Modeling and Forecasting U.S. Housing Sales

Grace Wu
gracewu@ucsb.edu

PSTAT 174

June 14, 2023

Abstract

Housing prices in the U.S. have changed drastically over time, with unpredictable drops and rises caused by changes in the economy. In this project, we will predict U.S. housing sales by applying time series methods to identify any patterns and trends in housing prices. We will be performing tests for the following two models: the Seasonal Autoregressive Integrated Moving Average SARIMA(p,d,q)$(P, D, Q)_s$ model and the Unit Root test to select the best fit model through estimating parameters determined by the autocorrelation and partial autocorrelation plots and model selection criteria values of our transformed time series data. To further confirm that the model we have selected is a good fit, we will run diagnostic checks as well. Something important to note in this prediction of U.S. housing sales is that while our model can determine trends in the housing sales time series data and account for them through transforming our data when selecting model parameters, the economy is unpredictable and subject to change any day. This means that unexpected societal and economic changes, such as the Covid-19 virus, caused drastic changes in housing sales, but our model couldn't have predicted that such a large outbreak would happen in 2020. Thus, in our future studies of housing sales, I hope to perform more advanced model testing that could possibly capture such economic changes.

Introduction

Last month, my parents bought another house. However, this decision was made over a long period of time. I overheard their discussion of "When is it a good time to purchase a house?", and this inspired me to write my project on predicting housing sales. At first, I didn't understand when my parents would say that it is not a good time to buy a house as the housing market would be bad at certain times of the year. However, I now understand what they mean after analyzing the housing sales time series data.

To take into account patterns and trends in the housing sales data, I will perform transformations on the data to make it stationary before applying two models to predict U.S. housing sales. For my first SARIMA model, I will estimate parameters by analyzing the ACF and PACF plot from my difference log transformation of the housing sales time series data and select the parameters by choosing the ones that result in the smallest AIC and BIC (model selection criteria) values. To select the best model fit from the two possible SARIMA models I chose, I will run diagnostic checks by examining the residual plot, normal QQ plot, Box-Pierce test, and Shapiro test. Then, I will perform the unit root test for my second model to further validate that the best fit SARIMA model I chose indicates a good fit. I will run the Dickey-Fuller, Augmented Dickey-Fuller, and Phillips-Perron Unit Root Test to show that the SARIMA model I selected through the difference log transformation does not have a unit root present, confirming that my selected SARIMA model is stationary.

By using the SARIMA model and Unit Root Test, I will forecast future U.S. housing sale prices and analyze how well my model was able to predict the prices.

Data

The original dataset contains 241 values of quarterly data for the average sales price of houses sold for the U.S. from January 1, 1963 to January 1, 2023, ranging from 0 to 600,000 U.S. dollars. This data was sourced and collected by the U.S. Census Bureau and U.S. Department of housing and Urban Development and retrieved from FRED, Federal Reserve Bank of St. Louis.

Link to dataset: https://fred.stlouisfed.org/series/ASPUS

This dataset is important because it shows us the past trend of housing sales price over time. Through studying this dataset, we can acquire valuable understanding into housing prices and evaluate how our forecasted housing prices can be explained by possible economy and market conditions as depicted by the patterns in the dataset.

Methodology

Describing the SARIMA Model

For our first model, we will be using the SARIMA (Seasonal Autoregressive Integrated Moving Average) model. SARIMA is an extension of the ARIMA model that additionally considers seasonal patterns in the data, so it deals with non-seasonal and seasonal patterns. In other words, SARIMA combines the idea of autoregressive (AR), integrated (I), and moving average (MA) models with seasonality.

To understand the model better, let's break down each component of SARIMA(p,d,q)$(P, D, Q)_s$:

(p,d,q) represents the non-seasonal part of the model. 'p' represents the non-seasonal autoregressive order, which is the number of lagged observations of the dependent variable, housing sales. 'd' represents the non-seasonal differencing order, which is the number of times the time series needs to be differenced to reach stationarity. 'q' represents the non-seasonal moving average order, which is the number of lagged forecast errors in the model.

(P,D,Q) represents the seasonal part of the model. 'P' indicates the seasonal autoregressive order, which is the number of lagged observations of the dependent variable, housing sales, at the seasonal lag. 'D' indicates the seasonal differencing order, which is the number of differencing operations required to convert the time series to be stationary. 'Q' indicates the seasonal moving average order, which is the number of lagged forecast errors at the seasonal lag.

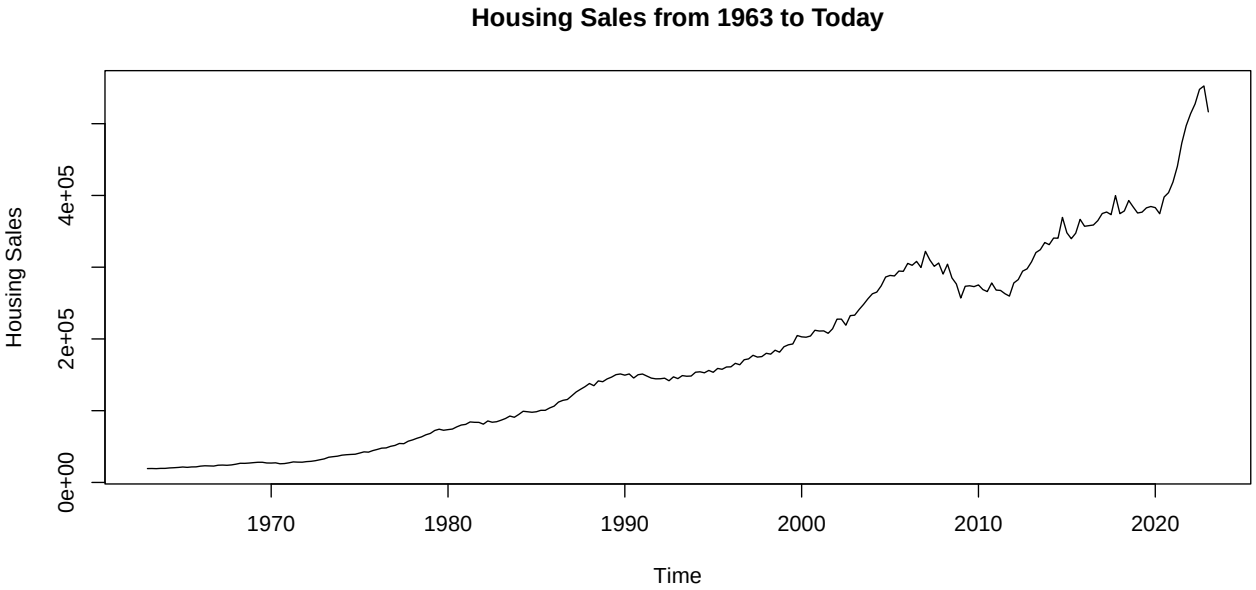
SARIMA is very useful for forecasting time series data with seasonal variations, which we will observe in the following report. Before we can forecast housing sales values using the SARIMA model, we will need to estimate parameters to find the best fit model by analyzing the autocorrelation and partial autocorrelation plots of our transformed stationary data and model selection criteria and performing diagnostic checks.

Plotting the Original Data: Housing Sales

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

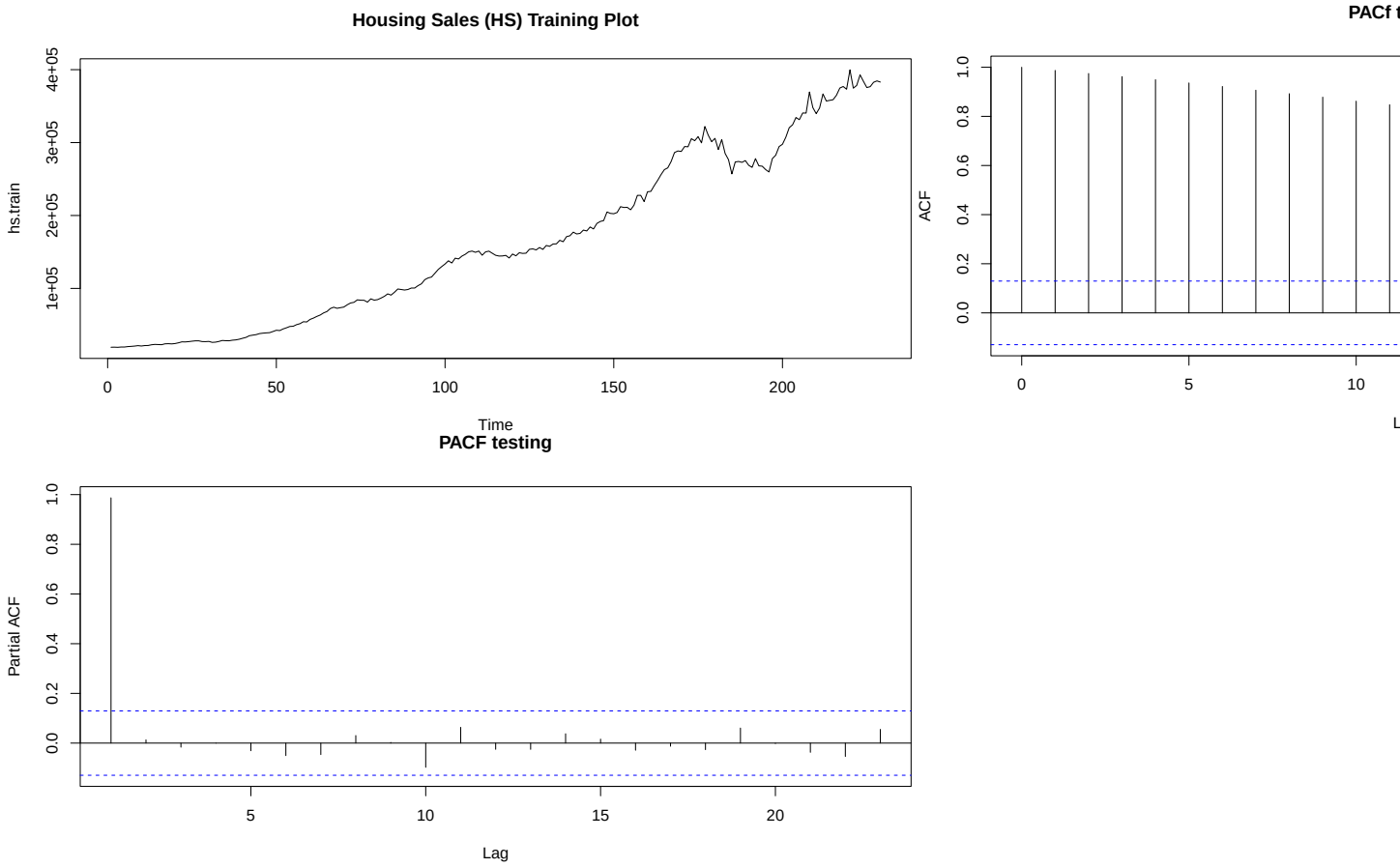
```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:astsa':
##
##     gas
```



As observed above in the plot of our housing sales time series data set, we see housing prices somewhat steadily increase over time, with the exception of significant changes in 2008 and 2020. The housing market crash of 2008 significantly decreased U.S. housing prices. In the years leading up to the crash, housing prices had increased sharply as a result of easy access to credit and the subprime mortgage crisis. Consequently, the housing market crashed in 2008. In 2020, we see a sharp spike in housing prices in 2020 that can be possibly explained by the economic and social changes caused by the Coronavirus-19.

Splitting into Training and Testing Sets

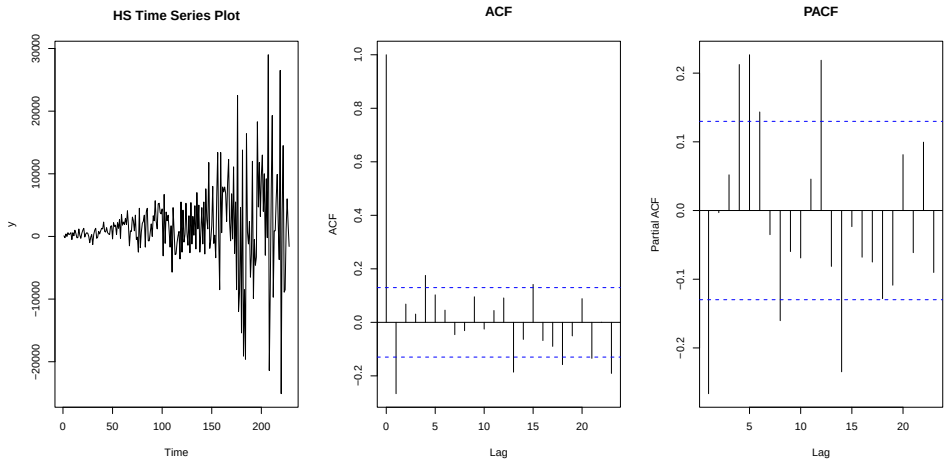


We will split the data set into a training set and testing set, with the training set including values from 1 to 229 and the testing set including values from 230 to 241. From the ACF and PACF plots of the training data set, there lies values outside of the confidence intervals. The ACF plot doesn't exhibit a significant lag and appears to be non-stationary. The PACF cuts off at lag 1 but since our ACF doesn't cut off at a significant lag, we can't identify a model and further transformations are needed.

Plotting our Transformed Data + ACF/PACF

Deseasonalizing

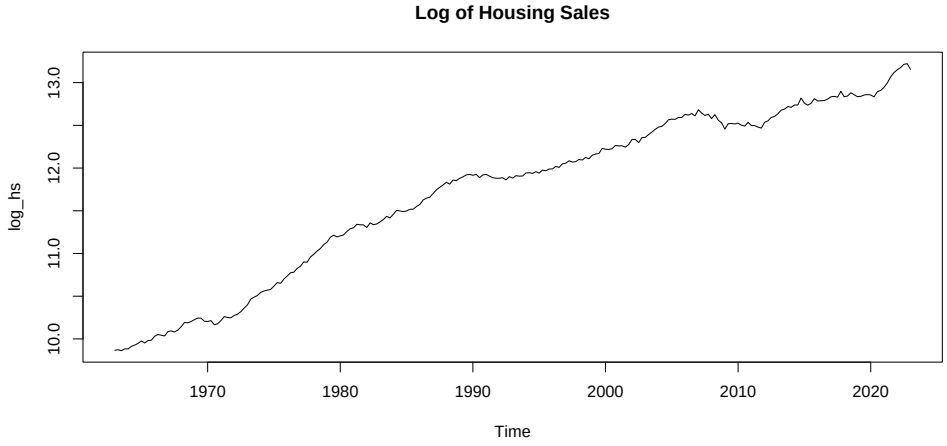
```
y <- diff(hs.train,lag=1) # 1 time period to lag when calculating the difference
par(mfrow=c(1, 3))
plot.ts(y, type="l", main="HS Time Series Plot")
acf(y, main="ACF"); pacf(y, main="PACF")
```



After performing deseasonalizing by taking the difference, we see the variance increase overtime. Thus, we will try a different method as we see deseasonalizing the time series data set doesn't transform it to become stationary.

Log Transformation

```
log_hs <- log(hs)
plot.ts(log_hs, main="Log of Housing Sales")
```



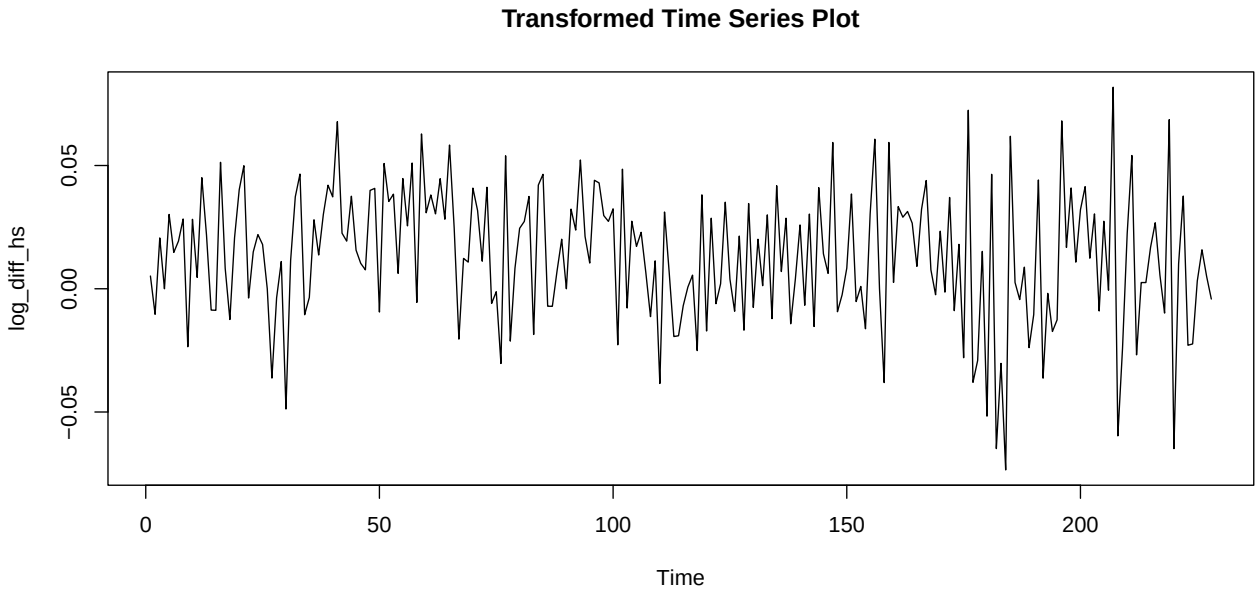

```
hist(log_hs, main="Histogram of Log of Housing Sales") # looks more symmetric
```

```
hist(hs.train, main="Histogram of the Training Set of Housing Sales")
```

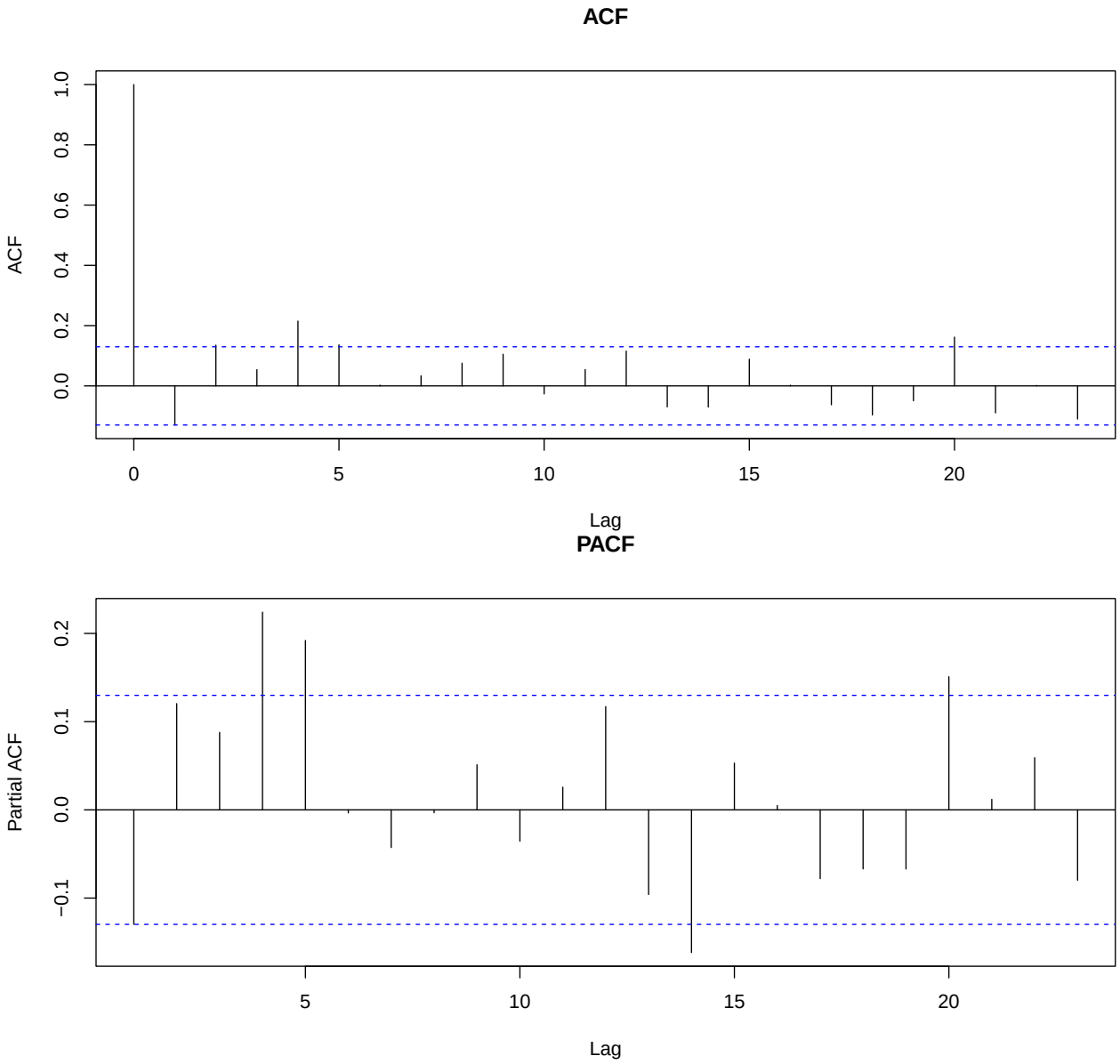
To stabilize the variance, we have taken the log of housing sales. Comparing the histograms of the training set of housing sales versus the log transformation of housing sales, we see that prior to the log transformation, housing sales displayed a right-skewed distribution. After taking the log transformation, housing sales appears to be more normal.

Difference of the Log (Final Transformation)

```
log_diff_hs <- diff(log(hs.train))
plot.ts(log_diff_hs, type="l", main="Transformed Time Series Plot")
```




```
acf(log_diff_hs, main="ACF"); pacf(log_diff_hs,main="PACF")
```



Above, we have plotted the transformed data and its autocorrelation and partial autocorrelation plots. Because there was a trend in taking the log transformation of housing sales, we have taken the difference of the log to make the housing sales data stationary. As seen in the plot, the housing sales data set no longer employs variance or trend and is stationary. Looking at the autocorrelation and partial autocorrelation plots, we see significant peaks from 0 to 5 (because our data is quarterly, the lag at 20 is 20/4=5).

Estimation Results

```
df <- data.frame(expand.grid(P=0:1, Q=0:5, p=0:2, q=0:2), AIC=NA, BIC=NA)
for (i in 1:nrow(df)) {
m <- df[i, ]
fit <- arima(log(hs.train), order=c(m$p, 1, m$q),
seasonal=list(order=c(m$P, 0, m$Q), period=4),
method="ML")
df[i, ]$AIC <- fit$aic; df[i, ]$BIC <- BIC(fit)
}
```

```
df[order(df$AIC)[1:3], ]; df[order(df$BIC)[1:3], ]
```

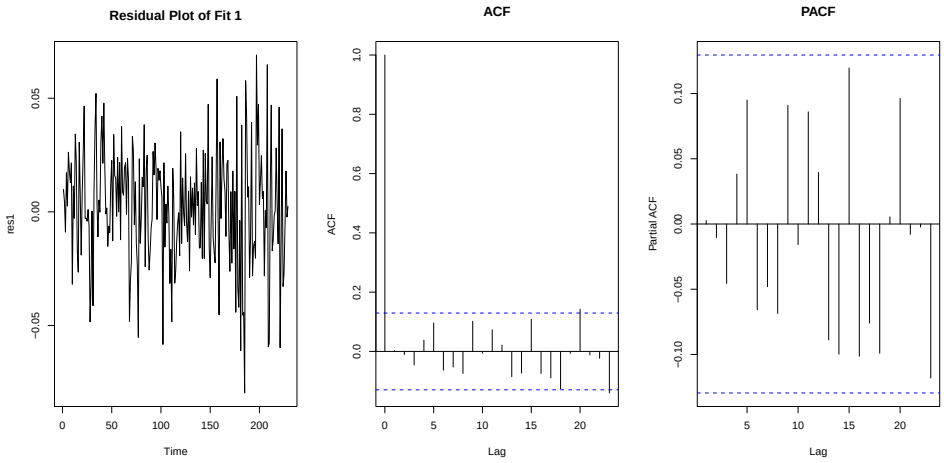
```
##    P Q p q       AIC       BIC
## 88 1 1 1 2 -1007.501 -986.9247
## 90 1 2 1 2 -1006.054 -982.0484
## 92 1 3 1 2 -1005.812 -978.3772
```

```
##    P Q p q       AIC       BIC
## 88 1 1 1 2 -1007.501 -986.9247
## 85 0 0 1 2 -1000.248 -986.5309
## 62 1 0 2 1 -1002.715 -985.5680
```

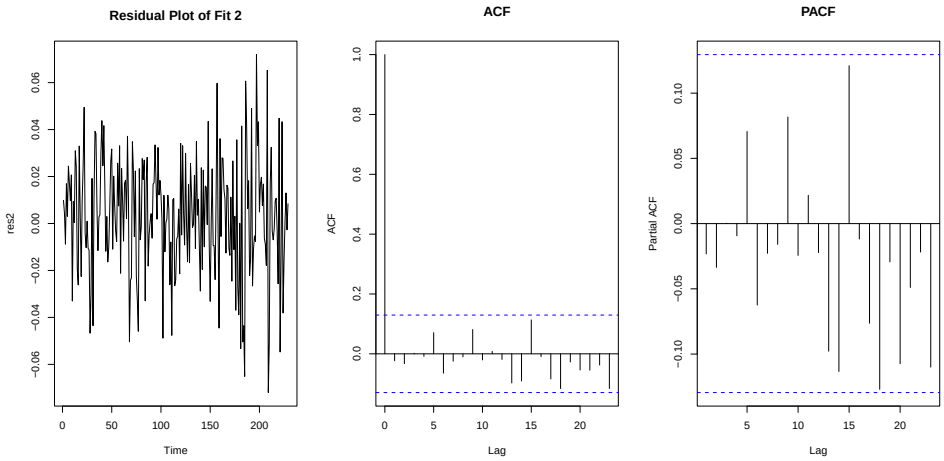
To estimate the parameters for a best fit model for our housing sales data, we will examine the model selection criteria values: AIC and BIC. As observed in the ACF and PACF plots from our difference log transformation, there are lags from 0 to 5 for P and Q. Note that if we make the parameters for P and Q both from 0 to 5, we will get an error because there are too many parameters. As a result, we will set P as 0 to 1 and Q as 0 to 5. Examining the AIC and BIC values, we see that when P=1, Q=1, p=1, and q=2, we get the lowest AIC and BIC values of -1007.501 and -986.9247, respectively. We will also play around with variations of the (p,d,q)(P,D,Q) parameters to find the best model for housing sales, and we will choose P=5, D=0, Q=1, p=2, d=1, and q=1 parameters as our second model for contention to be the best fit model. Comparing SARIMA $(1,1,2)(1,0,1)_4$ and $(2,1,1)(5,0,1)_4$, we will perform diagnostic checks to select the best model.

Diagnostic Checks

Residual Plots



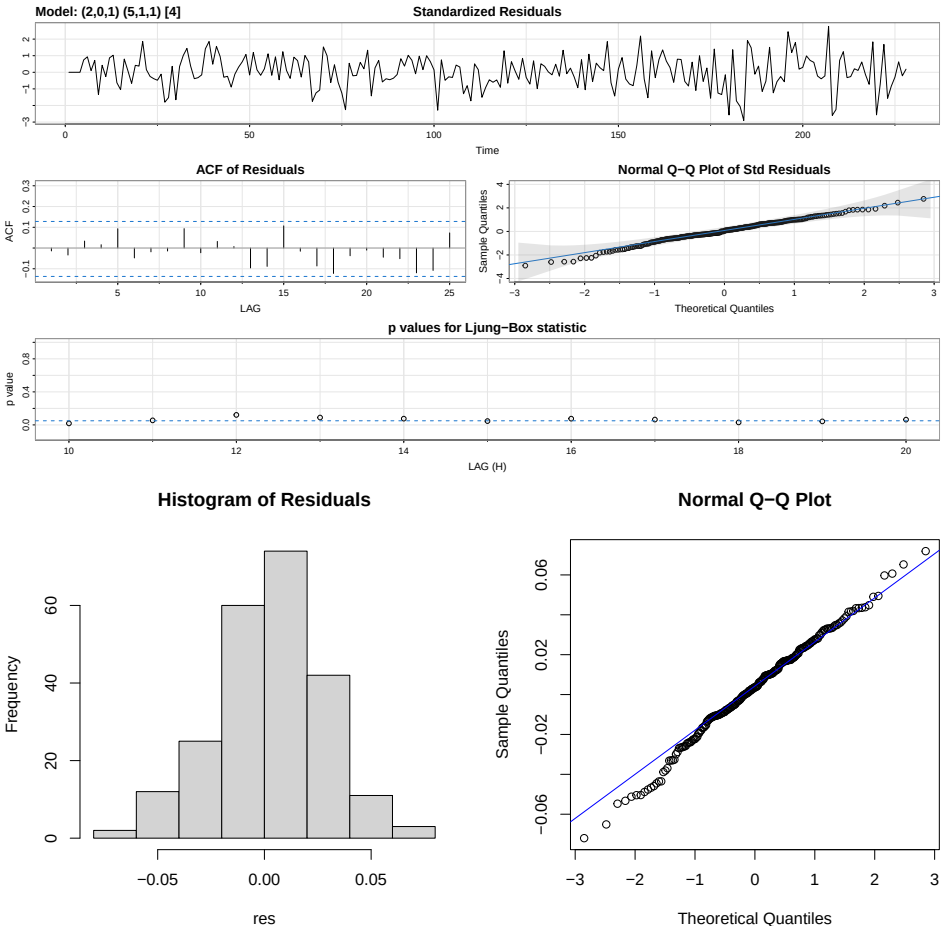
From the residual plot of our first potential best fit model, we see that there is a lag at 20 in the autocorrelation plot. This suggests that there still exists some correlation in the residuals that the SARIMA(1,0,2)$(1,1,1)_4$ model does not capture, violating the independence assumption of the residuals. The model may be missing lagged terms, causing it to fail to capture the long term dependencies. It could also be underfitting, meaning the model is too simple to capture the complexity of the data. Thus, this SARIMA model may not be appropriate for the housing sales time series data.



This second model –although it may be more complicated given the larger parameters– gives us complete white noise. We know this because in the ACF plot, when the lag is at 0, it has a corresponding value of 1 and 0 otherwise. Additionally, the PACF values are all equal to 0, so we know this second model provides us with white noise by definition. Hence, moving forward we will be using the SARIMA$(2,0,1)(5,1,1)_4$ model to run the diagnostic to confirm that this is our best fit model.

QQ plot

```
sarima(log_diff_hs,2,0,1,P=5,D=1,Q=1,S=4,details=TRUE)
```



Analyzing the histogram of the residuals, we see that the distribution is roughly normal. Our normal Q-Q Plot also shows that the observed points lie approximately on or near the reference line, so this suggests that our residuals appear to follow a normal distribution. The QQ plot is a strong indicator to assess the accuracy of our SARIMA model; thus, it appears that the SARIMA we chose in our model selection may be a good fit. We will run more diagnostic checks to further confirm that this model is a good fit.

Box-Pierce Test

```
Box.test(res)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 0.12369, df = 1, p-value = 0.7251
```

To perform a Box-Pierce Test, we will set the null hypothesis as the residuals are independent and the alternate hypothesis as the residuals are not independent. Because our p-value is $0.7251 > 0.05$, we will accept the null hypothesis and we can conclude that the residuals are independent. A higher p-value also indicates that the residuals have less autocorrelation, which is a good indicator to support our SARIMA model as a relatively good fit.

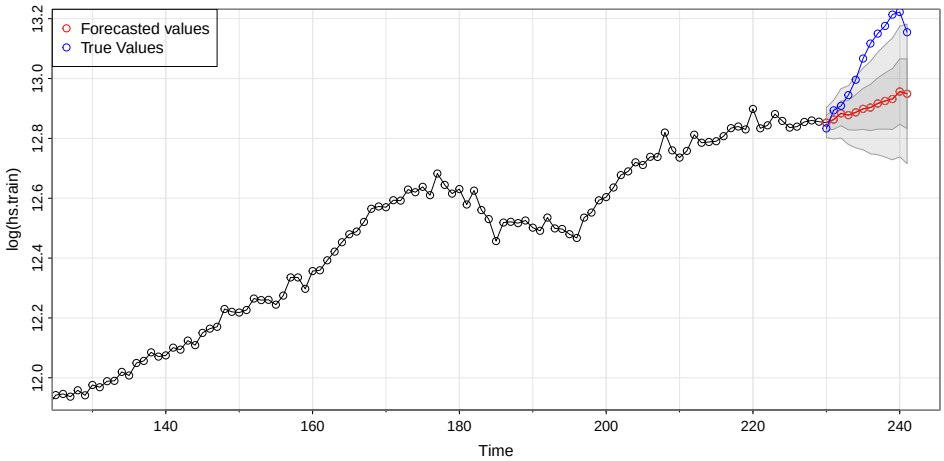
Shapiro Test

```
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99312, p-value = 0.369
```

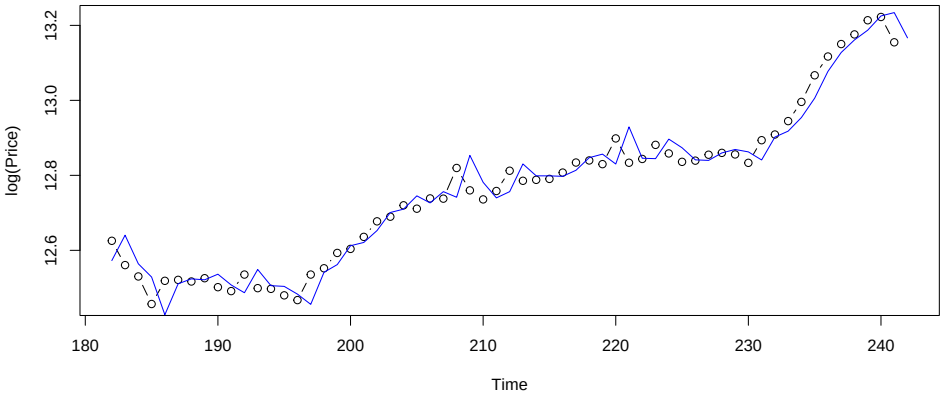
Looking at the Shapiro test, we will set the null hypothesis as the residuals being normally distributed. We get a large p-value of 0.369, meaning that we will accept the null hypothesis and conclude that the residuals are normal.

Forecasting



Although our forecasted values and true values differ, this doesn't necessarily mean that the SARIMA model we chose was a poor fit. The forecasted values we got from our code can't gauge economic and societal changes in the real world. So while our forecasted values predicted a gradual increase and we observed a large spike instead, this could be

due to the rise of the Coronavirus-19 that caused housing prices to signficantly increase. Thus, it is possible our forecasted and true values don't match up due to economic changes, as our SARIMA model can't take into account behavioral responses to Covid-19 or other possible economic upturns or downturns.



Unit Root Test

As applied above, the first difference operator is used to transform the non-stationary housing sales time series into a stationary housing sales time series by removing the trend or drift component. However, using the first difference operator may lead to overdifferencing, making the differenced series overly stationary. A unit root test allows us to test if the process is stationary through the absence of a unit root; hence, for our second model, we will be conducting the unit root test to see if the process used above is stationary. We will be performing two unit root tests: one on the log of the housing sales and the other on the difference of the log of the housing sales to show that the process for the difference of the log of the housing sales is stationary. To conduct the unit root test, we will test the null hypothesis that the process has a unit root against the alternate hypothesis that the process is stationary. We test the null hypothesis using three tests: DF, ADF, and PP tests.

The DF test, based on the Dickey-Fuller regression model, is a statistical test that detects the presence of a unit root in a time series by regressing the current value of the series on its lagged values. The test examines the significance of the coefficient of the lagged value to determine if it is significantly different from 1, which would indicate the presence of a unit root.

The ADF test is an extension of the DF test that further strengthens the accuracy of the unit root testing by including additional lagged terms in the regression model to account for potential serial correlation in the time series. So while the asymptotic null distribution is different between the DF and ADF test, the basic idea is the same. Hence, the choice of p is crucial to differentiate between the two.

The PP test is another test for unit roots that builds upon the ADF test by including a constant or non-stochastic trend. It addresses the potential issue of serial correlation in the residuals of the ADF regression model by applying a correction, and the PP test statistic is computed by accounting for autocorrelation and heteroscedasticity in the error residuals.

```
library(tseries)
adf.test(log(hs), k=0) # DF test
adf.test(log(hs)) # ADF test
pp.test(log(hs)) # PP test
```

The results of the DF, ADF, and PP tests indicate whether the time series (log(hs)) is stationary or exhibits a unit root. The DF test with a lag order of 0 has the p-value of 0.9149, and because the p-value is greater than the significance level, we will accept the null hypothesis and argue that there is evidence pointing towards the log(hs) time series being non-stationary. Similarly, the ADF test and PP test have a p-value of 0.7088 and 0.9504, respectively, so we can conclude that the log(hs) time series shows no evidence of stationarity as (indicated by the presence) there is a unit root.

```
adf.test(diff(log(hs)), k=0) # DF test
```

```
## Warning in adf.test(diff(log(hs)), k = 0): p-value smaller than printed p-value
```

```
adf.test(diff(log(hs))) # ADF test
```

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

```
pp.test(diff(log(hs))) # PP test
```

```
## Warning in pp.test(diff(log(hs))): p-value smaller than printed p-value
```

Now, we will perform the DF, ADF, and PP test for the time series diff(log(hs)). In each test, we observe a small p-value of 0.01, so we will reject the null hypothesis that the difference of the logged housing sales series has a unit root and accept the alternate hypothesis, concluding there's evidence of stationarity. This is consistent with our earlier findings that the process for the difference of the log of our housing sales time series is stationary whereas the process for the log of our housing sales data is not stationary. Thus, the unit root test supports our first model SARIMA(2,1,1)(5,0,1) as being our best fit model for predicting the future housing prices, as indicated by the absence of a unit root.

```
adf.test(res, k=0) # DF test
```

```
## Warning in adf.test(res, k = 0): p-value smaller than printed p-value
```

```
adf.test(res) # ADF test
```

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

```
pp.test(res) # PP test
```

```
## Warning in pp.test(res): p-value smaller than printed p-value
```

Conducting the DF, ADF, and PP test for the residuals of our best fit model in which the parameters were selected through observing the ACF and PACF graphs of the difference of the log of the housing sales data and concluding that the p-values are small, this further validates our finding that the difference of the log process is stationary and that it is our best fit model.

Conclusion

In this project, I examined the average U.S. housing sales price from January 1, 1963 to January 1, 2023. Uncovering trends and implementing time series methods, I was able to predict U.S. housing sales. I applied the Seasonal Autoregressive Integrated Moving Average SARIMA(p,d,q)$(P, D, Q)_s$ model and the Unit Root test to select the best model fit by first estimating the SARIMA model parameters through analyzing the ACF and PACF plots of the difference log transformation of our time series data and selecting the parameters with the lowest AIC and BIC values. Carrying out diagnostic checks, I confirmed that the model I selected was a good fit by showing that the models' residuals followed a normal distribution. By conducting DF, ADF, and PP tests, I demonstrated that my SARIMA model derived from the difference of log transformation was stationary as illustrated by the absence of a unit root.

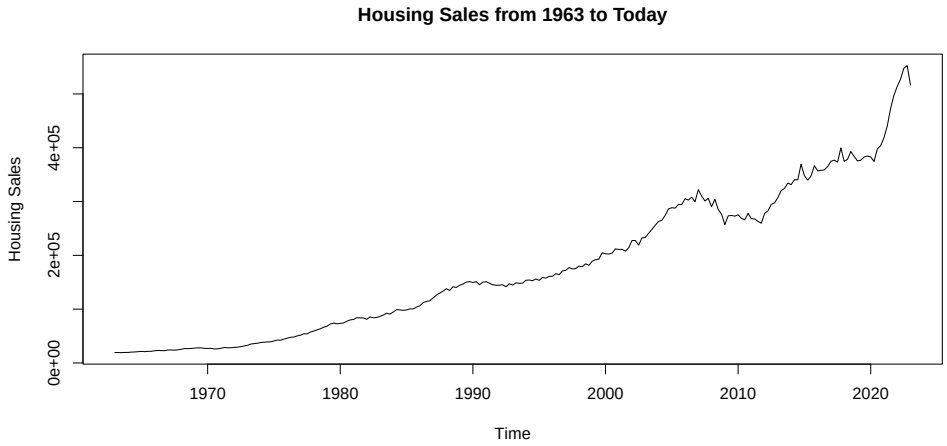
After forecasting U.S. housing sales, I found that while I was able to determine trends in the housing sales time series data and consider for them through transforming our data when selecting model parameters, the economy is unpredictable and this is something that our model cannot capture. Our model predicted an increase in housing prices over time, but it wasn't able to depict the sharp rise in housing prices in 2020 due to the Covid-19 virus. In future studies, I hope that with more advanced modeling techniques that another model can better exhibit the behavioral responses to housing sales caused by changes in the economy such as Covid-19 or the next large outbreak that will affect the supply and demand of housing.

References

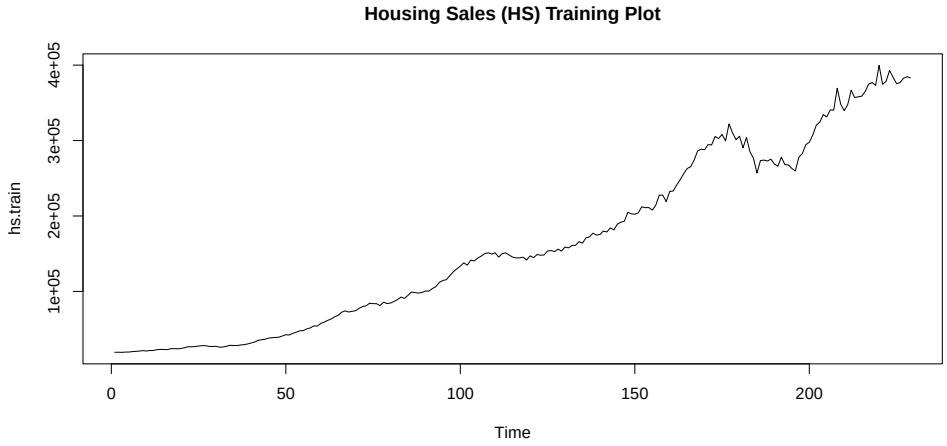
Shumway, Robert H., and Stoffer, David S.. "Time Series Analysis and Its Applications With R Examples". Fourth Edition. Springer International Publishing, 2017.

Appendix

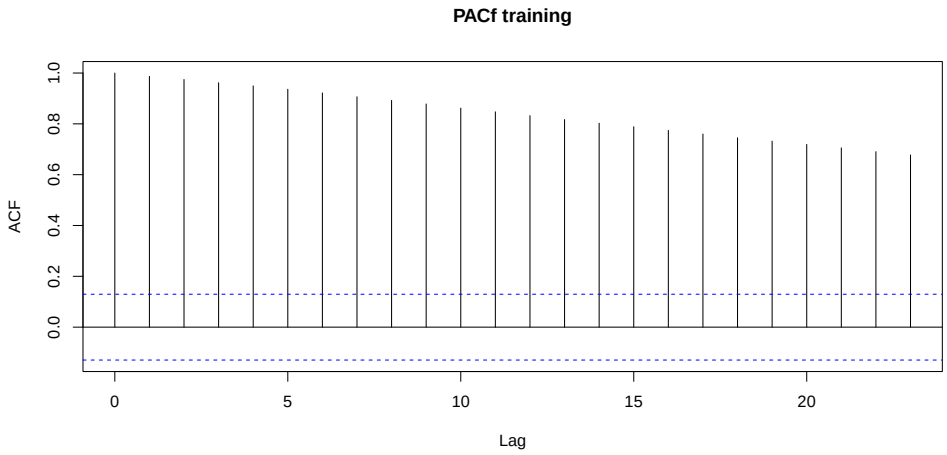
```
housing_sales <- read.csv("ASPUS.csv")
hs <- ts(housing_sales[,'ASPUS'],start=c(1963,1),frequency=4) # extracts a single column 'ASPUS' from t
plot.ts(hs,ylab="Housing Sales",main="Housing Sales from 1963 to Today")
```



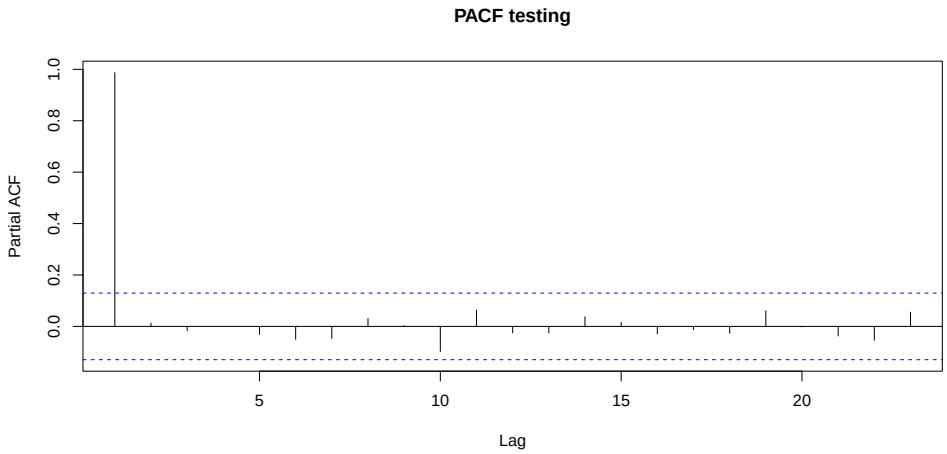

```
n <- length(hs)
hs.train <- hs[1:(n-12)]
hs.test <- hs[(n-11):n]
plot.ts(hs.train,main="Housing Sales (HS) Training Plot")
```




```
acf(hs.train,main='PACf training')
```



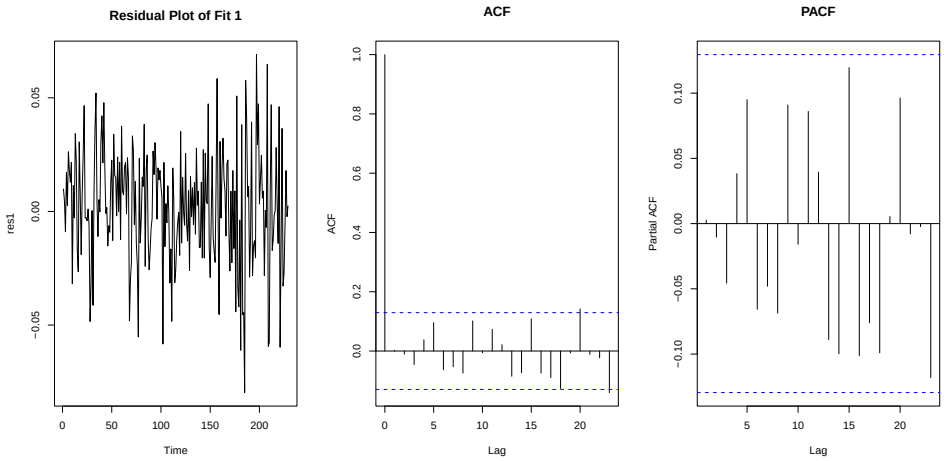

```
pacf(hs.train,main='PACF testing')
```



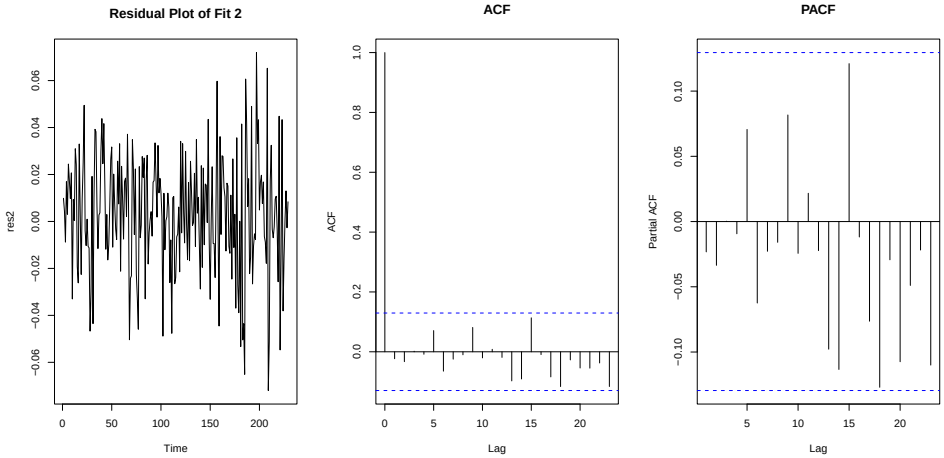

```
fit.x.1 <- arima(log(hs.train), order=c(1, 1, 2),
seasonal=list(order=c(1, 0, 1), period=4),
method="ML")
fit.x.1
```

```
fit.x.2 <- arima(log(hs.train), order=c(2, 1, 1),
seasonal=list(order=c(5, 0, 1), period=4),
method="ML")
fit.x.2
```

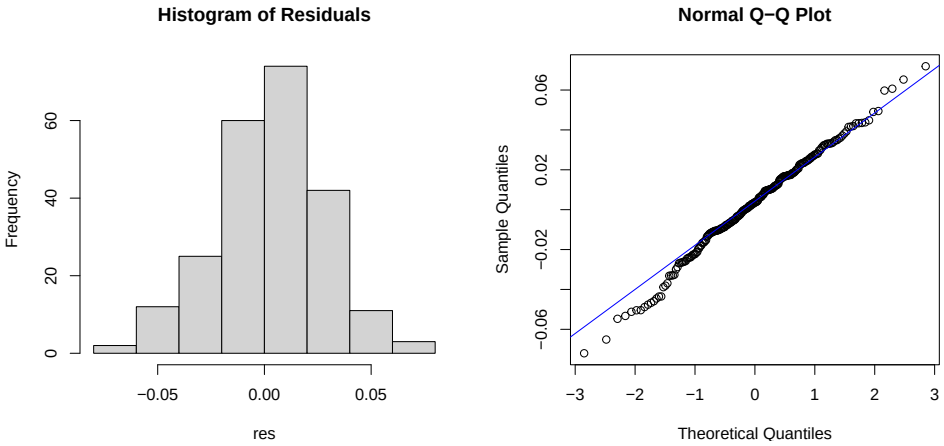
```
res1 <- residuals(fit.x.1)
par(mfrow=c(1, 3))
plot.ts(res1, type="l", main="Residual Plot of Fit 1")
acf(res1, main="ACF")
pacf(res1, main="PACF")
```



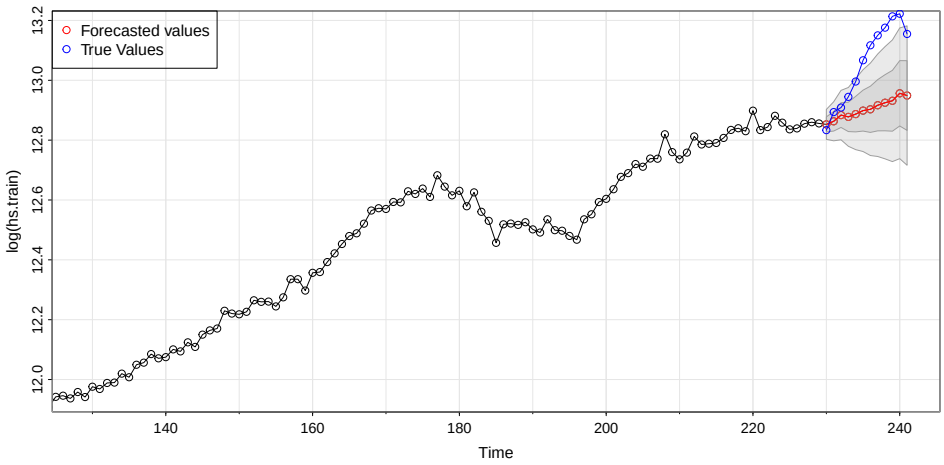

```
res2 <- residuals(fit.x.2)
par(mfrow=c(1, 3))
plot.ts(res2, type="l", main="Residual Plot of Fit 2")
acf(res2, main="ACF")
pacf(res2, main="PACF")
```




```
res <- residuals(fit.x.2)
par(mfrow=c(1, 2))
hist(res, main = "Histogram of Residuals")
qqnorm(res); qqline(res, col="blue")
```



```
n <- length(hs)
pred.tr <- sarima.for(log(hs.train), n.ahead=12, plot.all=F,
p=2, d=1, q=1, P=5, D=0, Q=1, S=4)
lines((n - 11):n, pred.tr$pred, col="red")
lines((n - 11):n, log(hs.test), col="blue")
points((n - 11):n, log(hs.test), col="blue")
legend("topleft", pch=1, col=c("red", "blue"),
legend=c("Forecasted values", "True Values"))
```




```
#n <- length(log_diff_hs)
hs_log_diff_fore=rep(NA,61)
for(i in 1:61){
  data=log_diff_hs[1:(i+181-1)]
  hs_log_diff_fore[i]=sarima.for(data,1, 0, 2,n.ahead=1, plot=F)$pred
}
hs_log_fore=hs_log_diff_fore+log(hs)[181:241]
plot(182:242, log(hs)[182:242], type="b", xlab="Time", ylab="log(Price)")
lines(182:242, hs_log_fore, col="blue")
```