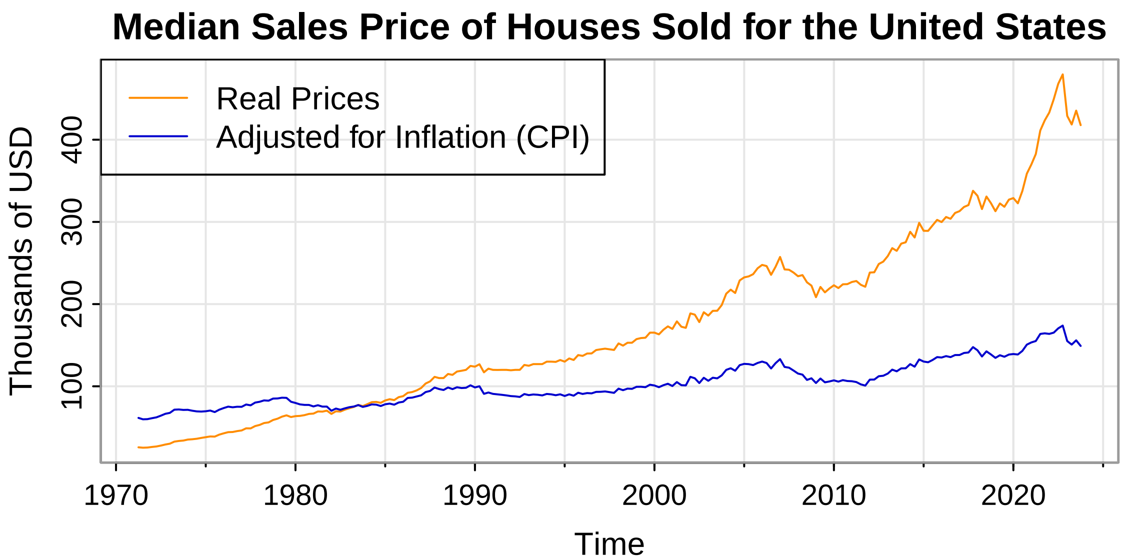
STAT 429 Project

Kunal Bhardwaj, NetID: kunal5, Section: GR

Understanding Housing Market Trends and Risks :

An Analytical Study



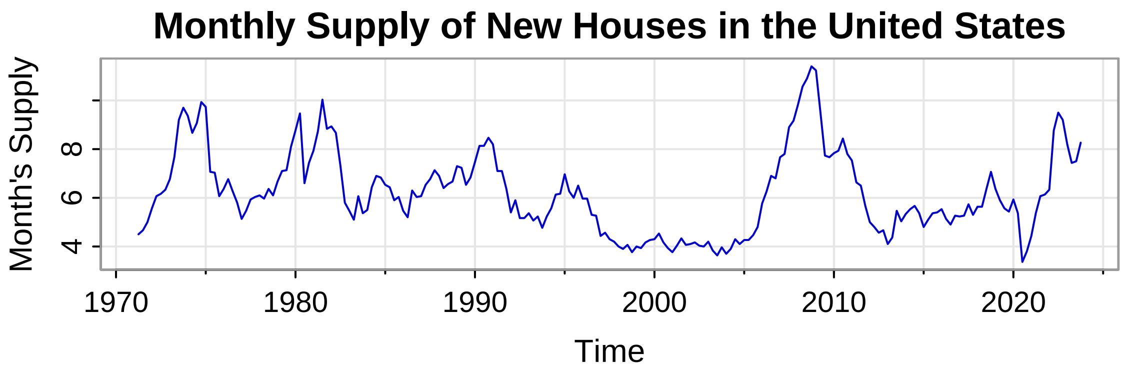
From the plot of Median Sales Price of Houses Sold, we can see that there exists an obvious trend in the data. The Median Prices will be adjusted for inflation based on 1982:1984 prices to make median housing prices from different years directly comparable. This is crucial for understanding long term trends in housing prices and assessing changes in affordability over time.

Moreover, adjusting median housing prices for inflation should lead to more accurate forecasts by avoid biases introduced by price shocks in commodity prices, the effect of which have not been included in the model.

Monthly Supply of New Houses in the United States (MSACSR) [Monthly] [Jan'63 - Dec'23]

Predictor variable 1

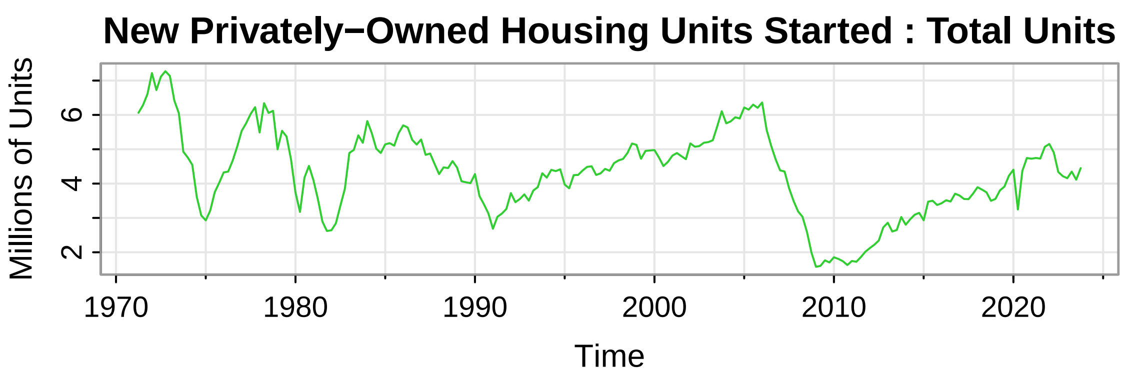
The months' supply is the ratio of new houses for sale to new houses sold. This statistic provides an indication of the size of the new for-sale inventory in relation to the number of new houses currently being sold. The months' supply indicates how long the current new for-sale inventory would last given the current sales rate if no additional new houses were built.



New Privately-Owned Housing Units Started: Total Units (HOUST) [Monthly] [Jan'59 - Jan'24]

Predictor variable 2

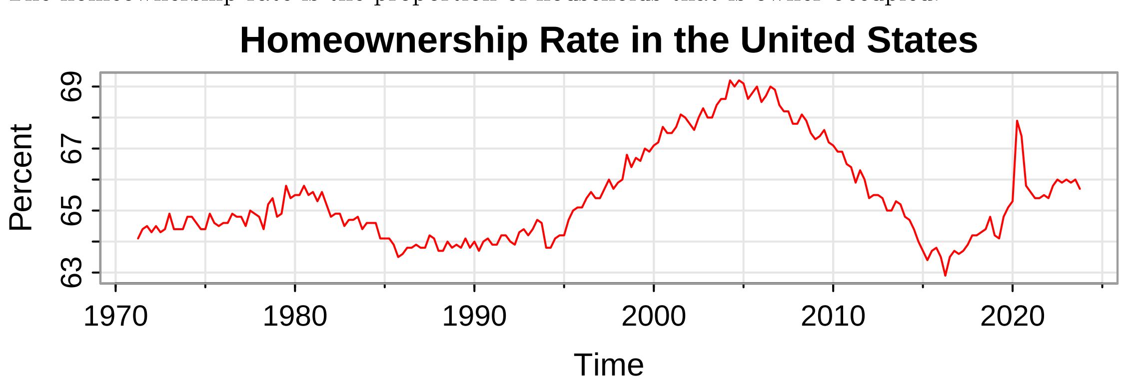
As provided by the Census, start occurs when excavation begins for the footings or foundation of a building. Increases in housing starts and permits indicate growing supply, which can help alleviate housing shortages and moderate price growth. Conversely, declines in construction activity may contribute to supply constraints and upward pressure on prices.



Homeownership Rate in the United States (RHORUSQ156N) [Quarterly] [Q1 '65 - Q4'23]

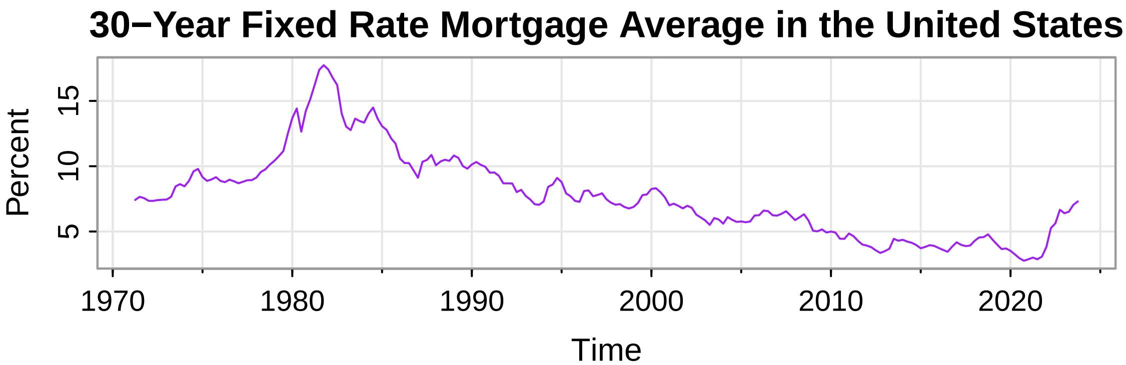
Predictor variable 3

The homeownership rate is the proportion of households that is owner-occupied.



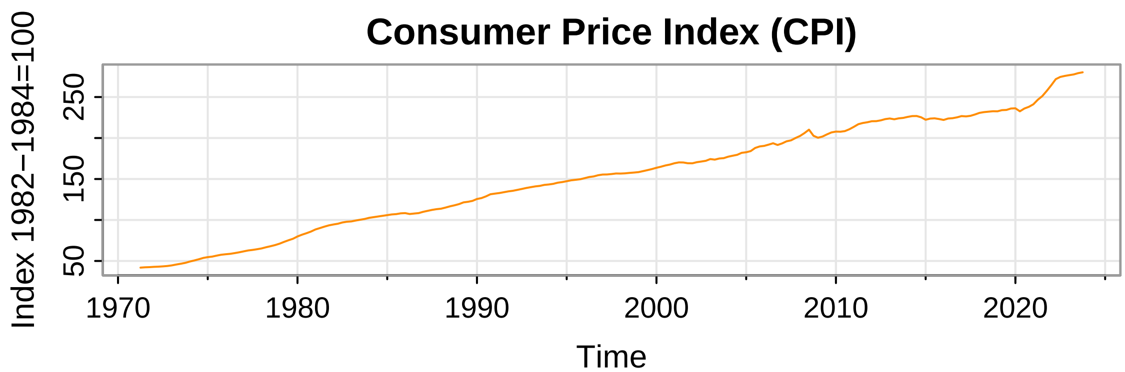
30-Year Fixed Rate Mortgage Average in the United States (MORTGAGE30US) [Weekly] [Apr'71 - Feb'24]

Predictor variable 4



Consumer Price Index for All Urban Consumers: All Items Less Shelter in U.S. City Average (CUSR0000SA0L2) [Monthly] [Jan'47 - Jan'24]

Additional Variable [NOT to be used as a predictor]



Questions to be answered

The primary objective is to predict the Median Sales price of houses sold in the United States based on four factors:

i) Monthly supply of New Houses in the United States
ii) New Privately-owned Housing units started
iii) Home ownership rate in the United States
iv) 30-Year Fixed Rate Mortgage Average in the United States

We will try to answer which (if any) of the four factors have an effect on the Median Sales price of houses sold. We will also try to answer which factor has the strongest effect on median prices. Analysis A will be used to answer these questions.

We will also try to gauge how volatile are prices to shocks in supply, housing starts, and Mortgage rates. Analysis C will consist of analyzing volatility patterns to better understand these factors.

Plans for Analysis

ANALYSIS A The time series data of median price [MSPUS] will be regressed on time *(t)* and the four other independent variables.

STEP 1) The Median Prices (outcome variable) will be pre processed by adjusting for inflation, followed by de-trending and log transformation to make it stationary and homoscedastic.

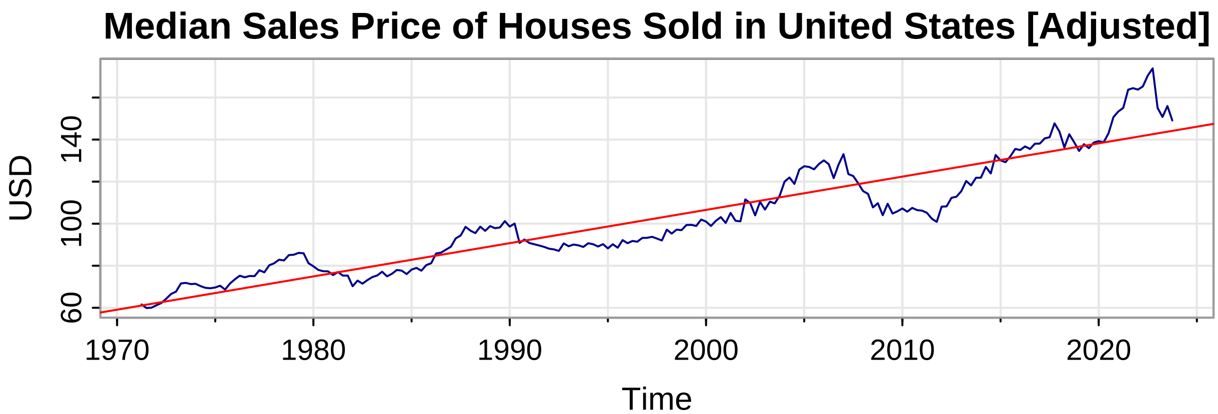
STEP 2) The four predictor variables will be converted into quarterly series if they are not already.

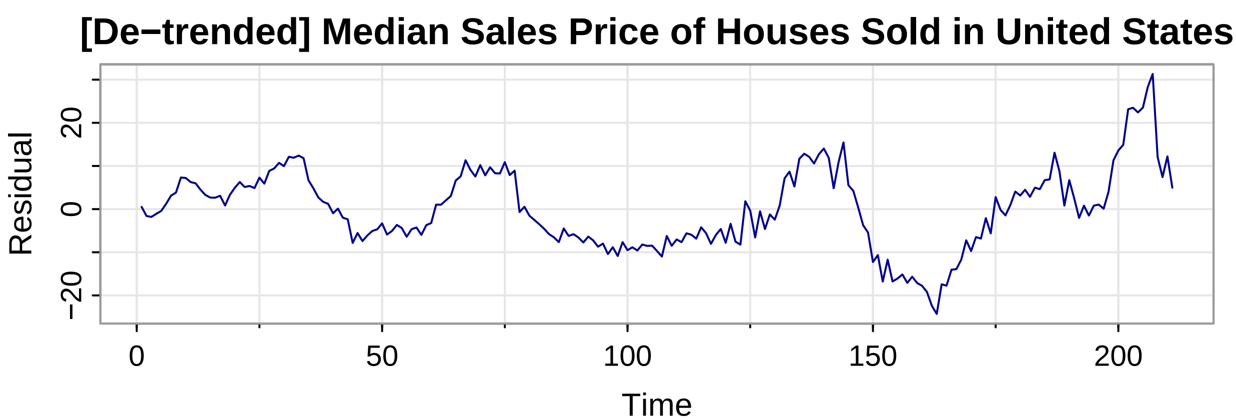
STEP 3) The Median Price will be regressed on time and other predictors to arrive at the full model.

STEP 3.5) Multiple model sizes will be analyzed to find the optimum model to be selected, based on AIC/BIC criteria.

STEP 4) Based on the p/ACF of the residuals, we may conduct regression with autocorrelated errors.

Viability Plots






```
##
##  Augmented Dickey-Fuller Test
##
## data:  resid(mspus_trend)
## Dickey-Fuller = -4.2328, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

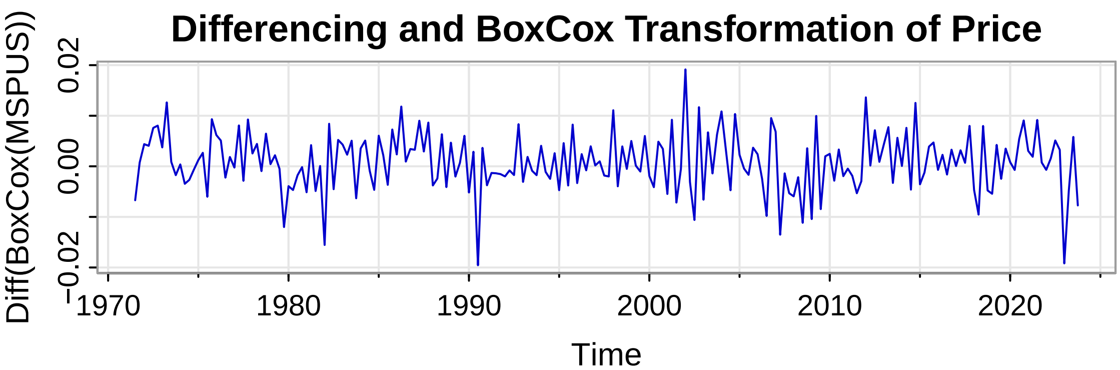
```
##
##  KPSS Test for Level Stationarity
```

```
##
## data:  resid(mspus_trend)
## KPSS Level = 0.28812, Truncation lag parameter = 4, p-value = 0.1
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(resid(mspus_trend) ~ time(resid(mspus_trend)))
## BP = 33.019, df = 1, p-value = 0.000000009128
```

ADF & KPSS tests conclude stationarity.

The series passes both these tests of stationarity. But the series exhibits an obvious heteroscedasticity (evidenced by BP-Test), where higher levels are associated with higher variation. A BoxCox transformation is recommended.




```
##
##  Augmented Dickey-Fuller Test
##
## data:  mspus.box.diff
## Dickey-Fuller = -4.7339, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  mspus.box.diff
## KPSS Level = 0.061236, Truncation lag parameter = 4, p-value = 0.1
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(mspus.box.diff ~ time(mspus.box.diff))
## BP = 1.0509, df = 1, p-value = 0.3053
```

The differenced and BoxCox transformed series passes tests of stationarity and is homoscedastic.

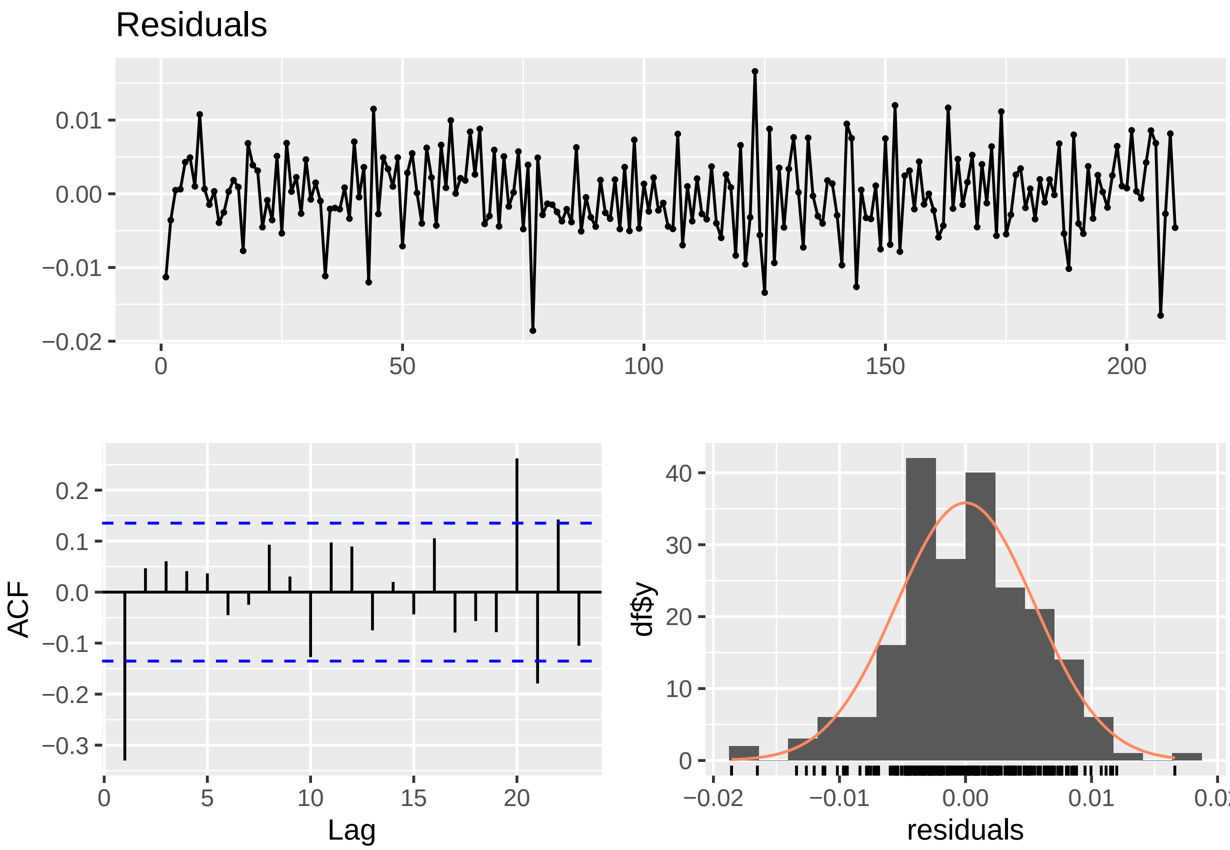
We can now proceed with regression.

We will use the step-wise algorithm to find the optimum model.

```
## Start:  AIC=-2178.18
## mspus.box.diff ~ time_mspus + houst.ts1 + mortgage.ts1 + msacsr.ts1 +
##     rhorusq156n.ts1
##
##                   Df   Sum of Sq       RSS     AIC
## - rhorusq156n.ts1  1 0.000048741 0.0062543 -2178.5
## - time_mspus       1 0.000049118 0.0062547 -2178.5
## - houst.ts1        1 0.000055033 0.0062606 -2178.3
## <none>                           0.0062056 -2178.2
## - mortgage.ts1     1 0.000101557 0.0063071 -2176.8
## - msacsr.ts1       1 0.000199669 0.0064052 -2173.5
##
## Step:  AIC=-2178.53
## mspus.box.diff ~ time_mspus + houst.ts1 + mortgage.ts1 + msacsr.ts1
##
##                   Df   Sum of Sq       RSS     AIC
## - houst.ts1        1 0.000033771 0.0062881 -2179.4
## <none>                           0.0062543 -2178.5
## + rhorusq156n.ts1  1 0.000048741 0.0062056 -2178.2
## - time_mspus       1 0.000087123 0.0063414 -2177.6
## - mortgage.ts1     1 0.000100425 0.0063547 -2177.2
## - msacsr.ts1       1 0.000217177 0.0064715 -2173.4
##
## Step:  AIC=-2179.4
## mspus.box.diff ~ time_mspus + mortgage.ts1 + msacsr.ts1
##
##                   Df  Sum of Sq       RSS     AIC
## <none>                          0.0062881 -2179.4
## + houst.ts1        1 0.00003377 0.0062543 -2178.5
## + rhorusq156n.ts1  1 0.00002748 0.0062606 -2178.3
## - mortgage.ts1     1 0.00010843 0.0063965 -2177.8
## - time_mspus       1 0.00016324 0.0064513 -2176.0
## - msacsr.ts1       1 0.00039291 0.0066810 -2168.7
```

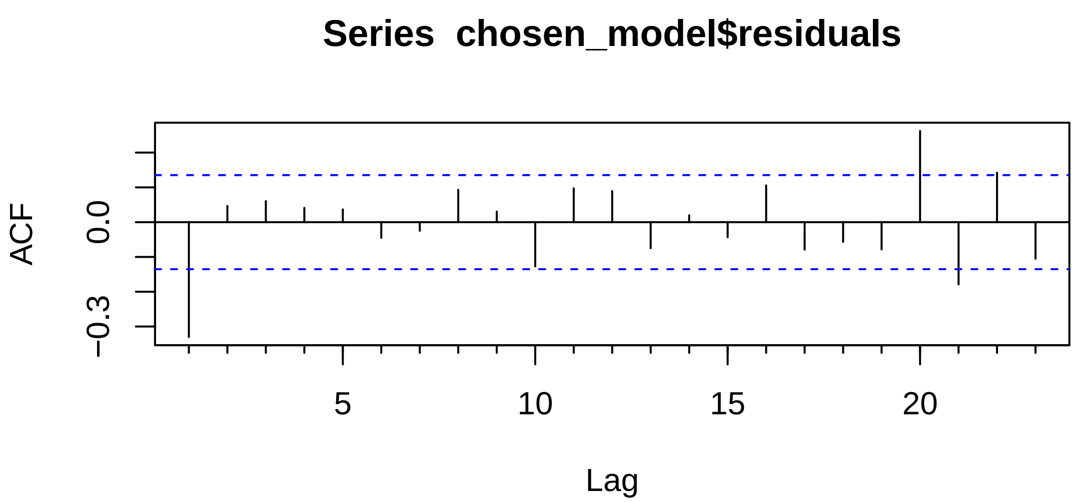
Thus we conclude that Predictor 1 (MSACSR) and Predictor 4 (MORTGAGE30US) are significant predictors of housing prices based on AIC criteria.

We will now carry out residual analysis:

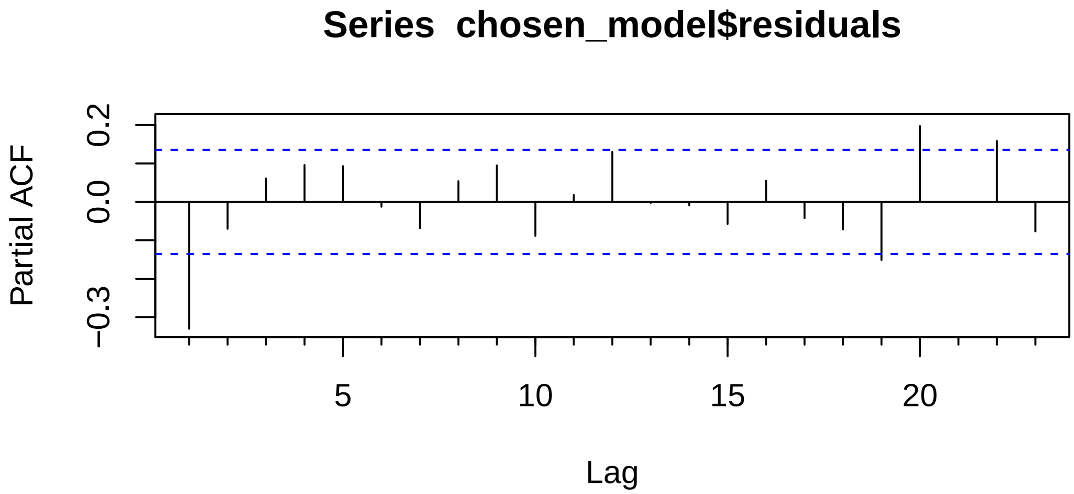



```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 31.428, df = 10, p-value = 0.0004985
##
## Model df: 0.   Total lags used: 10
```

The Ljung-Box test concludes that the residuals are not independently distributed; they exhibit serial correlation. We will carry out Regression with autocorrelated errors.

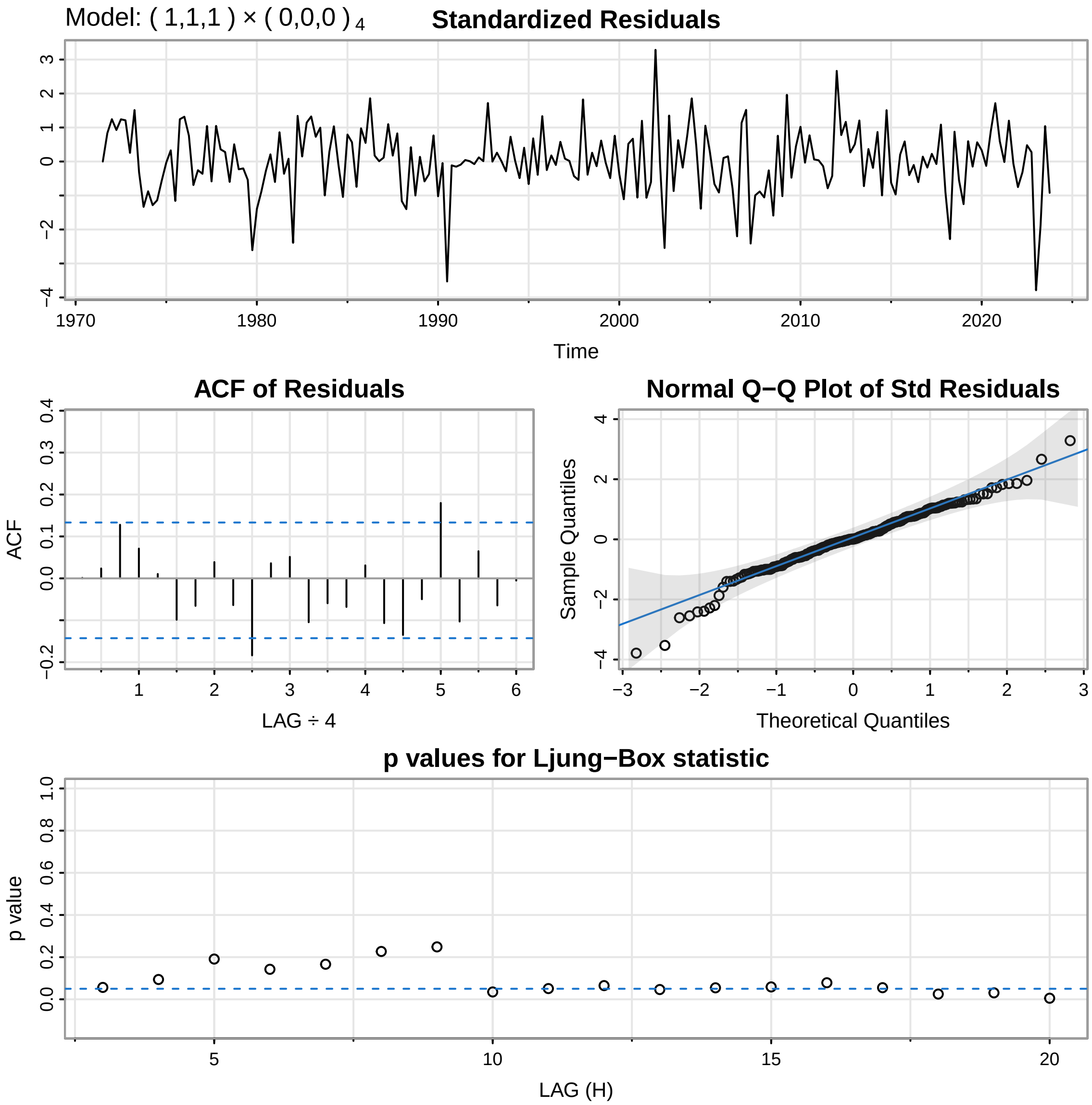


ACF cuts off after 1.



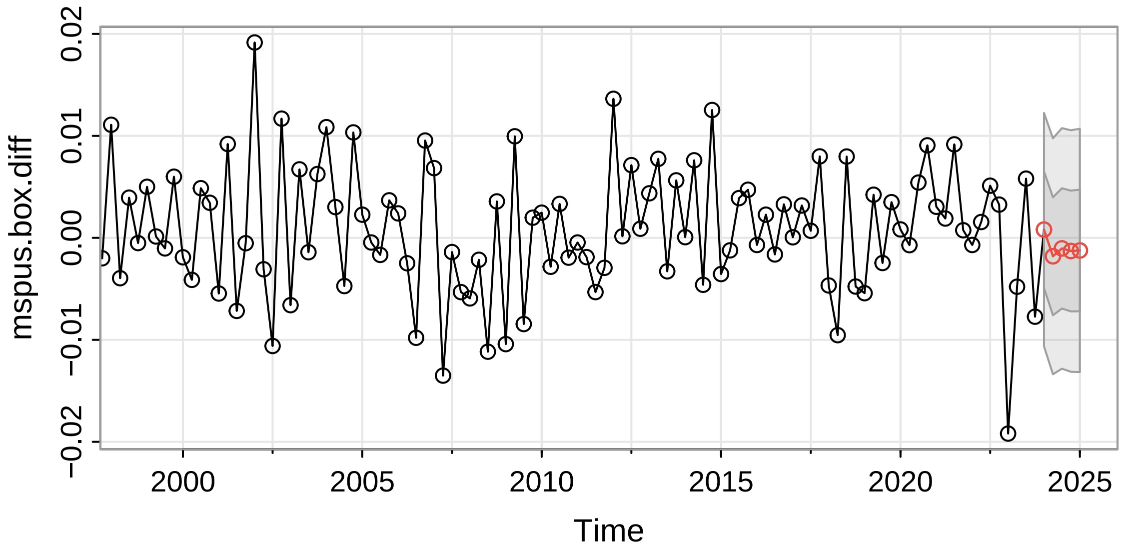
PACF cuts off after 1

ARMA(1,1) looks like a good fit for the residuals. We will fit a ARMA(1,1) model and carry out forecasting.

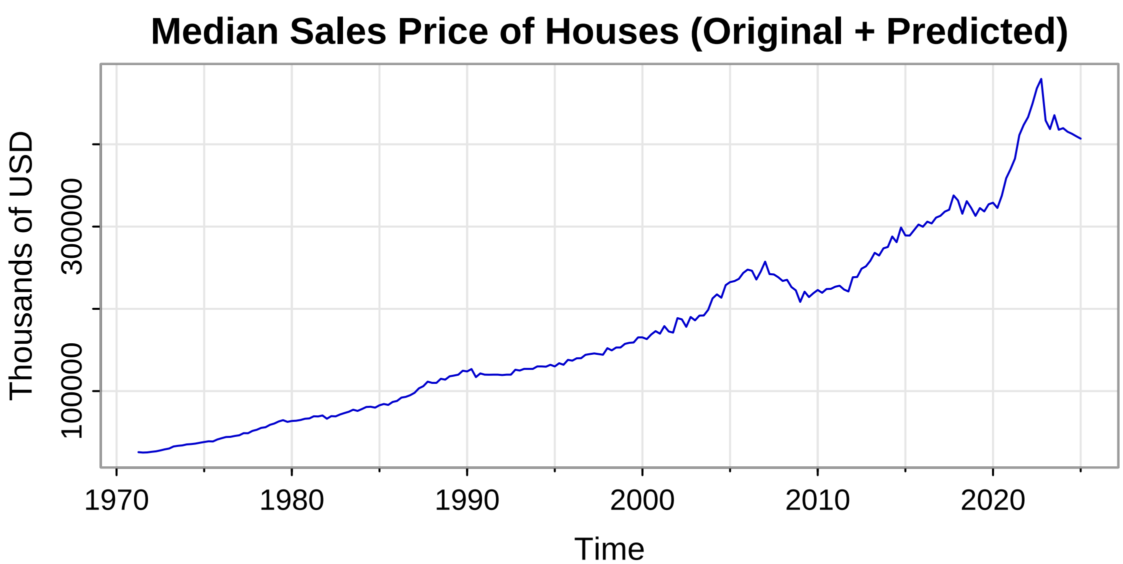


The ACF of residuals show that they now resemble white noise.

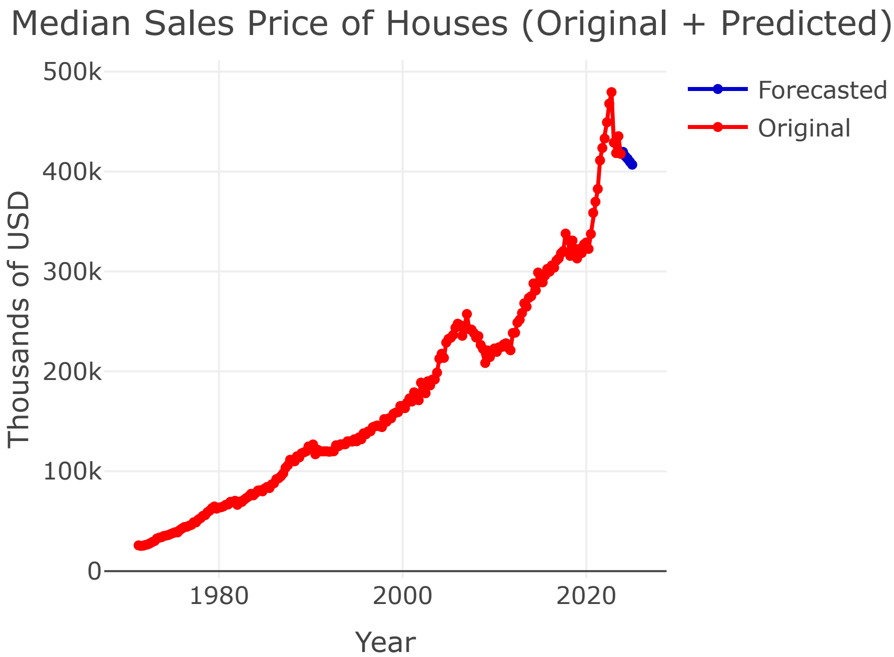
We will carry out the forecasting of the stationary series using *sarima.for()* function.




```
## [1] "Forecasted Values: "

##          Qtr1     Qtr2     Qtr3     Qtr4
## 2024 419656.1 415300.8 412832.8 409781.8
## 2025 406890.7
```




```
data_df <- data.frame(Time = time(new_mspus), Sales = as.numeric(new_mspus), Type = "Original")
data_df$Type[(length(mspus.ts) + 1):nrow(data_df)] <- "Forecasted"
p <- plot_ly(data_df, x = ~Time, y = ~Sales, type = 'scatter', mode = 'lines+markers',
             color = ~Type, colors = c('mediumblue', 'red'),
             line = list(shape = "linear"),
             marker = list(size = 5)) %>%
    layout(title = "Median Sales Price of Houses (Original + Predicted)",
           xaxis = list(title = "Year"),
           yaxis = list(title = "Thousands of USD"))
p
```



Basic Ideas about Analysis C

As can be seen from the plots of predictor variables *(i), (ii), & (iv)*, there exists "shocks" in these variables. Analysis C will focus on understanding how volatile are median prices to shocks in 'Monthly Supply of New Houses'*(supply side)*, 'New Privately Owned Housing Units Started' *(demand side)*, and '30 Year Fixed Rate Mortgage' *(Cost of Borrowing)*.

In particular, we can try to explore two ideas:

1) **Volatility Patterns:** How does volatility in housing prices and related variables behave over different frequencies? Spectral analysis can reveal cyclical patterns in volatility, while GARCH, APARCH, and IGARCH models can help identify and model the conditional heteroskedasticity in the data.

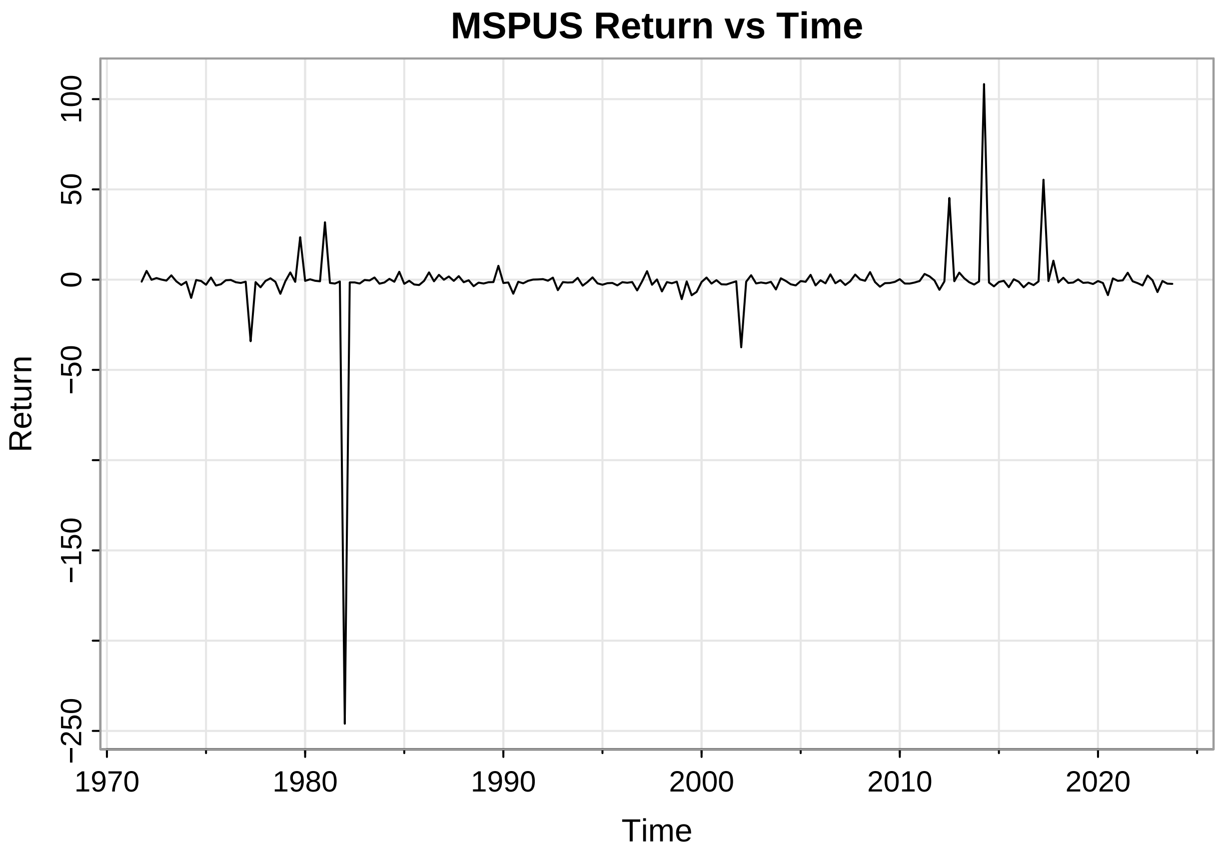
2) **Asymmetry in Volatility:** Are there asymmetries in the response of housing prices to shocks? APARCH models are particularly useful for capturing asymmetries in volatility, allowing for a more nuanced understanding of how positive and negative shocks impact housing market dynamics differently.

The spectral analysis will require decomposing the time series into its constituent frequencies while ARCH/GARCH models will be used to model the return/growth rate of median prices.

GARCH models assume stationarity. Therefore we will use the log-differenced series for GARCH model which is stationary.

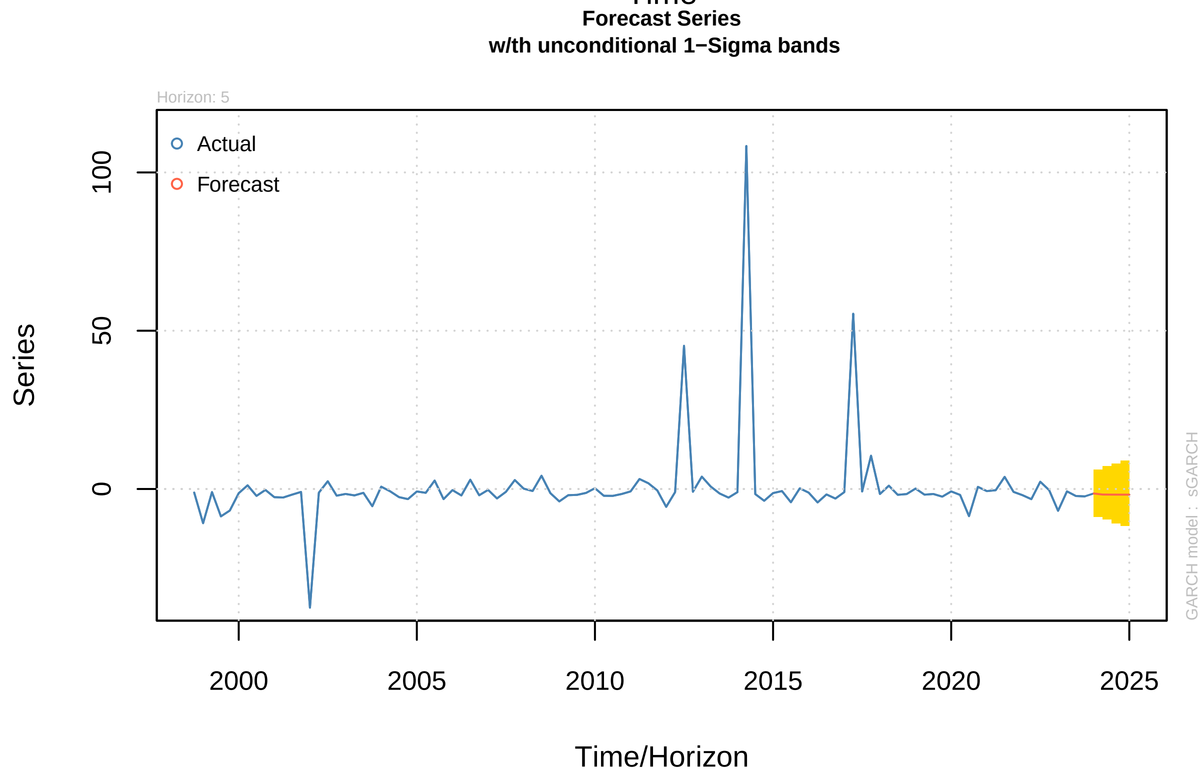
We will need to evaluate the mean model ARMA Order and GARCH order simultaneously to fit a ARMA-GARCH model.The mean model uses ARMA that generates the forecast for the mean of the time series, while the GARCH model generates the forecast for the variance.

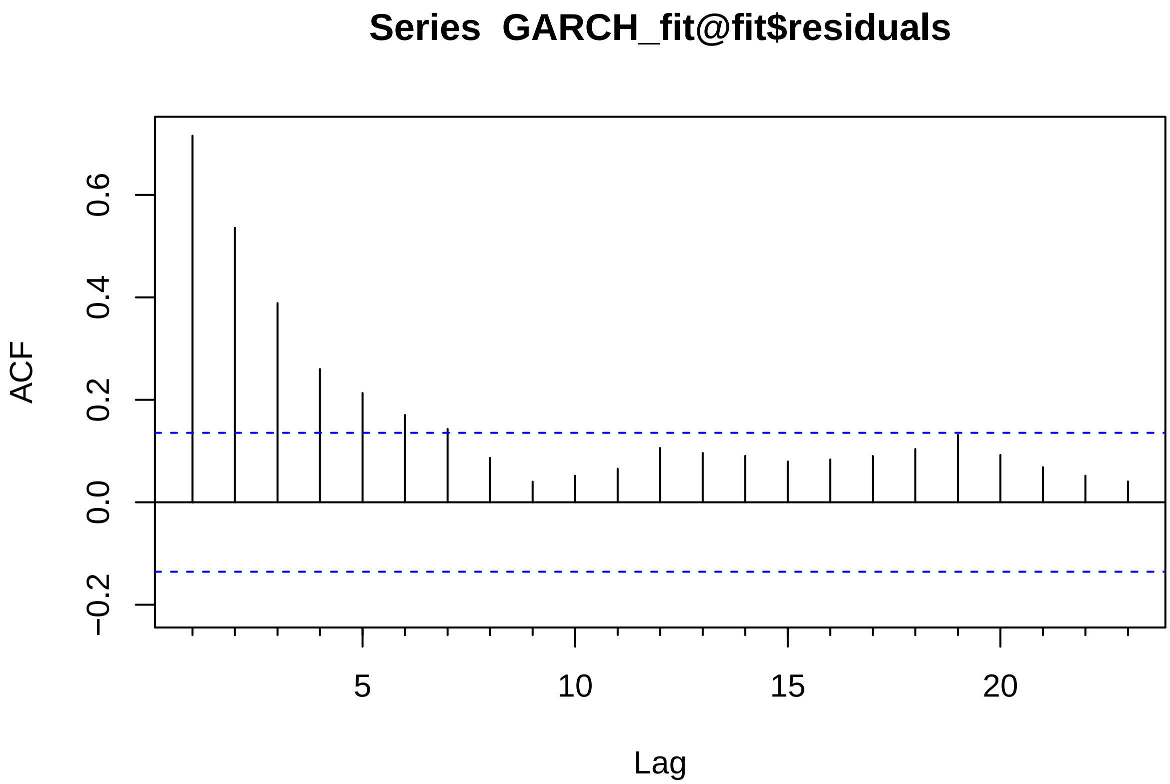
The mean model will use an ARMA(1,1) and we will use different GARCH orders to find the best log-likelihood value to evaluate goodness of fit.

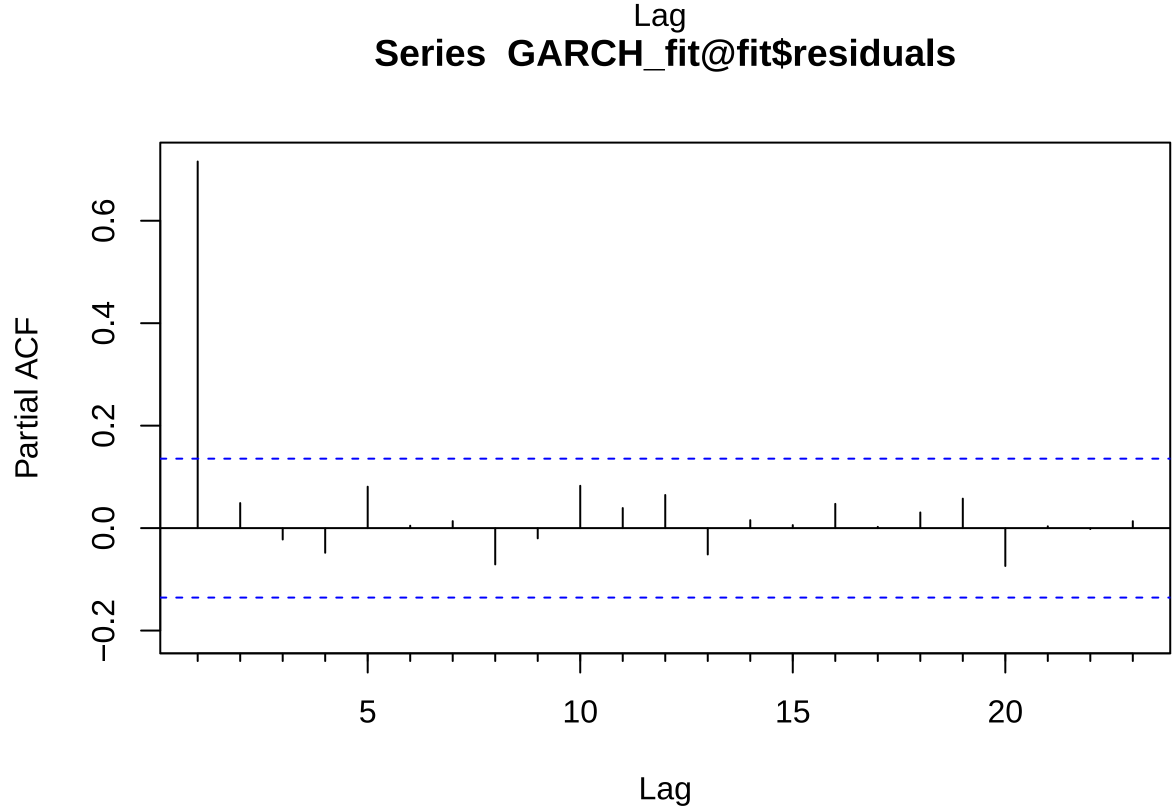



```
## Warning in adf.test(mspus_returns): p-value smaller than printed p-value

## Warning in kpss.test(mspus_returns): p-value greater than printed p-value
```


Series with 2 Conditional SD Superimposed
Returns
50
0
−50
−150
−250
1970
1980
1990
2000
2010
2020
Time
GARCH model : sGARCH


Forecast Series
w/th unconditional 1−Sigma bands
Horizon: 5
Actual
Forecast
Series
100
50
0
2000
2005
2010
2015
2020
2025
Time/Horizon
GARCH model : sGARCH



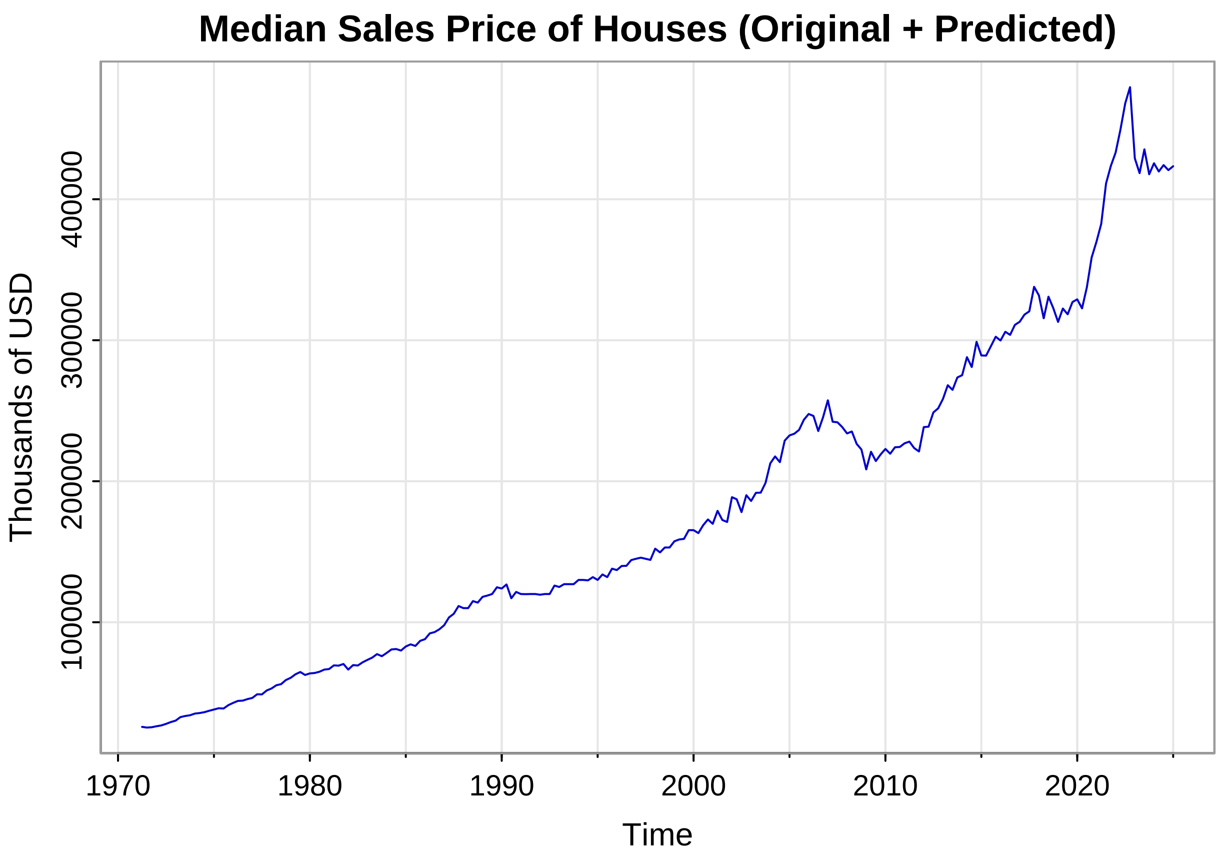



```
# Reverse calculating values and plotting them
X <- mspus.box.diff[210]

for (i in 1:length(GARCH_for@forecast$seriesFor)) {
```

```r
  diff <- 0
  diff <- X[i] * GARCH_for@forecast$seriesFor[i]
  X[i+1] <- X[i] + diff
}

GARCHmspus.box.diff <- ts(c(mspus.box.diff,X),start = c(1971, 3), frequency = 4)
cumsum(c(mspus.ts.boxcox[211],X[-1]))
```

```
## [1] 2.359119 2.362323 2.359925 2.361789 2.360336 2.361469
```

```r
garch_new_vals <- cumsum(c(mspus.ts.boxcox[211],X[-1]))
garch_forecast1_adj <- inv_boxcox(garch_new_vals[-1], lambda = optim_lambda)
garch_mspus_pred1 <- (cusr.ts[length(cusr.ts)]/100*garch_forecast1_adj)*1000
garch_new_mspus <- ts(c(mspus.ts,garch_mspus_pred1),start = c(1971, 2), frequency = 4)
tsplot(garch_new_mspus, main = "Median Sales Price of Houses (Original + Predicted)",
       ylab = "Thousands of USD", col = "mediumblue")
```



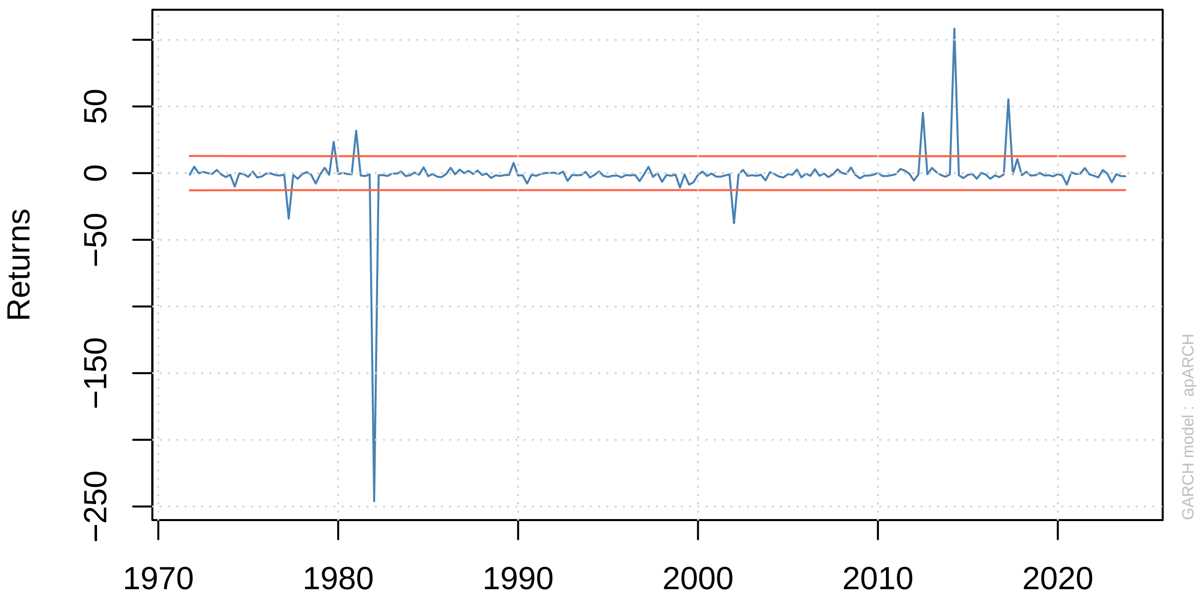

```r
print("Forecasted Values: ")
```

```
## [1] "Forecasted Values: "
```

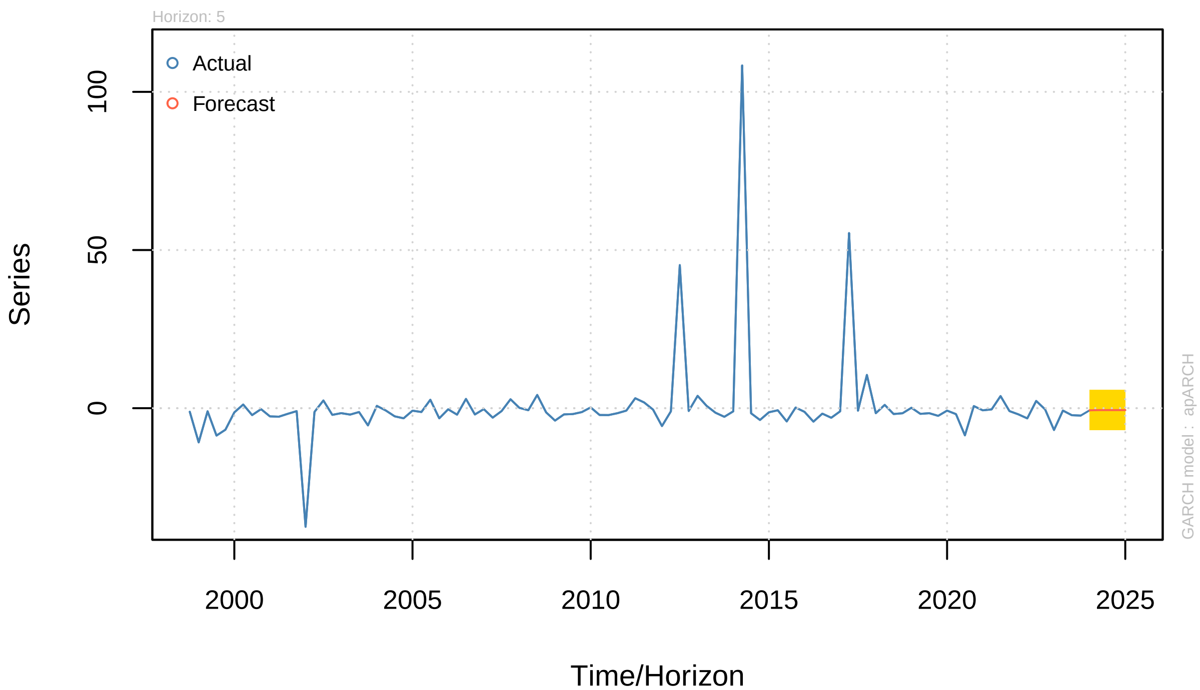
```r
tail(garch_new_mspus, 5)
```

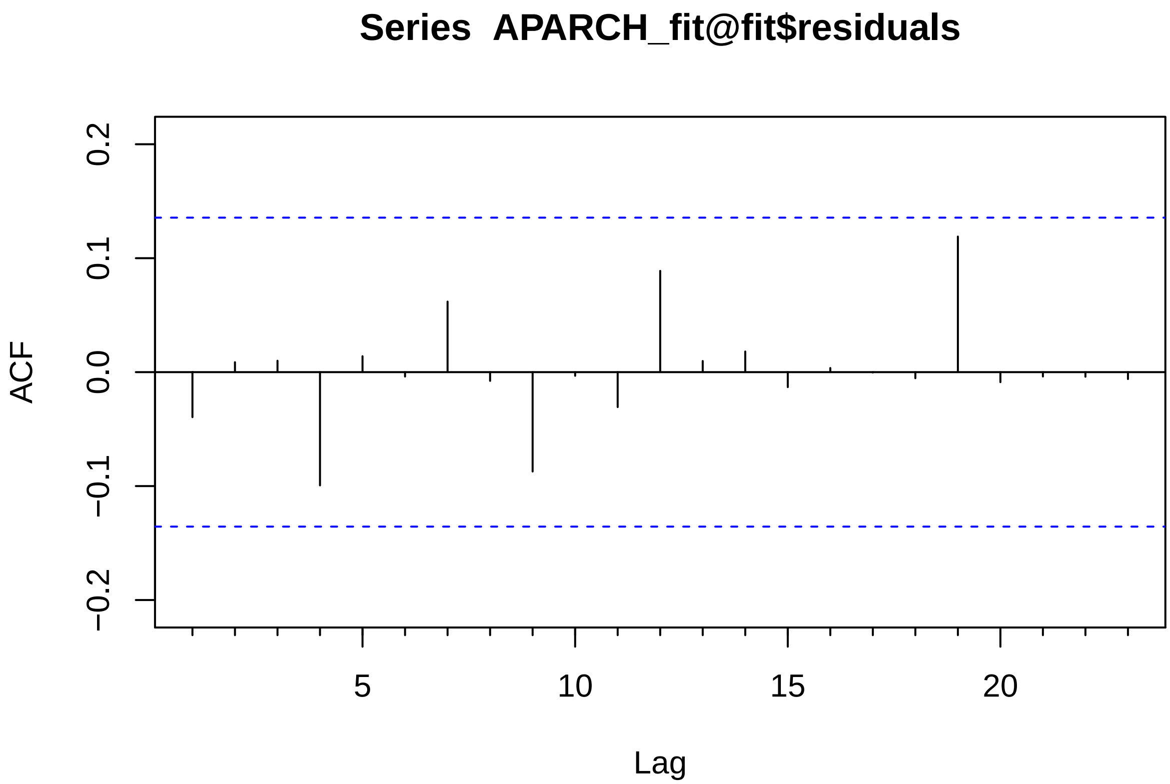
```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 2024 425531.7 419652.4 424211.7 420651.6
## 2025 423424.3
```

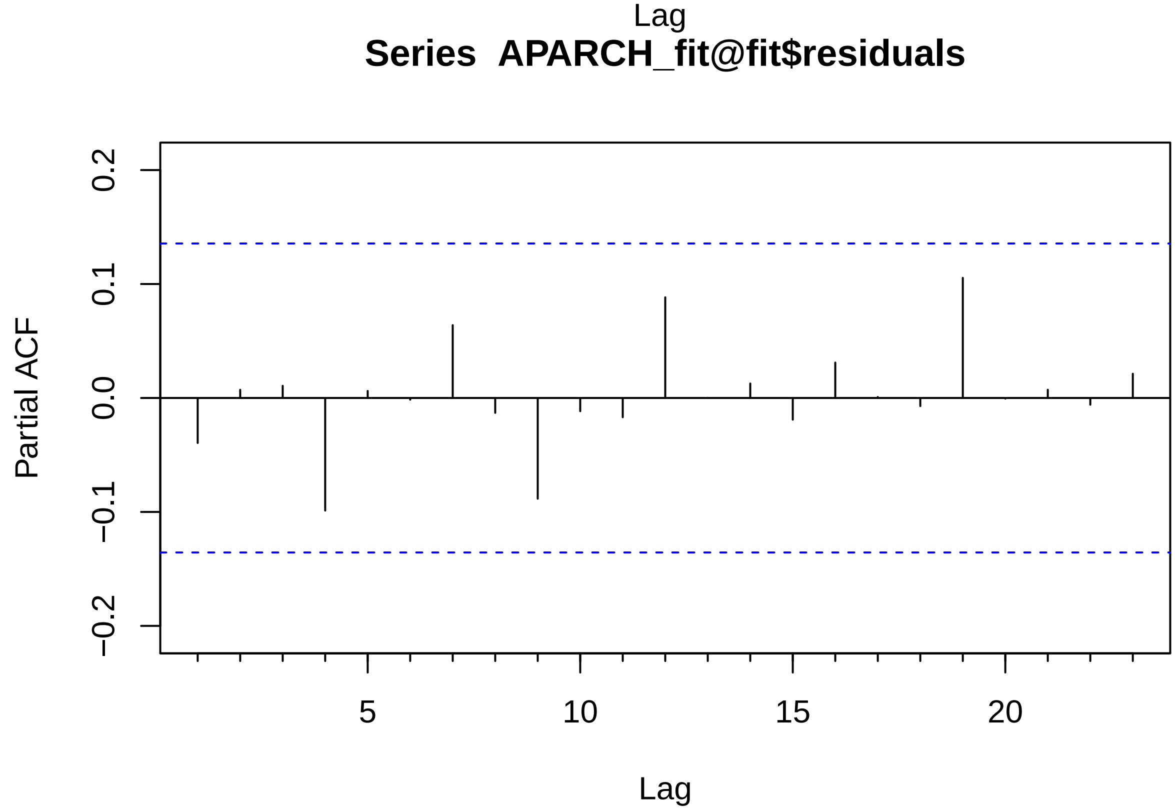
Series with 2 Conditional SD Superimposed



Forecast Series
w/th unconditional 1−Sigma bands





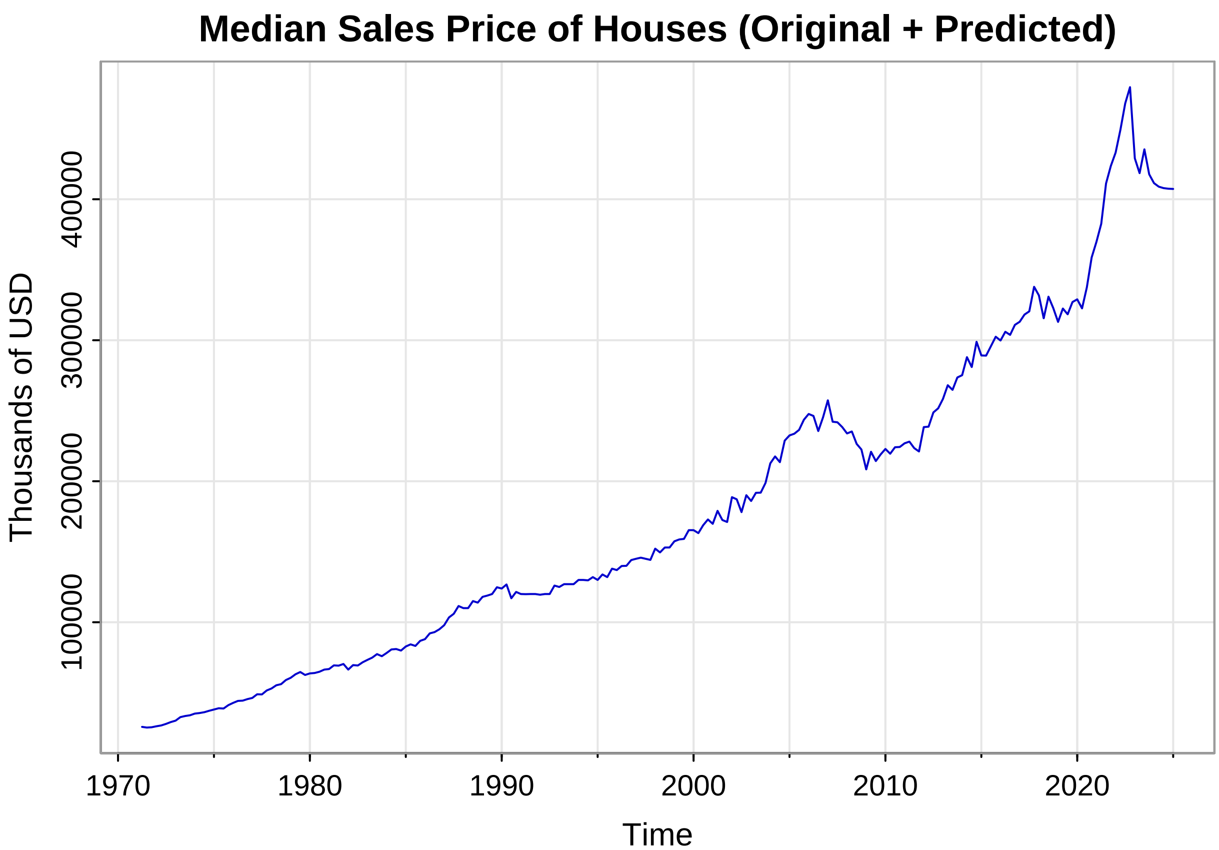



```
# Reverse calculating values and plotting them
X <- mspus.box.diff[210]

for (i in 1:length(APARCH_for@forecast$seriesFor)) {
```

```r
  diff <- 0
  diff <- X[i] * APARCH_for@forecast$seriesFor[i]
  X[i+1] <- X[i] + diff
}

APARCHmspus.box.diff <- ts(c(mspus.box.diff,X),start = c(1971, 3), frequency = 4)
cumsum(c(mspus.ts.boxcox[211],X[-1]))
```

```
## [1] 2.359119 2.356493 2.355425 2.354988 2.354810 2.354738
```

```r
aparch_new_vals <- cumsum(c(mspus.ts.boxcox[211],X[-1]))
aparch_forecast1_adj <- inv_boxcox(aparch_new_vals[-1], lambda = optim_lambda)
aparch_mspus_pred1 <- (cusr.ts[length(cusr.ts)]/100*aparch_forecast1_adj)*1000
aparch_new_mspus <- ts(c(mspus.ts,aparch_mspus_pred1),start = c(1971, 2), frequency = 4)
tsplot(aparch_new_mspus, main = "Median Sales Price of Houses (Original + Predicted)",
       ylab = "Thousands of USD", col = "mediumblue")
```




```r
print("Forecasted Values: ")
```

```
## [1] "Forecasted Values: "
```

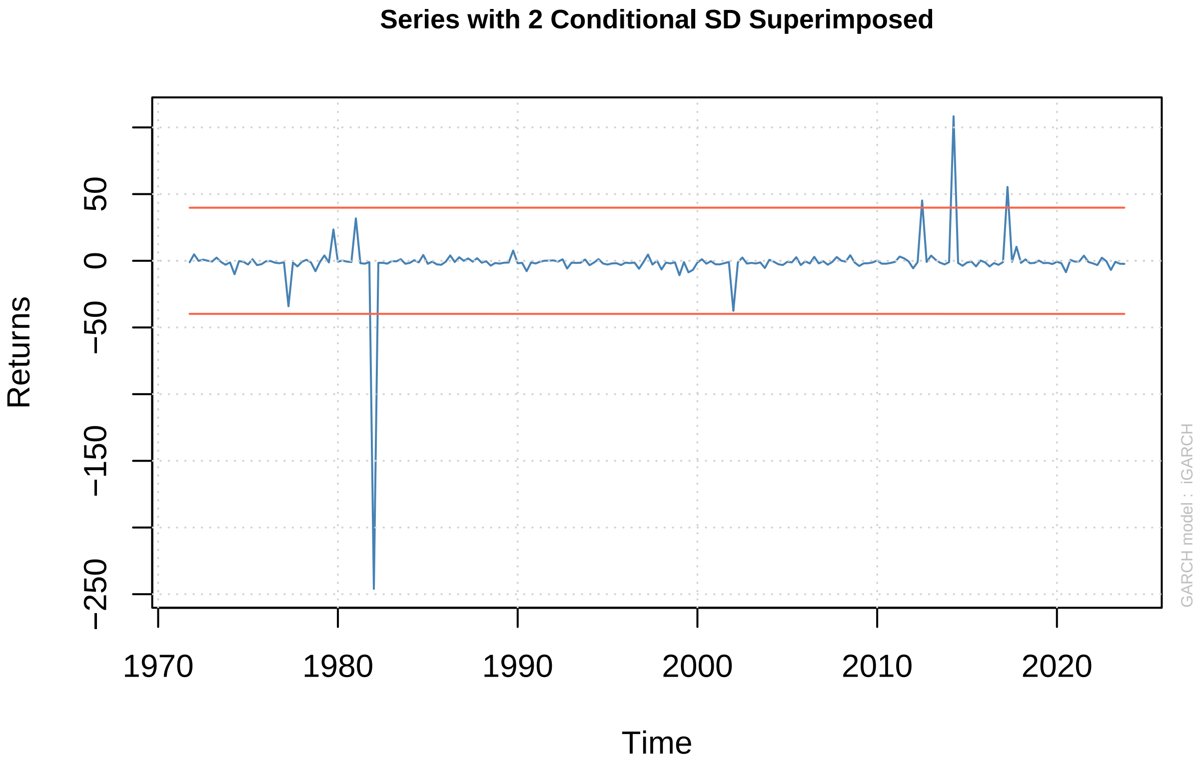
```r
tail(aparch_new_mspus, 5)
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 2024 411424.4 408907.8 407886.5 407470.6
## 2025 407301.0
```

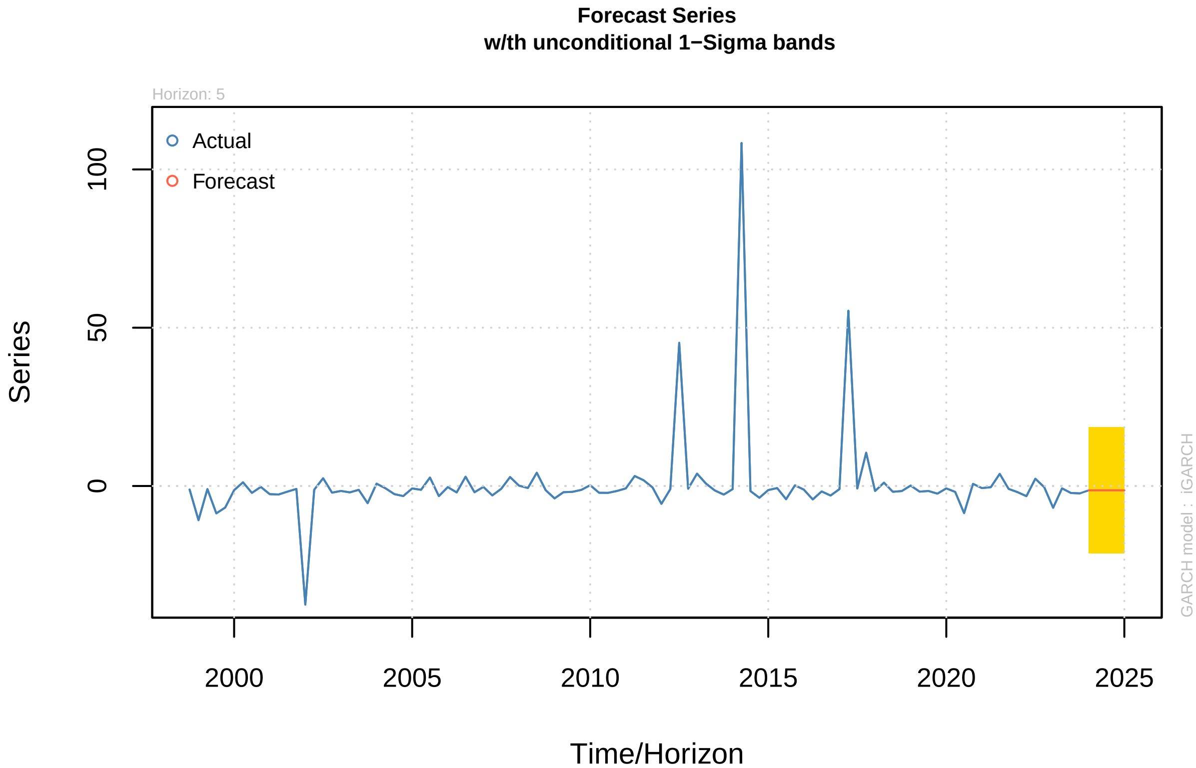
```
IGARCH_spec <- ugarchspec(variance.model = list(model = "iGARCH", garchOrder = c(0,1)),
                   mean.model = list(armaOrder = c(1,1), include.mean = TRUE)) # Integrated GARCH
IGARCH_fit <- ugarchfit(spec = IGARCH_spec, data = mspus_returns)
IGARCH_for <- ugarchforecast(IGARCH_fit, n.ahead = 5)
```

```
## Warning in `setfixed<-`(`*tmp*`, value = as.list(pars)): Unrecognized Parameter
## in Fixed Values: beta1...Ignored
```

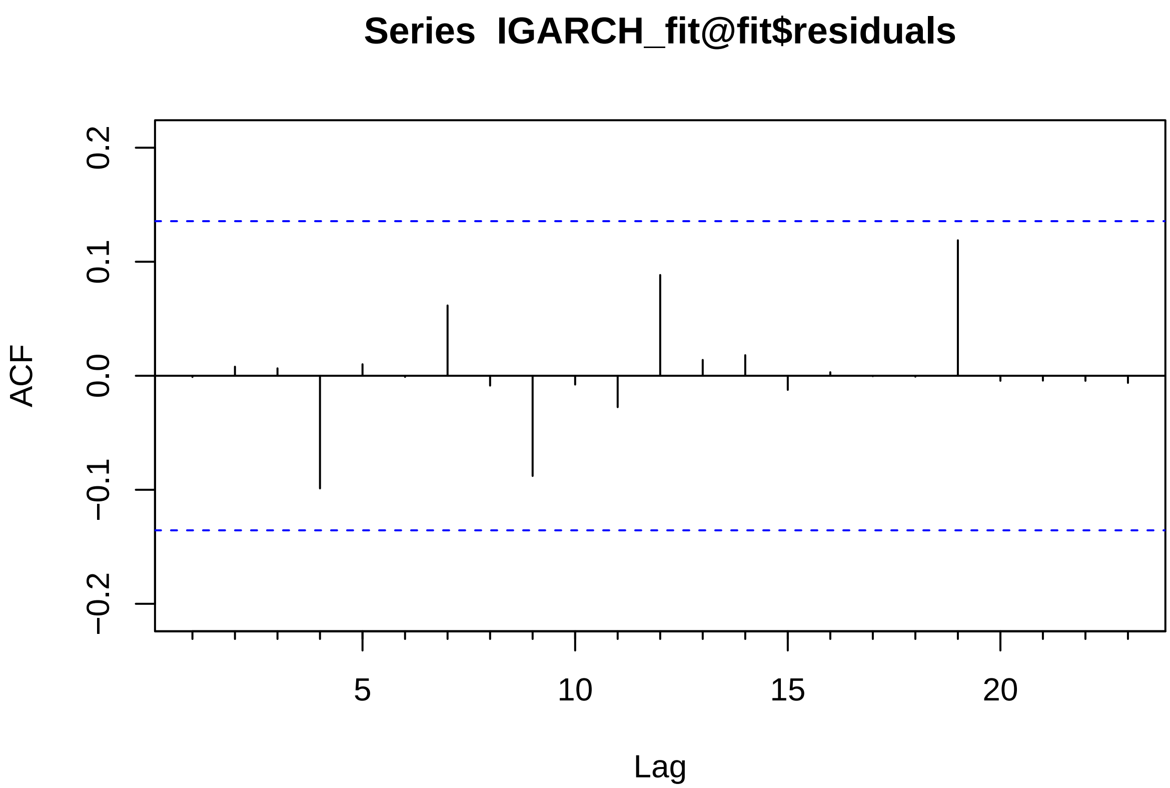
```
plot(IGARCH_fit, which = 1)
```



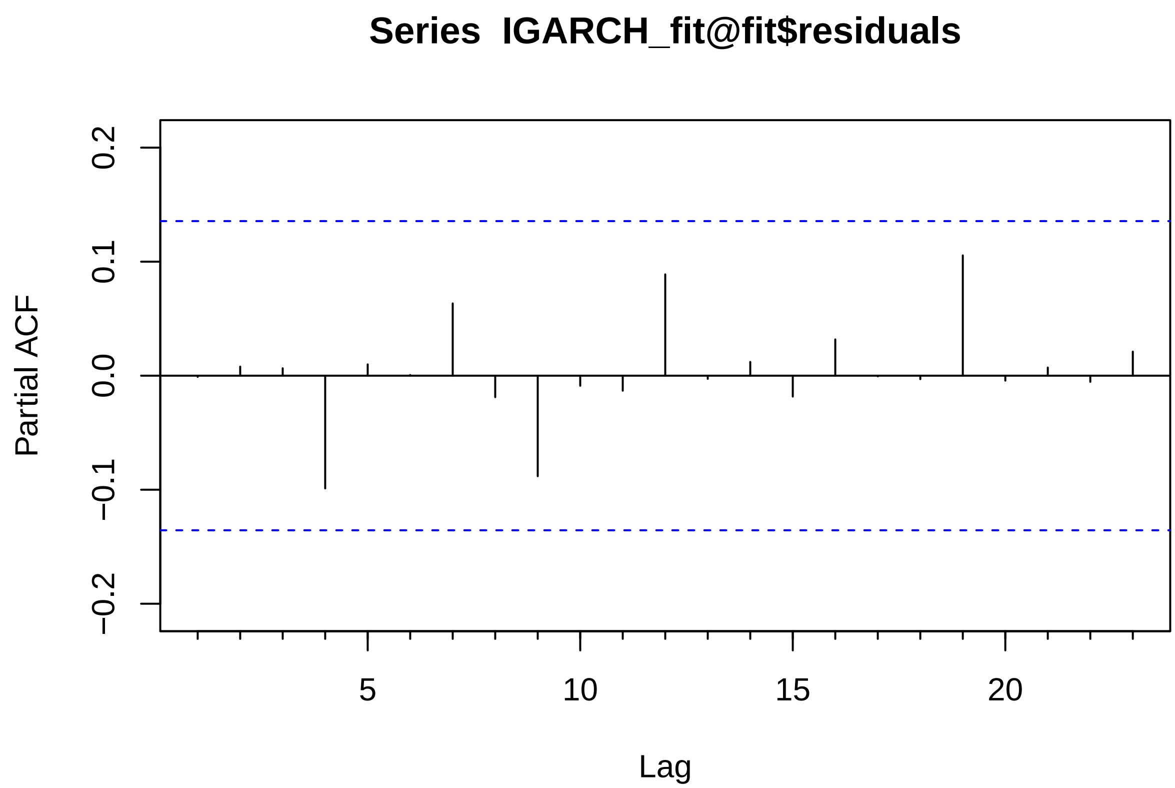

```
plot(IGARCH_for, which = 1)
```




```r
Acf(IGARCH_fit@fit$residuals)
```



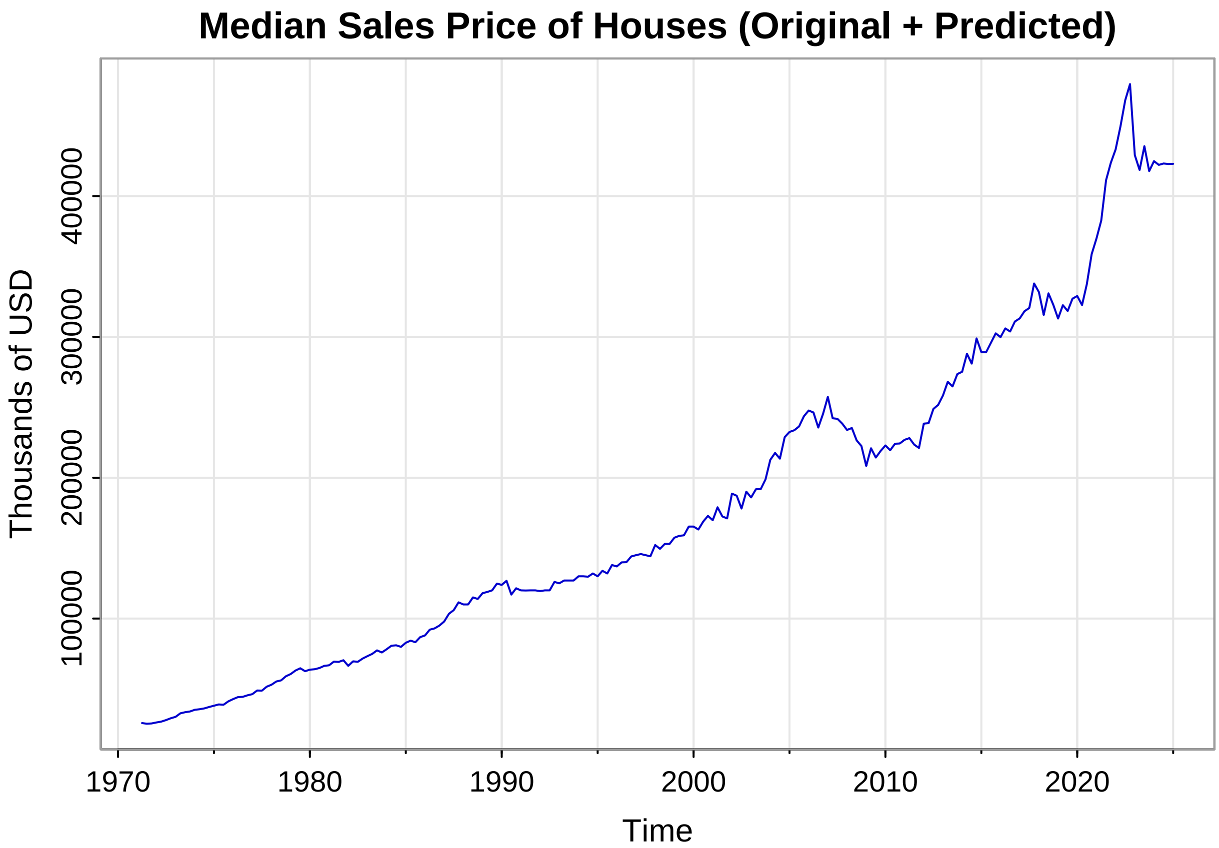
```r
Pacf(IGARCH_fit@fit$residuals)
```




```r
# Reverse calculating values and plotting them
X <- mspus.box.diff[210]

for (i in 1:length(IGARCH_for@forecast$seriesFor)) {
  diff <- 0
  diff <- X[i] * IGARCH_for@forecast$seriesFor[i]
  X[i+1] <- X[i] + diff
}

IGARCHmspus.box.diff <- ts(c(mspus.box.diff,X),start = c(1971, 3), frequency = 4)
cumsum(c(mspus.ts.boxcox[211],X[-1]))
```

```
## [1] 2.359119 2.362019 2.360936 2.361340 2.361190 2.361246
```

```r
igarch_new_vals <- cumsum(c(mspus.ts.boxcox[211],X[-1]))
igarch_forecast1_adj <- inv_boxcox(igarch_new_vals[-1], lambda = optim_lambda)
igarch_mspus_pred1 <- (cusr.ts[length(cusr.ts)]/100*igarch_forecast1_adj)*1000
igarch_new_mspus <- ts(c(mspus.ts,igarch_mspus_pred1),start = c(1971, 2), frequency = 4)
tsplot(igarch_new_mspus, main = "Median Sales Price of Houses (Original + Predicted)",
       ylab = "Thousands of USD", col = "mediumblue")
```



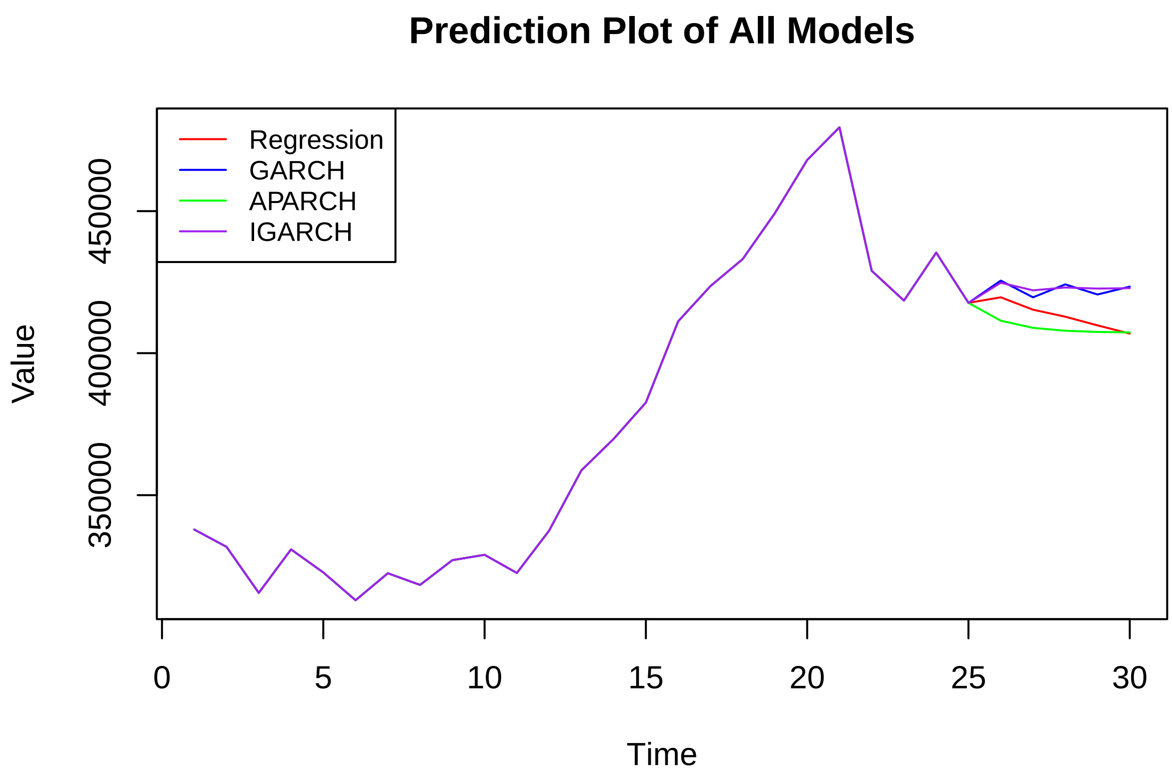

```
print("Forecasted Values: ")
```

```
## [1] "Forecasted Values: "
```

```
tail(igarch_new_mspus, 5)
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 2024 424779.4 422118.3 423107.9 422738.6
## 2025 422876.2
```

```
t <- 1:30
plot(t, tail(new_mspus,30), type = "l", col = "red", xlab = "Time", ylab = "Value",
     main = "Prediction Plot of All Models")

lines(t, tail(garch_new_mspus,30), col = "blue")
lines(t, tail(aparch_new_mspus,30), col = "green")
lines(t, tail(igarch_new_mspus,30), col = "purple")

legend("topleft", legend = c("Regression", "GARCH", "APARCH", "IGARCH"),
       col = c("red", "blue", "green", "purple"), lty = 1, cex = 0.8)
```


Prediction Plot of All Models
Regression
GARCH
APARCH
IGARCH
Value
350000
400000
450000
Time
0
5
10
15
20
25
30