Applied Research Project - Housing Analysis (California)

Xiaodan Chen

2021-06-06

Introduction:

There are two parts in this project. In the first part, I used *time series* method to show the trends and pattern of house inventory and house sales count monthly in California (2008 Jan. - 2021 Apr.). I compared the difference and found the relationship between sales count and house inventory using *cor.test()*, *linear model* and *anova()* methods. I also used *ARIMA()* and *ETS()* method to forecast the sales count in San Francisco in 3 years.
In the second part, I used *time series* method to show the trends and pattern of average house value in California (1996 Jan. - 2021 Apr.). I made the time series for the value of different types of house (single family, condo, one bedroom, two bedrooms, three bedrooms, four bedrooms and five bedrooms) in San Francisco (1996 Jan. - 2021 Apr.). At the end, I forecast the house value of single family house in San Francisco in 3 years.

The databases of this project are downloaded from zillow.com.

```
options(Ncpus = 8)
```

```
library(pacman)
p_load(fs, readr, lubridate, tidyverse, janitor, DataExplorer, summarytools, data.table, dtplyr, ggplot2
```

Part 1. Inventory and Sales Count

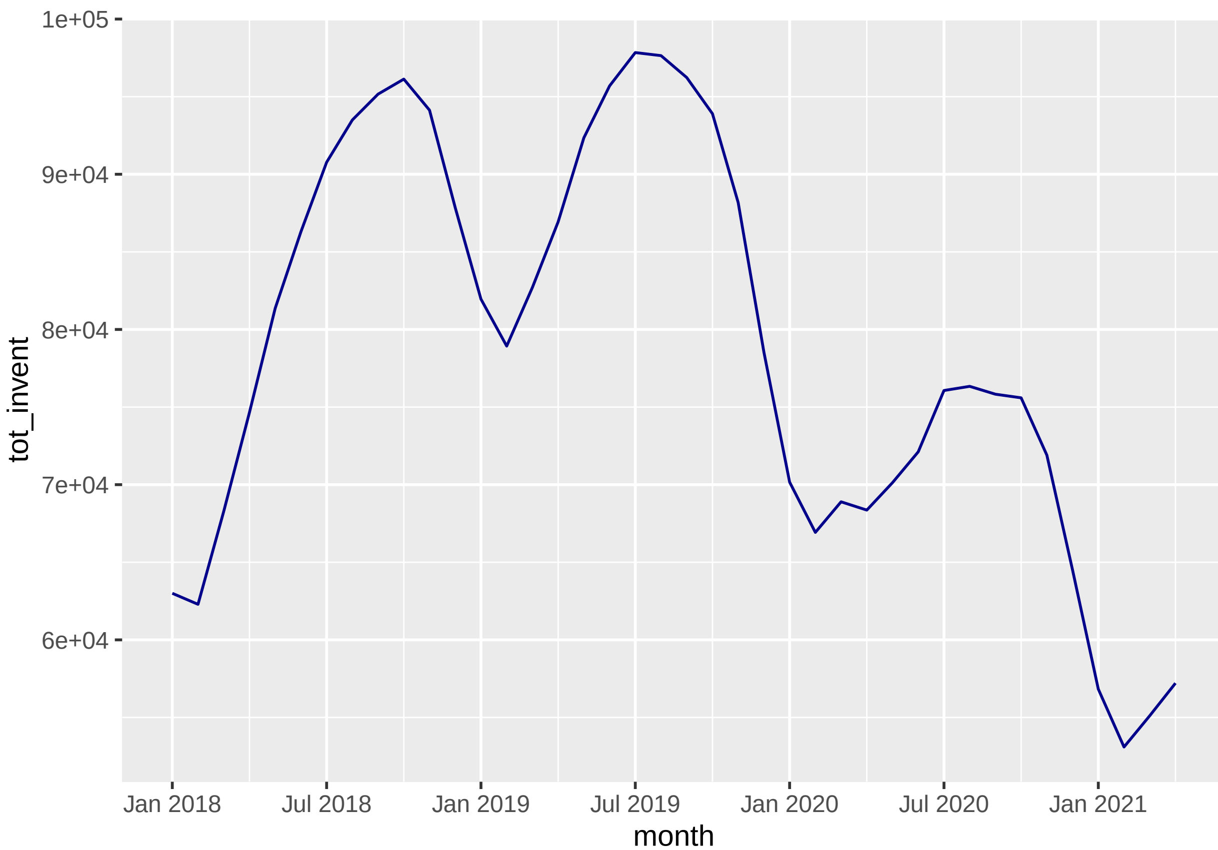
House Inventory

Downloading and cleaning the inventory data set for further analysis.

```
inventory <- read_csv('https://files.zillowstatic.com/research/public_v2/invt_fs/Metro_invt_fs_uc_sfrcon

inventory <- inventory %>%
  separate(RegionName, c('city', 'state'), sep = ',') %>%
  filter(StateName == 'CA') %>%
  pivot_longer(-c(1:6), names_to = 'date', values_to = 'inventory') %>%
  filter(!is.na(inventory))

inventory$month <- as.yearmon(inventory$date)
```

The time series of inventory shows there are peaks value in the middle of the years and bottoms at the second month of the years. It also shows there is a decresing trend over all.

```
invent <- inventory %>% group_by(month) %>%
  summarize(tot_invent = sum(inventory))

invent %>%
  ggplot(aes(x = month, y = tot_invent)) +
  geom_line(col = 'dark blue')
```



House Sales Count

Downloading and cleaning the sales count data set for further analysis.

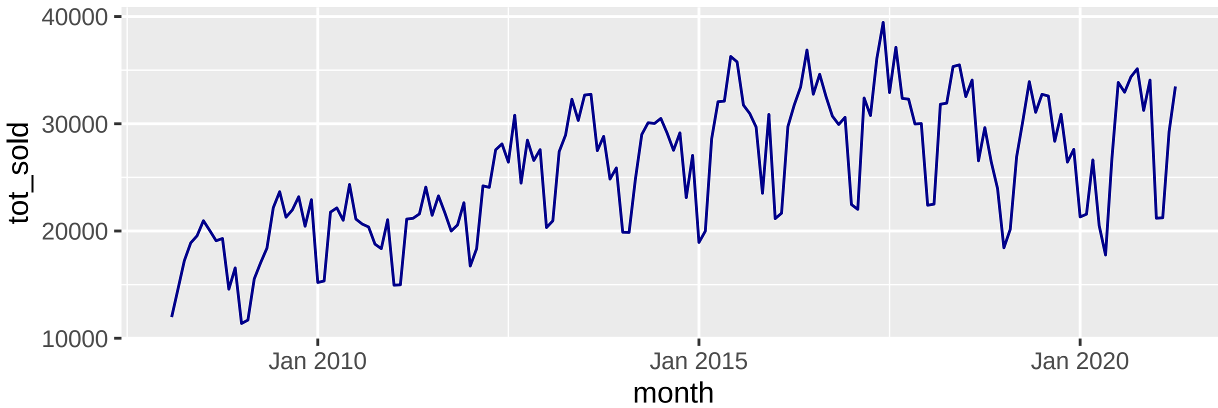
```
house_county <- read.csv('https://files.zillowstatic.com/research/public_v2/sales_count_now/Metro_sales_

sale_county <- house_county %>%
  pivot_longer(-c(1:5), names_to = 'date', values_to = 'sold') %>%
  filter(StateName == 'CA') %>%
  separate(RegionName, c('city', 'state'), sep = ',') %>%
  filter(!is.na(sold))

sale_county$date <- as.Date(sale_county$date, format = 'X%Y.%m.%d')
```

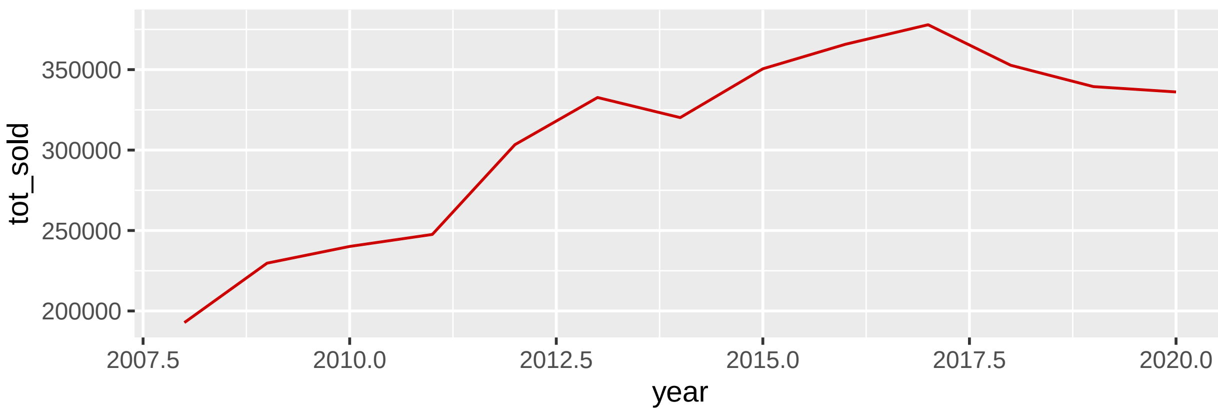
```r
sale_county <- sale_county %>% mutate(month = as.yearmon(date))

head(sale_county, n = 3)
```

```
## # A tibble: 3 x 9
##   RegionID SizeRank city       state RegionType StateName date        sold month
##      <int>    <int> <chr>      <chr> <chr>      <chr>     <date>     <dbl> <yea>
## 1   753899        2 Los Angel~ " CA" Msa        CA        2008-02-29  3625 Feb ~
## 2   753899        2 Los Angel~ " CA" Msa        CA        2008-03-31  4381 Mar ~
## 3   753899        2 Los Angel~ " CA" Msa        CA        2008-04-30  5197 Apr ~
```

The first plot is the trend of sales count for each month. There are more houses sold in the middle of the year than the other time. And the winter season has the lowest number of houses sold. The second plot shows an increase in sales count from 2008 to 2017 and start to decrease from 2017.

```r
sale1 <- sale_county %>%
  select(month, sold) %>%
  group_by(month) %>%
  summarize(tot_sold = sum(sold))

a <- ggplot(sale1, aes(x=month, y=tot_sold)) +
    geom_line(col = 'blue4') +
    theme(aspect.ratio=0.3)

sale2 <- sale_county %>%
  select(month, sold) %>%
  mutate(year = year(month)) %>%
  group_by(year) %>%
  summarize(tot_sold = sum(sold)) %>%
  filter(year < '2021')

b <- ggplot(sale2, aes(x=year, y=tot_sold)) +
    geom_line(col = 'red3') +
    theme(aspect.ratio=0.3)

ggarrange(a,b, nrow = 2)
```

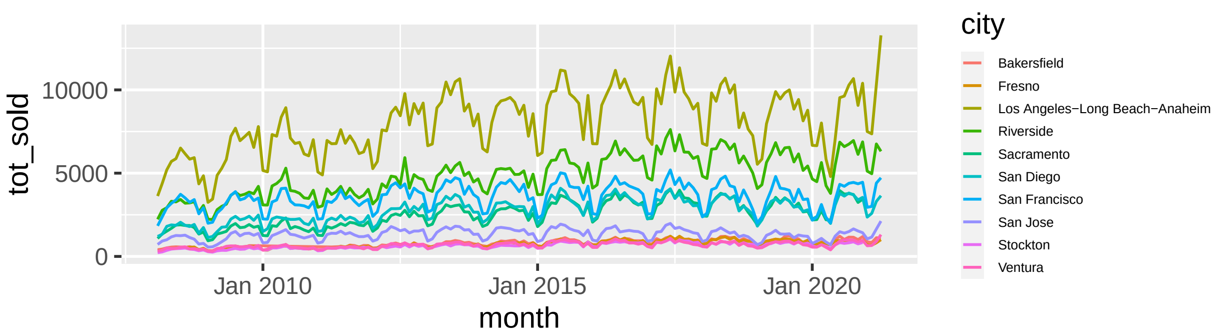


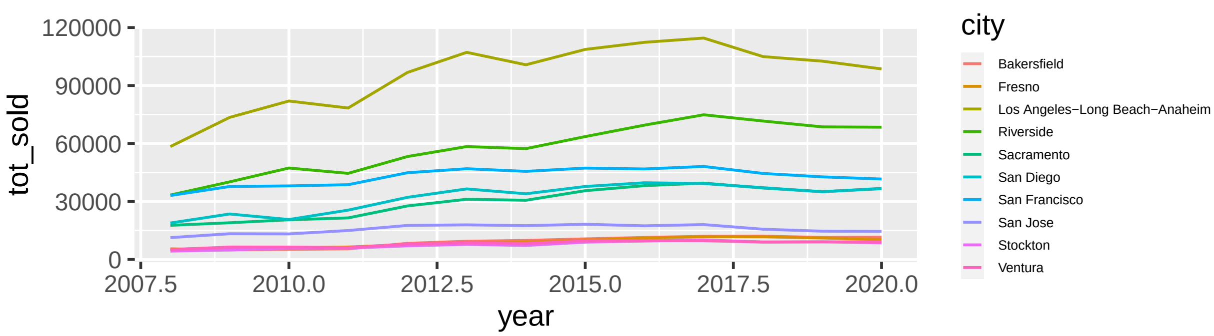


Time series of sales count in different cities.

```
sale_region_1 <- sale_county %>%
  select(month, sold, city) %>%
  group_by(month, city) %>%
  summarize(tot_sold = sum(sold))

a <- ggplot(sale_region_1, aes(x=month, y=tot_sold, colour = city)) +
    geom_line() +
    theme(aspect.ratio=0.3,
          legend.text = element_text(size = 5),
          legend.key.size = unit(0.3, "cm"))


sale_region_2 <- sale_county %>%
  select(month, sold, city) %>%
  mutate(year = year(month)) %>%
  group_by(year, city) %>%
  summarize(tot_sold = sum(sold)) %>%
  filter(year < '2021')

b <- ggplot(sale_region_2, aes(x=year, y=tot_sold, col = city)) +
    geom_line() +
    theme(aspect.ratio=0.3,
          legend.text = element_text(size = 5),
```

```
         legend.key.size = unit(0.3, "cm"))

ggarrange(a,b, nrow = 2)
```





House Inventory vs. Sales Count

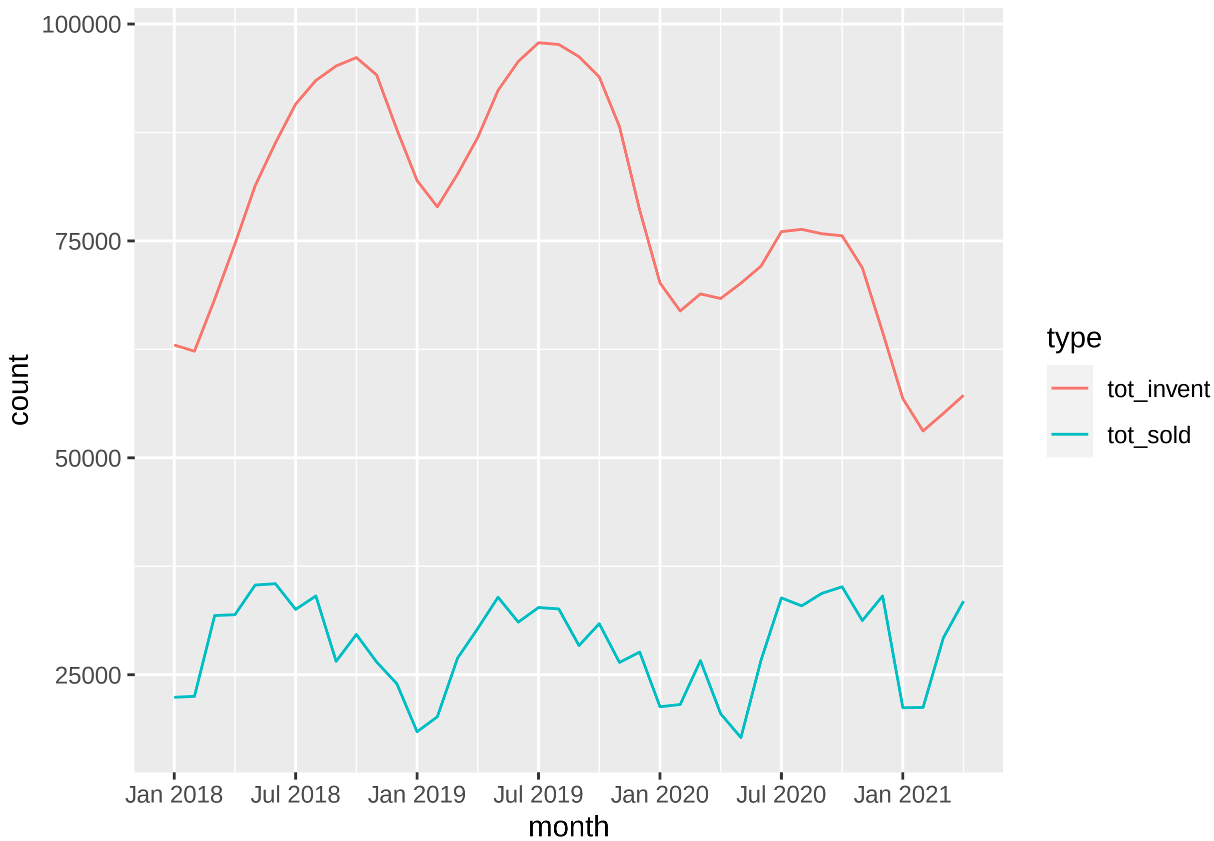
Using *inner_join()* function to join the sales count and inventory data frames.

```
sale_inventory <- invent %>%
  inner_join(sale1, by = 'month') %>%
  select(month, tot_invent, tot_sold)

head(sale_inventory, n = 3)
```

```
## # A tibble: 3 x 3
##   month     tot_invent tot_sold
##   <yearmon>      <dbl>    <dbl>
## 1 Jan 2018       63001    22402
## 2 Feb 2018       62291    22508
## 3 Mar 2018       68283    31822
```

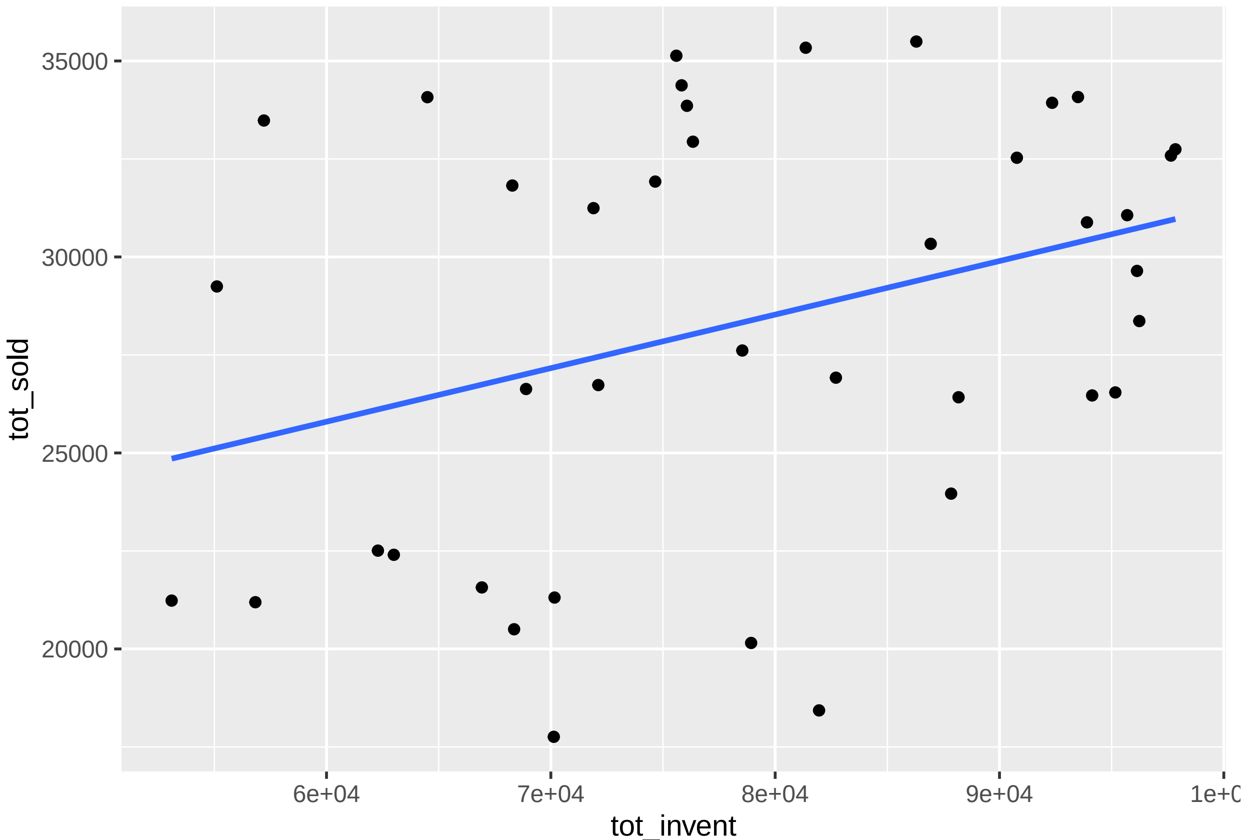
Comparing the time series of inventroy and sales count in California, they are having a similar pattern, while the sales count does not have a decreasing trend as inventory.

```r
sale_inventory0 <- sale_inventory %>%
  pivot_longer(tot_invent:tot_sold, names_to = 'type', values_to = 'count')

ggplot(sale_inventory0,aes(x=month, y=count, col = type)) +
    geom_line()
```



The scatter plot show the linear relationship between total inventory and total sales count of California is not clear nor clear.

```r
sale_inventory %>%
  ggplot(aes(x=tot_invent, y=tot_sold)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE)
```



The result of *cor.test()* shows the correlation coefficient for California house inventory and sales count is about 0.34, which agrees with the plot above that the linear relationship between them is not strong.

```
cor.test(sale_inventory$tot_invent, sale_inventory$tot_sold)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sale_inventory$tot_invent and sale_inventory$tot_sold
## t = 2.2255, df = 38, p-value = 0.03206
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.03137807 0.58879560
## sample estimates:
##       cor
## 0.3395683
```

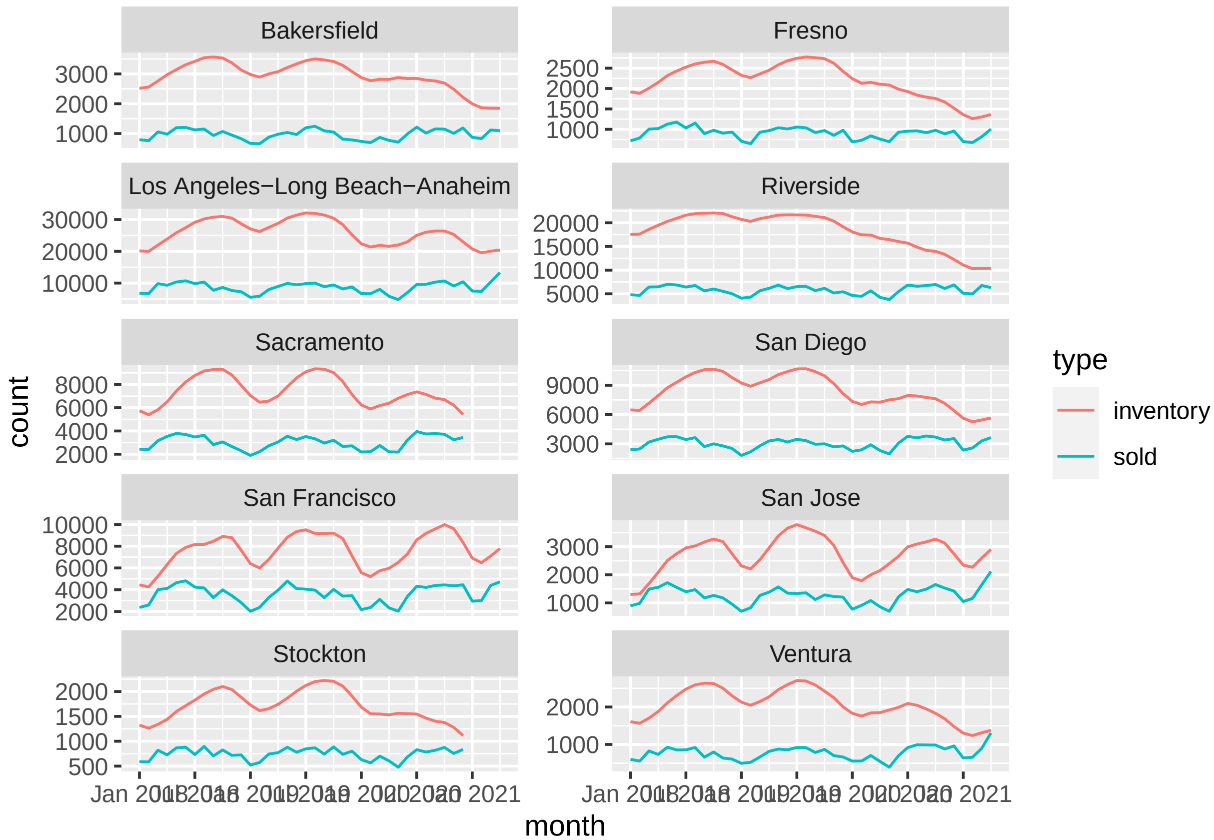
The result of simple linear regression model shows the inventory is a significant predictor, while the R-squared value shows this model is not a good model to explain the responser.

```
mod_lm <- lm(tot_sold ~ tot_invent, data = sale_inventory)
summary(mod_lm)
```

```
##
## Call:
## lm(formula = tot_sold ~ tot_invent, data = sale_inventory)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10364.8  -3852.1      5.1   4119.2   8064.1
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.760e+04  4.891e+03   3.599  0.00091 ***
## tot_invent  1.366e-01  6.139e-02   2.225  0.03206 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5081 on 38 degrees of freedom
## Multiple R-squared:  0.1153, Adjusted R-squared:  0.09203
## F-statistic: 4.953 on 1 and 38 DF,  p-value: 0.03206
```

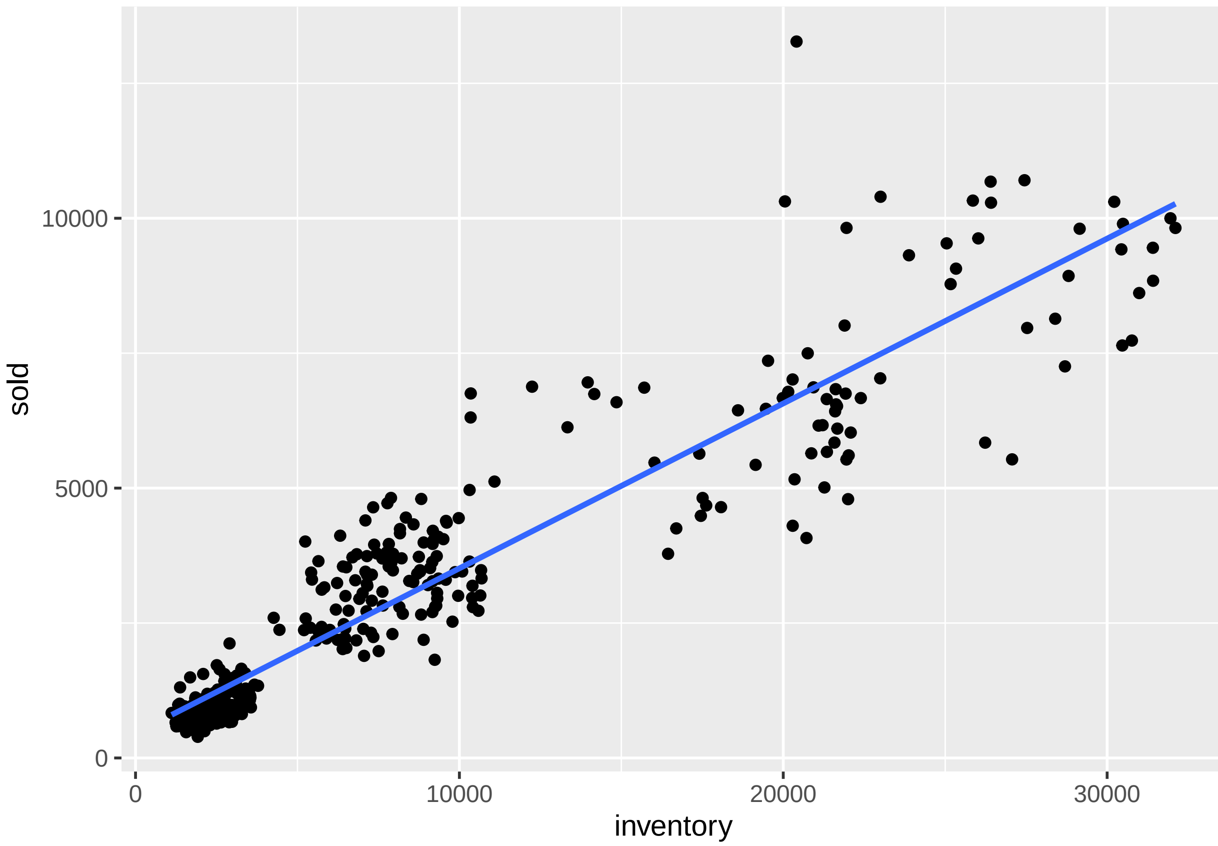
Comparing the time series of inventroy and sales count in cities of California. There is a similar pattern between the two variables for cities: Sacramento, San Francisco and San Jose. There is a decreasing inventory trend for cities: Bakersfield, Fresno, Riverside and Ventura.

```
sale_invent_ct <- inventory %>%
  inner_join(sale_county, by = c('month', 'city'))

sale_invent_ct %>%
  pivot_longer(c(inventory,sold), names_to = 'type', values_to = 'count') %>%
  ggplot(aes(x=month, y=count, col = type)) +
  geom_line() +
  facet_wrap(.~ city, nrow = 5, scales = 'free_y')
```

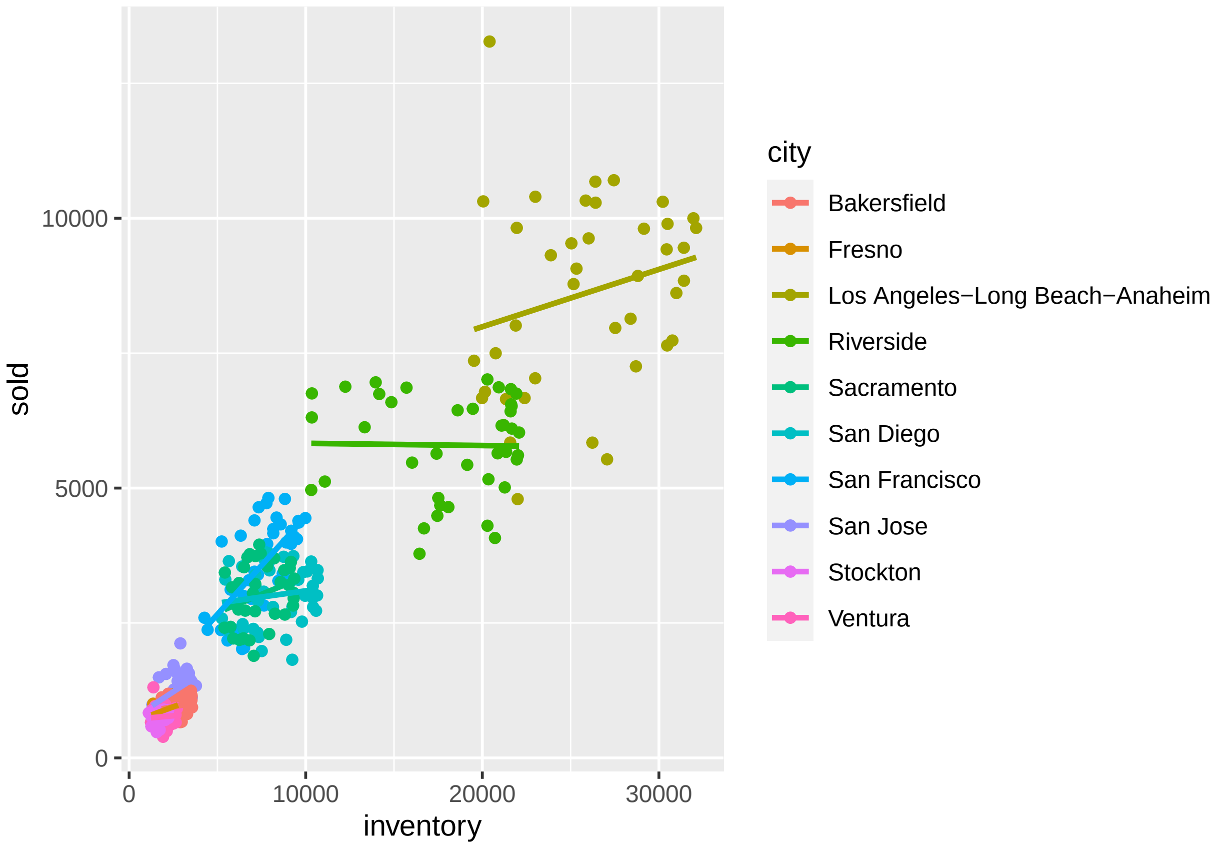


The scatter plots show a clear and strong linear relationship between the sales count and inventory for each cities.

```
sale_invent_ct %>%
  ggplot(aes(x=inventory, y=sold)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE)
```




```
sale_invent_ct %>%
  ggplot(aes(x=inventory, y=sold, col=city)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE)
```



The *anova()* function indicats that add the city variable to the linear regression model is necessary. Both explaining variables (inventory and city) are significant for estimating the response variable sales count.

```
mod_lm2 <- lm(sold ~ inventory * city, data = sale_invent_ct)
anova(mod_lm2)
```

```
## Analysis of Variance Table
##
## Response: sold
##                 Df     Sum Sq    Mean Sq   F value  Pr(>F)
## inventory        1 2310764376 2310764376 4479.0247 < 2e-16 ***
## city             9  123980108   13775568   26.7016 < 2e-16 ***
## inventory:city   9   12277856    1364206    2.6443 0.00559 **
## Residuals      372  191917750     515908
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The result of *cor.test()* (r = 0.94) shows the sales count and inventory have a strong positve linear relationship, which agrees with the conclusion above.

```
cor.test(sale_invent_ct$inventory, sale_invent_ct$sold)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sale_invent_ct$inventory and sale_invent_ct$sold
## t = 52.403, df = 390, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9221811 0.9470297
## sample estimates:
##       cor
## 0.935757
```

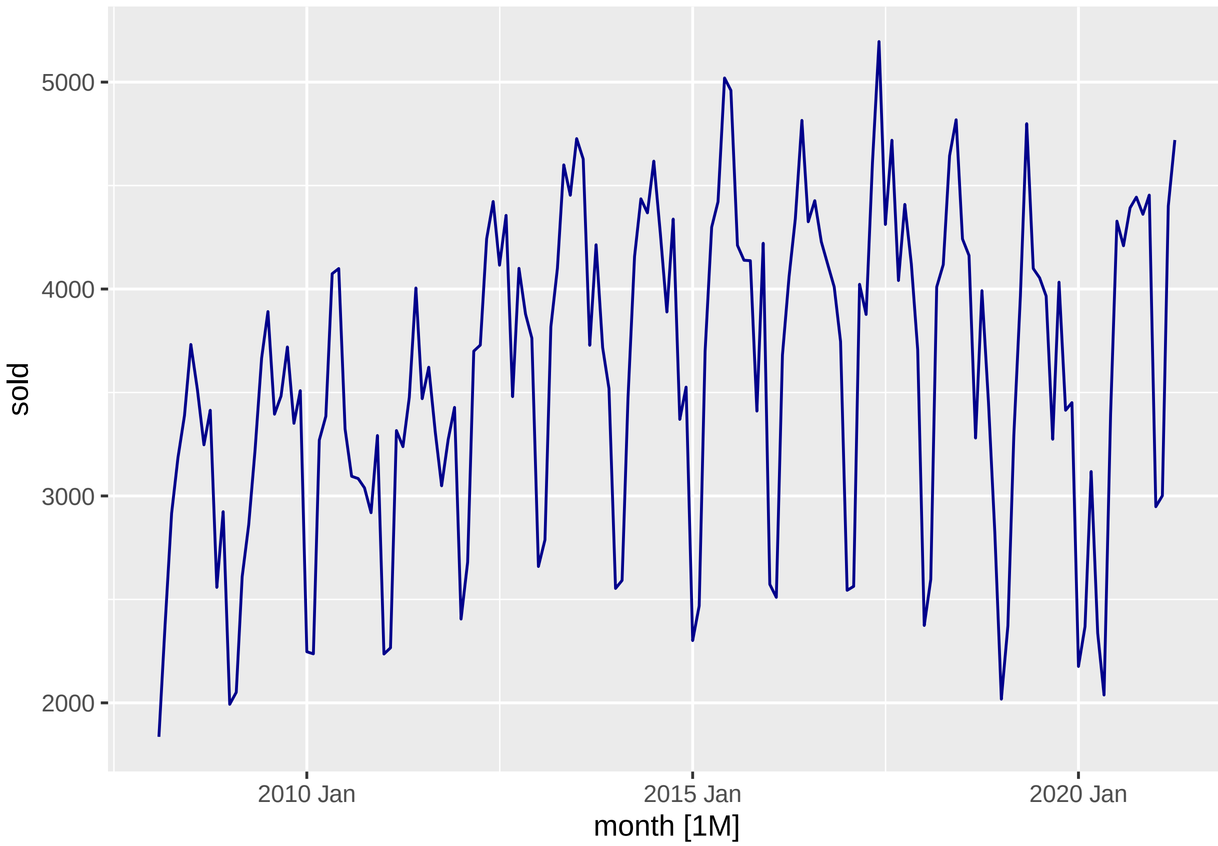
Forecast the sales count of San Francisco in 3 years.

```
cityname <- 'San Francisco'

tss <- sale_county %>%
  filter(city == cityname) %>%
  select(month, sold) %>%
  mutate(month = yearmonth(month)) %>%
  as_tsibble(index = month)
head(tss, n=3)
```

```
## # A tsibble: 3 x 2 [1M]
##      month  sold
##      <mth> <dbl>
## 1 2008 Feb  1836
## 2 2008 Mar  2374
## 3 2008 Apr  2916
```

Time series of sales count for San Francisco from 2008 February to 2021 April.

```
tss %>% autoplot(col = 'blue4')
```



Determining whether differencing is required using *unitroot_kpss()* test.

```
tss %>%
  features(sold, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1     0.658      0.0174
```

The p-value is less than 0.05, indicating that the null hypothesis is rejected. That is, the data are not stationary. We can difference the data, and apply the test again.

```
tss %>%
  mutate(diff_sold = difference(sold)) %>%
  features(diff_sold, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1    0.0234         0.1
```

Determining the appropriate *number* of first differences is carried out using the *unitroot_ndiffs()* feature.

```
tss %>%
  features(sold, unitroot_ndiffs)
```

```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1
```

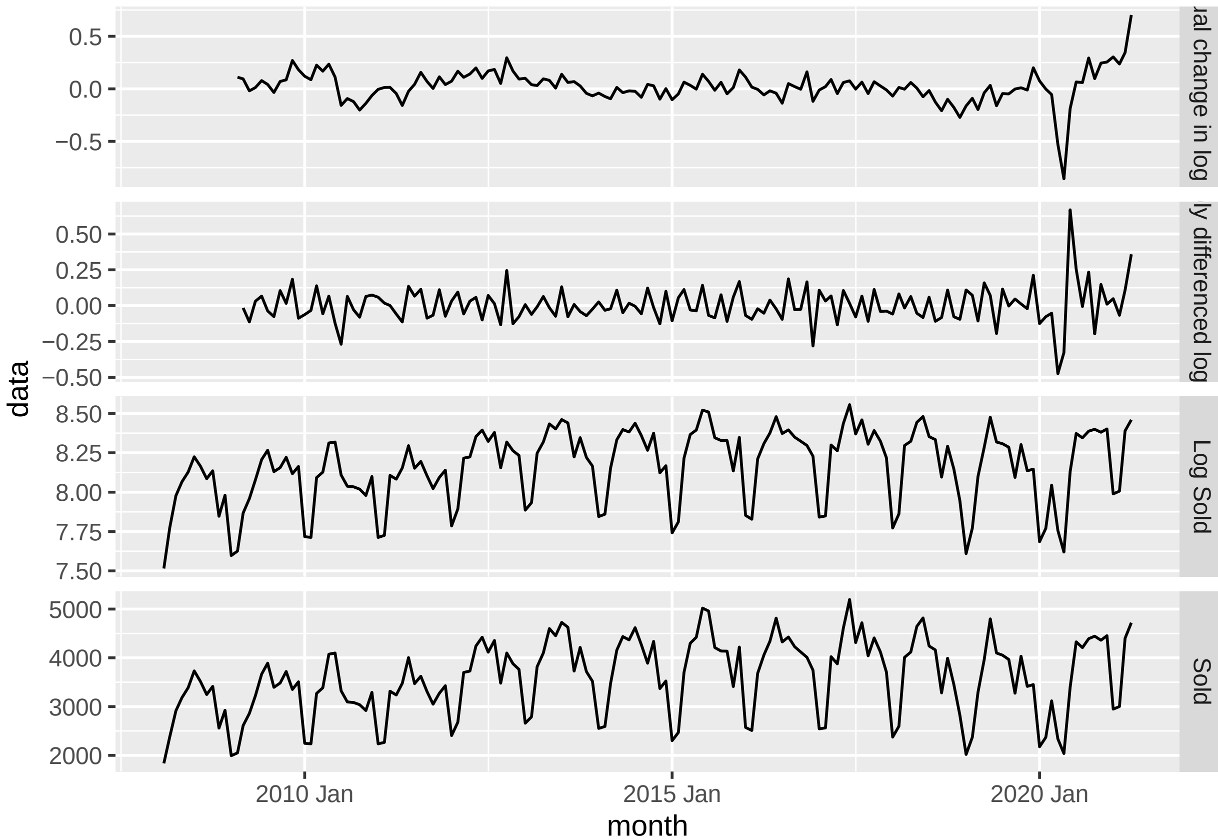
Determining whether seasonal differencing is required using *unitroot_nsdiffs()* function.

```
tss %>%
  features(sold, unitroot_nsdiffs)
```

```
## # A tibble: 1 x 1
##   nsdiffs
##     <int>
## 1       1
```

The time series shows stationary after transmution.

```
tss %>%
  transmute(
    `Sold` = sold,
    `Log Sold` = log(sold),
    `Annual change in log Sold` = difference(log(sold), 12),
    `Doubly differenced log Sold` =
                     difference(difference(log(sold), 12), 1)) %>%
  pivot_longer(-month, names_to="data_type", values_to="data") %>%
  mutate(
    data_type = as.factor(data_type)) %>%
  ggplot(aes(x = month, y = data)) +
  geom_line() +
  facet_grid(vars(data_type), scales = "free_y")
```



Comparing ARIMA() and ETS() model.

Splitting the data from 2008 Jan. to 2018 Dec. as training data, and the rest data to testing data.

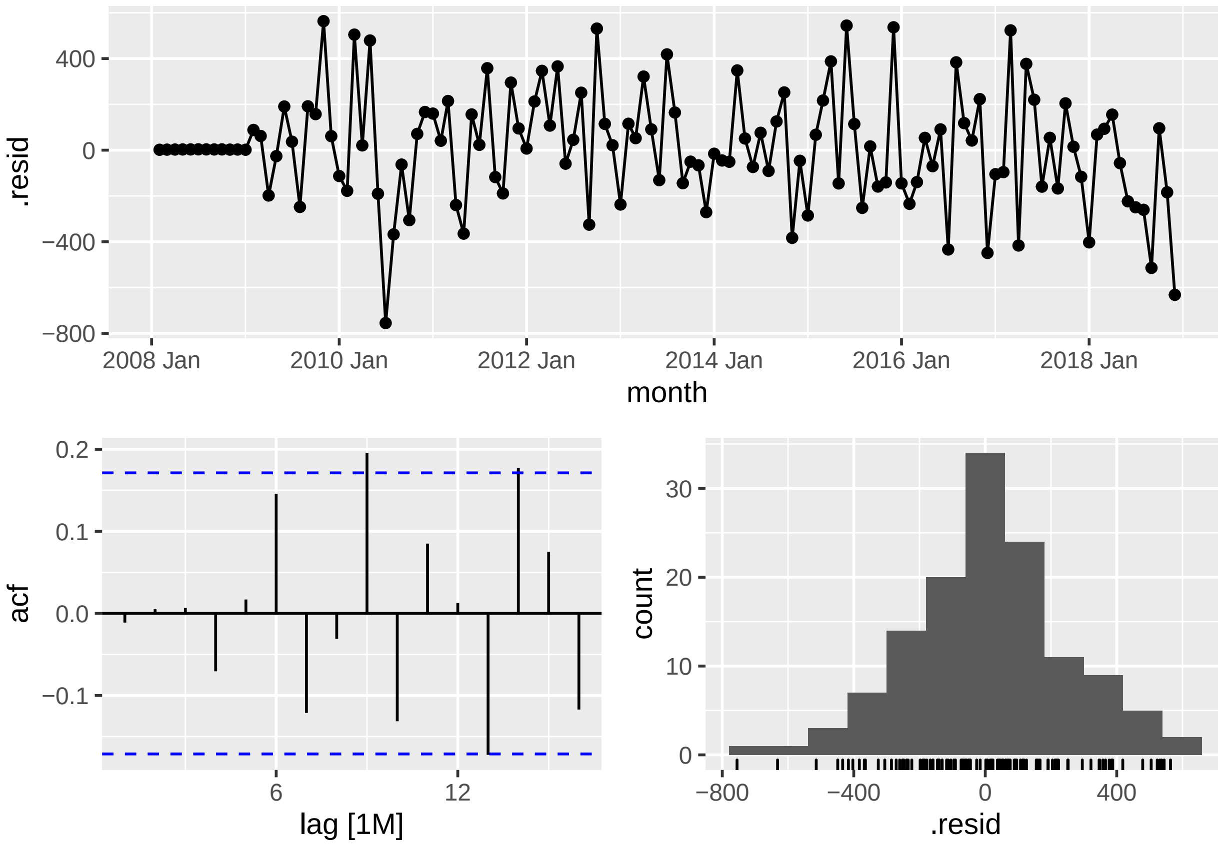
```
train <- tss %>%
  filter_index(. ~ "2018 Dec")
```

ARIMA()

```
fit_arima <- train %>% model(ARIMA(sold))
report(fit_arima)
```

```
## Series: sold
## Model: ARIMA(3,0,1)(0,1,2)[12] w/ drift
##
## Coefficients:
##           ar1     ar2     ar3     ma1      sma1      sma2  constant
##       -0.3457  0.5026  0.3654  0.8502  -0.4474  -0.3063   38.6205
## s.e.   0.1332  0.1125  0.1006  0.1279   0.1240   0.1144   17.7532
##
## sigma^2 estimated as 69123:  log likelihood=-833.21
## AIC=1682.43   AICc=1683.74   BIC=1704.66
```

```r
fit_arima %>% gg_tsresiduals(lag_max = 16)
```




```r
augment(fit_arima) %>%
  features(.innov, ljung_box, lag = 16, dof = 6)
```

```
## # A tibble: 1 x 3
##   .model      lb_stat lb_pvalue
##   <chr>         <dbl>     <dbl>
## 1 ARIMA(sold)    26.9   0.00269
```

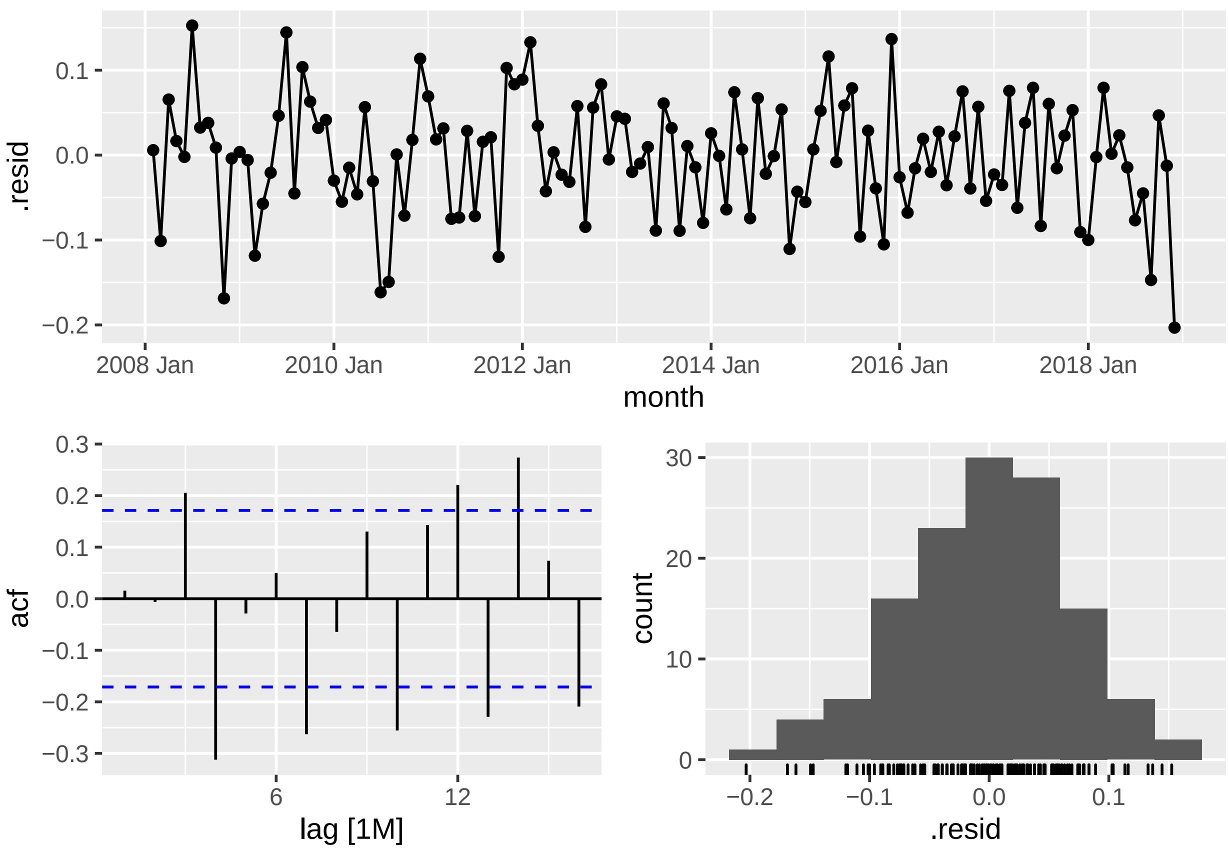
ETC()

```r
fit_ets <- train %>% model(ETS(sold))
report(fit_ets)
```

```
## Series: sold
## Model: ETS(M,Ad,M)
##   Smoothing parameters:
##     alpha = 0.4192802
##     beta  = 0.0001182539
##     gamma = 0.0001021992
##     phi   = 0.9794088
##
##   Initial states:
```

```
##         l        b        s1         s2         s3        s4        s5        s6
##   2663.68 30.63286 0.6583096 0.9793581 0.9617384 1.070028 1.013035 1.127003
##        s7       s8       s9       s10       s11       s12
##   1.150471 1.217332 1.141757 1.035703 0.9676088 0.677655
##
##   sigma^2:  0.0053
##
##      AIC     AICc      BIC
## 2111.121 2117.228 2162.875
```

```
fit_ets %>%
  gg_tsresiduals(lag_max = 16)
```




```
augment(fit_ets) %>%
  features(.innov, ljung_box, lag = 16, dof = 6)
```

```
## # A tibble: 1 x 3
##   .model     lb_stat lb_pvalue
##   <chr>        <dbl>     <dbl>
## 1 ETS(sold)     78.2  1.13e-12
```

The output below evaluates the forecasting performance of the two competing models over the train and test set. In this case the ARIMA model seems to be the slightly more accurate model based on the test set RMSE, MAPE and MASE.

```r
bind_rows(
    fit_arima %>% accuracy(),
    fit_ets %>% accuracy(),
    fit_arima %>% forecast(h = "3 years") %>%
      accuracy(tss),
    fit_ets %>% forecast(h = "3 years") %>%
      accuracy(tss)
  ) %>%
  select(-ME, -MPE, -ACF1)
```

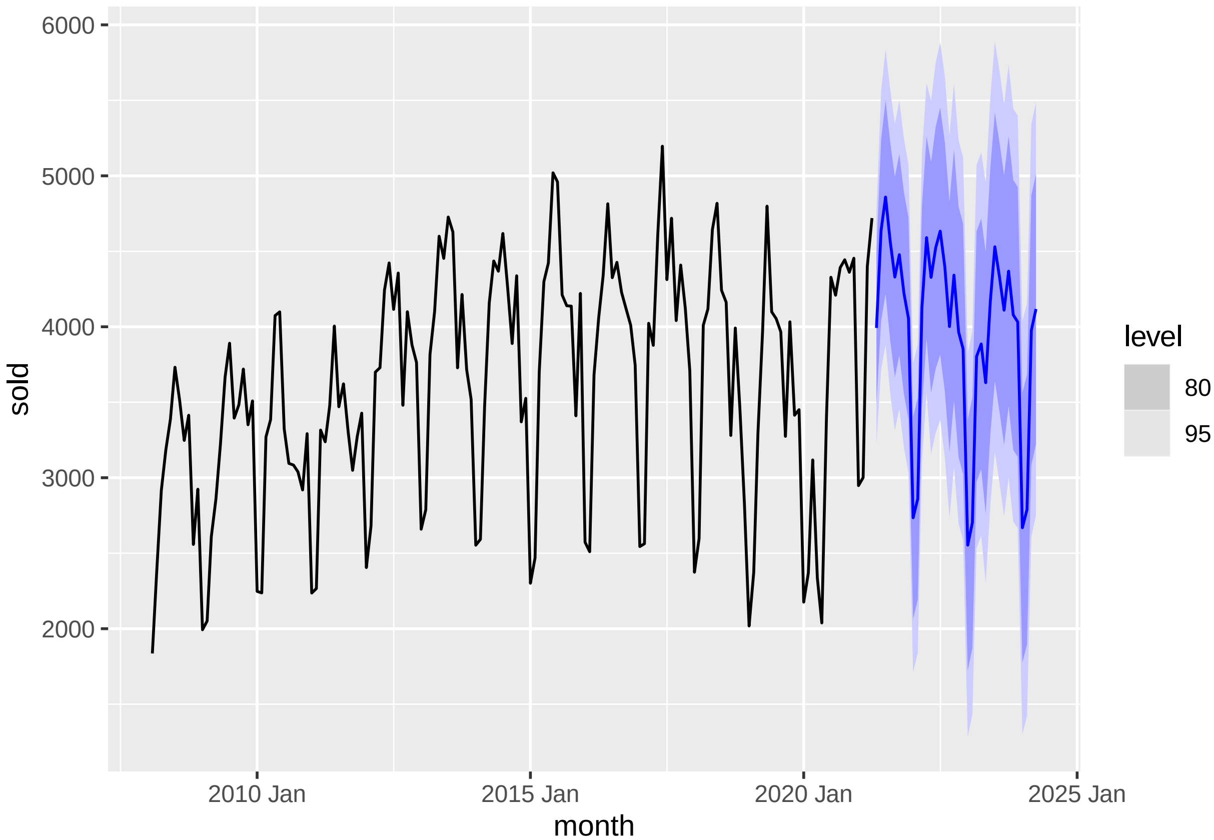
```
## # A tibble: 4 x 7
##   .model      .type     RMSE   MAE  MAPE  MASE RMSSE
##   <chr>       <chr>    <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ARIMA(sold) Training  243.  183.  5.13 0.635 0.658
## 2 ETS(sold)   Training  246.  195.  5.51 0.676 0.667
## 3 ARIMA(sold) Test      713.  457. 15.5  1.59  1.93
## 4 ETS(sold)   Test      745.  579. 16.9  2.01  2.02
```

Generating and ploting forecasts from the ARIMA model for the next 3 years.

```r
tss %>%
  model(ARIMA(sold)) %>%
  forecast(h="3 years") %>%
  head(n = 5)
```

```
## # A fable: 5 x 4 [1M]
## # Key:     .model [1]
##   .model         month            sold .mean
##   <chr>          <mth>          <dist> <dbl>
## 1 ARIMA(sold) 2021 May N(3992, 153230) 3992.
## 2 ARIMA(sold) 2021 Jun N(4638, 220239) 4638.
## 3 ARIMA(sold) 2021 Jul N(4859, 249542) 4859.
## 4 ARIMA(sold) 2021 Aug N(4566, 262357) 4566.
## 5 ARIMA(sold) 2021 Sep N(4329, 267960) 4329.
```

```r
tss %>%
  model(ARIMA(sold)) %>%
  forecast(h="3 years") %>%
  autoplot(tss)
```



Part 2. House Value and Forecasting

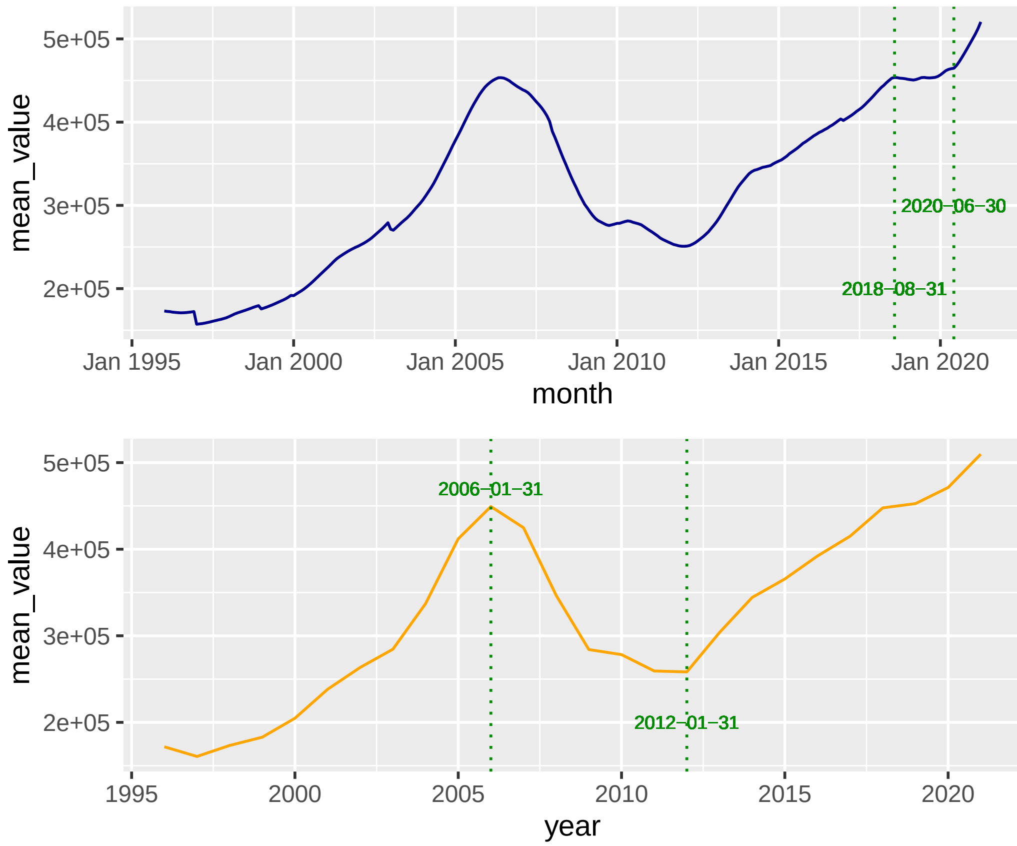
Downloading and cleaning the house value data set for further analysis.

```
house_value <- read.csv('https://files.zillowstatic.com/research/public_v2/zhvi/Metro_zhvi_uc_sfrcondo_t

house_value <- house_value %>%
  separate(RegionName, c('city', 'state'), sep = ',') %>%
  pivot_longer(-c(1:6), names_to = 'date', values_to = 'value') %>%
  filter(StateName == 'CA') %>%
  filter(!is.na(value))

house_value$date <- as.Date(house_value$date, format = 'X%Y.%m.%d')

house_value <- house_value %>% mutate(month = as.yearmon(date))

head(house_value, n = 3)
```

```
## # A tibble: 3 x 9
##   RegionID SizeRank city      state RegionType StateName date        value month
##      <int>    <int> <chr>     <chr> <chr>      <chr>     <date>      <dbl> <yea>
## 1   753899        2 Los Ange~ " CA" Msa        CA        1996-01-31 188830 Jan ~
## 2   753899        2 Los Ange~ " CA" Msa        CA        1996-02-29 189094 Feb ~
## 3   753899        2 Los Ange~ " CA" Msa        CA        1996-03-31 189114 Mar ~
```

Time series of average house value in California from 1996 January to 2021 April. It increases form 1996 to 2006 and from 2012 to 2021. It decreases from 2006 to 2012. In the monthly plot, there is a flat trend between August 2018 to June 2020.

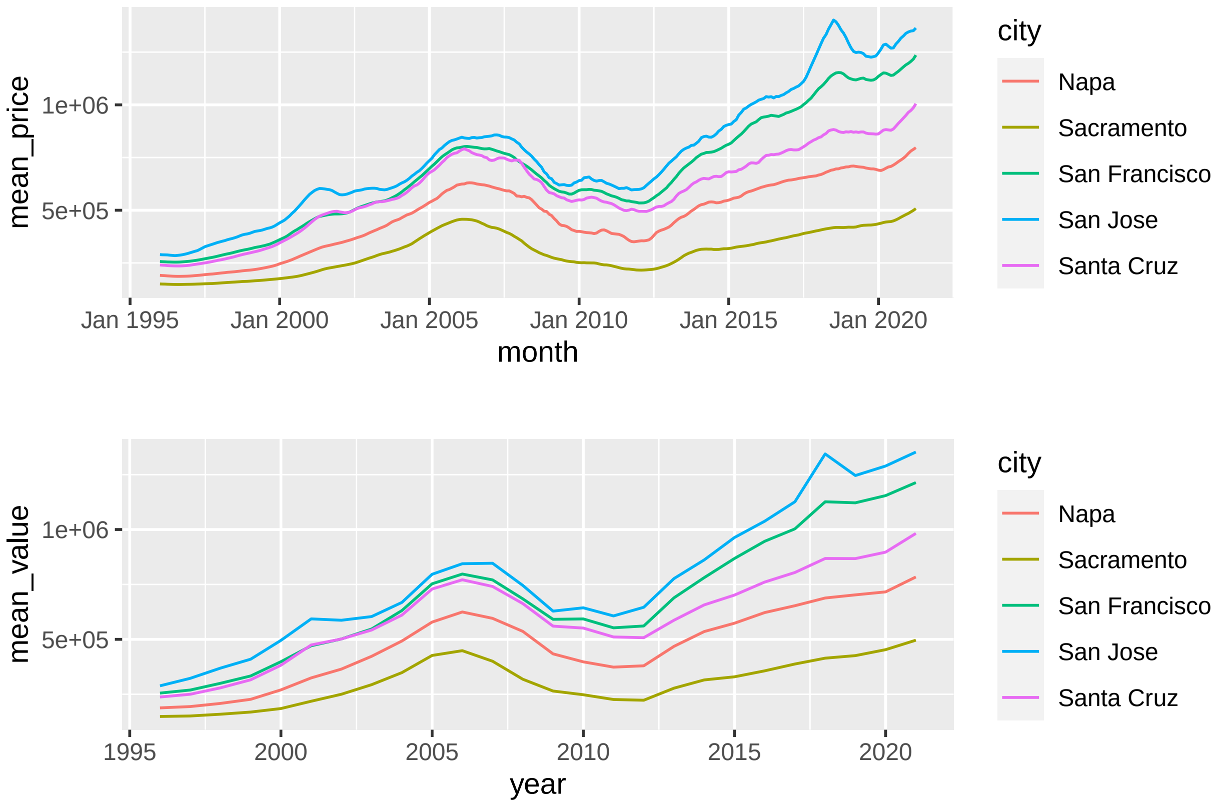
```
value_a <- house_value %>%
  select(city, month, value) %>%
  group_by(month) %>%
  summarize(mean_value = mean(value))

a <- ggplot(value_a, aes(x=month, y=mean_value)) +
  geom_line(col = 'dark blue') +
  geom_vline(aes(xintercept = as.numeric(as.yearmon('2018-08'))),
             linetype = 'dotted', col = 'green4') +
  geom_vline(aes(xintercept = as.numeric(as.yearmon('2020-06'))),
             linetype = 'dotted', col = 'green4') +
  geom_text(aes(x=as.numeric(as.yearmon('2018-08')), y = 200000),
          label = '2018-08-31', size = 2.5, col = 'green4')+
  geom_text(aes(x=as.numeric(as.yearmon('2020-06')), y = 300000),
          label = '2020-06-30', size = 2.5, col = 'green4')+
    theme(aspect.ratio=0.37)

value_b <- house_value %>%
  mutate(year = year(month)) %>%
  select(city, year, month, value) %>%
  group_by(year) %>%
  summarize(mean_value = mean(value))

b <- ggplot(value_b,aes(x=year, y=mean_value)) +
  geom_line(col = 'orange') +
  geom_vline(aes(xintercept = as.numeric(as.yearmon('2006-01'))),
             linetype = 'dotted', col = 'green4') +
  geom_vline(aes(xintercept = as.numeric(as.yearmon('2012-01'))),
             linetype = 'dotted', col = 'green4') +
  geom_text(aes(x=as.numeric(as.yearmon('2006-01')), y = 470000),
          label = '2006-01-31', size = 2.5, col = 'green4')+
  geom_text(aes(x=as.numeric(as.yearmon('2012-01')), y = 200000),
          label = '2012-01-31', size = 2.5, col = 'green4')+
    theme(aspect.ratio=0.37)

ggarrange(a,b, nrow = 2)
```



Time series of house value for cities in California. The plots show the cities have similar pattern and trends.

```
city1 <- c('San Francisco', 'San Jose', 'Sacramento', 'Rriveside',
           'Napa', 'Santa Cruz')

value_city <- house_value %>%
  filter(city %in% city1)

value_city1 <- value_city %>%
  select(month, value, city) %>%
  group_by(month, city) %>%
  summarize(mean_price = mean(value))

a <- value_city1 %>%
  ggplot(aes(x=month, y=mean_price, col = city)) +
  geom_line()+
  theme(aspect.ratio=0.35)

value_city2 <- value_city %>%
  mutate(year = year(month)) %>%
  select(year, value, city) %>%
  group_by(year, city) %>%
  summarize(mean_value = mean(value))
```

```
b <- value_city2 %>%
  ggplot(aes(x=year, y=mean_value, col = city)) +
  geom_line() +
  theme(aspect.ratio=0.35)

ggarrange(a,b, nrow = 2)
```



Downing the house value data set for different types of houses (single-family, condo, one-room, two-room, three-room, four-room, five-room).

```
value0sg <- read_csv('https://files.zillowstatic.com/research/public_v2/zhvi/City_zhvi_uc_sfr_tier_0.33_
value0cd <- read_csv('https://files.zillowstatic.com/research/public_v2/zhvi/City_zhvi_uc_condo_tier_0.3
value01 <- read_csv('https://files.zillowstatic.com/research/public_v2/zhvi/City_zhvi_bdrmcnt_1_uc_sfrc
value02 <- read_csv('https://files.zillowstatic.com/research/public_v2/zhvi/City_zhvi_bdrmcnt_2_uc_sfrc
value03 <- read_csv('https://files.zillowstatic.com/research/public_v2/zhvi/City_zhvi_bdrmcnt_3_uc_sfrc
value04 <- read_csv('https://files.zillowstatic.com/research/public_v2/zhvi/City_zhvi_bdrmcnt_4_uc_sfrc
value05 <- read_csv('https://files.zillowstatic.com/research/public_v2/zhvi/City_zhvi_bdrmcnt_5_uc_sfrc
```

Cleaning the data set to be tidy for further analysis.

```
value_1 <- value01 %>%
  filter(StateName == 'CA') %>%
  pivot_longer(-c(1:8) , names_to = 'date', values_to = 'room1') %>%
```

```
  filter(!is.na(room1))
value_2 <- value02 %>%
  filter(StateName == 'CA') %>%
  pivot_longer(-c(1:8)  , names_to = 'date', values_to = 'room2') %>%
  filter(!is.na(room2))
value_3 <- value03 %>%
  filter(StateName == 'CA') %>%
  pivot_longer(-c(1:8)  , names_to = 'date', values_to = 'room3') %>%
  filter(!is.na(room3))
value_4 <- value04 %>%
  filter(StateName == 'CA') %>%
  pivot_longer(-c(1:8)  , names_to = 'date', values_to = 'room4') %>%
  filter(!is.na(room4))
value_5 <- value05 %>%
  filter(StateName == 'CA') %>%
  pivot_longer(-c(1:8)  , names_to = 'date', values_to = 'room5') %>%
  filter(!is.na(room5))
value_s <- value0sg %>%
  filter(StateName == 'CA') %>%
  pivot_longer(-c(1:8)  , names_to = 'date', values_to = 'values') %>%
  filter(!is.na(values))
value_c <- value0cd %>%
  filter(StateName == 'CA') %>%
  pivot_longer(-c(1:8)  , names_to = 'date', values_to = 'valuec') %>%
  filter(!is.na(valuec))

value_1$month <- as.yearmon(value_1$date)
value_2$month <- as.yearmon(value_2$date)
value_3$month <- as.yearmon(value_3$date)
value_4$month <- as.yearmon(value_4$date)
value_5$month <- as.yearmon(value_5$date)
value_s$month <- as.yearmon(value_s$date)
value_c$month <- as.yearmon(value_c$date)

head(value_1, n =2)
```

```
## # A tibble: 2 x 11
##   RegionID SizeRank RegionName RegionType StateName State Metro CountyName date
##      <dbl>    <dbl> <chr>      <chr>      <chr>     <chr> <chr> <chr>      <chr>
## 1    12447        1 Los Angel~ City       CA        CA    Los ~ Los Angel~ 1996~
## 2    12447        1 Los Angel~ City       CA        CA    Los ~ Los Angel~ 1996~
## # ... with 2 more variables: room1 <dbl>, month <yearmon>
```

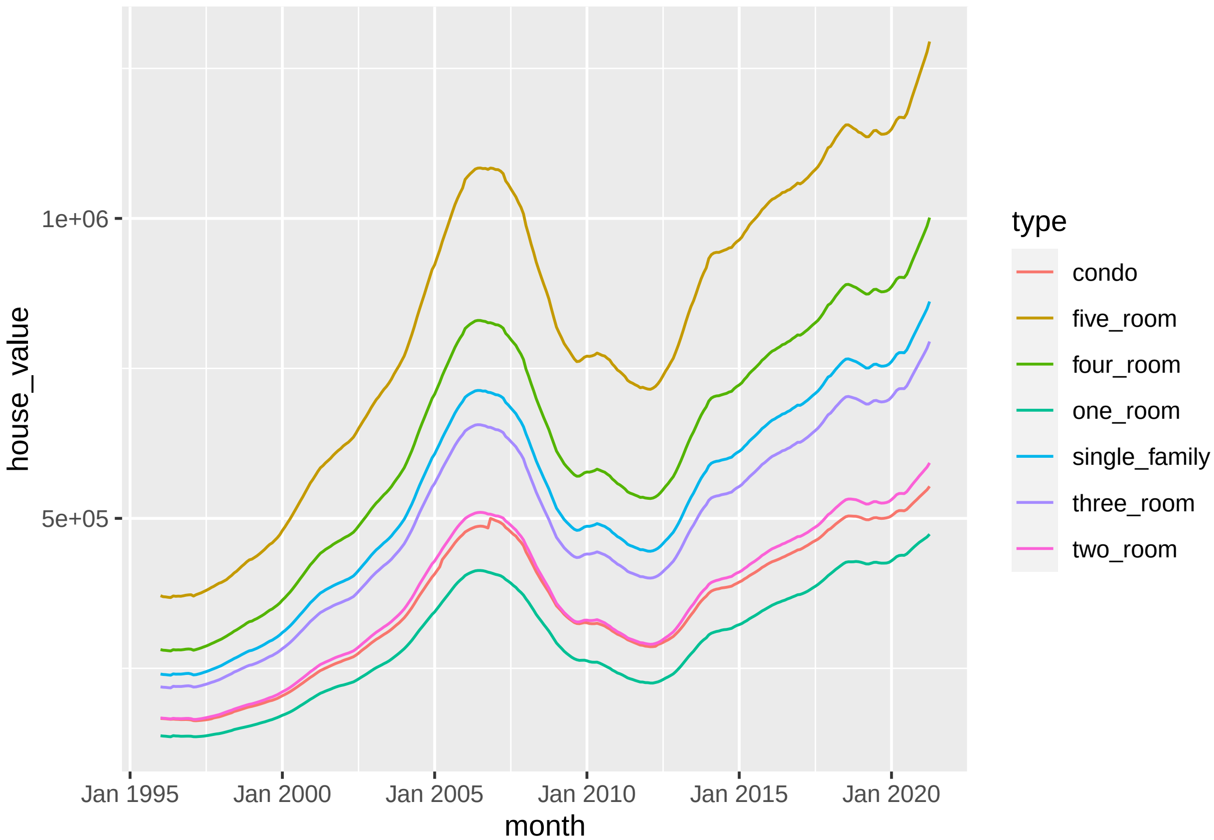
```
value_s_city <- value_s %>%
  mutate(year=year(month))%>%
  select(RegionName, date, month, year, values)

value_c_city <- value_c %>%
 mutate(year=year(month))%>%
  select(RegionName, date, month, year, valuec)

value_1_city <- value_1 %>%
  mutate(year=year(month))%>%
```

```
  select(RegionName, date, month, year, room1)

value_2_city <- value_2 %>%
 mutate(year=year(month))%>%
  select(RegionName, date, month, year, room2)

value_3_city <- value_3 %>%
  mutate(year=year(month))%>%
  select(RegionName, date, month, year, room3)

value_4_city <- value_4 %>%
  mutate(year=year(month))%>%
  select(RegionName, date, month, year, room4)

value_5_city <- value_5 %>%
 mutate(year=year(month))%>%
  select(RegionName, date, month, year, room5)

head(value_s_city, n = 2)
```

```
## # A tibble: 2 x 5
##   RegionName  date       month      year values
##   <chr>       <chr>      <yearmon> <int>  <dbl>
## 1 Los Angeles 1996-01-31 Jan 1996   1996 196175
## 2 Los Angeles 1996-02-29 Feb 1996   1996 196220
```

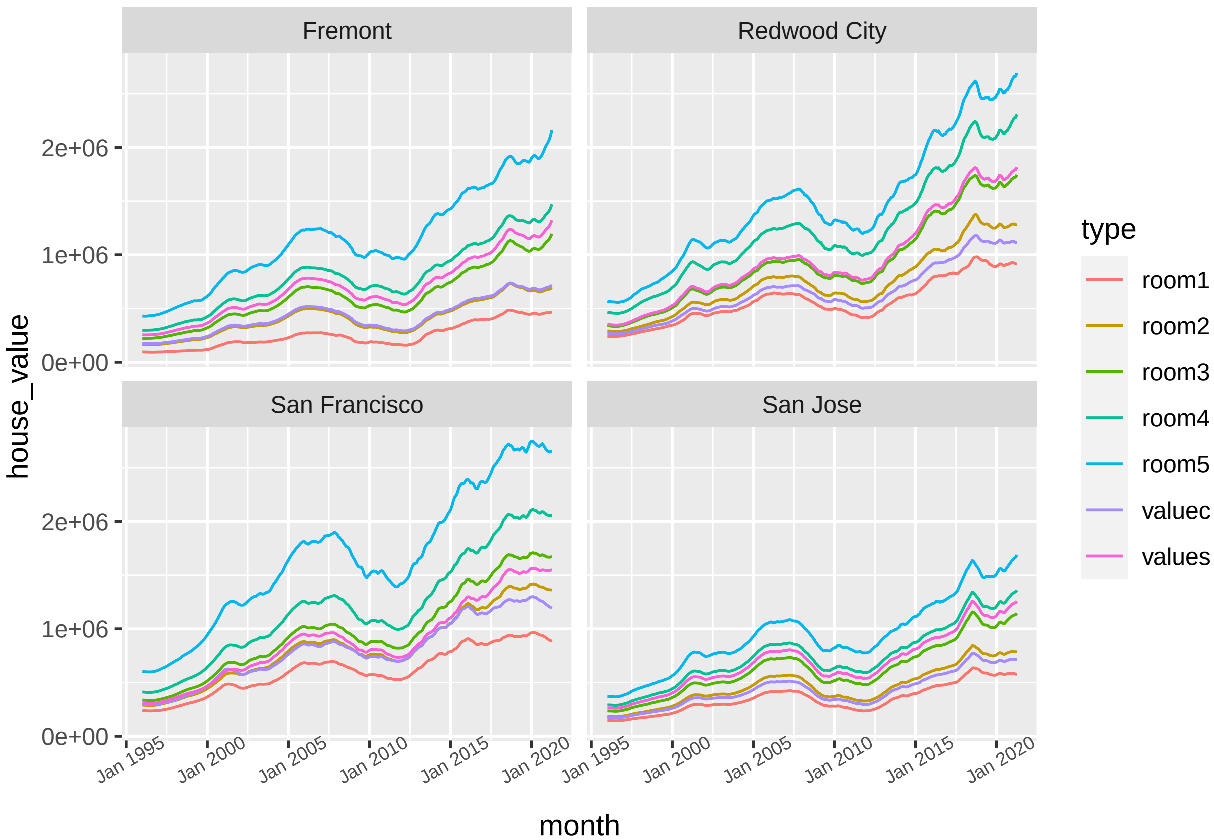
Using *inner_join()* function to join the different type of house value data sets together. Making the time series for different type of house value (on average) in California. The plot shows they have similar trends overall.

```
value_city <- value_s_city %>%
  inner_join(value_c_city, by = c('RegionName', 'month', 'date')) %>%
  inner_join(value_1_city, by = c('RegionName', 'month', 'date')) %>%
  inner_join(value_2_city, by = c('RegionName', 'month', 'date')) %>%
  inner_join(value_3_city, by = c('RegionName', 'month', 'date')) %>%
  inner_join(value_4_city, by = c('RegionName', 'month', 'date')) %>%
  inner_join(value_5_city, by = c('RegionName', 'month', 'date'))

value_city0 <-
  value_city %>%
  select(month, values, valuec, room1, room2, room3, room4, room5) %>%
  group_by(month) %>%
  summarize(single_family=mean(values), condo=mean(valuec),
            one_room = mean(room1), two_room = mean(room2),
            three_room = mean(room3), four_room = mean(room4),
            five_room = mean(room5))

value_city0 %>%
  pivot_longer(-1, names_to = 'type', values_to = 'house_value') %>%
  ggplot(aes(x = month, y = house_value, col = type)) +
  geom_line()
```



Making the time series for different type of house value (on average) for four cities. The plot shows the different types of house value have similar trends for each city.

```
city2 <- c('San Francisco', 'San Jose', 'Redwood City', 'Fremont')

value_city %>%
  filter(RegionName %in% city2) %>%
  select(RegionName, month, values, valuec, room1, room2, room3, room4, room5) %>%
  pivot_longer(-c(1:2), names_to = 'type', values_to = 'house_value') %>%
  ggplot(aes(x = month, y = house_value, col = type)) +
  geom_line() +
  facet_wrap(.~ RegionName, nrow = 2) +
  theme(axis.text.x = element_text(angle = 30, size = 7))
```



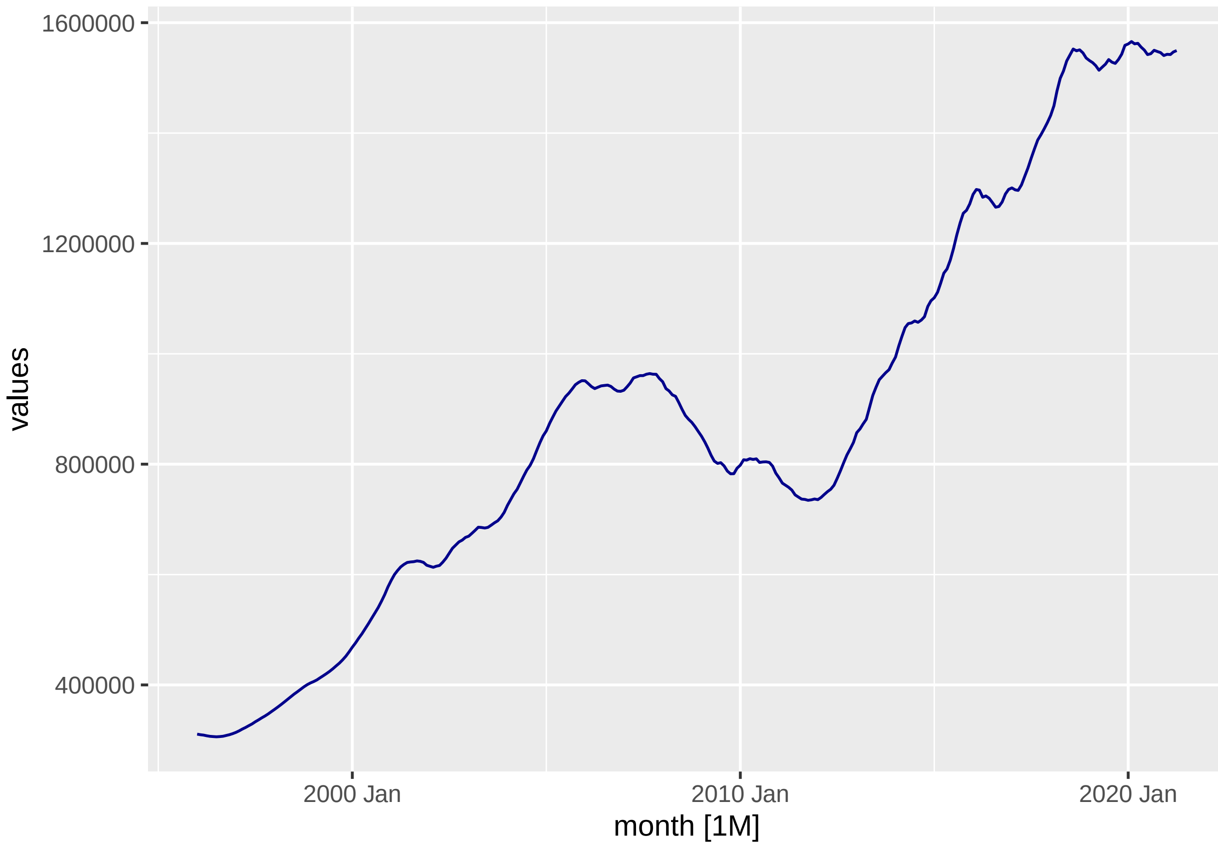
Forecast the single family house value of San Francisco in 3 years.

```
house_type <- 'values'
city <- 'San Francisco'

ts <- value_city %>%
  filter(RegionName == city) %>%
  select(month, house_type) %>%
  mutate(month = yearmonth(month)) %>%
  as_tsibble(index = month)
head(ts, n=3)
```

```
## # A tsibble: 3 x 2 [1M]
##      month values
##      <mth>  <dbl>
## 1 1996 Jan 310806
## 2 1996 Feb 309773
## 3 1996 Mar 309067
```

Time series of single family house value for San Francisco from 1996 January to 2021 April.

```
ts %>% autoplot(col = 'blue4')
```



Determining whether differencing is required using *unitroot_kpss()* test.

```
ts %>%
  features(values, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1      4.38        0.01
```

The p-value is less than 0.05, indicating that the null hypothesis is rejected. That is, the data are not stationary. We can difference the data, and apply the test again.

```
ts %>%
  mutate(diff_value = difference(values)) %>%
  features(diff_value, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1     0.241         0.1
```

Determining the appropriate *number* of first differences is carried out using the *unitroot_ndiffs()* feature.

```r
ts %>%
  features(values, unitroot_ndiffs)
```

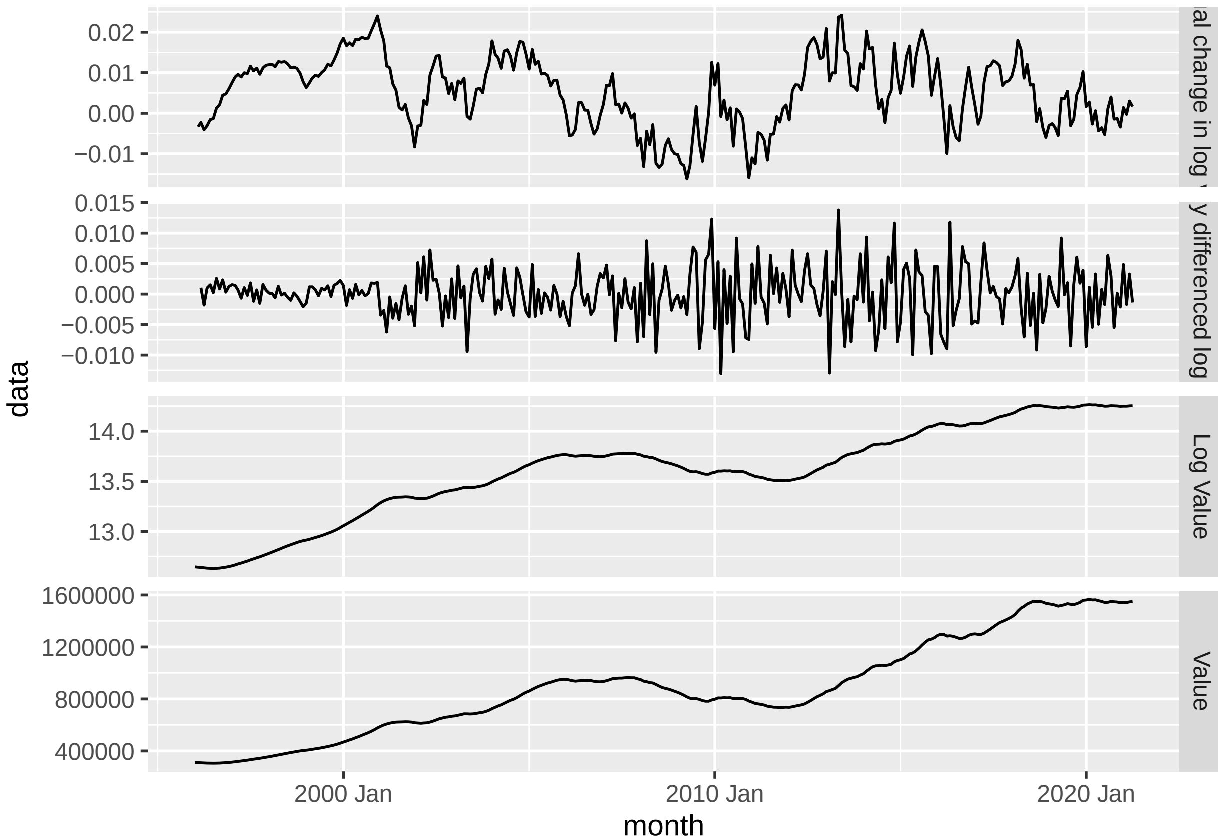
```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1
```

Determining whether seasonal differencing is required using *unitroot_nsdiffs()* function.

```r
ts %>%
  mutate(log_value = log(values)) %>%
  features(log_value, unitroot_nsdiffs)
```

```
## # A tibble: 1 x 1
##   nsdiffs
##     <int>
## 1       0
```

```r
ts %>%
  transmute(
    `Value` = values,
    `Log Value` = log(values),
    `Annual change in log value` = difference(log(values), 1),
    `Doubly differenced log value` =
                     difference(difference(log(values), 1), 1)) %>%
  pivot_longer(-month, names_to="data_type", values_to="data") %>%
  mutate(
    data_type = as.factor(data_type)) %>%
  ggplot(aes(x = month, y = data)) +
  geom_line() +
  facet_grid(vars(data_type), scales = "free_y")
```



Comparing ARIMA() and ETS() model.

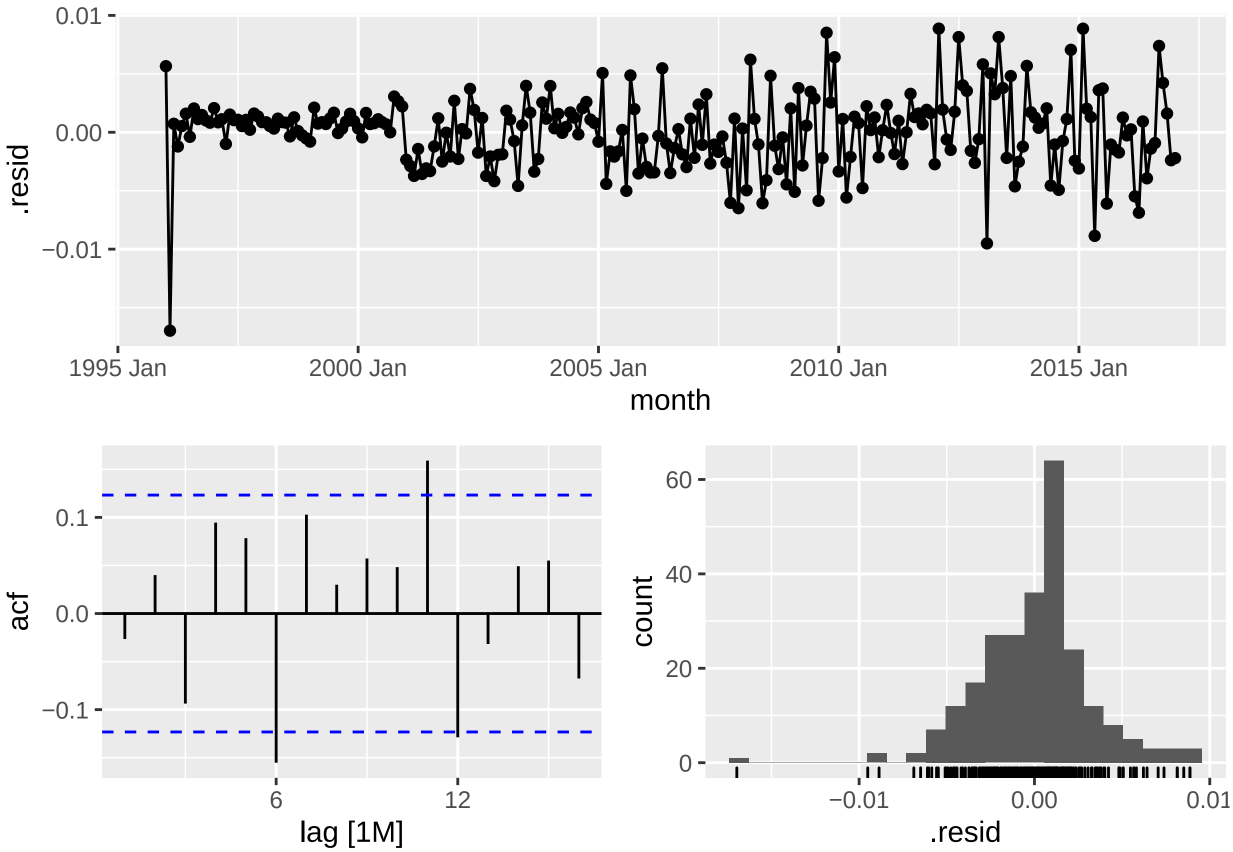
```
train <- ts %>%
  filter_index(. ~ "2016-12-31")
```

ARIMA()

```
fit_arima <- train %>% model(ARIMA(log(values)))
report(fit_arima)
```

```
## Series: values
## Model: ARIMA(3,2,0)(2,0,0)[12]
## Transformation: log(values)
##
## Coefficients:
##           ar1     ar2      ar3     sar1     sar2
##       -0.0047  0.1281  -0.3823  -0.7613  -0.4774
## s.e.   0.0590  0.0593   0.0589   0.0577   0.0581
##
## sigma^2 estimated as 1.055e-05:  log likelihood=1095.87
## AIC=-2179.75   AICc=-2179.4   BIC=-2158.6
```

```r
fit_arima %>% gg_tsresiduals(lag_max = 16)
```




```r
augment(fit_arima) %>%
  features(.innov, ljung_box, lag = 16, dof = 6)
```

```
## # A tibble: 1 x 3
##   .model              lb_stat lb_pvalue
##   <chr>                 <dbl>     <dbl>
## 1 ARIMA(log(values))     31.7  0.000443
```

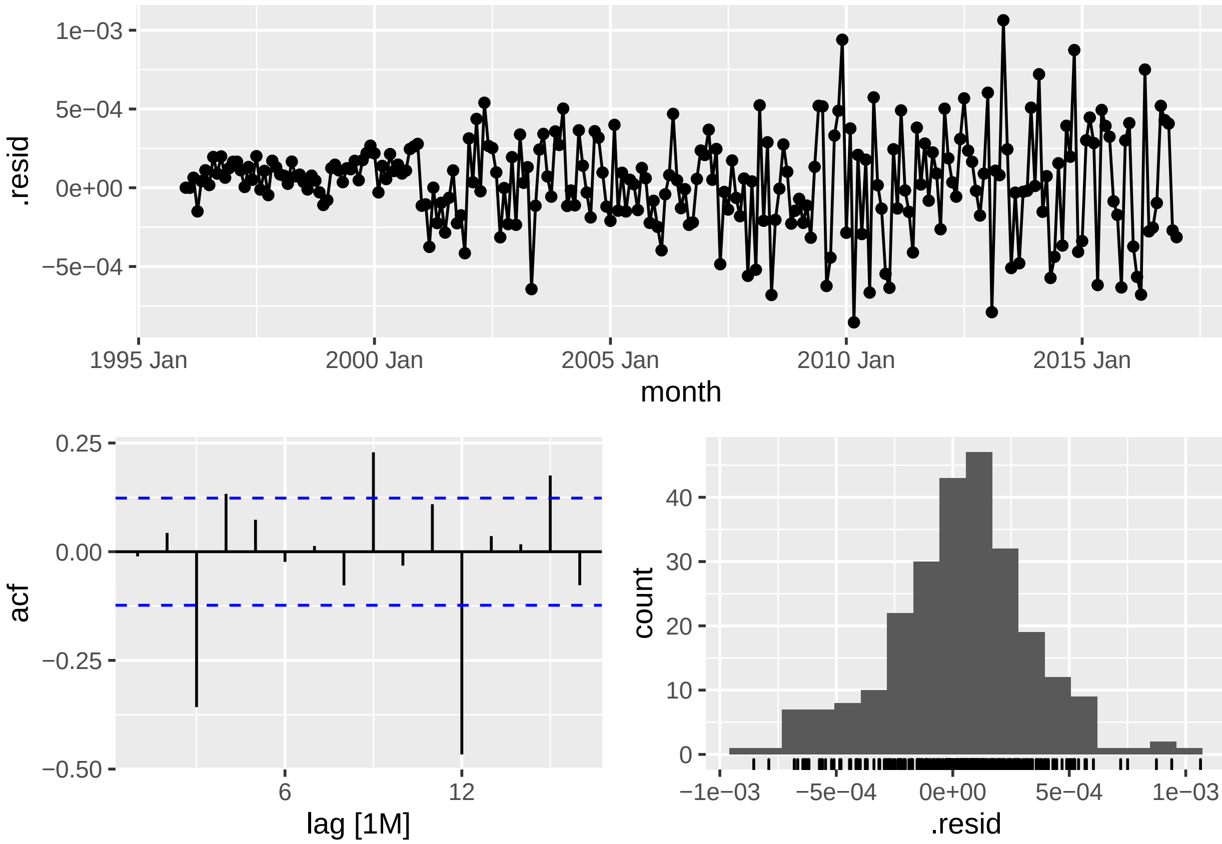
ETC()

```r
fit_ets <- train %>% model(ETS(log(values)))
report(fit_ets)
```

```
## Series: values
## Model: ETS(M,Ad,N)
## Transformation: log(values)
##   Smoothing parameters:
##     alpha = 0.9679686
##     beta  = 0.9458467
##     phi   = 0.9275335
##
##   Initial states:
```

```
##          l             b
##   12.65051 -0.003867577
##
##    sigma^2:  0
##
##        AIC       AICc        BIC
## -1362.309 -1361.968 -1341.109
```

```
fit_ets %>%
  gg_tsresiduals(lag_max = 16)
```




```
augment(fit_ets) %>%
  features(.innov, ljung_box, lag = 16, dof = 6)
```

```
## # A tibble: 1 x 3
##   .model           lb_stat lb_pvalue
##   <chr>              <dbl>     <dbl>
## 1 ETS(log(values))    127.         0
```

The output below evaluates the forecasting performance of the two competing models over the train and test set. The ARIMA model seems to be the slightly more accurate model based on the test set RMSE, MAPE and MASE.

```r
bind_rows(
    fit_arima %>% accuracy(),
    fit_ets %>% accuracy(),
    fit_arima %>% forecast(h = "3 years") %>%
      accuracy(ts),
    fit_ets %>% forecast(h = "3 years") %>%
      accuracy(ts)
  ) %>%
  select(-ME, -MPE, -ACF1)
```

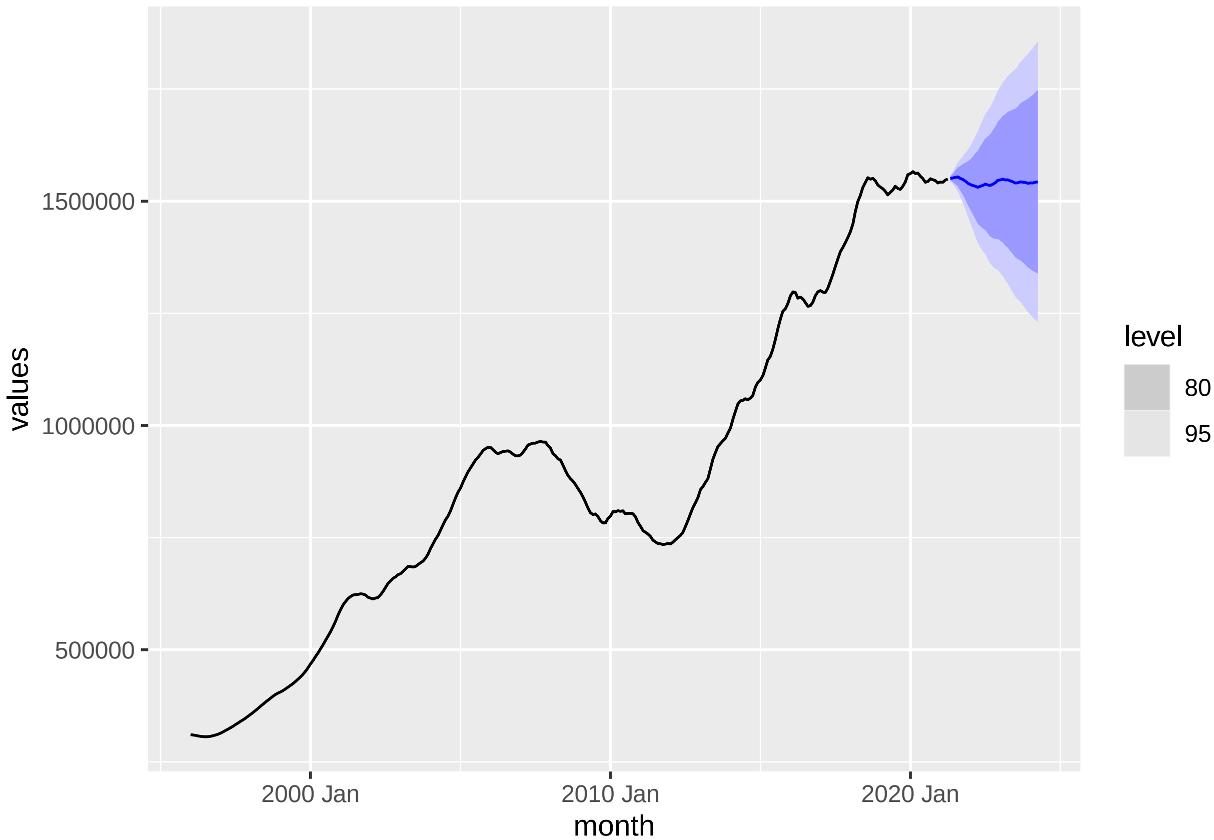
```
## # A tibble: 4 x 7
##   .model              .type       RMSE     MAE  MAPE   MASE  RMSSE
##   <chr>               <chr>      <dbl>   <dbl> <dbl>  <dbl>  <dbl>
## 1 ARIMA(log(values)) Training   2786.   1952. 0.236 0.0274 0.0325
## 2 ETS(log(values))   Training   3947.   2751. 0.321 0.0386 0.0460
## 3 ARIMA(log(values)) Test      82963.  60839. 4.03  0.854  0.968
## 4 ETS(log(values))   Test     149251. 133743. 8.84  1.88   1.74
```

Generating and ploting forecasts from the ARIMA model for the next 3 years.

```r
value_fc <- ts %>%
  model(ARIMA(values)) %>%
  forecast(h="3 years") %>%
  hilo(level = c(80, 95))
value_fc %>% head(n = 5)
```

```
## # A tsibble: 5 x 6 [1M]
## # Key:       .model [1]
##   .model    month              values  .mean                 `80%`
##   <chr>     <mth>              <dist>  <dbl>                <hilo>
## 1 ARIMA~ 2021 May   N(1550698, 1e+07) 1.55e6 [1546619, 1554777]80
## 2 ARIMA~ 2021 Jun N(1551676, 4.9e+07) 1.55e6 [1542704, 1560649]80
## 3 ARIMA~ 2021 Jul N(1553071, 1.4e+08) 1.55e6 [1537894, 1568249]80
## 4 ARIMA~ 2021 Aug N(1554121, 2.8e+08) 1.55e6 [1532842, 1575401]80
## 5 ARIMA~ 2021 Sep N(1550612, 4.7e+08) 1.55e6 [1522967, 1578258]80
## # ... with 1 more variable: `95%` <hilo>
```

```r
ts %>%
  model(ARIMA(values)) %>%
  forecast(h="3 years") %>%
  autoplot(ts)
```



values
1500000
1000000
500000
2000 Jan
2010 Jan
2020 Jan
month
level
80
95