Krysten Thompson w271: Homework 7

Professor Jeffrey Yau

In this homework, you are asked to use `quantmod` to get a time sereis `HOUST` from the Federal website, conduct Time Series EDA, examine seasonality, develop a model that can capture both trend and seasonality in the series, and plot the observed vs fitted value, and use the model to make a 12-step ahead forecast.

```
rm(list = ls())

# Load required libraries
library(car)
library(dplyr)
library(astsa)
library(forecast)
library(fpp2)
library(ggplot2)
library(plotly)
library(quantmod)

# Insert the function to *tidy up* the code when they are printed out
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

```
# Use Quantmod to load data
HOUST = getSymbols('HOUST', src='FRED', auto.assign = F)

#str(HOUST)   I commented these out to save space
#head(HOUST)
#tail(HOUST)

house.starts = ts(HOUST, frequency = 12, start = c(1959,1))
```

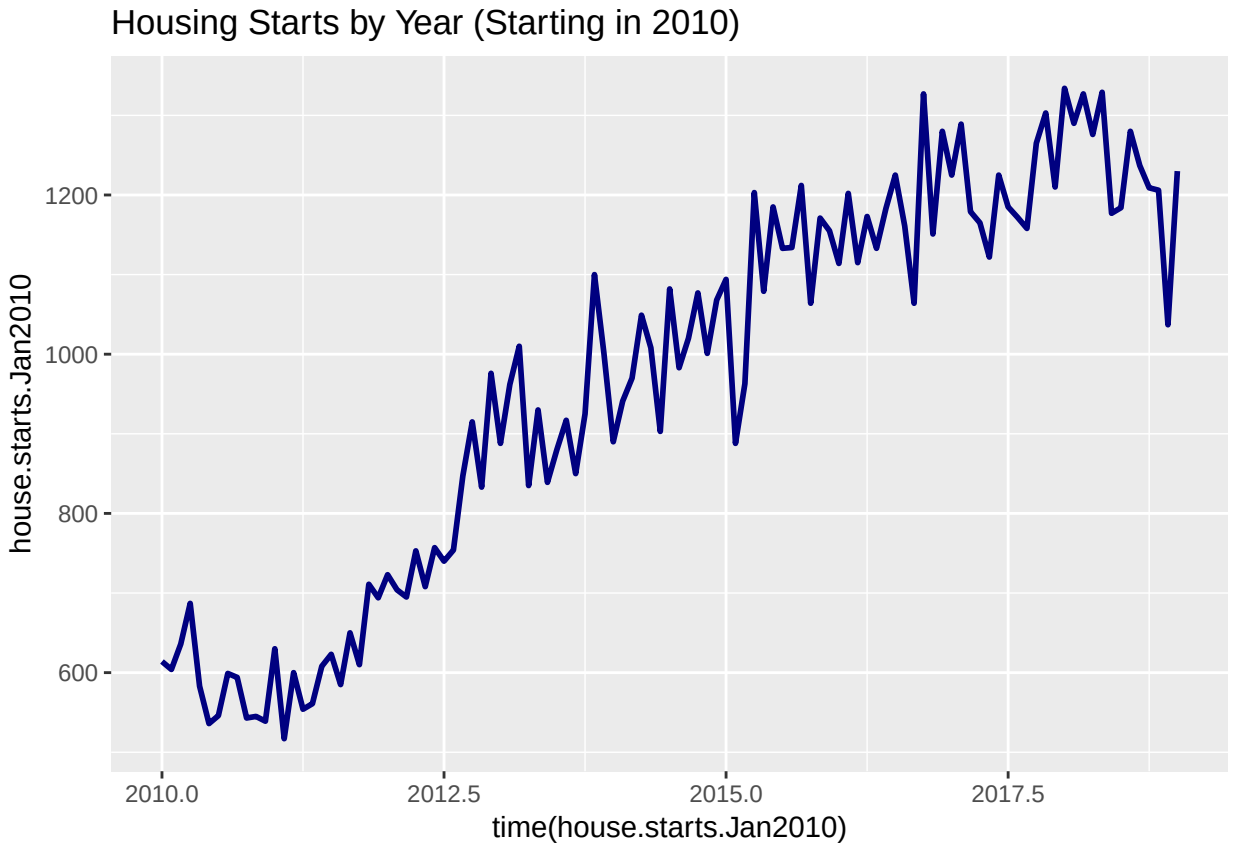
```
# subset your time series to one starting in Januagy 2010

house.starts.Jan2010 <- window(house.starts, start=c(2010, 1))
#house.starts.Jan2010
```

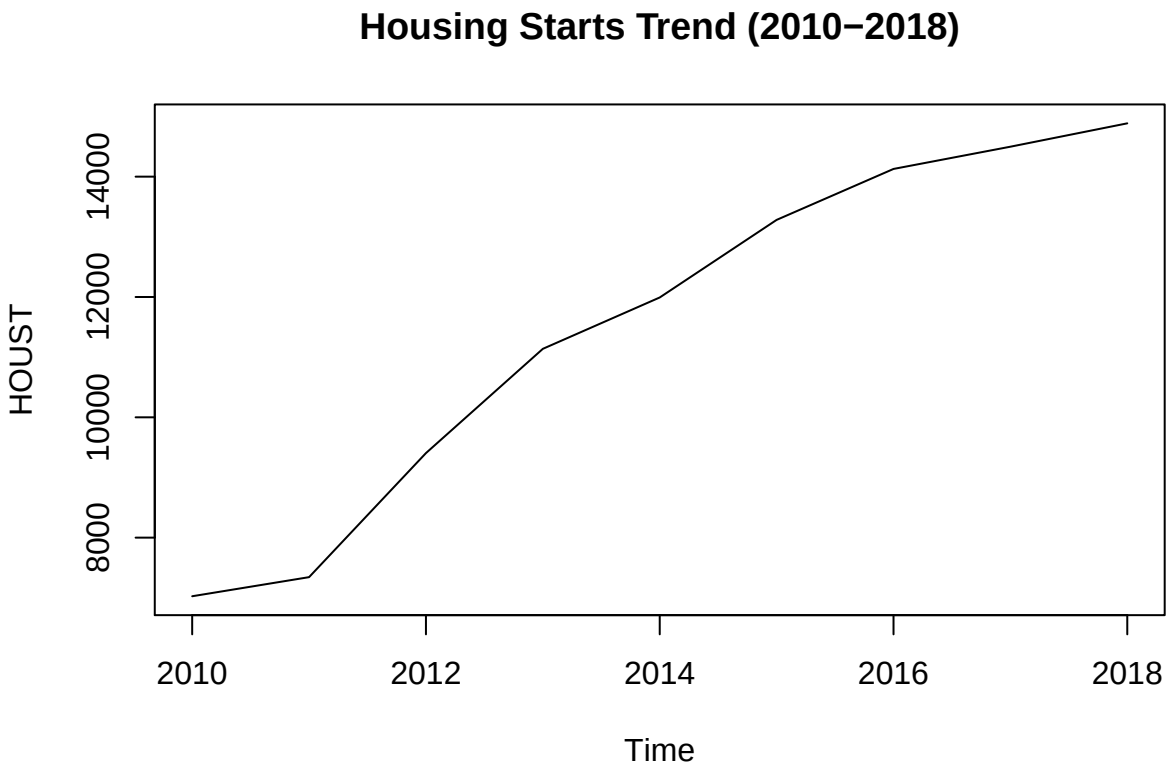
Examine your data

```
ggplot(house.starts.Jan2010, aes(x=time(house.starts.Jan2010), y=house.starts.Jan2010)) +
  geom_line(colour = "navy", size = 1) +
  ggtitle("Housing Starts by Year (Starting in 2010)") +
  theme(axis.title = element_text(size = rel(1.0)))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

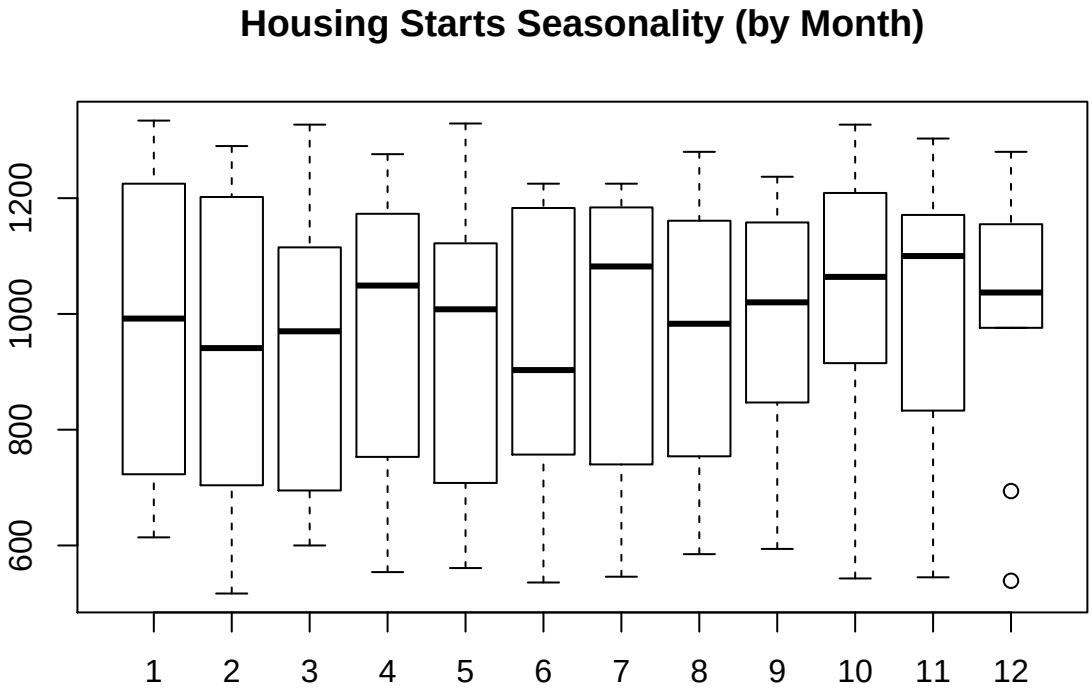



```
plot(aggregate(house.starts.Jan2010), main = "Housing Starts Trend (2010-2018)")
```

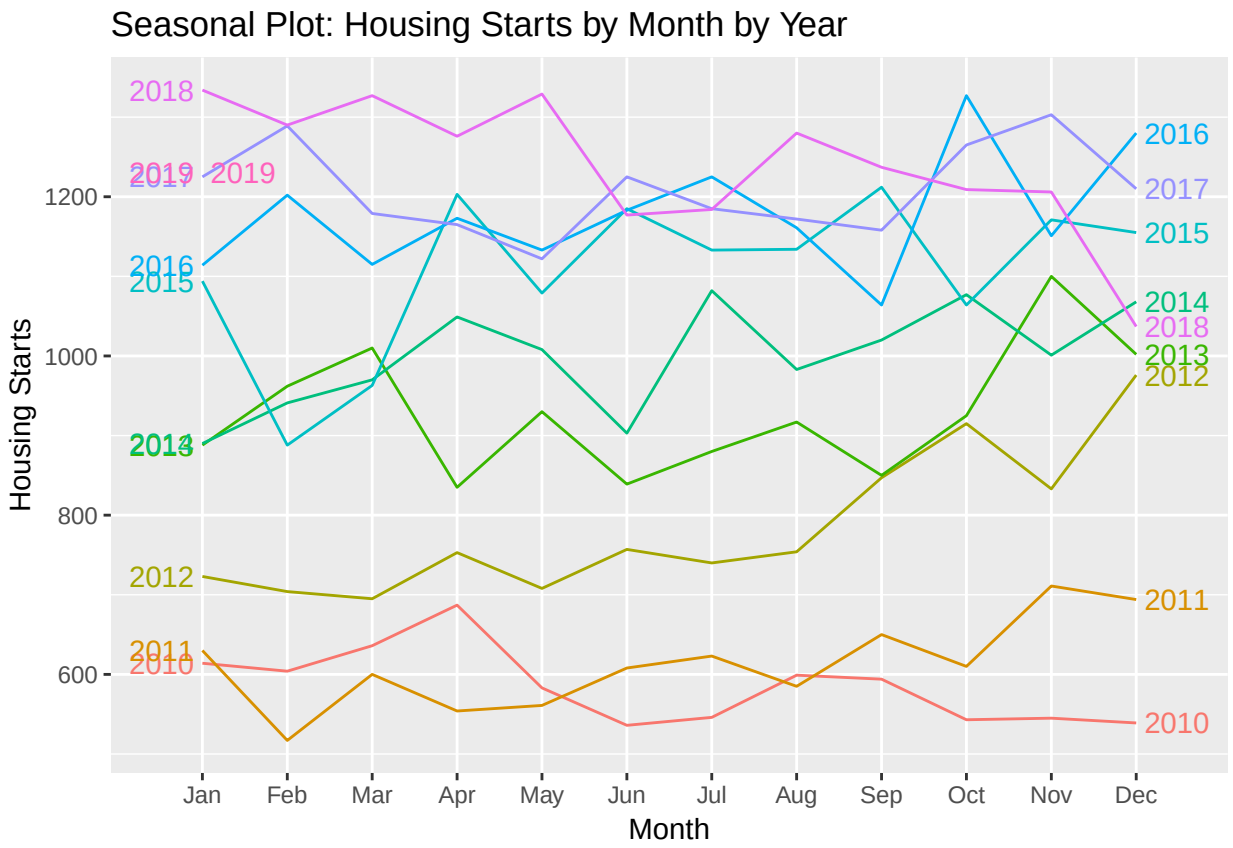


Examine seasonality

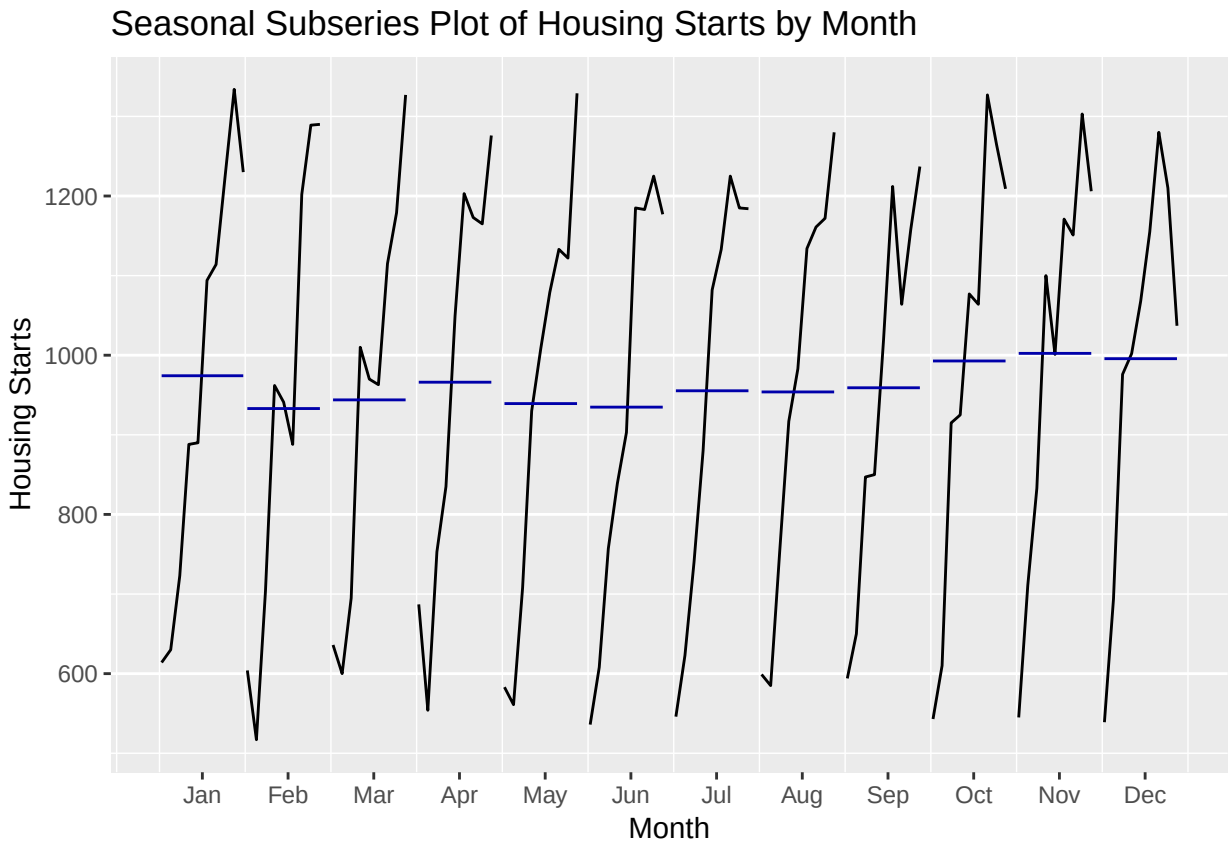
```
boxplot(house.starts.Jan2010 ~ cycle(house.starts.Jan2010),
        main= "Housing Starts Seasonality (by Month)")
```



```
ggseasonplot(house.starts.Jan2010, year.labels = TRUE, year.labels.left = TRUE) +
  ylab("Housing Starts") + ggtitle("Seasonal Plot: Housing Starts by Month by Year")
```




```
ggsubseriesplot(house.starts.Jan2010) + ylab("Housing Starts") +
  ggtitle("Seasonal Subseries Plot of Housing Starts by Month")
```



Estimate a model with trend, seasonlity, or both

```
mod.trend <- lm(house.starts.Jan2010 ~ time(house.starts.Jan2010))
summary(mod.trend)
```

```
##
## Call:
## lm(formula = house.starts.Jan2010 ~ time(house.starts.Jan2010))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -316.32  -61.23    4.37   58.05  196.34
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -1.772e+05  6.361e+03  -27.86   <2e-16 ***
## time(house.starts.Jan2010)   8.846e+01  3.158e+00   28.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86.44 on 107 degrees of freedom
## Multiple R-squared:   0.88,  Adjusted R-squared:  0.8789
```

```
## F-statistic: 784.8 on 1 and 107 DF,  p-value: < 2.2e-16
```

I was curious what model looked like when I called "tslm".

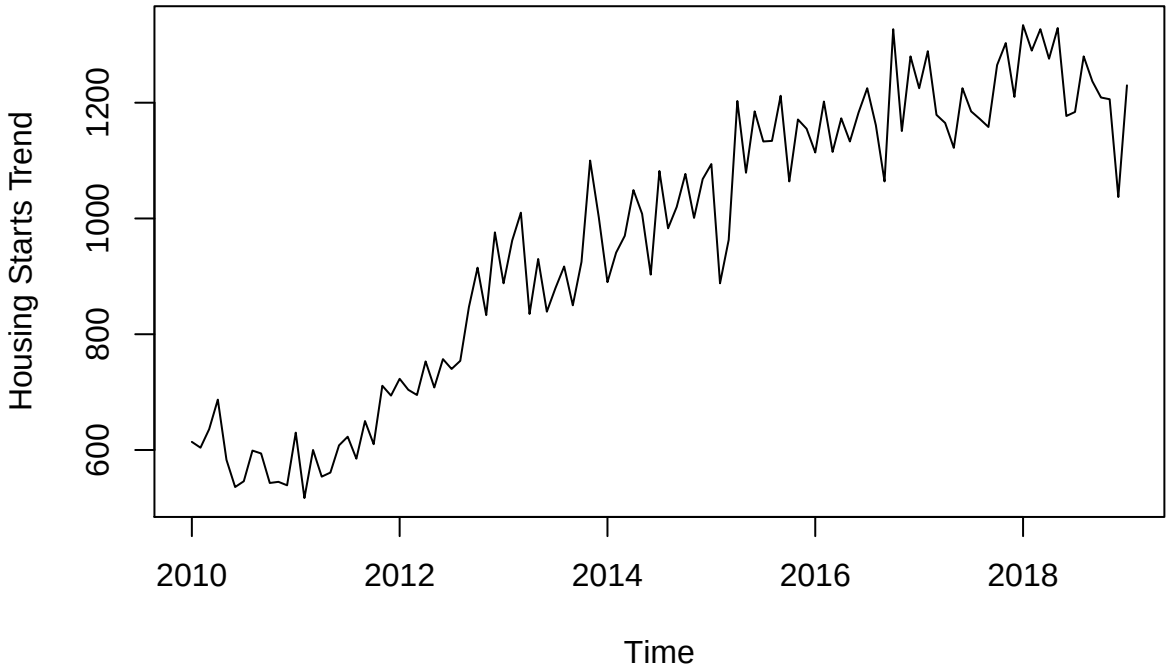
```
fit.tslm <- tslm(house.starts.Jan2010 ~ trend + season)
summary(fit.tslm)
```

```
##
## Call:
## tslm(formula = house.starts.Jan2010 ~ trend + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -312.999  -51.834   -0.611   65.055  186.250
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 568.1938    32.2604  17.613   <2e-16 ***
## trend         7.3819     0.2754  26.800   <2e-16 ***
## season2      -4.2903    41.4062  -0.104    0.918
## season3      -0.7834    41.3980  -0.019    0.985
## season4      14.0569    41.3916   0.340    0.735
## season5     -20.2139    41.3870  -0.488    0.626
## season6     -32.0403    41.3842  -0.774    0.441
## season7     -18.8667    41.3833  -0.456    0.649
## season8     -27.6930    41.3842  -0.669    0.505
## season9     -29.8527    41.3870  -0.721    0.472
## season10     -3.5680    41.3916  -0.086    0.931
## season11     -1.3944    41.3980  -0.034    0.973
## season12    -15.4430    41.4062  -0.373    0.710
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 90.07 on 96 degrees of freedom
## Multiple R-squared:  0.8831, Adjusted R-squared:  0.8685
## F-statistic: 60.45 on 12 and 96 DF,  p-value: < 2.2e-16
```

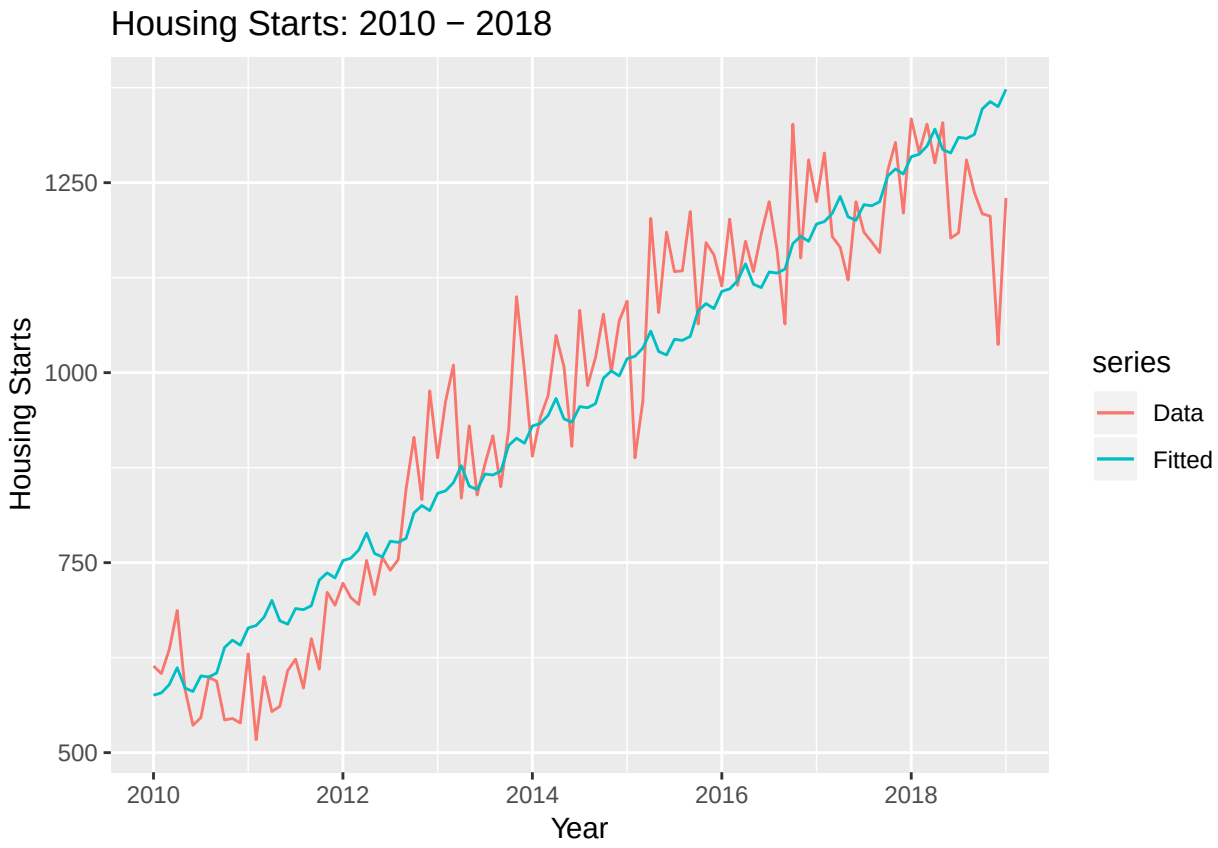
Plot the observed and fitted values

I tried several iterations here.

```
plot(house.starts.Jan2010, ylab = "Housing Starts Trend")
lines(mod.trend$fitted.values, col= 'red')
```




```
autoplot(house.starts.Jan2010, series="Data") +
  autolayer(fitted(fit.tslm), series="Fitted") +
  xlab("Year") + ylab("Housing Starts") +
  ggtitle("Housing Starts: 2010 - 2018")
```



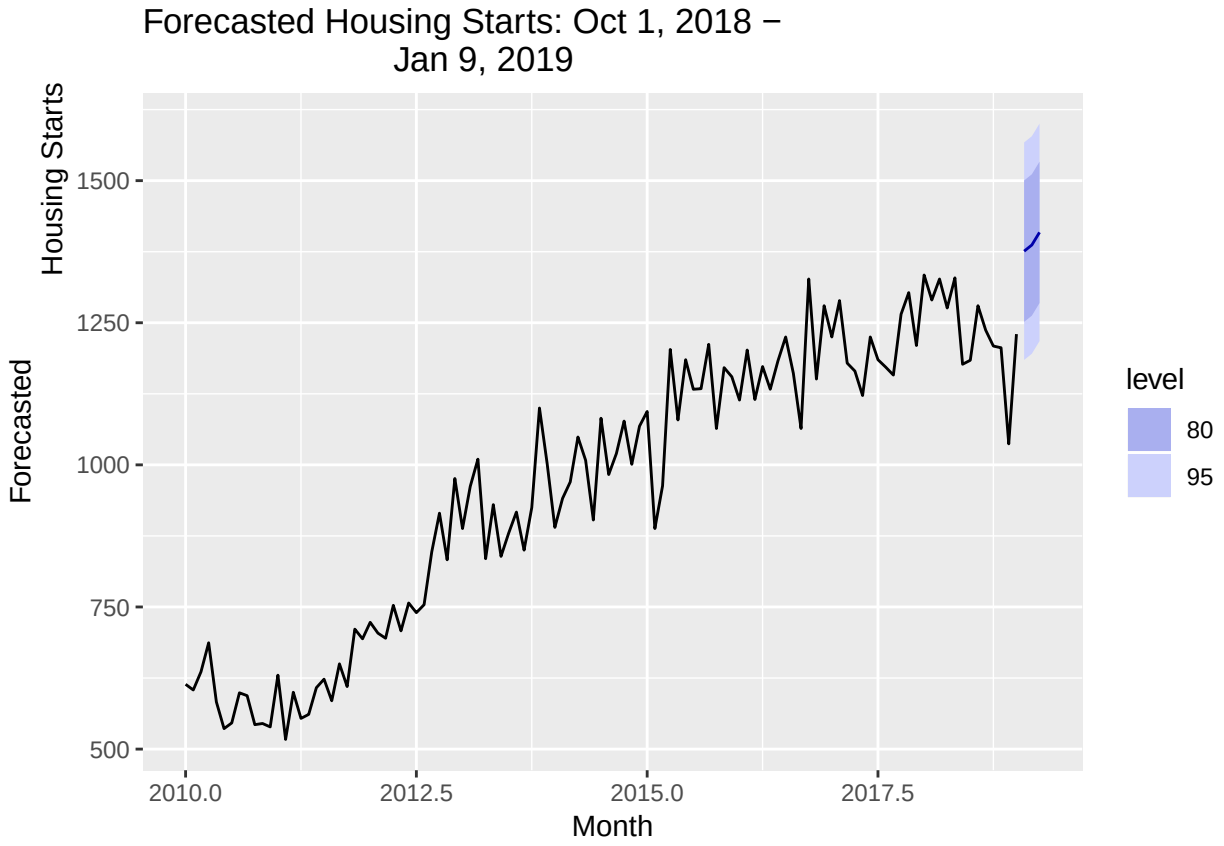
Make a 12-step ahead (out-of-sample) forecast

That is, forecast from 2018-10-01 to 2019-09-01

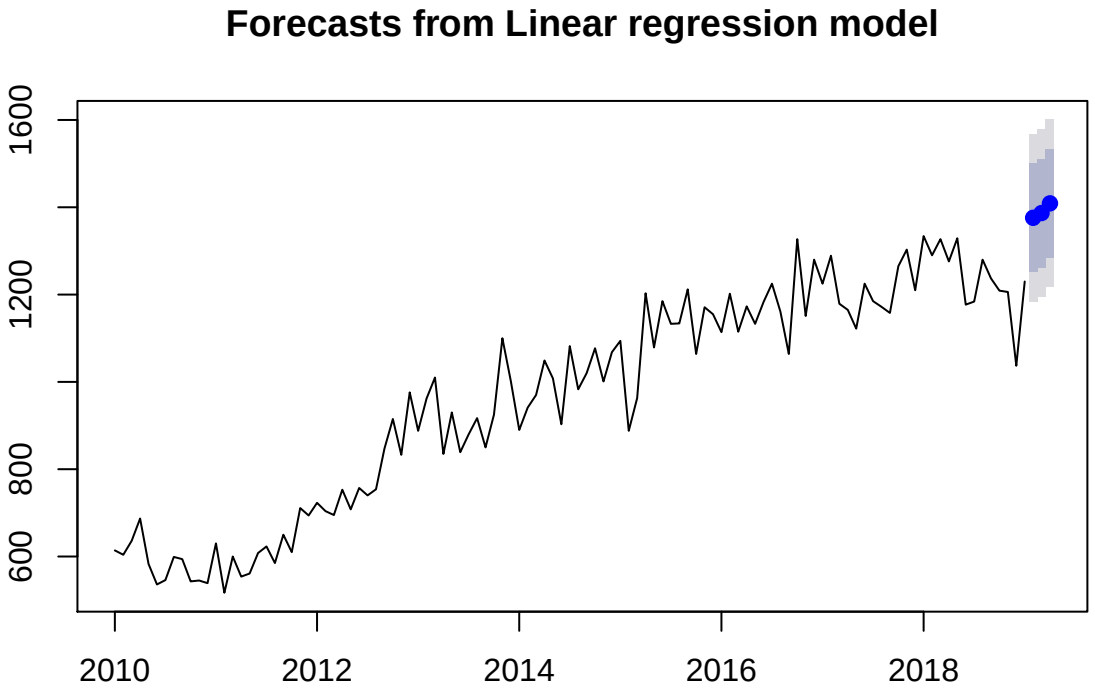
I played around with various iterations below for practice.

```
fcast <- forecast(fit.tslm, h=3)

autoplot(fcast) + ggtitle("Forecasted Housing Starts: Oct 1, 2018 -
                          Jan 9, 2019") + xlab("Month") + ylab("Forecasted
                                                              Housing Starts")
```




```
f1 <- forecast(fit.tslm, h=3)
plot(f1)
```

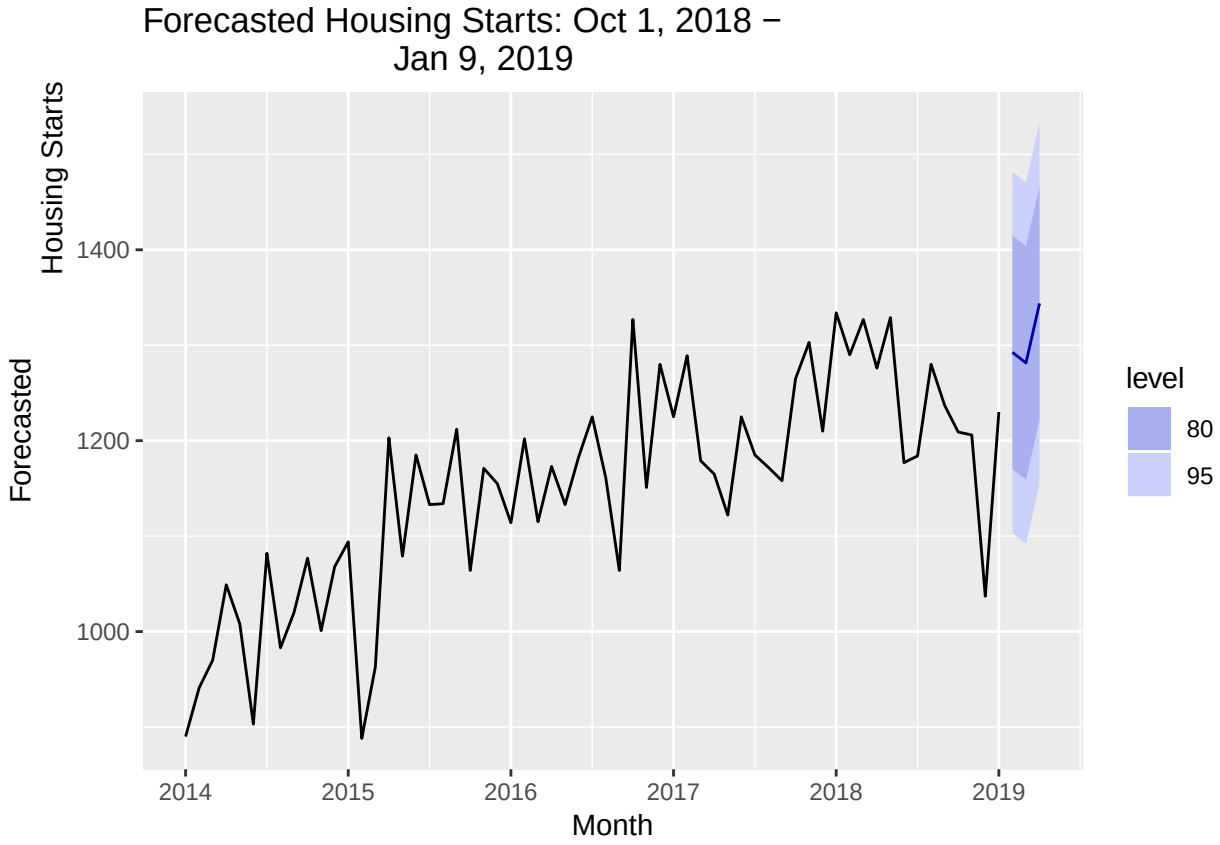


Wanted to see if forecast would change using smaller window of historical data.

```
house.2014 <- window(house.starts, start=c(2014, 1))
fit.2014 <- tslm(house.2014 ~ trend + season)
```

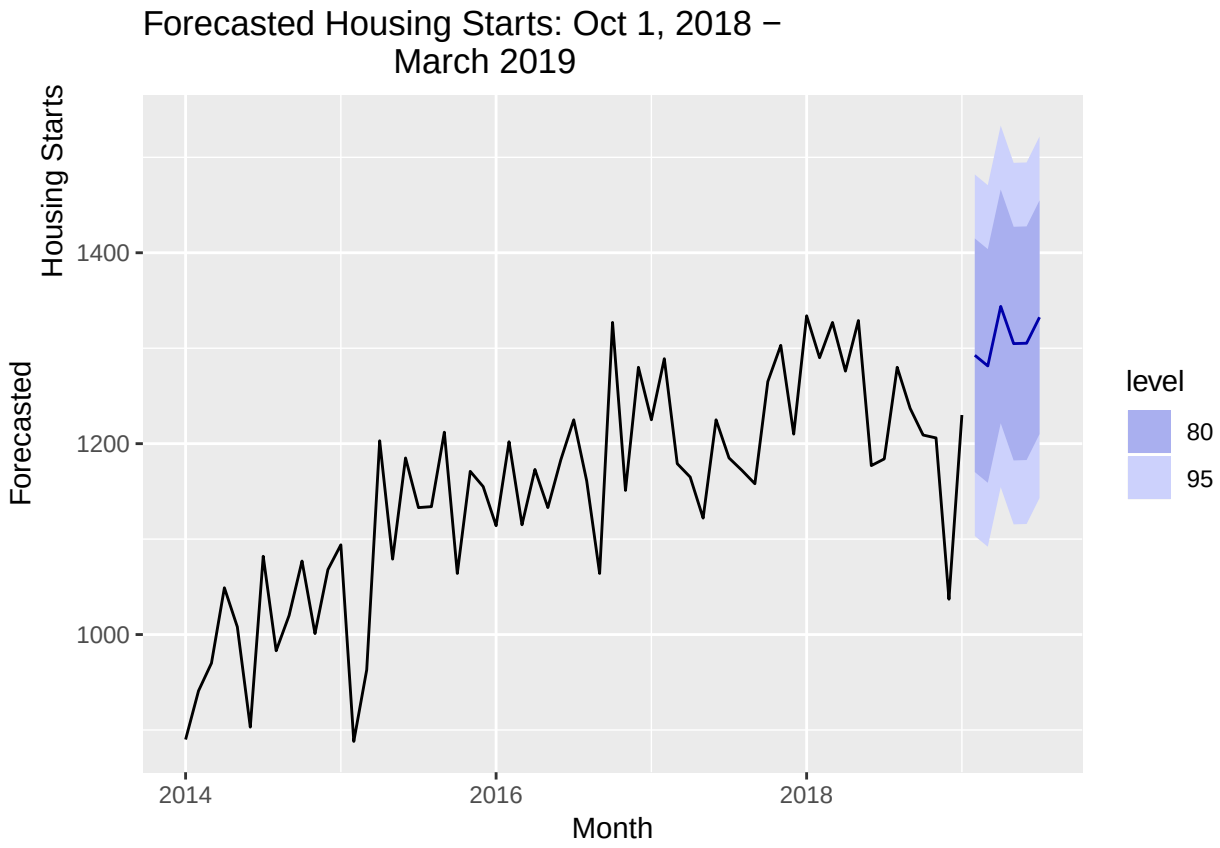
```
fcast <- forecast(fit.2014, h=3)

autoplot(fcast) + ggtitle("Forecasted Housing Starts: Oct 1, 2018 -
                          Jan 9, 2019") + xlab("Month") + ylab("Forecasted
                                                              Housing Starts")
```




```
fcast.6 <- forecast(fit.2014, h=6)

autoplot(fcast.6) + ggtitle("Forecasted Housing Starts: Oct 1, 2018 -
                          March 2019") + xlab("Month") + ylab("Forecasted
                                                              Housing Starts")
```


Forecasted Housing Starts: Oct 1, 2018 − March 2019
Housing Starts
Forecasted
1400
1200
1000
2014
2016
2018
Month
level
80
95