wine-price-time-series

February 19, 2022

1 Wine Price Time Series Analysis

In this notebook, I'm going to analyze wine prices using domestic wine production, exports, imports, average wine prices, and population datasets.

1.1 Contents

1.2 Setup

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import grangercausalitytests
from statsmodels.tsa.vector_ar.vecm import coint_johansen
from statsmodels.tsa.api import VAR
from statsmodels.stats.stattools import durbin_watson
```

[2]:
```
sns.set_palette('colorblind')
```

1.2.1 Import Data

[3]:
```
df = pd.read_excel('../data/master-data.xlsx')
df.head()
```

[3]:

	Unnamed: 0	month	population	price	price_adj	bulk	\
0	0	2000-01-31	281083000	5.458	5.299029	1.131505e+08	
1	1	2000-02-29	281299000	5.256	5.198813	7.179357e+07	
2	2	2000-03-31	281531000	5.471	5.311650	4.635628e+07	
3	3	2000-04-30	281763000	5.156	5.104950	3.296724e+07	
4	4	2000-05-31	281996000	5.530	5.426889	3.178035e+07	

	bottled	cider	effervescent	wine_gross	…	\
0	1.244070e+08	NaN	6.909175e+06	2.444667e+08	…	
1	1.375283e+08	NaN	4.377026e+06	2.136989e+08	…	
2	1.603837e+08	NaN	9.321474e+06	2.160614e+08	…	
3	1.423004e+08	2.045088e+06	7.881046e+06	1.811036e+08	…	
4	1.612658e+08	6.959646e+06	6.334834e+06	1.924214e+08	…	

	landed_duty_paid_value_ukfrspde_imports	\
0	59812261	
1	77087577	
2	87219165	
3	87040067	
4	79534639	

	landed_duty_paid_value_adj_ukfrspde_imports	\
0	5.807016e+07	
1	7.624884e+07	
2	8.467880e+07	
3	8.617828e+07	
4	7.805166e+07	

	customs_value_ukfrspde_imports	customs_value_adj_ukfrspde_imports	\
0	56706027	5.505440e+07	
1	73873201	7.306944e+07	
2	83500840	8.106878e+07	
3	83075769	8.225324e+07	
4	75599523	7.418991e+07	

	quantity_ukfrspde_imports	charges_insurance_freight_ukfrspde_imports	\
0	8768676	2406931	
1	8961916	2356486	
2	10474993	2804317	
3	11128077	2989501	
4	10874051	3037785	

	charges_insurance_freight_adj_ukfrspde_imports	\
0	2.336826e+06	
1	2.330847e+06	
2	2.722638e+06	
3	2.959902e+06	

```
4                                    2.981143e+06

   calculated_duties_ukfrspde_imports  calculated_duties_adj_ukfrspde_imports  \
0                              699303                             678934.951456
1                              857890                             848555.885262
2                              914008                             887386.407767
3                              974797                             965145.544554
4                              897331                             880599.607458

   frspger_25
0           0
1           0
2           0
3           0
4           0

[5 rows x 40 columns]
```

[4]:
```
df.set_index('month', inplace=True)
df.drop(columns=['Unnamed: 0'], axis=1, inplace=True)

df.head()
```

[4]:
```
            population  price  price_adj          bulk       bottled  \
month
2000-01-31   281083000  5.458   5.299029  1.131505e+08  1.244070e+08
2000-02-29   281299000  5.256   5.198813  7.179357e+07  1.375283e+08
2000-03-31   281531000  5.471   5.311650  4.635628e+07  1.603837e+08
2000-04-30   281763000  5.156   5.104950  3.296724e+07  1.423004e+08
2000-05-31   281996000  5.530   5.426889  3.178035e+07  1.612658e+08

                  cider  effervescent    wine_gross  bulk_adj  bottled_adj  \
month
2000-01-31          NaN  6.909175e+06  2.444667e+08  0.402552     0.442599
2000-02-29          NaN  4.377026e+06  2.136989e+08  0.255222     0.488904
2000-03-31          NaN  9.321474e+06  2.160614e+08  0.164658     0.569684
2000-04-30  2.045088e+06  7.881046e+06  1.811036e+08  0.117003     0.505036
2000-05-31  6.959646e+06  6.334834e+06  1.924214e+08  0.112698     0.571873

            …  landed_duty_paid_value_ukfrspde_imports  \
month       …
2000-01-31  …                                 59812261
2000-02-29  …                                 77087577
2000-03-31  …                                 87219165
2000-04-30  …                                 87040067
2000-05-31  …                                 79534639
```

```
            landed_duty_paid_value_adj_ukfrspde_imports  \
month
2000-01-31                                   5.807016e+07
2000-02-29                                   7.624884e+07
2000-03-31                                   8.467880e+07
2000-04-30                                   8.617828e+07
2000-05-31                                   7.805166e+07

            customs_value_ukfrspde_imports  \
month
2000-01-31                         56706027
2000-02-29                         73873201
2000-03-31                         83500840
2000-04-30                         83075769
2000-05-31                         75599523

            customs_value_adj_ukfrspde_imports  quantity_ukfrspde_imports  \
month
2000-01-31                         5.505440e+07                     8768676
2000-02-29                         7.306944e+07                     8961916
2000-03-31                         8.106878e+07                    10474993
2000-04-30                         8.225324e+07                    11128077
2000-05-31                         7.418991e+07                    10874051

            charges_insurance_freight_ukfrspde_imports  \
month
2000-01-31                                     2406931
2000-02-29                                     2356486
2000-03-31                                     2804317
2000-04-30                                     2989501
2000-05-31                                     3037785

            charges_insurance_freight_adj_ukfrspde_imports  \
month
2000-01-31                                    2.336826e+06
2000-02-29                                    2.330847e+06
2000-03-31                                    2.722638e+06
2000-04-30                                    2.959902e+06
2000-05-31                                    2.981143e+06

            calculated_duties_ukfrspde_imports  \
month
2000-01-31                              699303
2000-02-29                              857890
2000-03-31                              914008
2000-04-30                              974797
2000-05-31                              897331
```

```
            calculated_duties_adj_ukfrspde_imports  frspger_25
month
2000-01-31                              678934.951456           0
2000-02-29                              848555.885262           0
2000-03-31                              887386.407767           0
2000-04-30                              965145.544554           0
2000-05-31                              880599.607458           0

[5 rows x 38 columns]
```

1.3 Granger Causality Tests

[5]:
```
maxlag=16
test = 'ssr_chi2test'

def grangers_causation_matrix(data, variables, test=test, verbose=False):
    """Check Granger Causality of all possible combinations of the Time series.
    The rows are the response variable, columns are predictors. The values in␣
 ↪the table
    are the P-Values. P-Values lesser than the significance level (0.05),␣
 ↪implies
    the Null Hypothesis that the coefficients of the corresponding past values␣
 ↪is
    zero, that is, the X does not cause Y can be rejected.

    data      : pandas dataframe containing the time series variables
    variables : list containing names of the time series variables.
    """
    granger_df = pd.DataFrame(np.zeros((len(variables), len(variables))),␣
 ↪columns=variables, index=variables)
    for c in granger_df.columns:
        for r in granger_df.index:
            test_result = grangercausalitytests(data[[r, c]], maxlag=16,␣
 ↪verbose=False)
            p_values = [round(test_result[i+1][0][test][1],4) for i in␣
 ↪range(maxlag)]
            if verbose: print(f'Y = {r}, X = {c}, P Values = {p_values}')
            min_p_value = np.min(p_values)
            granger_df.loc[r, c] = min_p_value
    granger_df.columns = [var + '_x' for var in variables]
    granger_df.index = [var + '_y' for var in variables]

    return granger_df
```

[6]:
```
granger_results = grangers_causation_matrix(data=df.dropna(), variables=df.
 ↪columns)
granger_results.to_excel('../granger_results_master_df.xlsx')
```

1.4 Stationary Transformation

Before getting started, let's define the Dickey-Fuller test for checking for stationarity.

The hypothesis for the Augmented Dickey-Fuller test is as follows:

$$h_0 : \text{The series has a unit root}$$

$$h_1 : \text{The series does not have a unit root}$$

[7]:
```
def adf(col):
    print('Augmented Dickey-Fuller Test:')
    unit_root_test = adfuller(col, autolag='AIC')
    dfoutput = pd.Series(unit_root_test[0:4], index=['t-stat:','p-value:','lags:
 ↪','observations:'])
    for key, value in unit_root_test[4].items():
       dfoutput['critical value (%s):' % key] = value
    print (dfoutput)
```

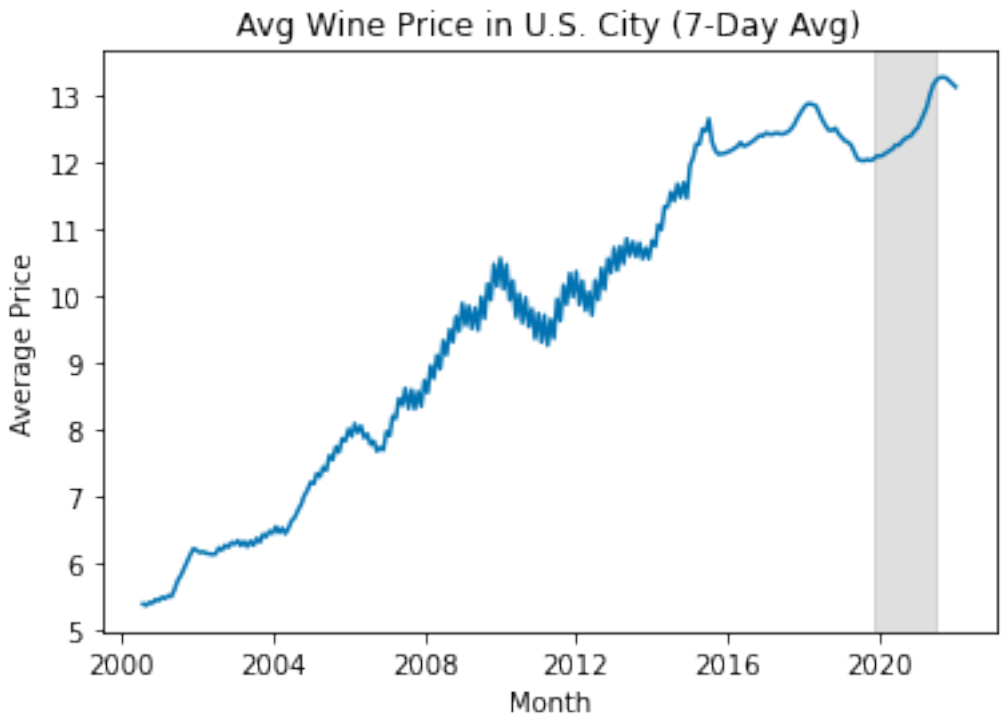
Let's also create a dataframe that contains all of the inputs to our multivariate timeseries model.

[8]:
```
ts_df = df[['frspger_25']].copy()
```

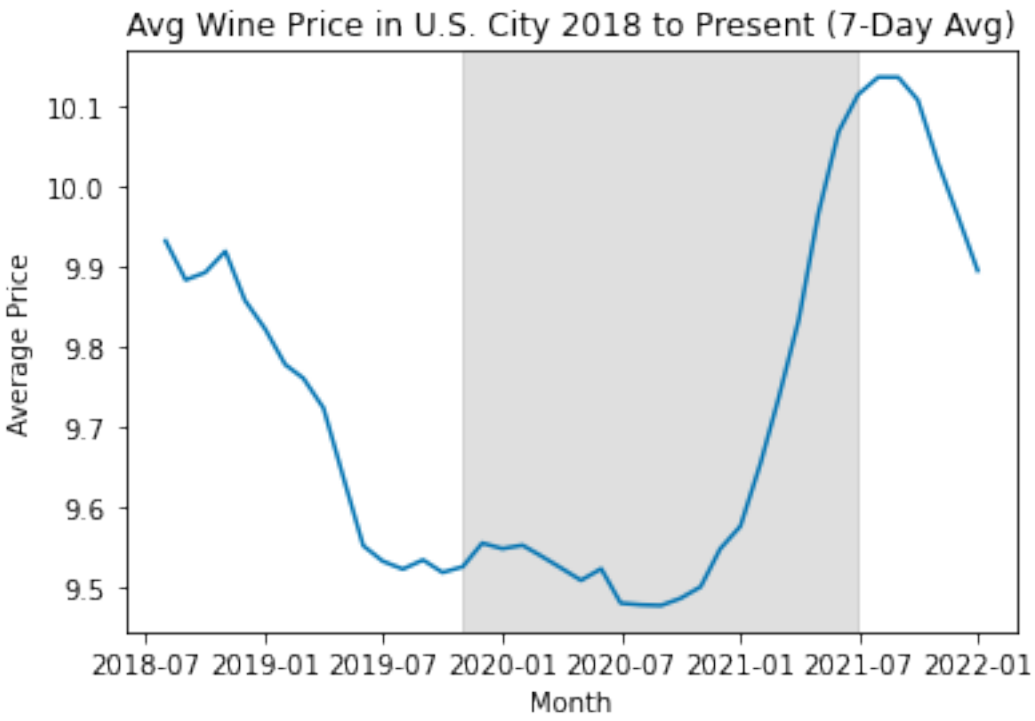
1.4.1 Price

Let's start with the price data.

[9]:
```
# Convert the additional 25% tariff indicator to a boolean
df['frspger_25'] = df['frspger_25'].astype('bool')

price_line_plot = sns.lineplot(data=df['price'].rolling(7).mean())
price_line_plot.set(title='Avg Wine Price in U.S. City (7-Day Avg)',␣
 ↪ylabel='Average Price', xlabel='Month')
# shade in the timespan of the additional tariff
price_line_plot.axvspan(
    xmin=df['frspger_25'].where(df['frspger_25']).first_valid_index(),
    xmax=df['frspger_25'].where(df['frspger_25']).last_valid_index(),
    color='gray',
    alpha=0.25
)
plt.show()
```



[10]:
```
price_line_2018_plot = sns.lineplot(data=df.loc['2018-01-01':
 ↪'2021-12-31']['price_adj'].rolling(7).mean())
price_line_2018_plot.set(title='Avg Wine Price in U.S. City 2018 to Present␣
 ↪(7-Day Avg)', ylabel='Average Price', xlabel='Month')
# shade in the timespan of the additional tariff
price_line_2018_plot.axvspan(
    xmin=df['frspger_25'].where(df['frspger_25']).first_valid_index(),
    xmax=df['frspger_25'].where(df['frspger_25']).last_valid_index(),
    color='gray',
    alpha=0.25
)
plt.show()
```



Alright, let's run the augmented Dickey-Fuller test.

[11]:
```
adf(df['price_adj'])
```

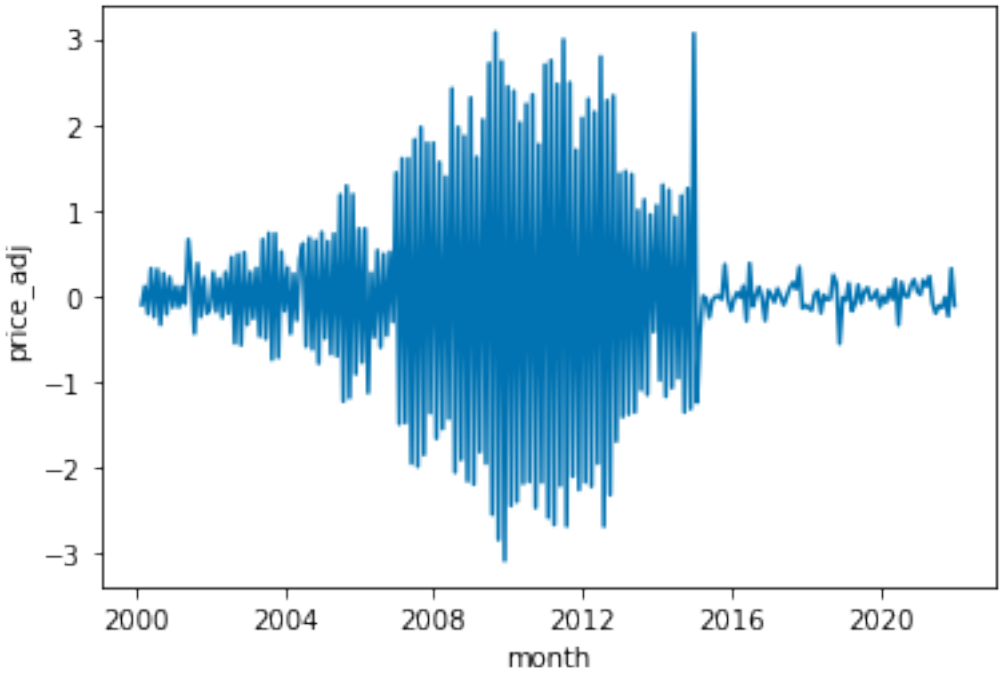
```
Augmented Dickey-Fuller Test:
t-stat:                    -1.863547
p-value:                    0.349399
lags:                       9.000000
observations:             254.000000
critical value (1%):       -3.456360
critical value (5%):       -2.872987
critical value (10%):      -2.572870
dtype: float64
```

Since the test statistic is greater than the critical value (at the 5% level), we fail to reject the null hypothesis; the series doesn't have a unit root and is therefore non-stationary.

So I do need to make some adjustments to the series to make it stationary.

[12]:
```
real_price_diff1 = df['price_adj'] - df['price_adj'].shift(1)

sns.lineplot(data=real_price_diff1)
```

[12]:
```
<AxesSubplot:xlabel='month', ylabel='price_adj'>
```



[13]:
```
adf(real_price_diff1.dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                 -8.068729e+00
p-value:                 1.570433e-12
lags:                    8.000000e+00
observations:            2.540000e+02
critical value (1%):    -3.456360e+00
critical value (5%):    -2.872987e+00
critical value (10%):   -2.572870e+00
dtype: float64
```

It looks like taking the first difference of the data worked to pass the Dickey-Fuller test.

[14]:
```
ts_df['price_adj_diff1'] = real_price_diff1
```

1.4.2 Domestic Wine Production

Bottled

[15]:
```
domestic_production_df = df[['bulk', 'bottled', 'cider', 'effervescent',␣
↪'population']].copy()
domestic_production_df.head()
```

[15]:

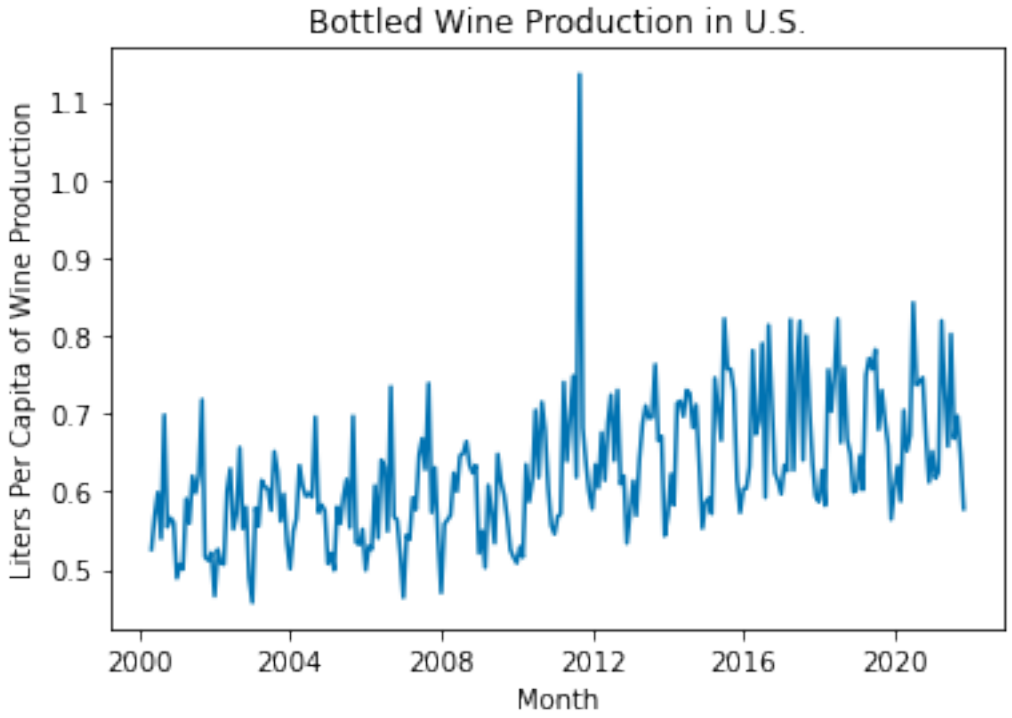
month	bulk	bottled	cider	effervescent	population
2000-01-31	1.131505e+08	1.244070e+08	NaN	6.909175e+06	281083000

2000-02-29	7.179357e+07	1.375283e+08	NaN	4.377026e+06	281299000
2000-03-31	4.635628e+07	1.603837e+08	NaN	9.321474e+06	281531000
2000-04-30	3.296724e+07	1.423004e+08	2.045088e+06	7.881046e+06	281763000
2000-05-31	3.178035e+07	1.612658e+08	6.959646e+06	6.334834e+06	281996000

I'm not interested in the distinction between carbonated wines and regular wine or cider and wine. The Alcohol and Tobacco Tax and Trade Bureau (TTB) data had Bottled wine production broken into Still Wine, Cider, and Effervescent. I'm going to subtract out the cider and add in the effervescent wine to the bottled category.

[16]:
```
domestic_production_df['bottled_total'] = domestic_production_df['bottled'] -␣
↪domestic_production_df['cider'] + domestic_production_df['effervescent']
```

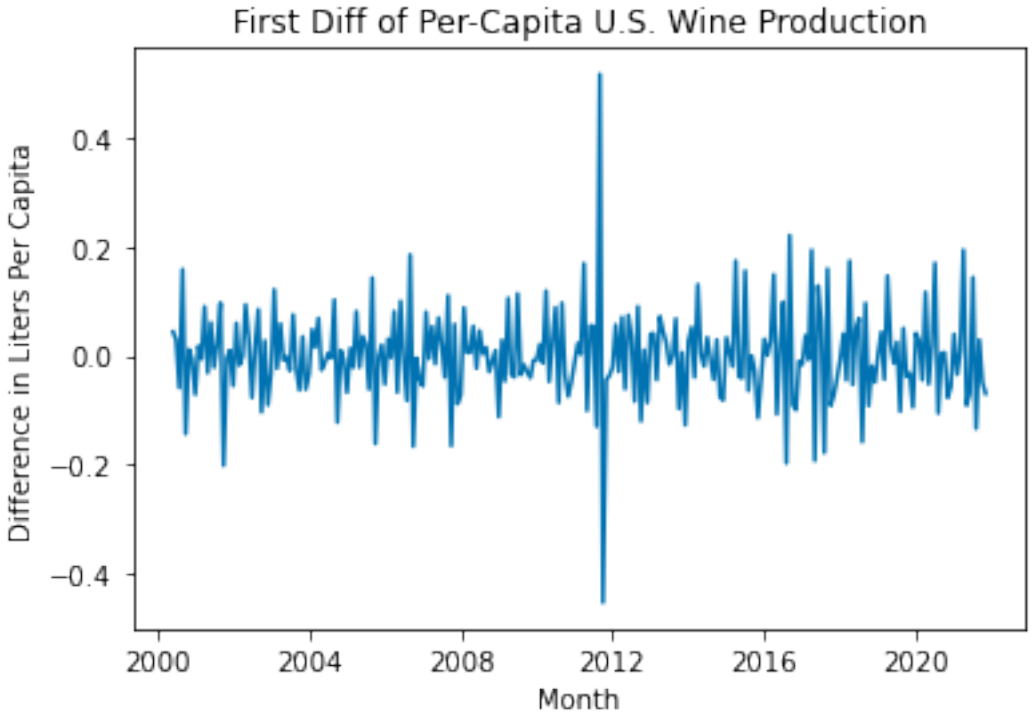
[17]:
```
for c in domestic_production_df.columns:
    if c != 'population':
        col_name = c + '_per_capita'
        domestic_production_df[col_name] = domestic_production_df[c] /␣
↪domestic_production_df['population']

bulk_bottled_lineplot = sns.
↪lineplot(data=domestic_production_df['bottled_total_per_capita'])
bulk_bottled_lineplot.set(title='Bottled Wine Production in U.S.',␣
↪xlabel='Month', ylabel='Liters Per Capita of Wine Production')
plt.show()
```



It looks like there's both an upward trend over time and some seasonality. Let's take a first diff and then run the ADF test. If it doesn't pass, let's look closer at the seasonality and then take another diff accordingly.

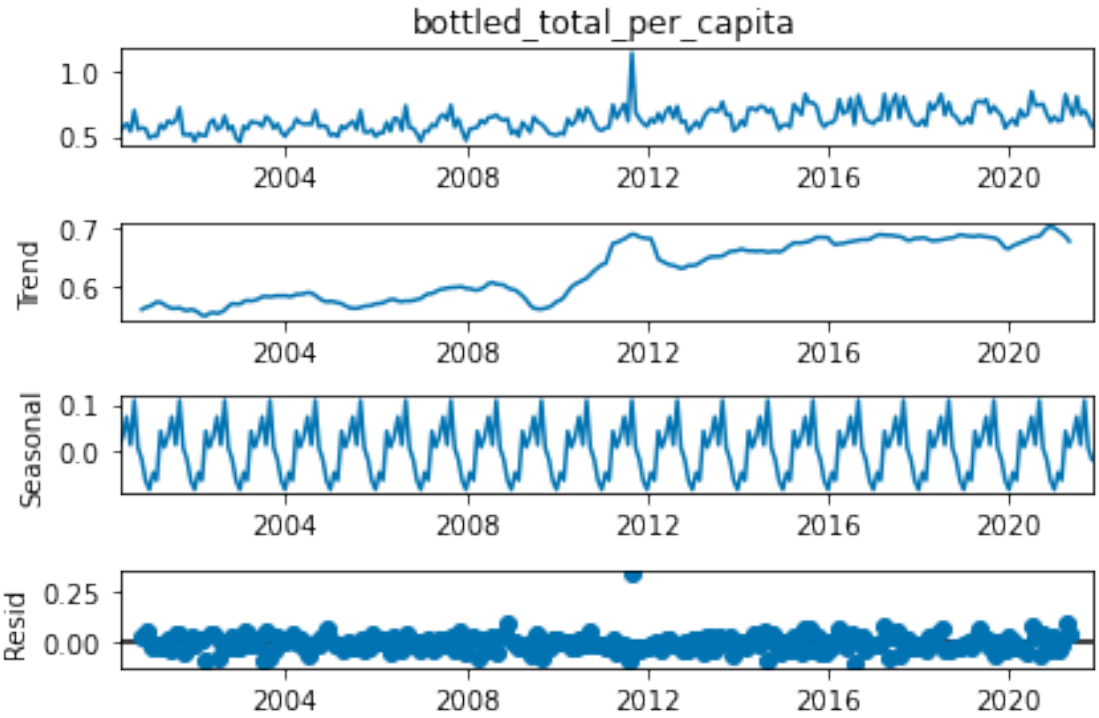
[18]:
```
domestic_production_df['bottled_total_per_capita_diff1'] =␣
 ↪domestic_production_df['bottled_total_per_capita'] -␣
 ↪domestic_production_df['bottled_total_per_capita'].shift(1)
wine_production_bottled_diff_plot = sns.
 ↪lineplot(data=domestic_production_df['bottled_total_per_capita_diff1'])
wine_production_bottled_diff_plot.set(title='First Diff of Per-Capita U.S. Wine␣
 ↪Production', xlabel='Month', ylabel='Difference in Liters Per Capita')

adf(domestic_production_df['bottled_total_per_capita_diff1'].dropna())
plt.show()
```

```
Augmented Dickey-Fuller Test:
t-stat:                    -4.794225
p-value:                    0.000056
lags:                      14.000000
observations:             243.000000
critical value (1%):       -3.457551
critical value (5%):       -2.873509
critical value (10%):      -2.573148
dtype: float64
```



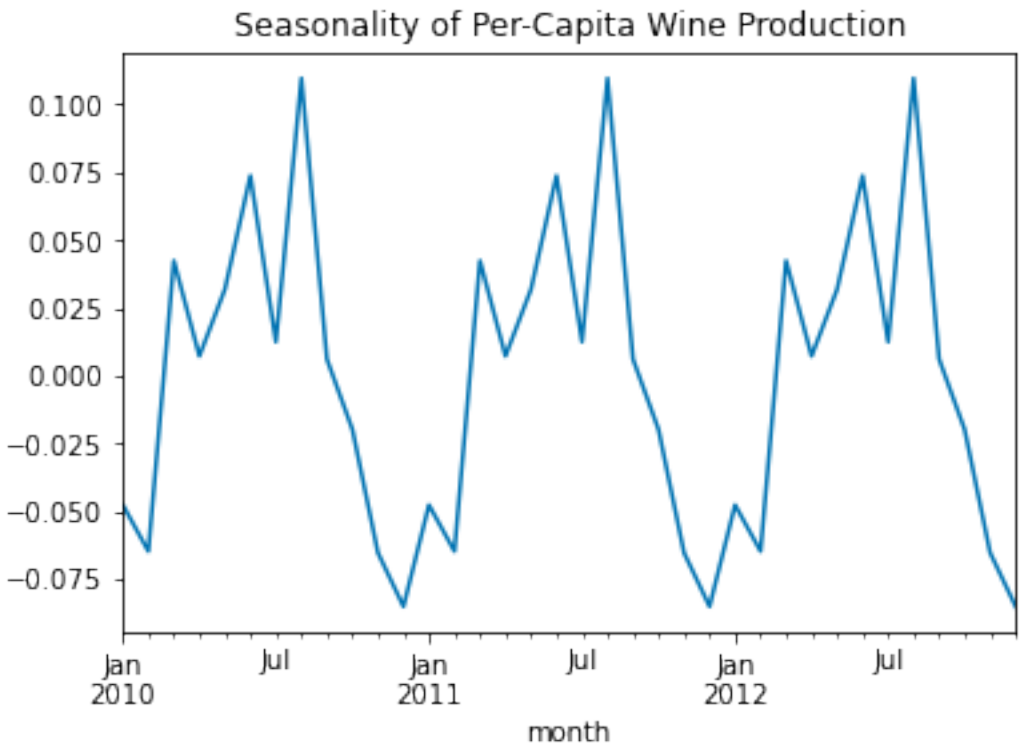
Alright, that failed the ADF test.

[19]:
```
decompose_result_mult =␣
↪seasonal_decompose(domestic_production_df['bottled_total_per_capita'].
↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot();
```



It definitely looks like there's seasonality in the data. Let's take a closer look at the seasonality to help identify the cycle.

[20]:
```
seasonality_bottled_plot = seasonal['2010-01-01':'2013-01-01'].plot()
seasonality_bottled_plot.set(title='Seasonality of Per-Capita Wine Production')
plt.show()
```



It looks like there's an annual cycle of wine production.

[21]:
```
domestic_production_df['bottled_total_per_capita_diff1_diff12'] =␣
↪domestic_production_df['bottled_total_per_capita_diff1'] -␣
↪domestic_production_df['bottled_total_per_capita_diff1'].shift(12)
```

[22]:
```
adf(domestic_production_df['bottled_total_per_capita_diff1_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                 -6.796768e+00
p-value:                 2.290410e-09
lags:                    1.500000e+01
observations:            2.300000e+02
critical value (1%):    -3.459106e+00
critical value (5%):    -2.874190e+00
critical value (10%):   -2.573512e+00
dtype: float64
```

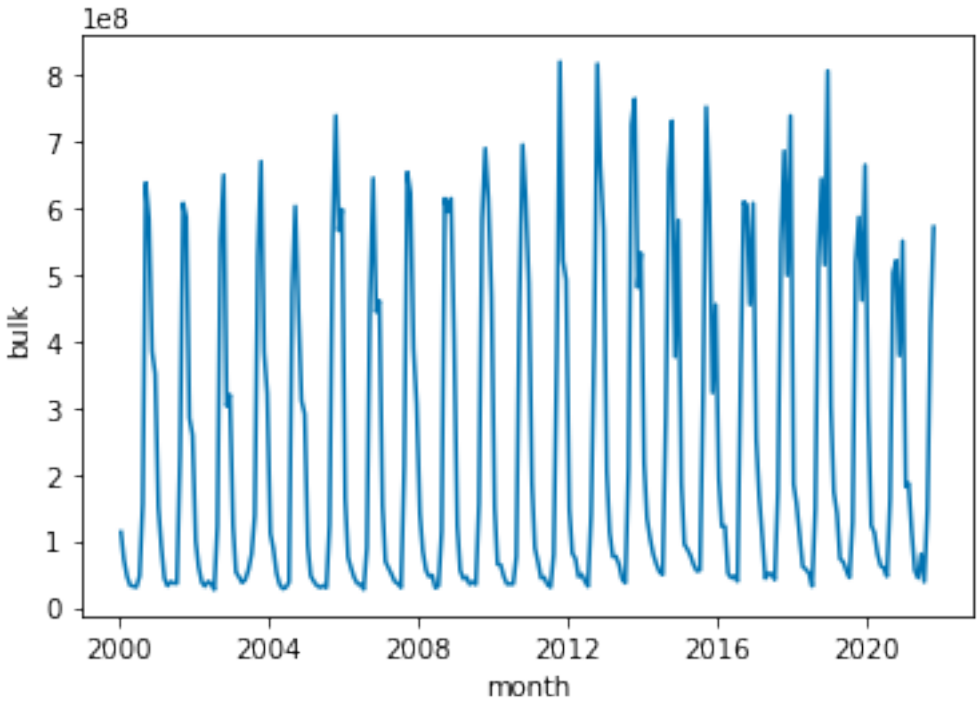
It looks like taking the second diff worked and we're now passing the ADF test.

[23]:
```
ts_df = ts_df.
↪merge(domestic_production_df['bottled_total_per_capita_diff1_diff12'],␣
↪left_index=True, right_index=True)
```

Bulk Wine

[24]:
```
sns.lineplot(data=domestic_production_df['bulk'])
```

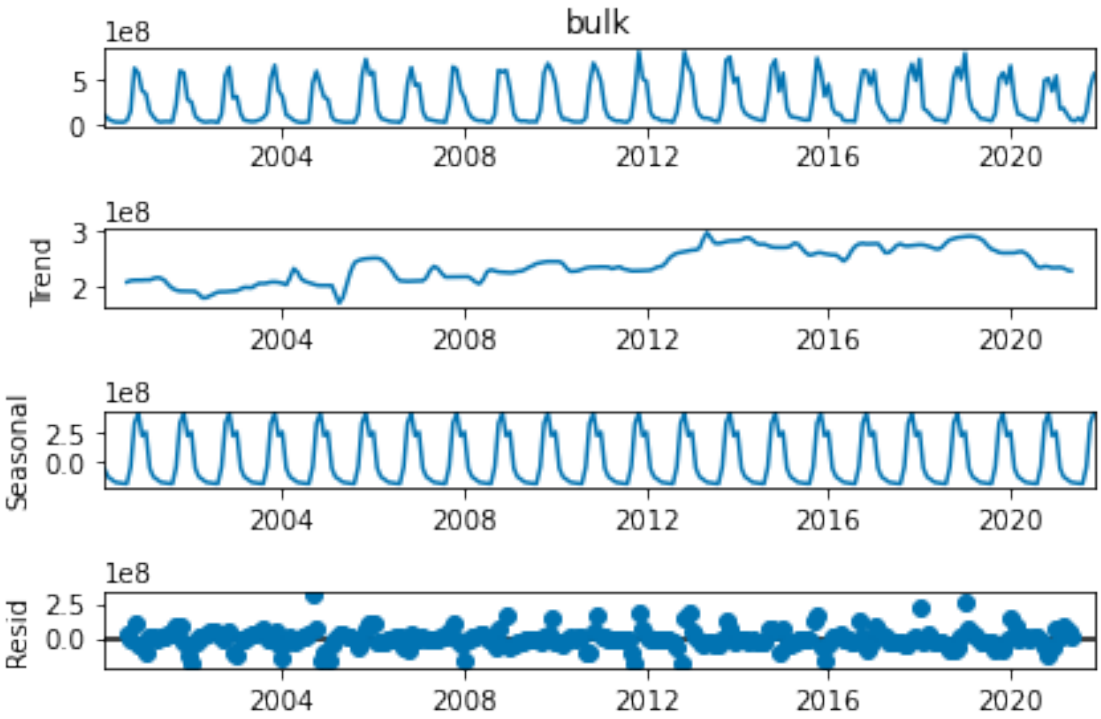
[24]:
```
<AxesSubplot:xlabel='month', ylabel='bulk'>
```

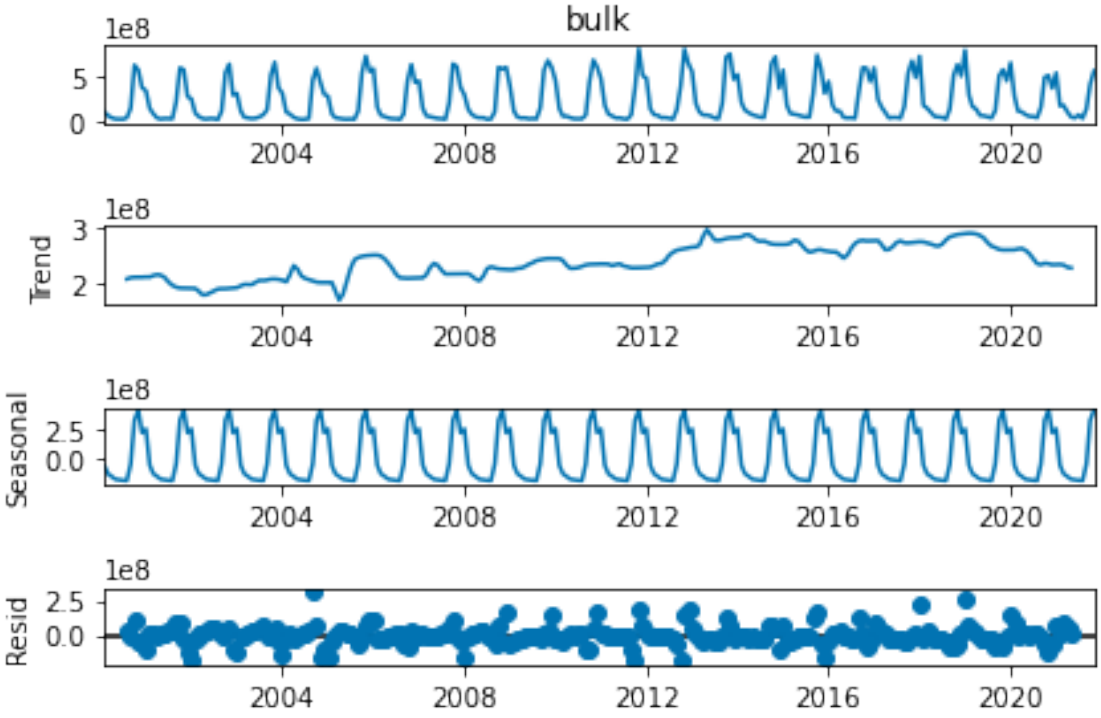


It doesn't really look like there's an upward trend in bulk wine production. But it does look like there's lots of seasonality.

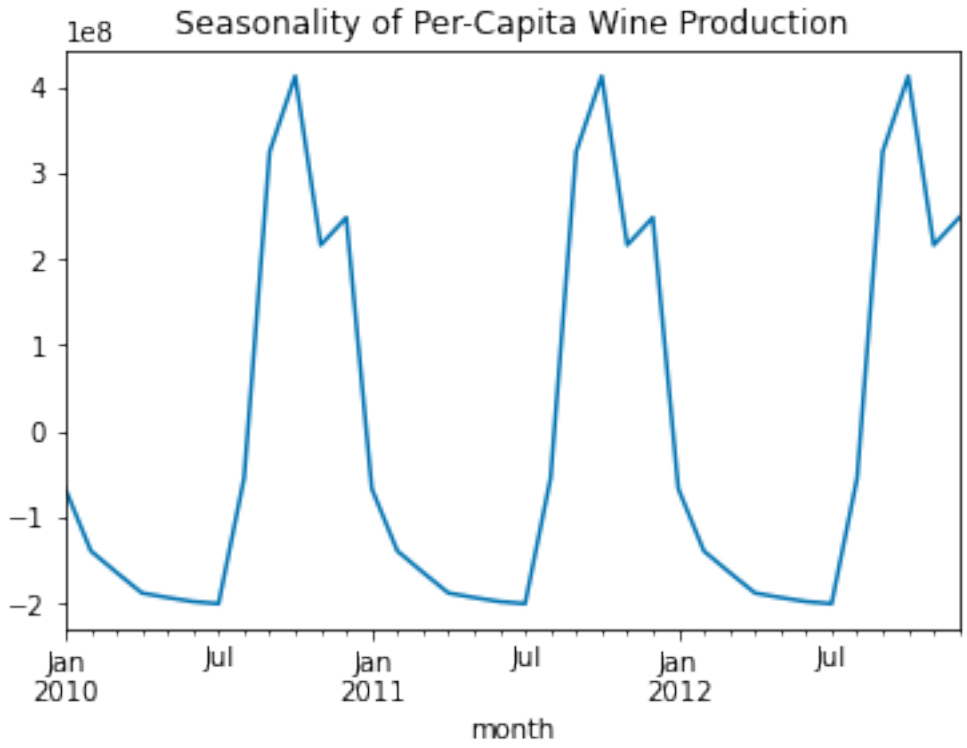
[25]:
```
decompose_result_mult = seasonal_decompose(domestic_production_df['bulk'].
↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[25]:


bulk
1e8
5
0
2004
2008
2012
2016
2020
Trend
3
2
Seasonal
2.5
0.0
Resid


bulk
1e8
5
0
2004
2008
2012
2016
2020
Trend
3
2
Seasonal
2.5
0.0
Resid

[26]:
```
seasonality_bulk_plot = seasonal['2010-01-01':'2013-01-01'].plot()
seasonality_bulk_plot.set(title='Seasonality of Per-Capita Wine Production')
plt.show()
```



[27]:
```
domestic_production_df['bulk_diff12'] = domestic_production_df['bulk'] -␣
↪domestic_production_df['bulk'].shift(12)
domestic_production_df['bulk_per_capita_diff12'] =␣
↪domestic_production_df['bulk_per_capita'] -␣
↪domestic_production_df['bulk_per_capita'].shift(12)
```

[28]:
```
adf(domestic_production_df['bulk_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                   -5.496346
p-value:                   0.000002
lags:                     12.000000
observations:            237.000000
critical value (1%):      -3.458247
critical value (5%):      -2.873814
critical value (10%):     -2.573311
dtype: float64
```

[29]:
```
adf(domestic_production_df['bulk_per_capita_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                  -5.662271e+00
p-value:                  9.321169e-07
lags:                     1.200000e+01
observations:             2.370000e+02
critical value (1%):     -3.458247e+00
critical value (5%):     -2.873814e+00
critical value (10%):    -2.573311e+00
dtype: float64
```

Correcting for the seasonality of bulk wine production allowed the data to pass the ADF test.

[30]:
```
ts_df = ts_df.merge(domestic_production_df[['bulk_diff12',␣
↪'bulk_per_capita_diff12']], left_index=True, right_index=True)
```

1.4.3 Wine Exports

Alright, let's look at the exports data. I think it'll be good to have a variable that's non-basic wine production (wine production that's used for some form of domestic consumption or input). So I'll start by defining a variable for non-basic wine quantity, `nonbasic_quantity`.

[31]:
```
df['nonbasic_quantity'] = df['bulk'] + df['bottled'] + df['effervescent'] -␣
↪df['cider'] - df['quantity_exports']
```

So we'll transform `nonbasic_quantity`, `quantity_exports`, and `fas_value_adj_exports` variables to be stationary. And we'll also create per-capita variables for those datapoints.

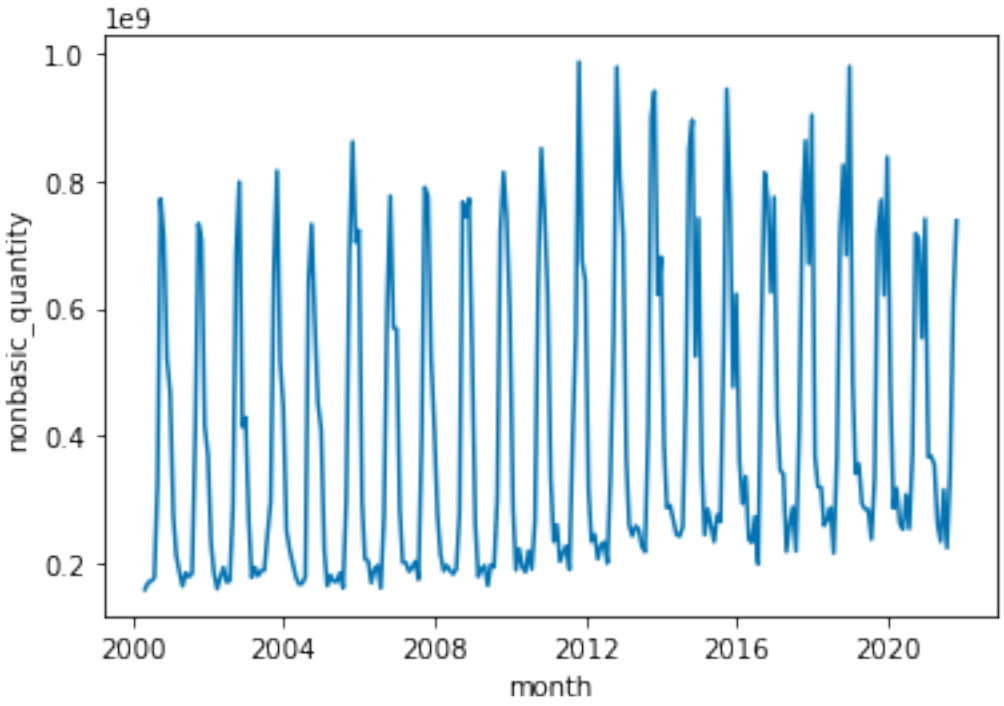
[32]:
```
exports_df = df[['nonbasic_quantity', 'quantity_exports',␣
↪'fas_value_adj_exports', 'population']].copy()
```

[33]:
```
for c in exports_df.columns:
    if c != 'population':
        col_name = c + '_per_capita'
        exports_df[col_name] = exports_df[c] / exports_df['population']
```

Non-Basic Wine Production

[34]:
```
sns.lineplot(data=exports_df['nonbasic_quantity'])
```

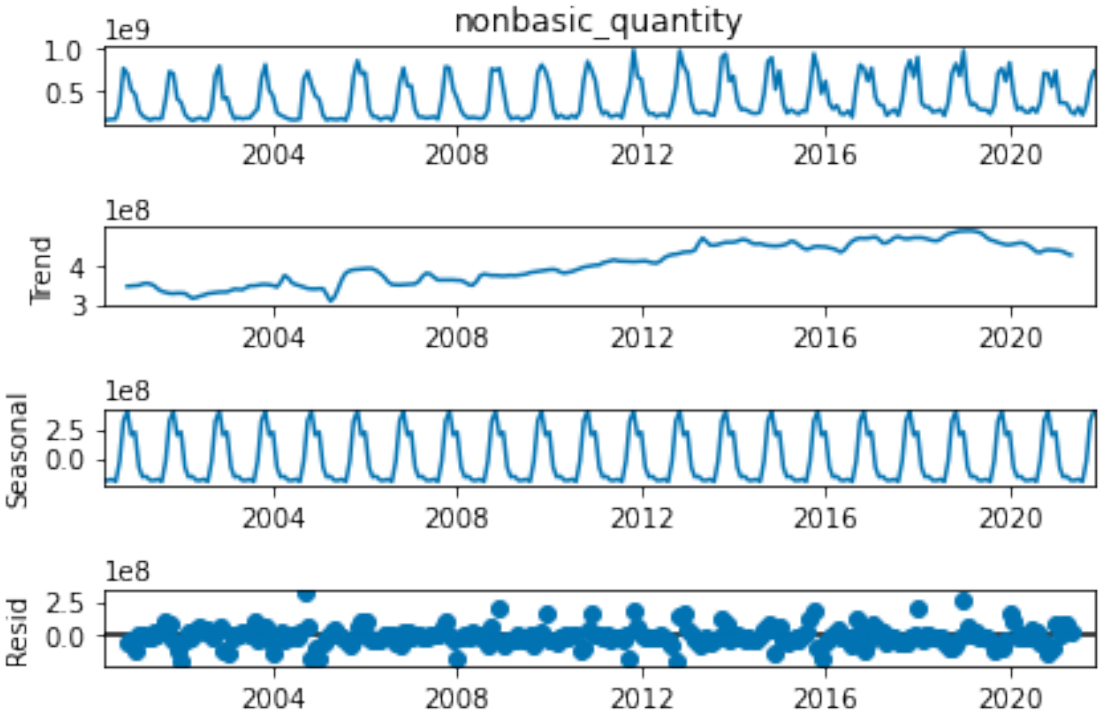
[34]:
```
<AxesSubplot:xlabel='month', ylabel='nonbasic_quantity'>
```

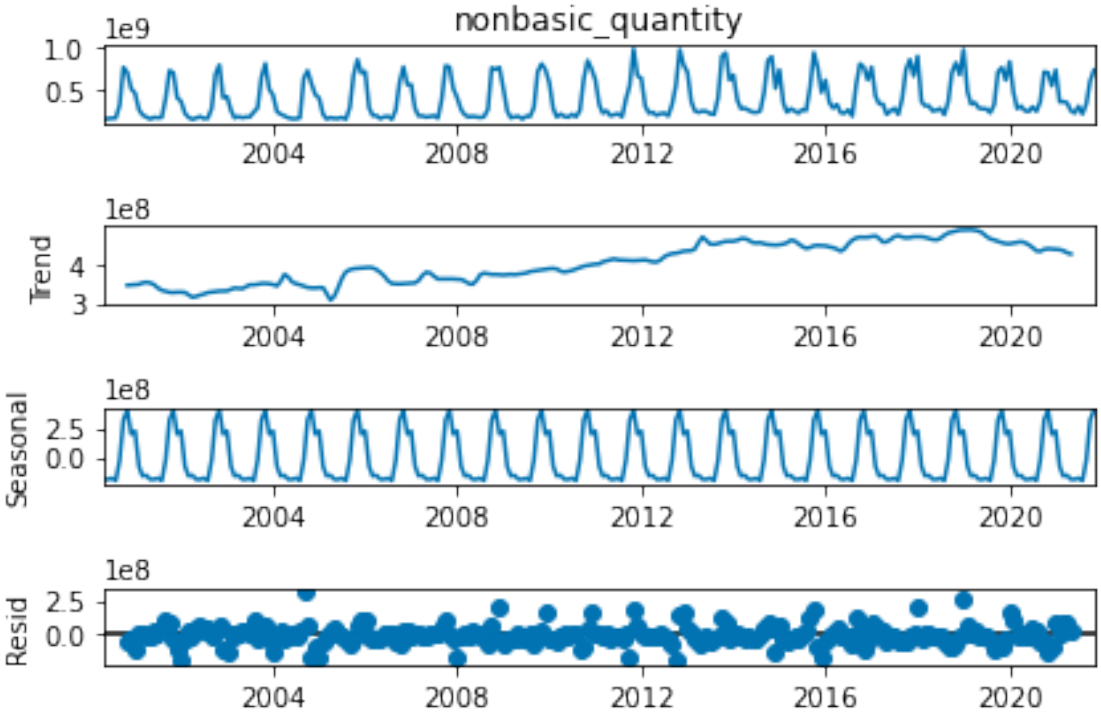


For non-basic wine in the U.S., we see both some seasonality and an upward trend. This makes sense since it includes both bottled and bulk wine production.

[35]:
```
decompose_result_mult = seasonal_decompose(exports_df['nonbasic_quantity'].
 ↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[35]:


nonbasic_quantity
1e9
1.0
0.5
2004
2008
2012
2016
2020
1e8
Trend
4
3
Seasonal
2.5
0.0
Resid


nonbasic_quantity
1e9
1.0
0.5
2004
2008
2012
2016
2020
1e8
Trend
4
3
Seasonal
2.5
0.0
Resid

[36]:
```
# diff non-basic quantity
exports_df['nonbasic_quantity_diff1'] = exports_df['nonbasic_quantity'] -␣
↪exports_df['nonbasic_quantity'].shift(1)
exports_df['nonbasic_quantity_diff1_diff12'] =␣
↪exports_df['nonbasic_quantity_diff1'] -␣
↪exports_df['nonbasic_quantity_diff1'].shift(12)
# diff non-basic quantity per-capita
exports_df['nonbasic_quantity_per_capita_diff1'] =␣
↪exports_df['nonbasic_quantity_per_capita'] -␣
↪exports_df['nonbasic_quantity_per_capita'].shift(1)
exports_df['nonbasic_quantity_per_capita_diff1_diff12'] =␣
↪exports_df['nonbasic_quantity_per_capita_diff1'] -␣
↪exports_df['nonbasic_quantity_per_capita_diff1'].shift(12)
```

[37]:
```
adf(exports_df['nonbasic_quantity_diff1_diff12'].dropna())
adf(exports_df['nonbasic_quantity_per_capita_diff1_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                  -7.302772e+00
p-value:                  1.325270e-10
lags:                     1.600000e+01
observations:             2.290000e+02
critical value (1%):     -3.459233e+00
critical value (5%):     -2.874245e+00
critical value (10%):    -2.573541e+00
dtype: float64
Augmented Dickey-Fuller Test:
t-stat:                  -7.348602e+00
p-value:                  1.019966e-10
lags:                     1.600000e+01
observations:             2.290000e+02
critical value (1%):     -3.459233e+00
critical value (5%):     -2.874245e+00
critical value (10%):    -2.573541e+00
dtype: float64
```

Alright, so the `nonbasic_quantity_*` variables are passing.

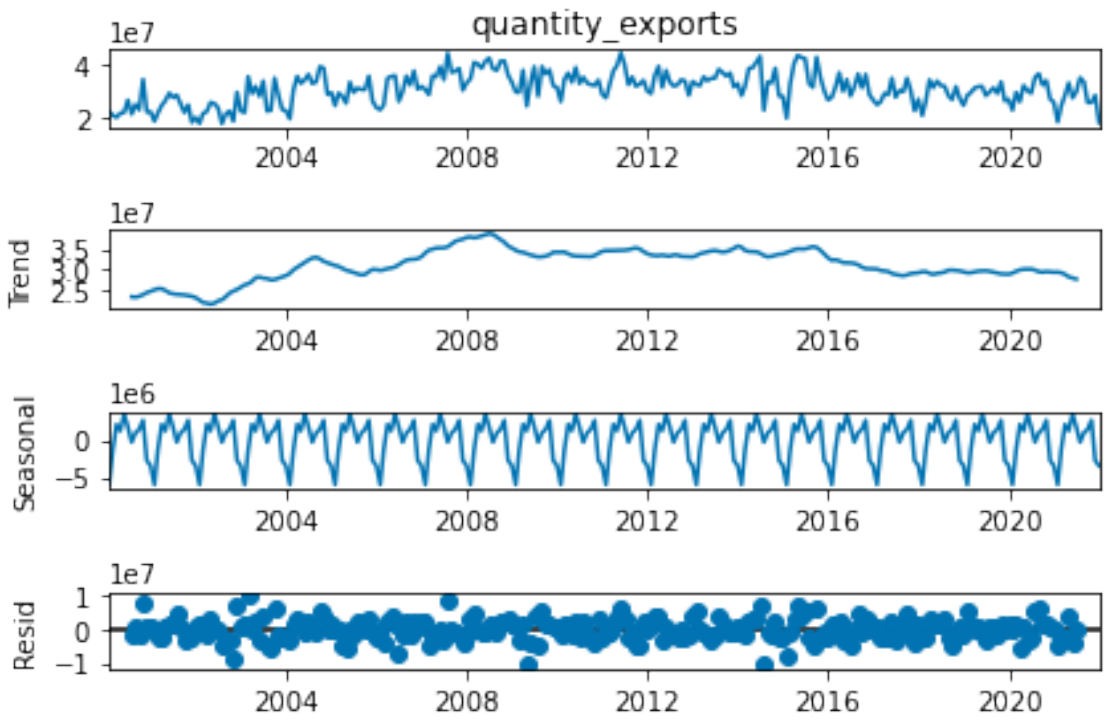
[38]:
```
ts_df = ts_df.merge(exports_df[['nonbasic_quantity_diff1_diff12',␣
↪'nonbasic_quantity_per_capita_diff1_diff12']], left_index=True,␣
↪right_index=True)
```

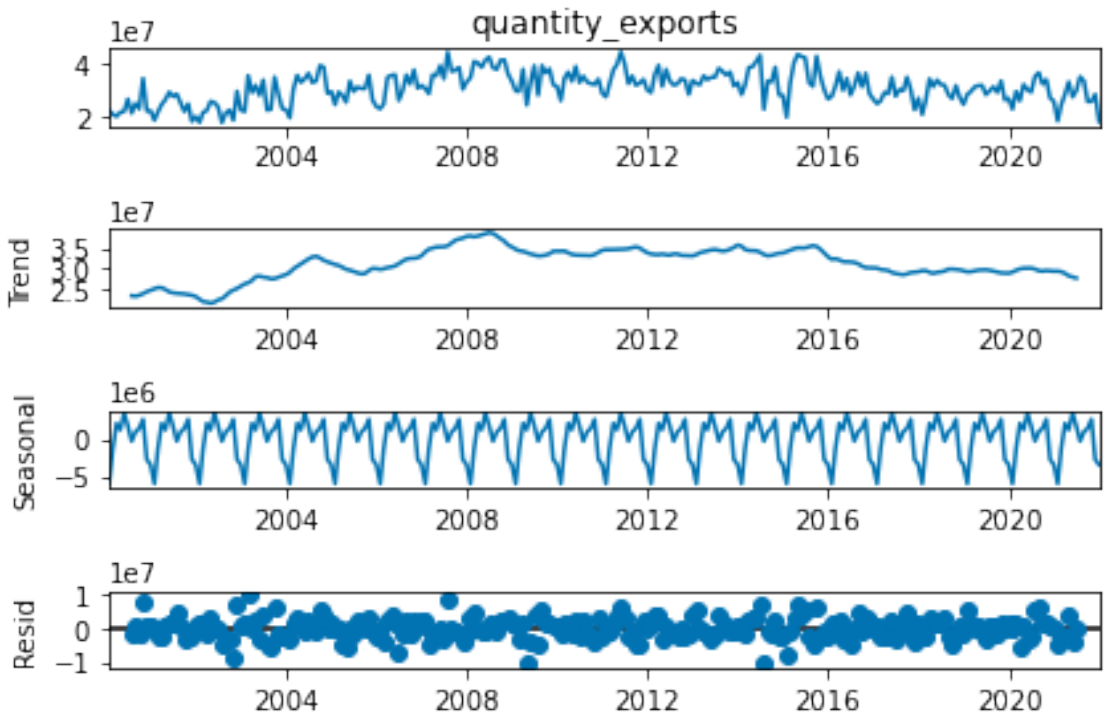
Exports Quantity

[39]:
```
decompose_result_mult = seasonal_decompose(exports_df['quantity_exports'].
↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
```

```
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[39]:





This looks like there may just be a seasonal trend in it.

[40]:
```
seasonality_bulk_plot = seasonal['2010-01-01':'2013-01-01'].plot()
seasonality_bulk_plot.set(title='Seasonality of Quantity of Wine Exports')
plt.show()
```

Let's do a 12-month diff and then run the ADF test.

[41]:
```
exports_df['quantity_exports_diff12'] = exports_df['quantity_exports'] -␣
↪exports_df['quantity_exports'].shift(12)
adf(exports_df['quantity_exports_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                    -4.698711
p-value:                    0.000085
lags:                      12.000000
observations:             239.000000
critical value (1%):       -3.458011
critical value (5%):       -2.873710
critical value (10%):      -2.573256
dtype: float64
```

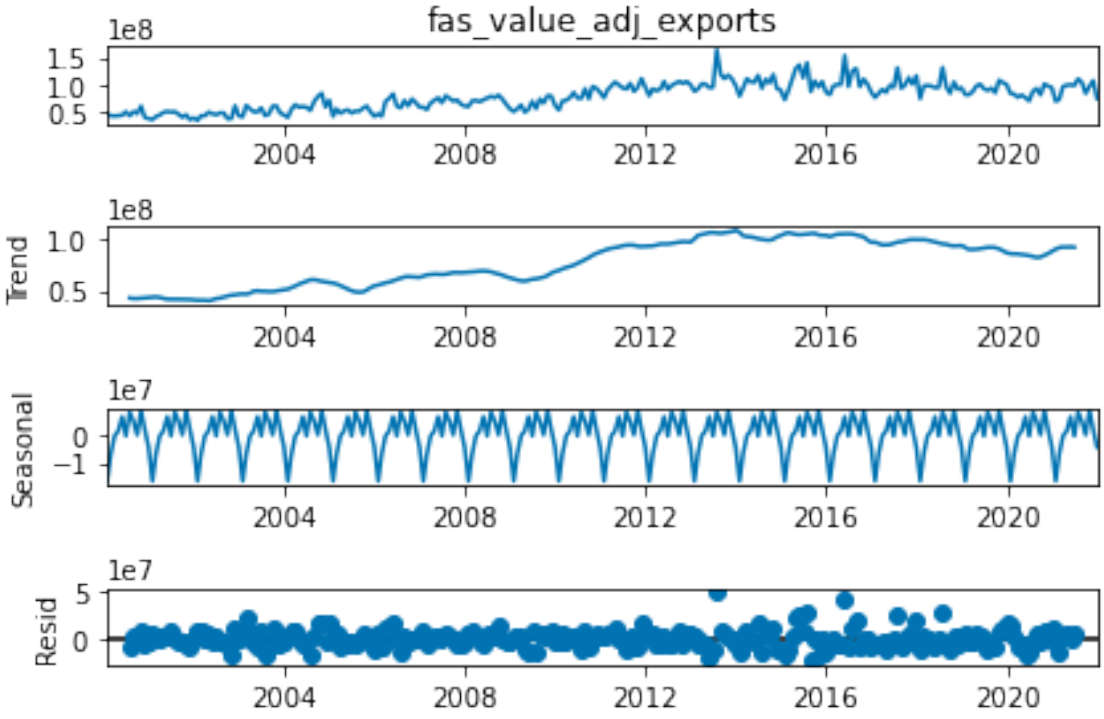
Cool, that passes.

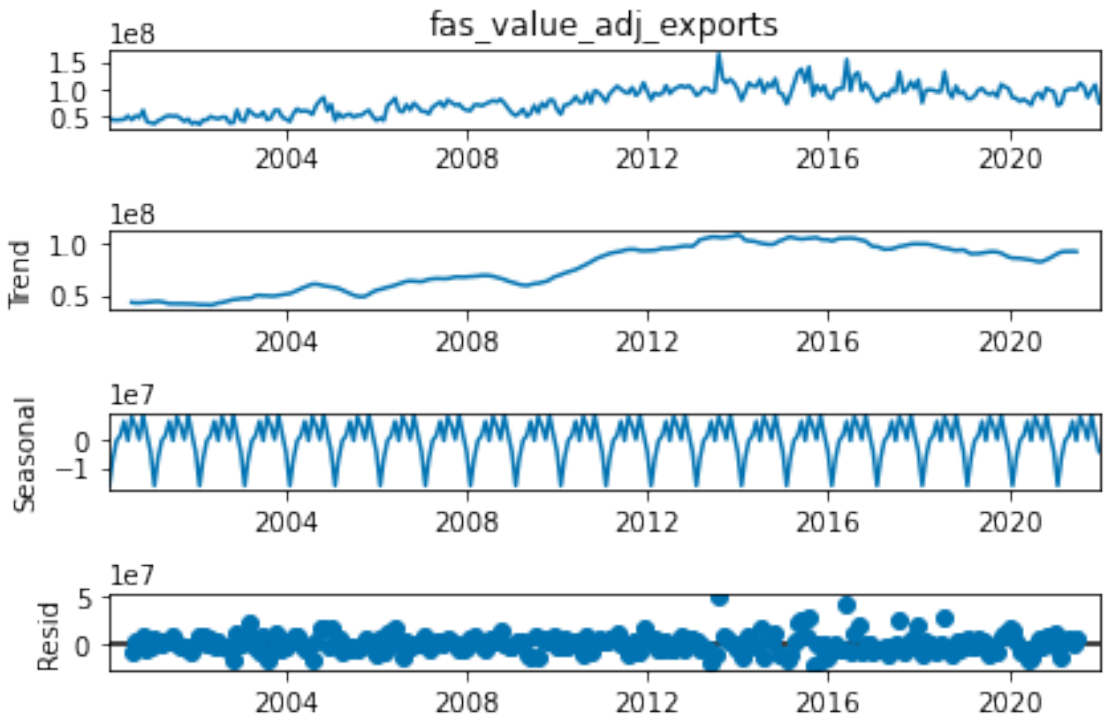
[42]:
```
ts_df = ts_df.merge(exports_df[['quantity_exports_diff12']], left_index=True,␣
↪right_index=True)
```

Real FAS Value of Exports

[43]:
```
decompose_result_mult = seasonal_decompose(exports_df['fas_value_adj_exports'].
↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[43]:





It looks like FAS Value has an upward trend and seasonality. From inspection, the seasonality seems to be in a 12-month cycle. So let's just take the diffs and then run the ADF test.

[44]:
```
exports_df['fas_value_adj_exports_diff1'] = exports_df['fas_value_adj_exports']␣
 ↪- exports_df['fas_value_adj_exports'].shift(1)
exports_df['fas_value_adj_exports_diff1_diff12'] =␣
 ↪exports_df['fas_value_adj_exports_diff1'] -␣
 ↪exports_df['fas_value_adj_exports_diff1'].shift(12)
adf(exports_df['fas_value_adj_exports_diff1_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                  -6.410001e+00
p-value:                  1.901197e-08
lags:                     1.600000e+01
observations:             2.340000e+02
critical value (1%):     -3.458608e+00
critical value (5%):     -2.873972e+00
critical value (10%):    -2.573396e+00
dtype: float64
```

That worked.

[45]:
```
ts_df = ts_df.merge(exports_df['fas_value_adj_exports_diff1_diff12'],␣
 ↪left_index=True, right_index=True)
```

1.4.4 Imports

Now we'll adjust imports for stationarity. It may be good to add a variable that represents the percentage of world totals that are affected by the additional tariffs. So I'll do that first and also adjust for per-capita values.

[46]:
```
imports_df = df[['dutiable_value_world_imports',
    'dutiable_value_adj_world_imports',
    'landed_duty_paid_value_world_imports',
    'landed_duty_paid_value_adj_world_imports',
    'customs_value_world_imports', 'customs_value_adj_world_imports',
    'quantity_world_imports', 'charges_insurance_freight_world_imports',
    'charges_insurance_freight_adj_world_imports',
    'calculated_duties_world_imports',
    'calculated_duties_adj_world_imports',
    'dutiable_value_ukfrspde_imports',
    'dutiable_value_adj_ukfrspde_imports',
    'landed_duty_paid_value_ukfrspde_imports',
    'landed_duty_paid_value_adj_ukfrspde_imports',
    'customs_value_ukfrspde_imports', 'customs_value_adj_ukfrspde_imports',
    'quantity_ukfrspde_imports',
    'charges_insurance_freight_ukfrspde_imports',
    'charges_insurance_freight_adj_ukfrspde_imports',
    'calculated_duties_ukfrspde_imports',
    'calculated_duties_adj_ukfrspde_imports', 'population']].copy()

# Proportion of imports' values provided by UK, Fr, Sp, and De
imports_df['dutiable_value_ukfrspde_proportion_imports'] =␣
 ↪imports_df['dutiable_value_adj_ukfrspde_imports'] /␣
 ↪imports_df['dutiable_value_adj_world_imports']
imports_df['landed_duty_paid_ukfrspde_proportion_imports'] =␣
 ↪imports_df['landed_duty_paid_value_adj_ukfrspde_imports'] /␣
 ↪imports_df['landed_duty_paid_value_adj_world_imports']
imports_df['customs_value_ukfrspde_proportion_imports'] =␣
 ↪imports_df['customs_value_adj_ukfrspde_imports'] /␣
 ↪imports_df['customs_value_adj_world_imports']
imports_df['quantity_ukfrspde_proportion_imports'] =␣
 ↪imports_df['quantity_ukfrspde_imports'] /␣
 ↪imports_df['quantity_world_imports']
imports_df['charges_insurance_freight_ukfrspde_proportion_imports'] =␣
 ↪imports_df['charges_insurance_freight_adj_ukfrspde_imports'] /␣
 ↪imports_df['charges_insurance_freight_adj_world_imports']
imports_df['calculated_duties_ukfrspde_proportion_imports'] =␣
 ↪imports_df['calculated_duties_adj_ukfrspde_imports'] /␣
 ↪imports_df['calculated_duties_adj_world_imports']

# Per-capita quantities of wine imports
```

```
imports_df['quantity_world_per_capita_imports'] =␣
 ↪imports_df['quantity_world_imports'] / imports_df['population']
imports_df['quantity_ukfrspde_per_capita_imports'] =␣
 ↪imports_df['quantity_ukfrspde_imports'] / imports_df['population']

# Incorporate new fields into original dataframe
df['dutiable_value_ukfrspde_proportion_imports'] =␣
 ↪imports_df['dutiable_value_ukfrspde_proportion_imports']
df['landed_duty_paid_ukfrspde_proportion_imports'] =␣
 ↪imports_df['landed_duty_paid_ukfrspde_proportion_imports']
df['customs_value_ukfrspde_proportion_imports'] =␣
 ↪imports_df['customs_value_ukfrspde_proportion_imports']
df['quantity_ukfrspde_proportion_imports'] =␣
 ↪imports_df['quantity_ukfrspde_proportion_imports']
df['charges_insurance_freight_ukfrspde_proportion_imports'] =␣
 ↪imports_df['charges_insurance_freight_ukfrspde_proportion_imports']
df['calculated_duties_ukfrspde_proportion_imports'] =␣
 ↪imports_df['calculated_duties_ukfrspde_proportion_imports']
df['quantity_world_per_capita_imports'] =␣
 ↪imports_df['quantity_world_per_capita_imports']
df['quantity_ukfrspde_per_capita_imports'] =␣
 ↪imports_df['quantity_ukfrspde_per_capita_imports']
```

U.K., France, Spain, Germany

[47]:
```
cols = []
for c in imports_df.columns:
    if 'ukfrspde' in c:
        cols.append(c)

imports_subset1_df = imports_df[cols].copy()
```

[48]:
```
imports_subset1_df = imports_subset1_df[['customs_value_adj_ukfrspde_imports',␣
 ↪'calculated_duties_adj_ukfrspde_imports',
    'charges_insurance_freight_adj_ukfrspde_imports',␣
 ↪'quantity_ukfrspde_imports',
    'quantity_ukfrspde_proportion_imports',␣
 ↪'quantity_ukfrspde_per_capita_imports']]
```

Customs Value

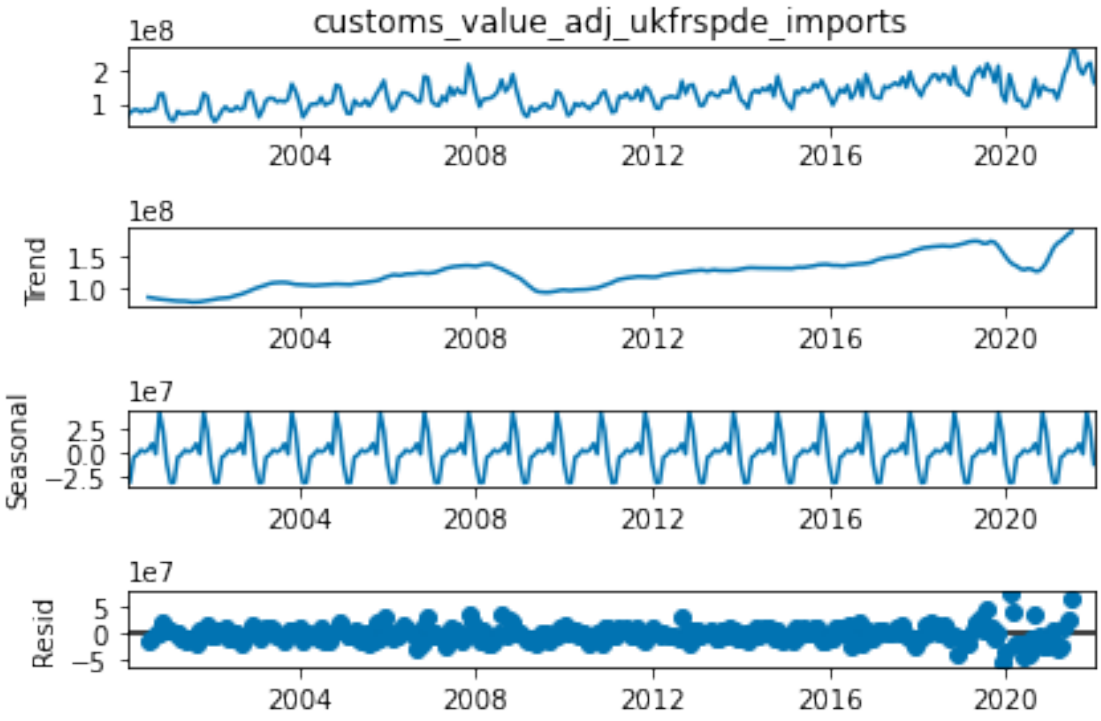
[49]:
```
adf(imports_subset1_df['customs_value_adj_ukfrspde_imports'].dropna())
```

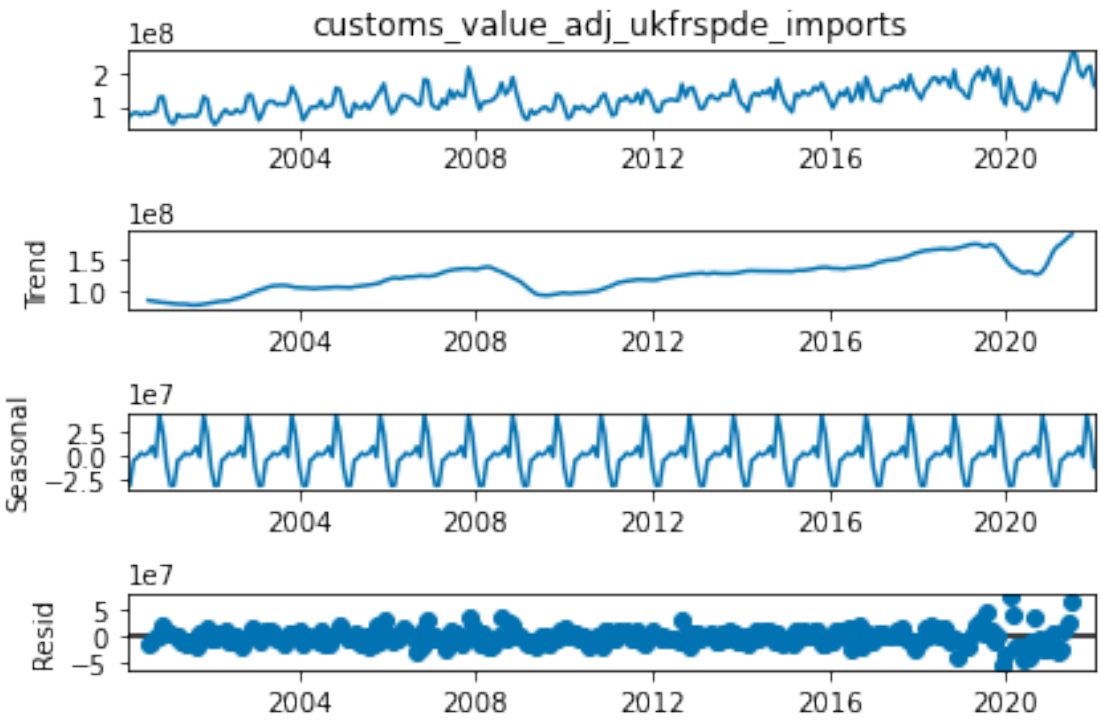
```
Augmented Dickey-Fuller Test:
t-stat:                   -1.522449
p-value:                   0.522378
lags:                     13.000000
observations:            250.000000
critical value (1%):      -3.456781
```

```
critical value (5%):      -2.873172
critical value (10%):     -2.572969
dtype: float64
```

[50]:
```
decompose_result_mult = seasonal_decompose(imports_subset1_df['customs_value_adj_ukfrspde_imports'].dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[50]:





[51]:
```
imports_subset1_df['customs_value_adj_ukfrspde_imports_diff1'] =␣
↪imports_subset1_df['customs_value_adj_ukfrspde_imports'] -␣
↪imports_subset1_df['customs_value_adj_ukfrspde_imports'].shift(1)
imports_subset1_df['customs_value_adj_ukfrspde_imports_diff1_diff12'] =␣
↪imports_subset1_df['customs_value_adj_ukfrspde_imports_diff1'] -␣
↪imports_subset1_df['customs_value_adj_ukfrspde_imports_diff1'].shift(12)
adf(imports_subset1_df['customs_value_adj_ukfrspde_imports_diff1_diff12'].
↪dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                 -7.935206e+00
p-value:                 3.428209e-12
lags:                    1.300000e+01
observations:            2.370000e+02
critical value (1%):    -3.458247e+00
critical value (5%):    -2.873814e+00
critical value (10%):   -2.573311e+00
dtype: float64
```

[52]:
```
ts_df = ts_df.
↪merge(imports_subset1_df['customs_value_adj_ukfrspde_imports_diff1_diff12'],␣
↪left_index=True, right_index=True)
```

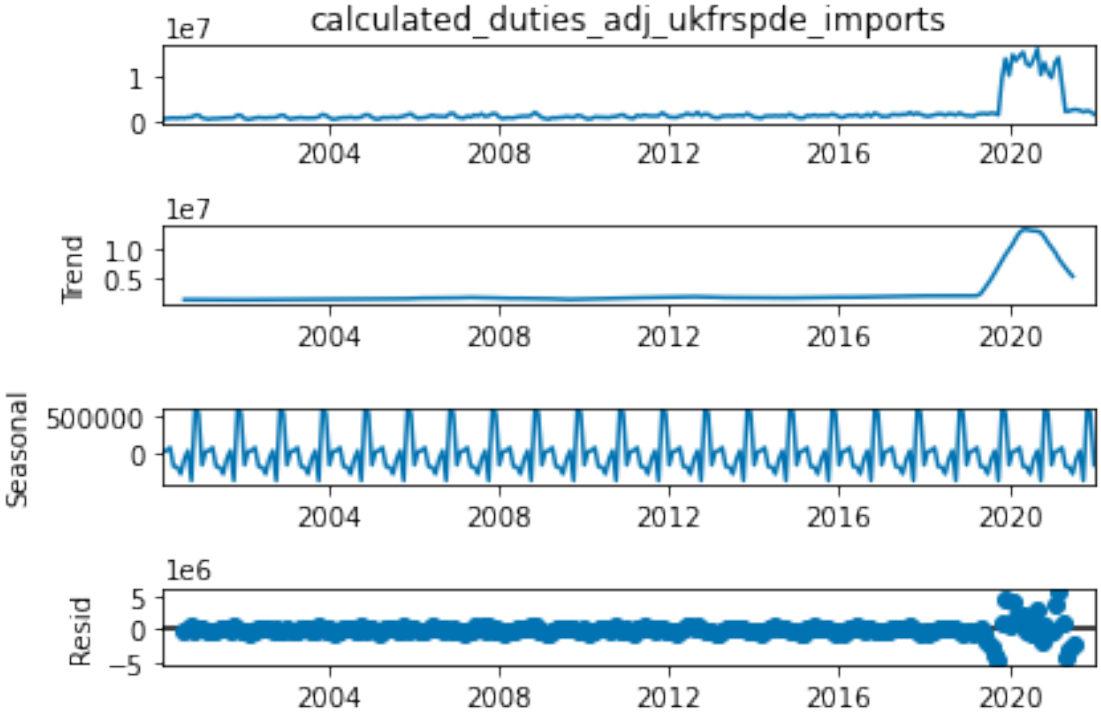
Calculated Duties

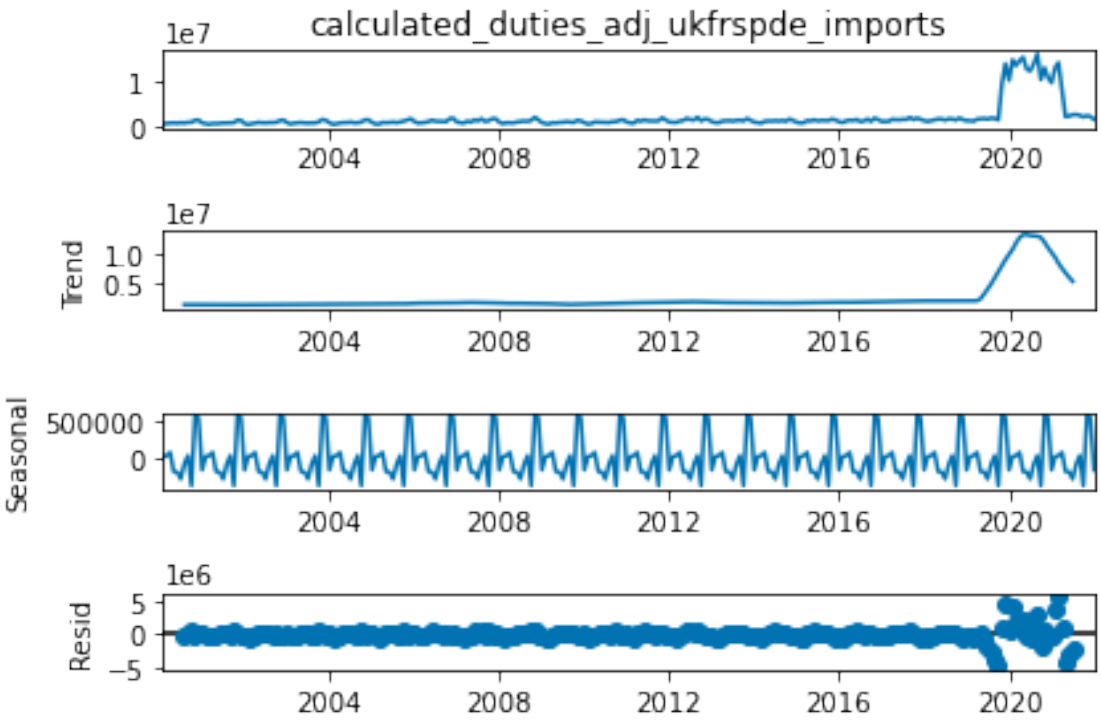
[53]:
```
adf(imports_subset1_df['calculated_duties_adj_ukfrspde_imports'])
```

```
Augmented Dickey-Fuller Test:
t-stat:                    -3.654396
p-value:                    0.004802
lags:                      15.000000
observations:             248.000000
critical value (1%):       -3.456996
critical value (5%):       -2.873266
critical value (10%):      -2.573019
dtype: float64
```

[54]:
```
decompose_result_mult =␣
↪seasonal_decompose(imports_subset1_df['calculated_duties_adj_ukfrspde_imports'].
↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[54]:





[55]:
```
imports_subset1_df['calculated_duties_adj_ukfrspde_imports_diff1'] =␣
 ↪imports_subset1_df['calculated_duties_adj_ukfrspde_imports'] -␣
 ↪imports_subset1_df['calculated_duties_adj_ukfrspde_imports'].shift(1)
imports_subset1_df['calculated_duties_adj_ukfrspde_imports_diff1_diff12'] =␣
 ↪imports_subset1_df['calculated_duties_adj_ukfrspde_imports_diff1'] -␣
 ↪imports_subset1_df['calculated_duties_adj_ukfrspde_imports_diff1'].shift(12)
adf(imports_subset1_df['calculated_duties_adj_ukfrspde_imports_diff1_diff12'].
 ↪dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                   -5.431072
p-value:                   0.000003
lags:                     16.000000
observations:            234.000000
critical value (1%):      -3.458608
critical value (5%):      -2.873972
critical value (10%):     -2.573396
dtype: float64
```

[56]:
```
ts_df = ts_df.
 ↪merge(imports_subset1_df['calculated_duties_adj_ukfrspde_imports_diff1_diff12'],␣
 ↪left_index=True, right_index=True)
```

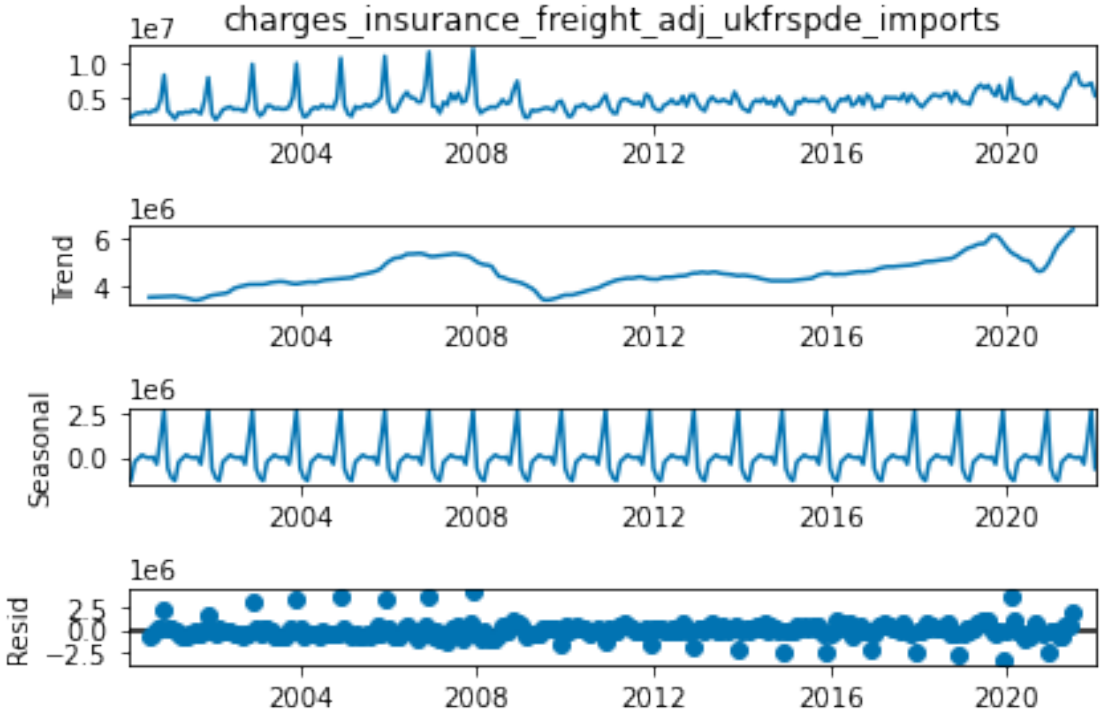
Charges, Insurance, and Freight

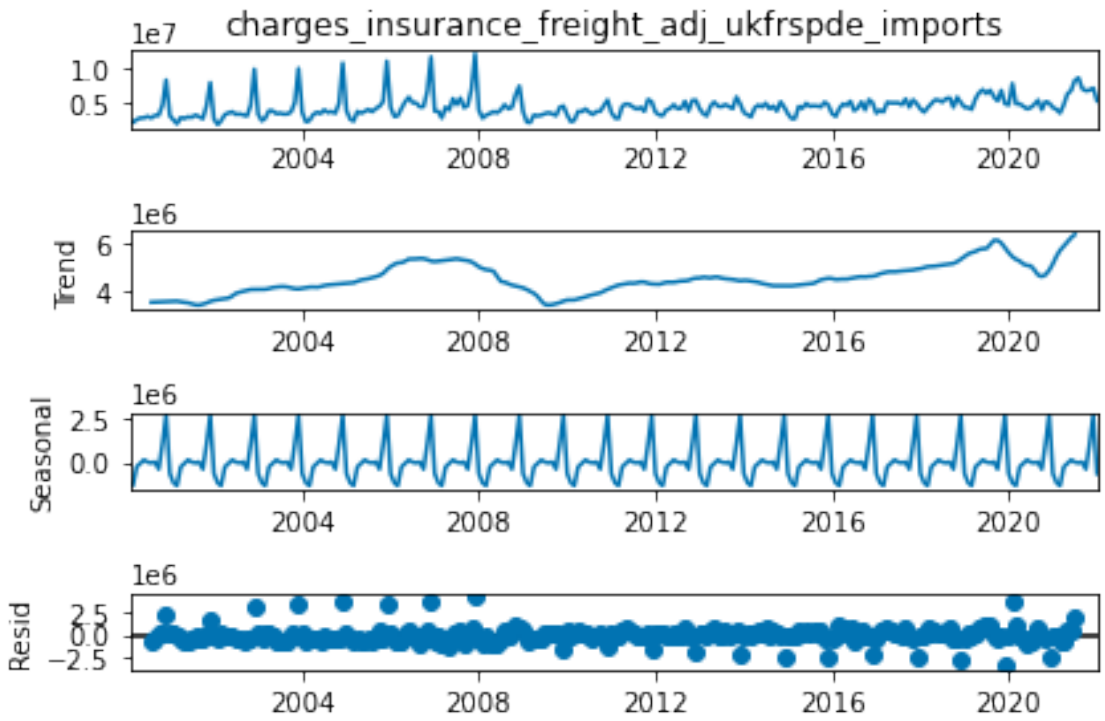
[57]:
```
adf(imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports'])
```

```
Augmented Dickey-Fuller Test:
t-stat:                    -1.451255
p-value:                    0.557473
lags:                      13.000000
observations:             250.000000
critical value (1%):       -3.456781
critical value (5%):       -2.873172
critical value (10%):      -2.572969
dtype: float64
```

[58]:
```
decompose_result_mult =␣
↪seasonal_decompose(imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports'].
↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[58]:





[59]:
```
imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports_diff1'] =␣
↪imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports'] -␣
↪imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports'].shift(1)
imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12']␣
↪= imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports_diff1']␣
↪- imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports_diff1'].
↪shift(12)
adf(imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12'].
↪dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                  -6.201207e+00
p-value:                  5.801371e-08
lags:                     1.600000e+01
observations:             2.340000e+02
critical value (1%):     -3.458608e+00
critical value (5%):     -2.873972e+00
critical value (10%):    -2.573396e+00
dtype: float64
```

[60]:
```
ts_df = ts_df.
↪merge(imports_subset1_df['charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12'],␣
↪left_index=True, right_index=True)
```

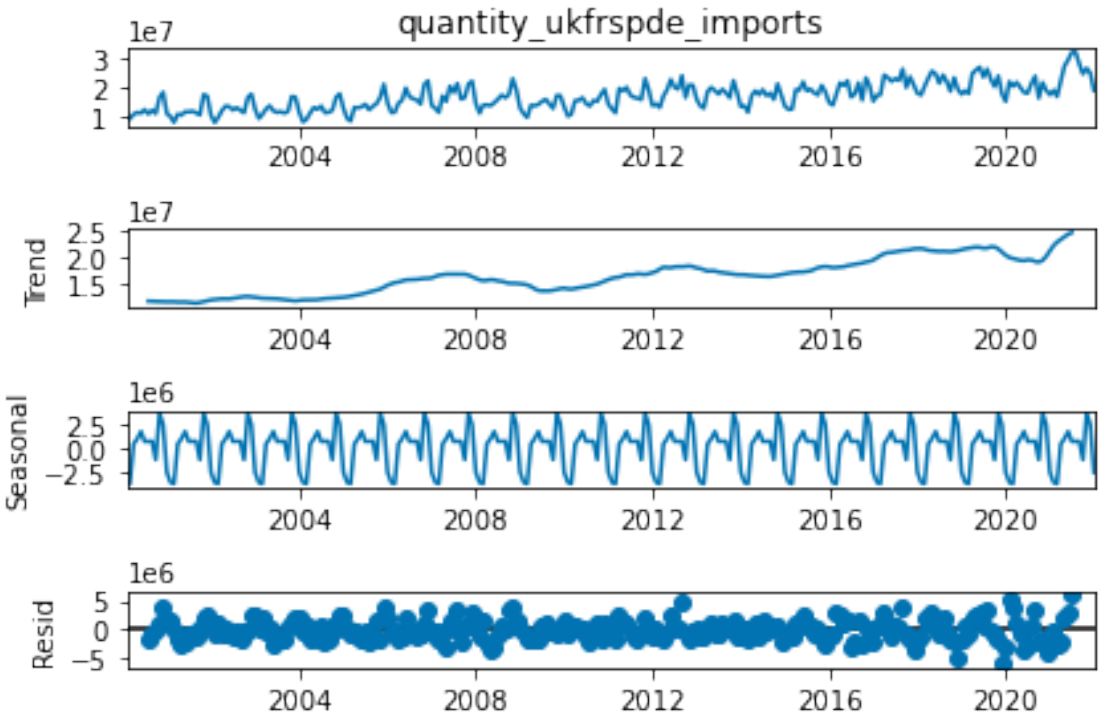
Quantity

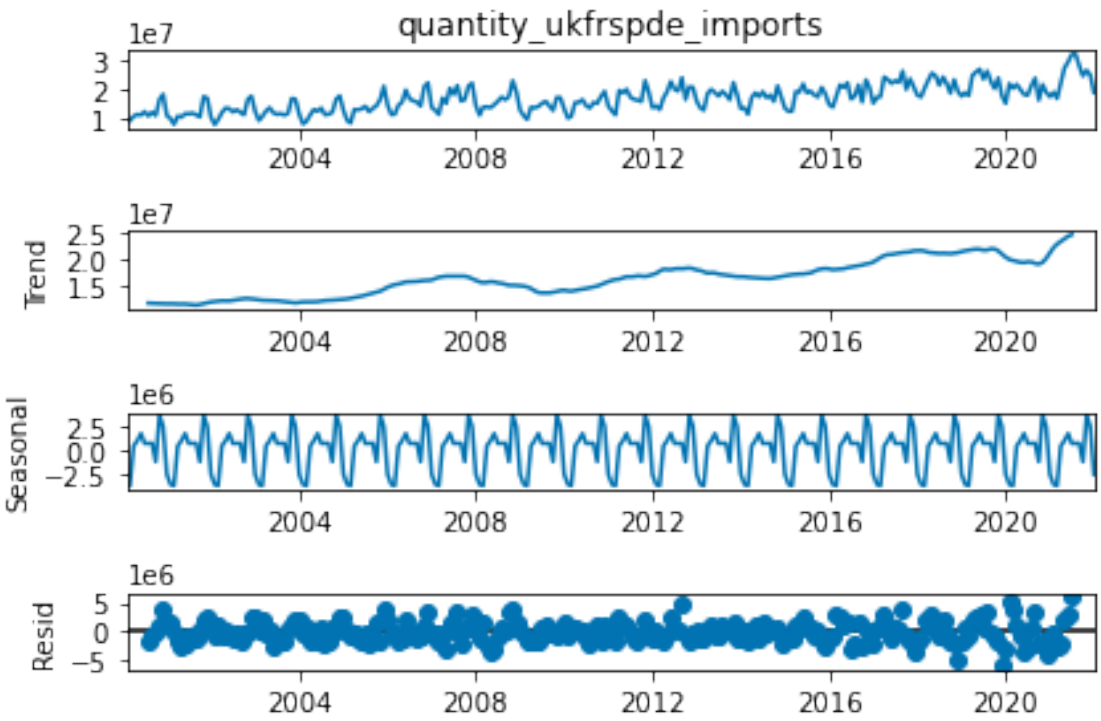
[61]:
```
adf(imports_subset1_df['quantity_ukfrspde_imports'])
```

```
Augmented Dickey-Fuller Test:
t-stat:                    -0.436341
p-value:                    0.903852
lags:                      13.000000
observations:             250.000000
critical value (1%):       -3.456781
critical value (5%):       -2.873172
critical value (10%):      -2.572969
dtype: float64
```

[62]:
```
decompose_result_mult =␣
↪seasonal_decompose(imports_subset1_df['quantity_ukfrspde_imports'].dropna(),␣
↪model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[62]:





[63]:
```
imports_subset1_df['quantity_ukfrspde_imports_diff1'] =␣
↪imports_subset1_df['quantity_ukfrspde_imports'] -␣
↪imports_subset1_df['quantity_ukfrspde_imports'].shift(1)
imports_subset1_df['quantity_ukfrspde_imports_diff1_diff12'] =␣
↪imports_subset1_df['quantity_ukfrspde_imports_diff1'] -␣
↪imports_subset1_df['quantity_ukfrspde_imports_diff1'].shift(12)
adf(imports_subset1_df['quantity_ukfrspde_imports_diff1_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                  -6.424159e+00
p-value:                  1.761400e-08
lags:                     1.300000e+01
observations:             2.370000e+02
critical value (1%):     -3.458247e+00
critical value (5%):     -2.873814e+00
critical value (10%):    -2.573311e+00
dtype: float64
```

[64]:
```
ts_df = ts_df.
↪merge(imports_subset1_df['quantity_ukfrspde_imports_diff1_diff12'],␣
↪left_index=True, right_index=True)
```

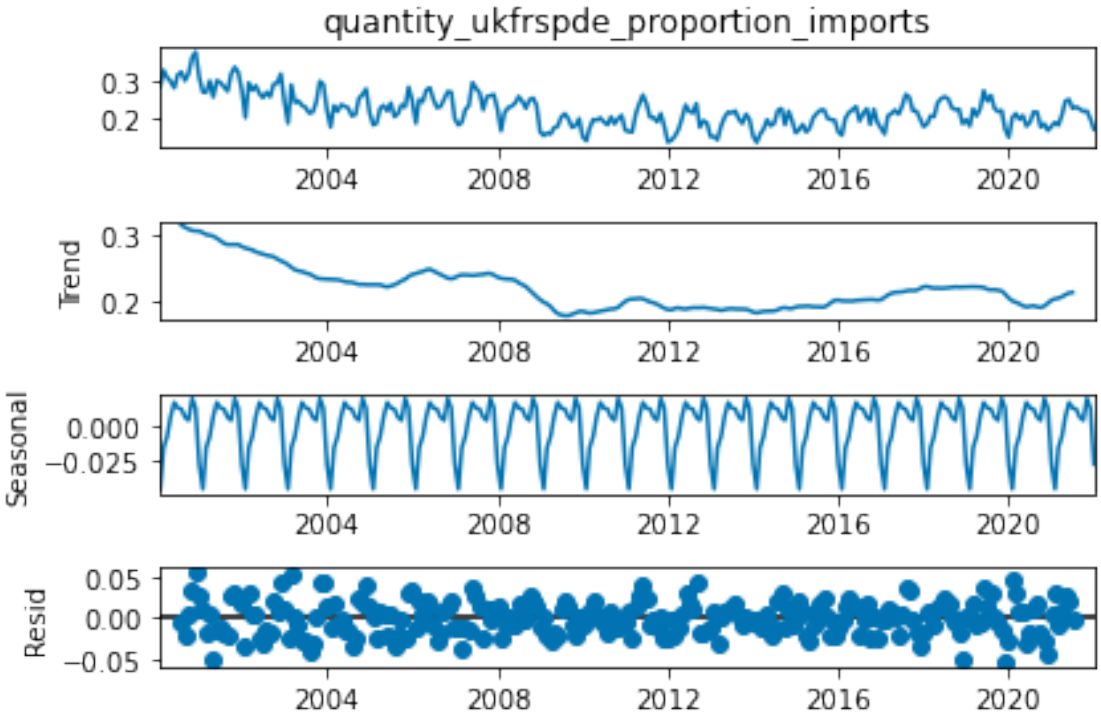
Quantity as a Proportion of World Imports

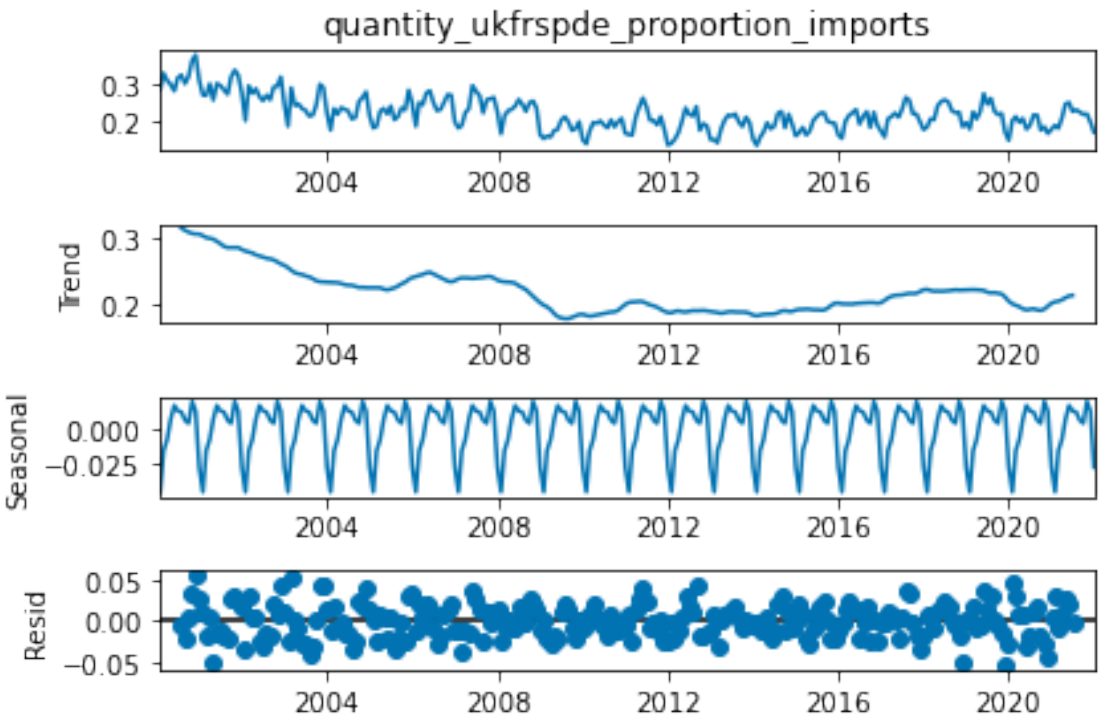
[65]:
```
adf(imports_subset1_df['quantity_ukfrspde_proportion_imports'])
```

```
Augmented Dickey-Fuller Test:
t-stat:                    -2.784062
p-value:                    0.060587
lags:                      13.000000
observations:             250.000000
critical value (1%):       -3.456781
critical value (5%):       -2.873172
critical value (10%):      -2.572969
dtype: float64
```

[66]:
```
decompose_result_mult = ␣
↪seasonal_decompose(imports_subset1_df['quantity_ukfrspde_proportion_imports'].
↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[66]:





[67]:
```
imports_subset1_df['quantity_ukfrspde_proportion_imports_diff12'] =␣
↪imports_subset1_df['quantity_ukfrspde_proportion_imports'] -␣
↪imports_subset1_df['quantity_ukfrspde_proportion_imports'].shift(12)
adf(imports_subset1_df['quantity_ukfrspde_proportion_imports_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                    -3.105613
p-value:                    0.026141
lags:                      13.000000
observations:             238.000000
critical value (1%):       -3.458128
critical value (5%):       -2.873762
critical value (10%):      -2.573283
dtype: float64
```

[68]:
```
imports_subset1_df['quantity_ukfrspde_proportion_imports_diff12_diff1'] =␣
↪imports_subset1_df['quantity_ukfrspde_proportion_imports_diff12'] -␣
↪imports_subset1_df['quantity_ukfrspde_proportion_imports_diff12'].shift(1)
adf(imports_subset1_df['quantity_ukfrspde_proportion_imports_diff12_diff1'].
↪dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                  -6.683635e+00
p-value:                  4.280055e-09
```

```
lags:                    1.300000e+01
observations:            2.370000e+02
critical value (1%):    -3.458247e+00
critical value (5%):    -2.873814e+00
critical value (10%):   -2.573311e+00
dtype: float64
```

[69]:
```
ts_df = ts_df.
↪merge(imports_subset1_df['quantity_ukfrspde_proportion_imports_diff12_diff1'],␣
↪left_index=True, right_index=True)
```

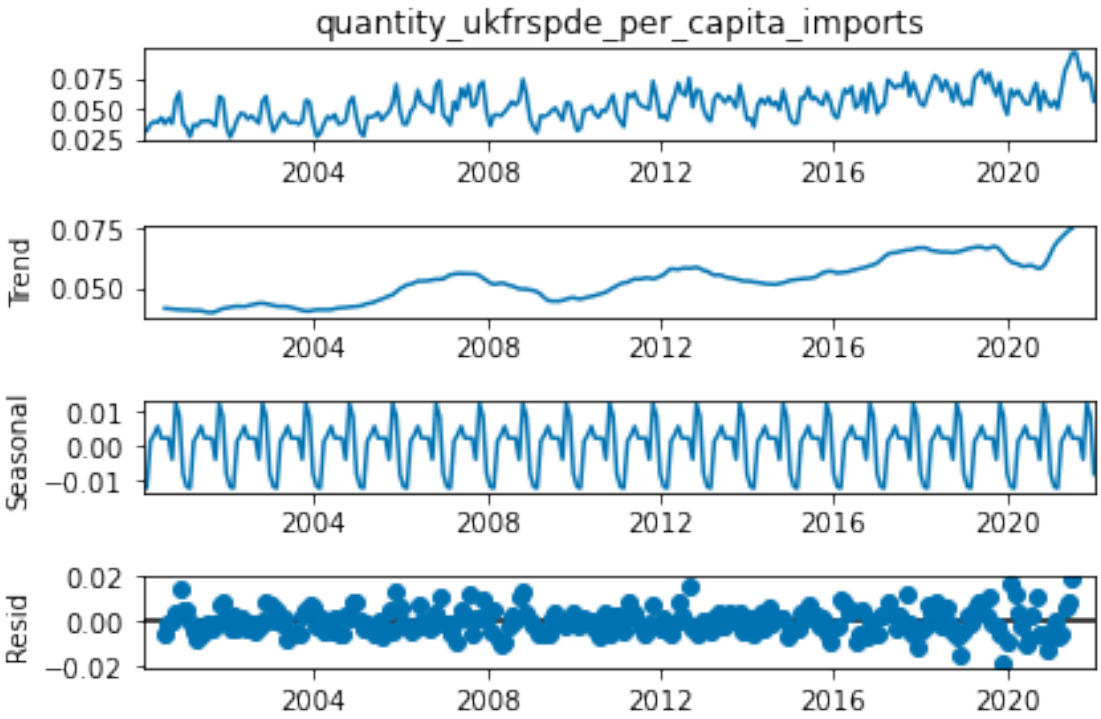
Quantity Per Capita

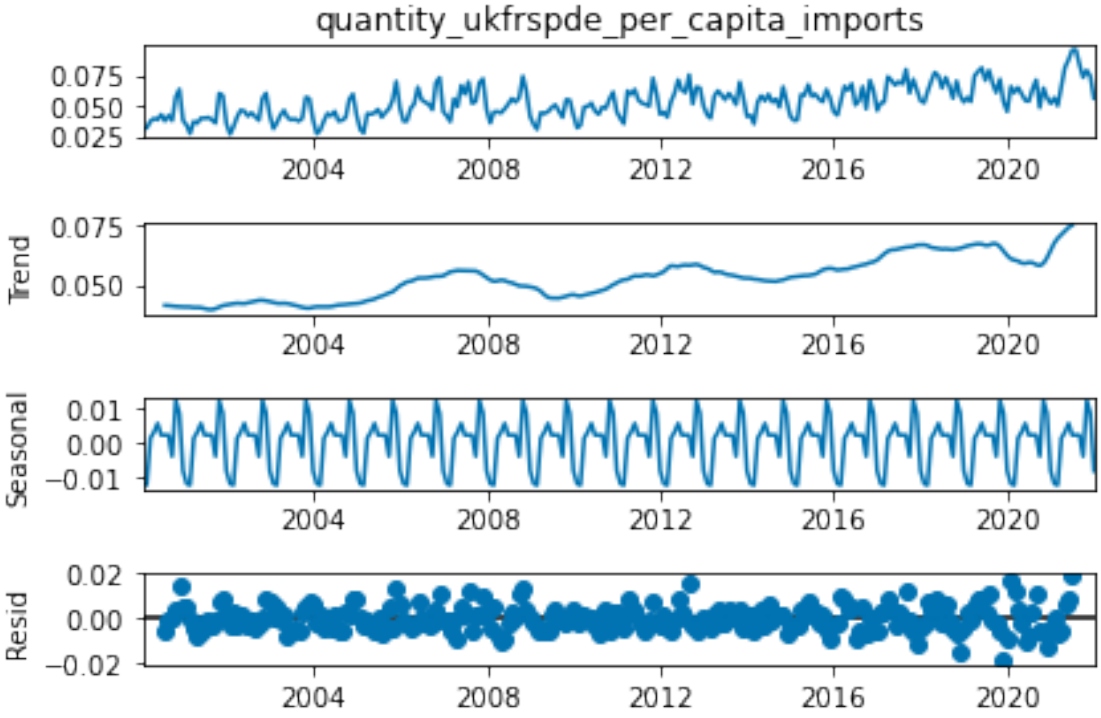
[70]:
```
adf(imports_subset1_df['quantity_ukfrspde_per_capita_imports'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                   -0.763268
p-value:                   0.829674
lags:                     13.000000
observations:            250.000000
critical value (1%):      -3.456781
critical value (5%):      -2.873172
critical value (10%):     -2.572969
dtype: float64
```

[71]:
```
decompose_result_mult =␣
↪seasonal_decompose(imports_subset1_df['quantity_ukfrspde_per_capita_imports'].
↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[71]:


quantity_ukfrspde_per_capita_imports
0.075
0.050
0.025
2004
2008
2012
2016
2020
Trend
0.075
0.050
Seasonal
0.01
0.00
−0.01
Resid
0.02
0.00
−0.02


quantity_ukfrspde_per_capita_imports
0.075
0.050
0.025
2004
2008
2012
2016
2020
Trend
0.075
0.050
Seasonal
0.01
0.00
−0.01
Resid
0.02
0.00
−0.02

[72]:
```
imports_subset1_df['quantity_ukfrspde_per_capita_imports_diff1'] =␣
↪imports_subset1_df['quantity_ukfrspde_per_capita_imports'] -␣
↪imports_subset1_df['quantity_ukfrspde_per_capita_imports'].shift(1)
imports_subset1_df['quantity_ukfrspde_per_capita_imports_diff1_diff12'] =␣
↪imports_subset1_df['quantity_ukfrspde_per_capita_imports_diff1'] -␣
↪imports_subset1_df['quantity_ukfrspde_per_capita_imports_diff1'].shift(12)
adf(imports_subset1_df['quantity_ukfrspde_per_capita_imports_diff1_diff12'].
↪dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                 -6.326419e+00
p-value:                 2.978744e-08
lags:                    1.300000e+01
observations:            2.370000e+02
critical value (1%):    -3.458247e+00
critical value (5%):    -2.873814e+00
critical value (10%):   -2.573311e+00
dtype: float64
```

[73]:
```
ts_df = ts_df.
↪merge(imports_subset1_df['quantity_ukfrspde_per_capita_imports_diff1_diff12'],␣
↪left_index=True, right_index=True)
```

World

[74]:
```
cols = []
for c in imports_df.columns:
    if 'world' in c:
        cols.append(c)

imports_subset2_df = imports_df[cols].copy()
```

[75]:
```
imports_subset2_df = imports_subset2_df[['customs_value_adj_world_imports',␣
↪'calculated_duties_adj_world_imports',
   'charges_insurance_freight_adj_world_imports', 'quantity_world_imports',␣
↪'quantity_world_per_capita_imports']]
```

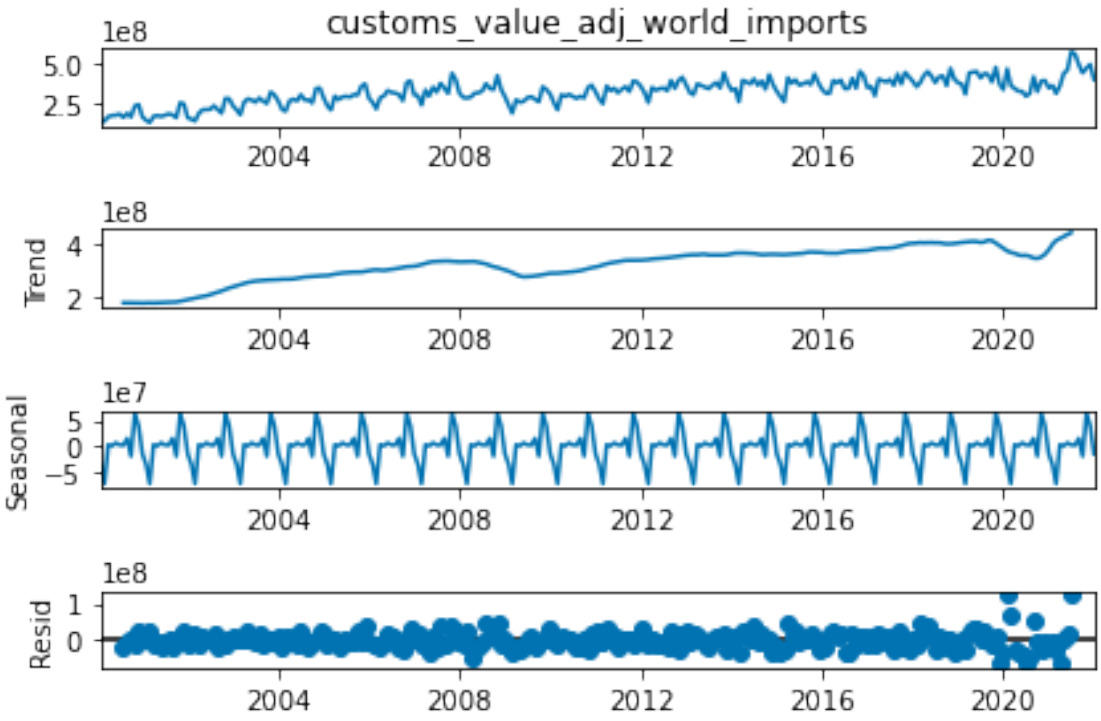
Customs Value

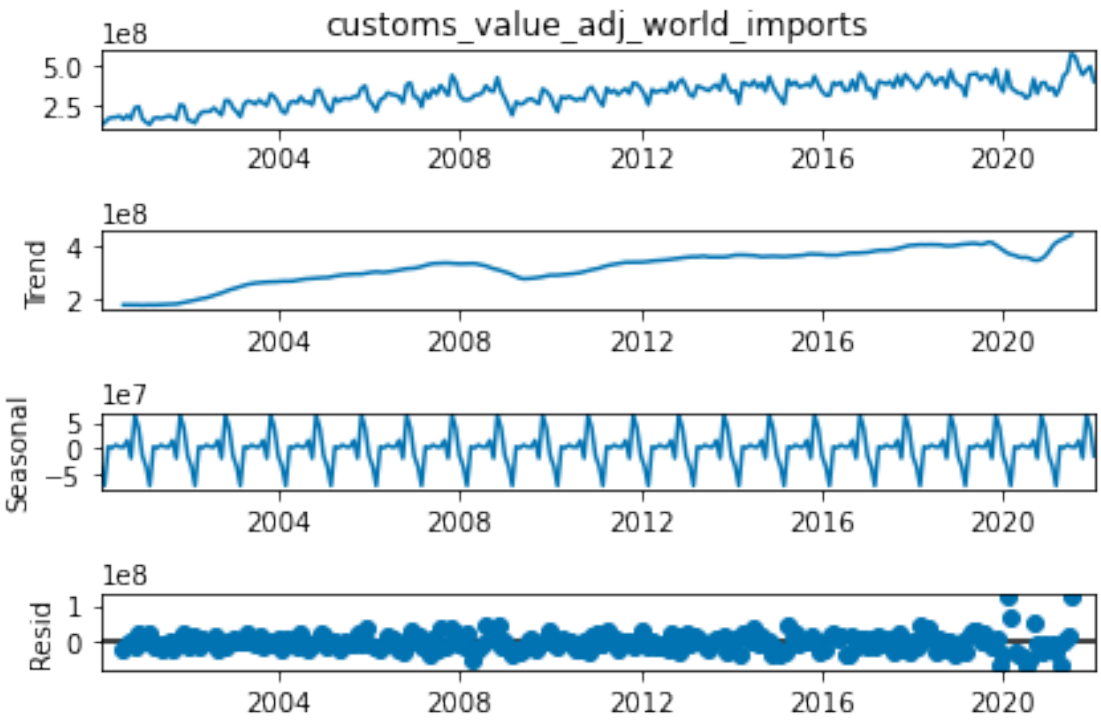
[76]:
```
adf(imports_subset2_df['customs_value_adj_world_imports'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                   -1.459830
p-value:                   0.553281
lags:                     13.000000
observations:            250.000000
critical value (1%):      -3.456781
critical value (5%):      -2.873172
critical value (10%):     -2.572969
dtype: float64
```

[77]:
```
decompose_result_mult = seasonal_decompose(imports_subset2_df['customs_value_adj_world_imports'].dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[77]:





[78]:
```
imports_subset2_df['customs_value_adj_world_imports_diff1'] =␣
↪imports_subset2_df['customs_value_adj_world_imports'] -␣
↪imports_subset2_df['customs_value_adj_world_imports'].shift(1)
imports_subset2_df['customs_value_adj_world_imports_diff1_diff12'] =␣
↪imports_subset2_df['customs_value_adj_world_imports_diff1'] -␣
↪imports_subset2_df['customs_value_adj_world_imports_diff1'].shift(12)
adf(imports_subset2_df['customs_value_adj_world_imports_diff1_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                  -6.834680e+00
p-value:                  1.855488e-09
lags:                     1.300000e+01
observations:             2.370000e+02
critical value (1%):     -3.458247e+00
critical value (5%):     -2.873814e+00
critical value (10%):    -2.573311e+00
dtype: float64
```

[79]:
```
ts_df = ts_df.
↪merge(imports_subset2_df['customs_value_adj_world_imports_diff1_diff12'],␣
↪left_index=True, right_index=True)
```

Calculated Duties

[80]:
```
adf(imports_subset2_df['calculated_duties_adj_world_imports'])
```

```
Augmented Dickey-Fuller Test:
t-stat:                   -3.584811
p-value:                   0.006058
lags:                     12.000000
observations:            251.000000
critical value (1%):      -3.456674
critical value (5%):      -2.873125
critical value (10%):     -2.572944
dtype: float64
```

[81]:
```
ts_df = ts_df.merge(imports_subset2_df['calculated_duties_adj_world_imports'],␣
↪left_index=True, right_index=True)
```

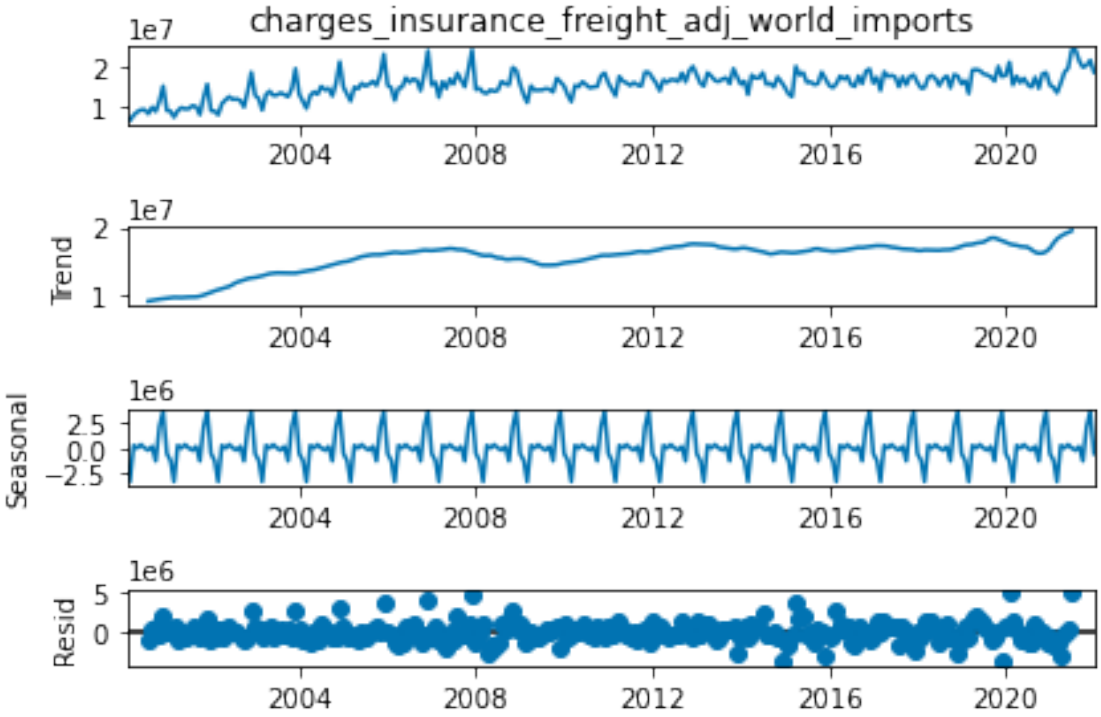
Charges, Insurance, and Freight

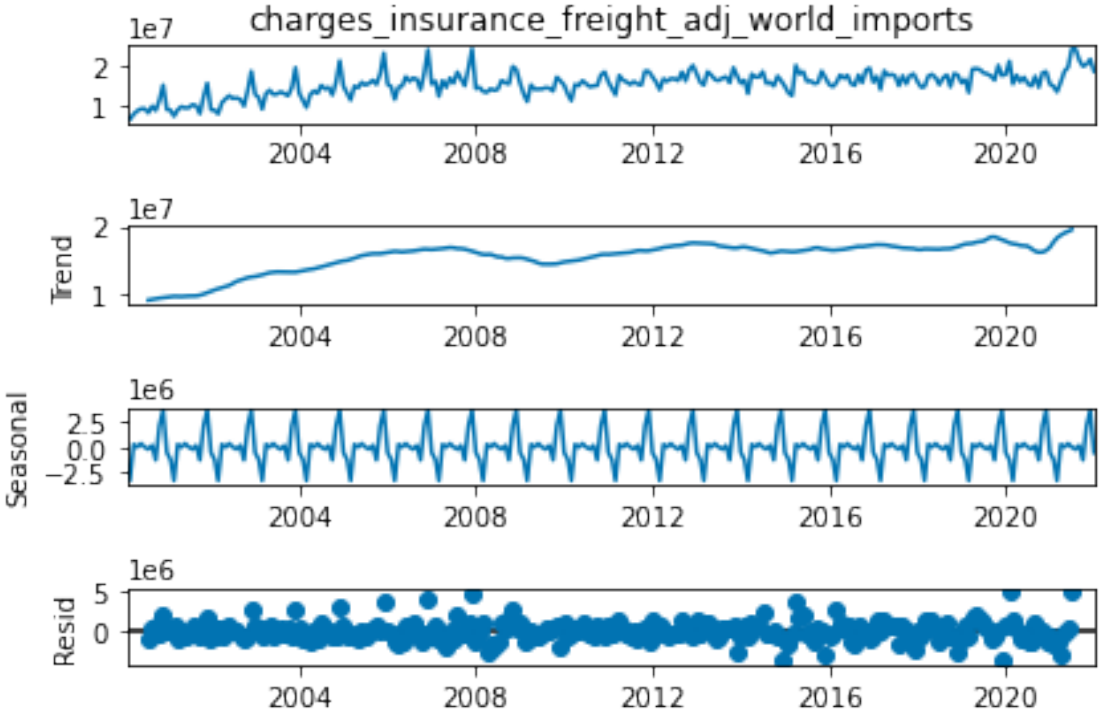
[82]:
```
adf(imports_subset2_df['charges_insurance_freight_adj_world_imports'])
```

```
Augmented Dickey-Fuller Test:
t-stat:                   -2.183202
p-value:                   0.212398
lags:                     16.000000
observations:            247.000000
critical value (1%):      -3.457105
critical value (5%):      -2.873314
critical value (10%):     -2.573044
dtype: float64
```

[83]:
```
decompose_result_mult =␣
↪seasonal_decompose(imports_subset2_df['charges_insurance_freight_adj_world_imports'].
↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[83]:


charges_insurance_freight_adj_world_imports
1e7
2
1
2004
2008
2012
2016
2020
Trend
Seasonal
1e6
2.5
0.0
−2.5
Resid
5
0


charges_insurance_freight_adj_world_imports
1e7
2
1
2004
2008
2012
2016
2020
Trend
Seasonal
1e6
2.5
0.0
−2.5
Resid
5
0

[84]:
```
imports_subset2_df['charges_insurance_freight_adj_world_imports_diff1'] =␣
↪imports_subset2_df['charges_insurance_freight_adj_world_imports'] -␣
↪imports_subset2_df['charges_insurance_freight_adj_world_imports'].shift(1)
imports_subset2_df['charges_insurance_freight_adj_world_imports_diff1_diff12']␣
↪= imports_subset2_df['charges_insurance_freight_adj_world_imports_diff1'] -␣
↪imports_subset2_df['charges_insurance_freight_adj_world_imports_diff1'].
↪shift(12)
adf(imports_subset2_df['charges_insurance_freight_adj_world_imports_diff1_diff12'].
↪dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                  -6.190991e+00
p-value:                  6.123570e-08
lags:                     1.600000e+01
observations:             2.340000e+02
critical value (1%):     -3.458608e+00
critical value (5%):     -2.873972e+00
critical value (10%):    -2.573396e+00
dtype: float64
```

[85]:
```
ts_df = ts_df.
↪merge(imports_subset2_df['charges_insurance_freight_adj_world_imports_diff1_diff12'],␣
↪left_index=True, right_index=True)
```

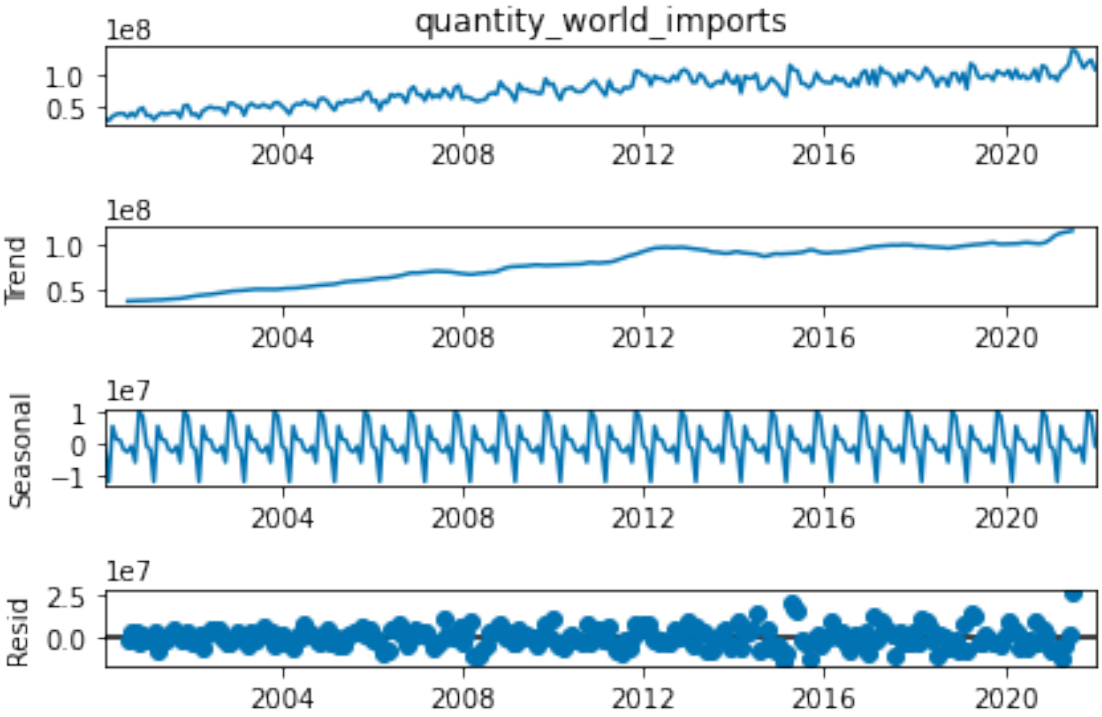
Quantity

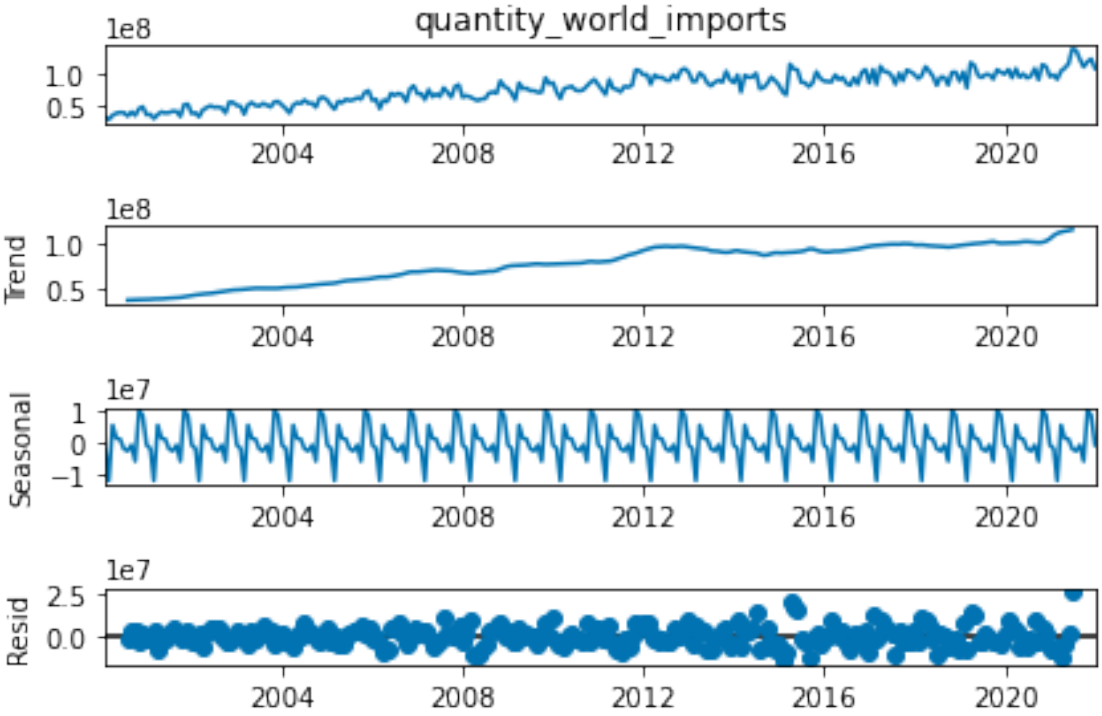
[86]:
```
adf(imports_subset2_df['quantity_world_imports'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                    -0.665185
p-value:                    0.855563
lags:                      13.000000
observations:             250.000000
critical value (1%):       -3.456781
critical value (5%):       -2.873172
critical value (10%):      -2.572969
dtype: float64
```

[87]:
```
decompose_result_mult =␣
↪seasonal_decompose(imports_subset2_df['quantity_world_imports'].dropna(),␣
↪model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[87]:


quantity_world_imports
1e8
1.0
0.5
2004
2008
2012
2016
2020
Trend
Seasonal
1e7
1
0
−1
Resid
2.5
0.0


quantity_world_imports
1e8
1.0
0.5
2004
2008
2012
2016
2020
Trend
Seasonal
1e7
1
0
−1
Resid
2.5
0.0

[88]:
```
imports_subset2_df['quantity_world_imports_diff1'] =␣
 ↪imports_subset2_df['quantity_world_imports'] -␣
 ↪imports_subset2_df['quantity_world_imports'].shift(1)
imports_subset2_df['quantity_world_imports_diff1_diff12'] =␣
 ↪imports_subset2_df['quantity_world_imports_diff1'] -␣
 ↪imports_subset2_df['quantity_world_imports_diff1'].shift(12)
adf(imports_subset2_df['quantity_world_imports_diff1_diff12'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                 -6.029991e+00
p-value:                 1.425555e-07
lags:                    1.600000e+01
observations:            2.340000e+02
critical value (1%):    -3.458608e+00
critical value (5%):    -2.873972e+00
critical value (10%):   -2.573396e+00
dtype: float64
```

[89]:
```
ts_df = ts_df.merge(imports_subset2_df['quantity_world_imports_diff1_diff12'],␣
 ↪left_index=True, right_index=True)
```

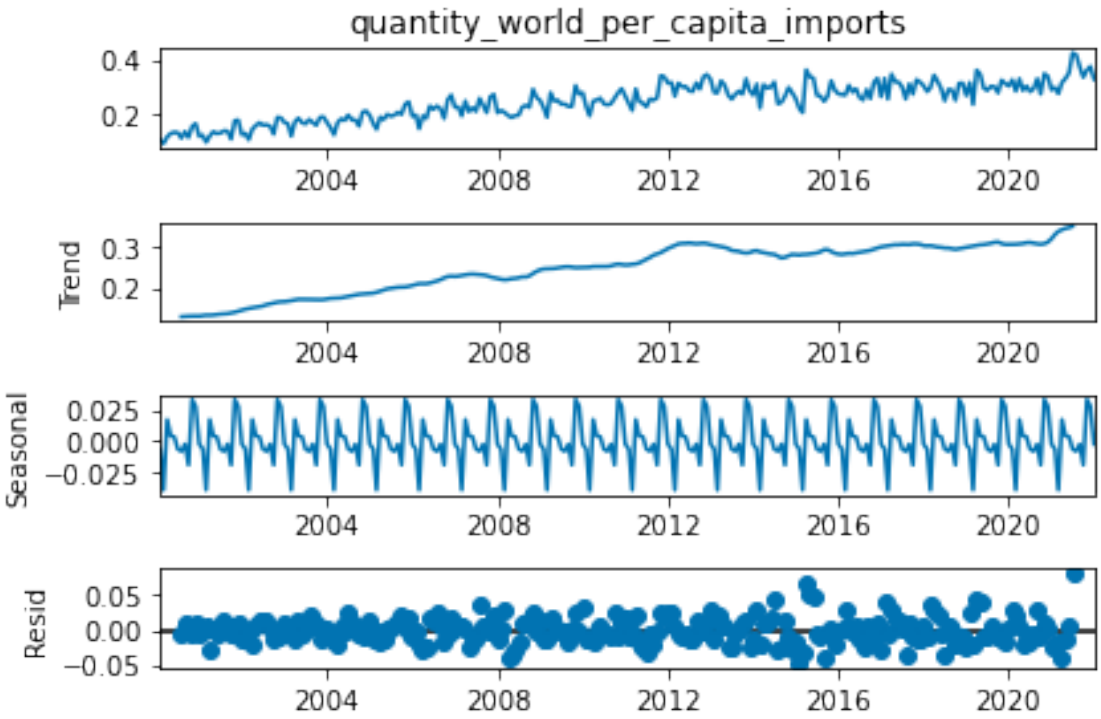
Quantity Per Capita

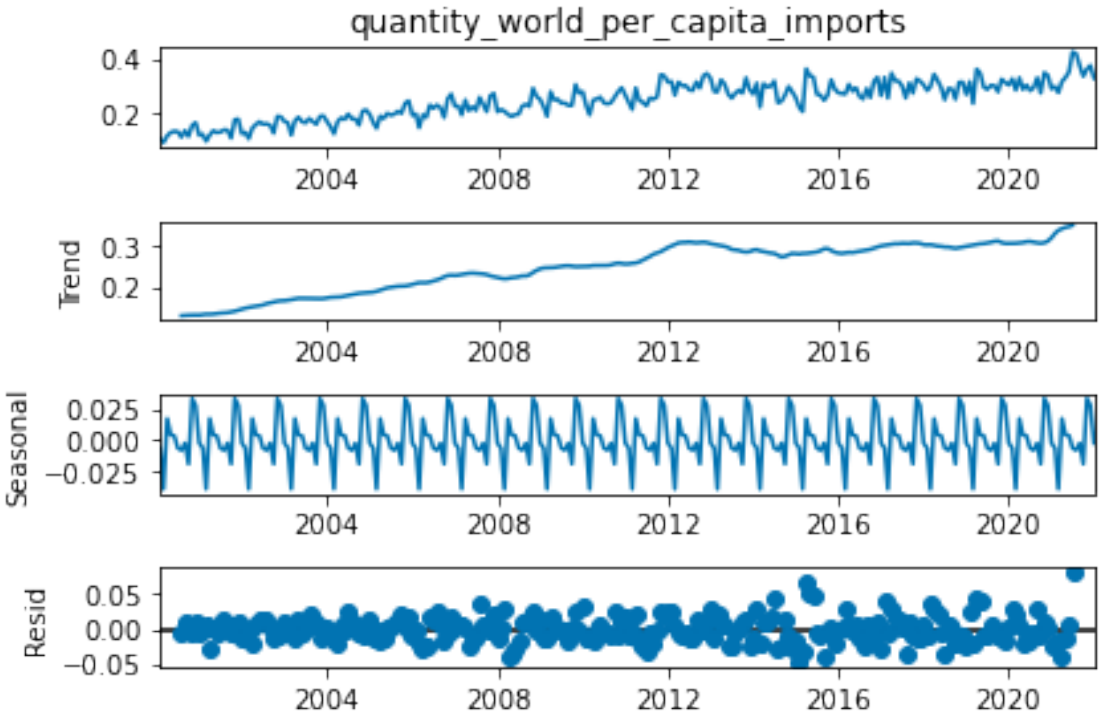
[90]:
```
adf(imports_subset2_df['quantity_world_per_capita_imports'].dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                   -0.972273
p-value:                   0.763242
lags:                     13.000000
observations:            250.000000
critical value (1%):      -3.456781
critical value (5%):      -2.873172
critical value (10%):     -2.572969
dtype: float64
```

[91]:
```
decompose_result_mult =␣
 ↪seasonal_decompose(imports_subset2_df['quantity_world_per_capita_imports'].
 ↪dropna(), model='additive')

trend = decompose_result_mult.trend
seasonal = decompose_result_mult.seasonal
residual = decompose_result_mult.resid

decompose_result_mult.plot()
```

[91]:


quantity_world_per_capita_imports
0.4
0.2
2004
2008
2012
2016
2020
Trend
0.3
0.2
Seasonal
0.025
0.000
−0.025
Resid
0.05
0.00
−0.05


quantity_world_per_capita_imports
0.4
0.2
2004
2008
2012
2016
2020
Trend
0.3
0.2
Seasonal
0.025
0.000
−0.025
Resid
0.05
0.00
−0.05

[92]:
```
imports_subset2_df['quantity_world_per_capita_imports_diff1'] =␣
↪imports_subset2_df['quantity_world_per_capita_imports'] -␣
↪imports_subset2_df['quantity_world_per_capita_imports'].shift(1)
imports_subset2_df['quantity_world_per_capita_imports_diff1_diff12'] =␣
↪imports_subset2_df['quantity_world_per_capita_imports_diff1'] -␣
↪imports_subset2_df['quantity_world_per_capita_imports_diff1'].shift(12)
adf(imports_subset2_df['quantity_world_per_capita_imports_diff1_diff12'].
↪dropna())
```

```
Augmented Dickey-Fuller Test:
t-stat:                  -6.062868e+00
p-value:                  1.200914e-07
lags:                     1.600000e+01
observations:             2.340000e+02
critical value (1%):     -3.458608e+00
critical value (5%):     -2.873972e+00
critical value (10%):    -2.573396e+00
dtype: float64
```

[93]:
```
ts_df = ts_df.
↪merge(imports_subset2_df['quantity_world_per_capita_imports_diff1_diff12'],␣
↪left_index=True, right_index=True)
```

Now we've made everything stationary for the multivariate time series.

[94]:
```
display(ts_df.dropna().head())
```

```
            frspger_25  price_adj_diff1  \
month
2001-05-31           0         0.659528
2001-06-30           0         0.242013
2001-07-31           0        -0.434246
2001-08-31           0         0.380452
2001-09-30           0        -0.242006

            bottled_total_per_capita_diff1_diff12   bulk_diff12  \
month
2001-05-31                               0.016915  6.595232e+06
2001-06-30                              -0.050785  5.953839e+06
2001-07-31                               0.082204 -1.084195e+07
2001-08-31                              -0.062038  7.302612e+07
2001-09-30                              -0.058162 -3.044768e+07

            bulk_per_capita_diff12  nonbasic_quantity_diff1_diff12  \
month
2001-05-31                0.022032                    9.853625e+06
2001-06-30                0.019816                   -8.823954e+06
2001-07-31               -0.039632                    4.362194e+05
2001-08-31                0.250287                    7.337013e+07
```

```
2001-09-30              -0.128668                   -1.192142e+08

            nonbasic_quantity_per_capita_diff1_diff12  \
month
2001-05-31                                   0.034201
2001-06-30                                  -0.031054
2001-07-31                                   0.001369
2001-08-31                                   0.251675
2001-09-30                                  -0.432685

            quantity_exports_diff12  fas_value_adj_exports_diff1_diff12  \
month
2001-05-31                7204114.0                        1.040327e+06
2001-06-30                 998658.0                       -7.739911e+06
2001-07-31                7048878.0                        8.425537e+06
2001-08-31                 295126.0                       -1.252128e+07
2001-09-30                -992607.0                       -1.852250e+06

            customs_value_adj_ukfrspde_imports_diff1_diff12  \
month
2001-05-31                                     8.508741e+06
2001-06-30                                    -5.426225e+06
2001-07-31                                     7.139594e+06
2001-08-31                                    -1.061431e+07
2001-09-30                                     1.310284e+07

            calculated_duties_adj_ukfrspde_imports_diff1_diff12  \
month
2001-05-31                                      176039.000669
2001-06-30                                      -75155.816947
2001-07-31                                      122605.114091
2001-08-31                                     -192850.754169
2001-09-30                                       63825.896916

            charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12  \
month
2001-05-31                                      189967.702126
2001-06-30                                     -283322.796341
2001-07-31                                      565432.829575
2001-08-31                                     -609574.553977
2001-09-30                                     -297341.446600

            quantity_ukfrspde_imports_diff1_diff12  \
month
2001-05-31                               1441077.0
2001-06-30                              -1241802.0
2001-07-31                               1692652.0
2001-08-31                              -1839244.0
```

```
2001-09-30                                  432505.0

            quantity_ukfrspde_proportion_imports_diff12_diff1  \
month
2001-05-31                                           0.060835
2001-06-30                                          -0.040281
2001-07-31                                          -0.018479
2001-08-31                                           0.011007
2001-09-30                                           0.027586

            quantity_ukfrspde_per_capita_imports_diff1_diff12  \
month
2001-05-31                                           0.005072
2001-06-30                                          -0.004398
2001-07-31                                           0.005988
2001-08-31                                          -0.006486
2001-09-30                                           0.001558

            customs_value_adj_world_imports_diff1_diff12  \
month
2001-05-31                                -7.017318e+06
2001-06-30                                 2.364912e+05
2001-07-31                                 2.828532e+07
2001-08-31                                -3.052870e+07
2001-09-30                                 5.153565e+06

            calculated_duties_adj_world_imports  \
month
2001-05-31                       2.579096e+06
2001-06-30                       2.717220e+06
2001-07-31                       2.990120e+06
2001-08-31                       2.938348e+06
2001-09-30                       2.574896e+06

            charges_insurance_freight_adj_world_imports_diff1_diff12  \
month
2001-05-31                                     -8.889423e+05
2001-06-30                                      1.200196e+03
2001-07-31                                      1.895395e+06
2001-08-31                                     -2.047266e+06
2001-09-30                                     -9.266976e+05

            quantity_world_imports_diff1_diff12  \
month
2001-05-31                          -3193275.0
2001-06-30                           1031146.0
2001-07-31                           7704403.0
2001-08-31                          -7049077.0
```

```
2001-09-30                              -2195867.0
```

```
            quantity_world_per_capita_imports_diff1_diff12
month
2001-05-31                                       -0.011262
2001-06-30                                        0.003642
2001-07-31                                        0.027200
2001-08-31                                       -0.024939
2001-09-30                                       -0.007474
```

1.4.5 Johansen Test

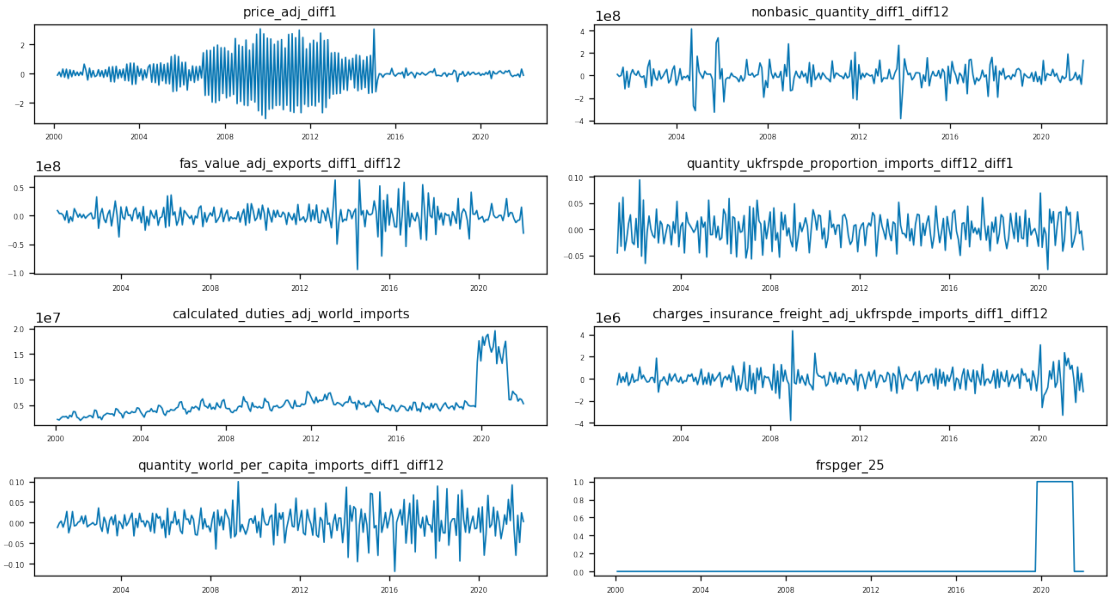
[95]:
```
johan_test_df = ts_df[['price_adj_diff1',
    'nonbasic_quantity_diff1_diff12',
    'fas_value_adj_exports_diff1_diff12',
    'quantity_ukfrspde_proportion_imports_diff12_diff1',
    'calculated_duties_adj_world_imports',
    'charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12',
    'quantity_world_per_capita_imports_diff1_diff12',
    'frspger_25'
    ]]
```

[96]:
```
johan_test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 264 entries, 2000-01-31 to 2021-12-31
Data columns (total 8 columns):
 #   Column                                                  Non-Null Count
Dtype
---  ------                                                  --------------
-----
 0   price_adj_diff1                                         263 non-null
float64
 1   nonbasic_quantity_diff1_diff12                          246 non-null
float64
 2   fas_value_adj_exports_diff1_diff12                      251 non-null
float64
 3   quantity_ukfrspde_proportion_imports_diff12_diff1       251 non-null
float64
 4   calculated_duties_adj_world_imports                     264 non-null
float64
 5   charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12  251 non-null
float64
 6   quantity_world_per_capita_imports_diff1_diff12          251 non-null
float64
 7   frspger_25                                              264 non-null
int64
dtypes: float64(7), int64(1)
```

```
memory usage: 26.7 KB
```

[97]:
```
fig, axes = plt.subplots(nrows=4, ncols=2, dpi=120, figsize=(11,6))
for i, ax in enumerate(axes.flatten()):
    data = johan_test_df[johan_test_df.columns[i]]
    ax.plot(data, linewidth=1)

    ax.set_title(johan_test_df.columns[i], size=9)
    ax.tick_params(labelsize=5)

plt.tight_layout()
plt.show()
```



Let's define a function that provides a printout of the results of a cointegration johansen test of the variables for our analysis.

[98]:
```
def johansen_test(var_df, critical_value=0.05):
    output = coint_johansen(var_df, -1, 5)
    d = {'0.90': 0, '0.95': 1, '0.99': 2}
    trace_stat = output.lr1
    critical_values = output.cvt[:, d[str(1-critical_value)]]

    print('Cointegration Johansen Test:')
    print('{:<62}'.format('Variable') + '{:<30}'.format('T-Stat > Critical␣
 ↪Values') + '{:<20}'.format('Significant'))
    print('--'*52)
    for col, trace, cvt in zip(var_df.columns, trace_stat, critical_values):
```

```
        print('{:<62}'.format(col) + '{:<30}'.format('{:<7}'.
↪format(format(trace, '.3f')) + ' > ' + '{:<7}'.format(format(cvt, '.3f'))) +␣
↪'{:<20}'.format(str(trace > cvt)))
```

[99]:
```
# drop boolean datapoint
johan_test_df = johan_test_df.drop(columns=['frspger_25'])

johansen_test(johan_test_df.dropna())
```

```
Cointegration Johansen Test:
Variable                                                        T-Stat > Critical
Values      Significant
------------------------------------------------------------------------------
------------------------
price_adj_diff1                                                 506.291 > 111.780
True
nonbasic_quantity_diff1_diff12                                  389.851 > 83.938
True
fas_value_adj_exports_diff1_diff12                              284.409 > 60.063
True
quantity_ukfrspde_proportion_imports_diff12_diff1               192.440 > 40.175
True
calculated_duties_adj_world_imports                             106.238 > 24.276
True
charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12     51.310  > 12.321
True
quantity_world_per_capita_imports_diff1_diff12                  1.311   > 4.130
False
```

1.5 Modeling

Before modeling, we need to do a train-test split to be able to forecast on the data. I should have done this much earlier on and used only the training data for checking for stationarity and granger causality. However, we're here now. New data will be published soon and we'll be able to incorporate that into the a new test set. For now, let's just pull out the most recent few observations for the test set (the most recent 3 months of data).

[100]:
```
obs = 3
ts_train, ts_test = johan_test_df.dropna()[0:-obs], johan_test_df.dropna()[-obs:
↪]
```

[101]:
```
ts_train
```

[101]:

month	price_adj_diff1	nonbasic_quantity_diff1_diff12	\
2001-05-31	0.659528	9.853625e+06	
2001-06-30	0.242013	-8.823954e+06	
2001-07-31	-0.434246	4.362194e+05	

```
2001-08-31         0.380452                    7.337013e+07
2001-09-30        -0.242006                   -1.192142e+08
…                       …                            …
2021-03-31         0.115665                   -4.419897e+07
2021-04-30         0.230637                   -3.726761e+07
2021-05-31        -0.083969                   -1.932481e+07
2021-06-30        -0.200000                    2.699895e+07
2021-07-31        -0.112057                   -3.781958e+07

            fas_value_adj_exports_diff1_diff12  \
month
2001-05-31                        1.040327e+06
2001-06-30                       -7.739911e+06
2001-07-31                        8.425537e+06
2001-08-31                       -1.252128e+07
2001-09-30                       -1.852250e+06
…                                       …
2021-03-31                        3.008267e+07
2021-04-30                       -1.641979e+06
2021-05-31                        5.475490e+06
2021-06-30                        4.892883e+06
2021-07-31                       -5.591900e+06

            quantity_ukfrspde_proportion_imports_diff12_diff1  \
month
2001-05-31                                           0.060835
2001-06-30                                          -0.040281
2001-07-31                                          -0.018479
2001-08-31                                           0.011007
2001-09-30                                           0.027586
…                                                       …
2021-03-31                                           0.042747
2021-04-30                                           0.027219
2021-05-31                                           0.032440
2021-06-30                                          -0.034139
2021-07-31                                          -0.022809

            calculated_duties_adj_world_imports  \
month
2001-05-31                         2.579096e+06
2001-06-30                         2.717220e+06
2001-07-31                         2.990120e+06
2001-08-31                         2.938348e+06
2001-09-30                         2.574896e+06
…                                         …
2021-03-31                         1.160764e+07
2021-04-30                         6.101831e+06
```

```
2021-05-31                          6.080915e+06
2021-06-30                          7.736218e+06
2021-07-31                          7.281041e+06

            charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12  \
month
2001-05-31                                          1.899677e+05
2001-06-30                                         -2.833228e+05
2001-07-31                                          5.654328e+05
2001-08-31                                         -6.095746e+05
2001-09-30                                         -2.973414e+05
…                                                           …
2021-03-31                                          1.199310e+06
2021-04-30                                          1.861444e+06
2021-05-31                                          8.662714e+05
2021-06-30                                          1.217117e+06
2021-07-31                                         -6.812625e+04

            quantity_world_per_capita_imports_diff1_diff12
month
2001-05-31                                       -0.011262
2001-06-30                                        0.003642
2001-07-31                                        0.027200
2001-08-31                                       -0.024939
2001-09-30                                       -0.007474
…                                                      …
2021-03-31                                       -0.004261
2021-04-30                                        0.056351
2021-05-31                                        0.004597
2021-06-30                                        0.091626
2021-07-31                                       -0.016523

[243 rows x 7 columns]
```

1.5.1 Lag Order Selection

[102]:
```
model = VAR(ts_train)
```

```
/opt/anaconda3/lib/python3.8/site-
packages/statsmodels/tsa/base/tsa_model.py:524: ValueWarning: No frequency
information was provided, so inferred frequency M will be used.
  warnings.warn('No frequency information was'
```

[103]:
```
lag_orders = model.select_order(maxlags=12)
lag_orders.summary()
```

[103]:
```
<class 'statsmodels.iolib.table.SimpleTable'>
```

It looks like the recommendation is for a lag order of 1. Let's go with that.

1.5.2 Fit the Model

[104]:
```
tsmf = model.fit(3)
tsmf.summary()
```

[104]:
```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Sat, 19, Feb, 2022
Time:                     22:43:22
--------------------------------------------------------------------
No. of Equations:         7.00000    BIC:                    110.021
Nobs:                     240.000    HQIC:                   108.688
Log likelihood:          -15164.4    FPE:                 6.50497e+46
AIC:                      107.788    Det(Omega_mle):      3.52059e+46
--------------------------------------------------------------------
Results for equation price_adj_diff1
================================================================================
================================================
                                                                  coefficient
std. error           t-stat            prob
--------------------------------------------------------------------------------
------------------------------------------------
const                                                               0.015916
0.043187            0.369           0.712
L1.price_adj_diff1                                                 -0.770097
0.066091          -11.652           0.000
L1.nonbasic_quantity_diff1_diff12                                   0.000000
0.000000            0.262           0.793
L1.fas_value_adj_exports_diff1_diff12                              -0.000000
0.000000           -1.476           0.140
L1.quantity_ukfrspde_proportion_imports_diff12_diff1               -0.558750
1.142021           -0.489           0.625
L1.calculated_duties_adj_world_imports                             -0.000000
0.000000           -0.552           0.581
L1.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12     -0.000000
0.000000           -0.063           0.949
L1.quantity_world_per_capita_imports_diff1_diff12                   0.473047
1.007933            0.469           0.639
L2.price_adj_diff1                                                 -0.033562
0.085113           -0.394           0.693
L2.nonbasic_quantity_diff1_diff12                                   0.000000
0.000000            0.144           0.885
L2.fas_value_adj_exports_diff1_diff12                              -0.000000
0.000000           -1.373           0.170
```

```
L2.quantity_ukfrspde_proportion_imports_diff12_diff1                  -0.156969
1.245603           -0.126           0.900
L2.calculated_duties_adj_world_imports                                 0.000000
0.000000            0.332           0.740
L2.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12        -0.000000
0.000000           -0.270           0.787
L2.quantity_world_per_capita_imports_diff1_diff12                      0.030526
1.146533            0.027           0.979
L3.price_adj_diff1                                                    -0.242961
0.065552           -3.706           0.000
L3.nonbasic_quantity_diff1_diff12                                     -0.000000
0.000000           -0.519           0.604
L3.fas_value_adj_exports_diff1_diff12                                 -0.000000
0.000000           -1.719           0.086
L3.quantity_ukfrspde_proportion_imports_diff12_diff1                   0.315225
1.127817            0.279           0.780
L3.calculated_duties_adj_world_imports                                 0.000000
0.000000            0.334           0.738
L3.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12        -0.000000
0.000000           -0.480           0.631
L3.quantity_world_per_capita_imports_diff1_diff12                      0.255544
1.003377            0.255           0.799
================================================================================
================================================

Results for equation nonbasic_quantity_diff1_diff12
================================================================================
================================================
coefficient        std. error           t-stat            prob
--------------------------------------------------------------------------------
------------------------------------------------
const
-5111511.767901   11611722.485505           -0.440           0.660
L1.price_adj_diff1
1115668.984279   17769922.736032            0.063           0.950
L1.nonbasic_quantity_diff1_diff12
-0.390860          0.065019           -6.011           0.000
L1.fas_value_adj_exports_diff1_diff12
0.039081          0.384537            0.102           0.919
L1.quantity_ukfrspde_proportion_imports_diff12_diff1
-141670217.650484  307054630.777165           -0.461           0.645
L1.calculated_duties_adj_world_imports
-1.299738          4.558890           -0.285           0.776
L1.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12
-11.735695         10.518319           -1.116           0.265
L1.quantity_world_per_capita_imports_diff1_diff12
89974219.019523  271002592.512322            0.332           0.740
```

```
L2.price_adj_diff1
19909241.419789   22884201.039596            0.870           0.384
L2.nonbasic_quantity_diff1_diff12
-0.311584          0.066496           -4.686           0.000
L2.fas_value_adj_exports_diff1_diff12
0.589944          0.447342            1.319           0.187
L2.quantity_ukfrspde_proportion_imports_diff12_diff1
-73970074.292114  334904850.728068           -0.221           0.825
L2.calculated_duties_adj_world_imports
1.867560          5.951371            0.314           0.754
L2.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12
-7.208763         11.180541           -0.645           0.519
L2.quantity_world_per_capita_imports_diff1_diff12
258283032.636053  308267816.697715            0.838           0.402
L3.price_adj_diff1
17287156.718374   17624810.968659            0.981           0.327
L3.nonbasic_quantity_diff1_diff12
-0.253240          0.065254           -3.881           0.000
L3.fas_value_adj_exports_diff1_diff12
0.083995          0.387425            0.217           0.828
L3.quantity_ukfrspde_proportion_imports_diff12_diff1
454082881.932832  303235562.190105            1.497           0.134
L3.calculated_duties_adj_world_imports
0.186536          4.651748            0.040           0.968
L3.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12
-3.837544         10.544972           -0.364           0.716
L3.quantity_world_per_capita_imports_diff1_diff12
338247851.975219  269777510.380507            1.254           0.210
==============================================================================
==============================================

Results for equation fas_value_adj_exports_diff1_diff12
==============================================================================
==============================================
                                                                 coefficient
std. error           t-stat            prob
------------------------------------------------------------------------------
------------------------------------------------
const                                                           26866.680981
2126527.278765            0.013           0.990
L1.price_adj_diff1                                          -3676081.809282
3254316.961751           -1.130           0.259
L1.nonbasic_quantity_diff1_diff12                               -0.006985
0.011907           -0.587           0.557
L1.fas_value_adj_exports_diff1_diff12                           -0.694014
0.070423           -9.855           0.000
L1.quantity_ukfrspde_proportion_imports_diff12_diff1        -7797701.458147
```

```
56232832.745863           -0.139           0.890
L1.calculated_duties_adj_world_imports                                    -0.192000
0.834898           -0.230           0.818
L1.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12          1.642468
1.926285            0.853           0.394
L1.quantity_world_per_capita_imports_diff1_diff12               -57272929.982329
49630397.756483           -1.154           0.249
L2.price_adj_diff1                                               -1851629.762872
4190926.697068           -0.442           0.659
L2.nonbasic_quantity_diff1_diff12                                         -0.003745
0.012178           -0.308           0.758
L2.fas_value_adj_exports_diff1_diff12                                     -0.354658
0.081924           -4.329           0.000
L2.quantity_ukfrspde_proportion_imports_diff12_diff1            -24982424.220039
61333217.509547           -0.407           0.684
L2.calculated_duties_adj_world_imports                                     0.002529
1.089912            0.002           0.998
L2.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12         -0.398519
2.047562           -0.195           0.846
L2.quantity_world_per_capita_imports_diff1_diff12               -47161358.851989
56455011.062429           -0.835           0.404
L3.price_adj_diff1                                                1664776.156503
3227741.737259            0.516           0.606
L3.nonbasic_quantity_diff1_diff12                                         -0.010717
0.011950           -0.897           0.370
L3.fas_value_adj_exports_diff1_diff12                                     -0.067587
0.070952           -0.953           0.341
L3.quantity_ukfrspde_proportion_imports_diff12_diff1            -24090942.779669
55533422.857279           -0.434           0.664
L3.calculated_duties_adj_world_imports                                     0.236561
0.851904            0.278           0.781
L3.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12          0.691952
1.931167            0.358           0.720
L3.quantity_world_per_capita_imports_diff1_diff12               -80309138.992383
49406040.812430           -1.625           0.104
==============================================================================
==============================================

Results for equation quantity_ukfrspde_proportion_imports_diff12_diff1
==============================================================================
==============================================
                                                                  coefficient
std. error           t-stat            prob
------------------------------------------------------------------------------
----------------------------------------------
const                                                             -0.000876
0.003360           -0.261           0.794
```

```
L1.price_adj_diff1                                                    0.007394
0.005142           1.438           0.150
L1.nonbasic_quantity_diff1_diff12                                     0.000000
0.000000           0.612           0.540
L1.fas_value_adj_exports_diff1_diff12                                -0.000000
0.000000          -0.432           0.666
L1.quantity_ukfrspde_proportion_imports_diff12_diff1                 -0.562307
0.088853          -6.329           0.000
L1.calculated_duties_adj_world_imports                               -0.000000
0.000000          -2.037           0.042
L1.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12         0.000000
0.000000           1.763           0.078
L1.quantity_world_per_capita_imports_diff1_diff12                    -0.023496
0.078420          -0.300           0.764
L2.price_adj_diff1                                                    0.002804
0.006622           0.423           0.672
L2.nonbasic_quantity_diff1_diff12                                     0.000000
0.000000           2.460           0.014
L2.fas_value_adj_exports_diff1_diff12                                 0.000000
0.000000           0.643           0.520
L2.quantity_ukfrspde_proportion_imports_diff12_diff1                 -0.257142
0.096912          -2.653           0.008
L2.calculated_duties_adj_world_imports                                0.000000
0.000000           0.876           0.381
L2.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12        0.000000
0.000000           0.276           0.783
L2.quantity_world_per_capita_imports_diff1_diff12                     0.063854
0.089204           0.716           0.474
L3.price_adj_diff1                                                   -0.004942
0.005100          -0.969           0.333
L3.nonbasic_quantity_diff1_diff12                                    -0.000000
0.000000          -0.097           0.922
L3.fas_value_adj_exports_diff1_diff12                                 0.000000
0.000000           0.288           0.774
L3.quantity_ukfrspde_proportion_imports_diff12_diff1                 -0.179775
0.087748          -2.049           0.040
L3.calculated_duties_adj_world_imports                                0.000000
0.000000           1.042           0.297
L3.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12        0.000000
0.000000           0.586           0.558
L3.quantity_world_per_capita_imports_diff1_diff12                     0.054921
0.078066           0.704           0.482
==============================================================================
==============================================

Results for equation calculated_duties_adj_world_imports
==============================================================================
```

```
================================================
                                                                coefficient
std. error           t-stat            prob
------------------------------------------------------------------------------
------------------------------------------------
const                                                          375160.336572
169808.259911            2.209           0.027
L1.price_adj_diff1                                            -447412.360200
259864.947885           -1.722           0.085
L1.nonbasic_quantity_diff1_diff12                                  -0.001186
0.000951           -1.247           0.212
L1.fas_value_adj_exports_diff1_diff12                               0.003861
0.005623            0.687           0.492
L1.quantity_ukfrspde_proportion_imports_diff12_diff1           676935.592993
4490325.411656            0.151           0.880
L1.calculated_duties_adj_world_imports                              0.809121
0.066669           12.136           0.000
L1.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12      -0.332286
0.153818           -2.160           0.031
L1.quantity_world_per_capita_imports_diff1_diff12           -1820572.421317
3963105.277724           -0.459           0.646
L2.price_adj_diff1                                            -183867.954302
334655.462428           -0.549           0.583
L2.nonbasic_quantity_diff1_diff12                                  -0.001507
0.000972           -1.549           0.121
L2.fas_value_adj_exports_diff1_diff12                               0.004816
0.006542            0.736           0.462
L2.quantity_ukfrspde_proportion_imports_diff12_diff1           3587568.383162
4897603.263318            0.733           0.464
L2.calculated_duties_adj_world_imports                             -0.037925
0.087032           -0.436           0.663
L2.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12      -0.111843
0.163503           -0.684           0.494
L2.quantity_world_per_capita_imports_diff1_diff12           3121206.183596
4508066.878555            0.692           0.489
L3.price_adj_diff1                                             302424.357740
257742.853016            1.173           0.241
L3.nonbasic_quantity_diff1_diff12                                  -0.000789
0.000954           -0.827           0.408
L3.fas_value_adj_exports_diff1_diff12                               0.002374
0.005666            0.419           0.675
L3.quantity_ukfrspde_proportion_imports_diff12_diff1           -227935.012057
4434475.868915           -0.051           0.959
L3.calculated_duties_adj_world_imports                              0.166977
0.068027            2.455           0.014
L3.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12      0.087215
0.154208            0.566           0.572
```

```
L3.quantity_world_per_capita_imports_diff1_diff12                1342533.644768
3945189.842240            0.340           0.734
================================================================================
================================================

Results for equation charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12
================================================================================
================================================
                                                                      coefficient
std. error           t-stat            prob
--------------------------------------------------------------------------------
------------------------------------------------
const                                                                  33954.273812
102765.063249            0.330           0.741
L1.price_adj_diff1                                                   -150652.956947
157265.835122           -0.958           0.338
L1.nonbasic_quantity_diff1_diff12                                        -0.000339
0.000575           -0.589           0.556
L1.fas_value_adj_exports_diff1_diff12                                     0.001631
0.003403            0.479           0.632
L1.quantity_ukfrspde_proportion_imports_diff12_diff1            -3414662.329888
2717468.368023           -1.257           0.209
L1.calculated_duties_adj_world_imports                                   -0.156597
0.040347           -3.881           0.000
L1.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12           -0.359888
0.093088           -3.866           0.000
L1.quantity_world_per_capita_imports_diff1_diff12               -1352546.343318
2398403.733370           -0.564           0.573
L2.price_adj_diff1                                                   -112647.762734
202527.779162           -0.556           0.578
L2.nonbasic_quantity_diff1_diff12                                         0.000307
0.000588            0.522           0.602
L2.fas_value_adj_exports_diff1_diff12                                     0.007634
0.003959            1.928           0.054
L2.quantity_ukfrspde_proportion_imports_diff12_diff1                 929623.090115
2963945.978758            0.314           0.754
L2.calculated_duties_adj_world_imports                                    0.115413
0.052670            2.191           0.028
L2.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12           -0.318866
0.098949           -3.223           0.001
L2.quantity_world_per_capita_imports_diff1_diff12                    582995.704304
2728205.201254            0.214           0.831
L3.price_adj_diff1                                                   -31801.467458
155981.579494           -0.204           0.838
L3.nonbasic_quantity_diff1_diff12                                         0.000209
0.000578            0.361           0.718
L3.fas_value_adj_exports_diff1_diff12                                     0.010311
```

```
0.003429            3.007           0.003
L3.quantity_ukfrspde_proportion_imports_diff12_diff1           -2974443.363238
2683669.177128           -1.108           0.268
L3.calculated_duties_adj_world_imports                                  0.041162
0.041169            1.000           0.317
L3.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12        -0.049839
0.093324           -0.534           0.593
L3.quantity_world_per_capita_imports_diff1_diff12              -1131504.292856
2387561.617317           -0.474           0.636
================================================================================
================================================

Results for equation quantity_world_per_capita_imports_diff1_diff12
================================================================================
================================================
                                                                     coefficient
std. error           t-stat            prob
--------------------------------------------------------------------------------
------------------------------------------------
const                                                                  -0.000049
0.003809           -0.013           0.990
L1.price_adj_diff1                                                     -0.013350
0.005830           -2.290           0.022
L1.nonbasic_quantity_diff1_diff12                                      -0.000000
0.000000           -1.385           0.166
L1.fas_value_adj_exports_diff1_diff12                                   0.000000
0.000000            1.306           0.192
L1.quantity_ukfrspde_proportion_imports_diff12_diff1                   -0.145107
0.100736           -1.440           0.150
L1.calculated_duties_adj_world_imports                                 -0.000000
0.000000           -1.496           0.135
L1.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12          0.000000
0.000000            0.051           0.960
L1.quantity_world_per_capita_imports_diff1_diff12                      -0.659773
0.088909           -7.421           0.000
L2.price_adj_diff1                                                     -0.003216
0.007508           -0.428           0.668
L2.nonbasic_quantity_diff1_diff12                                      -0.000000
0.000000           -1.681           0.093
L2.fas_value_adj_exports_diff1_diff12                                   0.000000
0.000000            1.442           0.149
L2.quantity_ukfrspde_proportion_imports_diff12_diff1                   -0.111825
0.109873           -1.018           0.309
L2.calculated_duties_adj_world_imports                                 -0.000000
0.000000           -0.525           0.599
L2.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12          0.000000
0.000000            0.035           0.972
```

```
L2.quantity_world_per_capita_imports_diff1_diff12                      -0.343311
0.101135           -3.395           0.001
L3.price_adj_diff1                                                      0.009804
0.005782            1.696           0.090
L3.nonbasic_quantity_diff1_diff12                                      -0.000000
0.000000           -1.311           0.190
L3.fas_value_adj_exports_diff1_diff12                                   0.000000
0.000000            3.622           0.000
L3.quantity_ukfrspde_proportion_imports_diff12_diff1                   -0.074487
0.099484           -0.749           0.454
L3.calculated_duties_adj_world_imports                                  0.000000
0.000000            2.269           0.023
L3.charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12          0.000000
0.000000            0.364           0.716
L3.quantity_world_per_capita_imports_diff1_diff12                      -0.187247
0.088507           -2.116           0.034
================================================================================
================================================

Correlation matrix of residuals
                                                                 price_adj_diff1
nonbasic_quantity_diff1_diff12  fas_value_adj_exports_diff1_diff12
quantity_ukfrspde_proportion_imports_diff12_diff1
calculated_duties_adj_world_imports
charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12
quantity_world_per_capita_imports_diff1_diff12
price_adj_diff1                                                          1.000000
-0.024413                          -0.048772
-0.041817                           -0.053355
-0.057014                                    -0.084999
nonbasic_quantity_diff1_diff12                                          -0.024413
1.000000                           -0.006295
0.021254                            -0.007641
0.018524                                     0.002964
fas_value_adj_exports_diff1_diff12                                      -0.048772
-0.006295                           1.000000
0.133028                            0.035530
0.280415                                     0.259451
quantity_ukfrspde_proportion_imports_diff12_diff1                       -0.041817
0.021254                           0.133028
1.000000                           -0.015431
0.405175                                    -0.318788
calculated_duties_adj_world_imports                                     -0.053355
-0.007641                           0.035530
-0.015431                            1.000000
-0.019621                                    0.044052
charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12             -0.057014
```

```
0.018524                                  0.280415
0.405175                                  -0.019621
1.000000                                              0.407009
quantity_world_per_capita_imports_diff1_diff12                                  -0.084999
0.002964                                  0.259451
-0.318788                                   0.044052
0.407009                                              1.000000
```

1.5.3 Serial Correlation of Residuals

I'll use the Durbin Watson statistic for serial correlation. The value ranges from `0` to `4` with `0` indicating a positive correlation and `4` indicating a negative correlation. The value `2` is what the test aims for.

[105]:
```
dw_output = durbin_watson(tsmf.resid)

for col, val in zip(ts_train.columns, dw_output):
    print('{:<62}'.format(str(col), ':'), round(val, 3))
```

```
price_adj_diff1                                                 1.957
nonbasic_quantity_diff1_diff12                                  2.14
fas_value_adj_exports_diff1_diff12                              2.04
quantity_ukfrspde_proportion_imports_diff12_diff1               2.063
calculated_duties_adj_world_imports                             1.994
charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12     2.032
quantity_world_per_capita_imports_diff1_diff12                  2.084
```

This looks like we don't have any serial correlation in the residuals. The only input that looks like it may have some serial correlation is the indicator variable for the additional 25% tariff.

1.5.4 Forecast

[106]:
```
ts_train.columns
```

[106]:
```
Index(['price_adj_diff1', 'nonbasic_quantity_diff1_diff12',
       'fas_value_adj_exports_diff1_diff12',
       'quantity_ukfrspde_proportion_imports_diff12_diff1',
       'calculated_duties_adj_world_imports',
       'charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12',
       'quantity_world_per_capita_imports_diff1_diff12'],
      dtype='object')
```

[107]:
```
lags = tsmf.k_ar
forecast_input = ts_test[['price_adj_diff1', 'nonbasic_quantity_diff1_diff12',
       'fas_value_adj_exports_diff1_diff12',
       'quantity_ukfrspde_proportion_imports_diff12_diff1',
       'calculated_duties_adj_world_imports',
```

```
       'charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12',
       'quantity_world_per_capita_imports_diff1_diff12']].dropna().values[-lags:
 ↪]
forecast_input
```

[107]:
```
array([[-1.35037399e-01,  3.27292104e+06, -1.21178253e+07,
        -5.17580615e-03,  7.00927776e+06, -2.15299312e+06,
        -7.97691163e-02],
       [-1.46302920e-02, -7.84754718e+07, -8.42806774e+06,
         3.32223558e-02,  5.81353470e+06,  1.06825123e+06,
         1.76854645e-02],
       [-2.32410052e-01,  1.34291529e+08, -7.36217880e+06,
        -8.11764728e-03,  6.28677720e+06, -1.05282745e+06,
        -4.84722042e-02]])
```

[108]:
```
pred = tsmf.forecast(y=forecast_input, steps=obs)
df_forecast = pd.DataFrame(pred, index=johan_test_df.index[-obs:],␣
 ↪columns=ts_train.columns)
df_forecast
```

[108]:

month	price_adj_diff1	nonbasic_quantity_diff1_diff12
2021-10-31	0.308678	-5.666268e+07
2021-11-30	-0.140625	6.329439e+06
2021-12-31	0.173194	-1.941216e+07

month	fas_value_adj_exports_diff1_diff12
2021-10-31	1.344912e+07
2021-11-30	-5.998065e+06
2021-12-31	1.421364e+06

month	quantity_ukfrspde_proportion_imports_diff12_diff1
2021-10-31	-0.016880
2021-11-30	0.009821
2021-12-31	-0.001334

month	calculated_duties_adj_world_imports
2021-10-31	6.530147e+06
2021-11-30	6.136224e+06
2021-12-31	5.986438e+06

month	charges_insurance_freight_adj_ukfrspde_imports_diff1_diff12
2021-10-31	165351.761234

```
2021-11-30                                          -38304.158790
2021-12-31                                          186515.103135

            quantity_world_per_capita_imports_diff1_diff12
month
2021-10-31                                        0.032204
2021-11-30                                       -0.014559
2021-12-31                                        0.003877
```

[109]:
```
df[['price_adj']].tail(18)
```

[109]:
```
            price_adj
month
2020-07-31   9.483945
2020-08-31   9.487003
2020-09-30   9.479693
2020-10-31   9.605364
2020-11-30   9.796169
2020-12-31   9.855172
2021-01-31   9.873563
2021-02-28  10.057515
2021-03-31  10.173180
2021-04-30  10.403817
2021-05-31  10.319847
2021-06-30  10.119847
2021-07-31  10.007790
2021-08-31   9.872753
2021-09-30   9.858122
2021-10-31   9.625712
2021-11-30   9.946201
2021-12-31   9.834599
```

Let's convert the forecasted differences to the actual amounts.

[110]:
```
def convert_forecasts(original_df, forecast_df):
    fc = forecast_df.copy()
    odf = original_df.merge(fc, how='left', left_index=True, right_index=True)

    for col in fc.columns:
        col_name_odf = str.split(col, '_diff')[0]
        col_name = col_name_odf + '_pred'
        if 'diff12' in col:
            odf[col_name] = fc[col] + odf[col_name_odf].shift(12)
            if 'diff1' in col:
                odf[col_name] = odf[col_name] + odf[col_name_odf].shift(1)
        elif 'diff1' in col:
            odf[col_name] = fc[col] + odf[col_name_odf].shift(1)
    ret_cols = []
```

```
    for col in odf.columns:
        if 'pred' in col:
            ret_cols.append(col)

    return odf[ret_cols][-obs:]
```

[111]:
```
df_results = convert_forecasts(df, df_forecast)
df_results.head()
```

[111]:

month	price_adj_pred	nonbasic_quantity_pred	fas_value_adj_exports_pred	quantity_ukfrspde_proportion_imports_pred	charges_insurance_freight_adj_ukfrspde_imports_pred	quantity_world_per_capita_imports_pred
2021-10-31	10.166800	1.263814e+09	1.969738e+08	0.386381	1.225568e+07	0.716993
2021-11-30	9.485087	1.297532e+09	1.865341e+08	0.395895	1.194401e+07	0.685081
2021-12-31	10.119395	NaN	2.018321e+08	0.374763	1.183037e+07	0.669497

1.5.5 Accuracy

[112]:
```
# accuracy metrics
def forecast_accuracy(forecast, actual):
    # mean abs percentage error
    mape = np.mean(np.abs(forecast - actual)/np.abs(actual))
    # root mean squared error
    rmse = np.mean((forecast - actual)**2)**.5
```

```
    # correlation coefficient
    corr = np.corrcoef(forecast, actual)[0,1]
    # minmax accuracy
    mins = np.amin(np.hstack([forecast[:, None], actual[:, None]]), axis=1)
    maxs = np.amax(np.hstack([forecast[:, None], actual[:, None]]), axis=1)
    minmax = 1 - np.mean(mins/maxs)

    return {'mape': mape, 'rmse': rmse, 'corr': corr, 'minmax': minmax}
```

[113]:
```
cols = [str.replace(c, '_pred', '') for c in df_results.columns]
for c in cols:
    print('\nPrediction Accuracy: ' + c)
    accuracy_prod = forecast_accuracy(df_results[c + '_pred'].values,␣
 ↪df[c][-obs:])
    for k, v in accuracy_prod.items():
        print('{:<10}'.format(k + ': ') + str(round(v,4)))
```

```
Prediction Accuracy: price_adj
mape:     0.0438
rmse:     0.4422
corr:     -0.8052
minmax:   0.0426

Prediction Accuracy: nonbasic_quantity
mape:     0.7137
rmse:     526341397.9576
corr:     nan
minmax:   nan

Prediction Accuracy: fas_value_adj_exports
mape:     1.1828
rmse:     104954413.9823
corr:     -0.918
minmax:   0.5254

Prediction Accuracy: quantity_ukfrspde_proportion_imports
mape:     0.9878
rmse:     0.1906
corr:     0.607
minmax:   0.4928

Prediction Accuracy: charges_insurance_freight_adj_ukfrspde_imports
mape:     0.8742
rmse:     5550194.6369
corr:     0.5941
minmax:   0.459
```

```
Prediction Accuracy: quantity_world_per_capita_imports
mape:     0.949
rmse:     0.336
corr:     0.5193
minmax:   0.4858
```

```
<ipython-input-112-edd26c8ed0cc>:10: FutureWarning: Support for multi-
dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in
a future version.  Convert to a numpy array before indexing instead.
  mins = np.amin(np.hstack([forecast[:, None], actual[:, None]]), axis=1)
<ipython-input-112-edd26c8ed0cc>:11: FutureWarning: Support for multi-
dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in
a future version.  Convert to a numpy array before indexing instead.
  maxs = np.amax(np.hstack([forecast[:, None], actual[:, None]]), axis=1)
```

1.6 Conclusion

The prediction accuracy looks pretty good for the Real Price variable, `price_adj`, with a RMSE of 0.44 and a minmax getting close to zero.

[ ]: