000
## INFO.l12              0.19454137  0.0271990 -0.14228098  0.10570286  0.2391922
## EDUHS.l12            -2.61581398 -0.2960907 -0.57228385  0.22270593 -1.2411842
## MAN.l12               3.89925922  0.2312172  0.52745161 -0.72126183  2.1687998
## AFS.l12              -1.70399998 -1.1178626 -1.19086603 -1.06689267 -0.4156994
## RT.l12               -1.96211264  0.4071131  0.34560067  0.47325358  0.4770030
## traffic_frequency.l12  0.62504252 -0.1588076  0.26736458 -0.04802203 -0.8018121
## Lead_STF_Real.l12     0.03414007  0.0198990 -0.08001404  0.09379728 -0.9853087
##                          RT.l12 traffic_frequency.l12 Lead_STF_Real.l12
## LH.l12                1.0000000           1.00000000       1.000000000
## INFO.l12              0.2504010          -0.03782207       0.005489628
## EDUHS.l12            -0.5858466          -0.06460442       0.030693665
## MAN.l12              -0.2309650           0.08110396      -0.080898918
## AFS.l12              -0.4617314          -1.01524948      -1.105052907
## RT.l12               -1.3905598           0.25316177      -0.607389401
## traffic_frequency.l12  0.2990137          -0.11633008       0.344918669
## Lead_STF_Real.l12     0.9054268          -0.20123568       0.179261944
##
## Weights W:
## (This is the loading matrix)
##
##                          LH.l12   INFO.l12   EDUHS.l12     MAN.l12    AFS.l12
## LH.d                 0.56086187 -2.2569204  0.62698816 -0.06084466 -0.9985306
## INFO.d               0.17921780 -2.8861671  0.57608474 -0.52708945 -0.1296603
## EDUHS.d              0.15710718 -0.3072272  0.38080993 -0.26146413 -0.2414772
## MAN.d                0.03472949 -0.5469909 -0.02210845  0.11810778 -0.1192807
## AFS.d                0.52035927 -1.8838866  0.87825031  0.29049453 -0.9491075
## RT.d                 0.30898772 -0.5865078  0.17046059  0.26620910 -0.2946883
## traffic_frequency.d  0.13137228 -0.4362515  0.22608531 -0.52408281 -0.8149320
## Lead_STF_Real.d      0.50804216 -0.3784209  0.19318619 -0.48308111 -0.5091792
##                          RT.l12 traffic_frequency.l12 Lead_STF_Real.l12
## LH.d                -0.186567657          -0.495853133      0.0083938407
## INFO.d              -0.024531227          -0.066811628      0.0113504334
## EDUHS.d              0.002159474          -0.108996524     -0.0009491958
## MAN.d                0.024771829          -0.004483752     -0.0037577871
## AFS.d               -0.176390218          -0.417310174      0.0064031040
## RT.d                -0.022427603          -0.117918241      0.0045777620
## traffic_frequency.d  0.082070001           0.249034419      0.0366606670
## Lead_STF_Real.d     -0.233612174           0.290082757     -0.0180960694
```

Confirm the number of cointegrating vectors based on the results of the Johansen procedure.

Cointegrating Vectors

```
BETA <- round(coefB(JOH, r = 6), 3)  # r = number of cointegrating vectors
BETA  # Short run adjustment coefficients
```

```
##                        [,1]  [,2]   [,3]  [,4]  [,5]  [,6]
## LH.l12                1.000 0.000  0.000 0.000 0.000 0.000
## INFO.l12              0.000 1.000  0.000 0.000 0.000 0.000
## EDUHS.l12             0.000 0.000  1.000 0.000 0.000 0.000
```

```
## MAN.l12               0.000  0.000   0.000  1.000  0.000  0.000
## AFS.l12               0.000  0.000   0.000  0.000  1.000  0.000
## RT.l12                0.000  0.000   0.000  0.000  0.000  1.000
## traffic_frequency.l12 -6.094  5.738 -11.091 -5.614 -2.364  2.826
## Lead_STF_Real.l12      5.424 -7.002   9.922  4.596  1.704 -3.521
```

Create VECM Using Data Up To January 2025

```r
VECM <- VECM(LDF, lag = 11, r = 6, estim = "ML", include = "const")  # lag = K - 1
# summary(VECM)
```

For each vector, we need at least one adjustment speed to have the opposite sign of the corresponding cointegrating vector coefficient and be statistically significant.

Model Evaluation

Rerun the model on data up to January 2024.

```r
NROW <- nrow(LDF)  # truncate the data by removing last 12 observations
LDF_2023 <- LDF[1:(NROW - 12), ]

# Fit the same VECM
JOH2023 <- ca.jo(LDF_2023, ecdet = "none", type = "eigen", K = 12)
summary(JOH2023)
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: maximal eigenvalue statistic (lambda max) , with linear trend
##
## Eigenvalues (lambda):
## [1] 0.70875070 0.47401855 0.39259558 0.27611749 0.25461687 0.18706449 0.09291123
## [8] 0.01507950
##
## Values of teststatistic and critical values of test:
##
##            test 10pct  5pct  1pct
## r <= 7 |   2.01  6.50  8.18 11.65
## r <= 6 |  12.87 12.91 14.90 19.19
## r <= 5 |  27.34 18.90 21.07 25.75
## r <= 4 |  38.79 24.78 27.14 32.14
## r <= 3 |  42.65 30.84 33.32 38.78
## r <= 2 |  65.81 36.25 39.43 44.59
## r <= 1 |  84.81 42.06 44.91 51.30
## r = 0  | 162.83 48.43 51.07 57.07
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
```

```
##
##                            LH.l12   INFO.l12    EDUHS.l12    MAN.l12
## LH.l12                 1.00000000  1.0000000  1.000000000  1.0000000
## INFO.l12              -0.23524305 -0.6265882 -0.006730193  0.2668923
## EDUHS.l12              0.67671557 -5.0884990 -0.525170875  1.9009301
## MAN.l12               -1.61477648  3.3440277  0.596735652 -3.6184392
## AFS.l12               -0.96607715 -4.3919307 -1.173662597 -1.5099433
## RT.l12                 1.64364471  4.5779936  0.522589643  3.0097473
## traffic_frequency.l12  0.07739906  2.4615662 -0.239585370 -0.8217406
## Lead_STF_Real.l12     -0.09667186  0.7287776  0.006780478  0.2595148
##                           AFS.l12      RT.l12 traffic_frequency.l12
## LH.l12                 1.00000000  1.00000000            1.00000000
## INFO.l12               0.04669676  0.17057879           -0.06032888
## EDUHS.l12             -0.19735117  0.01846803            0.22620783
## MAN.l12                0.18959471 -0.74078591           -0.34126508
## AFS.l12               -0.99971797 -1.04234608           -1.29526397
## RT.l12                 0.15494889 -0.12093835           -0.32293306
## traffic_frequency.l12 -0.09199165  0.06009914            0.38292133
## Lead_STF_Real.l12     -0.08151450  0.60027070            0.22993971
##                       Lead_STF_Real.l12
## LH.l12                       1.00000000
## INFO.l12                    -0.08271911
## EDUHS.l12                    0.20629398
## MAN.l12                     -0.18296425
## AFS.l12                     -1.01810383
## RT.l12                      -0.18325226
## traffic_frequency.l12        0.11434067
## Lead_STF_Real.l12           -0.08113835
##
## Weights W:
## (This is the loading matrix)
##
##                          LH.l12   INFO.l12   EDUHS.l12     MAN.l12    AFS.l12
## LH.d                -0.64597488 0.30401826 -0.92229848  0.07088857 -4.0407934
## INFO.d              -0.16356351 0.07442045 -1.24696782 -0.20080393 -1.2169005
## EDUHS.d             -0.14774003 0.09018019 -0.08932307 -0.07669147 -0.9551406
## MAN.d               -0.04972307 0.02157166 -0.34200131  0.01774512 -0.6981782
## AFS.d               -0.50402686 0.32941150 -0.83978502  0.09128764 -3.6820452
## RT.d                -0.39987844 0.12107439 -0.39872411  0.03779269 -0.7902962
## traffic_frequency.d -0.45011324 0.08598111  0.30395632 -0.02041786 -5.1889985
## Lead_STF_Real.d     -0.90116847 0.22936795  0.52742466  0.06542598 -4.6508246
##                         RT.l12 traffic_frequency.l12 Lead_STF_Real.l12
## LH.d                 0.2533510            0.76170562       0.388877160
## INFO.d              -0.1642948            0.08967581       0.254594013
## EDUHS.d              0.1425173            0.15815880       0.050042612
## MAN.d                0.1222295            0.01928131      -0.015585432
## AFS.d                0.2653150            0.67062575       0.368276948
## RT.d                 0.2082381            0.20236947       0.129276161
## traffic_frequency.d  0.9708586            0.08972961       0.648261347
## Lead_STF_Real.d     -0.2128882            0.07881669      -0.009223651
```

Forecast 2024 data.

```r
# vec2var: Transform a VECM to VAR object in levels
vec2var_ca.jo2023 <- vec2var(JOH2023, r = 6)

# forecasting horizon (12 months)
nhor <- 12

# Forecasting for 2024
pred_vec2var_ca.jo2024 <- predict(vec2var_ca.jo2023, n.ahead = nhor)
```

Compare actual and forecasted 2024 Lead_STF_Real. Calculate mean absolute error (MAE) for 2024.

```r
# Extract forecasted values
Forecast2024 <- pred_vec2var_ca.jo2024[["fcst"]][["Lead_STF_Real"]][, 1]

# Extract observed values
Actual2024 <- LDF$Lead_STF_Real[FD_1$year == 2024]

# Calculate individual absolute error
AE <- abs(Forecast2024 - Actual2024)

# Mean error
MAE <- mean(AE)

round(MAE, 4)
```

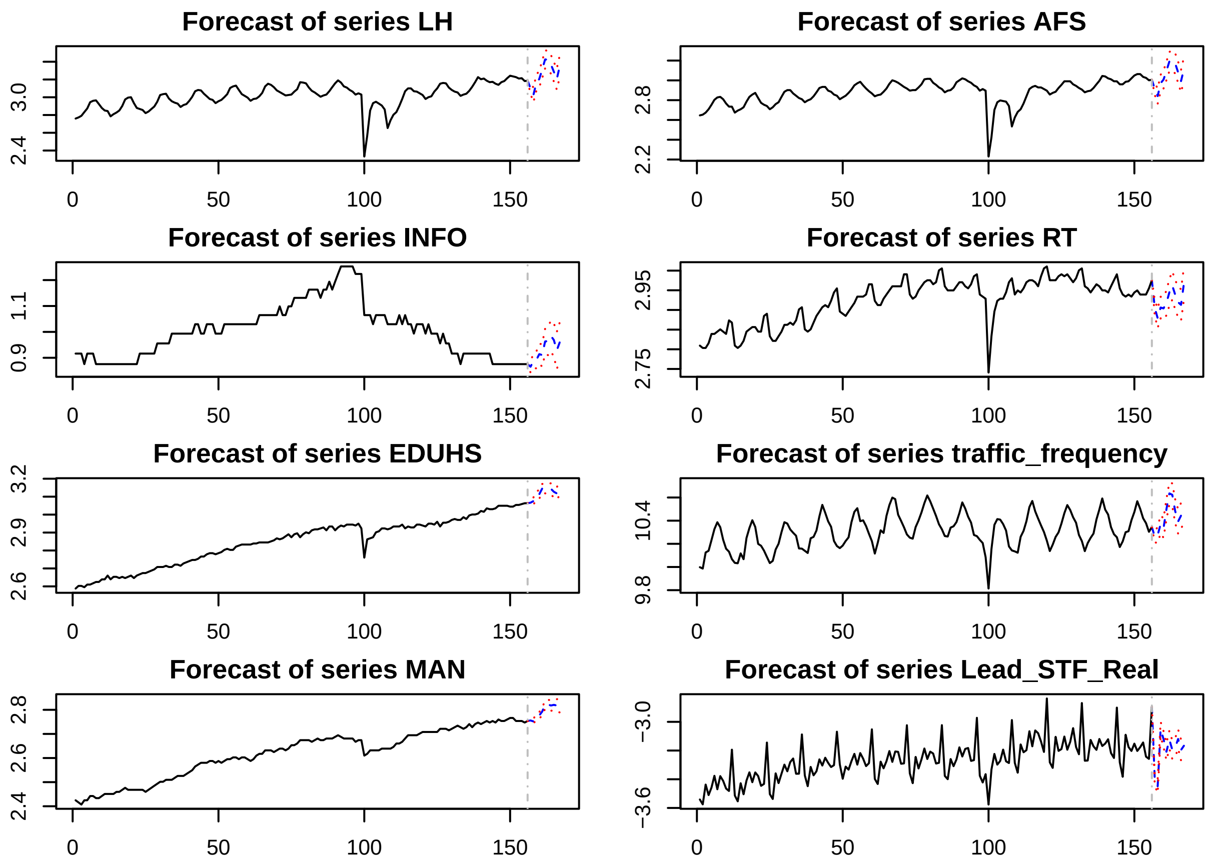
```
## [1] 0.1885
```

```r
round(Actual2024, 4) == round(LDF$Lead_STF_Real[FD_1$year == 2024], 4)  # Verification
```

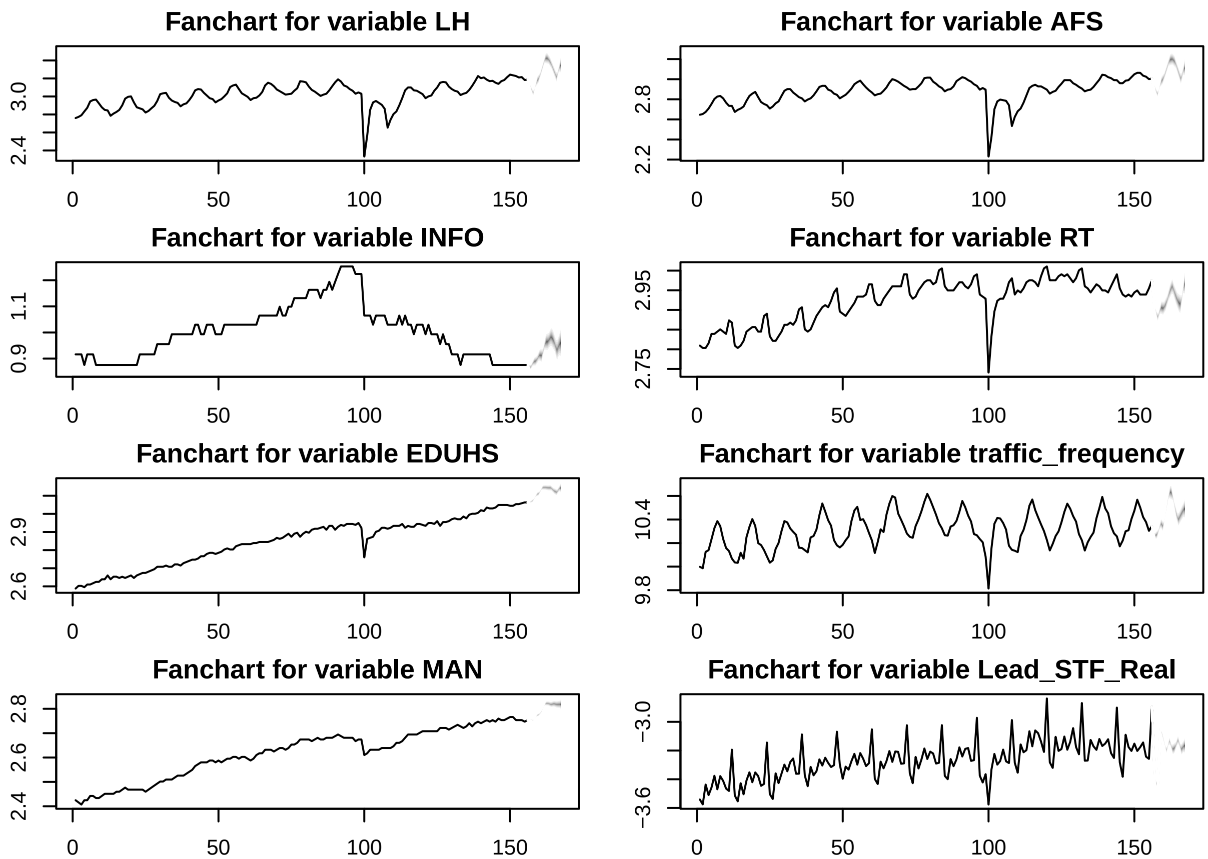
```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Forecast Lead_STF_Real for February through November of 2025 and save it.

```r
# vec2var: Transform a VECM to VAR in levels
vec2var_ca.jo <- vec2var(JOH, r = 6)

nhor <- 11  # Set forecasting horizon again (11 months)

# Forecasting
pred_vec2var_ca.jo <- predict(vec2var_ca.jo, n.ahead = nhor)

# Graphical representation
par(mai = rep(0.4, 4))
plot(pred_vec2var_ca.jo)
```




```r
par(mai = rep(0.4, 4))
fanchart(pred_vec2var_ca.jo)
```




```r
# Predicted Values
pred_vec2var_ca.jo$fcst$Lead_STF_Real <- exp(pred_vec2var_ca.jo$fcst$Lead_STF_Real)
pred_vec2var_ca.jo$fcst$Lead_STF_Real <- pred_vec2var_ca.jo$fcst$Lead_STF_Real *
    (4.35/3.85)
print(pred_vec2var_ca.jo$fcst$Lead_STF_Real)
```

```
##              fcst      lower      upper       CI
##  [1,] 0.03760063 0.03607739 0.03918818 1.177575
##  [2,] 0.03591039 0.03425275 0.03764825 1.184550
##  [3,] 0.05350185 0.05086462 0.05627581 1.188452
##  [4,] 0.04989126 0.04720601 0.05272925 1.194141
##  [5,] 0.04568014 0.04303075 0.04849266 1.199436
##  [6,] 0.04966853 0.04664215 0.05289127 1.203182
##  [7,] 0.04633854 0.04344787 0.04942154 1.205043
##  [8,] 0.04703595 0.04404951 0.05022486 1.206472
##  [9,] 0.04991698 0.04670235 0.05335287 1.207642
## [10,] 0.04624774 0.04300228 0.04973814 1.215144
## [11,] 0.04762490 0.04411286 0.05141654 1.219825
```

```r
# Saving predictions FRAME <-
# as.data.frame(pred_vec2var_ca.jo$fcst$Lead_STF_Real) FRAME <- FRAME |>
# mutate(year = 2025, month = 1:11, .before = fcst) write_csv(FRAME, file =
# 'Forecast2025-3.csv')
```