House Sales

Santi

2022-06-22

Let's load the data and run some basic analysis on it.

```
house_data <- read.csv('ma_lga_12345.csv')
head(house_data, 10)
```

```
##      saledate     MA  type bedrooms
## 1  30/09/2007 441854 house        2
## 2  31/12/2007 441854 house        2
## 3  31/03/2008 441854 house        2
## 4  30/06/2008 441854 house        2
## 5  30/09/2008 451583 house        2
## 6  31/12/2008 440256 house        2
## 7  31/03/2009 442566 house        2
## 8  30/06/2009 446113 house        2
## 9  30/09/2009 440123 house        2
## 10 31/12/2009 442131 house        2
```

```
# Check for possible duplicates and NA values.
sum(is.na(house_data))
```

```
## [1] 0
```

```
sum(duplicated(house_data))
```

```
## [1] 0
```

```
# Run summary on the data
summary(house_data)
```

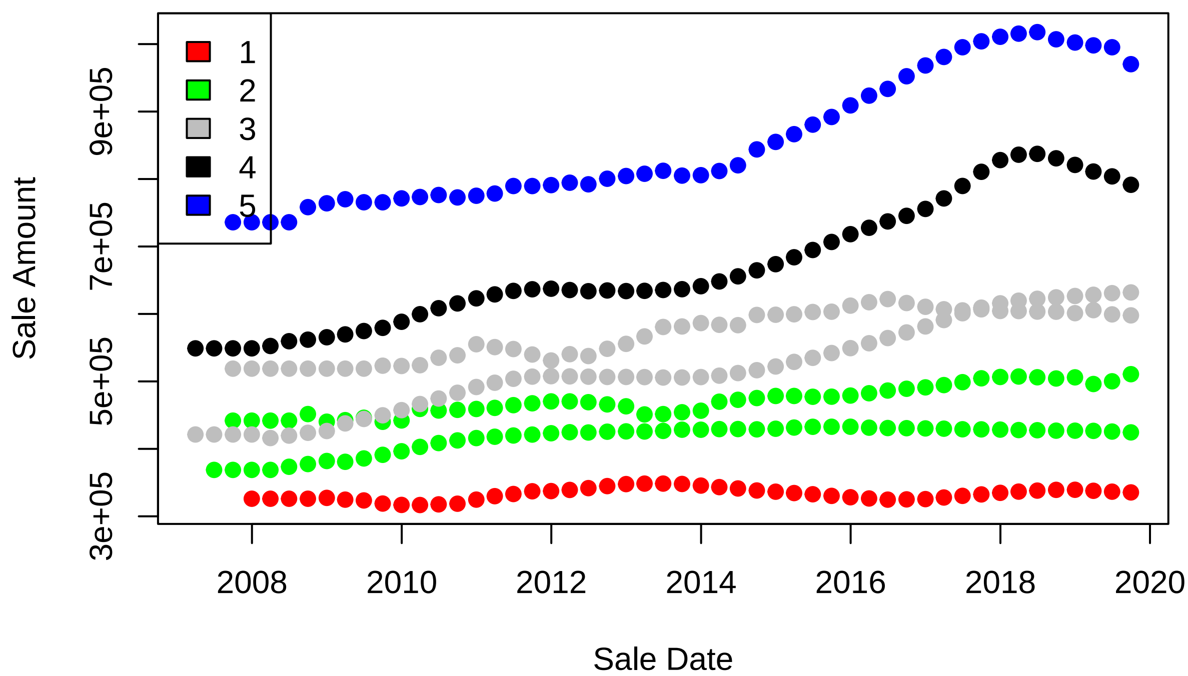
```
##    saledate               MA               type              bedrooms
##  Length:347         Min.   : 316751   Length:347         Min.   :1.000
##  Class :character   1st Qu.: 427740   Class :character   1st Qu.:2.000
##  Mode  :character   Median : 507744   Mode  :character   Median :3.000
##                     Mean   : 548132                      Mean   :2.867
##                     3rd Qu.: 627516                      3rd Qu.:4.000
##                     Max.   :1017752                      Max.   :5.000
```

We can see there doesn't seem to be anything unusual with the data when we look at the factors individually. Since we are doing time-series analysis, let's convert the *saledate* to a date format and the *bedrooms* into a factor.

```
house_data$saledate <- as.Date(house_data$saledate, '%d/%m/%Y')
house_data$bedrooms <- as.factor(house_data$bedrooms)
```

Now let's plot *saledate* with MA (the sale amount). Notices that the 2 and 3 bedroom housing units have two seemingly distinct lines. As we will soon see, this is attributed to the fact that they're the only housing units to have sales data for houses and units i.e codominium.

```
{plot(house_data$saledate, house_data$MA, xlab = "Sale Date", ylab = "Sale Amount",
     pch = 19, col = c('red','green','grey','black','blue')[house_data$bedrooms])
legend('topleft', c('1','2','3','4','5'), fill = c('red','green','grey','black','blue'))}
```



With that in mind let's filter the *bedroom* variable to run time-series on both types of housing units. We will do the time-series using fiscal quarters.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
one_br <- house_data %>% filter(bedrooms == '1')
ts_one <- ts(one_br[,2], start = c(2007,4), frequency = 4)

two_br_unit <- house_data %>% filter(bedrooms == '2', type == 'unit')
ts_two_unit <- ts(two_br_unit[,2], start = c(2007,2), frequency = 4)

two_br_house <- house_data %>% filter(bedrooms == '2', type == 'house')
ts_two_house <- ts(two_br_house[,2], start = c(2007,3), frequency = 4)
```

```r
three_br_unit <- house_data %>% filter(bedrooms == '3', type == 'unit')
ts_three_unit <- ts(three_br_unit[,2], start = c(2007,3), frequency = 4)

three_br_house <- house_data %>% filter(bedrooms == '3', type == 'house')
ts_three_house<- ts(three_br_house[,2], start = c(2007,1), frequency = 4)

four_br_house <- house_data %>% filter(bedrooms == '4', type == 'house')
ts_four_house <- ts(four_br_house[,2], start = c(2007,1), frequency = 4)

five_br_house <- house_data %>% filter(bedrooms == '5', type == 'house')
ts_five_house <- ts(five_br_house[,2], start = c(2007,3), frequency = 4)
```

Now that we have filtered and created our time-series object by unit type we can run our ARIMA model and create a price forecast. Since we will essentially run the same lines of code for each time-series object, let's make a function that:

- Runs ARIMA on the time-series object for each bedroom, unity type combination.
- Forecasts n points.
- Plots the forecast data.

```r
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
```

```
## The following object is masked from 'package:forecast':
##
##     accuracy
```

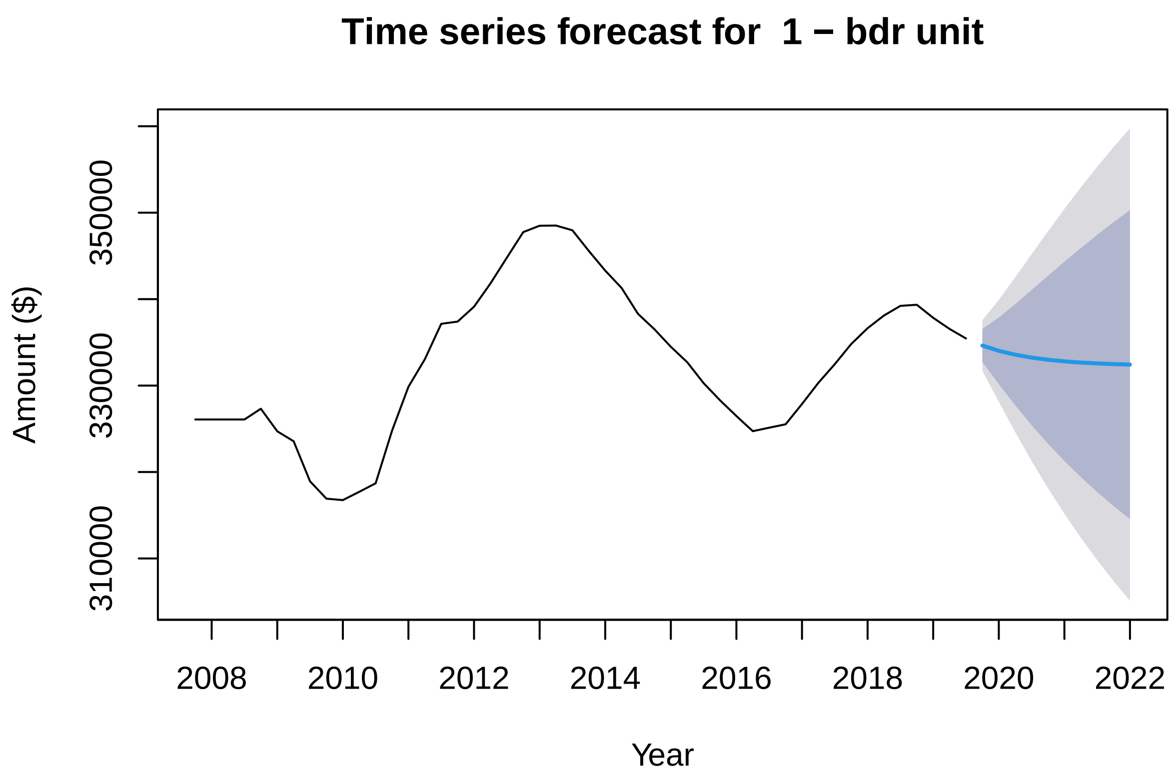
```r
housingForecast <- function(time_series, forecast_points, unit_type, brs)
{
  # Run ARIMA and create summary.
  arima_model <- auto.arima(time_series)
  summary(arima_model)

  # Forecast the number of points required
  data_forecast <- forecast(arima_model, forecast_points)
  print(data_forecast)

  # Plot the forecast data.
  plot(data_forecast, xlab = 'Year', ylab = 'Amount ($)',
       main = paste('Time series forecast for ',brs,'- bdr',unit_type),
       # This makes our x-axis consistent and more readble.
       xaxp = c(2007, (2020+forecast_points), ((2020+forecast_points) - 2007) %% 1000))
  return(summary(arima_model))
}
```

So now let's do some forcasting.

One-Bedroom Unit Forecast

```
housingForecast(ts_one, 10, 'unit', 1)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2019 Q4       334630.4 332713.2 336547.6 331698.2 337562.5
## 2020 Q1       334022.2 330169.8 337874.6 328130.5 339913.9
## 2020 Q2       333570.4 327722.7 339418.2 324627.0 342513.8
## 2020 Q3       333234.8 325419.1 341050.5 321281.7 345187.9
## 2020 Q4       332985.5 323271.0 342699.9 318128.5 347842.4
## 2021 Q1       332800.3 321274.2 344326.3 315172.7 350427.8
## 2021 Q2       332662.7 319417.7 345907.6 312406.2 352919.1
## 2021 Q3       332560.5 317688.0 347432.9 309815.0 355305.9
## 2021 Q4       332484.5 316071.4 348897.6 307382.9 357586.2
## 2022 Q1       332428.1 314555.2 350301.0 305093.9 359762.3
```



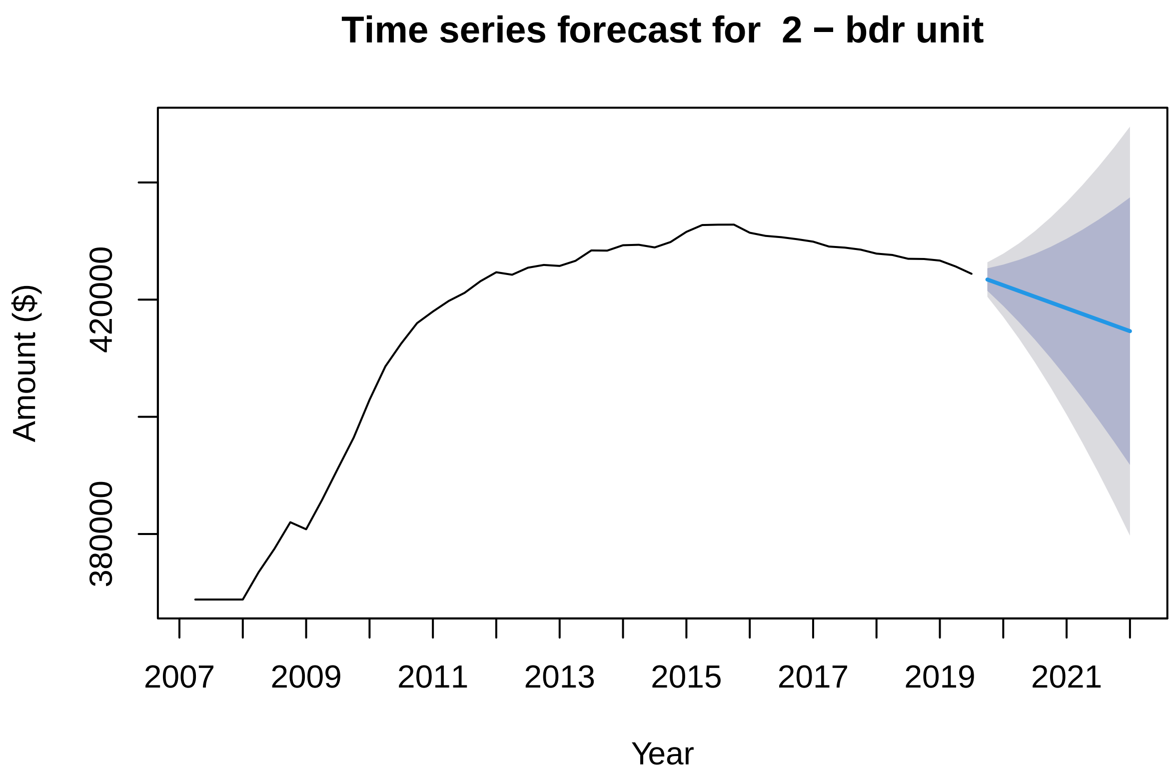

```
## Series: time_series
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.7429
## s.e.  0.0925
##
## sigma^2 = 2238012:  log likelihood = -410.16
## AIC=824.32   AICc=824.59   BIC=828.02
```

```
##
## Training set error measures:
##                    ME   RMSE      MAE        MPE      MAPE      MASE
## Training set 39.93949 1464.5 1005.183 0.01412482 0.3036909 0.1434905
##                     ACF1
## Training set -0.07697338
```

Two-Bedroom Unit Forecast

```
housingForecast(ts_two_unit, 10, 'unit', 2)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2019 Q4       423432.6 421513.3 425352.0 420497.2 426368.1
## 2020 Q1       422453.3 418930.1 425976.5 417065.0 427841.5
## 2020 Q2       421473.9 416151.4 426796.5 413333.8 429614.0
## 2020 Q3       420494.6 413175.0 427814.2 409300.2 431688.9
## 2020 Q4       419515.2 410012.7 429017.7 404982.4 434048.0
## 2021 Q1       418535.8 406676.9 430394.8 400399.2 436672.5
## 2021 Q2       417556.5 403178.4 431934.6 395567.1 439545.9
## 2021 Q3       416577.1 399526.4 433627.9 390500.3 442654.0
## 2021 Q4       415597.8 395728.9 435466.6 385211.0 445984.5
## 2022 Q1       414618.4 391792.9 437443.8 379709.9 449526.9
```



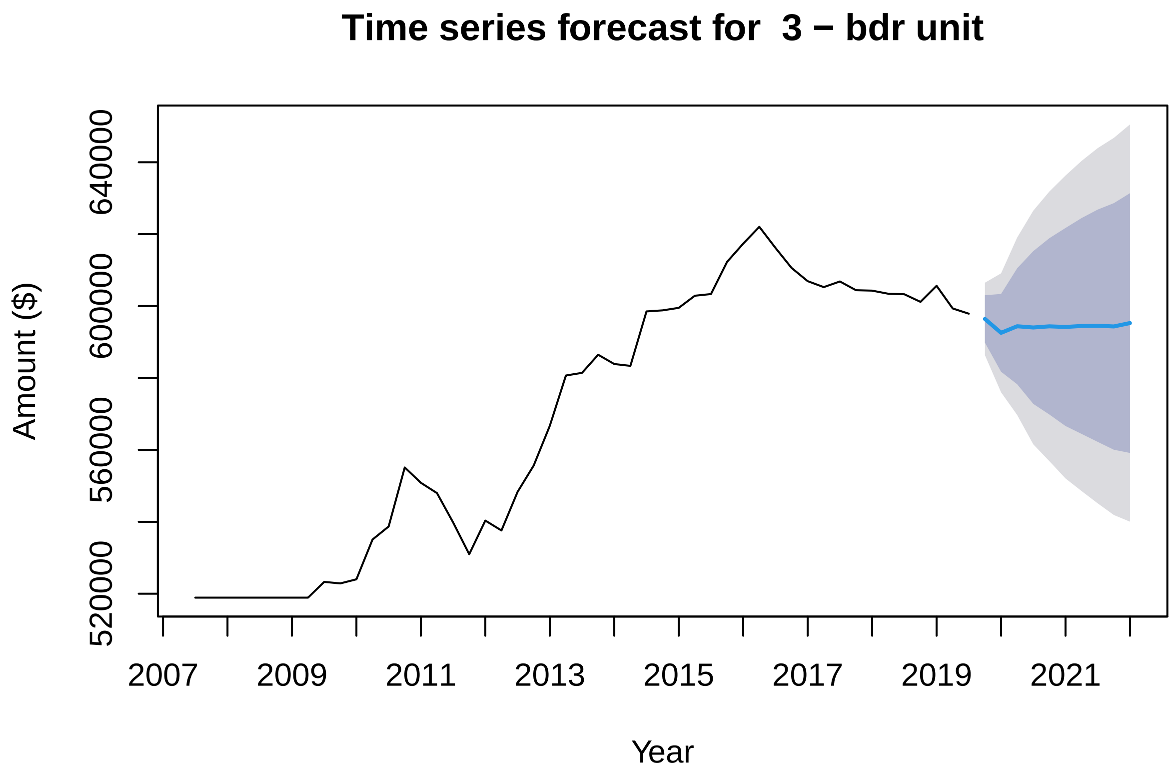

```
## Series: time_series
## ARIMA(0,2,1)
##
```

```
## Coefficients:
##           ma1
##       -0.4607
## s.e.   0.1415
##
## sigma^2 = 2243076:  log likelihood = -418.62
## AIC=841.24   AICc=841.51   BIC=844.99
##
## Training set error measures:
##                     ME     RMSE      MAE          MPE      MAPE      MASE
## Training set -43.95493 1452.065 963.0706 -0.006027837 0.2377975 0.1577489
##                    ACF1
## Training set 0.01884361
```

Three-Bedroom Unit Forecast

```
housingForecast(ts_three_unit, 10, 'unit', 3)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2019 Q4       596417.9 589828.9 603006.8 586340.9 606494.8
## 2020 Q1       592558.5 581742.2 603374.7 576016.5 609100.5
## 2020 Q2       594385.1 578255.5 610514.7 569717.0 619053.2
## 2020 Q3       594040.6 572794.0 615287.1 561546.7 626534.4
## 2020 Q4       594364.2 569846.2 618882.2 556867.1 631861.3
## 2021 Q1       594182.4 566642.1 621722.8 552063.1 636301.8
## 2021 Q2       594463.6 564437.5 624489.7 548542.7 640384.5
## 2021 Q3       594533.8 562229.4 626838.3 545128.5 643939.2
## 2021 Q4       594336.6 560046.8 628626.4 541894.9 646778.3
## 2022 Q1       595278.5 559158.1 631398.8 540037.1 650519.8
```



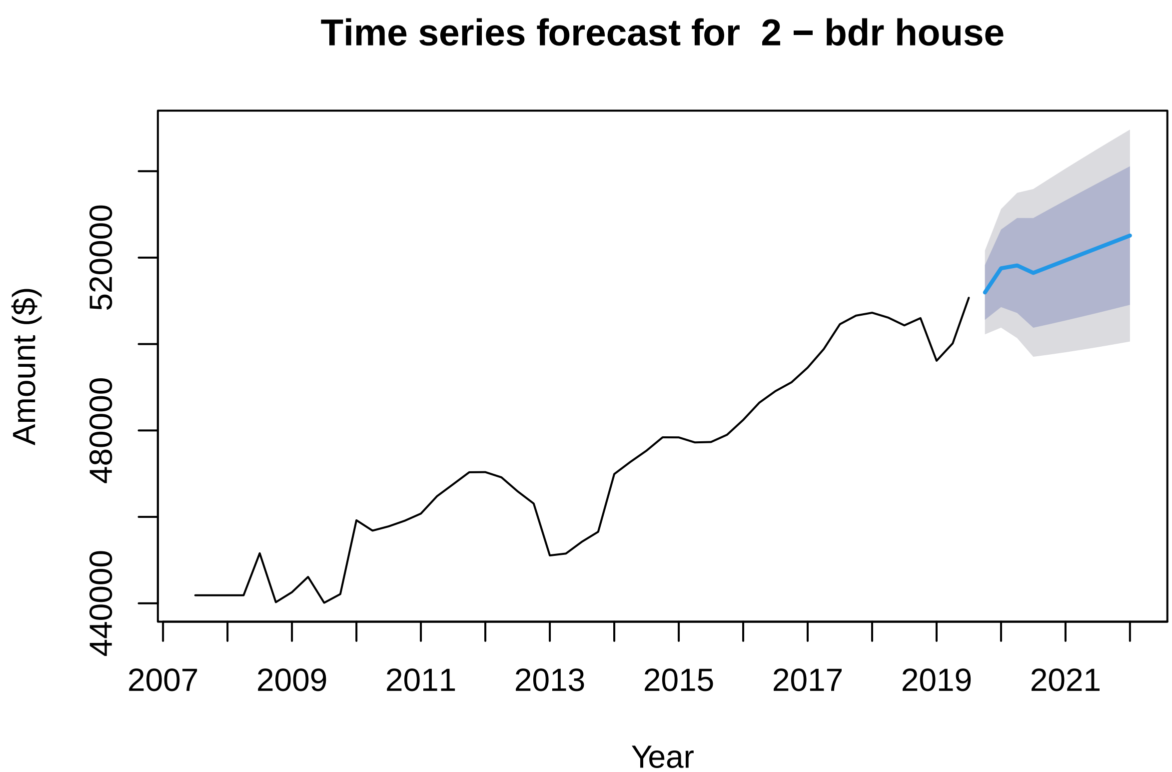

```
## Series: time_series
## ARIMA(2,1,0)(2,0,0)[4]
##
## Coefficients:
##          ar1     ar2     sar1     sar2
##       0.3018  0.4231  -0.5448  -0.3209
## s.e.  0.1326  0.1338   0.1428   0.1352
##
## sigma^2 = 26434054:  log likelihood = -477.03
## AIC=964.07   AICc=965.5   BIC=973.42
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE      MAPE      MASE
## Training set 793.4379 4872.033 3464.103 0.1435795 0.6040255 0.2840096
##                     ACF1
## Training set -0.03468343
```

Two-Bedroom House Forecast

```
housingForecast(ts_two_house, 10, 'house', 2)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2019 Q4       511943.7 505598.6 518288.9 502239.7 521647.8
## 2020 Q1       517522.0 508548.6 526495.3 503798.4 531245.5
## 2020 Q2       518183.2 507193.2 529173.3 501375.4 534991.1
```

```
## 2020 Q3       516466.9 503776.6 529157.1 497058.8 535874.9
## 2020 Q4       517907.3 504601.0 531213.6 497557.0 538257.6
## 2021 Q1       519347.7 505452.6 533242.9 498097.0 540598.5
## 2021 Q2       520788.2 506328.2 535248.1 498673.6 542902.8
## 2021 Q3       522228.6 507225.1 537232.2 499282.7 545174.6
## 2021 Q4       523669.0 508140.9 539197.2 499920.8 547417.3
## 2022 Q1       525109.5 509073.9 541145.0 500585.2 549633.8
```

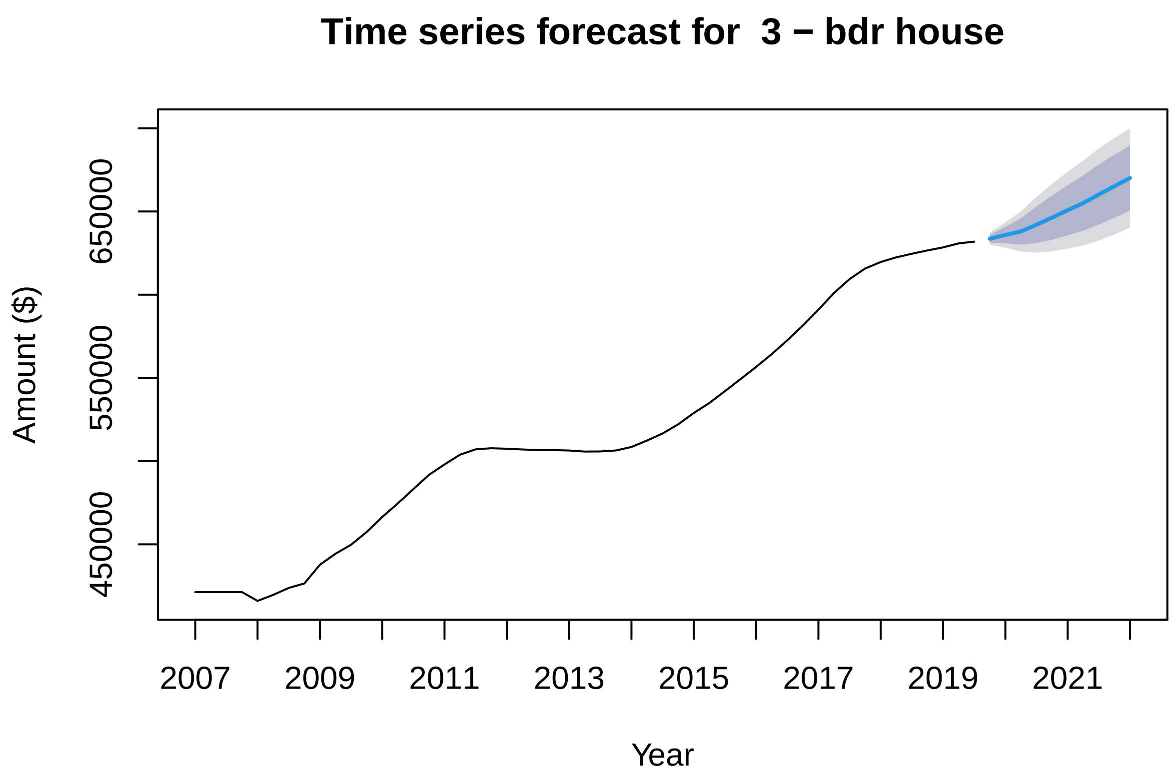



```
## Series: time_series
## ARIMA(0,1,0)(0,0,1)[4] with drift
##
## Coefficients:
##          sma1      drift
##       -0.3693  1440.4390
## s.e.   0.1853   464.6175
##
## sigma^2 = 24513602:  log likelihood = -475.73
## AIC=957.46   AICc=958.01   BIC=963.07
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE      MAPE      MASE
## Training set -5.730028 4797.163 3141.751 -0.01310796 0.6725938 0.3246236
##                     ACF1
## Training set -0.06523639
```

Three-Bedroom House Forecast

```
housingForecast(ts_three_house, 10, 'house', 3)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2019 Q4       633627.8 631258.1 635997.5 630003.7 637251.9
## 2020 Q1       635888.2 630895.1 640881.3 628251.9 643524.5
## 2020 Q2       638021.4 630119.5 645923.2 625936.5 650106.2
## 2020 Q3       642055.7 631089.5 653021.9 625284.3 658827.1
## 2020 Q4       646318.1 633085.2 659551.0 626080.2 666556.1
## 2021 Q1       650780.3 635698.3 665862.4 627714.4 673846.3
## 2021 Q2       655047.8 638383.0 671712.6 629561.2 680534.4
## 2021 Q3       660200.6 642139.7 678261.4 632578.9 687822.2
## 2021 Q4       665203.6 646246.2 684161.0 636210.7 694196.5
## 2022 Q1       670106.0 650566.7 689645.3 640223.2 699988.8
```



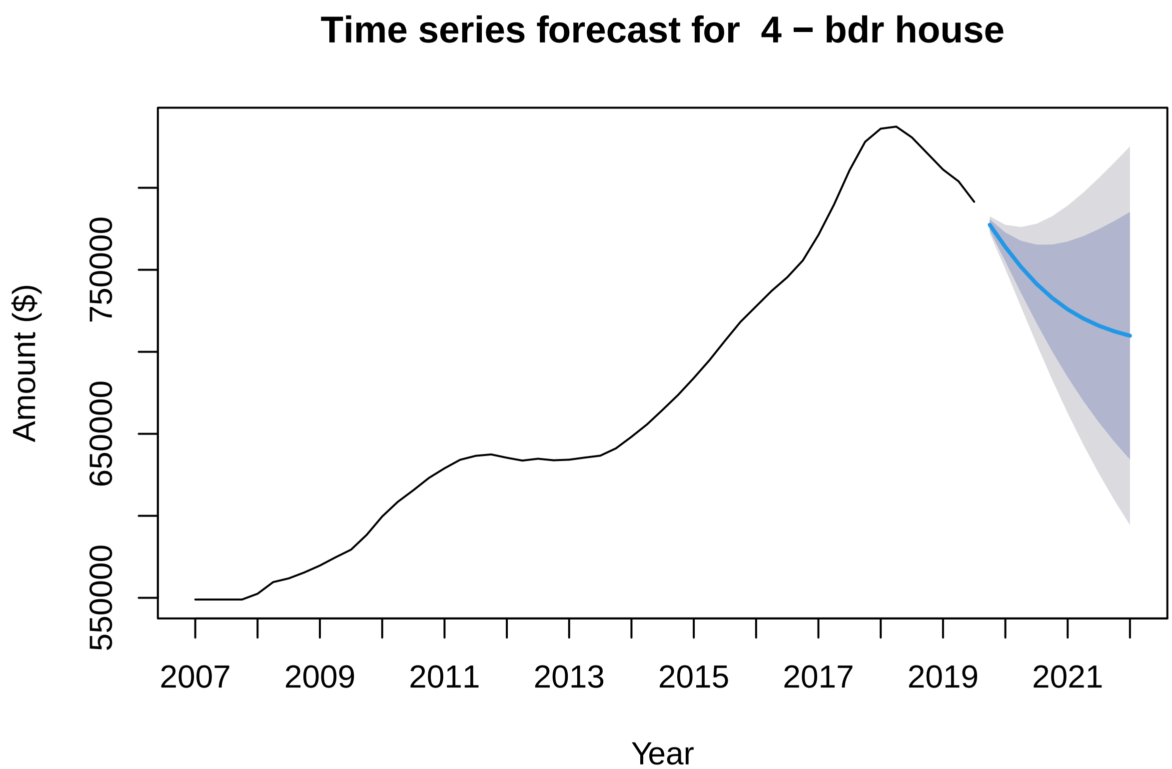

```
## Series: time_series
## ARIMA(1,1,0)(2,0,1)[4] with drift
##
## Coefficients:
##          ar1    sar1     sar2     sma1      drift
##       0.8558  0.2403  -0.3121  -0.8628  4285.7969
## s.e.  0.0891  0.2145   0.2077   0.2314   370.4374
##
## sigma^2 = 3398320:  log likelihood = -447.6
## AIC=907.19   AICc=909.14   BIC=918.66
```

```
##
## Training set error measures:
##                    ME     RMSE      MAE         MPE      MAPE       MASE
## Training set -58.09083 1731.623 1236.409 -0.01711588 0.2560346 0.06786317
##                     ACF1
## Training set -0.09240477
```

Four-Bedroom House Forecast

```
housingForecast(ts_four_house, 10, 'house', 4)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2019 Q4       777455.1 774064.3 780846.0 772269.2 782641.1
## 2020 Q1       763938.9 755107.6 772770.1 750432.7 777445.1
## 2020 Q2       751832.9 735969.4 767696.5 727571.7 776094.1
## 2020 Q3       741474.7 717562.5 765386.8 704904.2 778045.2
## 2020 Q4       732878.0 700355.1 765400.8 683138.6 782617.3
## 2021 Q1       725896.1 684532.8 767259.5 662636.4 789155.9
## 2021 Q2       720316.5 670112.9 770520.1 643536.7 797096.3
## 2021 Q3       715912.4 657020.0 774804.7 625844.3 805980.4
## 2021 Q4       712469.8 645133.6 779806.1 609487.9 815451.8
## 2022 Q1       709800.0 634317.0 785283.1 594358.6 825241.4
```



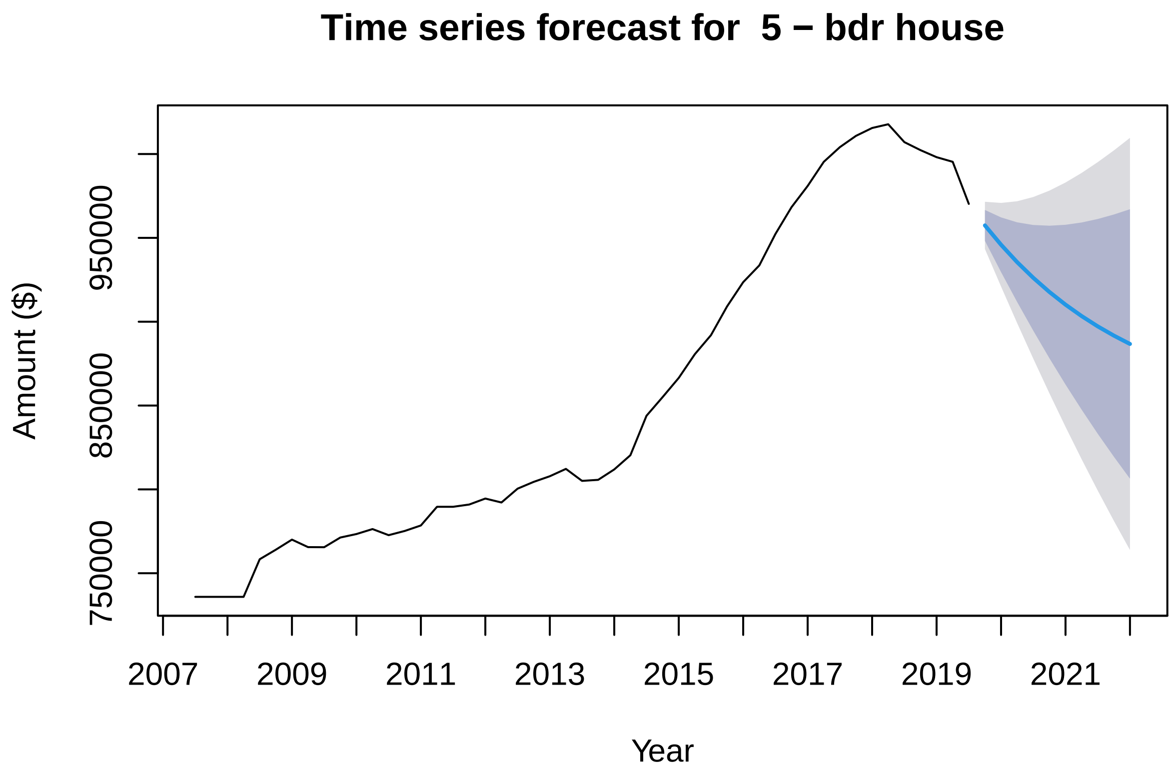

```
## Series: time_series
## ARIMA(2,1,0)
##
```

```
## Coefficients:
##          ar1      ar2
##       1.4048  -0.4918
## s.e.  0.1244   0.1247
##
## sigma^2 = 7000906:  log likelihood = -465.31
## AIC=936.62   AICc=937.14   BIC=942.35
##
## Training set error measures:
##                   ME     RMSE     MAE        MPE      MAPE       MASE
## Training set 270.726 2566.922 1880.71 0.05037792 0.2779408 0.07043778
##                      ACF1
## Training set -0.006186785
```

Five-Bedroom House Forecast

```r
housingForecast(ts_five_house, 10, 'house', 5)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
## 2019 Q4       957442.4 948235.3 966649.5 943361.3  971523.5
## 2020 Q1       945899.5 929595.6 962203.4 920964.8  970834.2
## 2020 Q2       935510.9 911754.4 959267.4 899178.5  971843.4
## 2020 Q3       926161.3 894638.7 957684.0 877951.6  974371.1
## 2020 Q4       917746.7 878238.4 957255.1 857324.0  978169.5
## 2021 Q1       910173.7 862543.1 957804.2 837329.1  983018.3
## 2021 Q2       903358.0 847535.2 959180.8 817984.4  988731.6
## 2021 Q3       897223.9 833190.7 961257.1 799293.6  995154.2
## 2021 Q4       891703.3 819482.0 963924.5 781250.4 1002156.1
## 2022 Q1       886734.8 806378.9 967090.6 763841.0 1009628.5
```




```
## Series: time_series
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.9000  -0.4386
## s.e.  0.0788   0.1666
##
## sigma^2 = 51615008:  log likelihood = -493.74
## AIC=993.48   AICc=994.03   BIC=999.1
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE      MAPE      MASE
## Training set 398.9056 6960.956 4467.352 0.06313621 0.5264191 0.1685397
##                     ACF1
## Training set -0.01247393
```