Forecasting Median house price in US cities using various Timeseries Forecasting models.

Team 18: Vishwas Shivakumar

Project Summary

In this project we will predict the median house price for a given city for the next 5 years based on the data from 2008-2021. This is done by applying Various methods and forecasting models in Timeseries.

Project Description

Objective: Buying a house is an expensive investment. But analyzing and predicting the House prices in the city of interest can be helpful for potential homeowners maximize the returns on their investment.

Usefulness: While buying a house we need to consider a lot of things one of the most important factor to consider is the cost. By having a forecast, the user can visualize where the best investment would be and the possible return in a few years on the given property. There are many websites which show this data, but Zillow stands out of the lot. For a normal user Zillow would work but for someone with more technical Knowledge or who wants to see the prediction with different models our project would be helpful.

Dataset: The dataset was collected from Zillow https://www.zillow.com/research/data/ where the data is updated monthly, the data consists of the median sale price of house in all US cities. Monthly data is available from 2008 April to 2021 December (165 records per city), For 95 US cities. For NY we have a gap of data between 2008 and 2012. We may have to remove the city from our list or predict the data with the missing data (yet to be decided).

The data doesn't require much cleaning, but it needs to be transposed as the month year is provided as a column in this dataset. Any other preprocessing required will be done as needed.

Description and Functionality: As a basic functionality the user can select the city(s) of their interest and we give the prediction for 5 years. Or the user can input their price range and we tell which cities are within their budget for investing.
For Advanced users they can view graphs with KPI's such as top 5 cities to buy a house in based on the Return on investment, editable graphs (Adding cities, selecting timeframe). Selecting their own model for prediction. Few more options can be added on later.

Task Division: As I am the only member in the team, I will be working on all the Tasks. The webapp will be developed on Shiny and Progress will be tracked on Kanban Board.

Project Part 2

Vishwas Shivakumar

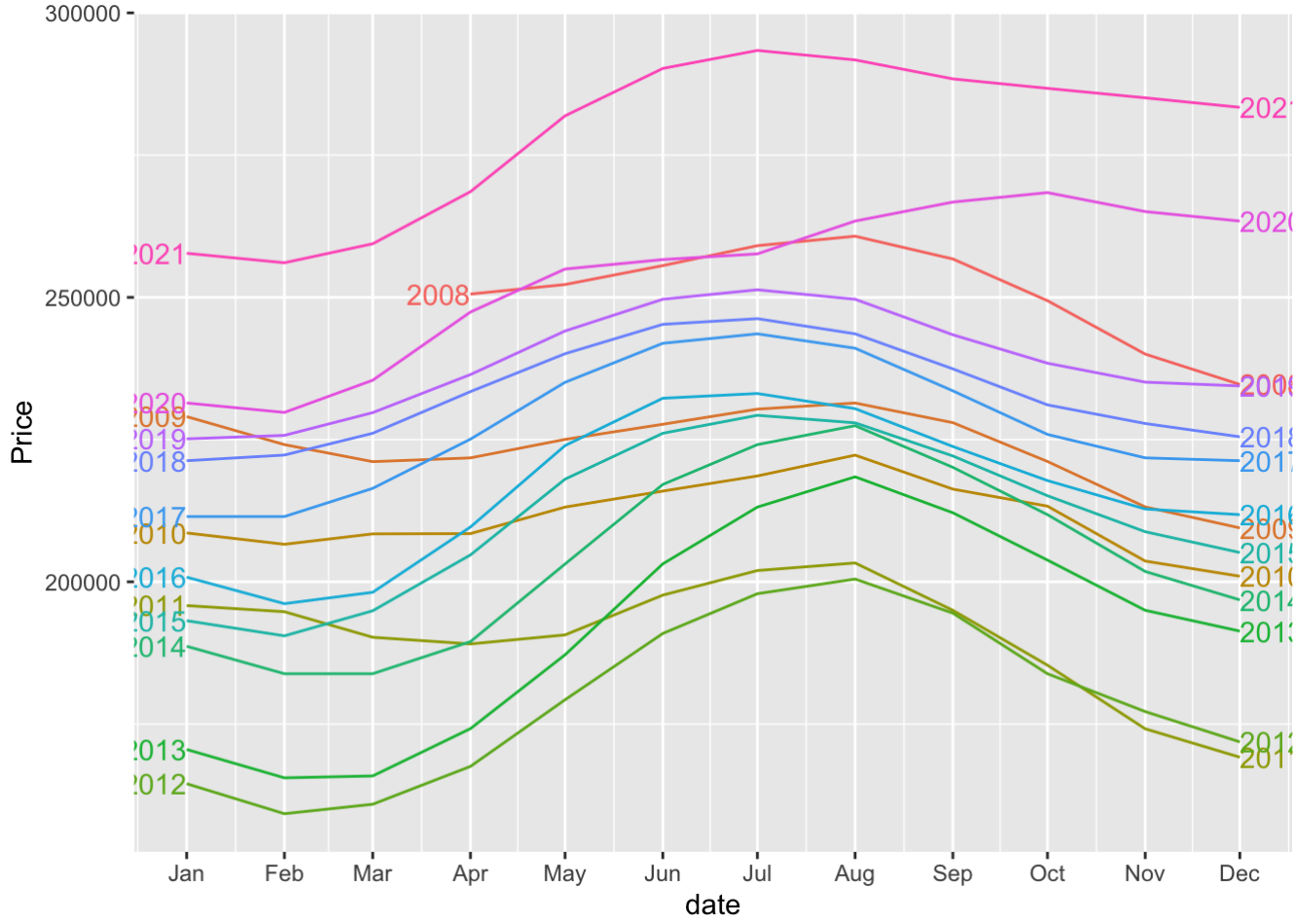
04/12/2022

Importing the dataset

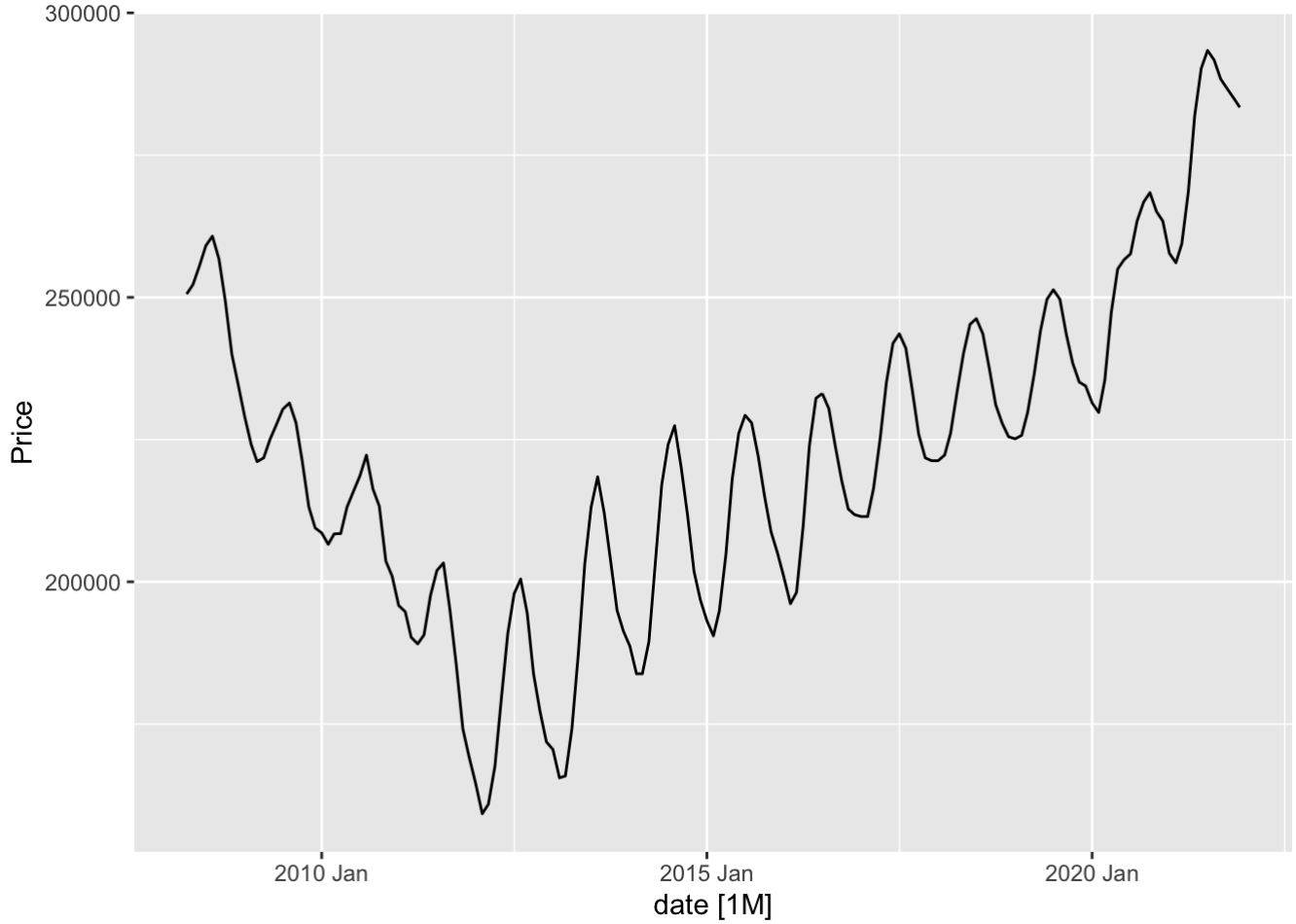
Filtering the dataset for one city and exploring the data here we have selected Chicago

```
## # A tsibble: 165 x 2 [1M]
##        date   Price
##       <mth>   <dbl>
##  1 2008 Apr 250586.
##  2 2008 May 252253.
##  3 2008 Jun 255582.
##  4 2008 Jul 259080.
##  5 2008 Aug 260747.
##  6 2008 Sep 256749.
##  7 2008 Oct 249421.
##  8 2008 Nov 240029.
##  9 2008 Dec 234702.
## 10 2009 Jan 229043.
## # … with 155 more rows
```

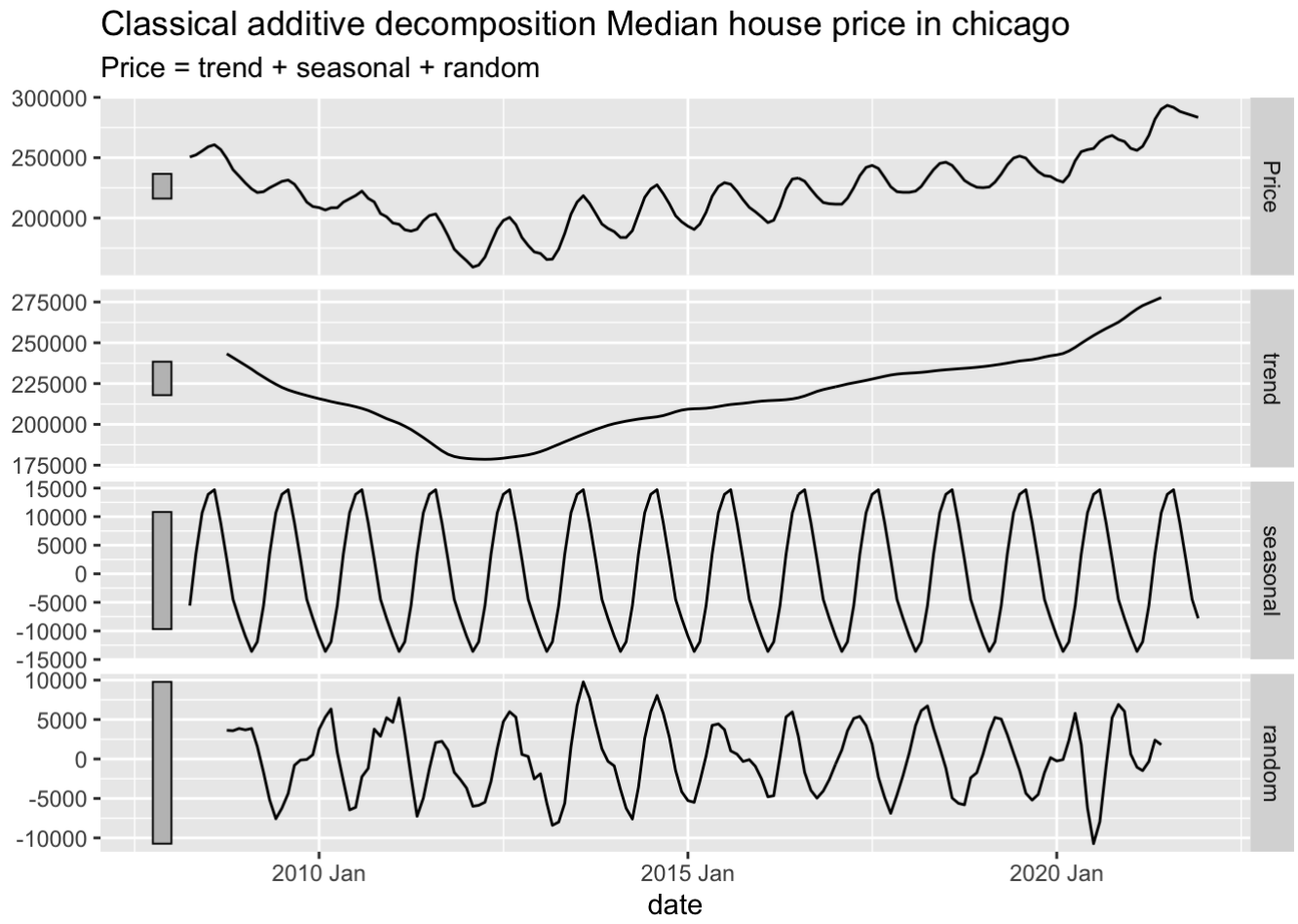
Seasonal plot for chicago and an autoplot to see the time series data for the same.

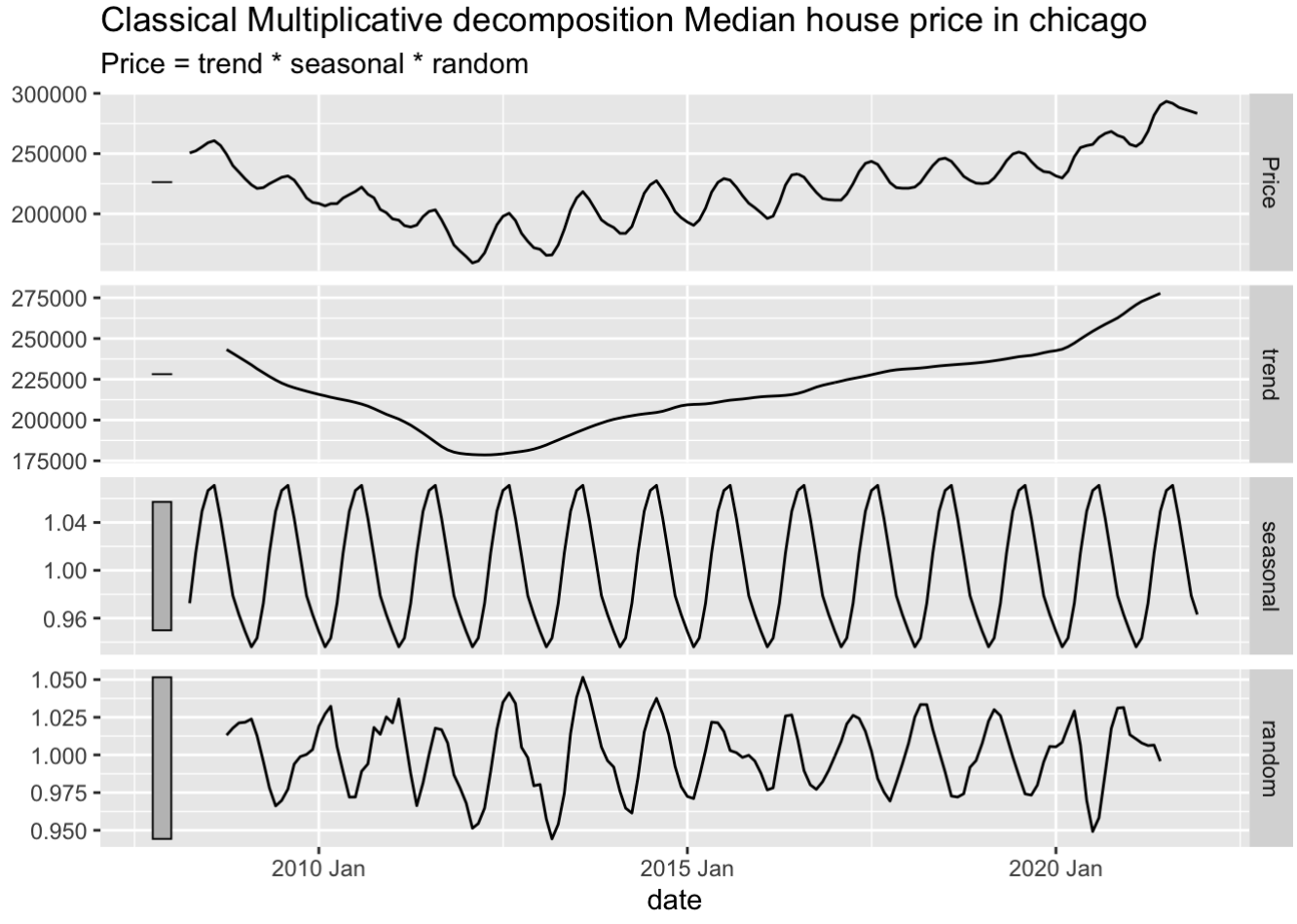



```
## Plot variable not specified, automatically selected `.vars = Price`
```



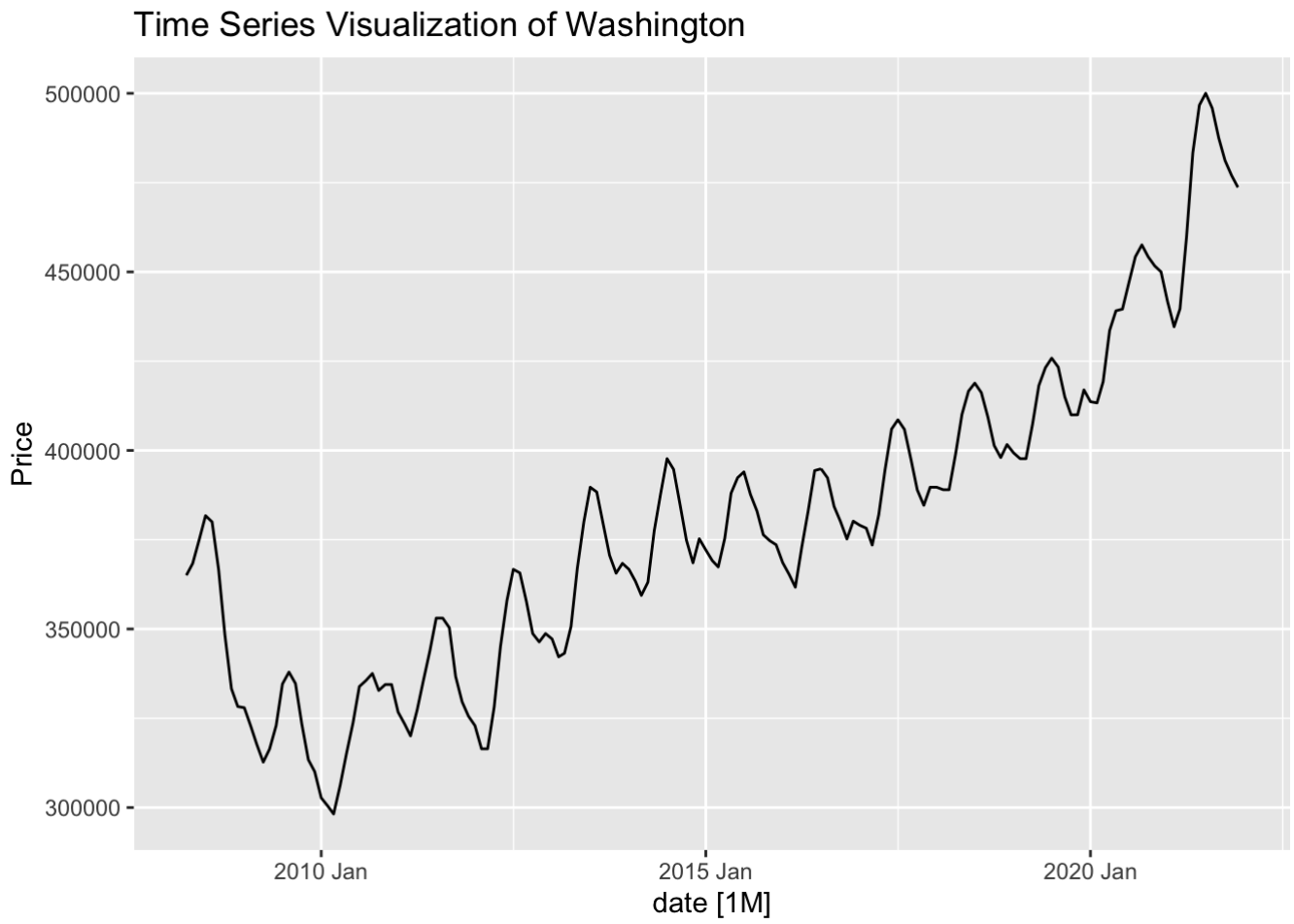
Classical additive and Multiplicative decomposition of the timeseries.


Classical additive decomposition Median house price in chicago
Price = trend + seasonal + random
Price
trend
seasonal
random
2010 Jan
2015 Jan
2020 Jan
date

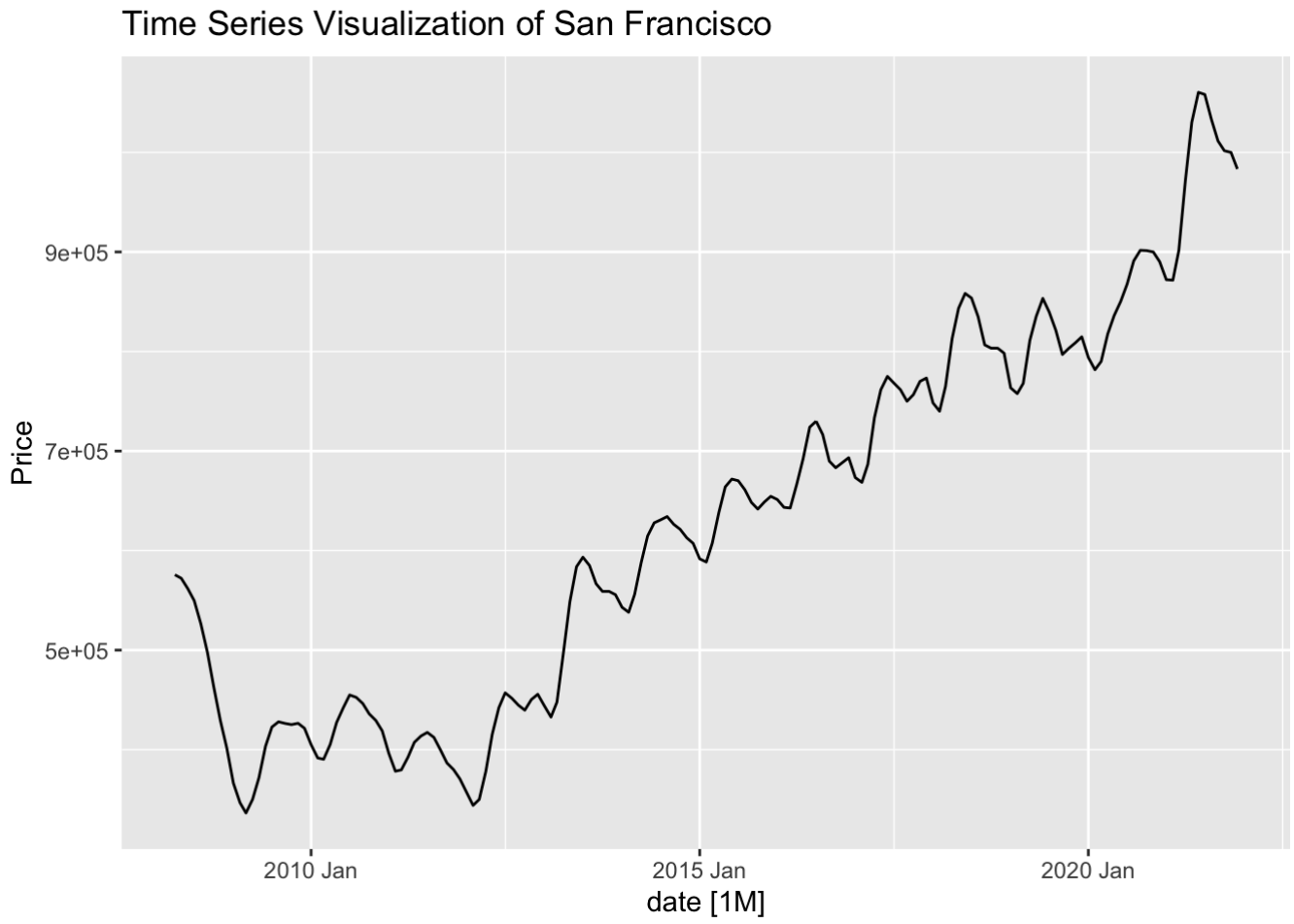

Classical Multiplicative decomposition Median house price in chicago
Price = trend * seasonal * random
Price
trend
seasonal
random
2010 Jan
2015 Jan
2020 Jan
date

Time Series Visulization for different cities

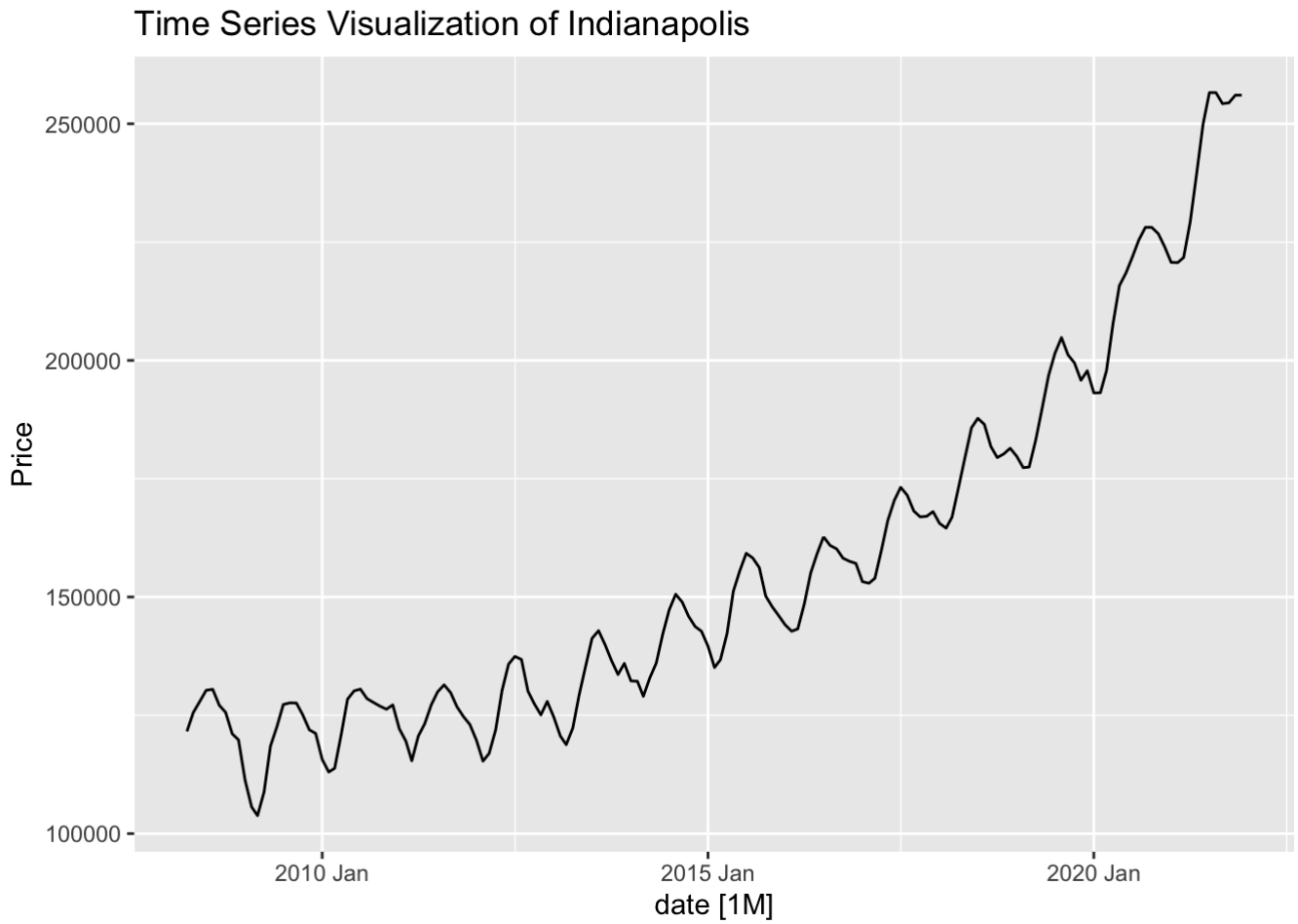
```
## Plot variable not specified, automatically selected `.vars = Price`
```



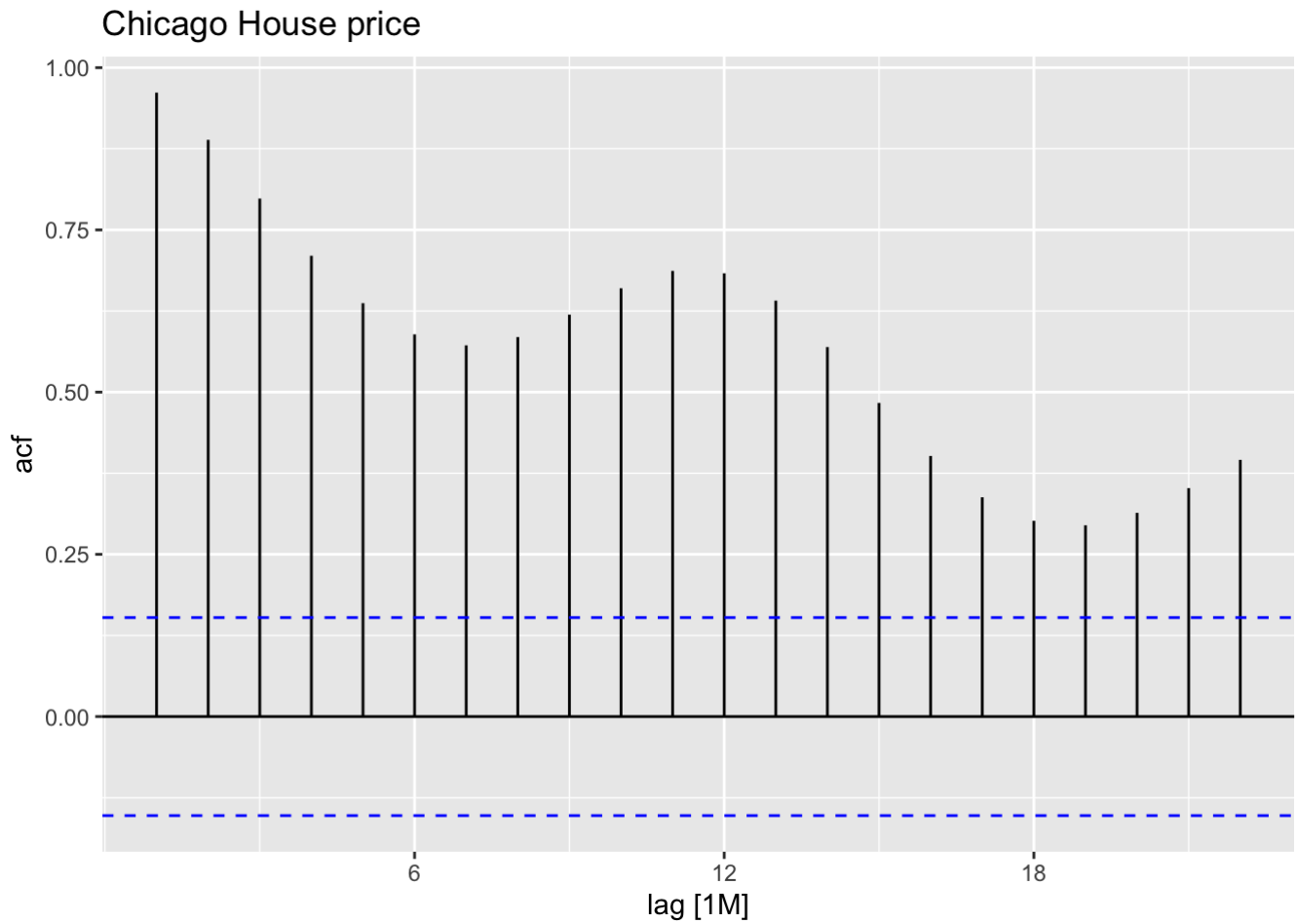

```
## Plot variable not specified, automatically selected `.vars = Price`
```




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

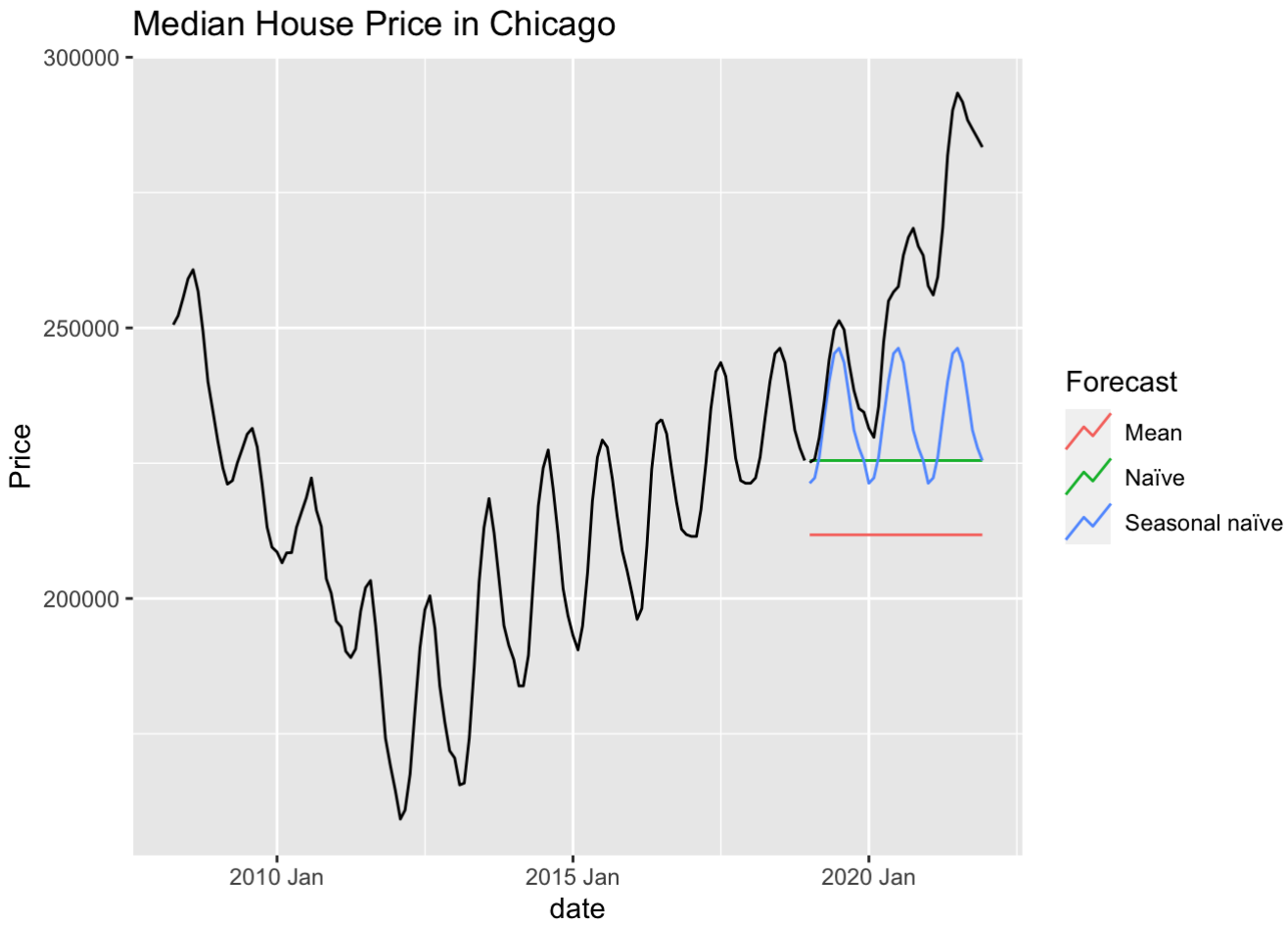


The Time series has a general upward trend but we can also see that in some housing markets there is a slight downward trend from 2008 to 2010, this can be attributed to the 2008 housing market crisis. We can also see a seasonal trend in the price, the price is at its highest mid year during the months of April to July. We can also see that the data is not stationary, we can see this in the Autocorrelation plot below since the time series has both trend and seasonality.



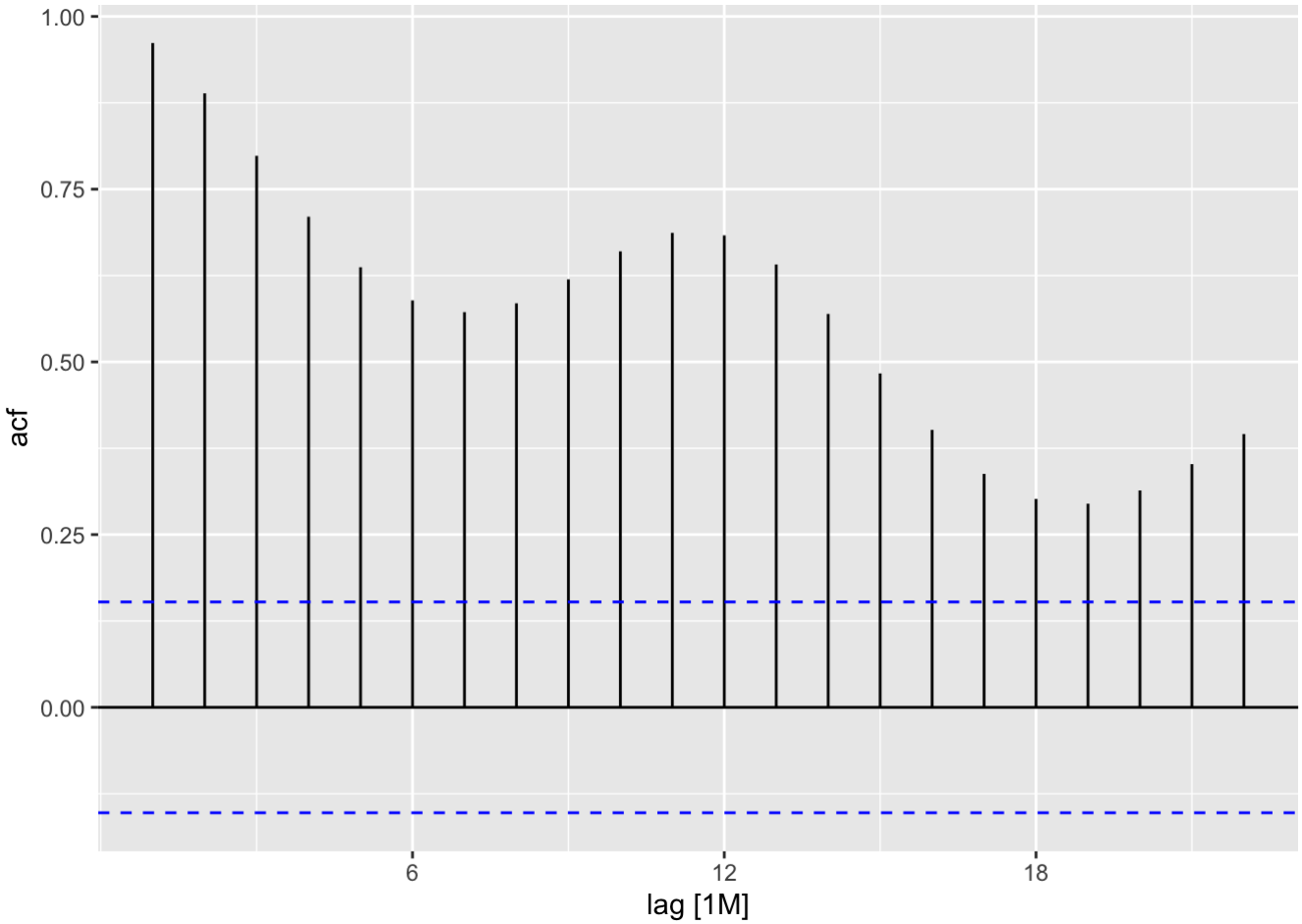
Trying some simple forecasting methods Mean , Naive and Seasonal Naive

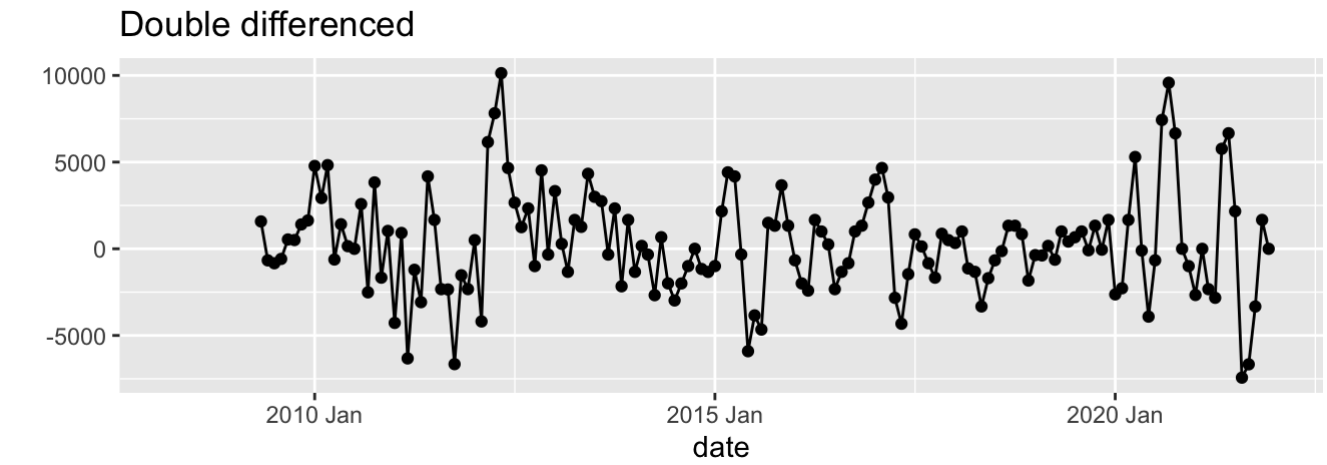
```
## Plot variable not specified, automatically selected `.vars = Price`
```

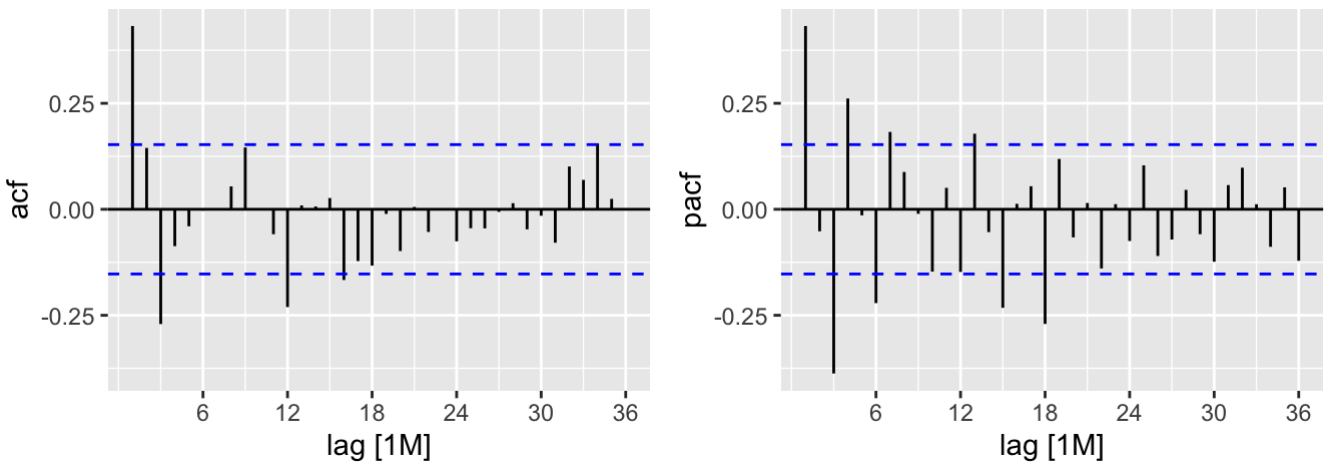


We can clearly see from the above Plot that the predictions are not accurate.And it should also be noted that the time series is not stationary.

We now use differencing to make the time series stationary we tried both first order and second order differencing here.


1.00
0.75
0.50
0.25
0.00
acf
6
12
18
lag [1M]


Double differenced
10000
5000
0
-5000
2010 Jan
2015 Jan
2020 Jan
date

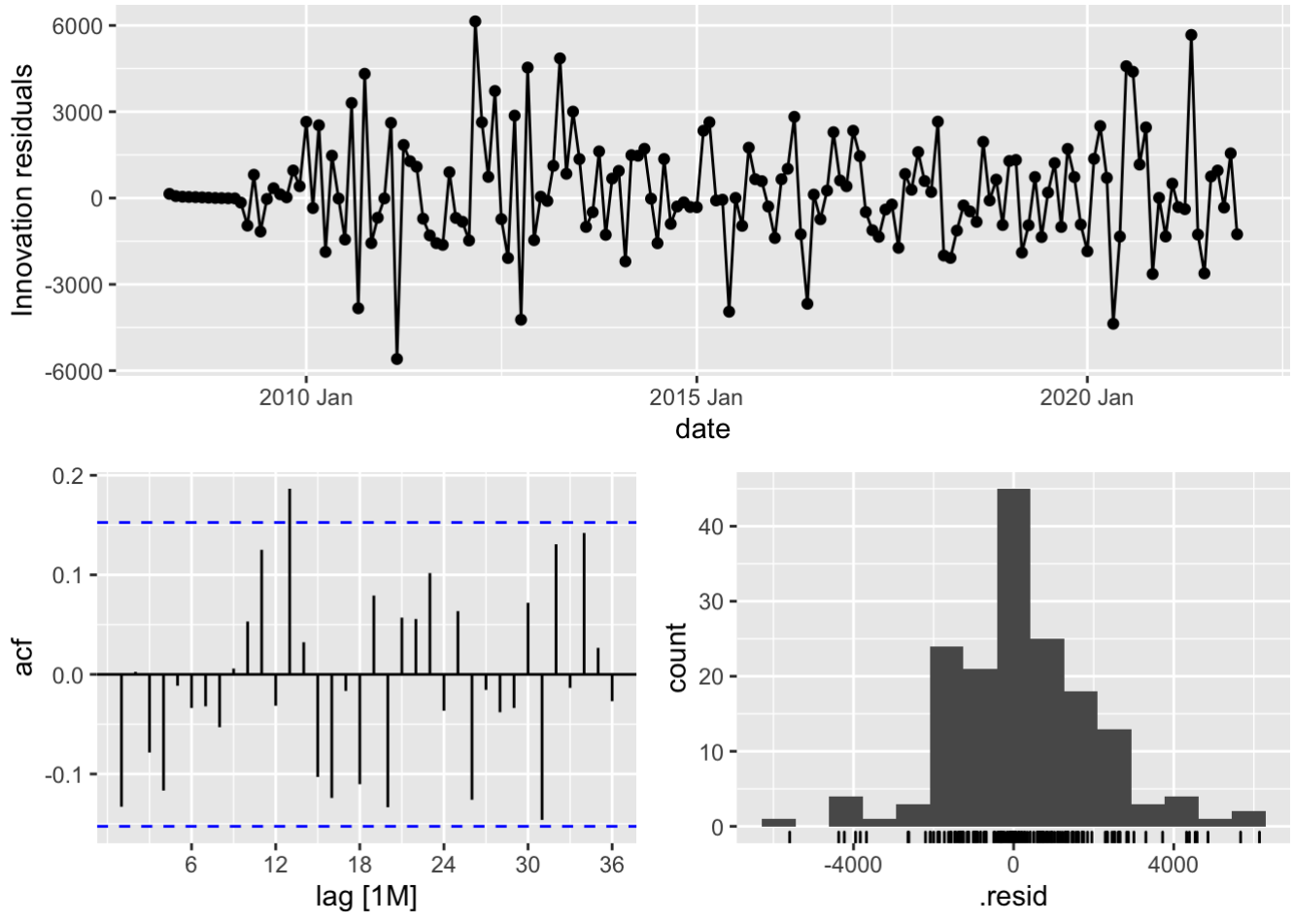

0.25
0.00
-0.25
acf
6
12
18
24
30
36
lag [1M]
0.25
0.00
-0.25
pacf
6
12
18
24
30
36
lag [1M]

Our aim now is to find an appropriate ARIMA model based on the ACF and PACF plots above. The significant spike at lag 1 in the ACF suggests a non-seasonal MA(2) component. The significant spike at lag 3 in the ACF suggests a seasonal MA(1) component. Consequently, we begin with an ARIMA(0,1,2)(0,1,1) model, indicating a first difference, a seasonal difference, and non-seasonal MA(2) and seasonal MA(1) component. If we had started with the PACF, we may have selected an ARIMA(2,1,0)(0,1,1) model — using the PACF to select the non-seasonal part of the model and the ACF to select the seasonal part of the model. We will also include an automatically selected model.

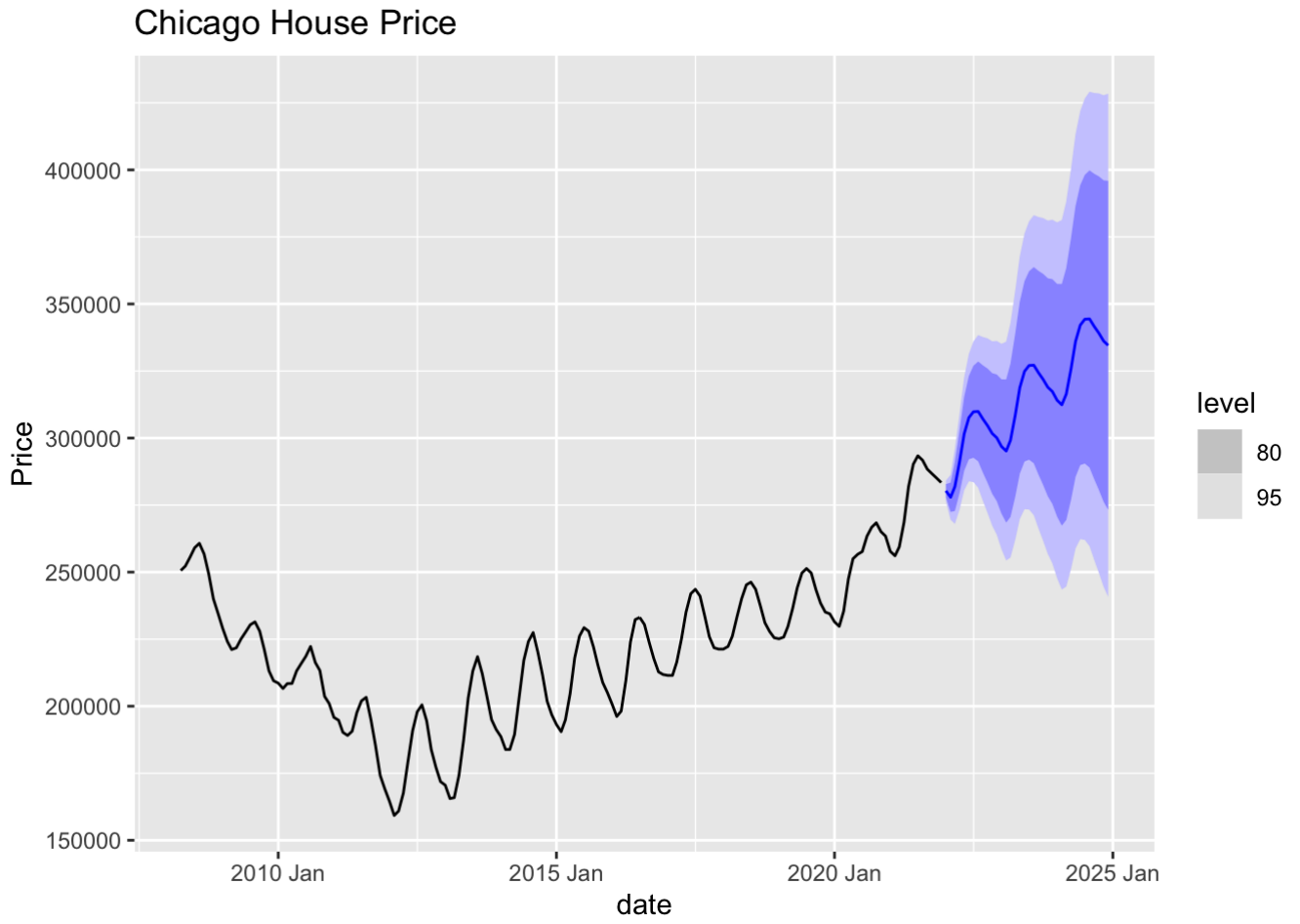
```
## # A mable: 3 x 2
## # Key:     Model name [3]
##   `Model name`                    Orders
##   <chr>                          <model>
## 1 arima012011  <ARIMA(0,1,2)(0,1,1)[12]>
## 2 arima210011  <ARIMA(2,1,0)(0,1,1)[12]>
## 3 auto         <ARIMA(0,1,2)(0,1,1)[12]>
```

```
## # A tibble: 3 × 6
##   .model        sigma2 log_lik   AIC  AICc   BIC
##   <chr>          <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1 arima012011 3768642.  -1371. 2751. 2751. 2763.
## 2 auto        3768642.  -1371. 2751. 2751. 2763.
## 3 arima210011 6028566.  -1403. 2814. 2814. 2826.
```

The residuals for the best model is shown below.



We now forecast using our Seasonal Arima model and the predictions are as excpected for the next 3 years(till 2024 dec)

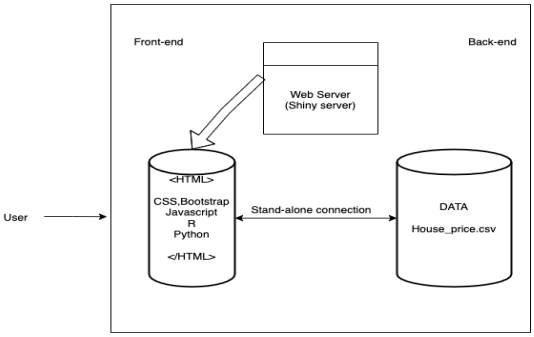


Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Web App Design

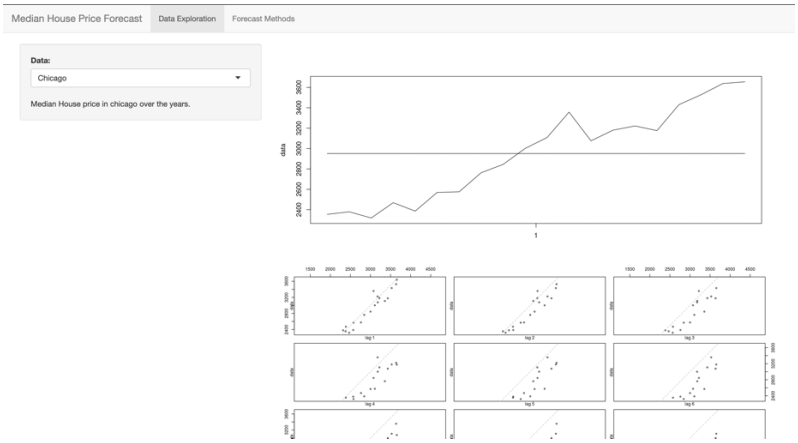
Web App Architecture

1. Our data will be stored in a website and we will access it through that url in our Webapp.
2. Backend will be built mostly using R and some Javascript.
3. The data will be accessed through a secure URL link and the users cannot change this data.
4. For the front end we will mostly use HTML,CSS and Bootstrap and some Javascript.
5. Our Application will be deployed in Shiny Server.
6. The interactivity to our app is provided to the users by allowing them to choose a city from the dropdowns, switch to different tabs, clicking and selecting the model they want to apply etc.,
7. Web app architecture.

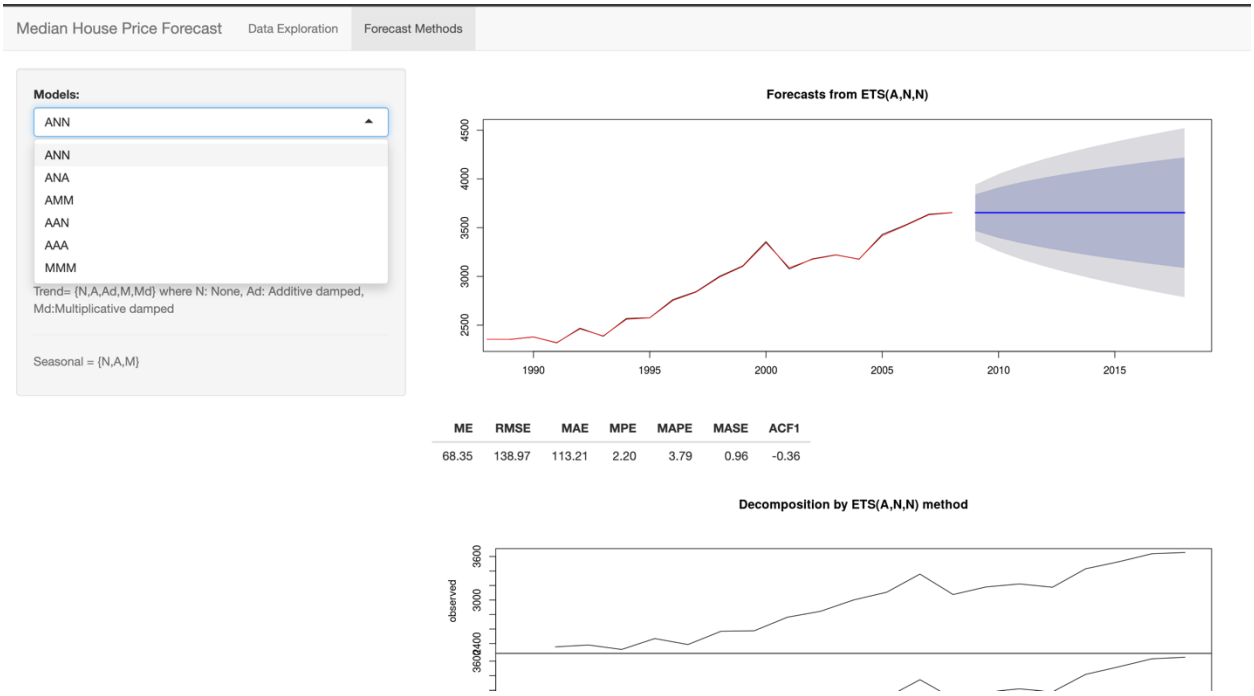


Web App Layout

Below is he initial screen the user sees, we will select a default city and display some basic time series data exploration in this tab.



We will be using 2 tabs one for data exploration and one for choosing different forecasting model. Menu will be on the left of the screen with dropdown options to select the city in the data exploration tab and to select the forecasting model in the forecast model tab.



Note:- The above screenshots are temporary and just represents the structure that will be followed, the final result may vary.

Team Work

- Create and test R scripts for different cities and choose different Forecasting models for the Users to choose from and also Suggest the best model to use.
- Start Build the Webapp on Shiny server.
- Test the webapp extensively
- Deploy the app.

Technical Report

Web URL

https://vishwas.shinyapps.io/MedianHousePrice/

Github URL

https://github.com/Vishwas336/TimeSeriesAnalysis_Spring2022

Technical description

Tools used

The webapp was written in Rstudio using Shiny and deployed on Shinyapps.io. We used the 2 file format for our Shiny webapp, ui.R which accepts the input from the user and displays the output. Server.R contains our R script for importing the data and other functions. The code was written in R.

Data

The dataset was collected from Zillow https://www.zillow.com/research/data/ where the data is updated monthly, the data consists of the median sale price of house in all US cities. Monthly data is available from 2008 April to 2021 December (165 records per city), For 95 US cities.

User functionalities

The user can select the city for which they want to see the data for, we provide the plot of the timeseries, decomposition of the time series, acf plot and double differenced acf, pacf plots so they can decide the model parameters for the Seasonal ARIMA model. The users can also choose the number of months for prediction to get an idea of what the price might be. We also provide some simple forecasting methods along with the auto seasonal ARIMA forecasting for the users.

Teamwork

This project was done by me as I am the only one in the team.