Yvonne Lee: MSDS 664 Week 6

Assignment #6

To do a time series anaylsis on the data, I had cleaned up the data first. I removed the unnecessary columns, removed an empty row, fixed a column name, and removed commas.

```
medianSalesPrice <- read.csv("C:/Users/ylee_/Desktop/medianSalesPrice.csv")
```

```
drops <- c("X","X.1","X.2","X.3","X.4")
medianSalesPrice <- medianSalesPrice[, !(names(medianSalesPrice) %in% drops)]
```

```
head(medianSalesPrice)
```

	ï..Period <chr>	United <chr>	Northeast <chr>	Midwest <chr>	South <chr>	West <chr>
1						
2	1963Q1	17,800	20,800	17,500	16,800	18,000
3	1963Q2	18,000	20,600	17,700	15,800	18,900
4	1963Q3	17,900	19,600	17,800	15,900	19,000
5	1963Q4	18,500	20,600	19,100	15,800	19,500
6	1964Q1	18,500	20,300	18,700	16,500	19,600

6 rows

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## 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
```

```
medianSalesPrice <- medianSalesPrice %>% rename(Period = 'ï..Period')
medianSalesPrice <- medianSalesPrice[-1,]
```

```
head(medianSalesPrice)
```

	Period <chr>	United <chr>	Northeast <chr>	Midwest <chr>	South <chr>	West <chr>
2	1963Q1	17,800	20,800	17,500	16,800	18,000
3	1963Q2	18,000	20,600	17,700	15,800	18,900
4	1963Q3	17,900	19,600	17,800	15,900	19,000
5	1963Q4	18,500	20,600	19,100	15,800	19,500
6	1964Q1	18,500	20,300	18,700	16,500	19,600
7	1964Q2	18,900	19,800	19,800	16,800	20,100

6 rows

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.0.4
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
x <- medianSalesPrice$Period
medianSalesPrice$Period <- as.yearqtr(sub("(..)", "\\1 ", x), format = "Q%q %Y")
```

```
medianSalesPrice$United <- as.numeric(gsub(",","",medianSalesPrice$United))
medianSalesPrice$Northeast <- as.numeric(gsub(",","",medianSalesPrice$Northeast))
medianSalesPrice$Midwest <- as.numeric(gsub(",","",medianSalesPrice$Midwest))
medianSalesPrice$South <- as.numeric(gsub(",","",medianSalesPrice$South))
medianSalesPrice$West <- as.numeric(gsub(",","",medianSalesPrice$West))
```

```
head(medianSalesPrice)
```

	Period <yearqtr>	United <dbl>	Northeast <dbl>	Midwest <dbl>	South <dbl>	West <dbl>
2	<NA>	17800	20800	17500	16800	18000
3	<NA>	18000	20600	17700	15800	18900
4	<NA>	17900	19600	17800	15900	19000
5	<NA>	18500	20600	19100	15800	19500
6	<NA>	18500	20300	18700	16500	19600
7	<NA>	18900	19800	19800	16800	20100

6 rows

I then created a time series object with the Northeast column and plotted it.

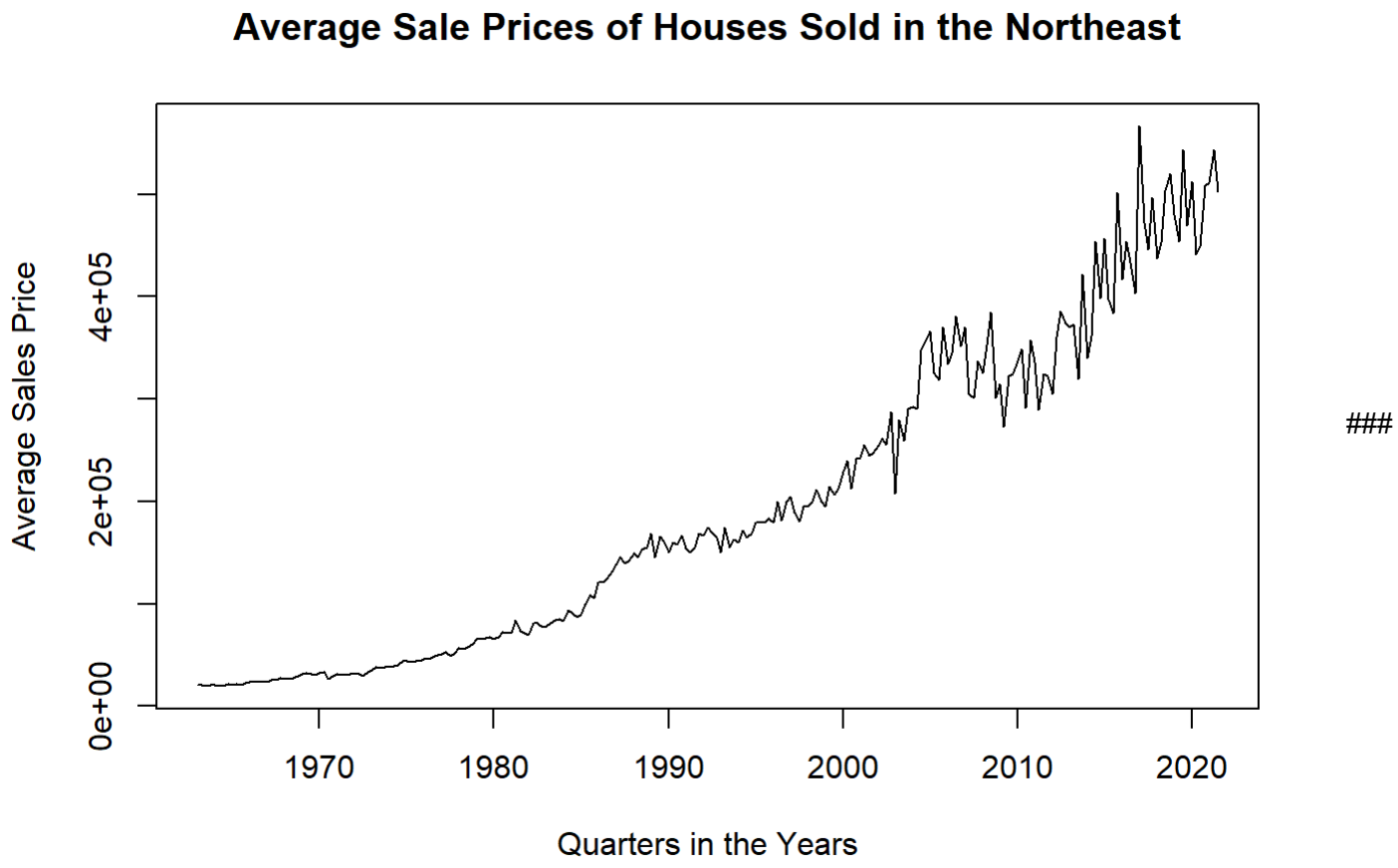
```
medianSalesPrice.ts <- ts(medianSalesPrice$Northeast, frequency = 4, start = 1963)
```

```
medianSalesPrice.ts
```

```
##        Qtr1   Qtr2   Qtr3   Qtr4
## 1963  20800  20600  19600  20600
## 1964  20300  19800  20200  21400
## 1965  21000  21900  21200  23100
## 1966  23700  23700  23700  23400
## 1967  23800  25400  25600  27500
## 1968  26600  26900  27800  29900
## 1969  31500  32100  31300  30500
## 1970  32000  33100  26800  29200
## 1971  31000  30500  30900  31300
## 1972  31900  31100  29400  33800
## 1973  35100  38400  36900  38600
## 1974  38600  39500  40700  44000
## 1975  44000  43400  44400  44400
## 1976  45700  46300  48400  50200
## 1977  50900  53200  49300  52000
## 1978  56800  55400  57800  60700
## 1979  65500  65800  66500  67100
## 1980  65400  67900  71900  71400
## 1981  71300  84000  73700  71100
## 1982  69900  81600  81300  76800
## 1983  77800  81000  84100  84900
## 1984  82700  94200  91200  86600
## 1985  89800  99800 108800 105700
## 1986 121500 121000 125000 131500
## 1987 139000 145000 139900 142000
## 1988 149000 145000 153000 155000
## 1989 168500 144900 165700 161400
## 1990 150000 159900 158000 167000
## 1991 153900 150000 155200 169000
## 1992 166900 175000 170000 165000
## 1993 150000 175000 155000 162600
## 1994 159900 172000 165000 169000
## 1995 179900 179900 179900 183500
## 1996 179000 199700 181000 200000
## 1997 204400 189000 180000 195000
## 1998 196000 200000 212000 200000
## 1999 195500 214700 206400 212600
## 2000 229300 239500 212800 241400
## 2001 242800 255200 244200 247800
## 2002 254200 261100 255400 287100
## 2003 208100 279900 259400 290000
## 2004 292000 290300 347700 357400
## 2005 366800 325700 318700 370300
## 2006 334600 344600 380500 351400
## 2007 370300 304900 301300 336900
## 2008 325900 352500 385200 300700
## 2009 314800 272500 322200 324600
## 2010 337400 348700 291000 358000
## 2011 336200 289100 324100 322800
## 2012 305400 360900 385700 374300
## 2013 370300 373200 320100 421400
```

```
## 2014 339800 361900 453900 398600
## 2015 456800 397800 383700 501900
## 2016 417100 454100 434000 403700
## 2017 566500 472200 445800 496500
## 2018 437500 453300 503700 519700
## 2019 480300 453500 543400 469500
## 2020 512100 441000 449500 508100
## 2021 511700 543800 502300
```

```
plot(medianSalesPrice.ts, xlab="Quarters in the Years", ylab="Average Sales Price", main="Averag
e Sale Prices of Houses Sold in the Northeast")
```



###

The plot shows an increase in average price as time goes on.

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.5
```

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

```
adf.test(medianSalesPrice.ts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  medianSalesPrice.ts
## Dickey-Fuller = -1.5766, Lag order = 6, p-value = 0.7539
## alternative hypothesis: stationary
```

With a hypothesis that states the time series is stationary and a null stating that is it not stationary, the adf.test() shows a p-value that is above 0.05. This means we fail to reject the null hypothesis and that the data may not have a time-dependent structure.

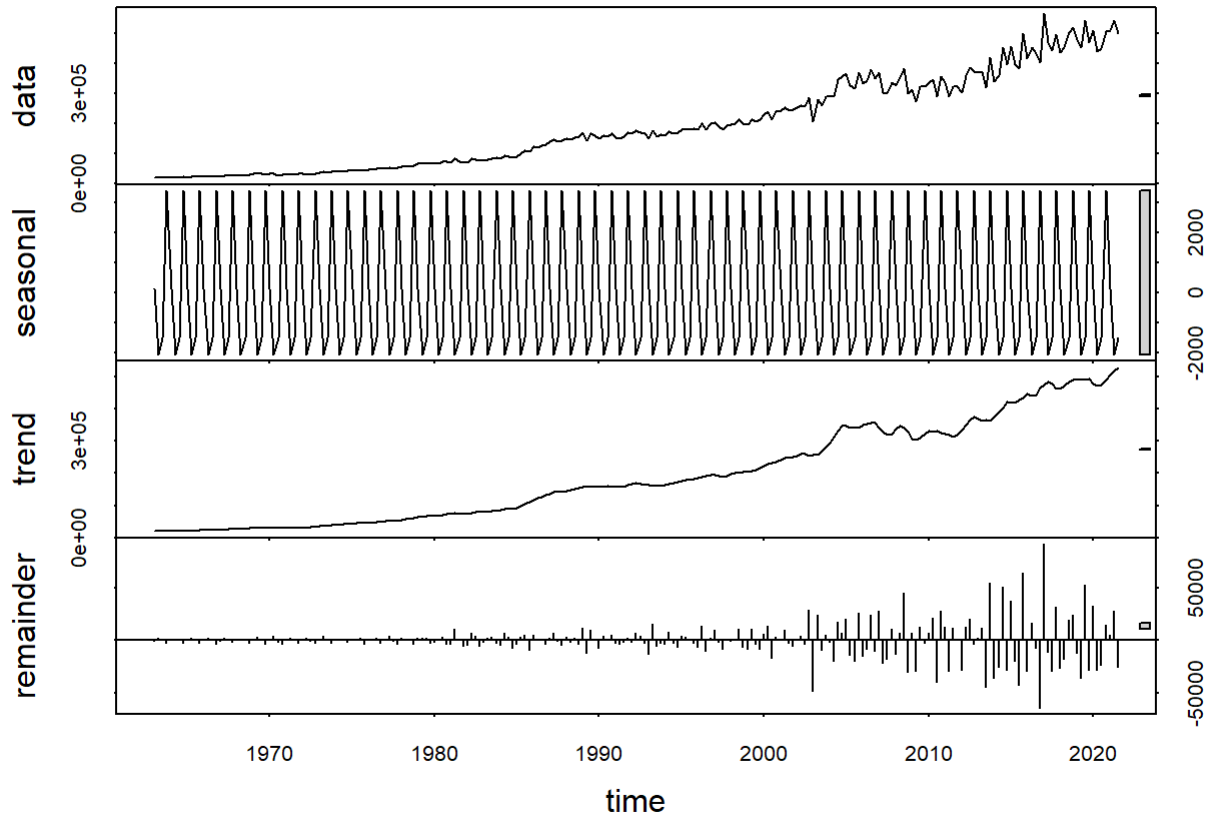
```
Box.test(medianSalesPrice.ts, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  medianSalesPrice.ts
## X-squared = 223.37, df = 1, p-value < 2.2e-16
```

The box.text() supports rejecting the null hypothesis.

```
median.stl <- stl(medianSalesPrice.ts, s.window = "periodic")
```

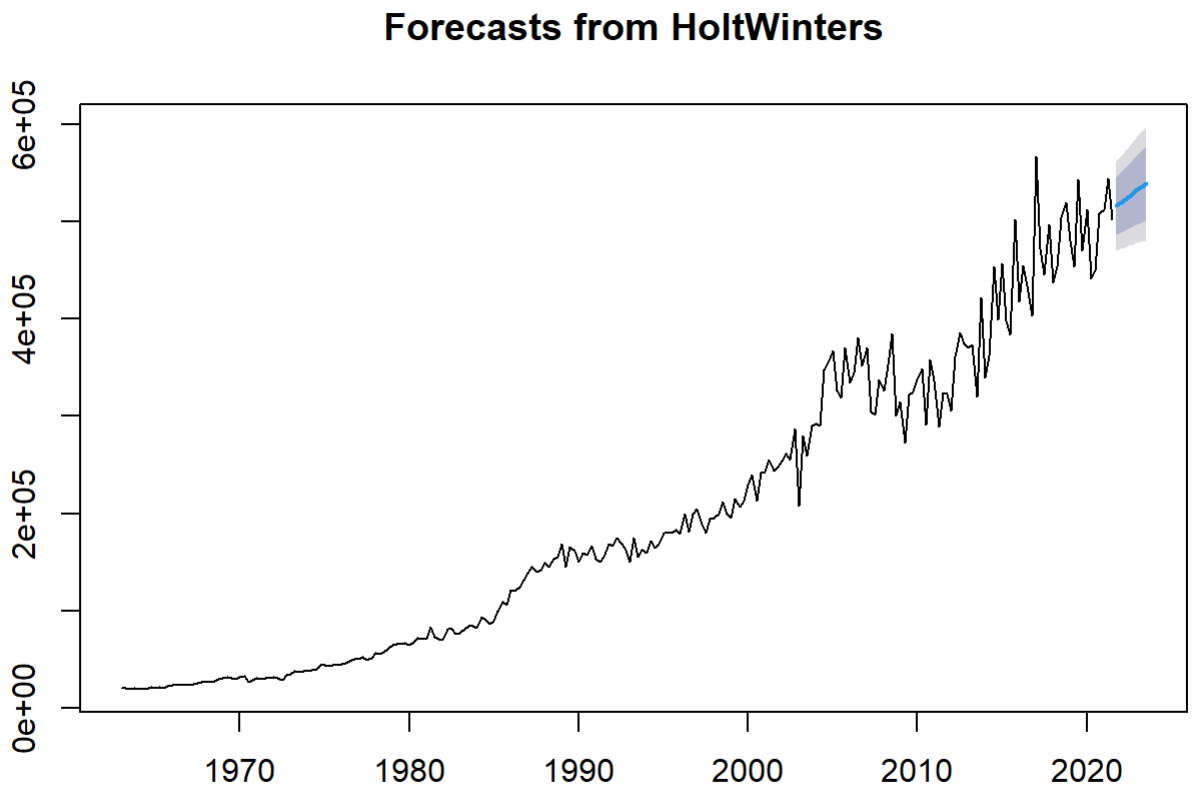
```
plot(median.stl)
```




```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.5
```

```
median.holt <- HoltWinters(medianSalesPrice.ts,gamma=FALSE)
plot(forecast(median.holt))
```




```
Box.test(medianSalesPrice.ts, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  medianSalesPrice.ts
## X-squared = 223.37, df = 1, p-value < 2.2e-16
```

The forecasting shows that prices will only continue to increase in the future.

Between 2017 and 2018 seems to be the peak of home prices in the Northeast. Although it was still a gradual increase in prices, there was a dip in prices between 2003 and 2004 and a small dip around 2008.

References:

Coghlan, A. (n.d.). Using R for time series analysis¶. Using R for Time Series Analysis - Time Series 0.2 documentation. Retrieved November 30, 2021, from https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html (https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html).

Ralph. (2013, January 11). Seasonal trend decomposition in R: R-bloggers. R. Retrieved November 30, 2021, from https://www.r-bloggers.com/2013/01/seasonal-trend-decomposition-in-r/ (https://www.r-bloggers.com/2013/01/seasonal-trend-decomposition-in-r/).

Zach. (2021, May 25). Augmented dickey-fuller test in R (with example). Statology. Retrieved November 30, 2021, from https://www.statology.org/dickey-fuller-test-in-r/ (https://www.statology.org/dickey-fuller-test-in-r/).