Assignment 6: VAR Models

Rachel Montgomery

2023-10-08

Assignment 6: Unraveling Multivariate Time Series with Vector Autoregression

Time series forecasting is a crucial part of many sectors, from predicting stock prices to anticipating energy demand. In this assignment, we'll explore the power of vector autoregression (VAR) models and learn how to forecast time series data effectively using R.

Section 1: Data Exploration

Our analysis begins by setting up our data and understanding its key components.

Loading and Formatting Data

We start by loading the "nashville_housing" and "housing_validation" dataset and converting them into a tsibble format. Additionally, we create a pandemic dummy variable to account for the impact of the COVID-19 pandemic.

```
#load in nashville housing
nashville_housing <- read.csv("nashville_housing.csv")

# convert date
nashville_housing$date <- yearmonth(nashville_housing$date)

# Convert to `tsibble`
housing_ts <- nashville_housing %>%
  as_tsibble(index = date)

# Set pandemic
housing_ts$pandemic <- rep(0, nrow(housing_ts))
housing_ts$pandemic[
  which(
    as.character(housing_ts$date) == "2020 May"
  ):which(
    as.character(housing_ts$date) == "2021 Jun"
  )
] <- 1
```

For validation purposes, we'll also prepare a validation dataset.

Section 2: Building a VAR Model with {fpp3}

The real fun begins as we attempt to model our multivariate time series data using VAR.

Fitting a VAR Model

Let’s fit a VAR model using the {fpp3} package to the housing data and examine how well it captures the data’s dynamics.

```
# Fit VAR model

fit_var <- housing_ts %>%
  model(
    var_model = VAR(
      vars(housing, unemployment, median_days, price_increased, price_decreased, pending_listin
g, median_price) ~
        xreg(pandemic)
      )
    )
```

Reporting Model Fit

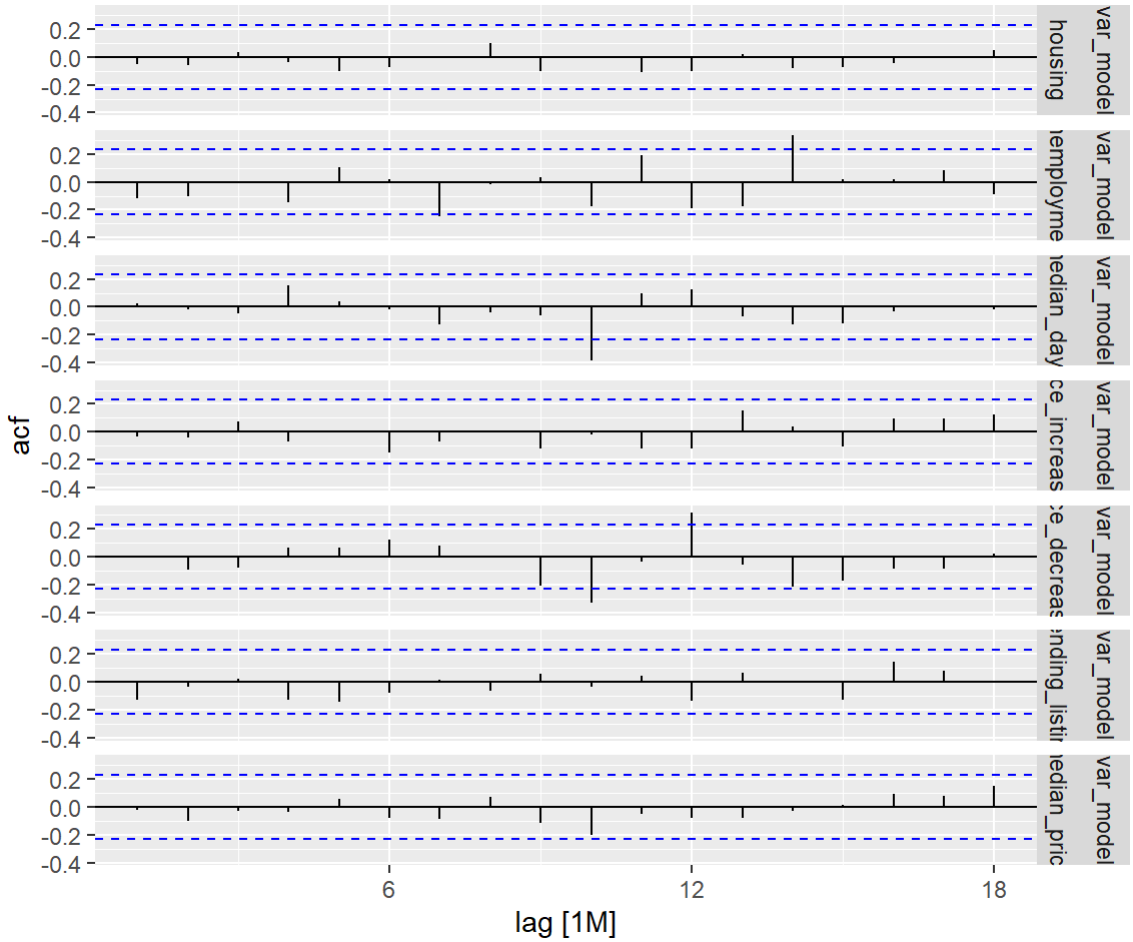
Understanding how well our model fits the data is essential. Let’s inspect the fit.

```
# Report fit
#report(fit_var) ##hiding because output is so long
```

How many lags were used? 5 lags were used in the VAR model.

Autocorrelation Analysis

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



To determine if any autocorrelations are significant, we typically look for bars (lags) that extend beyond the dashed blue lines, which represent the significance level. If a bar crosses this line, it means the autocorrelation for that specific lag is statistically significant.

- housing: significant autocorrelation around lag 12.
- price_increased: significant autocorrelation around lag 12.
- price_decreased: Significant autocorrelation observed around lag 6 and 12.
- pending_listing: Significant autocorrelation around lag 12.

Significant Autocorrelations

Identifying significant autocorrelations in the residuals.

Section 3: Building a VAR Model with {vars}

The heart of this assignment lies in building VAR models to analyze and forecast our multivariate data.

Fitting a VAR Model with {vars}

We will once again model our housing data, but this time using functions from the {vars} package.

```
# VAR model with {vars}

vars_var <- vars::VAR(
  y = housing_ts[,c(
    "housing", "unemployment", "median_days",
    "price_decreased", "pending_listing"
  )],
  exogen = housing_ts[,c("pandemic")],
  type = "none", # same as {fpp3}'s `VAR`
  p = 5 # lag
)

# Make dummy variable matrix
dummat <- matrix(
  rep(0, 2 * 24), nrow = 24,
  dimnames = list(NULL, c("outlier", "pandemic")))
```

Serial Test on Residual Autocorrelations

A serial test can give insights into the independence of residuals.

```
# Perform serial test

# Fit VAR(2)
var_2 <- vars::VAR(
  housing_ts[,-c(1, 9)], p = 2, type = "none",
  exogen = housing_ts[,c(9)]
)
serial.test(var_2, lags.pt = 4, type = "PT.adjusted")
```

```
    Portmanteau Test (adjusted)

data:  Residuals of VAR object var_2
Chi-squared = 151.63, df = 98, p-value = 0.0004178
```

```
# Fit VAR(2) with season
var_2_season <- vars::VAR(
  housing_ts[,-c(1, 9)], p = 2, type = "none",
  exogen = housing_ts[,c(9)], season = 12
)
serial.test(var_2_season, lags.pt = 10, type = "PT.adjusted")
```

```
    Portmanteau Test (adjusted)

data:  Residuals of VAR object var_2_season
Chi-squared = 501.17, df = 392, p-value = 0.0001533
```

```
## Set 'lags.pt' to `4`
## Set 'type' to "PT.adjusted"
```

Question: Interpret the p-value from the serial test. What implications does it have for our model?

Forecasting with VAR

Let's use our VAR model to predict future values.

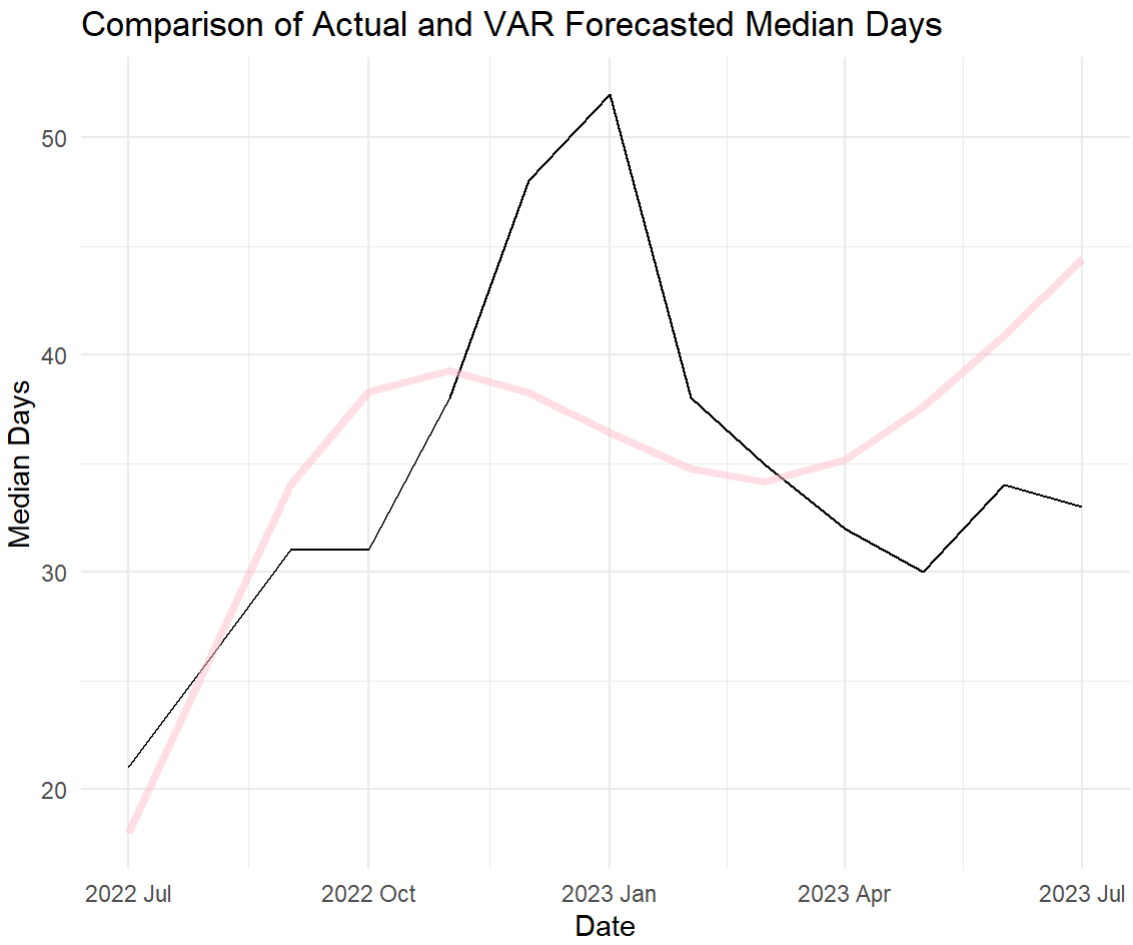
```
# dummy matrix
dummat <- matrix(
  rep(0, 13), nrow = 13,
  dimnames = list(NULL, c("pandemic"))
)

# forecast
var_fc <- predict(var_2, n.ahead = 13, dumvar = dummat)
```

Reformatting forecast

Now, let's format and plot our forecast against the validation data.

Plot the forecast against the validation data

```
Plot variable not specified, automatically selected `.vars = .mean`
```



Trustworthiness of VAR forecast

By visually examining the plot, here's an interpretation:

- **Beginning to late 2022**: The VAR forecast (pink line) closely follows the actual `median_days` (black line), suggesting that the model's forecast was quite accurate for this period.
- **Late 2022 to early 2023**: The actual values spike significantly while the forecasted values only show a mild increase. This suggests that the VAR model did not capture this particular behavior or event that led to the spike in the actual data.
- **Mid 2023 onwards**: Post the spike, the actual values decrease and then stabilize. The VAR model seems to anticipate this decline but not to the same extent as the actual data. The forecast then appears to mildly underestimate the actual `median_days` values.

In conclusion, the VAR forecast appears reasonably accurate at the beginning of the period. However, it missed the sharp spike and seems to mildly underestimate the data towards the end.

Section 4: VECM Models

VECM (Vector Error Correction Models) is another method for handling multivariate time series. Let's explore its capabilities.

Cointegration Analysis

We perform the Johansen Procedure to understand the relationships between variables and determine the rank of evidence. Cointegration can help us understand the long-term relationships between our variables.

```
# Cointegration
co_test <- ca.jo(
  # variables
  x = housing_ts[,c(
    "housing", "unemployment", "median_days",
    "price_decreased", "pending_listing"
  )],
  type = "trace", # tends to be more conservative
  K = 5, # lag -- same as your VAR model
  spec = "longrun", # generally use "longrun"
  ecdet = "trend", # trend-stationary
  # exogeneous dummy variables
  dumvar = housing_ts[,c("pandemic")]
)
```

Discussion of Cointegration Analysis

Now, let's discuss the results.

```
co_summ <- summary(co_test)
```

To determine the rank for which we have evidence of cointegration:

1. To determine the rank for which we have evidence of cointegration:
2. Start from the bottom (r = 0) and move upwards. Compare the test statistic to the critical values.

The results are:

- r $\leq$ 4: The test statistic is 0.06, which is less than the critical values at all significance levels (10.49, 12.25, 16.26). Therefore, we do not reject the hypothesis that r ≤ 4.
- r $\leq$ 3: The test statistic is 6.59, which is less than the critical values at all significance levels (22.76, 25.32, 30.45). So, we do not reject the hypothesis that r ≤ 3.
- r $\leq$ 2: The test statistic is 24.16, which is less than the critical values at all significance levels (39.06, 42.44, 48.45). So, we do not reject the hypothesis that r ≤ 2.
- r $\leq$ 1: The test statistic is 51.79, which is less than the critical values at the 1% significance level (70.05) but is below the critical values at the 5% and 10% significance levels. So, we reject the hypothesis that r ≤ 1 at the 10% and 5% significance levels.
- r = 0: The test statistic is 94.60, which is greater than the critical values at all significance levels (83.20, 87.31, 96.58). We can reject the hypothesis that r = 0 at the 10% and 5% significance levels but not at the 1% significance level.

Conclusion: We have evidence for a rank of 1 at the 5% significance level. The test statistic for this rank is 51.79. The critical value for this rank at the 5% significance level is 62.99.

Converting to VAR and Additional Forecasting

We convert our VECM to VAR and conduct further forecasting. We compare the VECM and VAR forecasts to make informed decisions.

```
# Convert VECM to VAR
vecm <- vars::vec2var(co_test, r = 2)

# Make dummy variable matrix
dummy_var_matrix <- matrix(
  rep(0, 1 * 13), nrow = 13,
  dimnames = list(NULL, c("pandemic")))

# Forecast
vecm_forecast <- predict(vecm, n.ahead = 13, dumvar = dummy_var_matrix)
```

Reformatting forecast

Use housing_validation's date variable, we'll format the median_days forecast to {fpp3} specifications.

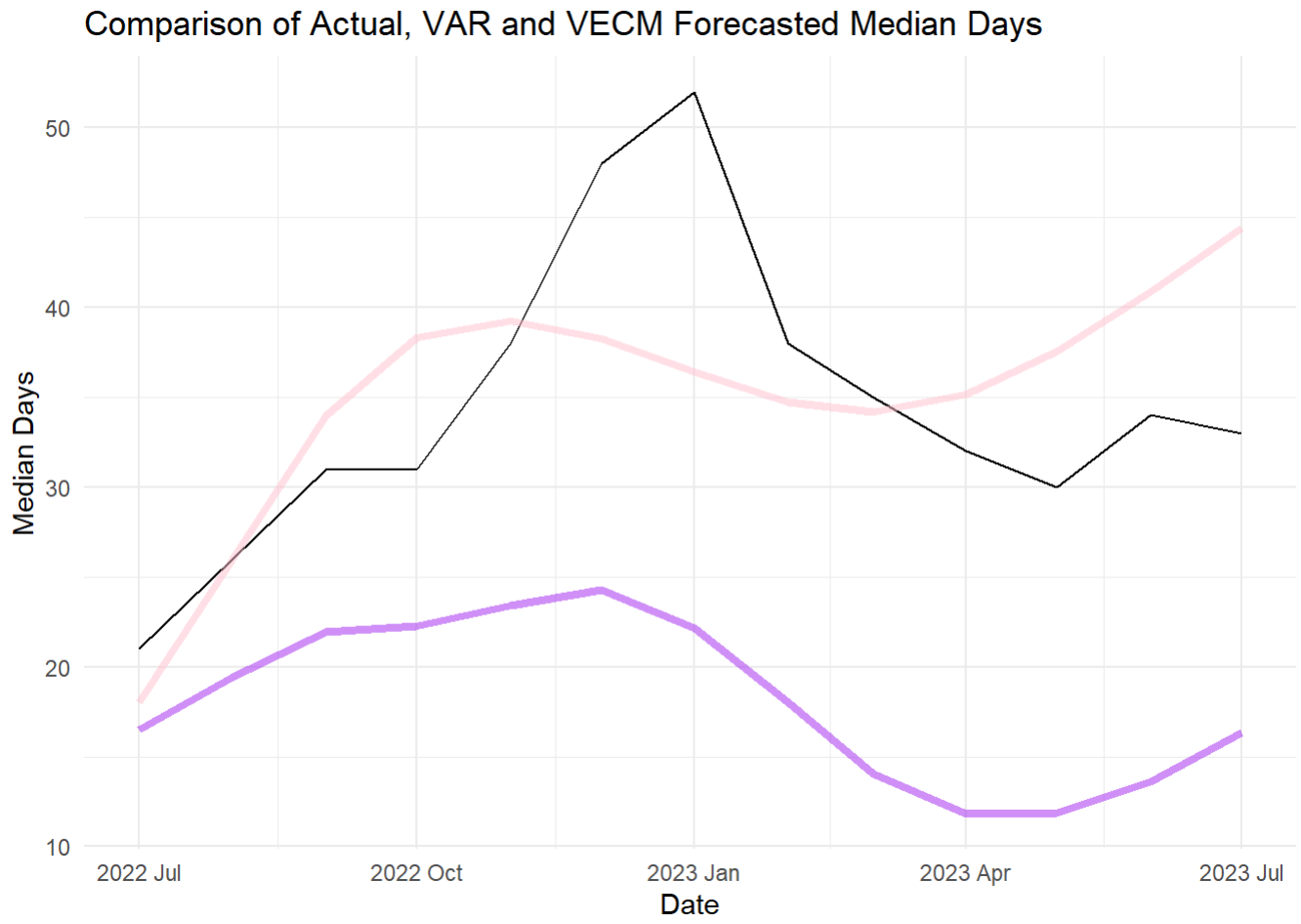
```
# Get forecast values
fc_median_days <- vecm_forecast$fcst$median_days

# Set up forecast as {fpp3} does
vecm_fc <- data.frame(
  .model = "VECM",
  date = validation_ts$date,
  median_days = distributional::dist_normal(
    mean = fc_median_days[,"fcst"],
    sd = fc_median_days[,"CI"]
  ),
  .mean = fc_median_days[,"fcst"]
) %>% as_tsibble(index = date)

# Add "housing" to dimnames
dimnames(vecm_fc$median_days) <- "median_days"
```

Plot the VECM forecast against the validation data and `VAR` forecast

With our VAR models in place, it's time to evaluate their performance and use them for forecasting.

```
validation_ts %>%
  autoplot(median_days) +
  autolayer(var_fc, alpha = 0.5, size = 1.5, color = "pink") +
  autolayer(vecm_fc, alpha = 0.5, size = 1.5, color = "purple") +
  labs(title = "Comparison of Actual, VAR and VECM Forecasted Median Days",
       x = "Date",
       y = "Median Days") +
  theme_minimal()
```

```
## Plot variable not specified, automatically selected `.vars = .mean`
## Plot variable not specified, automatically selected `.vars = .mean`
```



Section 4: Model Comparison and Conclusion

Comparing Forecasts

Based on the plotted forecasts, I prefer the VECM model.

- The VECM forecast (in purple) seems to capture the trend of the actual data more closely than the VAR forecast (in pink), especially around the peak observed around January 2023.
- The VECM also appears to be less volatile than the VAR model.

Conclusion