Assignment 6

Jaewoo Cho

Forecast median_days using VAR and VECM models

1. Loading nashville_housing and housing_validation and format them into tsibble

Set up a pandemic dummy variable between May 2020 and June 2021 in nashville_housing (add a pandemic dummy variable to your validation data too!)

2. Fit an {fpp3} VAR model to the nashville_housing data

```
# Fit VAR model
fit_var <- housing_ts %>%
  model(
    lag_2 = VAR(
      vars(housing, unemployment, median_days,
           price_decreased, pending_listing) ~
        xreg(outlier, pandemic)
    )
  )
## Make sure to include exogenous variables `outlier` and `pandemic`
fit_var
```

```
# A mable: 1 x 1
     lag_2
   <model>
1 <VAR(2)>
```

3. Report fit of VARmodel. How many lags were used?

```
# Report fit
report(fit_var)
```

```
Series: housing, unemployment, median_days, price_decreased, pending_listing
Model: VAR(2)

Coefficients for housing:
      lag(housing,1)  lag(unemployment,1)  lag(median_days,1)
              1.4460             -24.8325             -4.0360
s.e.          0.1395              29.5909             10.9709
      lag(price_decreased,1)  lag(pending_listing,1)  lag(housing,2)
```

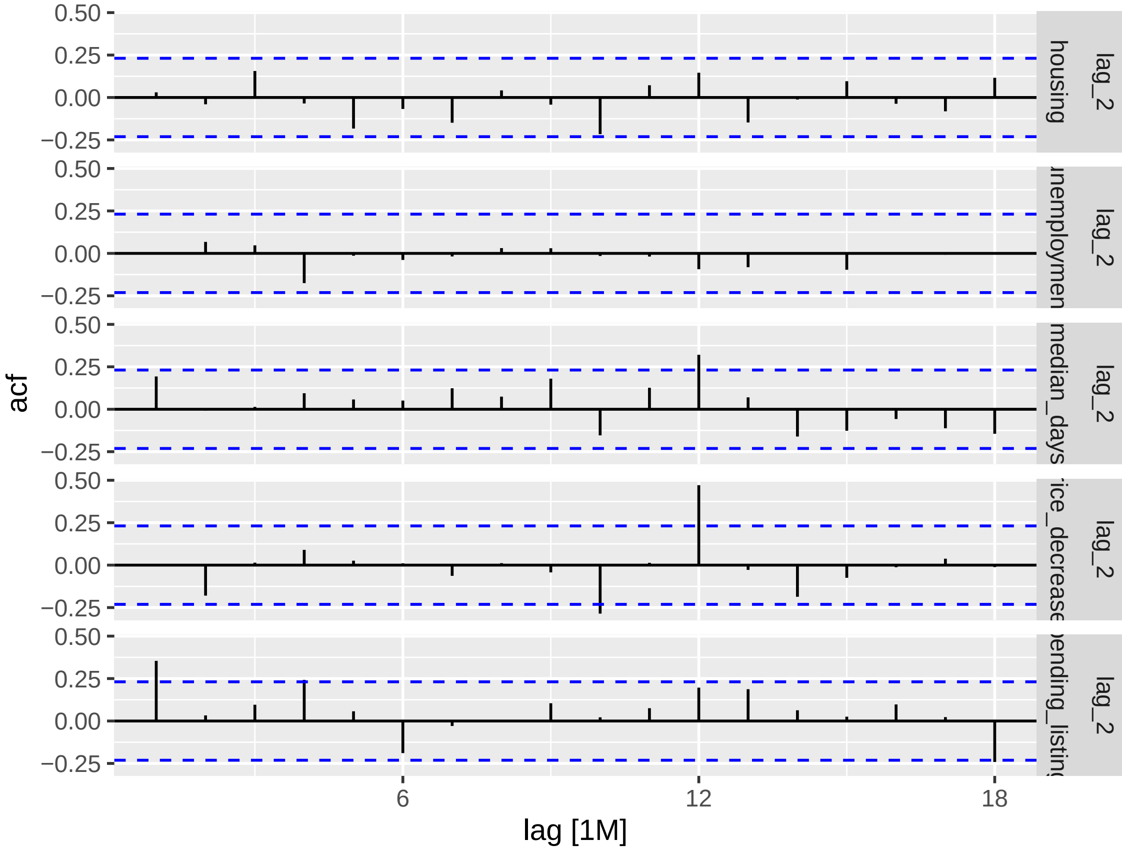
```
                    0.3521                  0.3178         -0.3131
s.e.                0.1783                  0.1373          0.1440
     lag(unemployment,2)  lag(median_days,2)  lag(price_decreased,2)
               -34.1893             -3.8458                 -0.7766
s.e.            24.2889              8.5064                  0.1728
     lag(pending_listing,2)     outlier   pandemic
                   -0.1170  -2353.7642  -178.1484
s.e.                0.1403    276.8817   145.2545

Coefficients for unemployment:
     lag(housing,1)  lag(unemployment,1)  lag(median_days,1)
             0.0011               0.6108             -0.0134
s.e.         0.0008               0.1651              0.0612
     lag(price_decreased,1)  lag(pending_listing,1)  lag(housing,2)
                   -0.0022                  0.0014          -3e-04
s.e.                0.0010                  0.0008           8e-04
     lag(unemployment,2)  lag(median_days,2)  lag(price_decreased,2)
                -0.0168             -0.0276                   0.000
s.e.             0.1355              0.0475                   0.001
     lag(pending_listing,2)  outlier  pandemic
                    -4e-04  -0.5911    0.0237
s.e.                 8e-04   1.5451    0.8106

Coefficients for median_days:
     lag(housing,1)  lag(unemployment,1)  lag(median_days,1)
             0.0055              -0.1309              0.7635
s.e.         0.0016               0.3479              0.1290
     lag(price_decreased,1)  lag(pending_listing,1)  lag(housing,2)
                   -0.0108                  0.0046         -0.0028
s.e.                0.0021                  0.0016          0.0017
     lag(unemployment,2)  lag(median_days,2)  lag(price_decreased,2)
                 0.0842             -0.2561                  0.0089
s.e.             0.2856              0.1000                  0.0020
     lag(pending_listing,2)  outlier  pandemic
                   -0.0030  -5.6353   -0.9413
s.e.                0.0016   3.2558    1.7080

Coefficients for price_decreased:
     lag(housing,1)  lag(unemployment,1)  lag(median_days,1)
             0.4722             -25.0214              7.7797
s.e.         0.1348              28.5962             10.6021
     lag(price_decreased,1)  lag(pending_listing,1)  lag(housing,2)
                    0.8169                  0.4004         -0.3002
s.e.                0.1723                  0.1327          0.1392
     lag(unemployment,2)  lag(median_days,2)  lag(price_decreased,2)
               -24.0572            -12.7665                 -0.4756
s.e.            23.4724              8.2205                  0.1670
     lag(pending_listing,2)   outlier  pandemic
                   -0.1865  -15.2563  -36.8159
s.e.                0.1356  267.5743  140.3718

Coefficients for pending_listing:
     lag(housing,1)  lag(unemployment,1)  lag(median_days,1)
            -0.0249              47.8599              3.2591
```

```
s.e.         0.0981             20.8126             7.7163
      lag(price_decreased,1)  lag(pending_listing,1)  lag(housing,2)
                      0.2271                  1.0216         -0.0535
s.e.                  0.1254                  0.0966          0.1013
      lag(unemployment,2)  lag(median_days,2)  lag(price_decreased,2)
                 -10.0297              8.7823                 -0.1648
s.e.              17.0834              5.9829                  0.1215
      lag(pending_listing,2)   outlier   pandemic
                     -0.1081  2690.234  -101.3304
s.e.                  0.0987   194.743   102.1638

Residual covariance matrix:
                   housing unemployment median_days price_decreased
housing         65333.6050     132.7881    -56.4621      34294.8379
unemployment      132.7881       2.0346      0.4186        -21.2753
median_days       -56.4621       0.4186      9.0334       -349.1250
price_decreased 34294.8379     -21.2753   -349.1250      61015.0513
pending_listing  1735.7366     -28.3019   -247.7888      17662.7114
                pending_listing
housing               1735.7366
unemployment           -28.3019
median_days           -247.7888
price_decreased      17662.7114
pending_listing      32320.0217

log likelihood = -1659.35
AIC = 3488.69   AICc = 2574.94  BIC = 3679.82
```

Answer: "How many lags were used?" As you can see there are 2 lags in the model shown by Model: VAR(2).

4. Plot the autocorrelations of the residuals

```r
# Plot autocorrelations
# Autocorrelation of residuals
fit_var %>% augment() %>%
  ACF(.innov) %>% autoplot()
```



5. Were any autocorrelations significant in 4.? Report which variables and at what lags. Be sure to report *all* variables *and* lags

> Report significant autocorrelations. Report all variables and lags that are significant. Housing: Significant autocorrelations with its own first lag and the outlier. Unemployment: Significant autocorrelation with its own first lag. Median_days: Significant autocorrelations with its own first and second lags. Price_decreased: Significant autocorrelations with housing's first lag, its own first lag, and pending_listing's first lag. Pending_listing: Significant autocorrelations with unemployment's first lag, its own first lag, and the outlier.

6. Fit a `VAR` model to the `nashville_housing` data using {vars}

```
var2 <- vars::VAR(y = housing_ts[,c("housing", "unemployment", "median_days", "price_decreased", "pendi
  type = "none",
  p = 2)
dummy_matrix <- matrix(rep(0, 2 * 24),
                 nrow = 24,
                 dimnames = list(NULL, c("outlier", "pandemic")))
var_fc <- predict(var2, n.ahead = 24, dumvar = dummy_matrix)
var_fc$fcst$housing
```

```
         fcst    lower    upper        CI
 [1,] 4777.020 4276.044 5277.995  500.9754
```

```
 [2,] 5159.572 4182.943 6136.200  976.6281
 [3,] 5126.324 3712.075 6540.574 1414.2496
 [4,] 4821.340 3052.146 6590.534 1769.1941
 [5,] 4390.288 2366.044 6414.531 2024.2434
 [6,] 3971.400 1780.880 6161.920 2190.5200
 [7,] 3675.272 1378.663 5971.881 2296.6088
 [8,] 3557.178 1187.657 5926.699 2369.5210
 [9,] 3612.312 1186.348 6038.275 2425.9638
[10,] 3791.928 1317.451 6266.404 2474.4764
[11,] 4026.594 1505.510 6547.677 2521.0834
[12,] 4248.431 1676.448 6820.414 2571.9827
[13,] 4407.996 1776.255 7039.737 2631.7414
[14,] 4482.861 1782.291 7183.431 2700.5698
[15,] 4477.307 1702.953 7251.661 2774.3535
[16,] 4415.174 1567.738 7262.610 2847.4361
[17,] 4329.316 1413.714 7244.919 2915.6024
[18,] 4251.142 1273.813 7228.470 2977.3285
[19,] 4202.890 1169.612 7236.169 3033.2789
[20,] 4194.010 1108.878 7279.142 3085.1323
[21,] 4221.631 1086.956 7356.305 3134.6745
[22,] 4274.093 1090.679 7457.507 3183.4138
[23,] 4335.933 1103.449 7568.418 3232.4842
[24,] 4392.712 1110.130 7675.295 3282.5824
```

7. Perform the serial test on the residual autocorrelations. Interpret the *p*-value. What does this mean?

```r
# Perform serial test
serial.test(var2, lags.pt = 10, type = "PT.adjusted")
```

```
    Portmanteau Test (adjusted)

data:  Residuals of VAR object var2
Chi-squared = 250.55, df = 200, p-value = 0.008801
```

```r
## Set 'lags.pt' to `4`
## Set 'type' to "PT.adjusted"
serial.test(var2, lags.pt = 4, type = "PT.adjusted")
```

```
    Portmanteau Test (adjusted)

data:  Residuals of VAR object var2
Chi-squared = 128.79, df = 50, p-value = 7.068e-09
```

> Answer: "Interpret the *p*-value. What does this mean?" The p-value means small p-value 7.068e-09 from the Portmanteau Test indicates that there's strong evidence against the residuals of the VAR model being white noise. This suggests that the model may not be a perfect fit for the data, as it leaves some temporal structure unaccounted for in its residuals.

8. Forecast 13 time points ahead using the VAR model

```
dummy_matrix_13 <- matrix(rep(0, 2 * 13),  nrow = 13, dimnames = list(NULL, c("outlier", "pandemic")))
# Forecasting 13 points
fcvar13 <- predict(var2, n.ahead = 13, dumvar = dummy_matrix_13)
fcvar13$fcst$housing
```

Don't forget to create a dummy variable matrix for the "pandemic" variable

```
          fcst    lower    upper        CI
 [1,] 4777.020 4276.044 5277.995  500.9754
 [2,] 5159.572 4182.943 6136.200  976.6281
 [3,] 5126.324 3712.075 6540.574 1414.2496
 [4,] 4821.340 3052.146 6590.534 1769.1941
 [5,] 4390.288 2366.044 6414.531 2024.2434
 [6,] 3971.400 1780.880 6161.920 2190.5200
 [7,] 3675.272 1378.663 5971.881 2296.6088
 [8,] 3557.178 1187.657 5926.699 2369.5210
 [9,] 3612.312 1186.348 6038.275 2425.9638
[10,] 3791.928 1317.451 6266.404 2474.4764
[11,] 4026.594 1505.510 6547.677 2521.0834
[12,] 4248.431 1676.448 6820.414 2571.9827
[13,] 4407.996 1776.255 7039.737 2631.7414
```

```
future_dates <- seq(from = as.Date("2023-01-01"), by = "month", length.out = 13)
fc_housing_13 <- fcvar13$fcst$housing
# Creating a tsibble
var_fc_tsbl <- data.frame(date = future_dates,
  housing_mean = fc_housing_13[,"fcst"],
  housing_sd = fc_housing_13[,"CI"]) %>%
  as_tsibble(index = date)
var_fc_tsbl
```

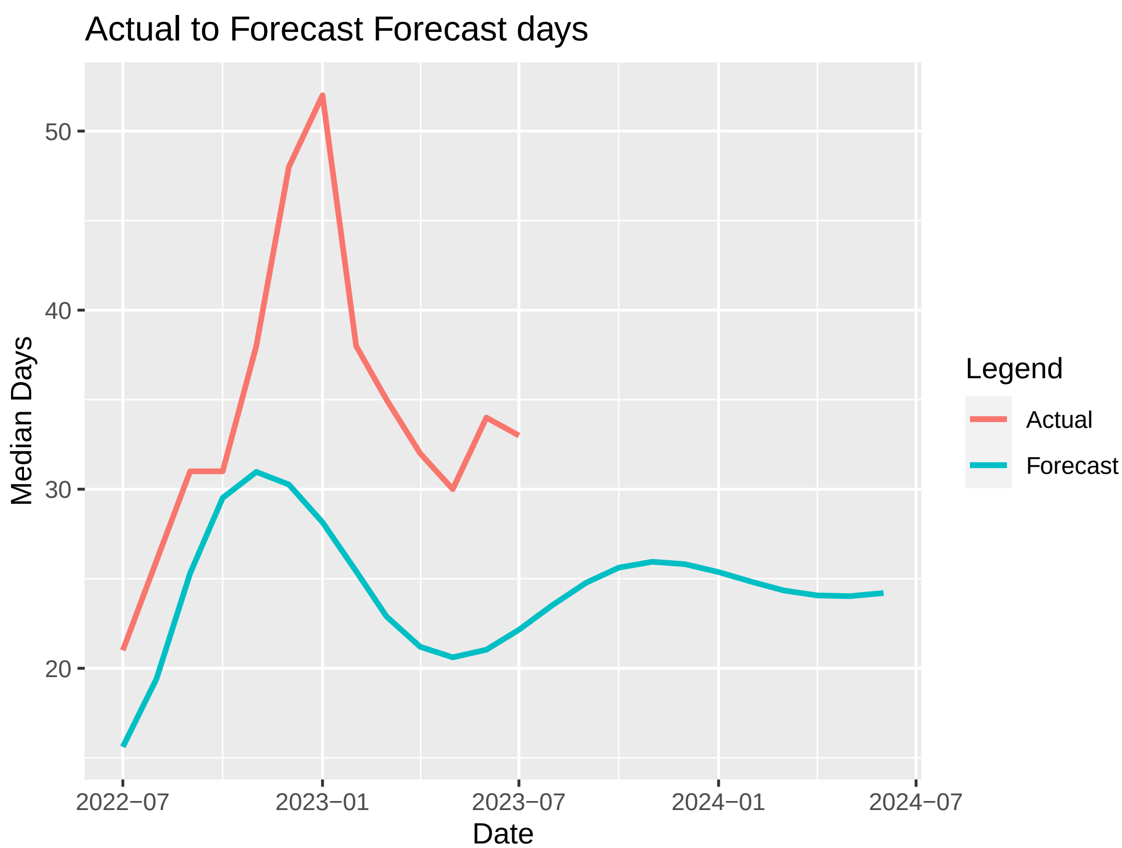
```
# A tsibble: 13 x 3 [1D]
   date       housing_mean housing_sd
   <date>            <dbl>      <dbl>
 1 2023-01-01        4777.       501.
 2 2023-02-01        5160.       977.
 3 2023-03-01        5126.      1414.
 4 2023-04-01        4821.      1769.
 5 2023-05-01        4390.      2024.
 6 2023-06-01        3971.      2191.
 7 2023-07-01        3675.      2297.
 8 2023-08-01        3557.      2370.
 9 2023-09-01        3612.      2426.
10 2023-10-01        3792.      2474.
11 2023-11-01        4027.      2521.
12 2023-12-01        4248.      2572.
13 2024-01-01        4408.      2632.
```

9. Format the `median_days` forecast to {fpp3} specifications

Use `housing_validation`'s `date` variable

```
         date median_days.x housing unemployment median_days.y price_increased
1  2022-07-01      15.60970    5390          3.7            21             136
2  2022-08-01      19.41130    6070          3.6            26             164
3  2022-09-01      25.30440    6350          3.2            31             154
4  2022-10-01      29.51464    7068          3.4            31             128
5  2022-11-01      30.96693    7269          3.2            38             116
6  2022-12-01      30.26227    6627          2.9            48              84
7  2023-01-01      28.15212    6418          3.5            52             154
8  2023-02-01      25.41781    6066          3.6            38             212
9  2023-03-01      22.88924    5638          3.1            35             258
10 2023-04-01      21.19915    5754          2.6            32             280
11 2023-05-01      20.61062    6040          3.2            30             284
12 2023-06-01      21.03586    6420          3.8            34             246
13 2023-07-01      22.15054    6597          3.7            33             200
14 2023-08-01      23.52990      NA           NA            NA              NA
15 2023-09-01      24.77723      NA           NA            NA              NA
16 2023-10-01      25.61912      NA           NA            NA              NA
17 2023-11-01      25.94830      NA           NA            NA              NA
18 2023-12-01      25.81344      NA           NA            NA              NA
19 2024-01-01      25.37116      NA           NA            NA              NA
20 2024-02-01      24.82138      NA           NA            NA              NA
21 2024-03-01      24.34619      NA           NA            NA              NA
22 2024-04-01      24.06722      NA           NA            NA              NA
23 2024-05-01      24.02804      NA           NA            NA              NA
24 2024-06-01      24.20068      NA           NA            NA              NA
   price_decreased pending_listing median_price
1             3216            2184       549999
2             3340            2224       534695
3             3416            2247       529450
4             3796            2010       525000
5             3508            1836       524450
6             1896            1581       519250
7             2220            1737       509763
8             2134            2228       517925
9             2084            2264       527500
10            2196            2268       564025
11            2286            2677       580000
12            2638            2645       591468
13            2716            2472       594900
14              NA              NA           NA
15              NA              NA           NA
16              NA              NA           NA
17              NA              NA           NA
18              NA              NA           NA
19              NA              NA           NA
20              NA              NA           NA
21              NA              NA           NA
22              NA              NA           NA
23              NA              NA           NA
24              NA              NA           NA
```

10. Plot the forecast against the validation data



11. Does the VAR forecast seem accurate?

```
# Accuracy measurements
subset_data <- median_days_fc[median_days_fc$date >= "2022-07-01" & median_days_fc$date <= "2023-07-01"
# Calculate the accuracy measures for the subset data
ME <- mean(subset_data$median_days.x - subset_data$median_days.y, na.rm = TRUE)
RMSE <- sqrt(mean((subset_data$median_days.x - subset_data$median_days.y)^2, na.rm = TRUE))
MAE <- mean(abs(subset_data$median_days.x - subset_data$median_days.y), na.rm = TRUE)
MPE <- mean((subset_data$median_days.x - subset_data$median_days.y) / subset_data$median_days.y, na.rm =
MAPE <- mean(abs(subset_data$median_days.x - subset_data$median_days.y) / subset_data$median_days.y, na
# Display the results
cat(paste("Mean Error (ME):", round(ME, 2)), "\n")
```

```
Mean Error (ME): -10.5
```

```
cat(paste("Root Mean Squared Error (RMSE):", round(RMSE, 2)), "\n")
```

```
Root Mean Squared Error (RMSE): 11.88
```

```r
cat(paste("Mean Absolute Error (MAE):", round(MAE, 2)), "\n")
```

```
Mean Absolute Error (MAE): 10.5
```

```r
cat(paste("Mean Percentage Error (MPE):", round(MPE, 2)), "\n")
```

```
Mean Percentage Error (MPE): -29.17
```

```r
cat(paste("Mean Absolute Percentage Error (MAPE):", round(MAPE, 2)), "\n")
```

```
Mean Absolute Percentage Error (MAPE): 29.17
```

> Answer: "Does the `VAR` forecast seem accurate?" - The VAR forecast appears to systematically under-predicting with relatively high errors, suggesting it may not be highly accurate. Based on the given information: - Mean Error (ME) -10.5: A negative mean error suggests that, on average, the model's forecasts are under-predicting the actual values. The magnitude (10.5) indicates the size of the average under-prediction. - Root Mean Squared Error (RMSE) 11.88: This metric gives more weight to larger errors. An RMSE of 11.88 indicates the average magnitude of the errors. It is somewhat close to the MAE, which means that there might not be a lot of very large individual errors, but it's still slightly higher. - Mean Absolute Error (MAE) 10.5: This metric indicates the average absolute forecast error. In this case, it's equal to the ME, which suggests that the errors might be predominantly in one direction (under-prediction, as mentioned above). - Mean Percentage Error (MPE) -29.17%: The negative value again confirms that the model, on average, tends to under-predict. A value of -29.17% indicates a significant under-prediction, which can be concerning. - Mean Absolute Percentage Error (MAPE) 29.17%: This indicates that the forecast is off by an average of 29.17% from the actual values. Depending on the context, this might be deemed high. For many industries or applications, an MAPE of around 10% or less might be considered good. A value of almost 30% could indicate that the model is not very accurate.

12. Perform co-integration

13. Print summary. What rank do you have evidence for? What is the test statistic? What critical value do you show evidence for at this rank?

```
######################
# Johansen-Procedure #
######################

Test type: trace statistic , with linear trend in cointegration

Eigenvalues (lambda):
[1] 6.226251e-01 4.318784e-01 2.202062e-01 7.313722e-02 2.274726e-02
[6] 8.072951e-19

Values of teststatistic and critical values of test:

           test 10pct  5pct  1pct
r <= 4 |   1.61 10.49 12.25 16.26
```

```
r <= 3 |   6.93 22.76 25.32 30.45
r <= 2 |  24.34 39.06 42.44 48.45
r <= 1 |  63.92 59.14 62.99 70.05
r = 0  | 132.13 83.20 87.31 96.58

Eigenvectors, normalised to first column:
(These are the cointegration relations)

                   housing.l2 unemployment.l2 median_days.l2 price_decreased.l2
housing.l2           1.000000        1.000000      1.0000000          1.0000000
unemployment.l2    -62.336693     -203.301132    385.6824888       -534.4968662
median_days.l2     447.715297      -76.737790   -117.6324624        -12.4991582
price_decreased.l2  -6.035573       -2.624682     -1.1025846         -1.0848765
pending_listing.l2   1.301636        1.107152      0.1842653          0.5672636
trend.l2           144.113544       -2.879399     18.4700815         46.8342678
                   pending_listing.l2      trend.l2
housing.l2                    1.00000      1.000000
unemployment.l2           -5193.93001 -10636.277472
median_days.l2              105.96295    144.618040
price_decreased.l2           23.55777      2.790984
pending_listing.l2          -38.34154    -31.648002
trend.l2                   2512.15664   -151.642492

Weights W:
(This is the loading matrix)

                     housing.l2 unemployment.l2 median_days.l2
housing.d         -1.436925e-02    0.1767928653  -0.0788199617
unemployment.d     4.872902e-05    0.0009323599  -0.0002217436
median_days.d     -9.122043e-04    0.0007389051   0.0008210301
price_decreased.d  3.170851e-02    0.2792681596  -0.0269561210
pending_listing.d  4.094582e-02   -0.0322138320   0.0385401499
                  price_decreased.l2 pending_listing.l2      trend.l2
housing.d              -0.0230093170       7.382314e-04 -5.185675e-17
unemployment.d          0.0002772236       3.459348e-06  1.288386e-19
median_days.d           0.0002870807       1.431888e-06  1.074801e-18
price_decreased.d      -0.0443802632       9.143612e-05 -7.973815e-18
pending_listing.d      -0.0228543351       3.806196e-04 -3.260871e-17
```

Answer: "What rank do you have evidence for?" - We have evidence for r <_ 1 since the test statistic value (63.92) exceeds all the critical values at 10%, 5%, and 1% significance levels (59.14, 62.99, and 70.05). This means we have evidence to suggest that there is at least one cointegrating relationship among the variables.

Answer: "What is the test statistic (i.e., numerical value) of the rank you reported above?" - The test statistic value for the rank r <_ 1 is 63.92.

Answer: "What critical value do you show evidence for at this rank?" - The test statistic value of 63.92 evidence is shown up to the 5% critical value of 62.99, but not for the 1% critical value of 70.05.

14. Convert `VECM` to `VAR` and forecast 13 months out (use code from 8.)

```
          fcst    lower     upper        CI
```

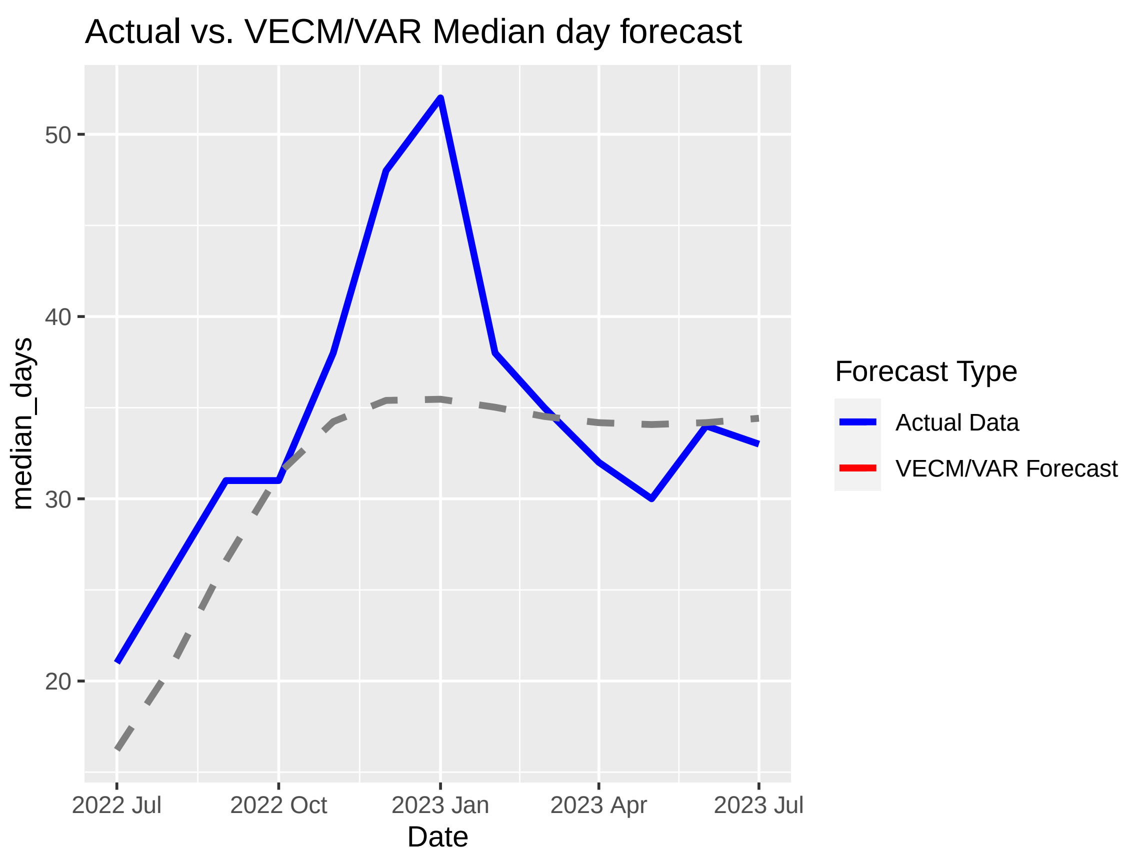
```
 [1,] 4746.046 4251.439  5240.652  494.6064
 [2,] 5327.426 4311.192  6343.660 1016.2340
 [3,] 5668.394 4115.593  7221.196 1552.8014
 [4,] 5806.616 3764.846  7848.386 2041.7698
 [5,] 5826.676 3370.117  8283.235 2456.5590
 [6,] 5804.341 3004.123  8604.558 2800.2174
 [7,] 5789.070 2702.919  8875.221 3086.1509
 [8,] 5803.269 2473.477  9133.062 3329.7923
 [9,] 5849.879 2305.242  9394.515 3544.6365
[10,] 5921.337 2180.424  9662.251 3740.9134
[11,] 6006.961 2081.473  9932.449 3925.4883
[12,] 6097.486 1995.107 10199.865 4102.3791
[13,] 6186.937 1913.392 10460.482 4273.5449
```

15. Format the `median_days` forecast to {fpp3} specifications

Use `housing_validation`'s `date` variable

Use code from 9.

16. Plot the forecast against the validation data and `VAR` forecast



17. Based on the plotted forecasts, which forecast would you prefer? What does the VECM do differently than the VAR? Would you trust this model to forecast into the future?

Answer: “Based on the plotted forecasts, which forecast would you prefer? What does the VECM do differently than the VAR? Would you trust this model to forecast into the future?” - I would trust the VECM forecast as the forecast calculates a long term calculation based on the cointegration compared to the VAR. I would trust this model to forecast into the future as it incorporates a more intercorlated calculation. - Cointegration: VECM is specifically designed for time series that are non-stationary but cointegrated. Statistical tests (like the Johansen test) that the variables are cointegrated, then VECM is a natural choice, as it captures the long-run equilibrium relationship between such variables.