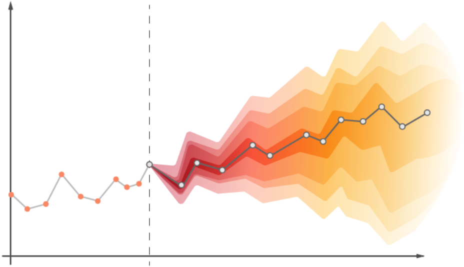
TSLM + ARIMA Error Models

DS-5740 Advanced Statistics



Overview

Overview: Week 5

Overview | Week 5

Goals for the Week

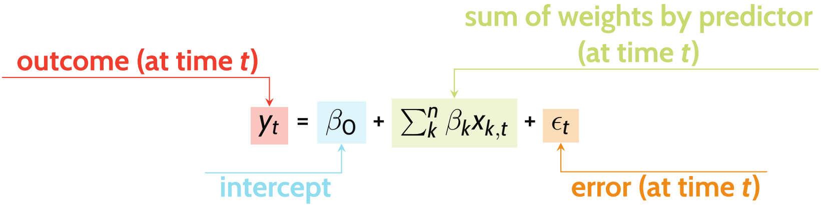
- Cover dynamic regression models (TSLM with ARIMA errors)
- Build dynamic model with multiple predictors
- Forecast number of houses on market in Nashville area

Dynamic Regression

Dynamic Regression | TSLM

$$\overbrace{y_t}^{\text{outcome (at time } t)} = \underbrace{\beta_0}_{\text{intercept}} + \overbrace{\sum_k^n \beta_k x_{k,t}}^{\text{sum of weights by predictor (at time } t)} + \underbrace{\epsilon_t}_{\text{error (at time } t)}$$

Dynamic Regression | TSLM



sum of weights by predictor (at time *t*)

outcome (at time *t*)

$$y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \epsilon_t$$

intercept

error (at time *t*)

- Pro: allows for external variables to be model
- Con: does not allow for time series dynamics (e.g., lagged time points and errors)

Dynamic Regression | ARIMA

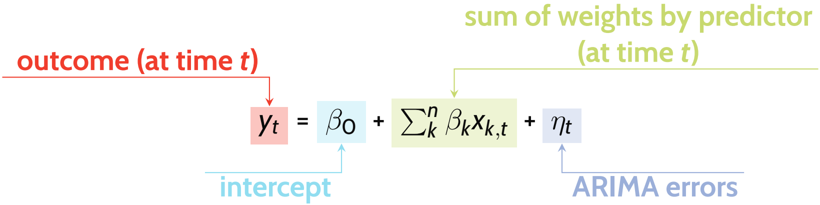
$$\overbrace{y_t}^{\text{outcome}} = \underbrace{c}_{\text{constant}} + \overbrace{\phi_1 y_{t-1} + \ldots + \phi_p y_{t-p}}^{\text{autoregressive}} + \overbrace{\theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q}}^{\text{moving average}} + \underbrace{\epsilon_t}_{\text{error}}$$

Dynamic Regression | ARIMA

$$\overset{\text{outcome}}{y_t} = \underset{\text{constant}}{c} + \overbrace{\phi_1 y_{t-1} + \ldots + \phi_p y_{t-p}}^{\text{autoregressive}} + \overbrace{\theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q}}^{\text{moving average}} + \underset{\text{error}}{\epsilon_t}$$

- Pro: allows for time series dynamics (e.g., lagged time points and errors)
- Con: does not allow for external predictors

Dynamic Regression | TSLM with ARIMA Errors



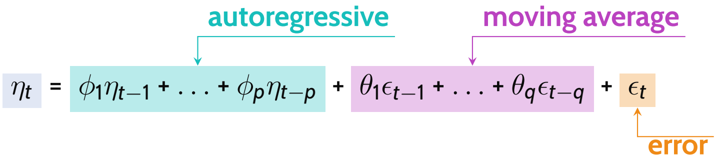
sum of weights by predictor (at time *t*)

outcome (at time *t*)

$$y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \eta_t$$

intercept

ARIMA errors



autoregressive

moving average

$$\eta_t = \phi_1 \eta_{t-1} + \ldots + \phi_p \eta_{t-p} + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t$$

error

Dynamic Regression | TSLM with ARIMA Errors

$$\overbrace{y_t}^{\text{outcome (at time } t\text{)}} = \underbrace{\beta_0}_{\text{intercept}} + \overbrace{\textstyle\sum_k^n \beta_k x_{k,t}}^{\text{sum of weights by predictor (at time } t\text{)}} + \underbrace{\eta_t}_{\text{ARIMA errors}}$$

$$\eta_t = \overbrace{\phi_1 \eta_{t-1} + \ldots + \phi_p \eta_{t-p}}^{\text{autoregressive}} + \overbrace{\theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q}}^{\text{moving average}} + \underbrace{\epsilon_t}_{\text{error}}$$

- Pro: allows for time series dynamics (e.g., lagged time points and errors)
- Pro: does not allow for external predictors

Dynamic Regression | ARIMA Errors vs. ARIMAX

Recall last week:

TSLM with SARIMA errors

```
Series: housing
Model: LM w/ ARIMA(1,1,1)(0,0,1)[12] errors

Coefficients:
         ar1     ma1    sma1      outlier
      0.4349  0.3094  0.3642   -1163.1577
s.e.  0.1763  0.1837  0.1490     143.3911

sigma^2 estimated as 110312:  log likelihood=-512.01
AIC=1034.03   AICc=1034.95   BIC=1045.34
```

SARIMAX

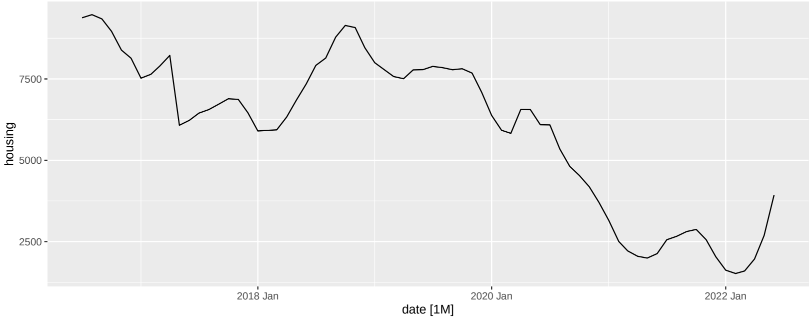
```
Call:
arimax(x = housing_ts$housing, order = c(1, 1, 1), seasonal = list(order = c(0,
    0, 1), period = 12), xreg = housing_ts$outlier)

Coefficients:
         ar1     ma1    sma1         xreg
      0.4349  0.3094  0.3642   -1163.1577
s.e.  0.1763  0.1837  0.1490     143.3911

sigma^2 estimated as 104096:  log likelihood = -512.01,  aic = 1032.03
```

Dynamic Regression

Nashville Area Housing

Example: Number of Houses on the Market

- FRED: Nashville-Davidson–Murfreesboro–Franklin, TN



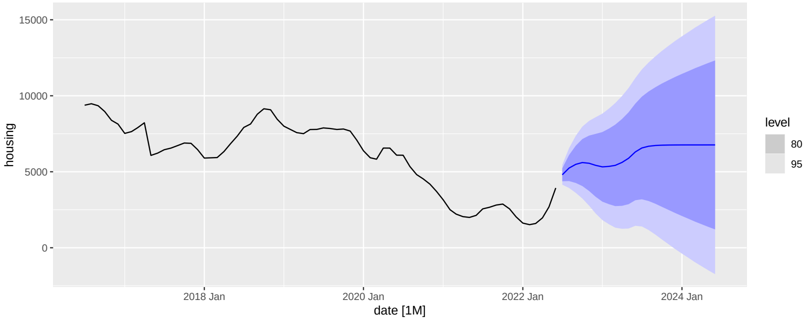
Dynamic Regression | Nashville Area Housing

```r
# Outlier dummy variable
housing_ts$outlier <- 0

# Set outlier to 1
housing_ts$outlier[which.min(difference(housing_ts$housing))] <- 1

# Final model
fit <- housing_ts %>%
  model(sarima_best = ARIMA(housing ~ outlier))

# Report fit
report(fit)
```

```
Series: housing
Model: LM w/ ARIMA(1,1,1)(0,0,1)[12] errors

Coefficients:
         ar1     ma1    sma1      outlier
      0.4349  0.3094  0.3642  -1163.1577
s.e.  0.1763  0.1837  0.1490    143.3911

sigma^2 estimated as 110312:  log likelihood=-512.01
AIC=1034.03   AICc=1034.95   BIC=1045.34
```

Dynamic Regression | Nashville Area Housing

```r
# Forecast next two years
new_two_years <- new_data(housing_ts, 24) %>% mutate(outlier = 0)
fc_two_years <- fit %>% forecast(new_data = new_two_years)

# Plot forecast
housing_ts %>% autoplot(housing) + autolayer(fc_two_years)
```



Can we make a better forecast?

Dynamic Regression | Nashville Area Housing

Example: Number of Houses on the Market

- FRED: Nashville-Davidson–Murfreesboro–Franklin, TN

Other variables?

Dynamic Regression | Nashville Area Housing

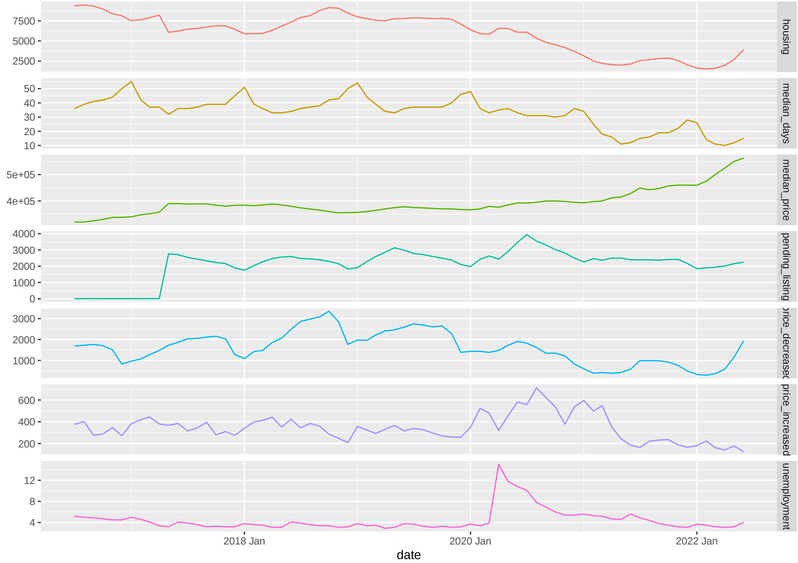
Example: Number of Houses on the Market

- FRED: Nashville-Davidson–Murfreesboro–Franklin, TN

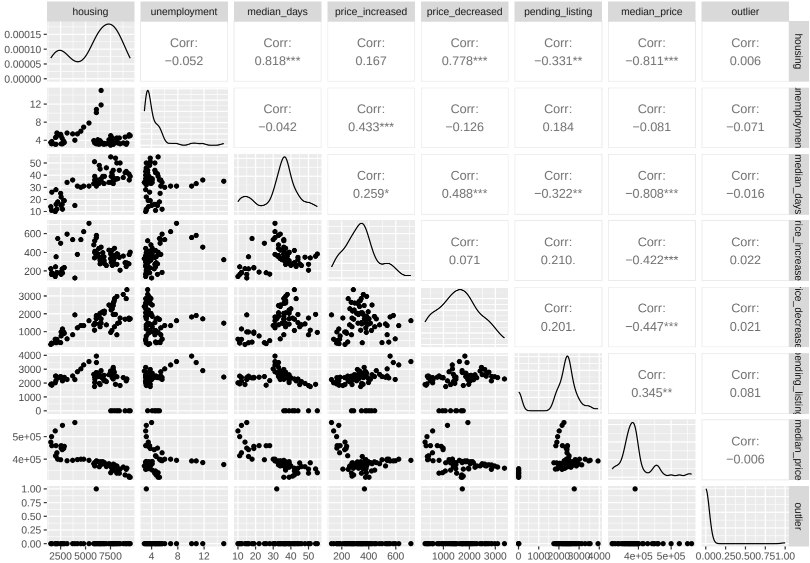
Other variables?

- Median days on market
- Median price
- Number of houses pending sale
- Number of houses decreased in price
- Number of houses increased in price
- Unemployment (Tennessee)
- Additional outliers – pandemic?

Dynamic Regression | Nashville Area Housing

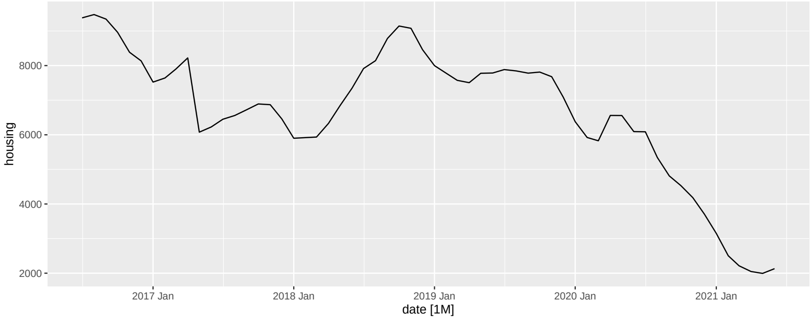


Dynamic Regression | Nashville Area Housing



Dynamic Regression | Nashville Area Housing

```r
# Set up training and testing indices
train <- 1:which(as.character(housing_ts$date) == "2021 Jun")

# Initialize training and testing data
housing_train <- housing_ts[train,]
housing_test <- housing_ts[-train,]

# Plot housing
housing_train %>% autoplot(housing)
```



Dynamic Regression | TSLM

Time Series Linear Model

Dynamic Regression | TSLM

```
# Fit linear model
fit_tslm <- housing_train %>%
  model(tslm = TSLM(
      housing ~ unemployment + median_days + price_increased +
      price_decreased + pending_listing + median_price + outlier))

# Report fit
report(fit_tslm)
```

```
Series: housing
Model: TSLM

Residuals:
     Min       1Q   Median       3Q      Max
-782.178 -217.483    5.184  224.990  751.783

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      1.655e+04  2.347e+03   7.053 4.08e-09 ***
unemployment     6.942e+01  2.467e+01   2.814  0.00689 **
median_days      3.791e+01  7.963e+00   4.762 1.58e-05 ***
price_increased -6.492e-01  5.332e-01  -1.218  0.22888
price_decreased  1.597e+00  1.182e-01  13.511  < 2e-16 ***
pending_listing -3.480e-01  1.226e-01  -2.838  0.00646 **
median_price    -3.584e-02  5.859e-03  -6.118 1.25e-07 ***
outlier          5.044e+02  3.727e+02   1.353  0.18176
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 365.5 on 52 degrees of freedom
Multiple R-squared: 0.9682, Adjusted R-squared: 0.9639
F-statistic: 226.2 on 7 and 52 DF, p-value: < 2.22e-16
```

Dynamic Regression | TSLM

```r
# Multicolinearity?
fit <- lm(
  housing ~ unemployment + median_days + price_increased +
      price_decreased + pending_listing + median_price + outlier,
  data = housing_train
)

# VIF
regclass::VIF(fit)
```

```
   unemployment     median_days price_increased price_decreased pending_listing
       1.378784        2.016963        1.458180        3.169702        7.034661
   median_price         outlier
       7.839214        1.022492
```

```r
# Coefficients
round(coefficients(fit), 3)[c("pending_listing", "median_price")]
```

```
pending_listing    median_price
         -0.348          -0.036
```

Dynamic Regression | TSLM

```r
# Multicolinearity?
fit <- lm(
  housing ~ unemployment + median_days + price_increased +
      price_decreased + pending_listing + outlier,
  data = housing_train
)

# VIF
regclass::VIF(fit)
```

```
   unemployment     median_days price_increased price_decreased pending_listing
       1.290513        1.582251        1.398928        1.602349        1.734942
        outlier
       1.022064
```

Dynamic Regression | TSLM

```r
# Remove price increases
fit_increase <- housing_train %>%
  model(
    tslm_all = TSLM(
      housing ~ unemployment + median_days + price_increased +
      price_decreased + pending_listing + outlier
    ),
    tslm_sig = TSLM(
      housing ~ unemployment + median_days +
      price_decreased + pending_listing + outlier
    )
  )

# Report fit
glance(fit_increase) %>%
  select(.model, AIC, AICc, BIC)
```

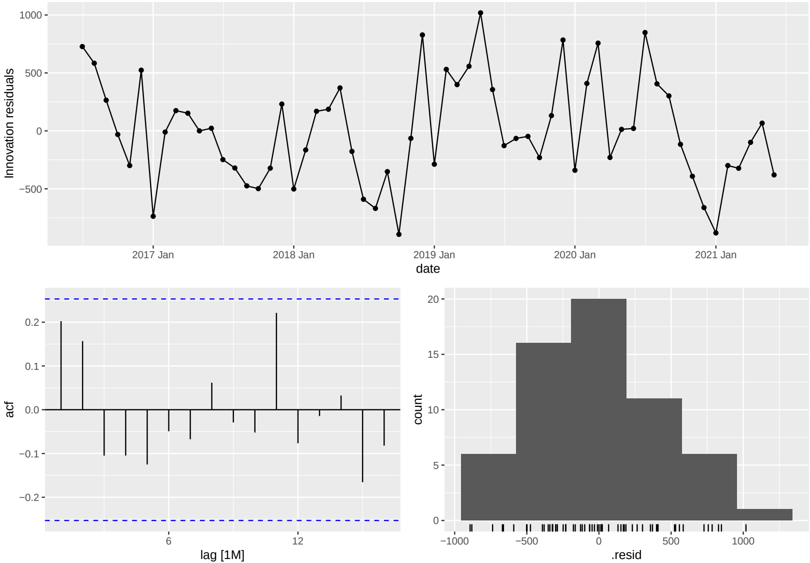
```
# A tibble: 2 x 4
  .model     AIC  AICc   BIC
  <chr>    <dbl> <dbl> <dbl>
1 tslm_all  748.  751.  765.
2 tslm_sig  746.  748.  761.
```

Dynamic Regression | TSLM

```r
# Check best fit
fit_tslm <- housing_train %>%
  model(
    tslm_sig = TSLM(
      housing ~ unemployment + median_days +
      price_decreased + pending_listing + outlier
    )
  )

# Report fit
report(fit_tslm)
```

```
Series: housing
Model: TSLM

Residuals:
    Min      1Q  Median      3Q     Max
-893.28 -320.92  -39.42  315.42 1018.56

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     2364.28418  413.02906   5.724 4.70e-07 ***
unemployment     107.71463   29.23658   3.684 0.000532 ***
median_days       60.53365    9.06723   6.676 1.38e-08 ***
price_decreased    2.10462    0.10394  20.248  < 2e-16 ***
pending_listing   -0.99884    0.07512 -13.297  < 2e-16 ***
outlier          550.95964  479.46898   1.149 0.255578
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 470.3 on 54 degrees of freedom
Multiple R-squared: 0.9453, Adjusted R-squared: 0.9402
F-statistic: 186.7 on 5 and 54 DF, p-value: < 2.22e-16
```

Dynamic Regression | TSLM



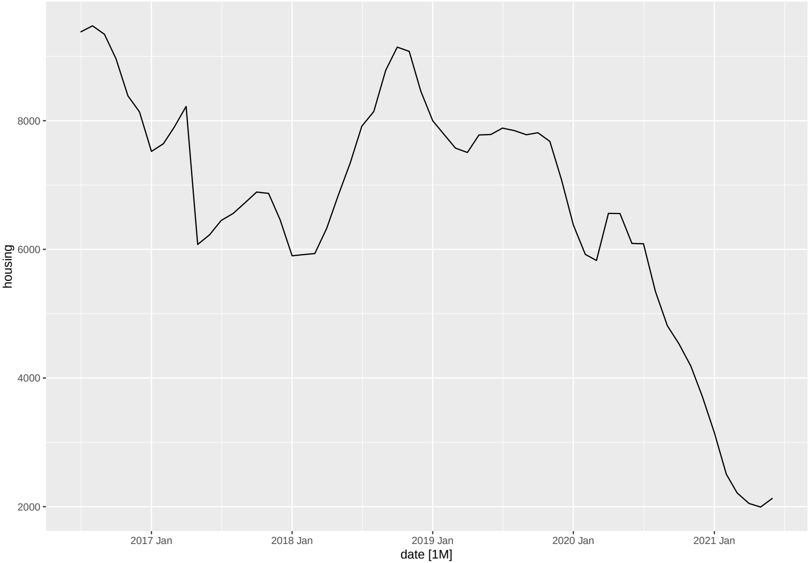
Dynamic Regression | TSLM

```
# Ljung-Box
fit_tslm %>% augment() %>%
  features(.innov, ljung_box, lag = 12, dof = 5)
```

```
# A tibble: 1 x 3
  .model   lb_stat lb_pvalue
  <chr>      <dbl>     <dbl>
1 tslm_sig    19.0   0.00812
```

$p < 0.05$: significantly different from white noise

Dynamic Regression | TSLM

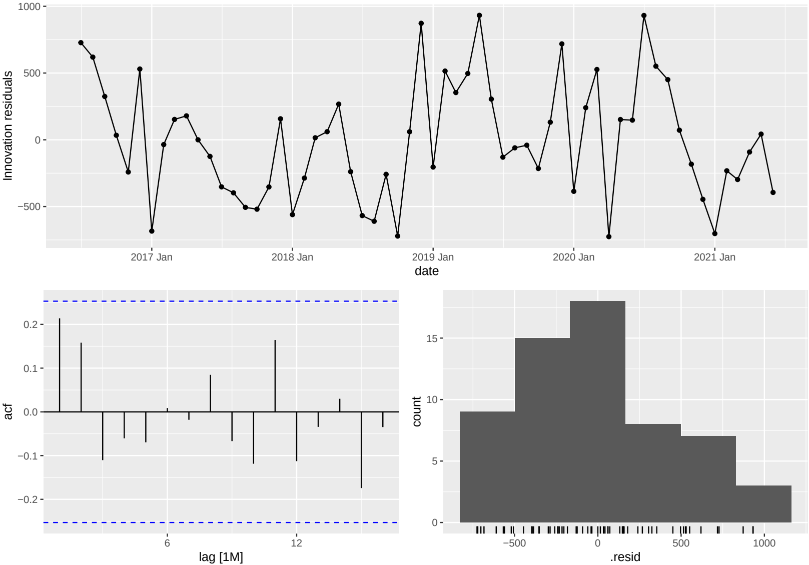


Dynamic Regression | TSLM

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

```
Series: housing
Model: TSLM

Residuals:
    Min      1Q  Median      3Q     Max
-725.53 -311.31  -17.91  276.64  932.24

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     2760.14945  450.77338   6.123 1.15e-07 ***
unemployment     135.72702   31.90622   4.254 8.58e-05 ***
median_days       52.40100    9.77082   5.363 1.83e-06 ***
price_decreased    1.95711    0.12637  15.487  < 2e-16 ***
pending_listing   -0.92351    0.08276 -11.159 1.58e-15 ***
outlier          372.63582  476.26269   0.782    0.437
pandemic        -536.57943  274.66178  -1.954    0.056 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 458.5 on 53 degrees of freedom
Multiple R-squared: 0.949,   Adjusted R-squared: 0.9432
F-statistic: 164.3 on 6 and 53 DF, p-value: < 2.22e-16
```

Dynamic Regression | TSLM



Dynamic Regression | TSLM

```r
# Ljung-Box
fit_tslm %>% augment() %>%
  features(.innov, ljung_box, lag = 12, dof = 6)
```

```
# A tibble: 1 x 3
  .model   lb_stat lb_pvalue
  <chr>      <dbl>     <dbl>
1 tslm_sig    16.4    0.0118
```

$p < 0.05$: significantly different from white noise

Dynamic Regression | ARIMA

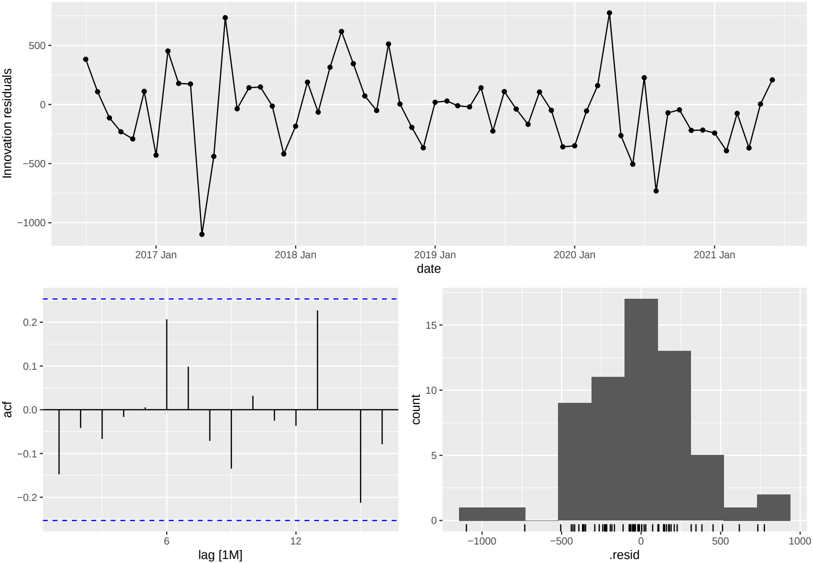
ARIMA

Dynamic Regression | ARIMA

```r
# Fit ARIMA model
fit_arima <- housing_train %>%
  model(arima = ARIMA(
    housing ~ outlier + pandemic
  ))

# Report fit
report(fit_arima)
```

```
Series: housing
Model: LM w/ ARIMA(2,0,0)(0,0,1)[12] errors

Coefficients:
         ar1      ar2    sma1     outlier   pandemic  intercept
      1.5563  -0.5858  0.3268  -1166.0005  -211.2371   6475.317
s.e.  0.1089   0.1163  0.1955    169.1823   278.1560   1551.560

sigma^2 estimated as 119628:  log likelihood=-435.6
AIC=885.2   AICc=887.36   BIC=899.86
```

Dynamic Regression | ARIMA



Dynamic Regression | ARIMA

```r
# Ljung-Box
fit_arima %>% augment() %>%
  features(.innov, ljung_box, lag = 12, dof = 5)
```

```
# A tibble: 1 x 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 arima     7.69     0.361
```

$p > 0.05$: not significantly different from white noise

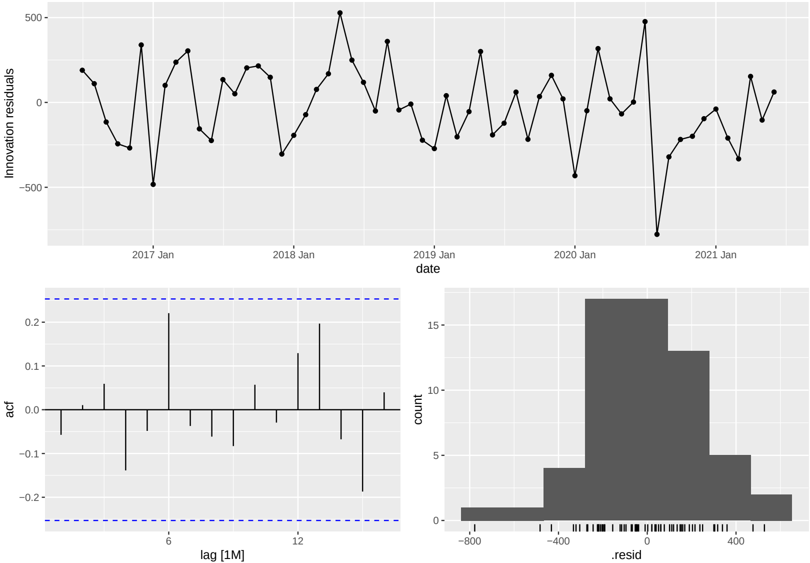
TSLM with ARIMA Errors

Dynamic Regression | TSLM with ARIMA Errors

```r
# Fit TSLM with ARIMA errors
fit_dynamic <- housing_train %>%
  model(
    dynamic = ARIMA(
      housing ~ unemployment + median_days + price_decreased +
        pending_listing + outlier + pandemic
    )
  )

# Report fit
report(fit_dynamic)
```

```
Series: housing
Model: LM w/ ARIMA(2,0,1)(1,0,0)[12] errors

Coefficients:
         ar1      ar2      ma1    sar1  unemployment  median_days
      1.9239  -0.9365  -0.6950  0.5190       62.0089     -13.9496
s.e.  0.0806   0.0797   0.1962  0.1294       19.1521      12.2262
      price_decreased  pending_listing    outlier  pandemic  intercept
               0.5472          -0.6461  -365.3717  337.4429   6771.728
s.e.           0.1916           0.1087   209.8392  251.7531   1421.673

sigma^2 estimated as 70875:  log likelihood=-418.49
AIC=860.99   AICc=867.62   BIC=886.12
```

Dynamic Regression | ARIMA



Dynamic Regression | ARIMA

```r
# Ljung-Box
fit_dynamic %>% augment() %>%
  features(.innov, ljung_box, lag = 12, dof = 10)
```

```
# A tibble: 1 x 3
  .model  lb_stat lb_pvalue
  <chr>     <dbl>     <dbl>
1 dynamic    6.69    0.0353
```

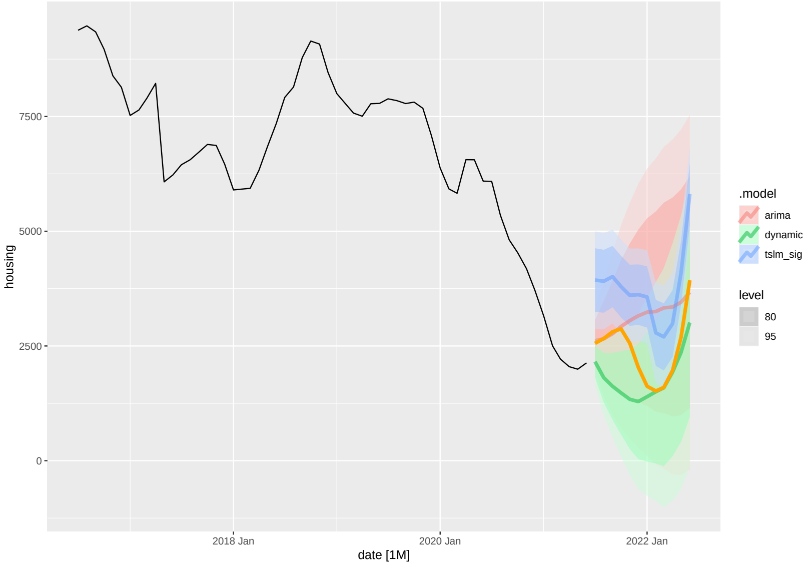
$p < 0.05$: significantly different from white noise

Forecast All Models

Dynamic Regression | Forecasting

```r
# Update test data with outlier and pandemic
housing_test <- housing_test %>%
  mutate(outlier = 0, pandemic = 0)

# Combine all models
all_models <- housing_train %>%
  model(
    tslm_sig = TSLM(
      housing ~ unemployment + median_days + price_decreased +
        pending_listing + outlier + pandemic
    ),
    arima = ARIMA(
      housing ~ outlier + pandemic
    ),
    dynamic = ARIMA(
      housing ~ unemployment + median_days + price_decreased +
        pending_listing + outlier + pandemic
    )
  )

# Forecast models
fc <- all_models %>% forecast(new_data = housing_test)
```

Dynamic Regression | Forecasting



Dynamic Regression | Forecasting

```r
# Point estimates
fc %>% accuracy(housing_test) %>%
  select(.model, RMSE, ME, MAE)
```

```
# A tibble: 3 x 4
  .model     RMSE     ME   MAE
  <chr>     <dbl>  <dbl> <dbl>
1 arima     1028.  -721.  772.
2 dynamic    780.   611.  611.
3 tslm_sig 1370. -1333. 1333.
```

```r
# Distributional estimates
fc %>% accuracy(
    housing_test,
    list(crps = CRPS)
  )
```

```
# A tibble: 3 x 3
  .model   .type  crps
  <chr>    <chr> <dbl>
1 arima    Test   558.
2 dynamic  Test   516.
3 tslm_sig Test  1036.
```

Dynamic Regression | Next Two Years

Forecast Next Two Years

Dynamic Regression | Next Two Years

Add outlier and pandemic variables

```r
# Outlier dummy variable
housing_ts$outlier <- 0

# Set outlier to 1
housing_ts$outlier[
  which.min(difference(housing_ts$housing))
] <- 1

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

Dynamic Regression | Next Two Years

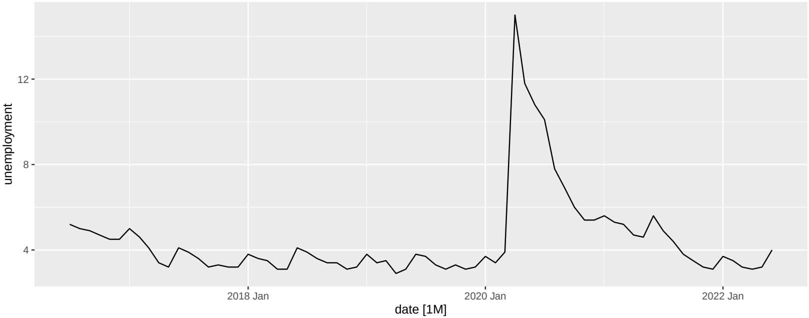
Add outlier and pandemic variables

```r
# Combine all models
all_models <- housing_ts %>%
  model(
    tslm_sig = TSLM(
      housing ~ unemployment + median_days + price_decreased +
        pending_listing + outlier + pandemic
    ),
    arima = ARIMA(
      housing ~ outlier + pandemic
    ),
    dynamic = ARIMA(
      housing ~ unemployment + median_days + price_decreased +
        pending_listing + outlier + pandemic
    )
  )
```

Dynamic Regression | Next Two Years

Create new predictor values

```
# Plot
housing_ts %>% autoplot(unemployment)
```



Dynamic Regression | Next Two Years

Create first six months of pandemic variable

```r
# Pandemic variable
unemployment_ts$pandemic <- 0

# April 2020-
unemployment_ts$pandemic[46:51] <- 6:1
```

Dynamic Regression | Next Two Years

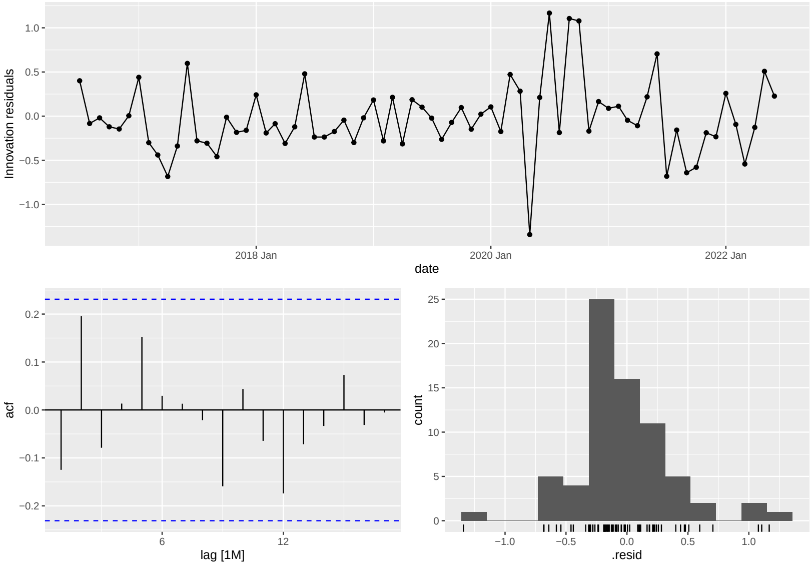
Fit ARIMA model

```
# Fit ARIMA
fit_unemployment <- unemployment_ts %>%
  model(
    arima = ARIMA(unemployment),
    arima_covariate = ARIMA(unemployment ~ pandemic)
  )

# Report fit
glance(fit_unemployment)
```

```
# A tibble: 2 x 8
  .model          sigma2 log_lik   AIC  AICc   BIC ar_roots    ma_roots
  <chr>            <dbl>   <dbl> <dbl> <dbl> <dbl> <list>      <list>
1 arima            1.92    -125.  256.  256.   263. <cpl [1]>  <cpl [0]>
2 arima_covariate  0.176   -40.3  92.6  93.9  106. <cpl [25]> <cpl [0]>
```

```
# Select pandemic model
fit_unemployment <- fit_unemployment %>%
  select(arima_covariate)
```

Check residuals



Dynamic Regression | Next Two Years

Ljung-Box test

```r
# Ljung-Box
fit_unemployment %>% augment() %>%
  features(.innov, ljung_box, lag = 12, dof = 4)
```

```
# A tibble: 1 x 3
  .model           lb_stat lb_pvalue
  <chr>              <dbl>     <dbl>
1 arima_covariate     9.65     0.290
```

$p > 0.05$: not significantly different from white noise

Dynamic Regression | Next Two Years

Create new data

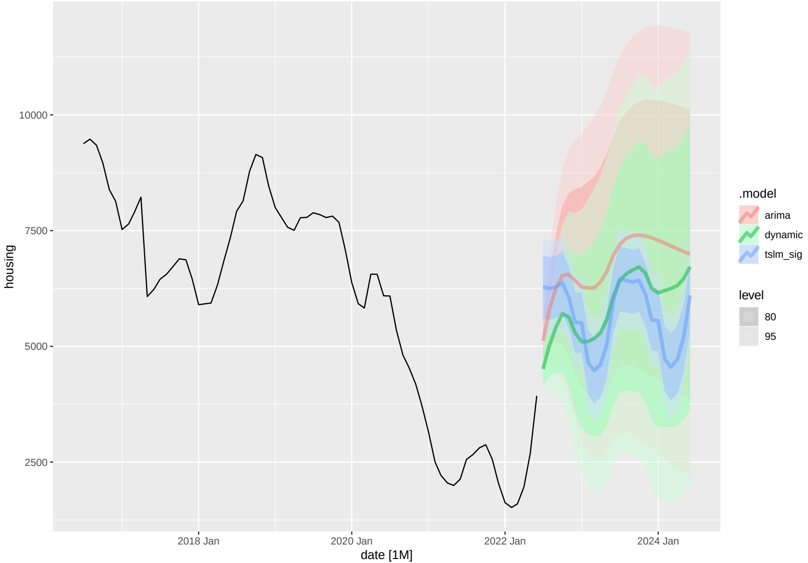
```
## New data
new_unemployment <- new_data(housing_test, n = 24)

## Add outlier
new_unemployment$pandemic <- 0

## Forecast
fc_unemployment <- fit_unemployment %>%
  forecast(new_data = new_unemployment)
```

Dynamic Regression | Next Two Years

Repeat for the rest of the variables

```r
# Make new data
new_final <- new_data(housing_ts, n = 24)

# Add variables
new_final <- new_final %>%
  mutate(
    unemployment = fc_unemployment$.mean,
    median_days = fc_median_days$.mean,
    price_decreased = fc_price_decreased$.mean,
    pending_listing = fc_pending_listing$.mean,
    outlier = 0,
    pandemic = 0
  )

# Forecast models
fc <- all_models %>% forecast(new_data = new_final)
```

Dynamic Regression | Next Two Years



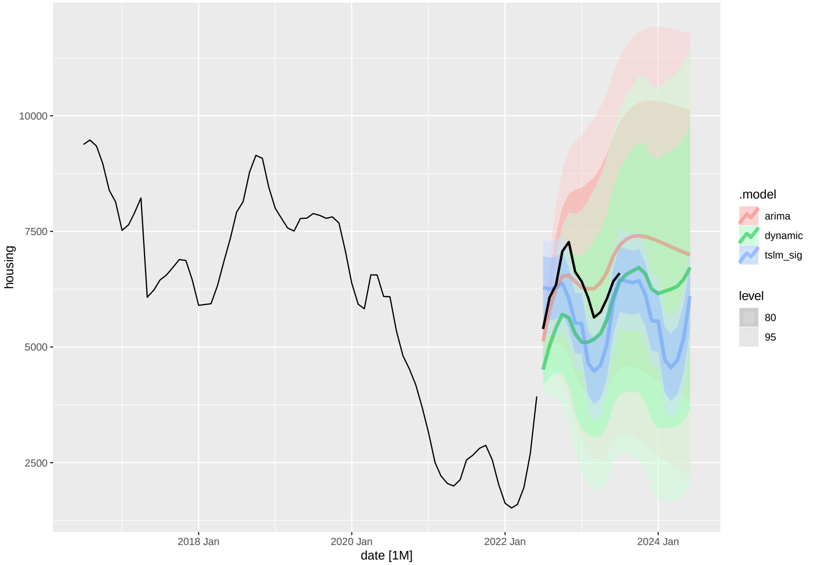
Dynamic Regression | Next Two Years

It’s been a year…

Dynamic Regression | Next Two Years

```r
# Load data
housing_validation <- read.csv("../data/housing_validation.csv")

# Convert date
housing_validation$date <- yearmonth(housing_validation$date)

# Convert to `tsibble`
housing_valid <- housing_validation %>%
  as_tsibble(index = date)
```

Dynamic Regression | Next Two Years



Preference?

Dynamic Regression | Next Two Years

```r
# Point estimates
fc %>%
  accuracy(housing_valid)
```

```
# A tibble: 3 x 10
  .model    .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>     <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 arima     Test  -73.6  467.  415. -1.46  6.60   NaN   NaN 0.777
2 dynamic   Test  955.  1047.  955. 15.0  15.0    NaN   NaN 0.738
3 tslm_sig  Test  563.   814.  703.  8.76 11.3    NaN   NaN 0.543
```

```r
# Distributional estimates
fc %>% accuracy(
    housing_valid,
    list(crps = CRPS)
  )
```

```
# A tibble: 3 x 3
  .model    .type  crps
  <chr>     <chr> <dbl>
1 arima     Test   406.
2 dynamic   Test   688.
3 tslm_sig  Test   514.
```