

# Import Relevant Packages and Custom Helper Functions

- Stored in `BoT_Exports/helper.py` to make code cleaner across .ipynb notebooks and .py scripts

```
In [1]: from helper import *

# Ignore FutureWarning
warnings.simplefilter(action='ignore', category=FutureWarning)
pd.set_option('mode.chained_assignment', None) # to avoid SettingWithCopyWarning
```

```
In [2]: df_export_ANALYSIS = pd.read_pickle("data/cleaned/total_export_FirstAnalysis")
display(df_export_ANALYSIS)
display(df_export_ANALYSIS.columns)
```

| class      | Agriculture | Rice   | Rubber | Durian | Other_Fruits | Horticultural_products,_n.i.e. | Anir   |
|------------|-------------|--------|--------|--------|--------------|--------------------------------|--------|
| 1995-01-01 | 584.19      | 196.61 | 229.04 | 0.54   | 1.97         |                                | 116.57 |
| 1995-02-01 | 550.96      | 186.2  | 216.29 | 0.58   | 2.6          |                                | 104.63 |
| 1995-03-01 | 605.24      | 175.64 | 238.01 | 3.29   | 4.13         |                                | 132.28 |
| 1995-04-01 | 453.51      | 107.99 | 181.01 | 11.73  | 6.4          |                                | 103.94 |
| 1995-05-01 | 587.16      | 191.15 | 216.77 | 15.06  | 12.38        |                                | 107.54 |
| ...        | ...         | ...    | ...    | ...    | ...          |                                | ...    |
| 2023-11-01 | 1552.72     | 637.49 | 361.63 | 48.69  | 191.81       |                                | 159.33 |
| 2023-12-01 | 1394.08     | 532.27 | 306.66 | 77.54  | 210.95       |                                | 129.23 |
| 2024-01-01 | 1536.73     | 602.49 | 326.87 | 48.63  | 240.71       |                                | 163.27 |
| 2024-02-01 | 1446.91     | 523.91 | 419.1  | 50.46  | 127.29       |                                | 169.21 |
| 2024-03-01 | 1576.87     | 484.76 | 444.62 | 64.81  | 159.33       |                                | 258.22 |

351 rows × 40 columns



```
Index(['Agriculture', 'Rice', 'Rubber', 'Durian', 'Other_Fruits',
       'Horticultural_products,_n.i.e.', 'Animal_products', 'Fishery',
       'Crustaceans', 'Fish', 'Cuttlefish,_squid,_octopus',
       'Fishery_products,_n.i.e.', 'Forestry', 'Mining', 'Crude_oil',
       'Mineral_products,_n.i.e.', 'Manufacturing',
       'Agro-manufacturing_Products', 'Apparels_and_Textile_Materials',
       'Footware_and_parts', 'Electronics', 'Electrical_Appliances',
       'Metal_&_Steel', 'Automotive',
       'Aircrafts,_ships,_floating_structures,_and_locomotive',
       'Machinery_&_Equipment', 'Jewellery',
       'Chemicals_&_Petro-chemical_Products', 'Petroleum_products',
       'Photographic_&_cinematographic_instruments_&_supplies',
       'Medicinal_and_surgical_equipment_and_supplies',
       'Toiletries_and_cosmetics', 'Furniture_and_parts',
       'Other_Manufacturing_products', 'Other_Exports', 'Re-Exports_1',
       'Total_Exports_(Customs_basis)', 'Coverage_Adjustment',
       'Timing_Adjustment', 'Exports,_f.o.b._(BOP_basis)'],
      dtype='object', name='class')
```

```
In [3]: #combine deseasonalised data (previously some had issues with certian timese
# import os
# COMBINED_deseasonalised_value_dict = {}

# directory = "/home/wheelfredie/scripts/BoT_Exports/data/cleaned/deseasonal

# for filename in os.listdir(directory):
#     if filename.endswith(".pickle") and "dict_deseasonalized_value" in fil
#         with open(directory + "/" + filename, "rb") as f:
#             temp_dict = pickle.load(f)
#             COMBINED_deseasonalised_value_dict.update(temp_dict)
#             print(filename)

# with open("data/cleaned/COMBINED_deseasonalised_value_dict.pkl", "wb") as
#     pickle.dump(COMBINED_deseasonalised_value_dict, f)

with open("data/cleaned/COMBINED_deseasonalised_value_dict.pkl", "rb") as f:
    COMBINED_deseasonalised_value_dict = pickle.load(f)

display(COMBINED_deseasonalised_value_dict)
```

| 'Rice': actor                   |        | observed   | seasadj    | trend    | irregular | seasonal_f |
|---------------------------------|--------|------------|------------|----------|-----------|------------|
| 1995-01-01                      | 196.61 | 178.265141 | 186.478635 | 0.955955 | 1.102908  |            |
| 1995-02-01                      | 186.20 | 205.441992 | 180.540166 | 1.137930 | 0.906339  |            |
| 1995-03-01                      | 175.64 | 181.019323 | 173.110849 | 1.045684 | 0.970283  |            |
| 1995-04-01                      | 107.99 | 117.286865 | 164.738659 | 0.711957 | 0.920734  |            |
| 1995-05-01                      | 191.15 | 200.064212 | 156.435754 | 1.278891 | 0.955443  |            |
| ...                             | ...    | ...        | ...        | ...      | ...       | ...        |
| 2023-11-01                      | 637.49 | 527.750224 | 494.900962 | 1.066375 | 1.207939  |            |
| 2023-12-01                      | 532.27 | 453.193097 | 505.805571 | 0.895983 | 1.174488  |            |
| 2024-01-01                      | 602.49 | 539.961243 | 513.968437 | 1.050573 | 1.115802  |            |
| 2024-02-01                      | 523.91 | 536.841248 | 523.435752 | 1.025611 | 0.975912  |            |
| 2024-03-01                      | 484.76 | 493.576523 | 535.992976 | 0.920864 | 0.982137  |            |
| [351 rows x 5 columns],         |        |            |            |          |           |            |
| 'Rubber': factor                |        | observed   | seasadj    | trend    | irregular | seasonal   |
| 1995-01-01                      | 229.04 | 192.088698 | 198.370291 | 0.968334 | 1.192366  |            |
| 1995-02-01                      | 216.29 | 195.091889 | 209.707634 | 0.930304 | 1.108657  |            |
| 1995-03-01                      | 238.01 | 214.174320 | 220.456802 | 0.971502 | 1.111291  |            |
| 1995-04-01                      | 181.01 | 232.583375 | 230.078404 | 1.010887 | 0.778259  |            |
| 1995-05-01                      | 216.77 | 280.782755 | 235.595643 | 1.191799 | 0.772020  |            |
| ...                             | ...    | ...        | ...        | ...      | ...       | ...        |
| 2023-11-01                      | 361.63 | 357.285263 | 340.461904 | 1.049413 | 1.012160  |            |
| 2023-12-01                      | 306.66 | 303.329300 | 353.451560 | 0.858192 | 1.010980  |            |
| 2024-01-01                      | 326.87 | 324.817541 | 364.819223 | 0.890352 | 1.006319  |            |
| 2024-02-01                      | 419.10 | 390.149911 | 373.874834 | 1.043531 | 1.074202  |            |
| 2024-03-01                      | 444.62 | 390.342579 | 380.276678 | 1.026470 | 1.139051  |            |
| [351 rows x 5 columns],         |        |            |            |          |           |            |
| 'Durian': factor                |        | observed   | seasadj    | trend    | irregular | seasonal   |
| 1995-01-01                      | 0.54   | 3.960392   | 4.407741   | 0.898508 | 0.136350  |            |
| 1995-02-01                      | 0.58   | 4.757544   | 4.438051   | 1.071990 | 0.121912  |            |
| 1995-03-01                      | 3.29   | 5.973443   | 4.432165   | 1.347748 | 0.550771  |            |
| 1995-04-01                      | 11.73  | 4.302768   | 4.446974   | 0.967572 | 2.726152  |            |
| 1995-05-01                      | 15.06  | 3.931984   | 4.471423   | 0.879358 | 3.830128  |            |
| ...                             | ...    | ...        | ...        | ...      | ...       | ...        |
| 2023-11-01                      | 48.69  | 249.311365 | 274.892655 | 0.906941 | 0.195298  |            |
| 2023-12-01                      | 77.54  | 231.338596 | 254.928560 | 0.907464 | 0.335180  |            |
| 2024-01-01                      | 48.63  | 254.311258 | 239.585180 | 1.061465 | 0.191222  |            |
| 2024-02-01                      | 50.46  | 262.254735 | 230.520720 | 1.137662 | 0.192408  |            |
| 2024-03-01                      | 64.81  | 187.690978 | 227.842494 | 0.823775 | 0.345302  |            |
| [351 rows x 5 columns],         |        |            |            |          |           |            |
| 'Other_Fruits': seasonal_factor |        | observed   | seasadj    | trend    | irregular | se         |
| 1995-01-01                      | 1.97   | 4.769296   | 6.646027   | 0.717616 | 0.413059  |            |
| 1995-02-01                      | 2.60   | 6.834375   | 7.331107   | 0.932243 | 0.380430  |            |
| 1995-03-01                      | 4.13   | 8.084921   | 7.968738   | 1.014580 | 0.510828  |            |
| 1995-04-01                      | 6.40   | 9.759477   | 8.480222   | 1.150852 | 0.655773  |            |
| 1995-05-01                      | 12.38  | 8.408256   | 8.721336   | 0.964102 | 1.472362  |            |
| ...                             | ...    | ...        | ...        | ...      | ...       | ...        |
| 2023-11-01                      | 191.81 | 194.431054 | 221.035915 | 0.879636 | 0.986519  |            |
| 2023-12-01                      | 210.95 | 203.190816 | 217.574793 | 0.933890 | 1.038187  |            |
| 2024-01-01                      | 240.71 | 250.989335 | 215.292060 | 1.165809 | 0.959045  |            |

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 2024-02-01 | 127.29 | 203.234945 | 217.047161 | 0.936363 | 0.626319 |
| 2024-03-01 | 159.33 | 217.754464 | 223.865179 | 0.972704 | 0.731696 |

[351 rows x 5 columns],  
 'Horticultural\_products,\_n.i.e.': observed seasadj tr  
 end irregular seasonal\_factor

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1995-01-01 | 116.57 | 104.869794 | 101.487538 | 1.033327 | 1.111569 |
| 1995-02-01 | 104.63 | 93.634193  | 100.278209 | 0.933744 | 1.117434 |
| 1995-03-01 | 132.28 | 103.519713 | 98.109437  | 1.055145 | 1.277824 |
| 1995-04-01 | 103.94 | 91.200984  | 95.555482  | 0.954430 | 1.139681 |
| 1995-05-01 | 107.54 | 101.720195 | 92.387003  | 1.101023 | 1.057214 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 159.33 | 199.831048 | 203.713209 | 0.980943 | 0.797324 |
| 2023-12-01 | 129.23 | 121.515259 | 187.463213 | 0.648209 | 1.063488 |
| 2024-01-01 | 163.27 | 167.330832 | 174.978106 | 0.956296 | 0.975732 |
| 2024-02-01 | 169.21 | 154.863195 | 168.324319 | 0.920029 | 1.092642 |
| 2024-03-01 | 258.22 | 195.036263 | 167.889546 | 1.161694 | 1.323959 |

[351 rows x 5 columns],  
 'Animal\_products': observed seasadj trend irregular  
 seasonal\_factor

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1995-01-01 | 39.46  | 44.439967  | 45.831093  | 0.969647 | 0.887939 |
| 1995-02-01 | 40.65  | 44.617423  | 45.674944  | 0.976847 | 0.911079 |
| 1995-03-01 | 51.90  | 50.680237  | 45.322375  | 1.118217 | 1.024068 |
| 1995-04-01 | 42.44  | 45.762553  | 44.741329  | 1.022825 | 0.927396 |
| 1995-05-01 | 44.26  | 42.693829  | 43.922808  | 0.972020 | 1.036684 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 153.77 | 150.142546 | 154.463840 | 0.972024 | 1.024160 |
| 2023-12-01 | 137.43 | 138.861057 | 156.381539 | 0.887963 | 0.989694 |
| 2024-01-01 | 154.76 | 164.845682 | 157.870525 | 1.044183 | 0.938817 |
| 2024-02-01 | 156.94 | 162.469465 | 158.642180 | 1.024125 | 0.965966 |
| 2024-03-01 | 165.14 | 157.057912 | 158.590992 | 0.990333 | 1.051459 |

[351 rows x 5 columns],  
 'Crustaceans': observed seasadj trend irregular sea  
 sonal\_factor

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1995-01-01 | 152.20 | 181.203725 | 178.292273 | 1.016330 | 0.839939 |
| 1995-02-01 | 120.85 | 170.793414 | 178.849027 | 0.954959 | 0.707580 |
| 1995-03-01 | 176.01 | 194.456417 | 178.935131 | 1.086743 | 0.905139 |
| 1995-04-01 | 146.65 | 164.048783 | 179.359795 | 0.914635 | 0.893941 |
| 1995-05-01 | 177.62 | 176.002336 | 180.007450 | 0.977750 | 1.009191 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 67.61  | 62.264018  | 63.433643  | 0.981561 | 1.085860 |
| 2023-12-01 | 62.49  | 67.053223  | 66.341121  | 1.010734 | 0.931946 |
| 2024-01-01 | 57.60  | 77.608888  | 68.890558  | 1.126553 | 0.742183 |
| 2024-02-01 | 58.28  | 75.691636  | 70.614082  | 1.071906 | 0.769966 |
| 2024-03-01 | 66.04  | 69.060681  | 71.431317  | 0.966812 | 0.956260 |

[351 rows x 5 columns],  
 'Fish': observed seasadj trend irregular seasonal\_fac  
 tor

|            |       |           |           |          |          |
|------------|-------|-----------|-----------|----------|----------|
| 1995-01-01 | 33.27 | 33.484448 | 34.262670 | 0.977287 | 0.993596 |
| 1995-02-01 | 30.25 | 34.803085 | 34.628606 | 1.005039 | 0.869176 |
| 1995-03-01 | 37.45 | 35.929433 | 35.188288 | 1.021062 | 1.042321 |
| 1995-04-01 | 30.94 | 32.092303 | 35.799669 | 0.896441 | 0.964094 |
| 1995-05-01 | 35.12 | 36.306830 | 36.459932 | 0.995801 | 0.967311 |

|                               |        |            |            |          |          |         |
|-------------------------------|--------|------------|------------|----------|----------|---------|
| ...                           | ...    | ...        | ...        | ...      | ...      | ...     |
| 2023-11-01                    | 37.64  | 35.176976  | 35.918243  | 0.979362 | 1.070018 |         |
| 2023-12-01                    | 33.40  | 33.885249  | 36.733420  | 0.922464 | 0.985680 |         |
| 2024-01-01                    | 40.46  | 45.194069  | 37.780551  | 1.196226 | 0.895250 |         |
| 2024-02-01                    | 33.84  | 37.868117  | 38.879445  | 0.973988 | 0.893628 |         |
| 2024-03-01                    | 42.83  | 40.689385  | 39.790368  | 1.022594 | 1.052609 |         |
| <br>[351 rows x 5 columns],   |        |            |            |          |          |         |
| 'Cuttlefish,_squid,_octopus': |        |            |            |          | observed | seasadj |
| regular seasonal_factor       |        |            |            |          |          |         |
| 1995-01-01                    | 23.25  | 29.718890  | 28.145450  | 1.055904 | 0.782331 |         |
| 1995-02-01                    | 24.12  | 27.860078  | 27.086428  | 1.028562 | 0.865755 |         |
| 1995-03-01                    | 30.05  | 30.581646  | 25.970615  | 1.177548 | 0.982616 |         |
| 1995-04-01                    | 21.91  | 23.273663  | 24.990717  | 0.931292 | 0.941407 |         |
| 1995-05-01                    | 24.95  | 23.956593  | 24.361498  | 0.983379 | 1.041467 |         |
| ...                           | ...    | ...        | ...        | ...      | ...      | ...     |
| 2023-11-01                    | 17.27  | 20.993389  | 22.667390  | 0.926149 | 0.822640 |         |
| 2023-12-01                    | 17.74  | 18.801511  | 23.235059  | 0.809187 | 0.943541 |         |
| 2024-01-01                    | 24.13  | 26.486435  | 23.790318  | 1.113328 | 0.911032 |         |
| 2024-02-01                    | 23.13  | 24.475575  | 24.256445  | 1.009034 | 0.945024 |         |
| 2024-03-01                    | 30.92  | 25.354145  | 24.578098  | 1.031575 | 1.219524 |         |
| <br>[351 rows x 5 columns],   |        |            |            |          |          |         |
| 'Fishery_products,_n.i.e.':   |        |            |            |          | observed | seasadj |
| ular seasonal_factor          |        |            |            |          |          |         |
| 1995-01-01                    | 3.70   | 3.976707   | 4.183499   | 0.950569 | 0.930418 |         |
| 1995-02-01                    | 3.68   | 3.998833   | 4.134897   | 0.967094 | 0.920269 |         |
| 1995-03-01                    | 4.76   | 4.432753   | 4.048388   | 1.094943 | 1.073825 |         |
| 1995-04-01                    | 4.74   | 4.765673   | 3.923586   | 1.214622 | 0.994613 |         |
| 1995-05-01                    | 3.76   | 3.488092   | 3.764040   | 0.926688 | 1.077953 |         |
| ...                           | ...    | ...        | ...        | ...      | ...      | ...     |
| 2023-11-01                    | 8.93   | 9.393188   | 8.552462   | 1.098302 | 0.950689 |         |
| 2023-12-01                    | 7.99   | 8.354163   | 8.548816   | 0.977230 | 0.956409 |         |
| 2024-01-01                    | 7.44   | 8.849094   | 8.431475   | 1.049531 | 0.840764 |         |
| 2024-02-01                    | 6.80   | 7.076978   | 8.272600   | 0.855472 | 0.960862 |         |
| 2024-03-01                    | 9.10   | 8.193094   | 8.140812   | 1.006422 | 1.110692 |         |
| <br>[351 rows x 5 columns],   |        |            |            |          |          |         |
| 'Forestry':                   |        |            |            |          | observed | seasadj |
| al_factor                     |        |            |            |          |          |         |
| 1995-01-01                    | 6.42   | 7.459445   | 7.476140   | 0.997767 | 0.860654 |         |
| 1995-02-01                    | 6.76   | 7.736112   | 7.706808   | 1.003802 | 0.873824 |         |
| 1995-03-01                    | 9.14   | 7.499167   | 8.101776   | 0.925620 | 1.218802 |         |
| 1995-04-01                    | 9.83   | 11.628200  | 8.546533   | 1.360575 | 0.845359 |         |
| 1995-05-01                    | 8.56   | 8.241011   | 9.053882   | 0.910218 | 1.038708 |         |
| ...                           | ...    | ...        | ...        | ...      | ...      | ...     |
| 2023-11-01                    | 134.74 | 144.489822 | 157.134588 | 0.919529 | 0.932522 |         |
| 2023-12-01                    | 148.04 | 164.417518 | 158.275435 | 1.038806 | 0.900391 |         |
| 2024-01-01                    | 145.39 | 159.990235 | 159.335117 | 1.004112 | 0.908743 |         |
| 2024-02-01                    | 167.74 | 162.843166 | 161.107259 | 1.010775 | 1.030071 |         |
| 2024-03-01                    | 191.10 | 152.852057 | 164.060967 | 0.931678 | 1.250229 |         |
| <br>[351 rows x 5 columns],   |        |            |            |          |          |         |
| 'Crude_oil':                  |        |            |            |          | observed | seasadj |
| l_factor                      |        |            |            |          |          |         |
| 1996-11-01                    | 6.75   | 7.574363   | 8.505354   | 0.890541 | 0.891164 |         |

## 1\_composite\_export\_value\_analysis

|            |       |           |           |          |          |
|------------|-------|-----------|-----------|----------|----------|
| 1996-12-01 | 8.86  | 8.276927  | 8.618530  | 0.960364 | 1.070446 |
| 1997-01-01 | 7.97  | 8.376525  | 8.801276  | 0.951740 | 0.951469 |
| 1997-02-01 | 14.30 | 18.514280 | 8.883700  | 2.084073 | 0.772377 |
| 1997-03-01 | 5.72  | 5.817404  | 8.978198  | 0.647948 | 0.983256 |
| ...        | ...   | ...       | ...       | ...      | ...      |
| 2023-11-01 | 57.49 | 58.886639 | 42.624698 | 1.381515 | 0.976283 |
| 2023-12-01 | 25.34 | 25.356326 | 43.199409 | 0.586960 | 0.999356 |
| 2024-01-01 | 23.93 | 20.270069 | 43.286035 | 0.468282 | 1.180558 |
| 2024-02-01 | 50.36 | 46.560456 | 42.978426 | 1.083345 | 1.081605 |
| 2024-03-01 | 36.38 | 32.630540 | 42.401481 | 0.769561 | 1.114906 |

|                             |       |           |           |          |          |
|-----------------------------|-------|-----------|-----------|----------|----------|
| [329 rows x 5 columns],     |       |           |           |          |          |
| 'Mineral_products,_n.i.e.': |       |           | observed  | seasadj  | trend    |
| egular seasonal_factor      |       |           |           |          | irr      |
| 1995-01-01                  | 22.02 | 24.752151 | 23.198225 | 1.066985 | 0.889620 |
| 1995-02-01                  | 20.88 | 19.699637 | 23.195050 | 0.849303 | 1.059918 |
| 1995-03-01                  | 19.80 | 18.914190 | 23.259030 | 0.813198 | 1.046833 |
| 1995-04-01                  | 31.93 | 26.161231 | 23.472592 | 1.114544 | 1.220508 |
| 1995-05-01                  | 22.58 | 24.566381 | 23.610870 | 1.040469 | 0.919142 |
| ...                         | ...   | ...       | ...       | ...      | ...      |
| 2023-11-01                  | 48.03 | 48.870382 | 52.506238 | 0.930754 | 0.982804 |
| 2023-12-01                  | 50.59 | 54.435088 | 53.460241 | 1.018235 | 0.929364 |
| 2024-01-01                  | 50.45 | 56.628818 | 54.279139 | 1.043289 | 0.890889 |
| 2024-02-01                  | 62.93 | 62.474226 | 54.930274 | 1.137337 | 1.007295 |
| 2024-03-01                  | 56.95 | 50.595780 | 55.447077 | 0.912506 | 1.125588 |

|                         |        |            |            |          |             |
|-------------------------|--------|------------|------------|----------|-------------|
| [351 rows x 5 columns], |        |            |            |          |             |
| 'Other_Exports':        |        |            | observed   | seasadj  | trend       |
| easonal_factor          |        |            |            |          | irregular s |
| 1995-01-01              | 50.89  | 48.766072  | 51.340328  | 0.949859 | 1.043553    |
| 1995-02-01              | 40.87  | 44.069222  | 53.847876  | 0.818402 | 0.927405    |
| 1995-03-01              | 67.21  | 62.599140  | 57.199729  | 1.094396 | 1.073657    |
| 1995-04-01              | 43.43  | 45.458114  | 59.992941  | 0.757724 | 0.955385    |
| 1995-05-01              | 62.26  | 64.298088  | 62.531128  | 1.028257 | 0.968303    |
| ...                     | ...    | ...        | ...        | ...      | ...         |
| 2023-11-01              | 266.27 | 415.072011 | 592.846084 | 0.700135 | 0.641503    |
| 2023-12-01              | 454.12 | 746.698984 | 603.888891 | 1.236484 | 0.608170    |
| 2024-01-01              | 493.70 | 692.673405 | 598.250608 | 1.157832 | 0.712746    |
| 2024-02-01              | 771.18 | 746.249096 | 577.322638 | 1.292603 | 1.033408    |
| 2024-03-01              | 463.03 | 388.970416 | 546.917156 | 0.711205 | 1.190399    |

|                         |       |           |           |          |               |
|-------------------------|-------|-----------|-----------|----------|---------------|
| [351 rows x 5 columns], |       |           |           |          |               |
| 'Re-Exports_1/':        |       |           | observed  | seasadj  | trend         |
| sonal_factor            |       |           |           |          | irregular sea |
| 1995-01-01              | 11.44 | 11.145798 | 7.352783  | 1.515861 | 1.026396      |
| 1995-02-01              | 4.80  | 5.899548  | 7.329847  | 0.804866 | 0.813622      |
| 1995-03-01              | 6.24  | 7.256359  | 7.320603  | 0.991224 | 0.859935      |
| 1995-04-01              | 7.11  | 7.260131  | 7.321362  | 0.991637 | 0.979321      |
| 1995-05-01              | 6.44  | 6.210331  | 7.357487  | 0.844083 | 1.036982      |
| ...                     | ...   | ...       | ...       | ...      | ...           |
| 2023-11-01              | 27.45 | 23.999586 | 16.293780 | 1.472929 | 1.143770      |
| 2023-12-01              | 11.43 | 12.964012 | 16.036918 | 0.808386 | 0.881671      |
| 2024-01-01              | 21.75 | 17.256317 | 15.726393 | 1.097284 | 1.260408      |
| 2024-02-01              | 13.40 | 14.683585 | 15.423437 | 0.952031 | 0.912584      |
| 2024-03-01              | 7.65  | 5.642510  | 15.122486 | 0.373121 | 1.355780      |

[351 rows x 5 columns],  
 'Total\_Exports\_(Customs\_basis)': observed seasadj  
 trend irregular seasonal\_factor

|            |          |              | observed     | seasadj  |
|------------|----------|--------------|--------------|----------|
| 1995-01-01 | 3995.51  | 4301.693482  | 4288.653453  | 1.003041 |
| 1995-02-01 | 3995.11  | 4262.542221  | 4370.725493  | 0.975248 |
| 1995-03-01 | 5188.91  | 4975.331390  | 4470.842562  | 1.112840 |
| 1995-04-01 | 4065.80  | 4461.954139  | 4581.907128  | 0.973820 |
| 1995-05-01 | 4880.42  | 4763.563971  | 4690.159987  | 1.015651 |
| ...        | ...      | ...          | ...          | ...      |
| 2023-11-01 | 23479.71 | 24122.577545 | 24193.857603 | 0.997054 |
| 2023-12-01 | 22791.55 | 23835.438442 | 24093.963796 | 0.989270 |
| 2024-01-01 | 22649.88 | 24324.217276 | 23907.123189 | 1.017446 |
| 2024-02-01 | 23384.86 | 23993.592370 | 23674.001751 | 1.013500 |
| 2024-03-01 | 24960.55 | 22934.207418 | 23469.551436 | 0.977190 |

[351 rows x 5 columns],  
 'Sugar': observed seasadj trend irregular seasonal\_factor

|            |        | observed   | seasadj    | trend    | irregular | seasonal_factor |
|------------|--------|------------|------------|----------|-----------|-----------------|
| 1995-01-01 | 52.60  | 41.989779  | 56.613824  | 0.741688 | 1.252686  |                 |
| 1995-02-01 | 82.98  | 46.562385  | 78.813038  | 0.590795 | 1.782125  |                 |
| 1995-03-01 | 192.59 | 91.033802  | 100.130590 | 0.909151 | 2.115588  |                 |
| 1995-04-01 | 198.44 | 143.936104 | 119.552781 | 1.203954 | 1.378667  |                 |
| 1995-05-01 | 242.68 | 149.099902 | 132.200683 | 1.127830 | 1.627634  |                 |
| ...        | ...    | ...        | ...        | ...      | ...       | ...             |
| 2023-11-01 | 239.89 | 326.243025 | 293.894757 | 1.110068 | 0.735311  |                 |
| 2023-12-01 | 172.22 | 285.162382 | 290.548260 | 0.981463 | 0.603937  |                 |
| 2024-01-01 | 261.95 | 291.850306 | 289.504936 | 1.008101 | 0.897549  |                 |
| 2024-02-01 | 350.90 | 273.702493 | 290.790268 | 0.941237 | 1.282049  |                 |
| 2024-03-01 | 375.59 | 304.053241 | 294.231731 | 1.033380 | 1.235277  |                 |

[351 rows x 5 columns],  
 'Fish,\_canned,\_prepared,\_or\_preserved': observed seasadj  
 trend irregular seasonal\_factor

|            |        |            | observed   | seasadj  |
|------------|--------|------------|------------|----------|
| 1995-01-01 | 68.16  | 75.515107  | 73.723153  | 1.024307 |
| 1995-02-01 | 67.12  | 72.325543  | 73.203174  | 0.988011 |
| 1995-03-01 | 77.62  | 73.727627  | 72.639878  | 1.014975 |
| 1995-04-01 | 64.69  | 68.119149  | 72.093832  | 0.944868 |
| 1995-05-01 | 74.19  | 71.900565  | 71.510311  | 1.005457 |
| ...        | ...    | ...        | ...        | ...      |
| 2023-11-01 | 270.30 | 254.145357 | 243.302431 | 1.044566 |
| 2023-12-01 | 237.37 | 242.166663 | 249.272467 | 0.971494 |
| 2024-01-01 | 231.05 | 256.002074 | 253.934842 | 1.008141 |
| 2024-02-01 | 268.64 | 270.237492 | 257.061586 | 1.051256 |
| 2024-03-01 | 262.02 | 248.876263 | 258.720735 | 0.961949 |

|                         |  |                                 |          |
|-------------------------|--|---------------------------------|----------|
| [351 rows x 5 columns], | 'Crustaceans_canned,_prepared,_or_preserved':                | observed                        | seasadj  |
| dj                      | trend irregular seasonal_factor                              |                                 |          |
| 1995-01-01              | 46.27 61.089155 62.006632                                    | 0.985204                        | 0.757418 |
| 1995-02-01              | 41.93 61.313029 62.819160                                    | 0.976024                        | 0.683868 |
| 1995-03-01              | 50.92 76.178292 63.676151                                    | 1.196339                        | 0.668432 |
| 1995-04-01              | 41.59 63.482014 64.439946                                    | 0.985135                        | 0.655146 |
| 1995-05-01              | 53.66 65.759859 65.002250                                    | 1.011655                        | 0.815999 |
| ...                     | ...  | ...                             | ...      |
| 2023-11-01              | 55.53 51.345197 51.264621                                    | 1.001572                        | 1.081503 |
| 2023-12-01              | 44.12 50.623542 50.259501                                    | 1.007243                        | 0.871531 |
| 2024-01-01              | 37.89 47.498265 50.020204                                    | 0.949582                        | 0.797713 |
| 2024-02-01              | 41.96 50.364090 50.454093                                    | 0.998216                        | 0.833133 |
| 2024-03-01              | 51.55 52.798032 51.239210                                    | 1.030422                        | 0.976362 |
| [351 rows x 5 columns], | 'Meat_of_poultry,_canned,_prepared,_or_preserved':           | observed                        |          |
| seasadj                 | trend irregular seasonal_factor                              |                                 |          |
| 1995-01-01              | 5.45 6.708869 6.660712                                       | 1.007230                        | 0.812358 |
| 1995-02-01              | 5.30 6.610690 6.743318                                       | 0.980332                        | 0.801732 |
| 1995-03-01              | 9.97 10.872629 10.904683                                     | 0.997060                        | 0.916982 |
| 1995-04-01              | 8.30 9.380218 11.047994                                      | 0.849043                        | 0.884841 |
| 1995-05-01              | 13.03 11.787394 11.144731                                    | 1.057665                        | 1.105418 |
| ...                     | ...  | ...                             | ...      |
| 2023-11-01              | 251.26 243.201210 242.701007                                 | 1.002061                        | 1.033136 |
| 2023-12-01              | 218.31 240.826521 242.795769                                 | 0.991889                        | 0.906503 |
| 2024-01-01              | 242.34 247.808343 242.549927                                 | 1.021680                        | 0.977933 |
| 2024-02-01              | 230.19 238.542183 242.404684                                 | 0.984066                        | 0.964987 |
| 2024-03-01              | 251.47 238.334831 242.402878                                 | 0.983218                        | 1.055112 |
| [351 rows x 5 columns], | 'Tapioca_&cassava_starch':                                   | observed                        | seasadj  |
| rregular                | seasonal_factor  | trend                           | i        |
| 1995-01-01              | 26.45 22.329353  | 16.279629                       | 1.371613 |
| 1995-02-01              | 17.67 15.772960  | 15.050518                       | 1.048001 |
| 1995-03-01              | 11.43 9.576113   | 13.602139                       | 0.704015 |
| 1995-04-01              | 10.76 11.749469  | 12.080432                       | 0.972603 |
| 1995-05-01              | 9.09 10.561313   | 10.621249                       | 0.994357 |
| ...                     | ...  | ...                             | ...      |
| 2023-11-01              | 155.92 144.864796  | 148.245564                      | 0.977195 |
| 2023-12-01              | 108.03 108.939471  | 150.269611                      | 0.724960 |
| 2024-01-01              | 126.88 140.539833  | 153.147070                      | 0.917679 |
| 2024-02-01              | 196.47 168.696079  | 156.475585                      | 1.078098 |
| 2024-03-01              | 190.09 156.720637  | 158.983173                      | 0.985769 |
| [351 rows x 5 columns], | 'Other_fruits_& vegetables,_canned,_prepared,_or_preserved': | ob                              |          |
| served                  | seasadj  | trend irregular seasonal_factor |          |
| 1995-01-01              | 23.52 27.568279  | 29.402266                       | 0.937624 |
| 1995-02-01              | 23.62 30.976414  | 29.621328                       | 1.045747 |
| 1995-03-01              | 28.40 29.904885  | 29.924762                       | 0.999336 |
| 1995-04-01              | 23.30 29.357368  | 30.219199                       | 0.971481 |
| 1995-05-01              | 30.95 32.730097  | 30.479313                       | 1.073846 |
| ...                     | ...  | ...                             | ...      |
| 2023-11-01              | 98.55 96.504762  | 91.410115                       | 1.055734 |
| 2023-12-01              | 96.85 94.071774  | 92.511370                       | 1.016867 |

|            |       |            |           |          |          |
|------------|-------|------------|-----------|----------|----------|
| 2024-01-01 | 87.44 | 102.188739 | 93.404493 | 1.094045 | 0.855672 |
| 2024-02-01 | 84.82 | 95.006450  | 94.063523 | 1.010024 | 0.892781 |
| 2024-03-01 | 92.52 | 91.072158  | 94.483110 | 0.963899 | 1.015898 |

[351 rows x 5 columns],  
 'Preparation\_of\_cereals\_flour\_or\_starch': observed seasadj

| trend      | irregular | seasonal_factor |            |          |          |
|------------|-----------|-----------------|------------|----------|----------|
| 1995-01-01 | 9.25      | 9.433526        | 9.539915   | 0.988848 | 0.980545 |
| 1995-02-01 | 8.07      | 9.528533        | 9.952214   | 0.957429 | 0.846930 |
| 1995-03-01 | 11.96     | 11.649352       | 10.414341  | 1.118588 | 1.026667 |
| 1995-04-01 | 9.07      | 9.786950        | 10.887362  | 0.898928 | 0.926744 |
| 1995-05-01 | 11.78     | 11.843162       | 11.286171  | 1.049352 | 0.994667 |
| ...        | ...       | ...             | ...        | ...      | ...      |
| 2023-11-01 | 163.34    | 157.656244      | 155.116190 | 1.016375 | 1.036052 |
| 2023-12-01 | 153.89    | 154.386945      | 155.651151 | 0.991878 | 0.996781 |
| 2024-01-01 | 145.13    | 156.128940      | 156.554300 | 0.997283 | 0.929552 |
| 2024-02-01 | 158.96    | 158.318817      | 157.794498 | 1.003323 | 1.004050 |
| 2024-03-01 | 164.97    | 155.082558      | 159.188495 | 0.974207 | 1.063756 |

[351 rows x 5 columns],  
 'Food\_products,\_n.i.e.': observed seasadj trend irre  
 gular seasonal\_factor

| 1995-01-01 | 49.60  | 52.919915  | 53.781840  | 0.983974 | 0.937265 |
|------------|--------|------------|------------|----------|----------|
| 1995-02-01 | 54.53  | 55.464204  | 54.465809  | 1.018331 | 0.983157 |
| 1995-03-01 | 72.73  | 66.083398  | 55.268217  | 1.195685 | 1.100579 |
| 1995-04-01 | 55.05  | 54.081144  | 56.251963  | 0.961409 | 1.017915 |
| 1995-05-01 | 60.70  | 56.017936  | 57.379704  | 0.976267 | 1.083582 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 501.45 | 512.007332 | 530.554960 | 0.965041 | 0.979381 |
| 2023-12-01 | 458.44 | 507.707658 | 521.138843 | 0.974227 | 0.902961 |
| 2024-01-01 | 442.61 | 515.058286 | 511.113997 | 1.007717 | 0.859340 |
| 2024-02-01 | 461.75 | 502.764313 | 505.320707 | 0.994941 | 0.918422 |
| 2024-03-01 | 612.53 | 512.790997 | 506.126093 | 1.013168 | 1.194502 |

[351 rows x 5 columns],  
 'Beverages': observed seasadj trend irregular seaso  
 nal\_factor

| 1995-01-01 | 18.04  | 19.626313  | 19.232066  | 1.020499 | 0.919174 |
|------------|--------|------------|------------|----------|----------|
| 1995-02-01 | 19.04  | 19.164498  | 18.961147  | 1.010725 | 0.993504 |
| 1995-03-01 | 23.87  | 19.212667  | 18.694984  | 1.027691 | 1.242409 |
| 1995-04-01 | 19.29  | 17.330376  | 18.479595  | 0.937811 | 1.113075 |
| 1995-05-01 | 24.05  | 19.409096  | 18.361211  | 1.057071 | 1.239110 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 233.63 | 246.556963 | 250.494391 | 0.984281 | 0.947570 |
| 2023-12-01 | 236.28 | 249.918700 | 252.462246 | 0.989925 | 0.945427 |
| 2024-01-01 | 238.05 | 256.000350 | 254.764529 | 1.004851 | 0.929882 |
| 2024-02-01 | 254.51 | 258.951421 | 257.037047 | 1.007448 | 0.982848 |
| 2024-03-01 | 294.41 | 256.598091 | 258.708514 | 0.991842 | 1.147358 |

[351 rows x 5 columns],  
 'Rubber\_products': observed seasadj trend irregular  
 seasonal\_factor

| 1995-01-01 | 38.25  | 43.405001  | 56.645755  | 0.766253 | 0.881235 |
|------------|--------|------------|------------|----------|----------|
| 1995-02-01 | 60.26  | 63.949381  | 56.707794  | 1.127700 | 0.942308 |
| 1995-03-01 | 117.35 | 109.592753 | 104.734768 | 1.046384 | 1.070782 |
| 1995-04-01 | 90.94  | 101.142044 | 105.295327 | 0.960556 | 0.899132 |

## 1\_composite\_export\_value\_analysis

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1995-05-01 | 32.25  | 31.188495  | 31.392868  | 0.993490 | 1.034035 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 449.19 | 450.754868 | 441.424824 | 1.021136 | 0.996528 |
| 2023-12-01 | 474.25 | 446.237956 | 443.870346 | 1.005334 | 1.062774 |
| 2024-01-01 | 442.81 | 455.757731 | 443.869745 | 1.026783 | 0.971591 |
| 2024-02-01 | 472.05 | 446.335591 | 440.209318 | 1.013917 | 1.057612 |
| 2024-03-01 | 465.83 | 411.847795 | 434.397664 | 0.948089 | 1.131073 |

|                         |        |            |            |           |           |
|-------------------------|--------|------------|------------|-----------|-----------|
| [351 rows x 5 columns], |        |            |            |           |           |
| 'Paper_products':       |        | observed   | seasadj    | trend     | irregular |
| seasonal_factor         |        |            |            |           |           |
| 1995-01-01              | 22.32  | 28.347992  | 20.198783  | 8.149210  | 0.787357  |
| 1995-02-01              | 13.17  | 19.006391  | 20.789197  | -1.782806 | 0.692925  |
| 1995-03-01              | 24.56  | 23.288049  | 21.571686  | 1.716363  | 1.054618  |
| 1995-04-01              | 20.93  | 20.769369  | 22.576227  | -1.806858 | 1.007734  |
| 1995-05-01              | 21.80  | 17.770953  | 23.797823  | -6.026871 | 1.226721  |
| ...                     | ...    | ...        | ...        | ...       | ...       |
| 2023-11-01              | 141.27 | 135.182962 | 132.500226 | 2.682736  | 1.045028  |
| 2023-12-01              | 133.51 | 131.665969 | 132.657079 | -0.991110 | 1.014005  |
| 2024-01-01              | 137.73 | 144.633860 | 132.334169 | 12.299691 | 0.952267  |
| 2024-02-01              | 128.45 | 130.318705 | 131.885247 | -1.566541 | 0.985660  |
| 2024-03-01              | 132.62 | 123.736545 | 131.638973 | -7.902428 | 1.071793  |

|                         |        |            |            |          |          |
|-------------------------|--------|------------|------------|----------|----------|
| [351 rows x 5 columns], |        |            |            |          |          |
| 'Wood_&_wood_products': |        | observed   | seasadj    | trend    | irreg    |
| ular seasonal_factor    |        |            |            |          |          |
| 1995-01-01              | 20.46  | 23.327954  | 24.385204  | 0.956644 | 0.877059 |
| 1995-02-01              | 22.00  | 24.485615  | 24.907255  | 0.983072 | 0.898487 |
| 1995-03-01              | 27.65  | 27.654743  | 25.495652  | 1.084685 | 0.999828 |
| 1995-04-01              | 19.59  | 23.630842  | 26.073852  | 0.906304 | 0.829001 |
| 1995-05-01              | 25.05  | 26.722699  | 26.575441  | 1.005541 | 0.937405 |
| ...                     | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01              | 136.17 | 137.567695 | 133.990396 | 1.026698 | 0.989840 |
| 2023-12-01              | 138.11 | 131.040240 | 133.639297 | 0.980552 | 1.053951 |
| 2024-01-01              | 127.45 | 132.870591 | 132.368070 | 1.003796 | 0.959204 |
| 2024-02-01              | 124.88 | 133.823300 | 130.766861 | 1.023373 | 0.933171 |
| 2024-03-01              | 138.39 | 122.678525 | 129.710622 | 0.945786 | 1.128070 |

|                               |       |           |           |          |          |
|-------------------------------|-------|-----------|-----------|----------|----------|
| [351 rows x 5 columns],       |       |           |           |          |          |
| 'Leather_&_leather_products': |       | observed  | seasadj   | trend    | i        |
| regular seasonal_factor       |       |           |           |          |          |
| 1995-01-01                    | 32.77 | 36.262360 | 37.003386 | 0.979974 | 0.903692 |
| 1995-02-01                    | 32.08 | 37.486940 | 38.226907 | 0.980643 | 0.855765 |
| 1995-03-01                    | 49.86 | 48.578840 | 39.505881 | 1.229661 | 1.026373 |
| 1995-04-01                    | 34.78 | 39.818232 | 40.817354 | 0.975522 | 0.873469 |
| 1995-05-01                    | 42.95 | 42.058543 | 41.947695 | 1.002643 | 1.021196 |
| ...                           | ...   | ...       | ...       | ...      | ...      |
| 2023-11-01                    | 86.49 | 79.462803 | 79.809775 | 0.995653 | 1.088434 |
| 2023-12-01                    | 58.60 | 67.081365 | 79.392473 | 0.844934 | 0.873566 |
| 2024-01-01                    | 59.45 | 79.646259 | 78.897023 | 1.009496 | 0.746426 |
| 2024-02-01                    | 63.99 | 76.972724 | 78.794109 | 0.976884 | 0.831333 |
| 2024-03-01                    | 74.79 | 80.709033 | 78.969512 | 1.022028 | 0.926662 |

|                         |  |          |         |       |               |
|-------------------------|--|----------|---------|-------|---------------|
| [351 rows x 5 columns], |  |          |         |       |               |
| 'Animal_food':          |  | observed | seasadj | trend | irregular sea |
| sonal_factor            |  |          |         |       |               |

## 1\_composite\_export\_value\_analysis

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1995-01-01 | 14.81  | 15.666074  | 16.706795  | 0.937707 | 0.945355 |
| 1995-02-01 | 17.27  | 17.933416  | 16.842049  | 1.064800 | 0.963007 |
| 1995-03-01 | 20.37  | 19.885289  | 17.097646  | 1.163043 | 1.024375 |
| 1995-04-01 | 13.96  | 14.783477  | 17.378633  | 0.850670 | 0.944297 |
| 1995-05-01 | 16.31  | 17.178024  | 17.729601  | 0.968889 | 0.949469 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 220.40 | 219.249489 | 220.873404 | 0.992648 | 1.005247 |
| 2023-12-01 | 224.41 | 228.373245 | 224.963256 | 1.015158 | 0.982646 |
| 2024-01-01 | 211.52 | 226.597209 | 229.923230 | 0.985534 | 0.933463 |
| 2024-02-01 | 232.34 | 236.871413 | 235.096935 | 1.007548 | 0.980870 |
| 2024-03-01 | 261.03 | 242.572515 | 239.532769 | 1.012690 | 1.076091 |

|                                      |           |                 |           |          |          |
|--------------------------------------|-----------|-----------------|-----------|----------|----------|
| [351 rows x 5 columns],              |           |                 |           |          |          |
| 'Other_agro-manufacturing_products': |           |                 |           | observed | seasadj  |
| trend                                | irregular | seasonal_factor |           |          |          |
| 1995-01-01                           | 20.85     | 19.177887       | 22.275534 | 0.860939 | 1.087190 |
| 1995-02-01                           | 18.71     | 19.241455       | 21.955582 | 0.876381 | 0.972380 |
| 1995-03-01                           | 28.82     | 23.952413       | 21.641415 | 1.106786 | 1.203219 |
| 1995-04-01                           | 22.24     | 20.481606       | 21.346359 | 0.959489 | 1.085852 |
| 1995-05-01                           | 27.55     | 24.190007       | 20.948368 | 1.154744 | 1.138900 |
| ...                                  | ...       | ...             | ...       | ...      | ...      |
| 2023-11-01                           | 103.73    | 115.368258      | 90.151430 | 1.279716 | 0.899121 |
| 2023-12-01                           | 92.14     | 90.673107       | 91.560307 | 0.990310 | 1.016178 |
| 2024-01-01                           | 93.64     | 97.135642       | 92.371848 | 1.051572 | 0.964013 |
| 2024-02-01                           | 89.21     | 98.530085       | 92.453385 | 1.065727 | 0.905409 |
| 2024-03-01                           | 97.53     | 93.329297       | 91.836477 | 1.016255 | 1.045009 |

|                                   |           |                 |            |          |          |
|-----------------------------------|-----------|-----------------|------------|----------|----------|
| [351 rows x 5 columns],           |           |                 |            |          |          |
| 'Apparels_and_Textile_Materials': |           |                 |            | observed | seasadj  |
| end                               | irregular | seasonal_factor |            |          | tr       |
| 1995-01-01                        | 518.17    | 576.939570      | 554.001078 | 1.041405 | 0.898136 |
| 1995-02-01                        | 488.79    | 558.536771      | 551.622323 | 1.012535 | 0.875126 |
| 1995-03-01                        | 613.60    | 625.483367      | 548.165058 | 1.141049 | 0.981001 |
| 1995-04-01                        | 442.69    | 526.069060      | 544.409535 | 0.966311 | 0.841505 |
| 1995-05-01                        | 499.84    | 481.230332      | 541.077835 | 0.889392 | 1.038671 |
| ...                               | ...       | ...             | ...        | ...      | ...      |
| 2023-11-01                        | 511.14    | 511.683997      | 515.692956 | 0.992226 | 0.998937 |
| 2023-12-01                        | 495.33    | 507.087263      | 517.753239 | 0.979400 | 0.976814 |
| 2024-01-01                        | 495.20    | 532.933184      | 519.972633 | 1.024925 | 0.929197 |
| 2024-02-01                        | 519.91    | 526.198970      | 522.118050 | 1.007816 | 0.988048 |
| 2024-03-01                        | 556.99    | 512.721596      | 524.050194 | 0.978383 | 1.086340 |

|                         |                 |            |            |          |               |
|-------------------------|-----------------|------------|------------|----------|---------------|
| [351 rows x 5 columns], |                 |            |            |          |               |
| 'Footware_and_parts':   |                 |            |            | observed | seasadj       |
| ar                      | seasonal_factor |            |            |          | trend irregul |
| 1995-01-01              | 179.20          | 164.413796 | 177.190814 | 0.927891 | 1.089933      |
| 1995-02-01              | 179.19          | 164.300843 | 180.321420 | 0.911155 | 1.090621      |
| 1995-03-01              | 179.66          | 178.290226 | 183.761331 | 0.970227 | 1.007683      |
| 1995-04-01              | 135.52          | 152.797591 | 187.487743 | 0.814974 | 0.886925      |
| 1995-05-01              | 198.12          | 196.437317 | 190.755937 | 1.029784 | 1.008566      |
| ...                     | ...             | ...        | ...        | ...      | ...           |
| 2023-11-01              | 49.57           | 47.715363  | 44.507799  | 1.072067 | 1.038869      |
| 2023-12-01              | 40.53           | 37.985140  | 44.116111  | 0.861026 | 1.066996      |
| 2024-01-01              | 42.75           | 44.460057  | 43.828849  | 1.014402 | 0.961537      |
| 2024-02-01              | 41.70           | 41.967033  | 43.626816  | 0.961955 | 0.993637      |
| 2024-03-01              | 42.23           | 40.447744  | 43.481559  | 0.930228 | 1.044063      |

| [351 rows x 5 columns],<br>'Computer': |        |            |            |          | observed | seasadj | trend | irregular | season |
|--|--------|------------|------------|----------|----------|---------|-------|-----------|--------|
| al_factor                              |        |            |            |          |          |         |       |           |        |
| 1995-01-01                             | 0.72   | 0.629698   | 0.523508   | 1.202844 |          |         |       | 1.143405  |        |
| 1995-02-01                             | 0.56   | 0.533709   | 0.500108   | 1.067187 |          |         |       | 1.049261  |        |
| 1995-03-01                             | 0.60   | 0.475614   | 0.471423   | 1.008891 |          |         |       | 1.261527  |        |
| 1995-04-01                             | 0.28   | 0.339739   | 0.445455   | 0.762678 |          |         |       | 0.824161  |        |
| 1995-05-01                             | 0.32   | 0.448186   | 0.422513   | 1.060762 |          |         |       | 0.713990  |        |
| ...                                    | ...    | ...        | ...        | ...      |          |         |       | ...       |        |
| 2023-11-01                             | 200.34 | 206.830179 | 209.412398 | 0.987669 |          |         |       | 0.968621  |        |
| 2023-12-01                             | 182.81 | 193.015223 | 221.208573 | 0.872549 |          |         |       | 0.947127  |        |
| 2024-01-01                             | 239.80 | 240.698231 | 231.493513 | 1.039762 |          |         |       | 0.996268  |        |
| 2024-02-01                             | 244.72 | 272.044505 | 239.787909 | 1.134521 |          |         |       | 0.899559  |        |
| 2024-03-01                             | 350.21 | 301.488283 | 245.747460 | 1.226822 |          |         |       | 1.161604  |        |

| [351 rows x 5 columns],<br>'Computer_parts_&_accessories': |           |                 |            |          | observed | seasadj | tre      |
|--|-----------|-----------------|------------|----------|----------|---------|----------|
| nd   | irregular | seasonal_factor |            |          |          |         |          |
| 1995-01-01   | 326.05    | 330.298335      | 344.550499 | 0.958635 |          |         | 0.987138 |
| 1995-02-01   | 333.87    | 347.850533      | 365.652074 | 0.951316 |          |         | 0.959809 |
| 1995-03-01   | 435.51    | 394.139309      | 388.584534 | 1.014295 |          |         | 1.104965 |
| 1995-04-01   | 371.19    | 413.946051      | 412.011790 | 1.004695 |          |         | 0.896711 |
| 1995-05-01   | 406.50    | 443.767422      | 433.172591 | 1.024459 |          |         | 0.916020 |
| ...  | ...       | ...             | ...        | ...      |          |         | ...      |
| 2023-11-01   | 891.00    | 928.063445      | 946.141587 | 0.980893 |          |         | 0.960064 |
| 2023-12-01   | 1253.74   | 995.662149      | 958.594545 | 1.038669 |          |         | 1.259202 |
| 2024-01-01   | 766.94    | 1005.650825     | 964.815153 | 1.042325 |          |         | 0.762631 |
| 2024-02-01   | 828.48    | 955.439765      | 959.833960 | 0.995422 |          |         | 0.867119 |
| 2024-03-01   | 1129.36   | 918.738621      | 949.230442 | 0.967877 |          |         | 1.229251 |

| [351 rows x 5 columns],<br>'Integrated_circuits_&_parts': |                 |            |            |          | observed | seasadj | trend    |
|---|-----------------|------------|------------|----------|----------|---------|----------|
| irregular   | seasonal_factor |            |            |          |          |         |          |
| 1995-01-01  | 158.74          | 160.859682 | 167.323072 | 0.961372 |          |         | 0.986823 |
| 1995-02-01  | 165.06          | 169.649698 | 173.085214 | 0.980151 |          |         | 0.972946 |
| 1995-03-01  | 202.41          | 182.976986 | 179.976718 | 1.016670 |          |         | 1.106205 |
| 1995-04-01  | 175.08          | 188.092282 | 186.857647 | 1.006607 |          |         | 0.930820 |
| 1995-05-01  | 202.25          | 208.311817 | 193.105121 | 1.078748 |          |         | 0.970900 |
| ...   | ...             | ...        | ...        | ...      |          |         | ...      |
| 2023-11-01  | 762.06          | 740.588663 | 755.505616 | 0.980256 |          |         | 1.028992 |
| 2023-12-01  | 770.18          | 754.276753 | 744.774438 | 1.012759 |          |         | 1.021084 |
| 2024-01-01  | 721.61          | 751.096465 | 731.739858 | 1.026453 |          |         | 0.960742 |
| 2024-02-01  | 653.42          | 713.304332 | 716.509908 | 0.995526 |          |         | 0.916047 |
| 2024-03-01  | 663.35          | 634.886221 | 701.139161 | 0.905507 |          |         | 1.044833 |

| [351 rows x 5 columns],<br>'Printed_circuits': |        |            |            |           | observed | seasadj | trend | irregular |
|--|--------|------------|------------|-----------|----------|---------|-------|-----------|
| seasonal_factor                                |        |            |            |           |          |         |       |           |
| 1995-01-01                                     | 40.94  | 44.370518  | 51.319085  | -6.948567 |          |         |       | 0.922685  |
| 1995-02-01                                     | 41.02  | 43.865675  | 53.131372  | -9.265696 |          |         |       | 0.935128  |
| 1995-03-01                                     | 56.18  | 57.623843  | 54.339380  | 3.284463  |          |         |       | 0.974944  |
| 1995-04-01                                     | 53.74  | 56.727899  | 54.803026  | 1.924873  |          |         |       | 0.947329  |
| 1995-05-01                                     | 62.90  | 60.597951  | 54.065948  | 6.532003  |          |         |       | 1.037989  |
| ...  | ...    | ...        | ...        | ...       |          |         |       | ...       |
| 2023-11-01                                     | 111.18 | 110.696041 | 107.117432 | 3.578609  |          |         |       | 1.004372  |

|            |        |            |            |           |          |
|------------|--------|------------|------------|-----------|----------|
| 2023-12-01 | 106.60 | 105.843979 | 106.404686 | -0.560708 | 1.007143 |
| 2024-01-01 | 103.41 | 106.134428 | 105.388148 | 0.746280  | 0.974330 |
| 2024-02-01 | 103.07 | 104.432942 | 104.557677 | -0.124734 | 0.986949 |
| 2024-03-01 | 105.01 | 98.921276  | 104.037187 | -5.115911 | 1.061551 |

[351 rows x 5 columns],  
'Telecommunication\_equipments': observed seasadj tren

| d          | irregular | seasonal_factor |            |          |          |
|------------|-----------|-----------------|------------|----------|----------|
| 1995-01-01 | 49.68     | 53.699065       | 55.911821  | 0.960424 | 0.925156 |
| 1995-02-01 | 55.97     | 55.799713       | 56.684153  | 0.984397 | 1.003052 |
| 1995-03-01 | 63.90     | 58.842967       | 57.511525  | 1.023151 | 1.085941 |
| 1995-04-01 | 48.61     | 58.267336       | 58.125477  | 1.002441 | 0.834258 |
| 1995-05-01 | 60.19     | 60.328426       | 58.361271  | 1.033707 | 0.997705 |
| ...        | ...       | ...             | ...        | ...      | ...      |
| 2023-11-01 | 939.75    | 881.609193      | 859.914129 | 1.025229 | 1.065949 |
| 2023-12-01 | 833.24    | 753.282905      | 865.523772 | 0.870320 | 1.106145 |
| 2024-01-01 | 877.76    | 907.862247      | 871.433604 | 1.041803 | 0.966843 |
| 2024-02-01 | 735.32    | 855.192164      | 879.327384 | 0.972553 | 0.859830 |
| 2024-03-01 | 845.57    | 876.485250      | 891.730973 | 0.982903 | 0.964728 |

[351 rows x 5 columns],  
'Air\_conditioning\_machines': observed seasadj trend

| irregular  | seasonal_factor |            |            |          |          |
|------------|-----------------|------------|------------|----------|----------|
| 1995-01-01 | 45.70           | 49.490727  | 49.855499  | 0.992683 | 0.923405 |
| 1995-02-01 | 61.51           | 47.890877  | 52.468225  | 0.912760 | 1.284378 |
| 1995-03-01 | 93.18           | 56.340459  | 55.099533  | 1.022522 | 1.653874 |
| 1995-04-01 | 60.83           | 45.998752  | 57.661385  | 0.797739 | 1.322427 |
| 1995-05-01 | 78.47           | 63.365609  | 59.598629  | 1.063206 | 1.238369 |
| ...        | ...             | ...        | ...        | ...      | ...      |
| 2023-11-01 | 329.01          | 385.386615 | 405.705461 | 0.949917 | 0.853714 |
| 2023-12-01 | 387.15          | 427.230695 | 419.668786 | 1.018019 | 0.906185 |
| 2024-01-01 | 521.29          | 464.237801 | 433.502414 | 1.070900 | 1.122894 |
| 2024-02-01 | 566.42          | 441.530878 | 445.669211 | 0.990714 | 1.282855 |
| 2024-03-01 | 644.31          | 444.758225 | 455.383703 | 0.976667 | 1.448675 |

[351 rows x 5 columns],  
'Refrigerators': observed seasadj trend irregular s

| easonal_factor |        |            |            |          |          |
|----------------|--------|------------|------------|----------|----------|
| 1995-01-01     | 18.11  | 18.857844  | 19.029178  | 0.990996 | 0.960343 |
| 1995-02-01     | 17.86  | 19.019706  | 20.139635  | 0.944392 | 0.939026 |
| 1995-03-01     | 25.22  | 22.169738  | 21.441824  | 1.033948 | 1.137587 |
| 1995-04-01     | 19.09  | 22.432092  | 22.906968  | 0.979269 | 0.851013 |
| 1995-05-01     | 25.32  | 23.952765  | 24.427221  | 0.980577 | 1.057080 |
| ...            | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01     | 163.77 | 173.074856 | 170.462460 | 1.015325 | 0.946238 |
| 2023-12-01     | 148.72 | 165.134880 | 169.137568 | 0.976335 | 0.900597 |
| 2024-01-01     | 166.15 | 168.518427 | 167.713380 | 1.004800 | 0.985946 |
| 2024-02-01     | 159.70 | 164.102576 | 167.351346 | 0.980587 | 0.973172 |
| 2024-03-01     | 193.00 | 171.162233 | 168.211818 | 1.017540 | 1.127585 |

[351 rows x 5 columns],  
'Parts\_of\_electrical\_appliances': observed seasadj tr

| end        | irregular | seasonal_factor |           |          |          |
|------------|-----------|-----------------|-----------|----------|----------|
| 1995-01-01 | 71.87     | 80.849570       | 81.531348 | 0.991638 | 0.888935 |
| 1995-02-01 | 79.11     | 85.329435       | 83.118081 | 1.026605 | 0.927113 |
| 1995-03-01 | 99.04     | 94.203474       | 85.396096 | 1.103136 | 1.051341 |

## 1\_composite\_export\_value\_analysis

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1995-04-01 | 74.65  | 84.493183  | 88.269007  | 0.957224 | 0.883503 |
| 1995-05-01 | 87.63  | 81.681547  | 91.812955  | 0.889652 | 1.072825 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 787.03 | 788.127134 | 734.976134 | 1.072317 | 0.998608 |
| 2023-12-01 | 724.88 | 693.781132 | 714.002359 | 0.971679 | 1.044825 |
| 2024-01-01 | 604.74 | 668.644900 | 693.634224 | 0.963973 | 0.904426 |
| 2024-02-01 | 693.82 | 728.444272 | 677.536799 | 1.075136 | 0.952468 |
| 2024-03-01 | 699.57 | 621.744272 | 666.390674 | 0.933003 | 1.125173 |

[351 rows x 5 columns],

|            |                               |            | observed   | seasadj    | trend     |
|------------|-------------------------------|------------|------------|------------|-----------|
|            | 'Electrical_control_devices': |            |            |            |           |
| irregular  | seasonal_factor               |            |            |            |           |
| 1995-01-01 | 1.37                          | 0.084493   | 0.176462   | -0.091969  | 16.214358 |
| 1995-02-01 | 2.26                          | 2.203839   | 1.132203   | 1.071636   | 1.025483  |
| 1995-03-01 | 1.72                          | 3.145507   | 2.389212   | 0.756295   | 0.546812  |
| 1995-04-01 | 1.20                          | 2.477288   | 3.992402   | -1.515114  | 0.484401  |
| 1995-05-01 | 2.73                          | 3.279324   | 5.915435   | -2.636112  | 0.832489  |
| ...        | ...                           | ...        | ...        | ...        | ...       |
| 2023-11-01 | 207.56                        | 205.098827 | 203.563478 | 1.535349   | 1.012000  |
| 2023-12-01 | 189.42                        | 189.656333 | 205.641277 | -15.984945 | 0.998754  |
| 2024-01-01 | 185.48                        | 206.584130 | 208.539962 | -1.955832  | 0.897842  |
| 2024-02-01 | 214.16                        | 219.521151 | 211.446439 | 8.074713   | 0.975578  |
| 2024-03-01 | 274.43                        | 262.787114 | 213.497244 | 49.289870  | 1.044305  |

[351 rows x 5 columns],

|            |   | observed   | seasadj    | trend    | irregular | seasonal |
|------------|---|------------|------------|----------|-----------|----------|
|            | 'Other_electrical_apparatus_(excl._printed_circuits_and_electrical_control_devices)': |            |            |          |           |          |
| _factor    |   |            |            |          |           |          |
| 1995-01-01 | 70.91   | 71.971262  | 77.738445  | 0.925813 | 0.985254  |          |
| 1995-02-01 | 78.12   | 81.064298  | 80.024847  | 1.012989 | 0.963679  |          |
| 1995-03-01 | 93.82   | 95.276571  | 82.294950  | 1.157745 | 0.984712  |          |
| 1995-04-01 | 70.44   | 75.886753  | 84.493490  | 0.898137 | 0.928225  |          |
| 1995-05-01 | 80.77   | 87.156638  | 86.255995  | 1.010442 | 0.926722  |          |
| ...        | ...   | ...        | ...        | ...      | ...       | ...      |
| 2023-11-01 | 409.30  | 401.088829 | 398.275750 | 1.007063 | 1.020472  |          |
| 2023-12-01 | 374.09  | 388.845484 | 397.932978 | 0.977163 | 0.962053  |          |
| 2024-01-01 | 378.32  | 406.574515 | 397.913511 | 1.021766 | 0.930506  |          |
| 2024-02-01 | 389.97  | 401.712456 | 398.056336 | 1.009185 | 0.970769  |          |
| 2024-03-01 | 413.64  | 387.013074 | 398.404490 | 0.971407 | 1.068801  |          |

[351 rows x 5 columns],

|            |                               | observed  | seasadj   | trend    | i        |
|------------|-------------------------------|-----------|-----------|----------|----------|
|            | 'Appliances_Washing_machine': |           |           |          |          |
| irregular  | seasonal_factor               |           |           |          |          |
| 1995-01-01 | 1.21                          | 1.222044  | 1.710976  | 0.714238 | 0.990145 |
| 1995-02-01 | 1.61                          | 1.705839  | 1.872197  | 0.911143 | 0.943817 |
| 1995-03-01 | 1.91                          | 1.819677  | 2.035972  | 0.893764 | 1.049637 |
| 1995-04-01 | 2.64                          | 3.310471  | 2.194089  | 1.508814 | 0.797470 |
| 1995-05-01 | 2.38                          | 2.563743  | 2.307796  | 1.110905 | 0.928330 |
| ...        | ...                           | ...       | ...       | ...      | ...      |
| 2023-11-01 | 87.14                         | 93.998488 | 95.389655 | 0.985416 | 0.927036 |
| 2023-12-01 | 74.26                         | 75.532924 | 94.174614 | 0.802052 | 0.983147 |
| 2024-01-01 | 98.38                         | 93.356382 | 93.682674 | 0.996517 | 1.053811 |
| 2024-02-01 | 103.23                        | 97.200780 | 93.804795 | 1.036203 | 1.062029 |
| 2024-03-01 | 105.74                        | 90.467522 | 94.124525 | 0.961147 | 1.168817 |

[351 rows x 5 columns],

| 'Other_Electrical_Applications':<br>irregular seasonal_factor |        |            | observed   | seasadj  | trend    |
|---|--------|------------|------------|----------|----------|
| 1995-01-01  | 82.33  | 91.652348  | 92.924203  | 0.986313 | 0.898286 |
| 1995-02-01  | 87.98  | 93.409447  | 95.134833  | 0.981864 | 0.941875 |
| 1995-03-01  | 101.49 | 98.376281  | 97.933487  | 1.004521 | 1.031651 |
| 1995-04-01  | 71.25  | 84.138508  | 100.944419 | 0.833513 | 0.846818 |
| 1995-05-01  | 130.21 | 125.698623 | 103.976239 | 1.208917 | 1.035890 |
| ...   | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01  | 192.76 | 198.490885 | 199.594116 | 0.994473 | 0.971128 |
| 2023-12-01  | 179.83 | 196.276449 | 197.319437 | 0.994714 | 0.916208 |
| 2024-01-01  | 191.06 | 200.287528 | 195.773730 | 1.023056 | 0.953929 |
| 2024-02-01  | 196.07 | 198.755960 | 195.232136 | 1.018049 | 0.986486 |
| 2024-03-01  | 210.57 | 190.536291 | 195.636347 | 0.973931 | 1.105144 |
| [351 rows x 5 columns],                                       |        |            |            |          |          |
| 'Iron_&_steel':<br>asonal_factor                              |        |            | observed   | seasadj  | trend    |
| 1995-01-01  | 41.23  | 45.293830  | 45.417264  | 0.997282 | 0.910279 |
| 1995-02-01  | 37.08  | 42.004583  | 45.620145  | 0.920746 | 0.882761 |
| 1995-03-01  | 57.52  | 50.254533  | 46.221551  | 1.087253 | 1.144573 |
| 1995-04-01  | 43.48  | 46.897377  | 46.736569  | 1.003441 | 0.927131 |
| 1995-05-01  | 49.76  | 47.674232  | 47.239097  | 1.009211 | 1.043750 |
| ...   | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01  | 476.20 | 479.351647 | 355.537302 | 1.348246 | 0.993425 |
| 2023-12-01  | 408.05 | 395.050864 | 360.679026 | 1.095298 | 1.032905 |
| 2024-01-01  | 847.20 | 889.177080 | 356.222619 | 2.496128 | 0.952791 |
| 2024-02-01  | 347.24 | 371.489230 | 342.350778 | 1.085113 | 0.934724 |
| 2024-03-01  | 296.41 | 285.594398 | 323.092906 | 0.883939 | 1.037870 |
| [351 rows x 5 columns],                                       |        |            |            |          |          |
| 'Aluminium':<br>nal_factor                                    |        |            | observed   | seasadj  | trend    |
| 1995-01-01  | 9.42   | 10.825066  | 10.509622  | 1.030015 | 0.870203 |
| 1995-02-01  | 10.65  | 10.578403  | 10.819733  | 0.977695 | 1.006768 |
| 1995-03-01  | 10.76  | 10.502087  | 11.199122  | 0.937760 | 1.024558 |
| 1995-04-01  | 6.76   | 7.742040   | 11.628219  | 0.665798 | 0.873155 |
| 1995-05-01  | 12.34  | 11.907824  | 12.051736  | 0.988059 | 1.036293 |
| ...   | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01  | 227.28 | 236.390773 | 231.530369 | 1.020993 | 0.961459 |
| 2023-12-01  | 226.80 | 239.112187 | 235.367733 | 1.015909 | 0.948509 |
| 2024-01-01  | 218.90 | 239.714591 | 237.677215 | 1.008572 | 0.913169 |
| 2024-02-01  | 229.40 | 237.923478 | 238.858353 | 0.996086 | 0.964176 |
| 2024-03-01  | 258.84 | 237.083227 | 239.945654 | 0.988071 | 1.091769 |
| [351 rows x 5 columns],                                       |        |            |            |          |          |
| 'Copper':<br>factor   |        |            | observed   | seasadj  | trend    |
| 1995-01-01  | 5.12   | 5.862006   | 5.802822   | 1.010199 | 0.873421 |
| 1995-02-01  | 5.06   | 5.420370   | 5.892561   | 0.919867 | 0.933516 |
| 1995-03-01  | 6.70   | 6.574570   | 6.072646   | 1.082653 | 1.019078 |
| 1995-04-01  | 5.51   | 6.194251   | 6.273269   | 0.987404 | 0.889535 |
| 1995-05-01  | 6.22   | 6.296248   | 6.494633   | 0.969454 | 0.987890 |
| ...   | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01  | 299.16 | 307.722613 | 300.536727 | 1.023910 | 0.972174 |
| 2023-12-01  | 296.94 | 296.186593 | 300.915847 | 0.984284 | 1.002544 |
| 2024-01-01  | 284.82 | 306.342647 | 302.383011 | 1.013095 | 0.929743 |

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 2024-02-01 | 303.63 | 301.987562 | 305.565761 | 0.988290 | 1.005439 |
| 2024-03-01 | 337.96 | 302.070974 | 309.866269 | 0.974843 | 1.118810 |

[351 rows x 5 columns],  
'Other\_metal\_&\_steel': observed seasadj trend irregular  
lar seasonal\_factor

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1995-01-01 | 37.06  | 44.545280  | 46.393277  | 0.960167 | 0.831962 |
| 1995-02-01 | 46.00  | 46.409504  | 47.302597  | 0.981120 | 0.991176 |
| 1995-03-01 | 69.16  | 67.768451  | 48.333043  | 1.402114 | 1.020534 |
| 1995-04-01 | 47.79  | 52.386499  | 49.244659  | 1.063801 | 0.912258 |
| 1995-05-01 | 45.62  | 48.905729  | 49.972826  | 0.978646 | 0.932815 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 413.65 | 417.595029 | 410.336306 | 1.017690 | 0.990553 |
| 2023-12-01 | 384.03 | 399.389623 | 409.566086 | 0.975153 | 0.961542 |
| 2024-01-01 | 397.04 | 423.143771 | 407.850334 | 1.037498 | 0.938310 |
| 2024-02-01 | 407.21 | 408.508490 | 405.878420 | 1.006480 | 0.996821 |
| 2024-03-01 | 423.69 | 394.178383 | 404.706779 | 0.973985 | 1.074869 |

[351 rows x 5 columns],  
'Passenger\_car': observed seasadj trend irregular s  
easonal\_factor

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1995-01-01 | 0.18   | 0.181000   | 0.182735   | 0.990504 | 0.994476 |
| 1995-02-01 | 0.21   | 0.253311   | 0.175024   | 1.447293 | 0.829022 |
| 1995-03-01 | 0.08   | 0.104947   | 0.165142   | 0.635498 | 0.762286 |
| 1995-04-01 | 0.02   | 0.035884   | 0.153390   | 0.233943 | 0.557344 |
| 1995-05-01 | 0.13   | 0.144449   | 0.139622   | 1.034574 | 0.899972 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 528.69 | 470.793737 | 607.466248 | 0.775012 | 1.122976 |
| 2023-12-01 | 733.42 | 667.489951 | 618.419389 | 1.079348 | 1.098773 |
| 2024-01-01 | 636.21 | 626.452072 | 624.865991 | 1.002538 | 1.015576 |
| 2024-02-01 | 651.54 | 651.523461 | 631.423148 | 1.031833 | 1.000025 |
| 2024-03-01 | 737.94 | 624.339013 | 643.585929 | 0.970094 | 1.181954 |

[351 rows x 5 columns],  
'Pick\_up\_and\_trucks': observed seasadj trend irregular  
ar seasonal\_factor

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1995-01-01 | 2.52   | 3.508505   | 11.360490  | 0.308834 | 0.718255 |
| 1995-02-01 | 9.45   | 10.967642  | 11.067926  | 0.990939 | 0.861626 |
| 1995-03-01 | 9.86   | 11.154626  | 10.673568  | 1.045070 | 0.883938 |
| 1995-04-01 | 1.47   | 1.769177   | 10.218933  | 0.173127 | 0.830895 |
| 1995-05-01 | 8.63   | 9.757693   | 9.700966   | 1.005848 | 0.884430 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 658.18 | 624.423057 | 695.244723 | 0.898134 | 1.054061 |
| 2023-12-01 | 793.36 | 754.251072 | 677.273078 | 1.113659 | 1.051851 |
| 2024-01-01 | 687.55 | 653.214267 | 667.057248 | 0.979248 | 1.052564 |
| 2024-02-01 | 592.17 | 631.170451 | 664.426289 | 0.949948 | 0.938209 |
| 2024-03-01 | 768.41 | 624.898424 | 663.615787 | 0.941657 | 1.229656 |

[351 rows x 5 columns],  
'Motorcycle': observed seasadj trend irregular seas  
onal\_factor

|            |       |           |           |          |          |
|------------|-------|-----------|-----------|----------|----------|
| 1995-01-01 | 3.45  | 4.071092  | 16.843450 | 0.241702 | 0.847438 |
| 1995-02-01 | 7.88  | 8.658665  | 17.866602 | 0.484628 | 0.910071 |
| 1995-03-01 | 17.61 | 17.051275 | 19.120701 | 0.891770 | 1.032767 |
| 1995-04-01 | 20.48 | 23.301394 | 20.191825 | 1.154001 | 0.878917 |
| 1995-05-01 | 24.44 | 23.343266 | 20.835392 | 1.120366 | 1.046983 |

| 2023-11-01   | 243.16          | 242.999209      | 208.924297 | 1.163097        |         | 1.000662 |     |
|--|-----------------|-----------------|------------|-----------------|---------|----------|-----|
| 2023-12-01   | 191.24          | 189.732199      | 210.924203 | 0.899528        |         | 1.007947 |     |
| 2024-01-01   | 214.27          | 204.399102      | 216.687925 | 0.943288        |         | 1.048292 |     |
| 2024-02-01   | 286.33          | 238.565487      | 225.005706 | 1.060264        |         | 1.200216 |     |
| 2024-03-01   | 293.78          | 225.960815      | 233.448335 | 0.967926        |         | 1.300137 |     |
| <br>[351 rows x 5 columns],                              |                 |                 |            |                 |         |          |     |
| 'Other_vehicles,_n.i.e.':                                |                 |                 |            | observed        | seasadj | trend    | irr |
| regular  | seasonal_factor |                 |            |                 |         |          |     |
| 1995-01-01   | 4.14            | 4.729256        | 4.512395   | 1.048059        |         | 0.875402 |     |
| 1995-02-01   | 3.81            | 4.642272        | 4.245358   | 1.093494        |         | 0.820719 |     |
| 1995-03-01   | 3.89            | 4.260450        | 3.945555   | 1.079810        |         | 0.913049 |     |
| 1995-04-01   | 2.60            | 2.426572        | 3.663443   | 0.662375        |         | 1.071471 |     |
| 1995-05-01   | 3.41            | 3.226970        | 3.432707   | 0.940066        |         | 1.056719 |     |
| <br>[351 rows x 5 columns],                              |                 |                 |            |                 |         |          |     |
| 'Vehicle_parts_&_accessories':                           |                 |                 |            | observed        | seasadj | trend    | tre |
| nd   | irregular       | seasonal_factor |            |                 |         |          |     |
| 1995-01-01   | 40.97           | 45.942913       | 47.308409  | 0.971136        |         | 0.891759 |     |
| 1995-02-01   | 43.03           | 48.519656       | 50.270956  | 0.965163        |         | 0.886857 |     |
| 1995-03-01   | 55.99           | 55.180763       | 53.924733  | 1.023292        |         | 1.014665 |     |
| 1995-04-01   | 48.42           | 57.687248       | 57.888217  | 0.996528        |         | 0.839354 |     |
| 1995-05-01   | 54.53           | 57.483367       | 61.927904  | 0.928230        |         | 0.948622 |     |
| <br>[351 rows x 5 columns],                              |                 |                 |            |                 |         |          |     |
| 'Aircrafts,_ships,_floating_structures,_and_locomotive': |                 |                 |            | observ          |         |          |     |
| ed   | seasadj         | trend           | irregular  | seasonal_factor |         |          |     |
| 1995-01-01   | 29.99           | 33.703209       | 45.517611  | 0.740443        |         | 0.889826 |     |
| 1995-02-01   | 34.92           | 35.521391       | 47.582305  | 0.746525        |         | 0.983070 |     |
| 1995-03-01   | 85.56           | 79.883849       | 49.404555  | 1.616933        |         | 1.071055 |     |
| 1995-04-01   | 37.49           | 51.303909       | 51.939301  | 0.987767        |         | 0.730744 |     |
| 1995-05-01   | 51.28           | 55.479815       | 54.106745  | 1.025377        |         | 0.924300 |     |
| <br>[351 rows x 5 columns],                              |                 |                 |            |                 |         |          |     |
| 'Machinery_&_Equipment':                                 |                 |                 |            | observed        | seasadj | trend    | ir  |
| regular  | seasonal_factor |                 |            |                 |         |          |     |
| 1995-01-01   | 236.49          | 258.161217      | 257.145787 | 1.003949        |         | 0.916055 |     |

## 1\_composite\_export\_value\_analysis

|            |         |             |             |          |          |
|------------|---------|-------------|-------------|----------|----------|
| 1995-02-01 | 238.35  | 259.373288  | 262.366469  | 0.988592 | 0.918946 |
| 1995-03-01 | 286.10  | 277.093451  | 269.351582  | 1.028743 | 1.032504 |
| 1995-04-01 | 242.49  | 268.119063  | 277.779994  | 0.965221 | 0.904412 |
| 1995-05-01 | 285.15  | 281.076810  | 287.537072  | 0.977532 | 1.014491 |
| ...        | ...     | ...         | ...         | ...      | ...      |
| 2023-11-01 | 1996.65 | 2004.368993 | 1978.282173 | 1.013187 | 0.996149 |
| 2023-12-01 | 1815.06 | 1778.158229 | 1966.351141 | 0.904293 | 1.020753 |
| 2024-01-01 | 1823.19 | 1960.005260 | 1956.769717 | 1.001654 | 0.930196 |
| 2024-02-01 | 1972.76 | 1979.375043 | 1952.093965 | 1.013975 | 0.996658 |
| 2024-03-01 | 2073.37 | 1902.674300 | 1955.406618 | 0.973033 | 1.089714 |

|                         |         |            |            |          |           |
|-------------------------|---------|------------|------------|----------|-----------|
| [351 rows x 5 columns], |         |            |            |          |           |
| 'Jewellery':            |         | observed   | seasadj    | trend    | irregular |
| seasonal_factor         |         |            |            |          | seaso     |
| 1995-01-01              | 156.47  | 166.117109 | 161.705604 | 1.027281 | 0.941926  |
| 1995-02-01              | 121.59  | 154.406477 | 163.822271 | 0.942524 | 0.787467  |
| 1995-03-01              | 201.12  | 171.201388 | 165.804068 | 1.032552 | 1.174757  |
| 1995-04-01              | 146.12  | 163.569740 | 168.048691 | 0.973347 | 0.893319  |
| 1995-05-01              | 164.38  | 169.357316 | 170.278111 | 0.994592 | 0.970611  |
| ...                     | ...     | ...        | ...        | ...      | ...       |
| 2023-11-01              | 692.52  | 716.719195 | 737.022017 | 0.972453 | 0.966236  |
| 2023-12-01              | 539.92  | 696.059412 | 737.451709 | 0.943871 | 0.775681  |
| 2024-01-01              | 680.71  | 790.377991 | 738.396435 | 1.070398 | 0.861246  |
| 2024-02-01              | 1112.66 | 755.244950 | 740.635599 | 1.019725 | 1.473244  |
| 2024-03-01              | 679.43  | 730.989615 | 744.710797 | 0.981575 | 0.929466  |

|  |           |                 |             |          |          |
|--|-----------|-----------------|-------------|----------|----------|
| [351 rows x 5 columns],                |           |                 |             |          |          |
| 'Chemicals_& Petro-chemical_Products': |           | observed        | seasadj     |          |          |
| trend                                  | irregular | seasonal_factor |             |          |          |
| 1995-01-01                             | 138.59    | 151.432819      | 152.217074  | 0.994848 | 0.915191 |
| 1995-02-01                             | 136.73    | 156.868697      | 158.682680  | 0.988568 | 0.871621 |
| 1995-03-01                             | 176.21    | 167.279959      | 167.497107  | 0.998704 | 1.053384 |
| 1995-04-01                             | 169.39    | 175.573454      | 177.631871  | 0.988412 | 0.964781 |
| 1995-05-01                             | 249.00    | 252.631491      | 188.820341  | 1.337946 | 0.985625 |
| ...                                    | ...       | ...             | ...         | ...      | ...      |
| 2023-11-01                             | 1731.72   | 1754.709388     | 1729.348524 | 1.014665 | 0.986898 |
| 2023-12-01                             | 1562.99   | 1650.622594     | 1711.556241 | 0.964399 | 0.946909 |
| 2024-01-01                             | 1615.16   | 1743.716343     | 1692.985783 | 1.029965 | 0.926275 |
| 2024-02-01                             | 1603.84   | 1673.617463     | 1677.040303 | 0.997959 | 0.958307 |
| 2024-03-01                             | 1750.68   | 1639.597131     | 1667.940967 | 0.983007 | 1.067750 |

|                         |                 |            |            |          |          |
|-------------------------|-----------------|------------|------------|----------|----------|
| [351 rows x 5 columns], |                 |            |            |          |          |
| 'Petroleum_products':   |                 | observed   | seasadj    | trend    | irregul  |
| ar                      | seasonal_factor |            |            |          |          |
| 1995-01-01              | 15.02           | 15.514684  | 46.759413  | 0.331798 | 0.968115 |
| 1995-02-01              | 41.14           | 46.232489  | 43.896400  | 1.053218 | 0.889850 |
| 1995-03-01              | 45.09           | 50.944694  | 39.902349  | 1.276734 | 0.885077 |
| 1995-04-01              | 23.09           | 31.670843  | 35.092064  | 0.902507 | 0.729062 |
| 1995-05-01              | 19.71           | 23.276048  | 29.781075  | 0.781572 | 0.846793 |
| ...                     | ...             | ...        | ...        | ...      | ...      |
| 2023-11-01              | 958.77          | 990.408448 | 984.964563 | 1.005527 | 0.968055 |
| 2023-12-01              | 1014.69         | 988.421512 | 944.819450 | 1.046149 | 1.026576 |
| 2024-01-01              | 765.07          | 866.458886 | 883.895804 | 0.980273 | 0.882985 |
| 2024-02-01              | 633.93          | 759.531385 | 820.329753 | 0.925885 | 0.834633 |
| 2024-03-01              | 724.46          | 795.764407 | 775.723266 | 1.025835 | 0.910395 |

| 'Medicinal_and_surgical_equipment_and_supplies': |                 |                 |                 |          | observed | s       |
|--|-----------------|-----------------|-----------------|----------|----------|---------|
| easadj   | trend           | irregular       | seasonal_factor |          |          |         |
| 1995-01-01                                       | 20.84           | 21.785688       | 21.720121       | 1.003019 | 0.956591 |         |
| 1995-02-01                                       | 19.48           | 21.293526       | 21.630632       | 0.984415 | 0.914832 |         |
| 1995-03-01                                       | 21.91           | 21.822580       | 21.527795       | 1.013693 | 1.004006 |         |
| 1995-04-01                                       | 17.15           | 17.490947       | 21.405163       | 0.817137 | 0.980507 |         |
| 1995-05-01                                       | 22.08           | 21.244930       | 21.264439       | 0.999083 | 1.039307 |         |
| ...  | ...             | ...             | ...             | ...      | ...      |         |
| 2023-11-01                                       | 166.14          | 168.721817      | 172.606031      | 0.977497 | 0.984698 |         |
| 2023-12-01                                       | 164.92          | 172.690955      | 174.405641      | 0.990168 | 0.955001 |         |
| 2024-01-01                                       | 182.95          | 186.603574      | 176.635209      | 1.056435 | 0.980421 |         |
| 2024-02-01                                       | 173.53          | 178.439105      | 178.298857      | 1.000787 | 0.972489 |         |
| 2024-03-01                                       | 184.18          | 176.230646      | 178.772283      | 0.985783 | 1.045108 |         |
| <br>[351 rows x 5 columns],                      |                 |                 |                 |          |          |         |
| 'Toiletries_and_cosmetics':                      |                 |                 |                 |          | observed | seasadj |
| rregular   | seasonal_factor |                 |                 |          | trend    | i       |
| 1995-01-01                                       | 8.56            | 10.154263       | 10.495449       | 0.967492 | 0.842996 |         |
| 1995-02-01                                       | 9.14            | 10.326658       | 10.735467       | 0.961920 | 0.885088 |         |
| 1995-03-01                                       | 14.36           | 14.073457       | 11.025635       | 1.276431 | 1.020361 |         |
| 1995-04-01                                       | 10.83           | 11.179251       | 11.298326       | 0.989461 | 0.968759 |         |
| 1995-05-01                                       | 11.66           | 11.473887       | 11.518056       | 0.996165 | 1.016221 |         |
| ...  | ...             | ...             | ...             | ...      | ...      |         |
| 2023-11-01                                       | 306.43          | 312.836055      | 309.380338      | 1.011170 | 0.979523 |         |
| 2023-12-01                                       | 302.98          | 316.490339      | 313.696624      | 1.008906 | 0.957312 |         |
| 2024-01-01                                       | 308.17          | 321.112503      | 314.421436      | 1.021281 | 0.959695 |         |
| 2024-02-01                                       | 319.35          | 313.488439      | 312.010590      | 1.004737 | 1.018698 |         |
| 2024-03-01                                       | 322.83          | 288.801024      | 309.087089      | 0.934368 | 1.117828 |         |
| <br>[351 rows x 5 columns],                      |                 |                 |                 |          |          |         |
| 'Furniture_and_parts':                           |                 |                 |                 |          | observed | seasadj |
| lar  | seasonal_factor |                 |                 |          | trend    | irregu  |
| 1995-01-01                                       | 61.75           | 67.047271       | 66.151113       | 1.013547 | 0.920992 |         |
| 1995-02-01                                       | 57.44           | 63.878359       | 66.220109       | 0.964637 | 0.899209 |         |
| 1995-03-01                                       | 75.61           | 74.576364       | 66.367779       | 1.123683 | 1.013860 |         |
| 1995-04-01                                       | 54.71           | 65.399507       | 66.536153       | 0.982917 | 0.836551 |         |
| 1995-05-01                                       | 69.34           | 68.294700       | 66.645034       | 1.024753 | 1.015306 |         |
| ...  | ...             | ...             | ...             | ...      | ...      |         |
| 2023-11-01                                       | 142.73          | 136.520024      | 131.841610      | 1.035485 | 1.045488 |         |
| 2023-12-01                                       | 135.50          | 134.727478      | 131.862936      | 1.021724 | 1.005734 |         |
| 2024-01-01                                       | 118.50          | 128.820183      | 131.435527      | 0.980102 | 0.919887 |         |
| 2024-02-01                                       | 126.65          | 130.047806      | 130.901954      | 0.993475 | 0.973873 |         |
| 2024-03-01                                       | 135.37          | 128.243395      | 130.915972      | 0.979586 | 1.055571 |         |
| <br>[351 rows x 5 columns],                      |                 |                 |                 |          |          |         |
| 'Other_Manufacturing_products':                  |                 |                 |                 |          | observed | seasadj |
| d  | irregular       | seasonal_factor |                 |          | tren     |         |
| 1995-01-01                                       | 258.65          | 277.956074      | 287.537364      | 0.966678 | 0.930543 |         |
| 1995-02-01                                       | 260.53          | 298.277739      | 295.146081      | 1.010611 | 0.873448 |         |
| 1995-03-01                                       | 348.30          | 354.770991      | 304.599920      | 1.164711 | 0.981760 |         |
| 1995-04-01                                       | 249.90          | 272.584446      | 315.213730      | 0.864761 | 0.916780 |         |
| 1995-05-01                                       | 323.11          | 314.595273      | 326.891255      | 0.962385 | 1.027066 |         |
| ...  | ...             | ...             | ...             | ...      | ...      |         |
| 2023-11-01                                       | 877.52          | 874.222213      | 860.587073      | 1.015844 | 1.003772 |         |
| 2023-12-01                                       | 824.87          | 839.350318      | 856.725104      | 0.979720 | 0.982748 |         |

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 2024-01-01 | 790.58 | 853.542889 | 856.736920 | 0.996272 | 0.926233 |
| 2024-02-01 | 850.15 | 861.954482 | 860.779759 | 1.001365 | 0.986305 |
| 2024-03-01 | 932.87 | 860.812399 | 867.216407 | 0.992615 | 1.083709 |

[351 rows x 5 columns],  
 'Coverage\_Adjustment': observed seasadj trend irreg  
 ular seasonal\_factor

|            |         |             |             |             |          |
|------------|---------|-------------|-------------|-------------|----------|
| 1995-01-01 | -67.51  | -62.554744  | -68.445005  | 5.890261    | 1.079215 |
| 1995-02-01 | -46.11  | -49.701502  | -70.507466  | 20.805964   | 0.927739 |
| 1995-03-01 | -105.91 | -98.411305  | -73.556724  | -24.854581  | 1.076197 |
| 1995-04-01 | -60.80  | -69.499132  | -77.424911  | 7.925779    | 0.874831 |
| 1995-05-01 | -91.42  | -77.890124  | -80.620844  | 2.730719    | 1.173705 |
| ...        | ...     | ...         | ...         | ...         | ...      |
| 2023-11-01 | -379.79 | -346.290689 | -345.316168 | -0.974521   | 1.096738 |
| 2023-12-01 | -411.25 | -399.231989 | -354.421557 | -44.810432  | 1.030103 |
| 2024-01-01 | -637.67 | -645.149553 | -362.748549 | -282.401004 | 0.988406 |
| 2024-02-01 | -348.41 | -353.982280 | -369.532959 | 15.550679   | 0.984258 |
| 2024-03-01 | -416.71 | -401.945151 | -374.486265 | -27.458886  | 1.036733 |

[351 rows x 5 columns],  
 'Exports,\_f.o.b.\_(BOP\_basis)': observed seasadj t  
 rend irregular seasonal\_factor

|            |          |              |              |          |         |
|------------|----------|--------------|--------------|----------|---------|
| 1995-01-01 | 3928.00  | 4243.919157  | 4235.054166  | 1.002093 | 0.92556 |
| 0          | 3949.00  | 4221.918657  | 4309.358550  | 0.979709 | 0.93535 |
| 7          | 5083.00  | 4855.120313  | 4398.666963  | 1.103771 | 1.04693 |
| 6          | 4005.00  | 4408.883868  | 4498.883273  | 0.979995 | 0.90839 |
| 3          | 4789.00  | 4627.514602  | 4598.960977  | 1.006209 | 1.03489 |
| 7          | ...      | ...          | ...          | ...      | ...     |
| ...        | ...      | ...          | ...          | ...      | ...     |
| 2023-11-01 | 23099.92 | 23596.578998 | 23688.218268 | 0.996131 | 0.97895 |
| 2          | 22380.30 | 23288.243046 | 23617.376905 | 0.986064 | 0.96101 |
| 3          | 22012.20 | 23717.439598 | 23428.258435 | 1.012343 | 0.92810 |
| 2          | 23036.45 | 23536.171581 | 23182.778518 | 1.015244 | 0.97876 |
| 8          | 24543.84 | 22526.500319 | 23005.343488 | 0.979186 | 1.08955 |
| 4          | ...      | ...          | ...          | ...      | ...     |

[351 rows x 5 columns],  
 'Electronics': observed seasadj trend irregular s  
 easonal\_factor

|            |         |             |             |          |          |
|------------|---------|-------------|-------------|----------|----------|
| 1995-01-01 | 576.14  | 601.055200  | 623.252180  | 0.964385 | 0.958548 |
| 1995-02-01 | 596.49  | 629.921553  | 651.170644  | 0.967368 | 0.946927 |
| 1995-03-01 | 758.60  | 690.991431  | 681.718661  | 1.013602 | 1.097843 |
| 1995-04-01 | 648.90  | 714.215023  | 712.583384  | 1.002290 | 0.908550 |
| 1995-05-01 | 732.16  | 754.801626  | 739.930729  | 1.020098 | 0.970003 |
| ...        | ...     | ...         | ...         | ...      | ...      |
| 2023-11-01 | 2904.33 | 2884.722243 | 2861.359995 | 1.008165 | 1.006797 |
| 2023-12-01 | 3146.57 | 2845.736892 | 2883.820730 | 0.986794 | 1.105714 |

|            |         |             |             |          |          |
|------------|---------|-------------|-------------|----------|----------|
| 2024-01-01 | 2709.52 | 3021.104407 | 2901.508753 | 1.041218 | 0.896864 |
| 2024-02-01 | 2565.01 | 2876.544768 | 2905.766370 | 0.989944 | 0.891698 |
| 2024-03-01 | 3093.50 | 2811.410580 | 2905.587708 | 0.967588 | 1.100337 |

[351 rows x 5 columns],  
 'Photographic\_&\_cinematographic\_instruments\_&\_supplies': observ  
 ed seasadj trend irregular seasonal\_factor

|            |        |            |            |          |          |
|------------|--------|------------|------------|----------|----------|
| 1996-11-01 | 50.73  | 54.575032  | 48.372696  | 1.128220 | 0.929546 |
| 1996-12-01 | 35.99  | 45.784966  | 46.256731  | 0.989801 | 0.786066 |
| 1997-01-01 | 38.98  | 39.209009  | 43.863557  | 0.893886 | 0.994159 |
| 1997-02-01 | 41.45  | 43.150772  | 41.619945  | 1.036781 | 0.960585 |
| 1997-03-01 | 38.83  | 38.311014  | 39.634888  | 0.966598 | 1.013547 |
| ...        | ...    | ...        | ...        | ...      | ...      |
| 2023-11-01 | 257.71 | 229.626395 | 237.335315 | 0.967519 | 1.122301 |
| 2023-12-01 | 210.06 | 218.213626 | 237.909262 | 0.917214 | 0.962635 |
| 2024-01-01 | 225.98 | 246.617053 | 239.871457 | 1.028122 | 0.916319 |
| 2024-02-01 | 194.39 | 241.089071 | 242.411187 | 0.994546 | 0.806300 |
| 2024-03-01 | 223.52 | 255.570698 | 244.876640 | 1.043671 | 0.874592 |

[329 rows x 5 columns],  
 'Electrical\_Appliances': observed seasadj trend ir  
 regular seasonal\_factor

|            |         |             |             |          |          |
|------------|---------|-------------|-------------|----------|----------|
| 1995-01-01 | 291.50  | 315.570730  | 326.631451  | 0.966137 | 0.923723 |
| 1995-02-01 | 328.45  | 330.143902  | 338.457057  | 0.975438 | 0.994869 |
| 1995-03-01 | 416.38  | 360.941142  | 352.002629  | 1.025393 | 1.153595 |
| 1995-04-01 | 300.11  | 314.799228  | 366.034510  | 0.860026 | 0.953338 |
| 1995-05-01 | 407.51  | 383.269964  | 379.345464  | 1.010345 | 1.063245 |
| ...        | ...     | ...         | ...         | ...      | ...      |
| 2023-11-01 | 2176.58 | 2256.136448 | 2218.051401 | 1.017170 | 0.964738 |
| 2023-12-01 | 2078.35 | 2213.995419 | 2218.985145 | 0.997751 | 0.938733 |
| 2024-01-01 | 2145.42 | 2225.129989 | 2228.034259 | 0.998696 | 0.964177 |
| 2024-02-01 | 2323.37 | 2252.791472 | 2246.760710 | 1.002684 | 1.031329 |
| 2024-03-01 | 2541.25 | 2152.777238 | 2273.836320 | 0.946760 | 1.180452 |

[351 rows x 5 columns],  
 'Metal\_&\_Steel': observed seasadj trend irregular  
 seasonal\_factor

|            |         |             |             |          |          |
|------------|---------|-------------|-------------|----------|----------|
| 1995-01-01 | 92.82   | 108.321214  | 108.804251  | 0.995560 | 0.856896 |
| 1995-02-01 | 98.79   | 106.354120  | 109.098102  | 0.974848 | 0.928878 |
| 1995-03-01 | 144.15  | 136.665595  | 110.191672  | 1.240253 | 1.054764 |
| 1995-04-01 | 103.55  | 113.586203  | 111.319969  | 1.020358 | 0.911642 |
| 1995-05-01 | 113.93  | 111.188668  | 112.665720  | 0.986890 | 1.024655 |
| ...        | ...     | ...         | ...         | ...      | ...      |
| 2023-11-01 | 1416.29 | 1456.062331 | 1281.759511 | 1.135987 | 0.972685 |
| 2023-12-01 | 1315.82 | 1327.495661 | 1288.894098 | 1.029949 | 0.991205 |
| 2024-01-01 | 1747.95 | 1871.687947 | 1285.655434 | 1.455824 | 0.933890 |
| 2024-02-01 | 1287.47 | 1309.275712 | 1271.736276 | 1.029518 | 0.983345 |
| 2024-03-01 | 1316.90 | 1210.800265 | 1252.897845 | 0.966400 | 1.087628 |

[351 rows x 5 columns],  
 'Automotive': observed seasadj trend irregular se  
 asonal\_factor

|            |       |           |           |          |          |
|------------|-------|-----------|-----------|----------|----------|
| 1995-01-01 | 51.25 | 63.202547 | 70.626374 | 0.894886 | 0.810885 |
| 1995-02-01 | 64.37 | 71.822939 | 76.588678 | 0.937775 | 0.896232 |
| 1995-03-01 | 87.42 | 85.739140 | 82.601837 | 1.037981 | 1.019604 |
| 1995-04-01 | 72.98 | 89.304055 | 88.074416 | 1.013961 | 0.817208 |

## 1\_composite\_export\_value\_analysis

|            |         |             |             |          |          |
|------------|---------|-------------|-------------|----------|----------|
| 1995-05-01 | 91.14   | 95.889952   | 91.753442   | 1.045083 | 0.950465 |
| ...        | ...     | ...         | ...         | ...      | ...      |
| 2023-11-01 | 3576.63 | 3464.276518 | 3572.377005 | 0.969740 | 1.032432 |
| 2023-12-01 | 3640.17 | 3648.988031 | 3535.714395 | 1.032037 | 0.997583 |
| 2024-01-01 | 3470.70 | 3514.455166 | 3512.270715 | 1.000622 | 0.987550 |
| 2024-02-01 | 3501.28 | 3480.136569 | 3503.189498 | 0.993419 | 1.006075 |
| 2024-03-01 | 3951.42 | 3412.225722 | 3502.531129 | 0.974217 | 1.158018 |

|                                |         |             |             |          |          |
|--------------------------------|---------|-------------|-------------|----------|----------|
| [351 rows x 5 columns],        |         |             |             |          |          |
| 'Agro-manufacturing_Products': |         |             | observed    | seasadj  | trend    |
| nd irregular seasonal_factor   |         |             |             |          |          |
| 1995-01-01                     | 448.80  | 507.501387  | 501.857049  | 1.011247 | 0.884333 |
| 1995-02-01                     | 483.77  | 486.441968  | 497.050371  | 0.978657 | 0.994507 |
| 1995-03-01                     | 748.10  | 648.065296  | 655.559346  | 0.988568 | 1.154359 |
| 1995-04-01                     | 632.93  | 671.964974  | 642.271382  | 1.046232 | 0.941909 |
| 1995-05-01                     | 686.07  | 617.862684  | 625.112899  | 0.988402 | 1.110392 |
| ...                            | ...     | ...         | ...         | ...      | ...      |
| 2023-11-01                     | 3107.13 | 3157.824078 | 3141.958127 | 1.005050 | 0.983947 |
| 2023-12-01                     | 2846.54 | 3089.105091 | 3147.846698 | 0.981339 | 0.921477 |
| 2024-01-01                     | 2885.93 | 3181.671090 | 3143.501579 | 1.012142 | 0.907049 |
| 2024-02-01                     | 3159.12 | 3170.950513 | 3135.326759 | 1.011362 | 0.996269 |
| 2024-03-01                     | 3465.33 | 3081.840525 | 3129.738073 | 0.984696 | 1.124435 |

|                         |         |             |             |          |             |
|-------------------------|---------|-------------|-------------|----------|-------------|
| [351 rows x 5 columns], |         |             |             |          |             |
| 'Agriculture':          |         | observed    | seasadj     | trend    | irregular s |
| easonal_factor          |         |             |             |          |             |
| 1995-01-01              | 584.19  | 534.357489  | 553.959278  | 0.964615 | 1.093257    |
| 1995-02-01              | 550.96  | 550.444206  | 555.005956  | 0.991781 | 1.000937    |
| 1995-03-01              | 605.24  | 579.367787  | 552.130304  | 1.049332 | 1.044656    |
| 1995-04-01              | 453.51  | 488.499177  | 546.058446  | 0.894591 | 0.928374    |
| 1995-05-01              | 587.16  | 607.965365  | 534.785316  | 1.136840 | 0.965779    |
| ...                     | ...     | ...         | ...         | ...      | ...         |
| 2023-11-01              | 1552.72 | 1709.812490 | 1735.727090 | 0.985070 | 0.908123    |
| 2023-12-01              | 1394.08 | 1424.826397 | 1738.069198 | 0.819775 | 0.978421    |
| 2024-01-01              | 1536.73 | 1767.732439 | 1727.891972 | 1.023057 | 0.869323    |
| 2024-02-01              | 1446.91 | 1702.420910 | 1703.269093 | 0.999502 | 0.849913    |
| 2024-03-01              | 1576.87 | 1638.344876 | 1673.196652 | 0.979171 | 0.962477    |

|                         |        |            |            |          |                    |
|-------------------------|--------|------------|------------|----------|--------------------|
| [351 rows x 5 columns], |        |            |            |          |                    |
| 'Fishery':              |        | observed   | seasadj    | trend    | irregular seasonal |
| l_factor                |        |            |            |          |                    |
| 1995-01-01              | 212.42 | 247.018458 | 243.124514 | 1.016016 | 0.859936           |
| 1995-02-01              | 178.90 | 236.297214 | 243.985527 | 0.968489 | 0.757097           |
| 1995-03-01              | 248.28 | 264.017566 | 244.535164 | 1.079671 | 0.940392           |
| 1995-04-01              | 204.25 | 221.061824 | 245.527348 | 0.900355 | 0.923950           |
| 1995-05-01              | 241.44 | 246.588557 | 246.835139 | 0.999001 | 0.979121           |
| ...                     | ...    | ...        | ...        | ...      | ...                |
| 2023-11-01              | 131.44 | 127.075481 | 127.971463 | 0.992999 | 1.034346           |
| 2023-12-01              | 121.63 | 129.181601 | 131.700237 | 0.980876 | 0.941543           |
| 2024-01-01              | 129.63 | 158.630589 | 135.864291 | 1.167566 | 0.817182           |
| 2024-02-01              | 122.05 | 144.106992 | 139.948627 | 1.029714 | 0.846940           |
| 2024-03-01              | 148.89 | 142.087348 | 143.569382 | 0.989677 | 1.047877           |

|                         |  |          |         |       |                     |
|-------------------------|--|----------|---------|-------|---------------------|
| [351 rows x 5 columns], |  |          |         |       |                     |
| 'Mining':               |  | observed | seasadj | trend | irregular seasonal_ |
| factor                  |  |          |         |       |                     |

```

1995-01-01    22.02   23.977146   23.232542   1.032050   0.918375
1995-02-01    20.88   18.491921   23.256415   0.795132   1.129142
1995-03-01    19.80   19.891098   23.367219   0.851239   0.995420
1995-04-01    31.93   26.177024   23.588068   1.109757   1.219772
1995-05-01    22.58   25.452841   23.725735   1.072795   0.887131
...
2023-11-01    48.03   54.574050   77.395464   0.705132   0.880089
2023-12-01    75.94   77.687723   79.429592   0.978070   0.977503
2024-01-01    74.38   72.642362   81.731838   0.888789   1.023920
2024-02-01   113.29  101.801734   84.191358   1.209171   1.112849
2024-03-01    93.32   83.482423   86.694210   0.962953   1.117840

[351 rows x 5 columns],
'Manufacturing':          observed      seasadj       trend  irregula
r seasonal_factor
1995-01-01    3108.12   3423.991245   3417.212664   1.001984   0.90774
8
1995-02-01    3191.95   3420.748620   3507.233634   0.975341   0.93311
4
1995-03-01    4233.01   3991.474083   3614.358877   1.104338   1.06051
3
1995-04-01    3315.75   3645.275817   3734.057172   0.976224   0.90960
2
1995-05-01    3951.96   3901.760922   3855.068353   1.012112   1.01286
6
...
...
2023-11-01   21319.05   21453.865838   21339.634277   1.005353   0.99371
6
2023-12-01   20586.33   21219.311233   21316.164307   0.995456   0.97017
0
2024-01-01   20248.29   21708.280953   21217.729614   1.023120   0.93274
5
2024-02-01   20750.29   21262.011038   21071.453192   1.009043   0.97593
3
2024-03-01   22479.68   20526.314574   20965.972137   0.979030   1.09516
4

[351 rows x 5 columns]}

```

Analyse each of the individual component export timeseries (represented by cols)

## 1. Agriculture

```
In [4]: df = df_export_ANALYSIS.copy()
name = df.columns[0]
name
```

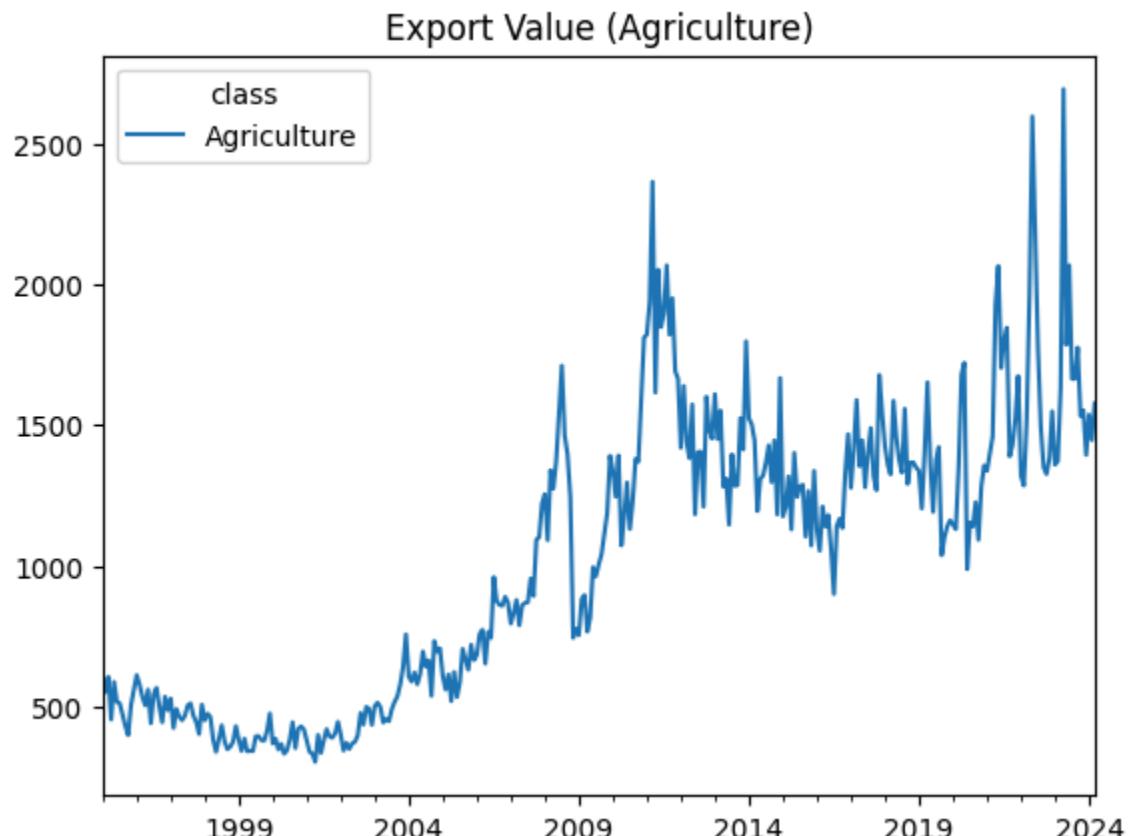
```
Out[4]: 'Agriculture'
```

```
In [5]: df = df_export_ANALYSIS[[name]]
display(df)
df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
```

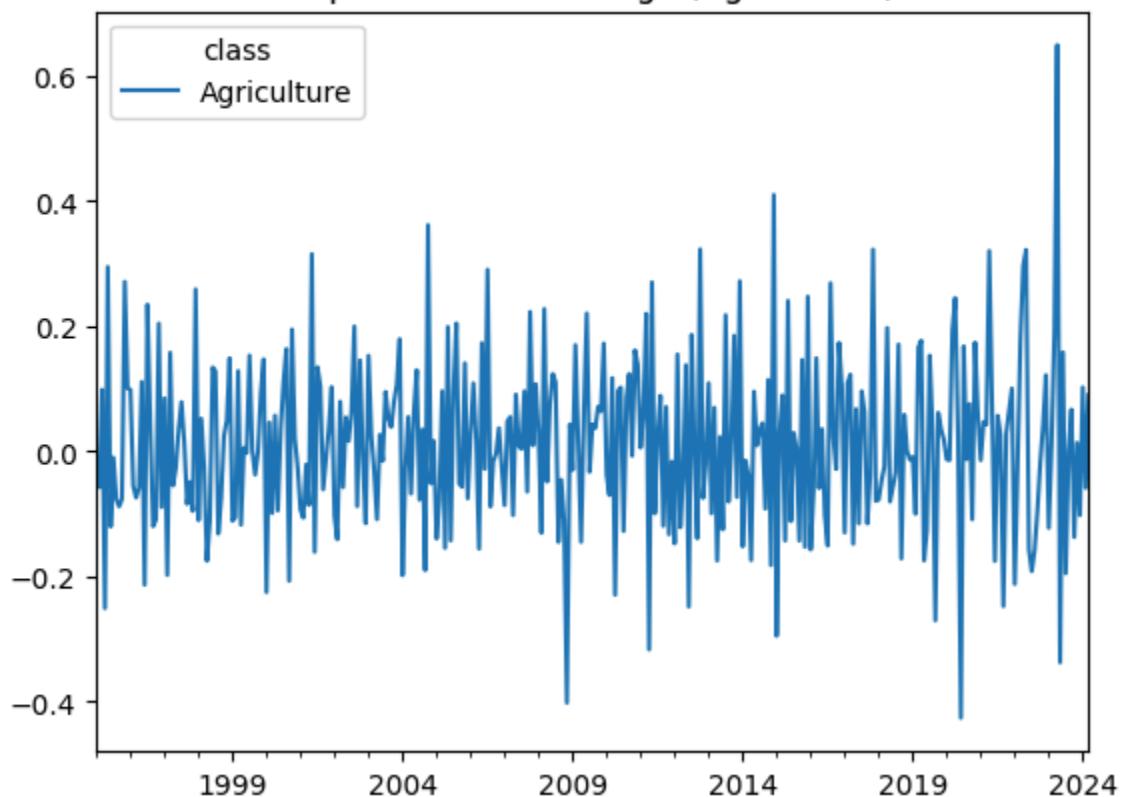
| class      | Agriculture |
|------------|-------------|
| 1995-01-01 | 584.19      |
| 1995-02-01 | 550.96      |
| 1995-03-01 | 605.24      |
| 1995-04-01 | 453.51      |
| 1995-05-01 | 587.16      |
| ...        | ...         |
| 2023-11-01 | 1552.72     |
| 2023-12-01 | 1394.08     |
| 2024-01-01 | 1536.73     |
| 2024-02-01 | 1446.91     |
| 2024-03-01 | 1576.87     |

351 rows × 1 columns

```
Out[5]: <Axes: title={'center': 'Export Value % Change (Agriculture)'}>
```



### Export Value % Change (Agriculture)



```
In [6]: ts = df.pct_change().dropna()  
ts
```

Out[6]:

| class      | Agriculture |
|------------|-------------|
| 1995-02-01 | -0.056882   |
| 1995-03-01 | 0.098519    |
| 1995-04-01 | -0.250694   |
| 1995-05-01 | 0.294701    |
| 1995-06-01 | -0.120393   |
| ...        | ...         |
| 2023-11-01 | 0.013478    |
| 2023-12-01 | -0.102169   |
| 2024-01-01 | 0.102326    |
| 2024-02-01 | -0.058449   |
| 2024-03-01 | 0.089819    |

350 rows × 1 columns

## check if the data timeseries & statistical assumptions validity (stationary, seasonality, autocorrelation)

```
In [7]: #freq 12 is used since the data is monthly, so the seasonality is 12 months
check_time_series_assumptions(ts, freq=12)
```

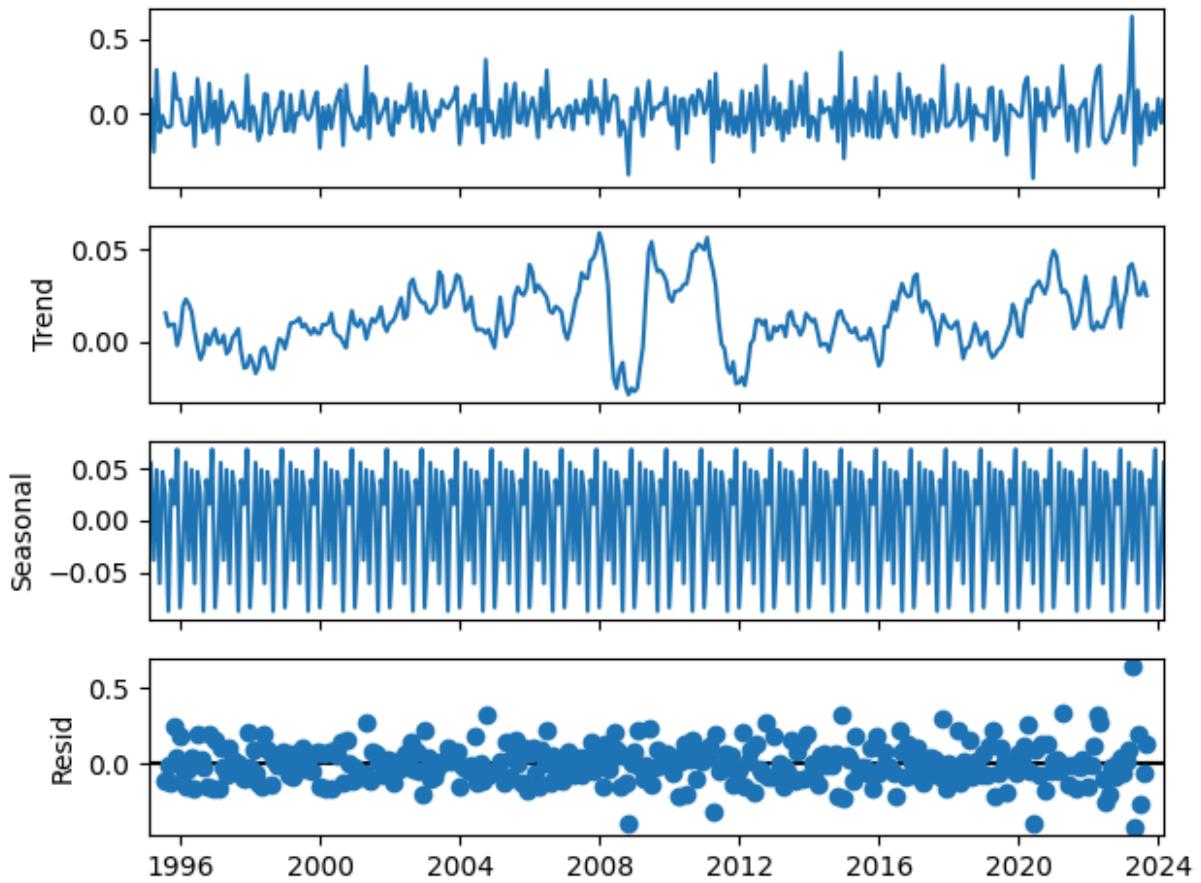
```
Running ADF Test for Stationarity...
ADF Statistic: -5.195082665088777
p-value: 8.997610064288814e-06
Critical Value 1%: -3.4502011472639724
Critical Value 5%: -2.8702852297358983
Critical Value 10%: -2.5714292194077513
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.11911592573952522
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

```
Decomposing the Series...
```

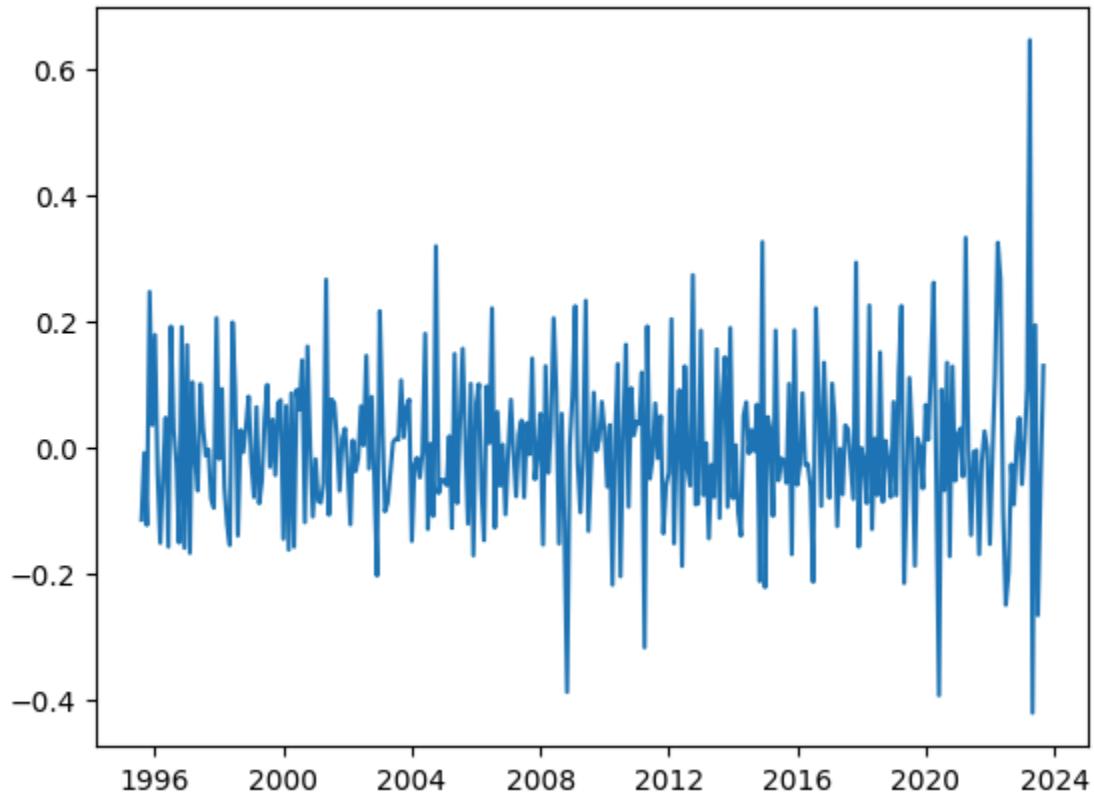
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

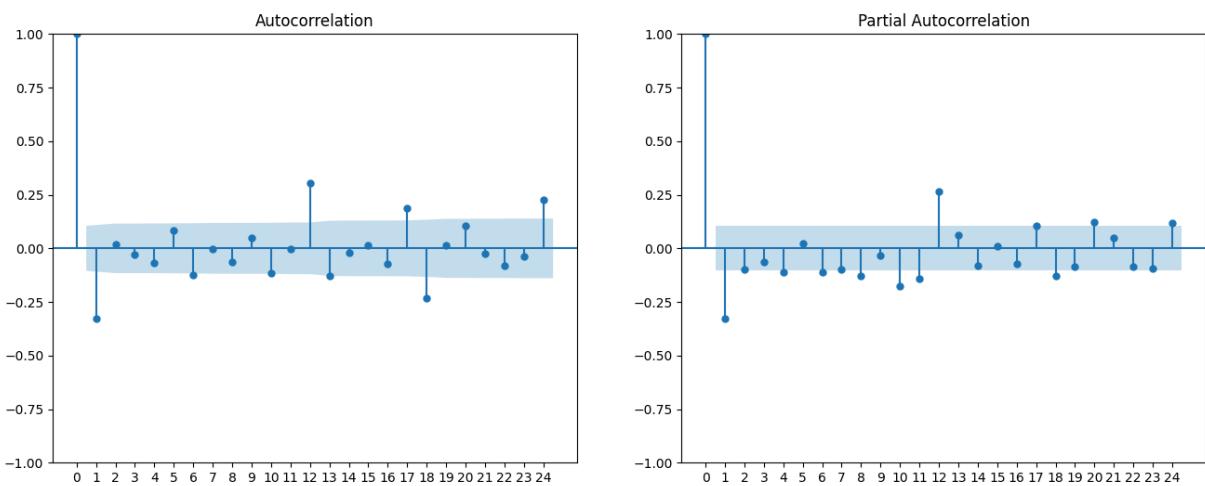
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

Out[7]: True

## Deseasonalise (by YoY and industry X-13)

**Setup for X-13 (ubuntu version, Windows OS can just be downloaded from the US Census page & add to PATH)**

Step 1: Download the X-13ARIMA-SEATS Executable Open a terminal on your Ubuntu machine. Download the Linux/Unix executable with ASCII output from the web:

```
https://www.census.gov/data/software/x13as.X-13ARIMA-SEATS.html
```

Step 2: Extract the Downloaded File Extract the tar.gz file to a directory of choice, cd there:

```
sudo tar -xzf x13as_V1.1_B39.tar.gz -C /opt/x13as
```

The extracted directory should contain the `x13as` executable.

Step 3: Change File permission to be executable w/o sudo

```
sudo chmod +x  
/home/wheelfredie/scripts/BoT_Exports/data/others/x13as/x13as_ascii
```

Step 4: Add to `$PATH`

```
nano ~/.bashrc
```

add:

```
export  
PATH="/home/wheelfredie/scripts/BoT_Exports/data/others/x13as:$PATH
```

reload bash and exit terminal

```
source ~/.bashrc
```

Step 5: Command should work

```
x13as_ascii
```

otherwise investigate path via `echo \$PATH' or directory permissions

X-13 deseasonality, ran and saved as .pkl files

```
In [8]: # X13-ARIMA-SEATS deseasonalization
# #DEsesonality using X13 by US Census Bureau
# from tqdm import tqdm
# import pickle
# import json
# import pandas as pd
# import numpy as np
# from statsmodels.tsa.x13 import x13_arima_analysis
# from statsmodels.tsa.x13 import X13Warning
# import warnings

# # Suppress specific X13 warnings
# warnings.filterwarnings("ignore", category=X13Warning)

# df = pd.read_csv('data/cleaned/total_export_ANALYSIS.csv', index_col=0)
# df.index = pd.to_datetime(df.index)

# x13_path='/home/wheelfredie/scripts/BoT_Exports/data/others/x13as'
# freq='M'

# dict_deseasonalized_value = {}
# dict_deseasonalized_pct_change = {}

# cols = df.columns.to_list()
# for col in tqdm(cols,
#                  desc='Deseasonalizing timeseries...'):
#     try:
#         #deseasonalize the value
#         ts = df[col].replace(0, 1e-10).ffill().bfill().dropna()

#         result = x13_arima_analysis(ts,
#                                     x12path=x13_path,
#                                     freq=freq)

#         # Convert components to DataFrame
#         df1 = pd.DataFrame({
#             'observed': result.observed,
#             'seasadj': result.seasadj,
#             'trend': result.trend,
#             'irregular': result.irregular,
```

```

#           'seasonal_factor': result.observed / result.seasadj
#       })
#   dict_deseasonalized_value[col] = df1

# except:
#     pass

# try:
#     #deseasonalize the pct_change of the value
#     ts = ts.pct_change().replace(0, 1e-10).dropna()

#     result = x13_arima_analysis(ts,
#                                 x12path=x13_path,
#                                 freq=freq)

#     # Convert components to DataFrame
#     df2 = pd.DataFrame({
#         'observed': result.observed,
#         'seasadj': result.seasadj,
#         'trend': result.trend,
#         'irregular': result.irregular,
#         'seasonal_factor': result.observed / result.seasadj
#     })

#     dict_deseasonalized_pct_change[col] = df2
# except:
#     pass

# #convert dictionary to pickle and json
# with open('/data/cleaned/deseasonalised_x13/dict_deseasonalized_value.pickle', 'wb') as handle:
#     pickle.dump(dict_deseasonalized_value, handle, protocol=pickle.HIGHEST_PROTOCOL)

# with open('/data/cleaned/deseasonalised_x13/dict_deseasonalized_pct_change.pickle', 'wb') as handle:
#     pickle.dump(dict_deseasonalized_pct_change, handle, protocol=pickle.HIGHEST_PROTOCOL)

#read the pickle file (UPDATED via 'update_sarimax.py' for latest code version)
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value.pickle', 'rb') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()

```

In [9]: `check_time_series_assumptions(adj_ts, freq=12)`

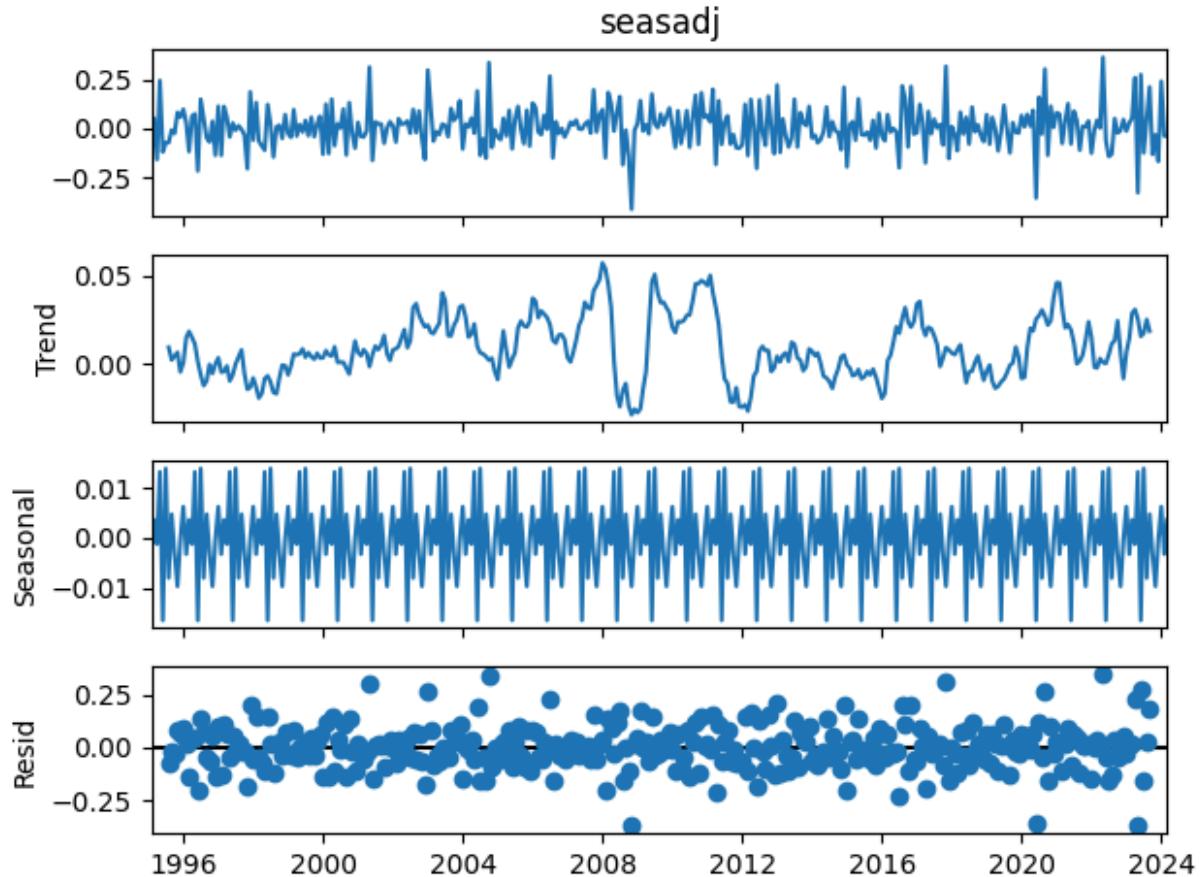
Running ADF Test for Stationarity...  
 ADF Statistic: -6.504929905828064  
 p-value: 1.137401698060269e-08  
 Critical Value 1%: -3.4499043309021955  
 Critical Value 5%: -2.870154885338925  
 Critical Value 10%: -2.5713597190254385  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.07228608151793027  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

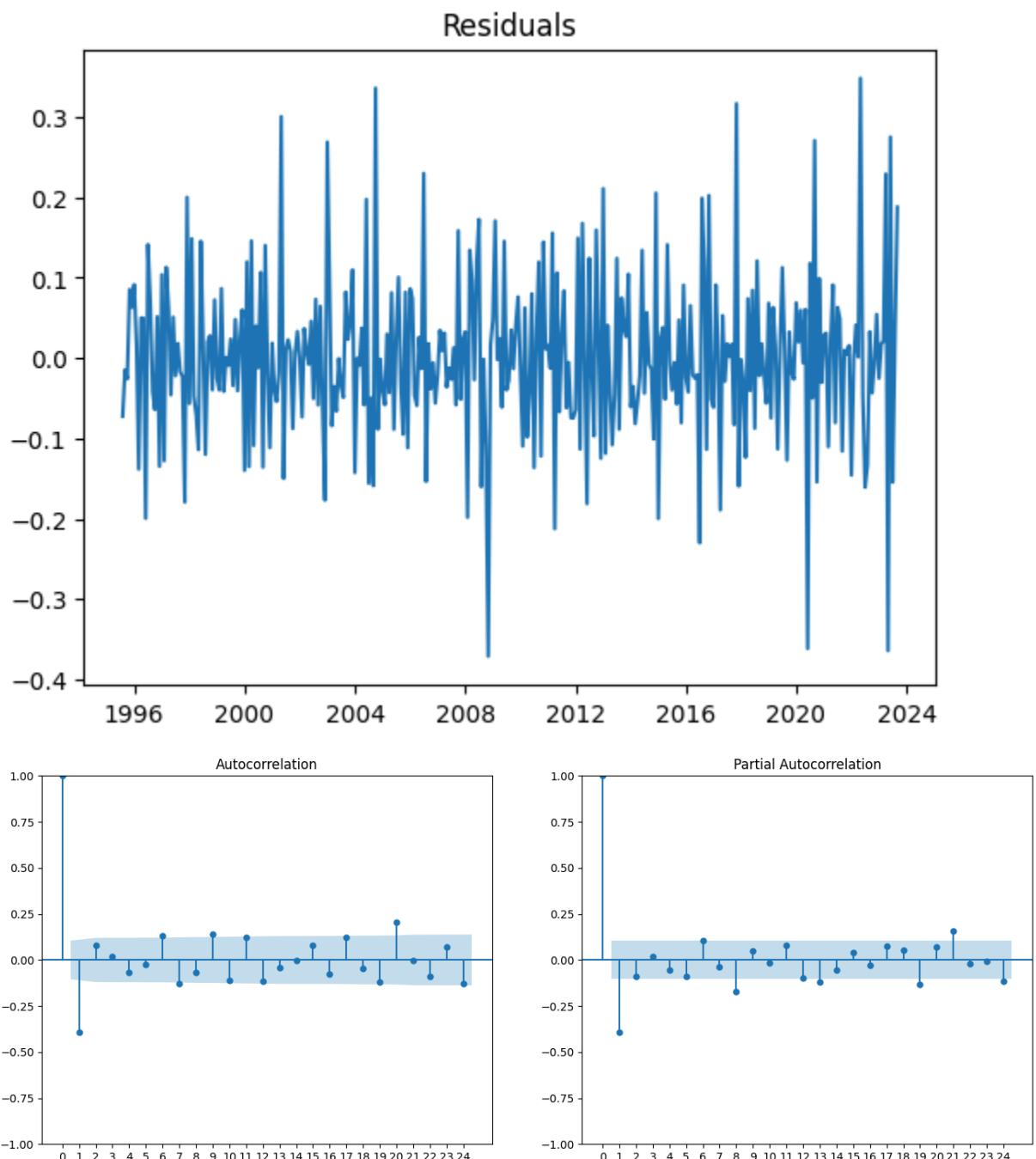
Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...



## Gridsearch for SARIMA parameters & validity

runtime: ~5minutes

```
In [10]: # #this is actualling done on a rolling walk forward basis for the timeserie
# model = auto_arima(adj_ts, seasonal=True, m=12, trace=True, error_action='stop'
# display(model.summary())
```

# Runtime ~40 minutes for rolling walkforward forecast, results saved as pickle

```
In [11]: # #Walk Forward parameters
# window_size = 180 # 15 years of monthly data
# total_observations = len(adj_ts) # Total data points
# steps = total_observations - window_size # Remaining points after initial

# # Generate periods DataFrame
# periods_df = generate_periods_df(adj_ts, window_size, steps)
# # Display the DataFrame with walk-forward periods
# display(periods_df)

# # ###
# # #FOR TESTING
# # periods_df = periods_df[:5] ###REMOVE----REMOVE---FOR--LIVE RUN#
# # ###

# # Perform walk-forward forecasting
# rolling_forecasts = walk_forward_forecasting(adj_ts, periods_df)
# # Display the first few predictions
# display(rolling_forecasts)

## consolidated df of actual vs forecasted values with error metrics
# forecast = pd.DataFrame(rolling_forecasts, columns=['forecast'])
# actual = adj_ts[adj_ts.index.isin(forecast.index)]

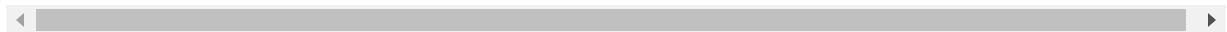
# df_accuracy = calculate_accuracy_metrics(actual, forecast)
# df_accuracy

...
As runtime is long,
67 + 4 extra = 71 Timeseries
Estimated runtime = 71 * (40/60)hours = >47+ hours with non linear compute s

Therefore this is pre-ran in parallel in bash terminal and loaded from the p
Pre-ran script path: "BoT_Exports/run_sarimax_RollWalkForward.py"
...
df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
display(df_accuracy)
```

| Agriculture | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|-------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01  | -0.095051 | -0.025370 | -0.069681 | 0.069681  | 0.004855      | 0.733089             |
| 2010-03-01  | 0.089923  | 0.027485  | 0.062439  | 0.062439  | 0.003899      | 0.694354             |
| 2010-04-01  | -0.075884 | -0.019145 | -0.056739 | 0.056739  | 0.003219      | 0.747703             |
| 2010-05-01  | -0.005640 | 0.037153  | -0.042793 | 0.042793  | 0.001831      | 7.587469             |
| 2010-06-01  | 0.091125  | -0.019013 | 0.110138  | 0.110138  | 0.012130      | 1.208641             |
| ...         | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-11-01  | -0.031345 | 0.042831  | -0.074176 | 0.074176  | 0.005502      | 2.366407             |
| 2023-12-01  | -0.166677 | 0.061904  | -0.228580 | 0.228580  | 0.052249      | 1.371399             |
| 2024-01-01  | 0.240665  | 0.079977  | 0.160688  | 0.160688  | 0.025821      | 0.667683             |
| 2024-02-01  | -0.036947 | -0.069714 | 0.032768  | 0.032768  | 0.001074      | 0.886891             |
| 2024-03-01  | -0.037638 | -0.004297 | -0.033341 | 0.033341  | 0.001112      | 0.885822             |

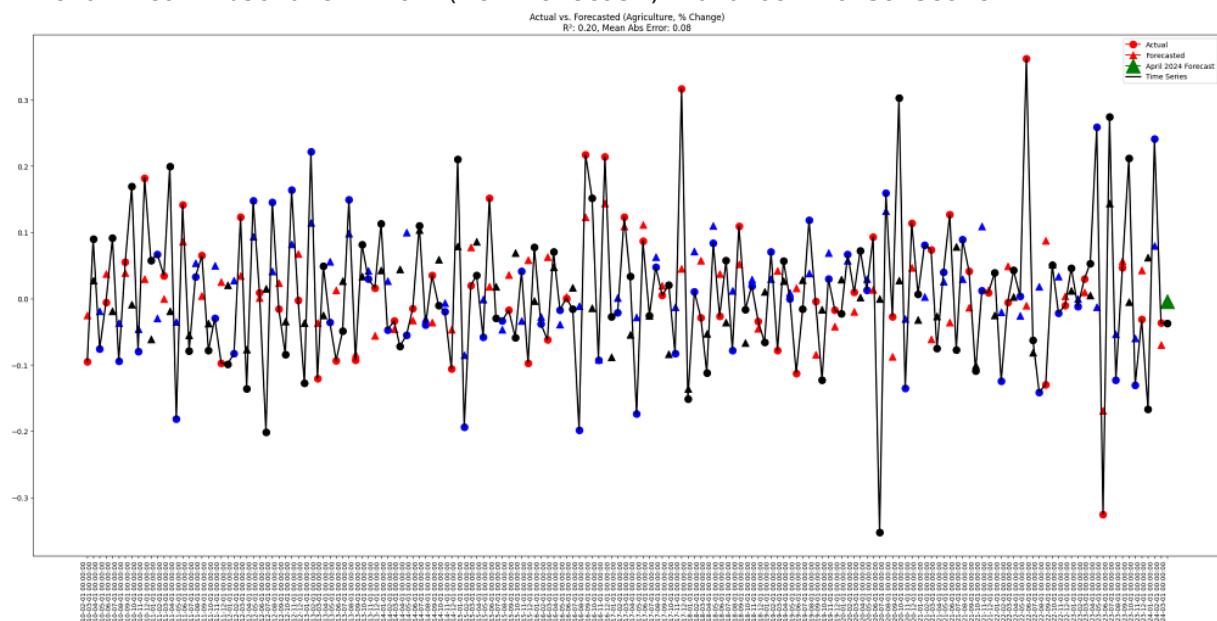
170 rows × 6 columns

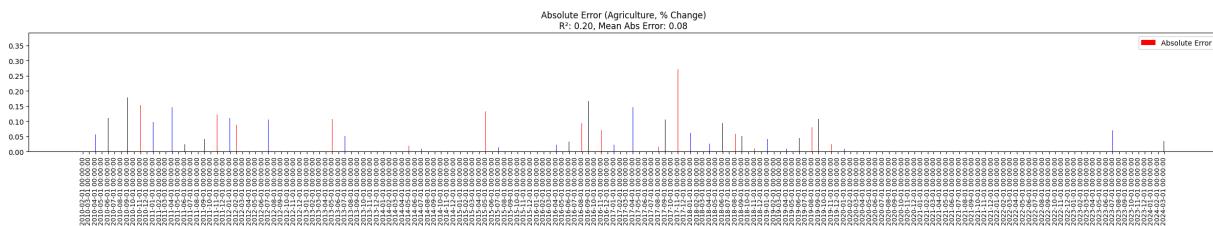


## Forecast are all Out-Of-Sample on a rolling basis

```
In [12]: average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Total Mean Absolute Error (Per Forecast): 0.07682719459738918'





## These is repeated for analysis on the other timeseries

(However it is compressed into a cell for simplicity of read unless there is an issue with the timeseries in terms or statistical properties which requires further analysis/transformation/stabilization)

## Rice

```
In [13]: df = df_export_ANALYSIS.copy()
name = df.columns[1]
display(f"Component: {name}")
```

'Component: Rice'

```
In [14]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_values.pkl', 'rb') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
```

```

display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

| class      | Rice      |
|------------|-----------|
| 1995-02-01 | -0.052947 |
| 1995-03-01 | -0.056713 |
| 1995-04-01 | -0.385163 |
| 1995-05-01 | 0.770071  |
| 1995-06-01 | -0.315093 |
| ...        | ...       |
| 2023-11-01 | 0.216445  |
| 2023-12-01 | -0.165054 |
| 2024-01-01 | 0.131926  |
| 2024-02-01 | -0.130425 |
| 2024-03-01 | -0.074727 |

350 rows × 1 columns

```

'-----'
-----
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -6.487194758089419
p-value: 1.252349578834437e-08
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True

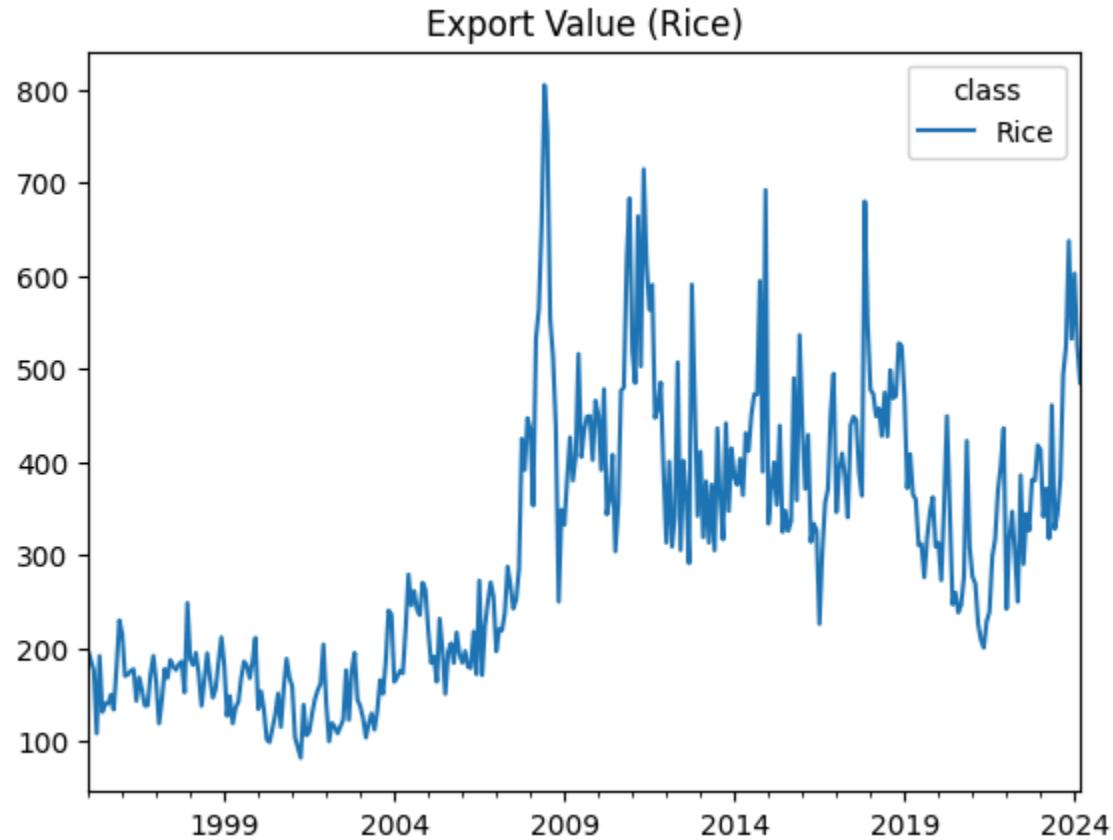
Running KPSS Test for Stationarity...
KPSS Statistic: 0.031331742245834
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True

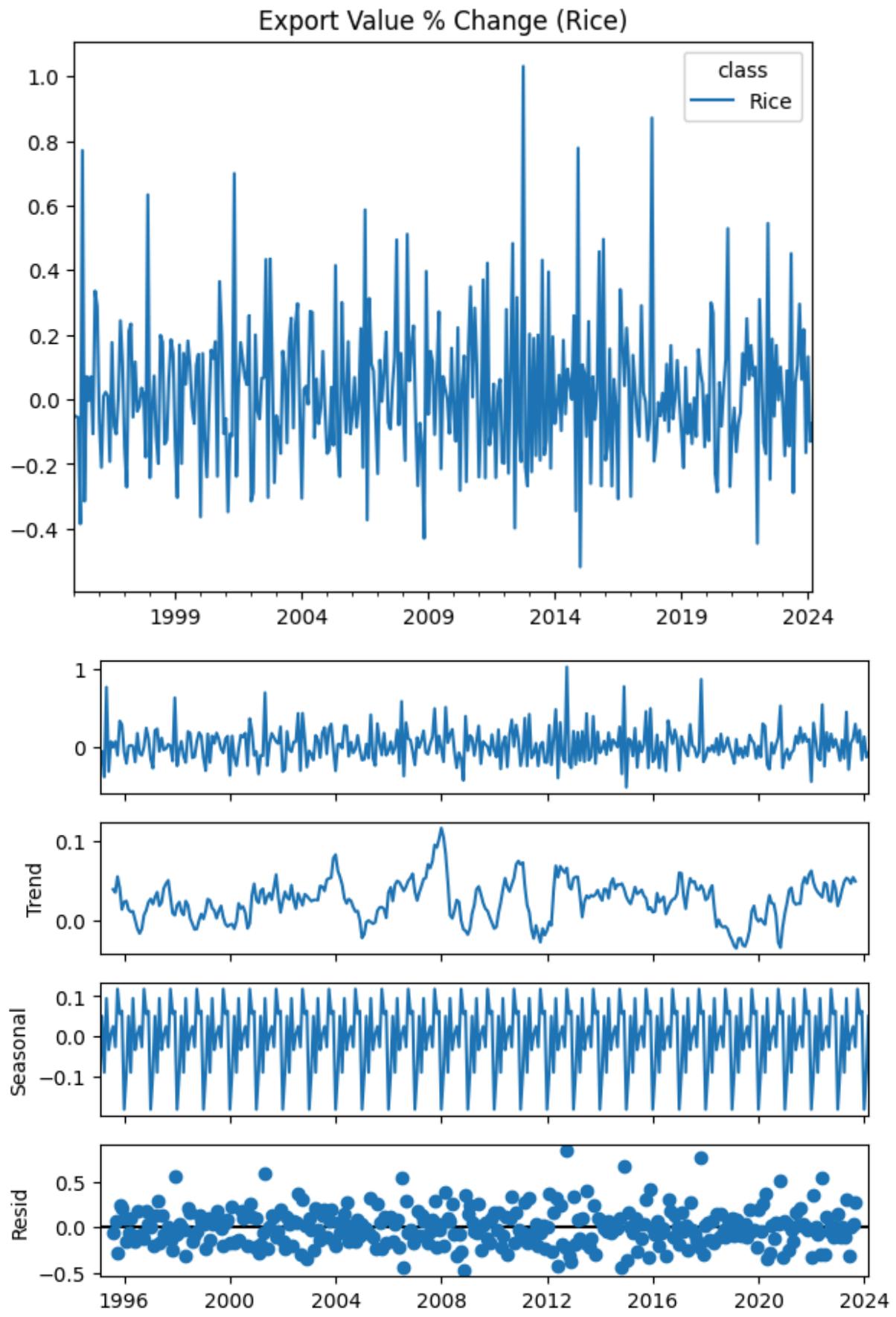
Decomposing the Series...

```

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th  
e test statistic is outside of the range of p-values available in the  
look-up table. The actual p-value is greater than the p-value returned.
```

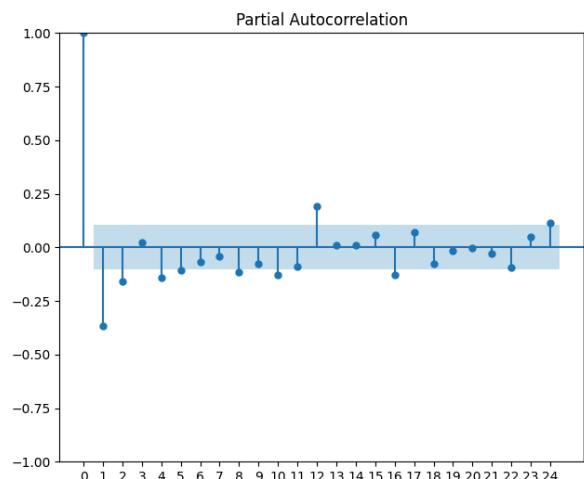
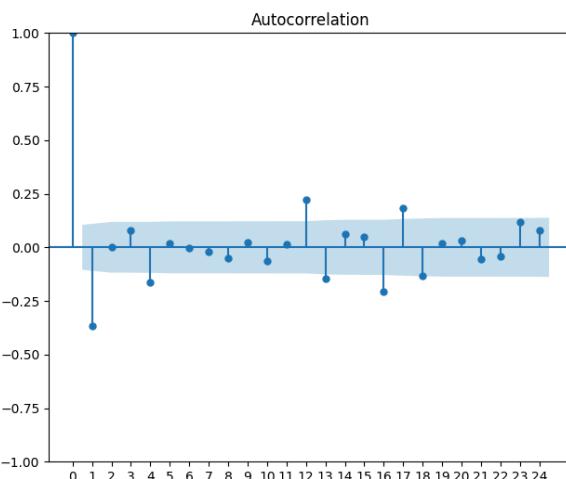
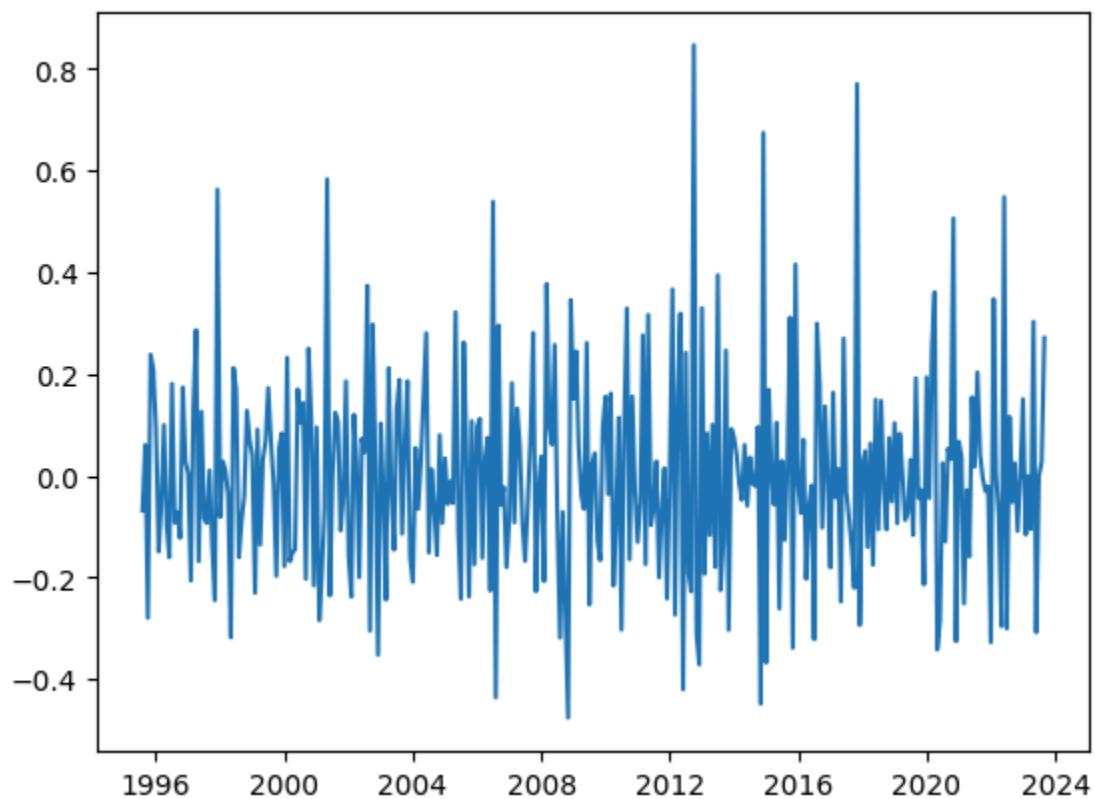
```
result = kpss(series, regression='c')
```





Plotting Residuals for Homoscedasticity Check...

### Residuals



Time series assumptions are met.

True

'-----'  
-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

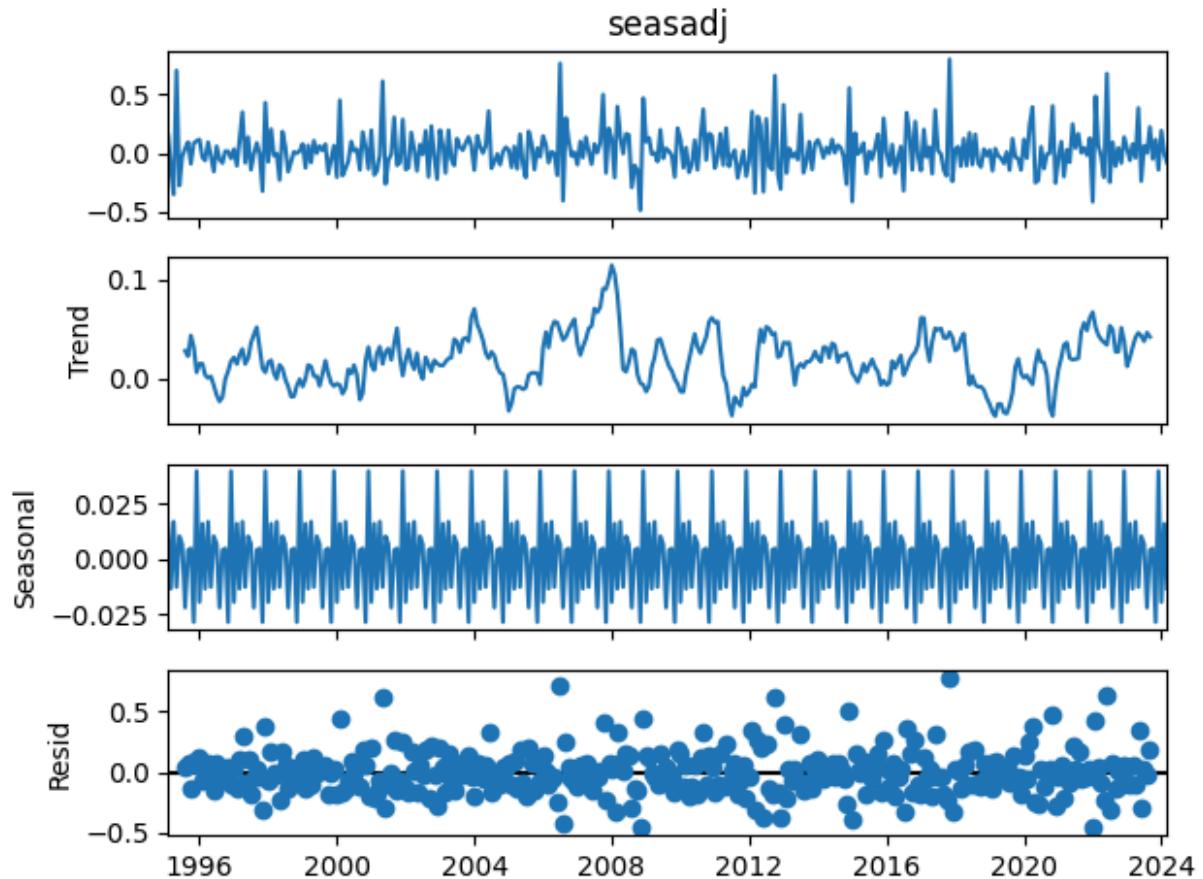
Running ADF Test for Stationarity...  
 ADF Statistic: -10.68959662059167  
 p-value: 3.755641134637406e-19  
 Critical Value 1%: -3.4494474563375737  
 Critical Value 5%: -2.8699542285903887  
 Critical Value 10%: -2.5712527305187987  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.036948282148716996  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

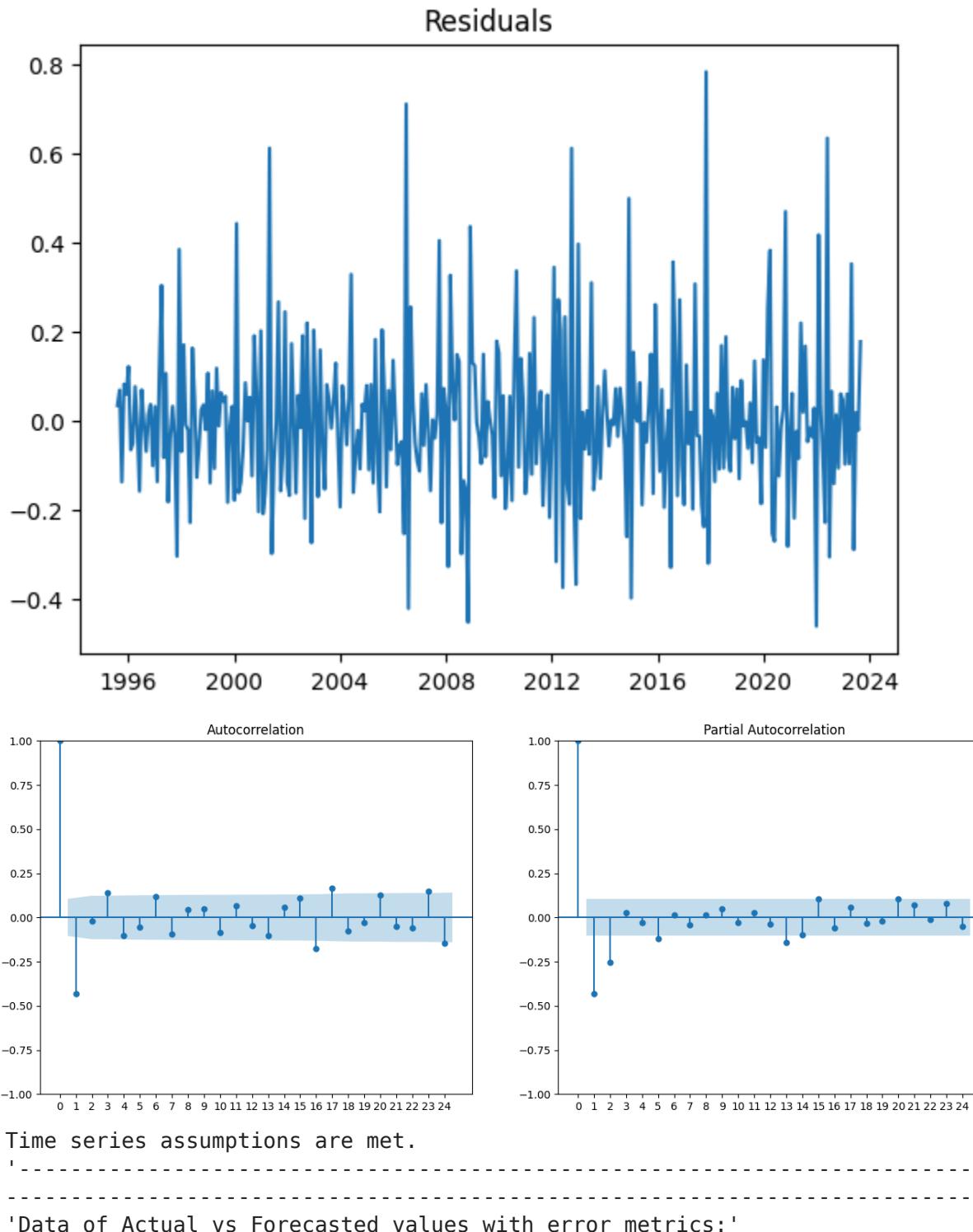
Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

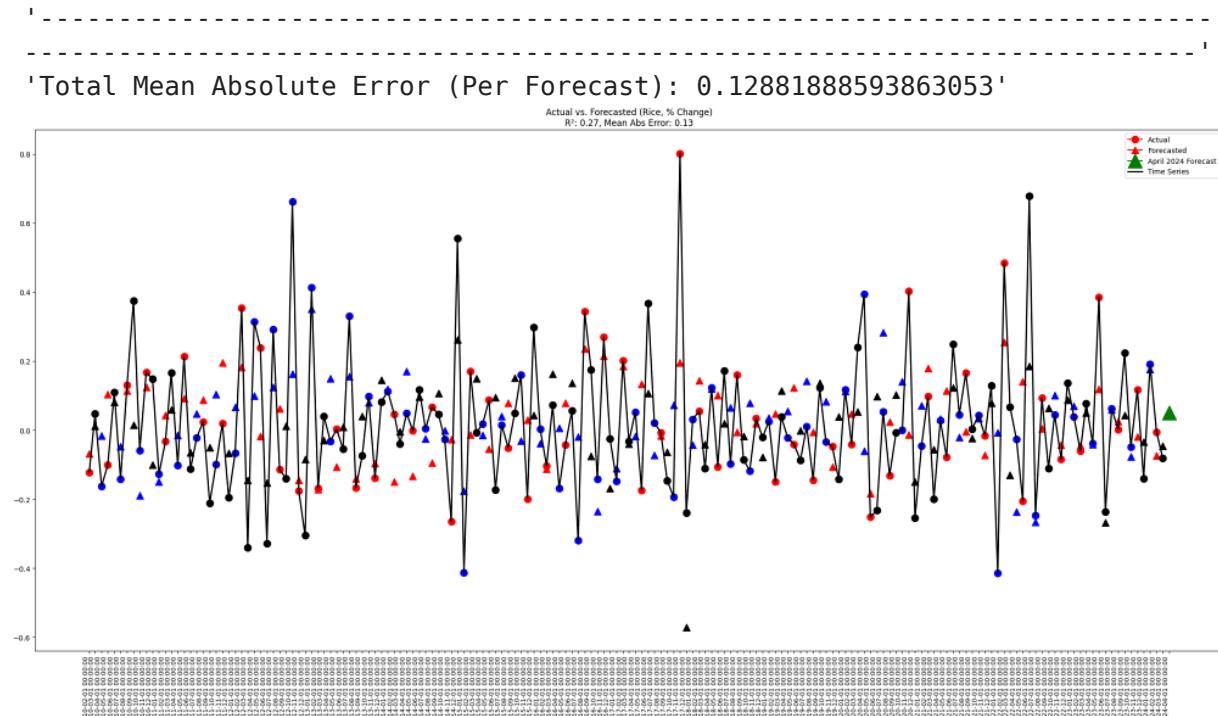


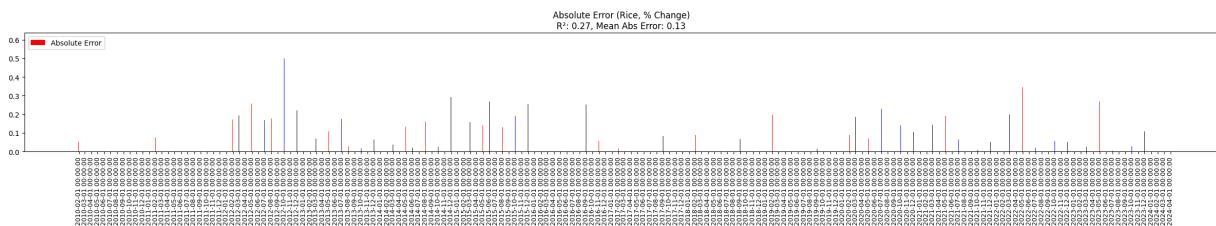
Plotting Residuals for Homoscedasticity Check...



| Rice       | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01 | -0.122352 | -0.068625 | -0.053727 | 0.053727  | 0.002887      | 0.439119             |
| 2010-03-01 | 0.047302  | 0.011891  | 0.035410  | 0.035410  | 0.001254      | 0.748604             |
| 2010-04-01 | -0.163013 | -0.016417 | -0.146596 | 0.146596  | 0.021490      | 0.899287             |
| 2010-05-01 | -0.100385 | 0.103554  | -0.203940 | 0.203940  | 0.041591      | 2.031569             |
| 2010-06-01 | 0.109553  | 0.081706  | 0.027847  | 0.027847  | 0.000775      | 0.254188             |
| ...        | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01 | -0.141274 | -0.033351 | -0.107922 | 0.107922  | 0.011647      | 0.763925             |
| 2024-01-01 | 0.191460  | 0.175817  | 0.015642  | 0.015642  | 0.000245      | 0.081701             |
| 2024-02-01 | -0.005778 | -0.074062 | 0.068284  | 0.068284  | 0.004663      | 11.817517            |
| 2024-03-01 | -0.080591 | -0.046221 | -0.034370 | 0.034370  | 0.001181      | 0.426471             |
| 2024-04-01 | NaN       | 0.050789  | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns





## Rubber

```
In [15]: df = df_export_ANALYSIS.copy()
name = df.columns[2]
display(f"Component: {name}")
```

'Component: Rubber'

```
In [16]: df = df_export_ANALYSIS[[name]]
```

```
df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value.pkl', 'rb') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f'data/cleaned/SARIMA_RollWalkForward/combined/{name}.pkl')
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Rubber    |
|------------|-----------|
| 1995-02-01 | -0.055667 |
| 1995-03-01 | 0.100421  |
| 1995-04-01 | -0.239486 |
| 1995-05-01 | 0.197558  |
| 1995-06-01 | 0.156848  |
| ...        | ...       |
| 2023-11-01 | 0.051189  |
| 2023-12-01 | -0.152006 |
| 2024-01-01 | 0.065904  |
| 2024-02-01 | 0.282161  |
| 2024-03-01 | 0.060892  |

350 rows × 1 columns

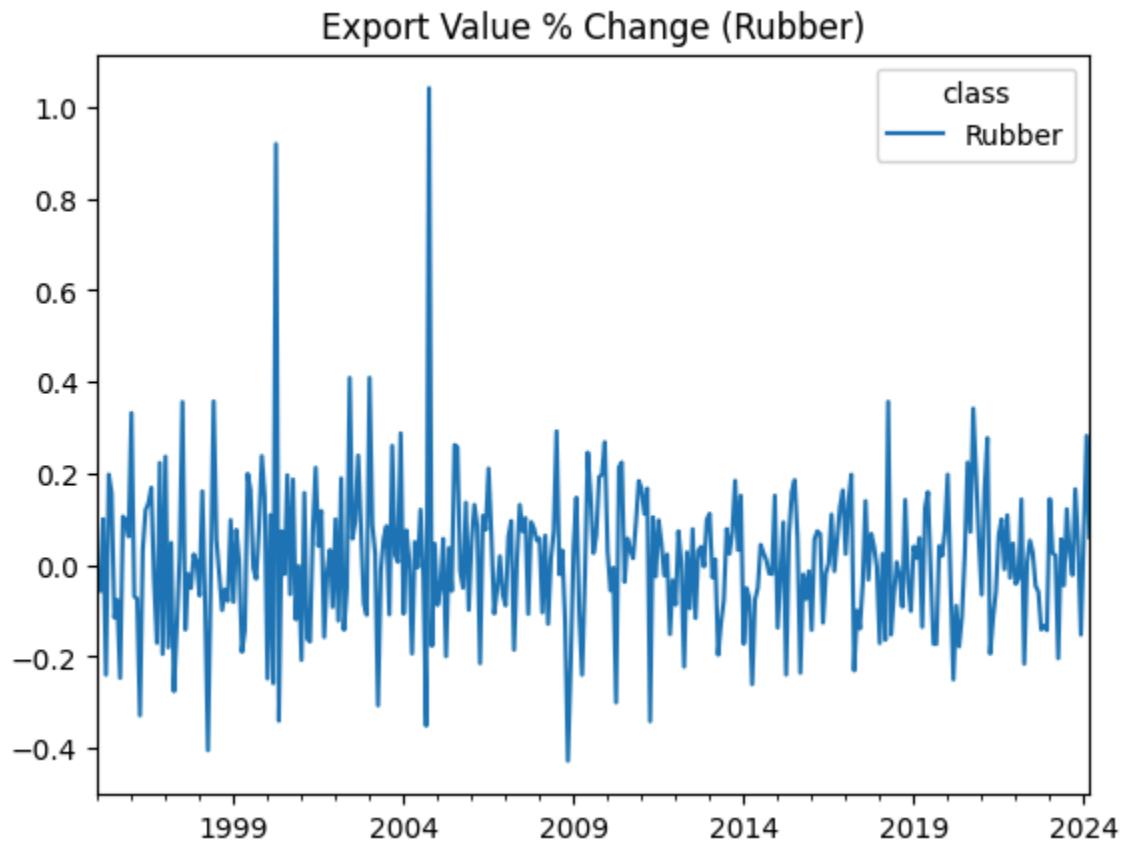
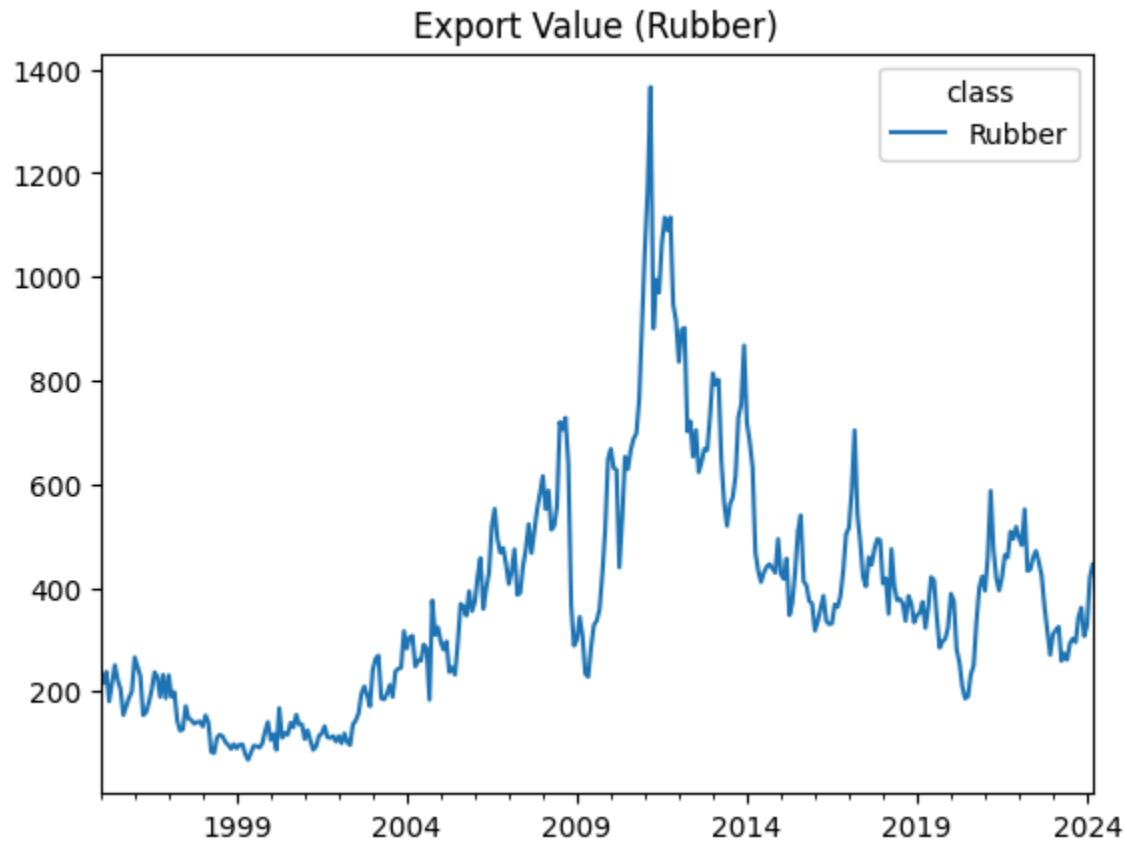
```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -13.106077679704486
p-value: 1.671466152784553e-24
Critical Value 1%: -3.449336554273722
Critical Value 5%: -2.8699055166063085
Critical Value 10%: -2.571226758215748
is_stationary: True
```

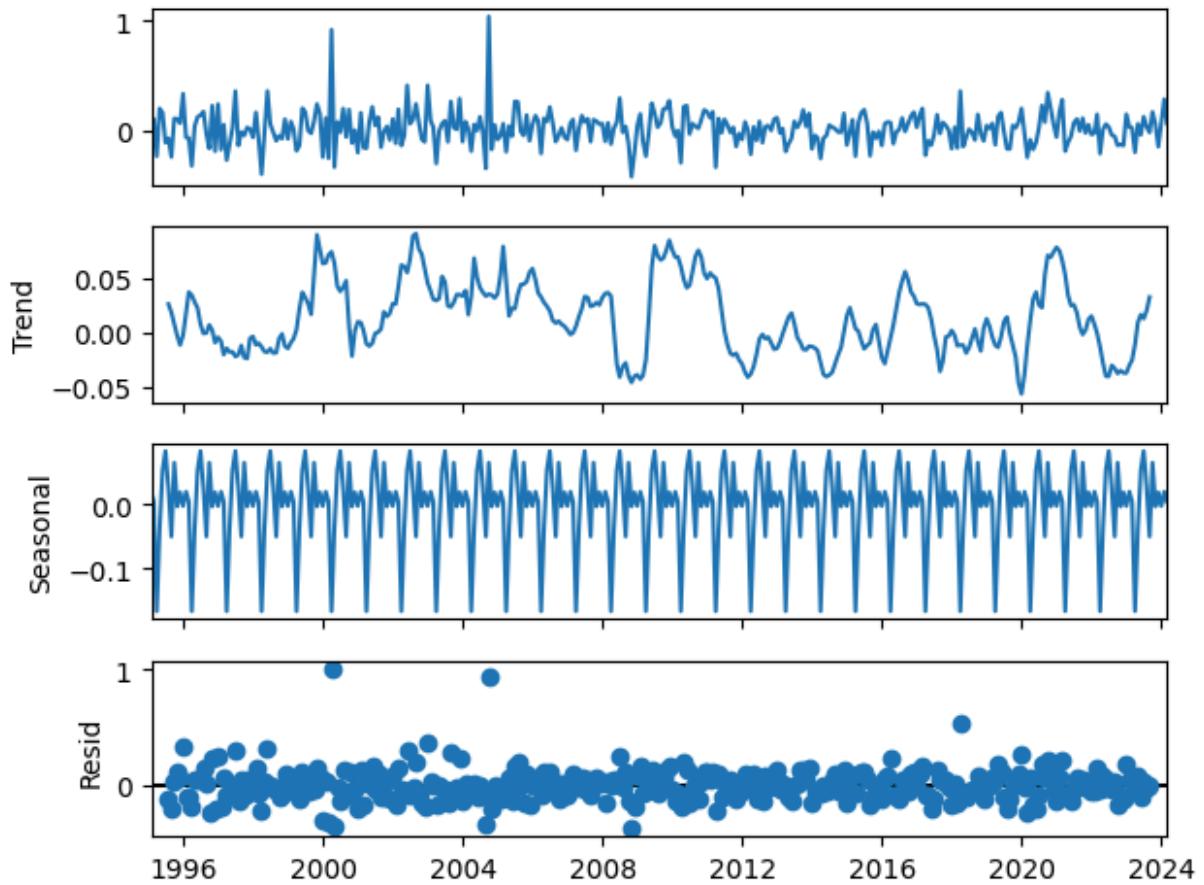
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.1310051968820044
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

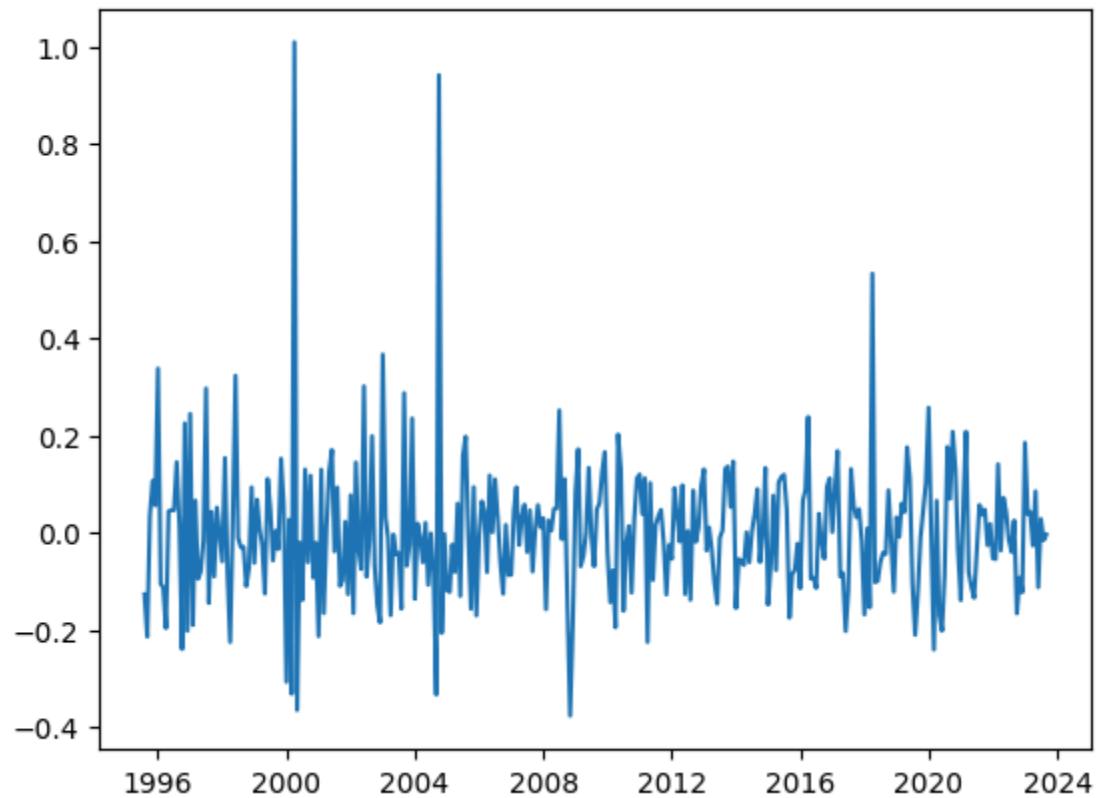
```
result = kpss(series, regression='c')
```

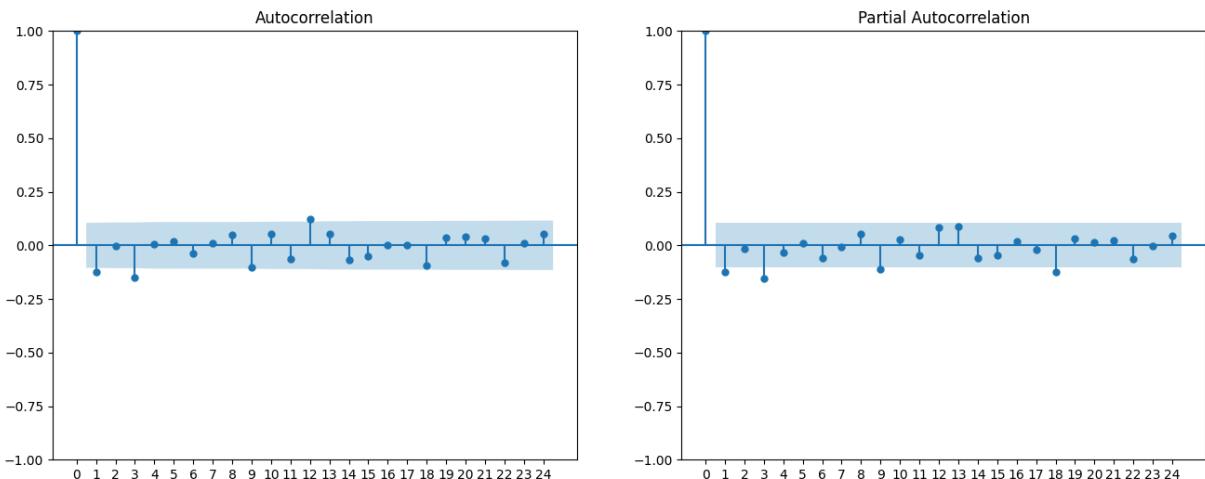




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

Running ADF Test for Stationarity...

ADF Statistic: -22.12214900240001

p-value: 0.0

Critical Value 1%: -3.449226932880019

Critical Value 5%: -2.869857365438656

Critical Value 10%: -2.571201085130664

is\_stationary: True

Running KPSS Test for Stationarity...

KPSS Statistic: 0.14365447105696572

p-value: 0.1

Critical Value 10%: 0.347

Critical Value 5%: 0.463

Critical Value 2.5%: 0.574

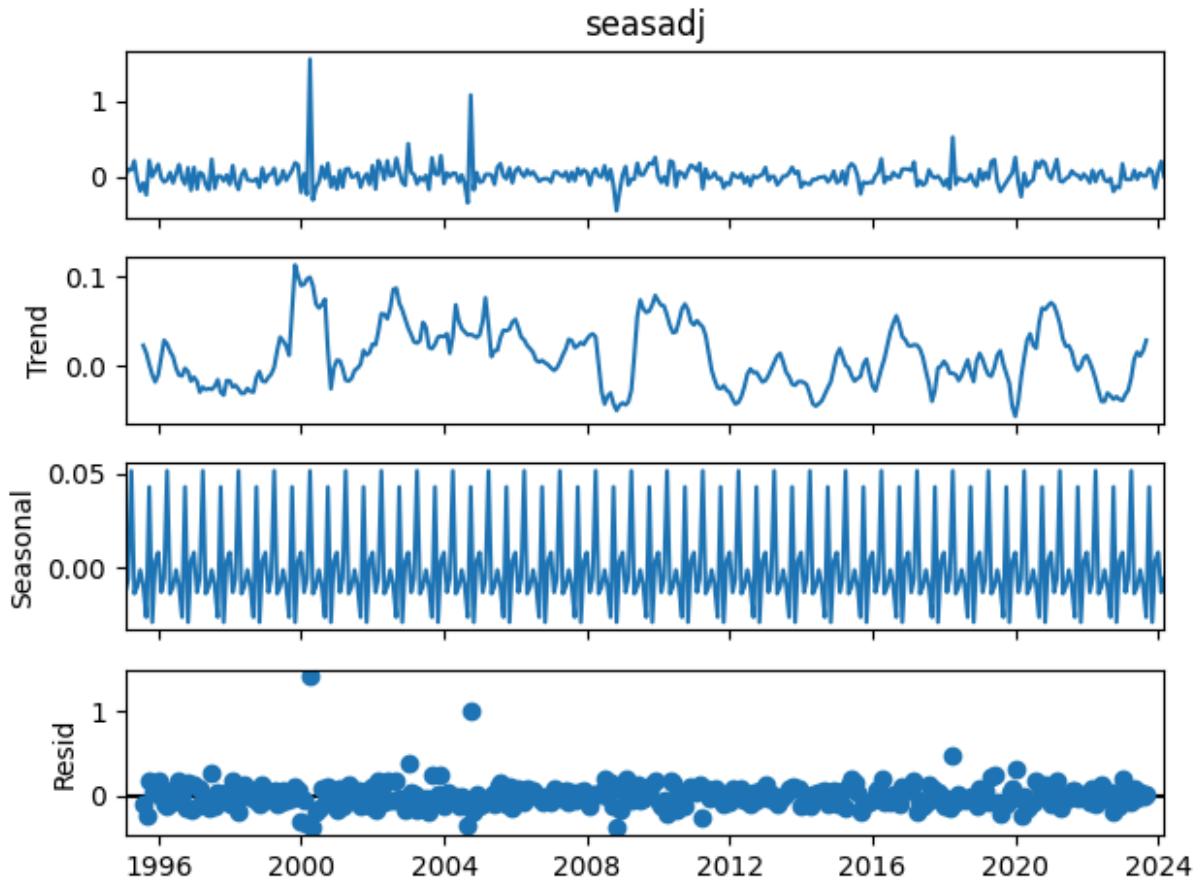
Critical Value 1%: 0.739

is\_stationary: True

Decomposing the Series...

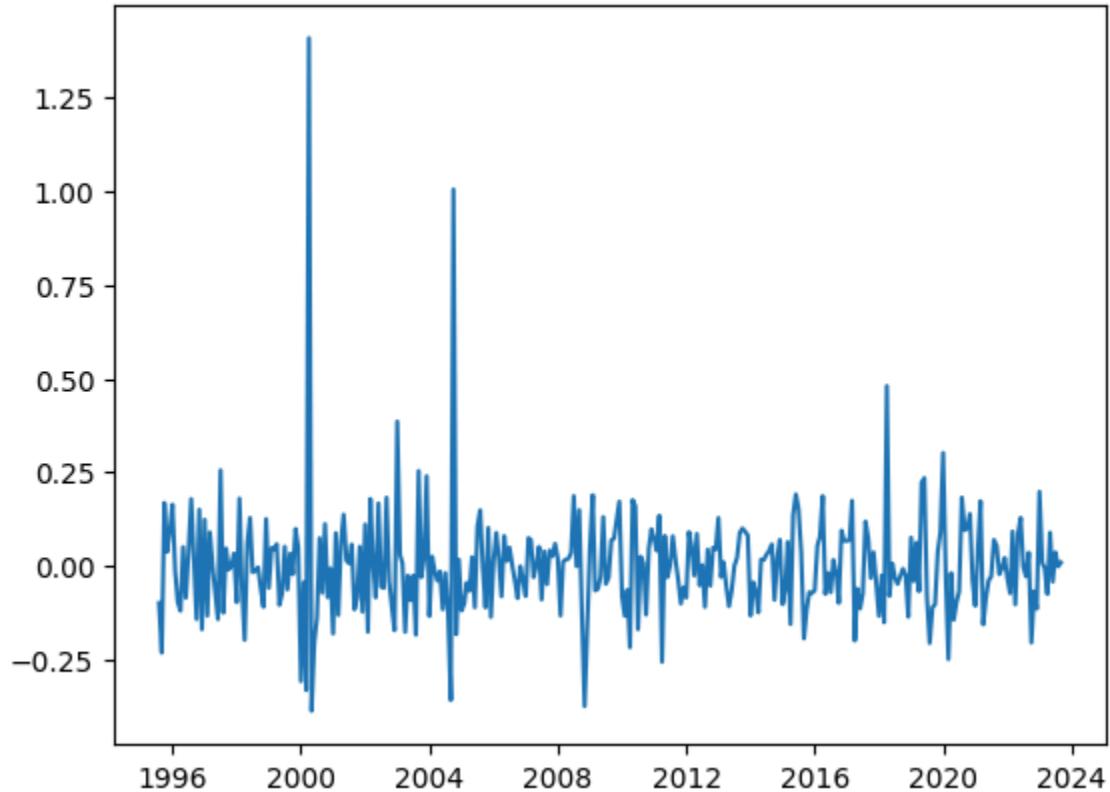
/home/wheelfredie/scripts/BoT\_Exports/helper.py:61: InterpolationWarning: The test statistic is outside of the range of p-values available in the look-up table. The actual p-value is greater than the p-value returned.

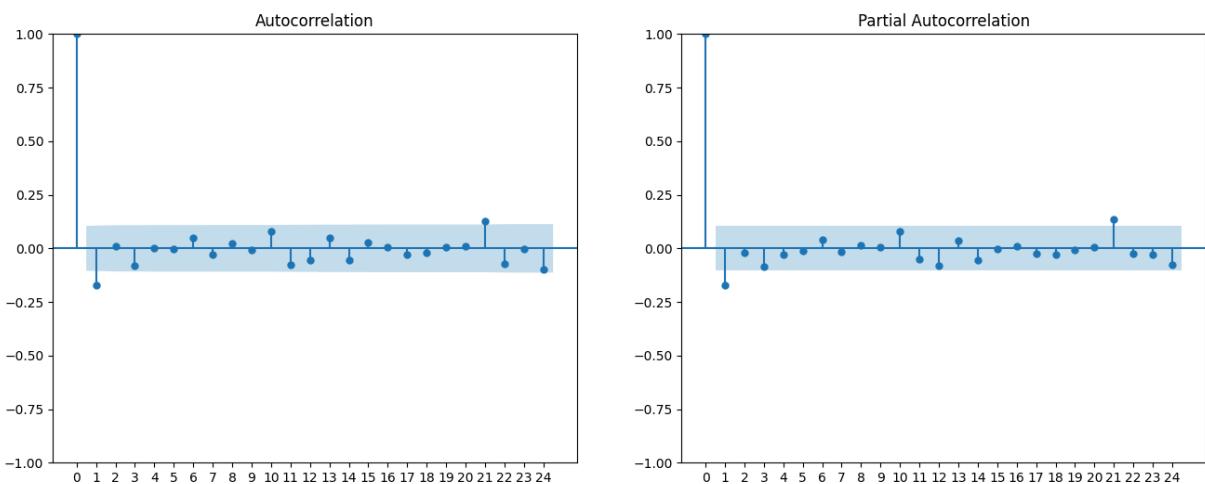
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals





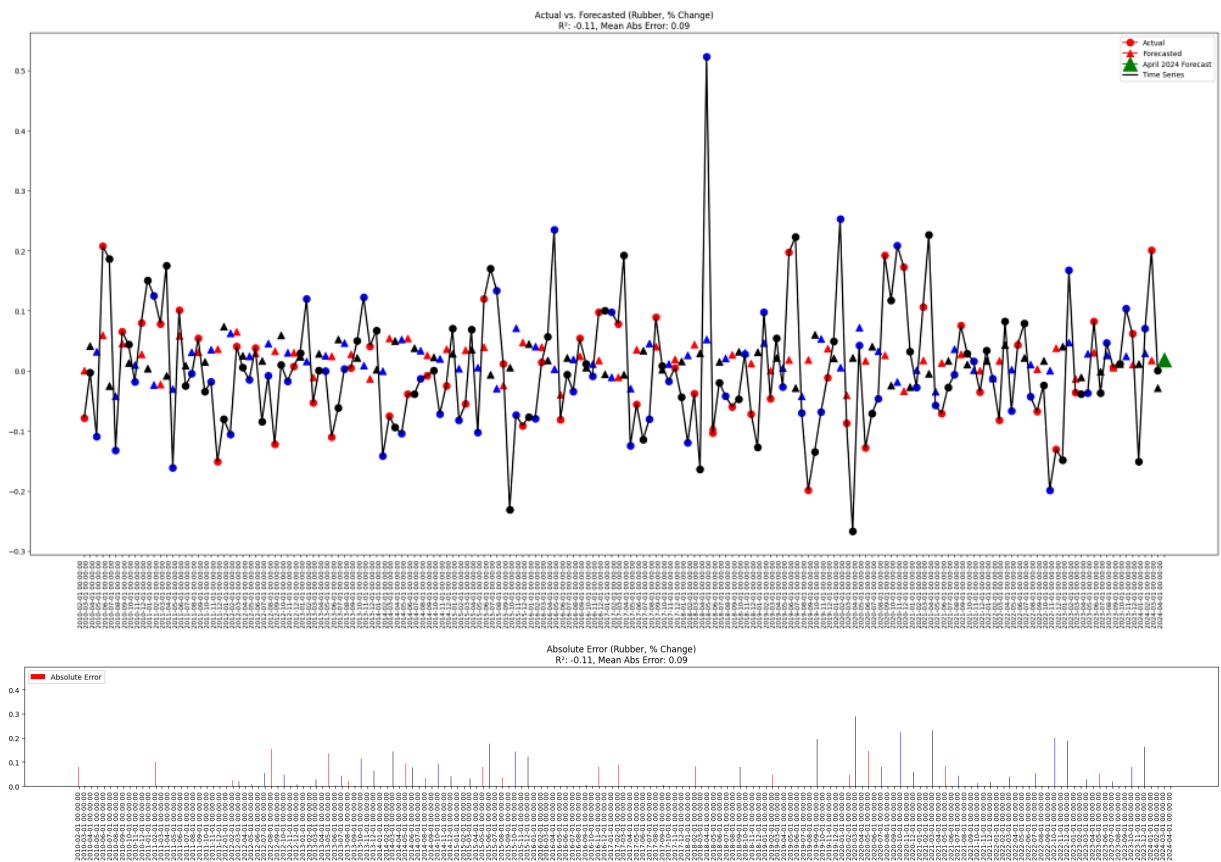
Time series assumptions are met.

'Data of Actual vs Forecasted values with error metrics:'

| Rubber     | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01 | -0.078375 | 0.000334  | -0.078709 | 0.078709  | 0.006195      | 1.004258             |
| 2010-03-01 | -0.002378 | 0.041979  | -0.044357 | 0.044357  | 0.001968      | 18.650189            |
| 2010-04-01 | -0.109102 | 0.031840  | -0.140942 | 0.140942  | 0.019865      | 1.291838             |
| 2010-05-01 | 0.207537  | 0.059429  | 0.148108  | 0.148108  | 0.021936      | 0.713648             |
| 2010-06-01 | 0.186257  | -0.024825 | 0.211081  | 0.211081  | 0.044555      | 1.133282             |
| ...        | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01 | -0.151016 | 0.011612  | -0.162629 | 0.162629  | 0.026448      | 1.076894             |
| 2024-01-01 | 0.070841  | 0.029946  | 0.040896  | 0.040896  | 0.001672      | 0.577284             |
| 2024-02-01 | 0.201136  | 0.017526  | 0.183609  | 0.183609  | 0.033712      | 0.912864             |
| 2024-03-01 | 0.000494  | -0.028519 | 0.029013  | 0.029013  | 0.000842      | 58.749997            |
| 2024-04-01 | NaN       | 0.018832  | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns

'Total Mean Absolute Error (Per Forecast): 0.08708156791004627'



## Durian

```
In [17]: df = df_export_ANALYSIS.copy()
name = df.columns[3]
display(f"Component: {name}")
```

'Component: Durian'

```
In [18]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

| class      | Durian    |
|------------|-----------|
| 1995-02-01 | 0.074074  |
| 1995-03-01 | 4.672414  |
| 1995-04-01 | 2.565350  |
| 1995-05-01 | 0.283887  |
| 1995-06-01 | -0.636786 |
| ...        | ...       |
| 2023-11-01 | -0.252418 |
| 2023-12-01 | 0.592524  |
| 2024-01-01 | -0.372840 |
| 2024-02-01 | 0.037631  |
| 2024-03-01 | 0.284384  |

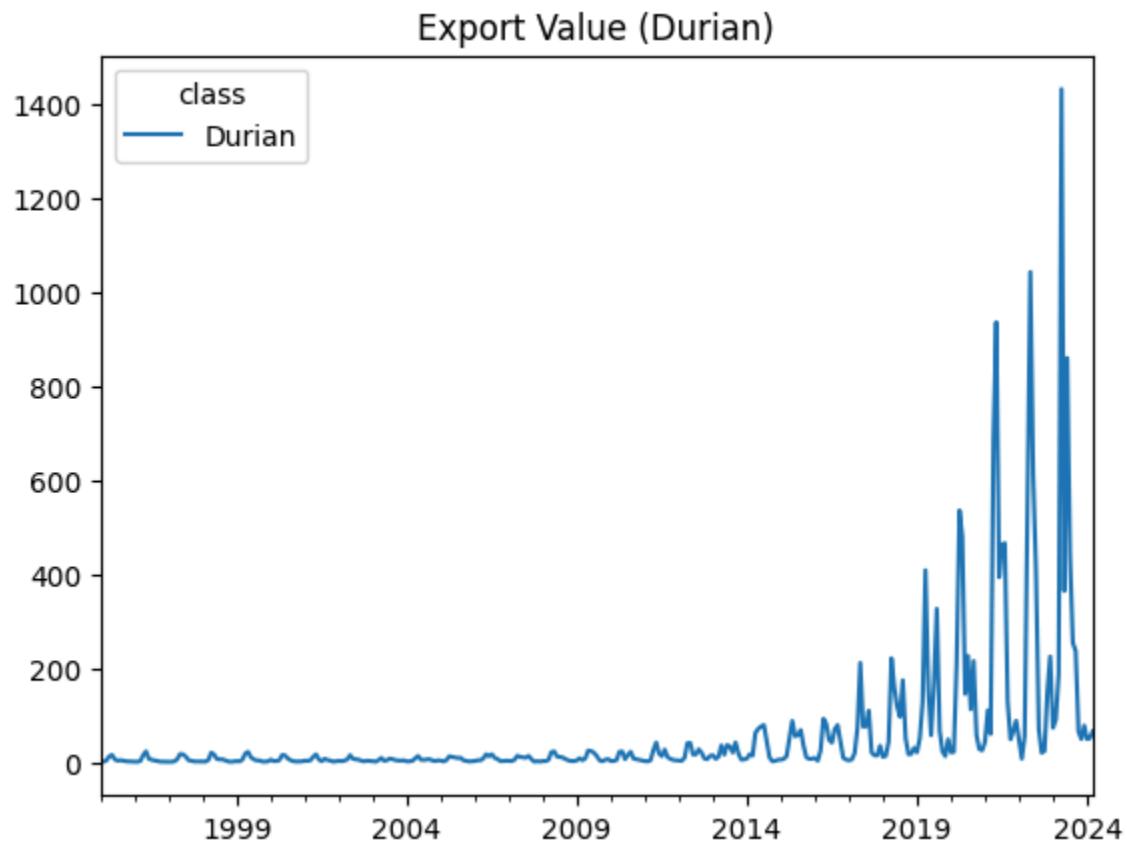
350 rows × 1 columns

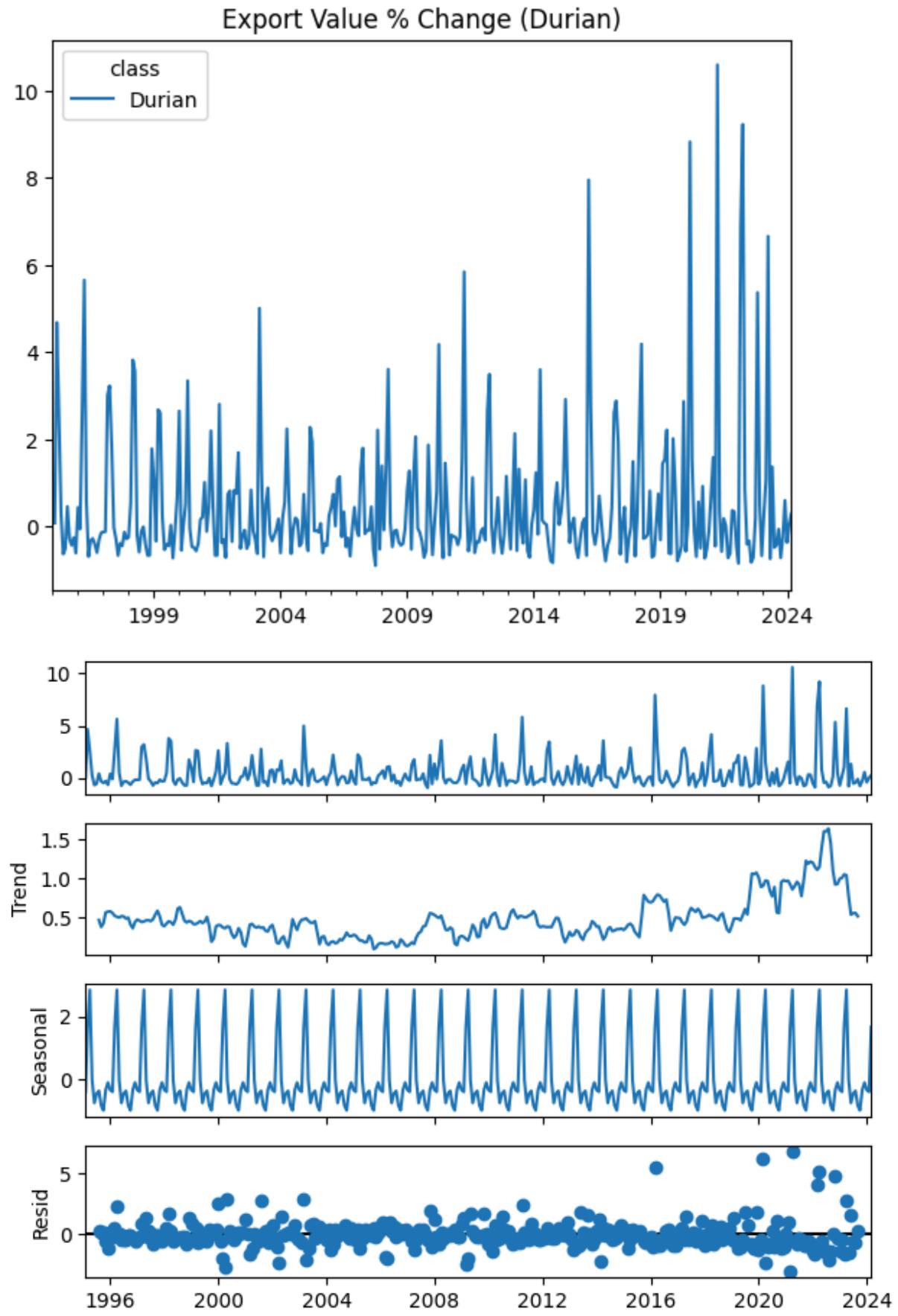
'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

```
Running ADF Test for Stationarity...
ADF Statistic: -2.666297288716745
p-value: 0.08005407384840707
Critical Value 1%: -3.4500219858626227
Critical Value 5%: -2.870206553997666
Critical Value 10%: -2.571387268879483
is_stationary: False
```

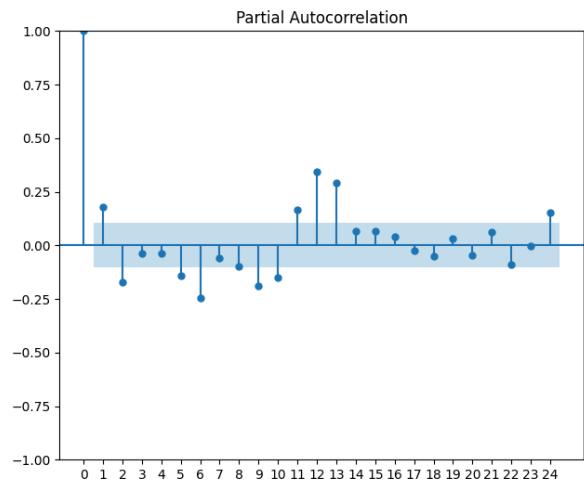
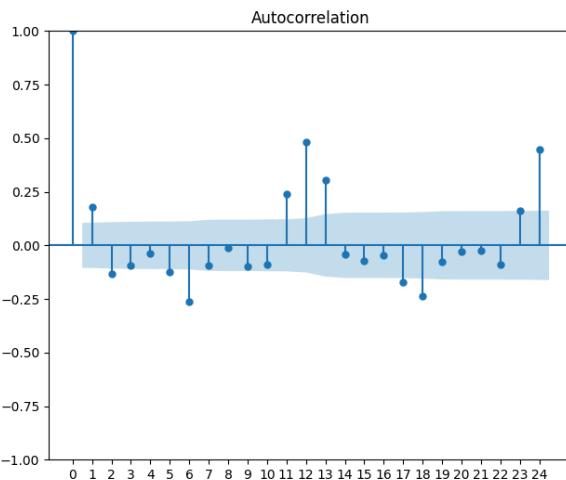
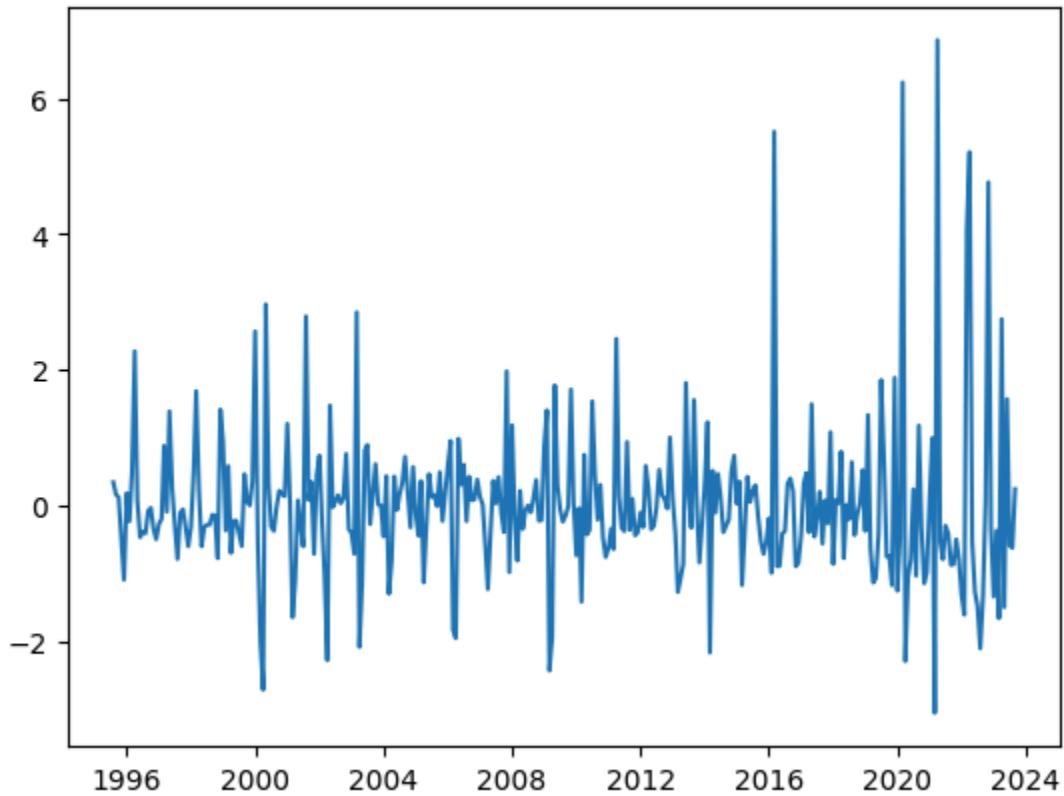
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.3833605025381945
p-value: 0.08432736959560583
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...





### Residuals



Time series assumptions are not met. Further investigation needed.

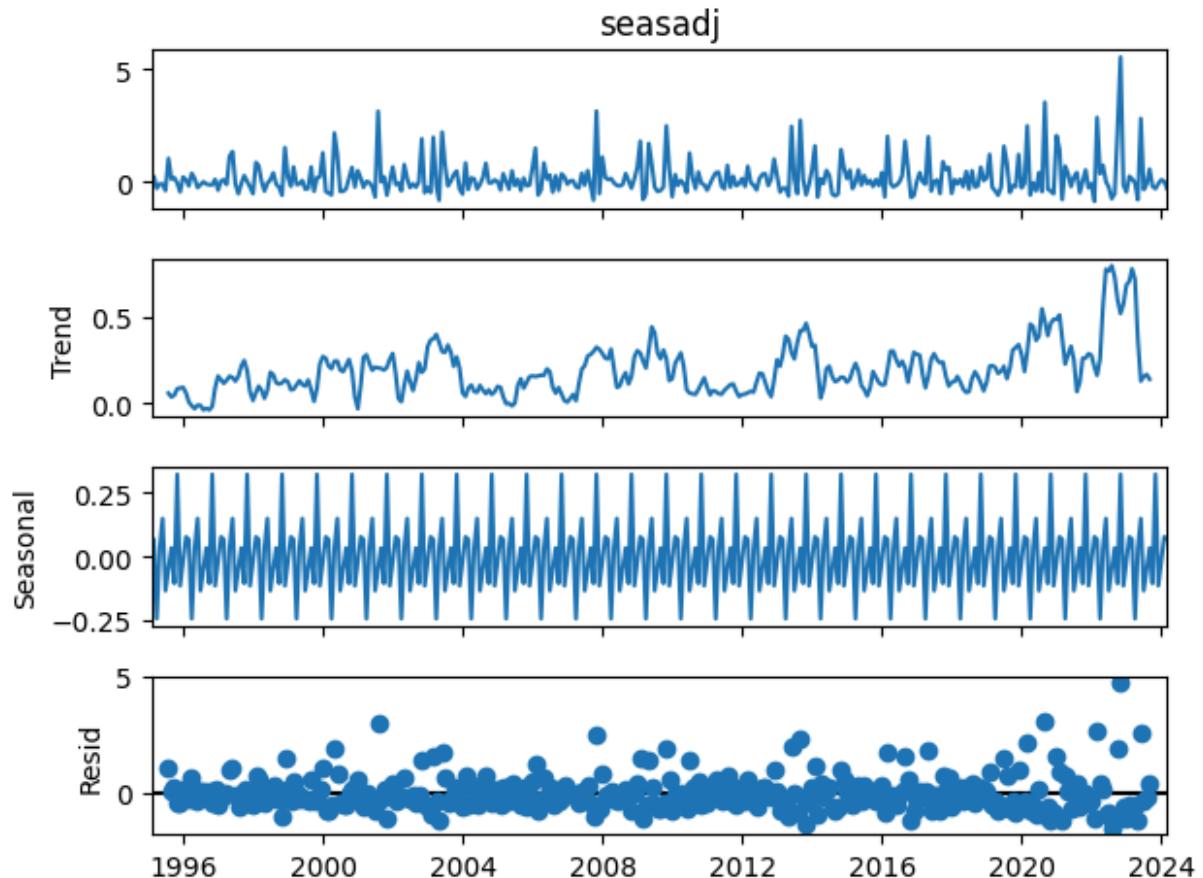
False

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

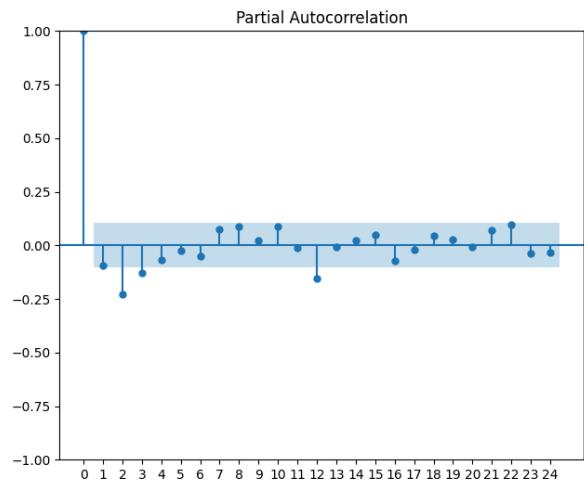
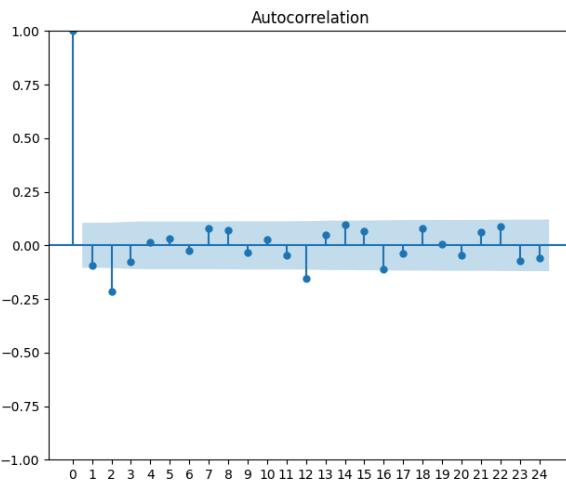
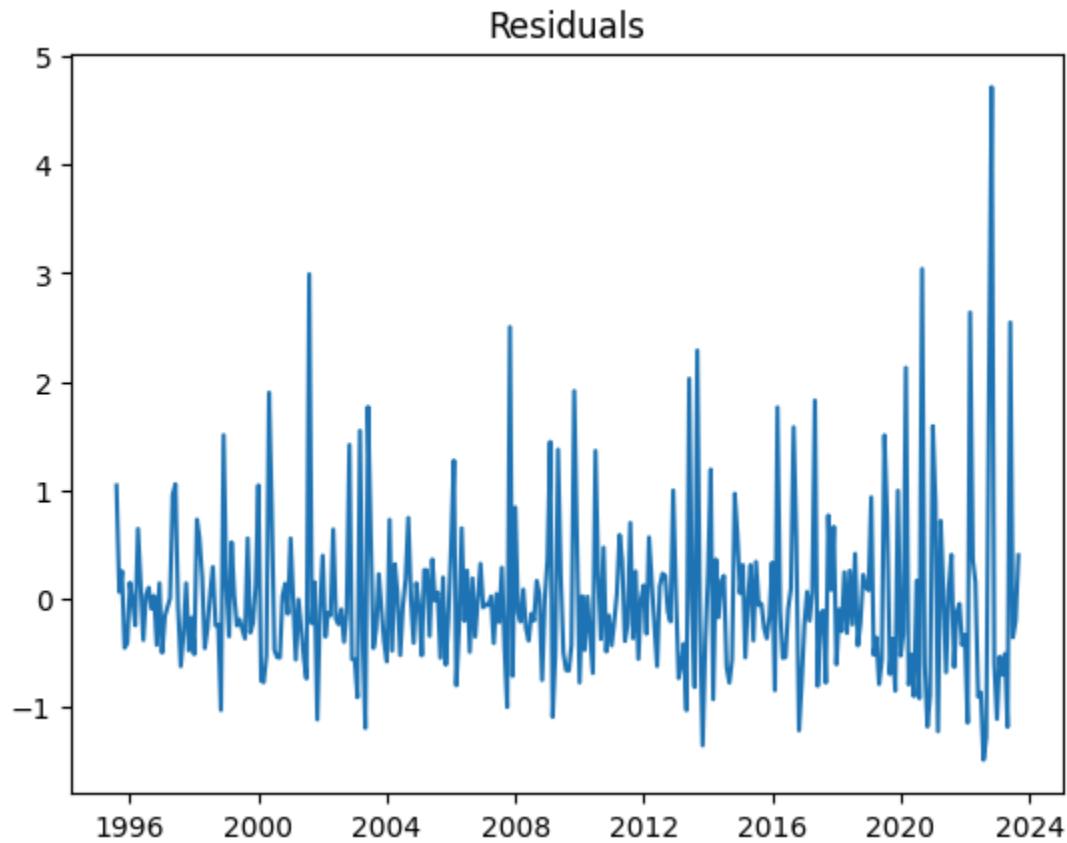
```
Running ADF Test for Stationarity...
ADF Statistic: -5.654259579766532
p-value: 9.702019584150628e-07
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.515421804117269
p-value: 0.03819328736097545
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: False
```

Decomposing the Series...



Plotting Residuals for Homoscedasticity Check...



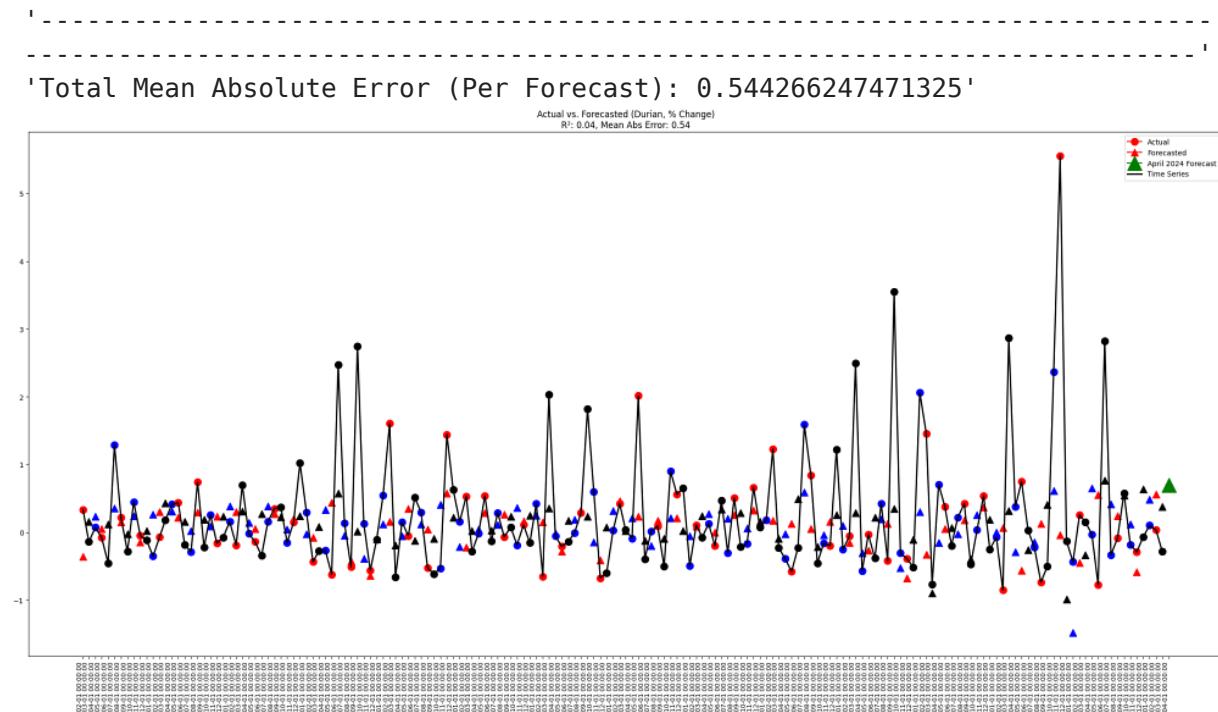
Time series assumptions are not met. Further investigation needed.

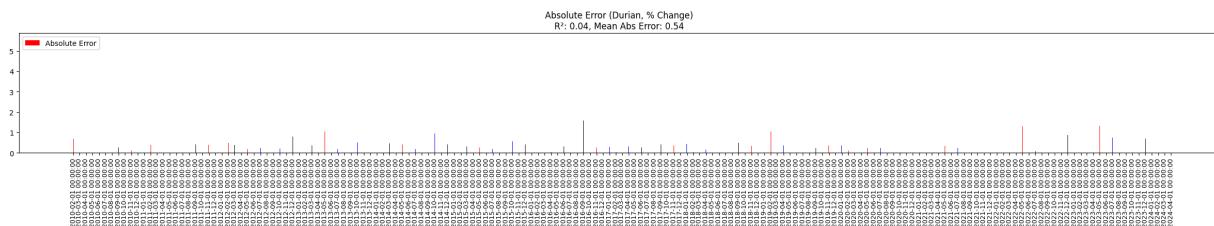
'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Durian     | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01 | 0.332411  | -0.361639 | 0.694050  | 0.694050  | 0.481706      | 2.087926             |
| 2010-03-01 | -0.143789 | 0.157522  | -0.301311 | 0.301311  | 0.090788      | 2.095505             |
| 2010-04-01 | 0.069102  | 0.229410  | -0.160309 | 0.160309  | 0.025699      | 2.319892             |
| 2010-05-01 | -0.078682 | 0.046134  | -0.124816 | 0.124816  | 0.015579      | 1.586333             |
| 2010-06-01 | -0.458265 | 0.116006  | -0.574271 | 0.574271  | 0.329787      | 1.253141             |
| ...        | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01 | -0.072090 | 0.630673  | -0.702763 | 0.702763  | 0.493875      | 9.748452             |
| 2024-01-01 | 0.099303  | 0.479382  | -0.380079 | 0.380079  | 0.144460      | 3.827463             |
| 2024-02-01 | 0.031235  | 0.560893  | -0.529658 | 0.529658  | 0.280538      | 16.957061            |
| 2024-03-01 | -0.284318 | 0.379214  | -0.663532 | 0.663532  | 0.440275      | 2.333767             |
| 2024-04-01 | NaN       | 0.690984  | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns





In this event, even with adjustment, SARIMA Time series analysis does not do well, which can be seen due to generally tight forecast except for huge jumps in actual observed values. If the jumps are due to forecast, we can regulate it via percentile algorithms. However as the shock is external from the historical timeseries, further factors like demand suvey, durian supply ssurvey and news sentiment might serve a better option. I also found that true simpler smoothing prediction models does not out perform SARIMA (which i will check below)

```
In [19]: df_smoothing = df_accuracy[["actual", "forecast"]].copy(deep=True)
df_smoothing["forecast"] = df_smoothing["actual"].shift(1).rolling(window=24)
df_smoothing = calculate_accuracy_metrics(df_smoothing["actual"], df_smoothing["forecast"])
display(df_smoothing)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_smoothing["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_smoothing, f"Rolling Average{name}")
```

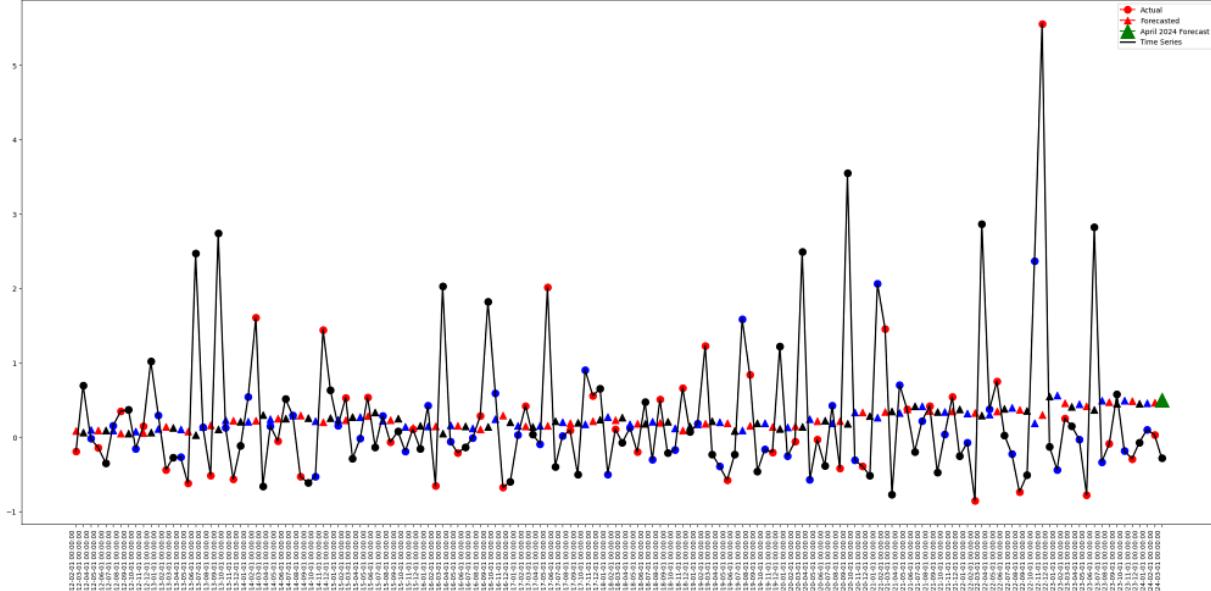
|            | actual    | forecast | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|----------|-----------|-----------|---------------|----------------------|
| 2012-02-01 | -0.193172 | 0.084174 | -0.277347 | 0.277347  | 0.076921      | 1.435748             |
| 2012-03-01 | 0.693883  | 0.062275 | 0.631608  | 0.631608  | 0.398929      | 0.910251             |
| 2012-04-01 | -0.021601 | 0.097178 | -0.118779 | 0.118779  | 0.014108      | 5.498829             |
| 2012-05-01 | -0.141372 | 0.093399 | -0.234771 | 0.234771  | 0.055117      | 1.660658             |
| 2012-06-01 | -0.348385 | 0.090787 | -0.439172 | 0.439172  | 0.192872      | 1.260593             |
| ...        | ...       | ...      | ...       | ...       | ...           | ...                  |
| 2023-11-01 | -0.292977 | 0.484634 | -0.777611 | 0.777611  | 0.604679      | 2.654173             |
| 2023-12-01 | -0.072090 | 0.449995 | -0.522085 | 0.522085  | 0.272573      | 7.242162             |
| 2024-01-01 | 0.099303  | 0.457572 | -0.358269 | 0.358269  | 0.128356      | 3.607827             |
| 2024-02-01 | 0.031235  | 0.464780 | -0.433545 | 0.433545  | 0.187961      | 13.879992            |
| 2024-03-01 | -0.284318 | 0.501653 | -0.785971 | 0.785971  | 0.617750      | 2.764407             |

146 rows × 6 columns

'Total Mean Absolute Error (Per Forecast): 0.6416991421054048'

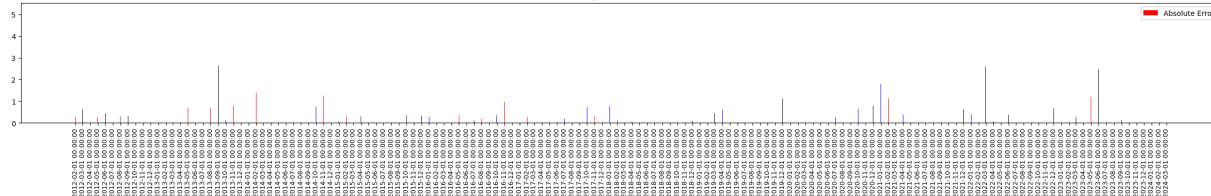
Actual vs. Forecasted (Rolling Average Durian, % Change)

R²: -0.05, Mean Abs Error: 0.64



Absolute Error (Rolling AverageDurian, % Change)

 $R^2: -0.05$ , Mean Abs Error: 0.64



```
In [20]: import pandas as pd
from statsmodels.tsa.holtwinters import SimpleExpSmoothing

model = SimpleExpSmoothing(df_accuracy['actual'])
fit_model = model.fit(smoothing_level=0.2, optimized=False)
df_accuracy['forecast'] = fit_model.fittedvalues

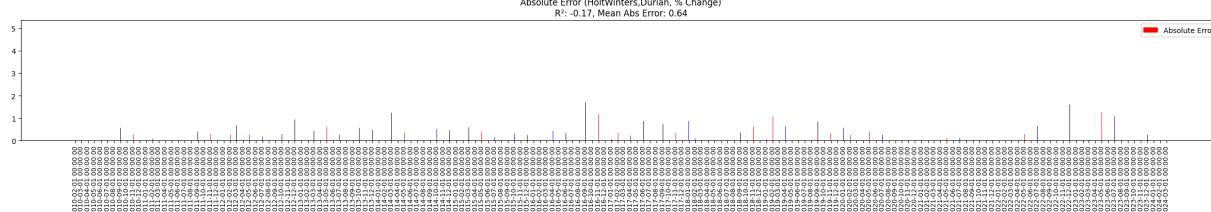
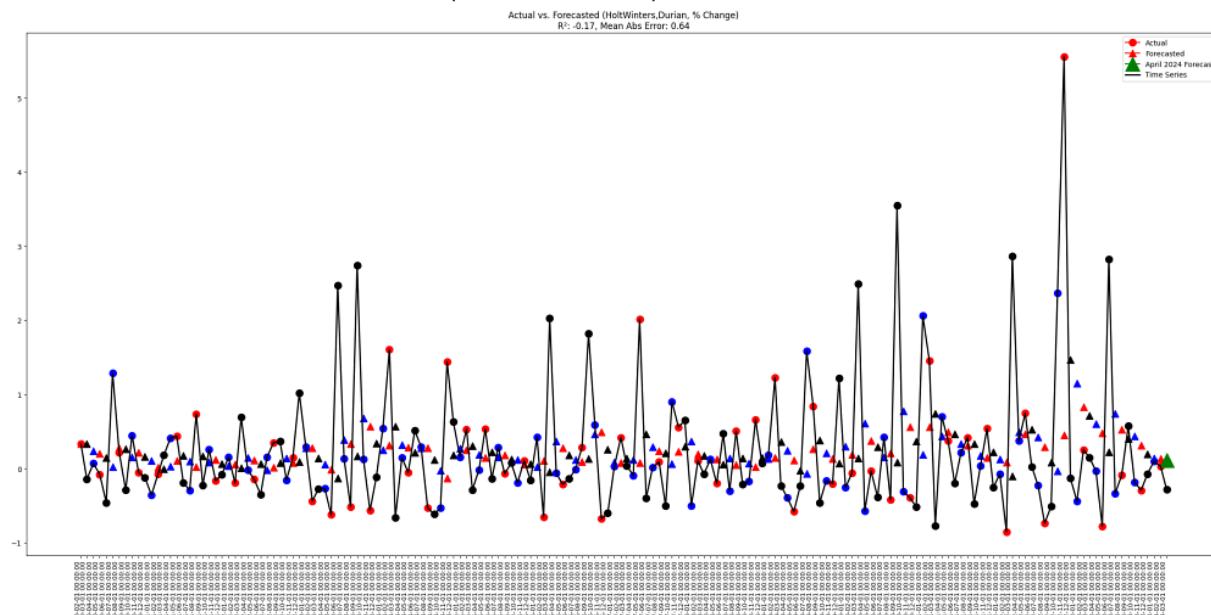
# Calculate accuracy metrics
df_smoothing = calculate_accuracy_metrics(df_accuracy['actual'], df_accuracy)

# Plot Actual vs Forecasted values with error metrics
average_abs_error = df_smoothing['abs_error'].mean()
print(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_smoothing, f"HoltWinters,{name}")
```

Total Mean Absolute Error (Per Forecast): 0.6356185380573713

```
/home/wheelfredie/.local/share/virtualenvs/BoT_Exports-70xff3EB/lib/python3.10/site-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
    self._init_dates(dates, freq)
```

'Total Mean Absolute Error (Per Forecast): 0.6356185380573713'



Durian seems to lack AR properties and have poor predictability across models, with SARIMA still outperforming

## Other\_Fruits

```
In [21]: df = df_export_ANALYSIS.copy()
name = df.columns[4]
display(f"Component: {name}")

'Component: Other_Fruits'

In [22]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

'Timeseries of component in % form:'
```

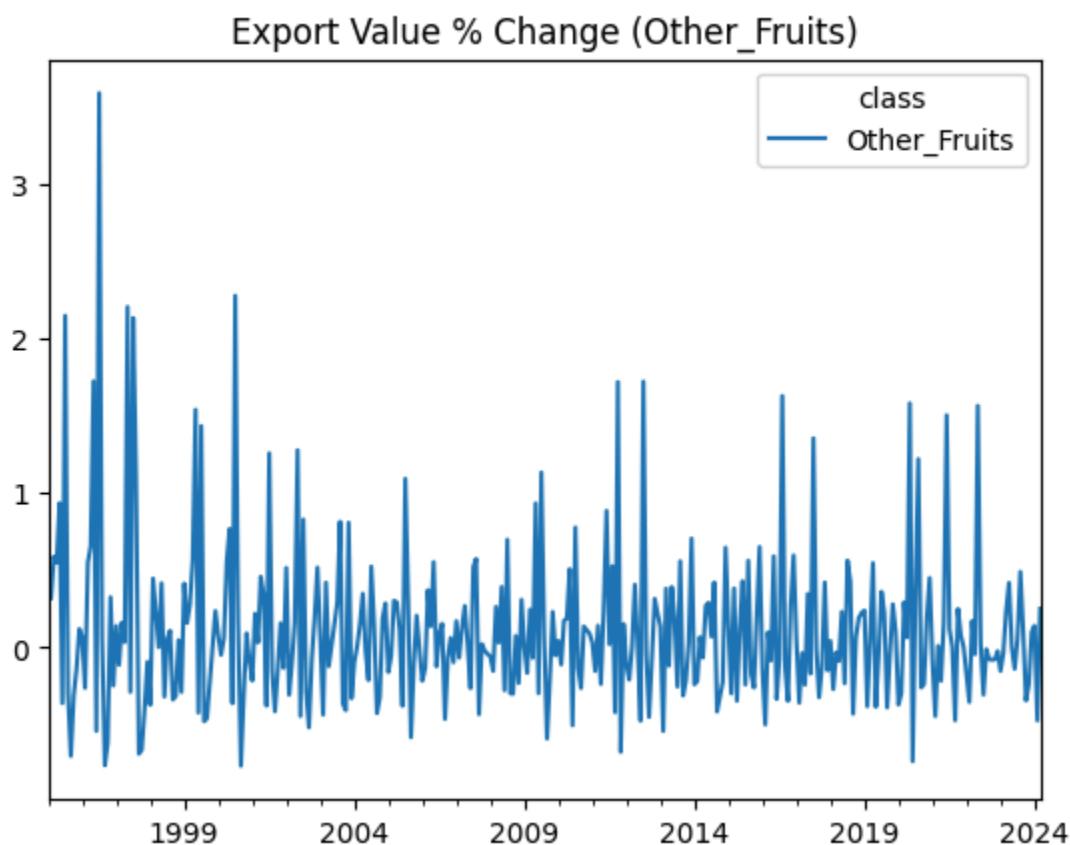
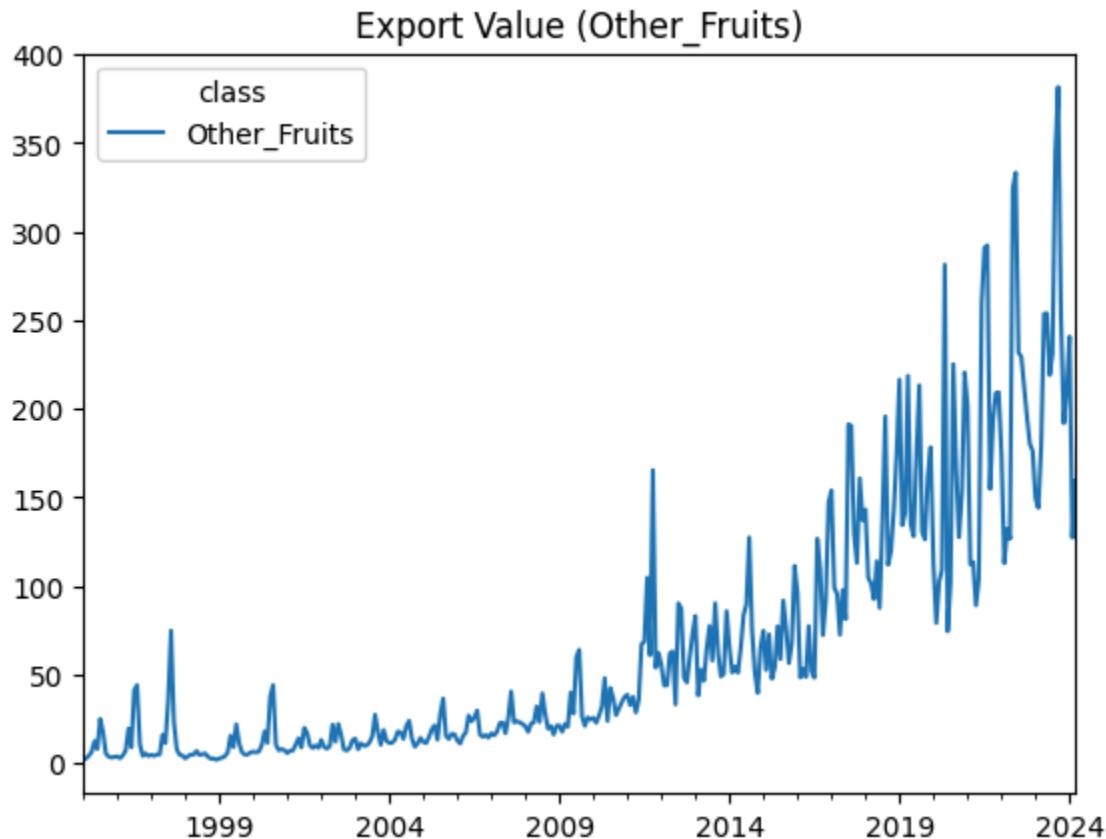
| class      | Other_Fruits |
|------------|--------------|
| 1995-02-01 | 0.319797     |
| 1995-03-01 | 0.588462     |
| 1995-04-01 | 0.549637     |
| 1995-05-01 | 0.934375     |
| 1995-06-01 | -0.359451    |
| ...        | ...          |
| 2023-11-01 | -0.235817    |
| 2023-12-01 | 0.099786     |
| 2024-01-01 | 0.141076     |
| 2024-02-01 | -0.471189    |
| 2024-03-01 | 0.251709     |

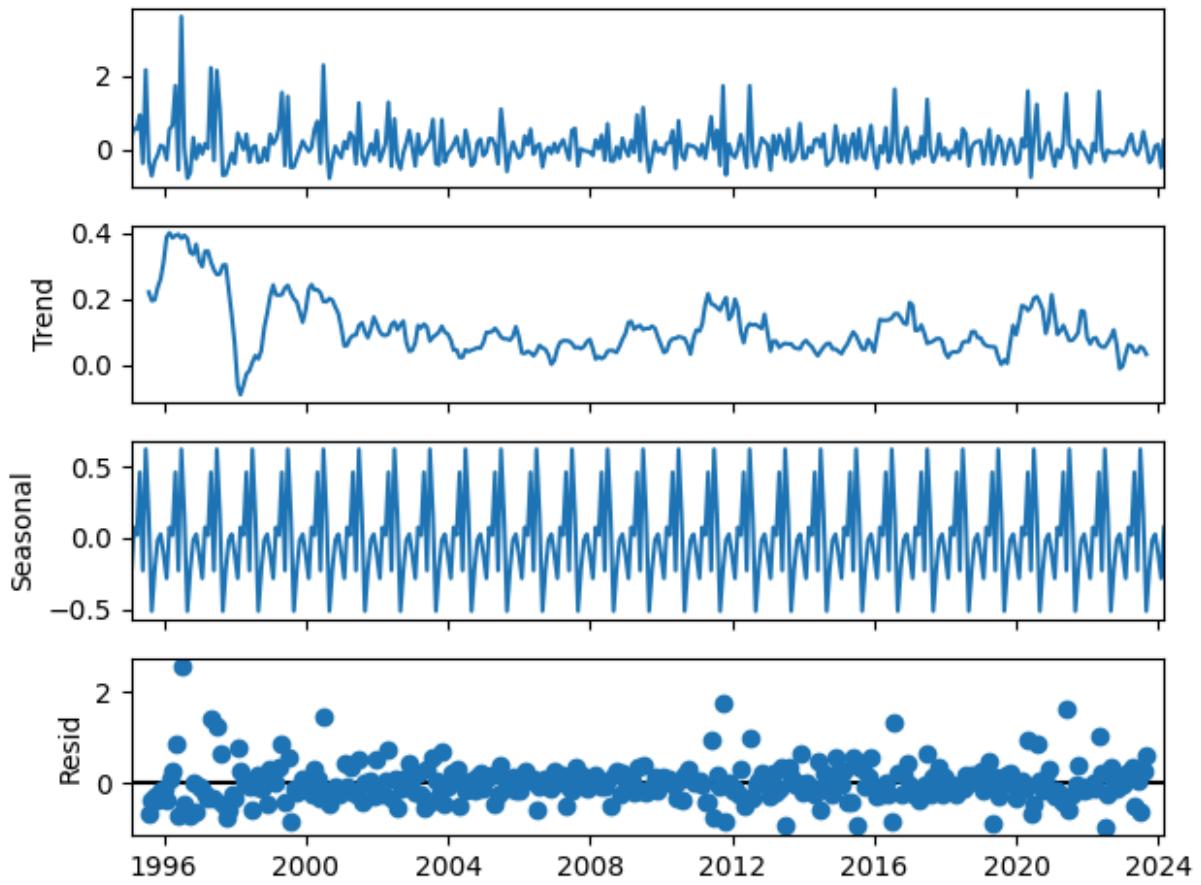
350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -3.8686106871899493
p-value: 0.002278391766330927
Critical Value 1%: -3.4499043309021955
Critical Value 5%: -2.870154885338925
Critical Value 10%: -2.5713597190254385
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.3667018257796163
p-value: 0.09150783371568263
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

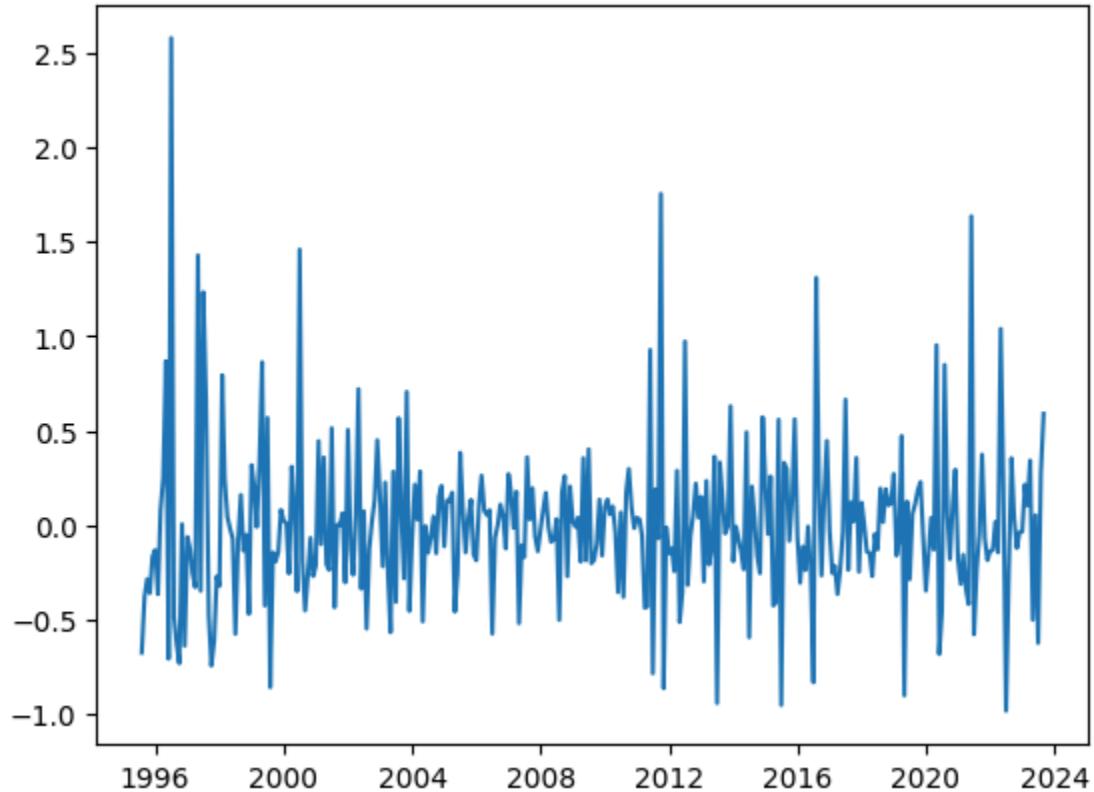
Decomposing the Series...

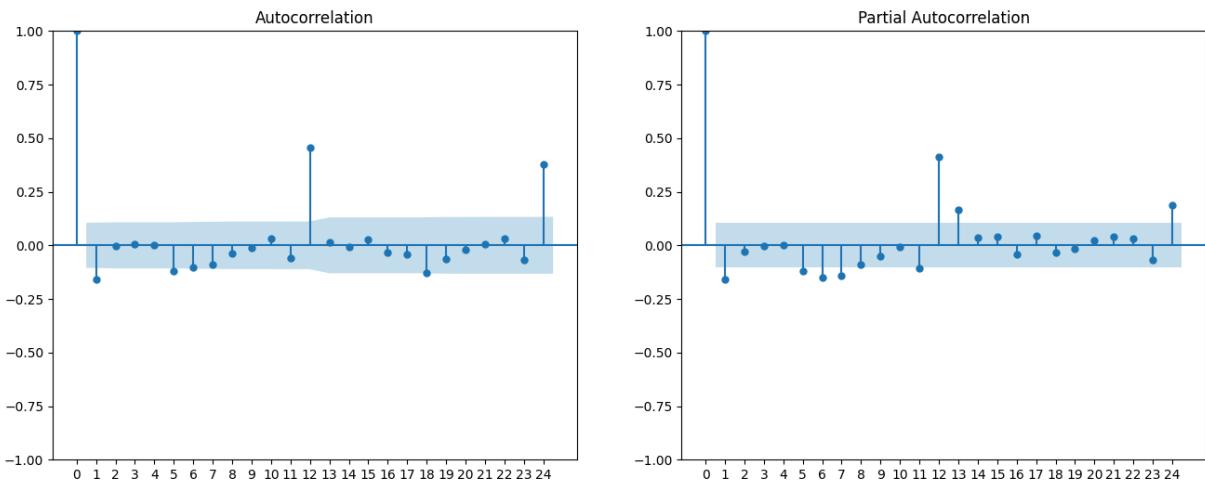




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

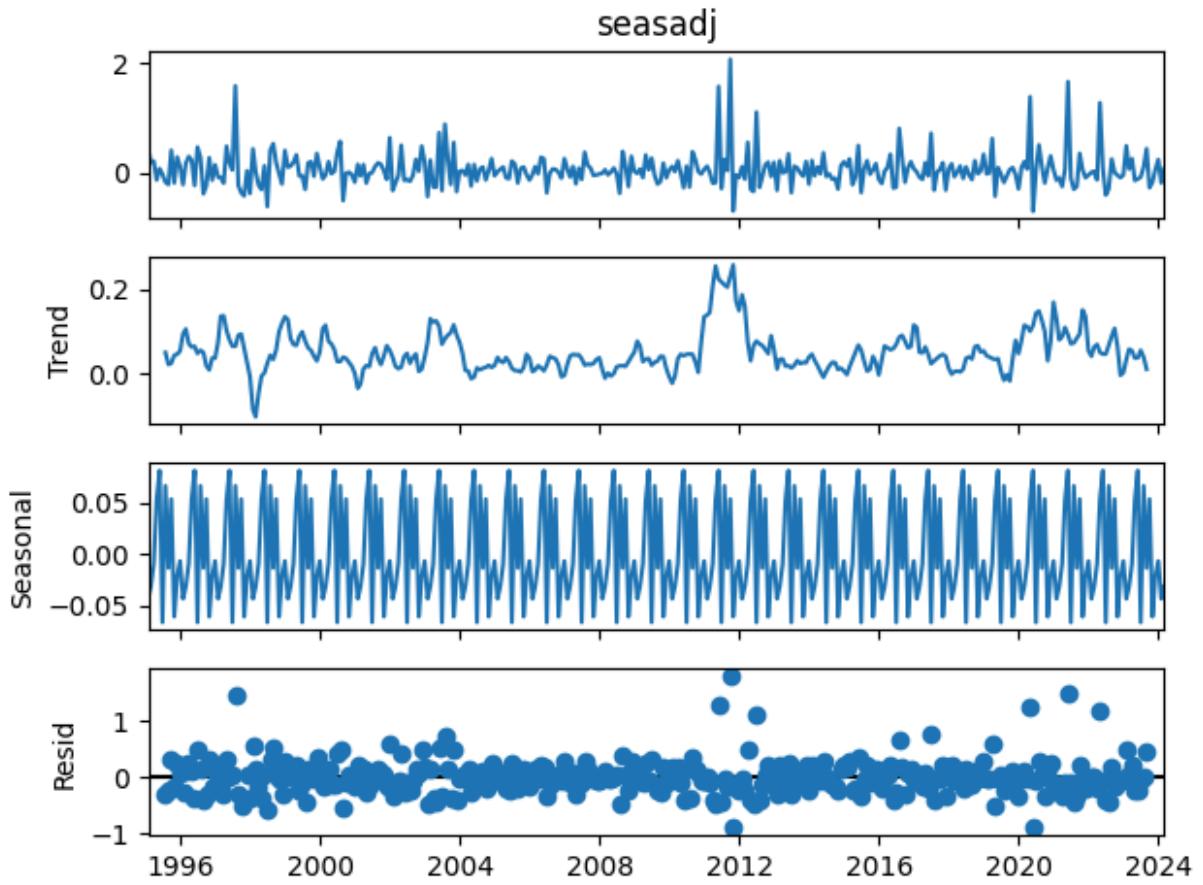
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted): '
Running ADF Test for Stationarity...
ADF Statistic: -5.944683573091165
p-value: 2.2185558340651744e-07
Critical Value 1%: -3.4499043309021955
Critical Value 5%: -2.870154885338925
Critical Value 10%: -2.5713597190254385
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.050524861457597346
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

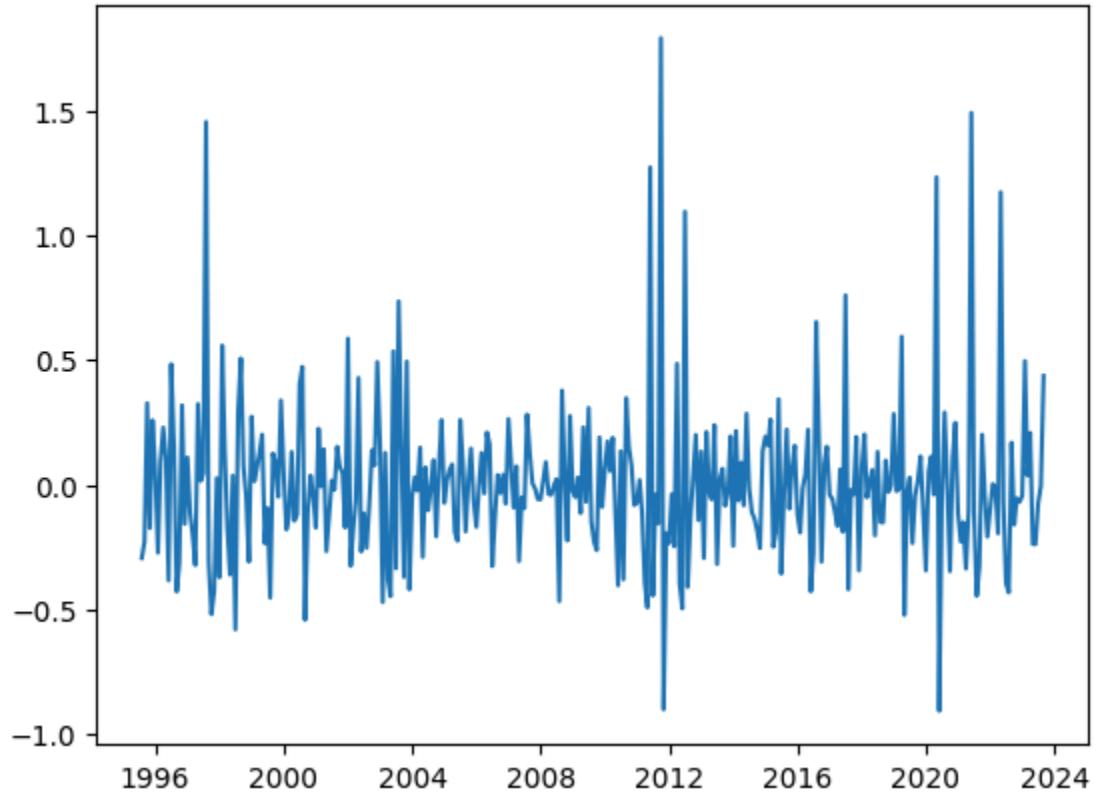
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

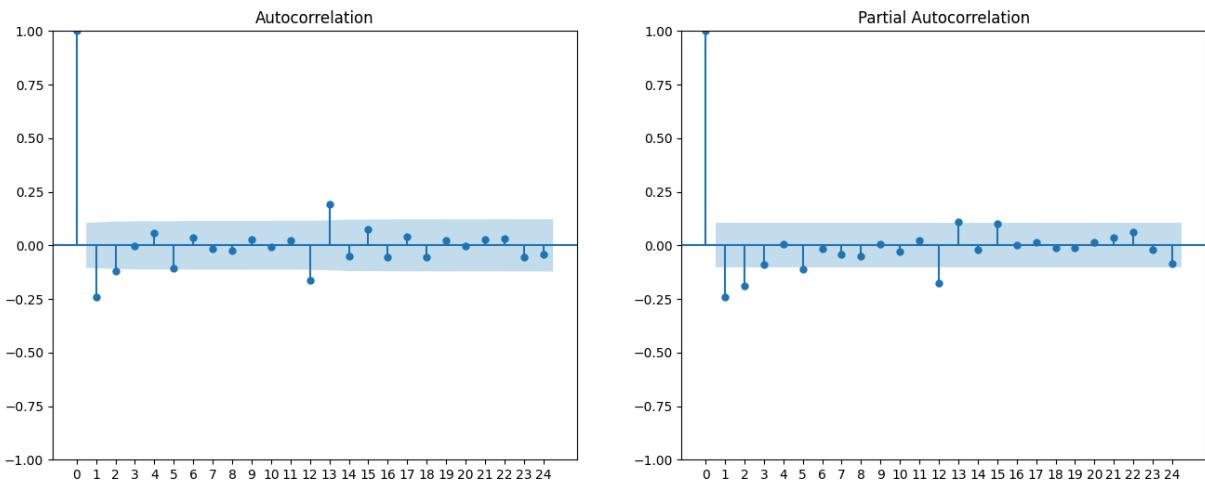
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

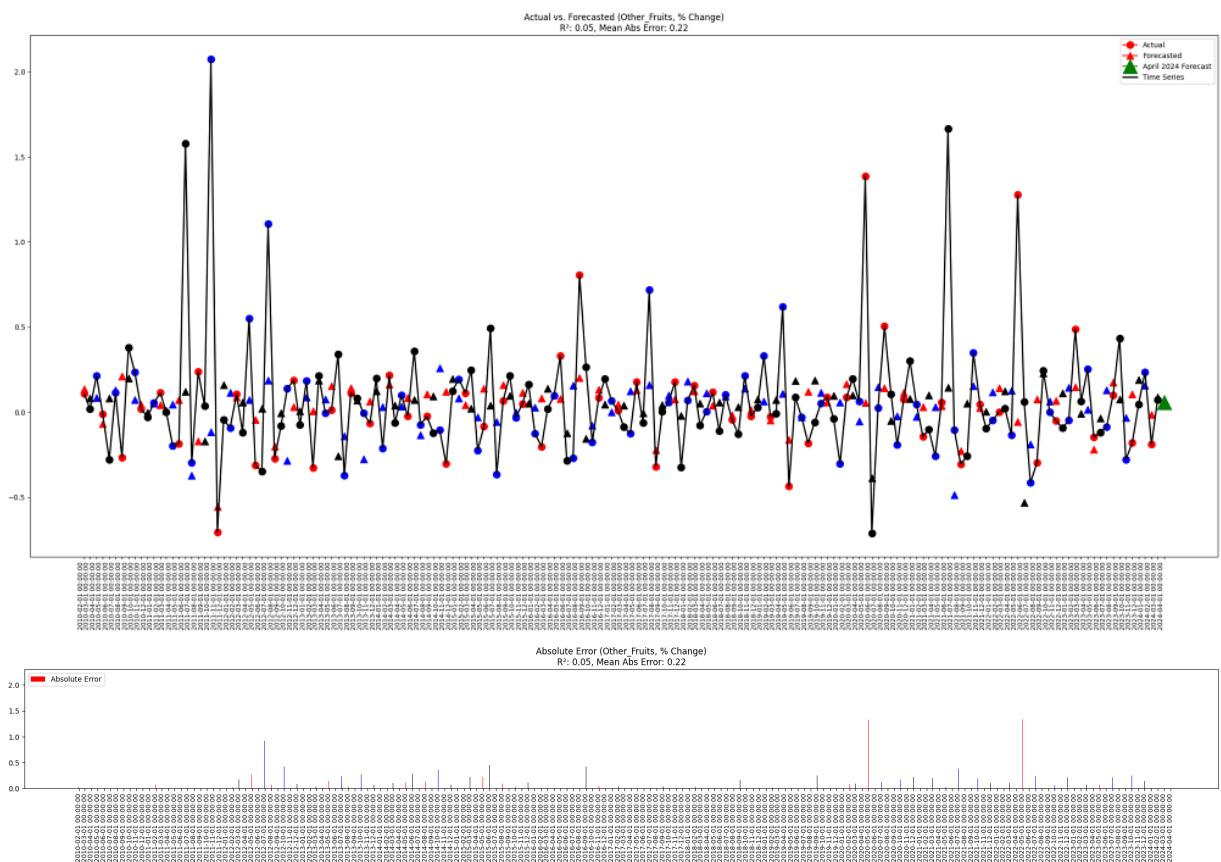
'Data of Actual vs Forecasted values with error metrics:'

| Other_Fruits | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_err |
|--------------|-----------|-----------|-----------|-----------|---------------|--------------------|
| 2010-02-01   | 0.108537  | 0.135683  | -0.027146 | 0.027146  | 0.000737      | 0.25010            |
| 2010-03-01   | 0.017583  | 0.082404  | -0.064821 | 0.064821  | 0.004202      | 3.68653            |
| 2010-04-01   | 0.214976  | 0.083721  | 0.131255  | 0.131255  | 0.017228      | 0.61055            |
| 2010-05-01   | -0.011749 | -0.067711 | 0.055963  | 0.055963  | 0.003132      | 4.76339            |
| 2010-06-01   | -0.278874 | 0.080185  | -0.359059 | 0.359059  | 0.128923      | 1.28753            |
| ...          | ...       | ...       | ...       | ...       | ...           | ...                |
| 2023-12-01   | 0.045053  | 0.190074  | -0.145021 | 0.145021  | 0.021031      | 3.21886            |
| 2024-01-01   | 0.235240  | 0.155411  | 0.079829  | 0.079829  | 0.006373      | 0.33935            |
| 2024-02-01   | -0.190265 | -0.014585 | -0.175679 | 0.175679  | 0.030863      | 0.92334            |
| 2024-03-01   | 0.071442  | 0.087865  | -0.016423 | 0.016423  | 0.000270      | 0.22987            |
| 2024-04-01   | NaN       | 0.056845  | NaN       | NaN       | NaN           | NaN                |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.2153698095069689'



## Horticultural\_products,\_n.i.e.

```
In [23]: df = df_export_ANALYSIS.copy()
name = df.columns[5]
display(f"Component: {name}")

'Component: Horticultural_products,_n.i.e.'
```

```
In [24]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value.pkl', 'rb') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
display(check_time_series_assumptions(adj_ts, freq=12))
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

class Horticultural\_products\_n.i.e.

|            |           |
|------------|-----------|
| 1995-02-01 | -0.102428 |
| 1995-03-01 | 0.264265  |
| 1995-04-01 | -0.214243 |
| 1995-05-01 | 0.034635  |
| 1995-06-01 | -0.273945 |
| ...        | ...       |
| 2023-11-01 | -0.160582 |
| 2023-12-01 | -0.188916 |
| 2024-01-01 | 0.263406  |
| 2024-02-01 | 0.036381  |
| 2024-03-01 | 0.526033  |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

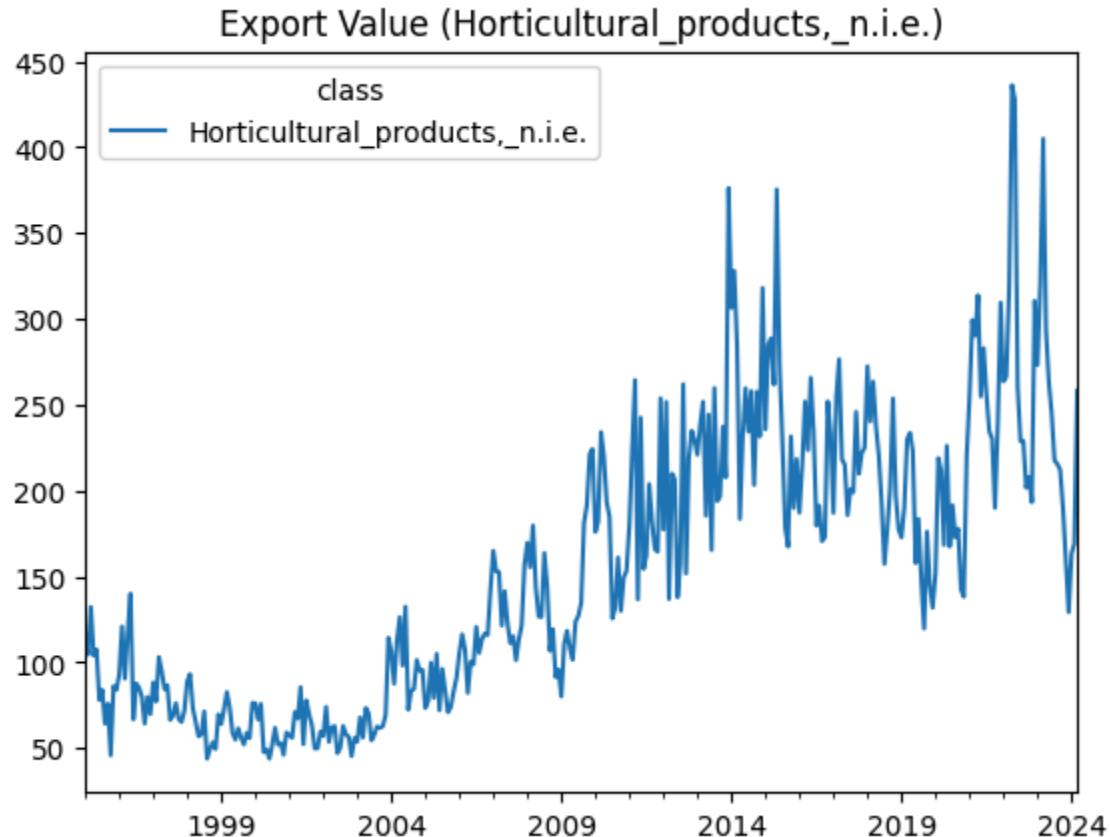
```
Running ADF Test for Stationarity...
ADF Statistic: -6.594816686859109
p-value: 6.968154769514926e-09
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

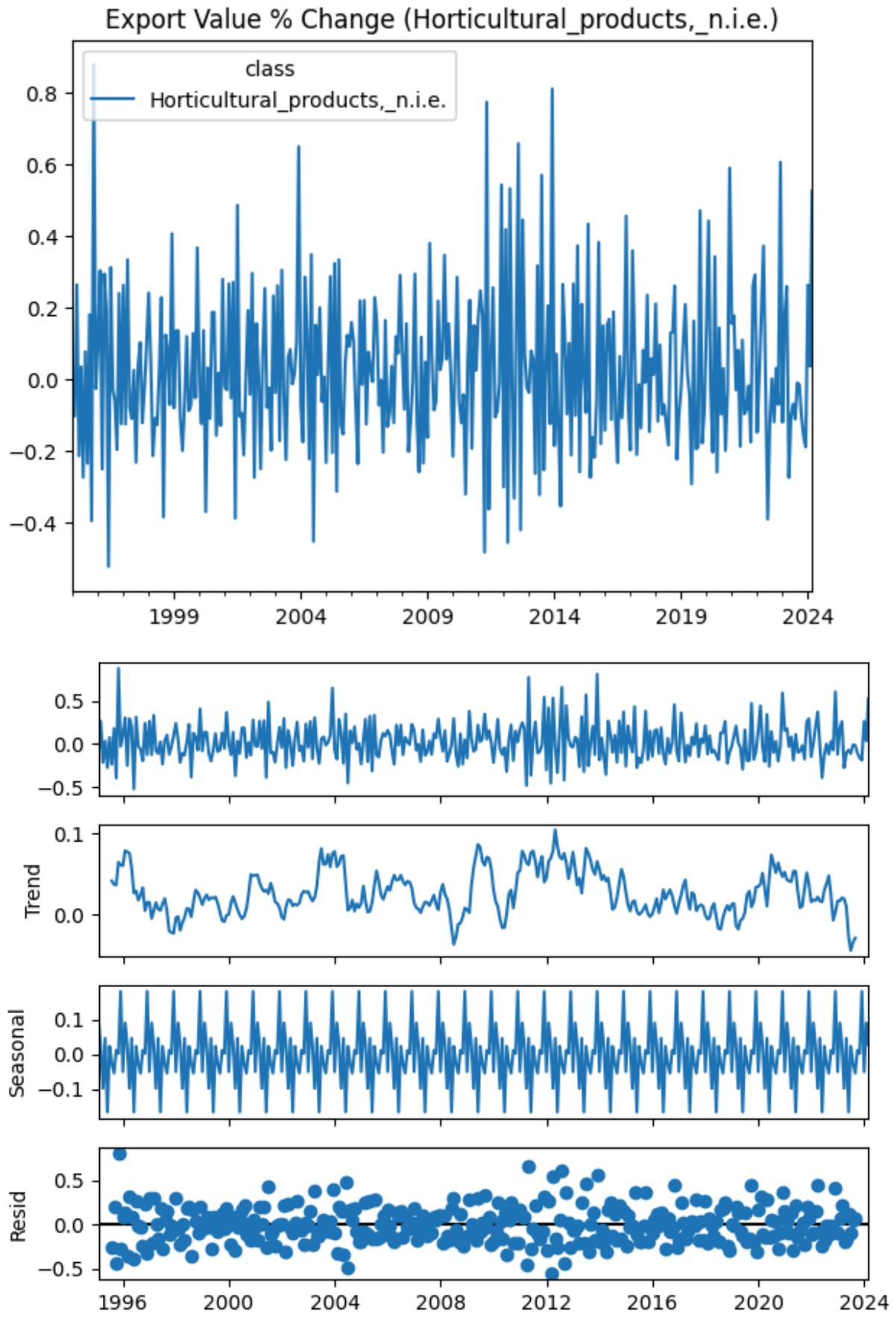
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.08675134072371464
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

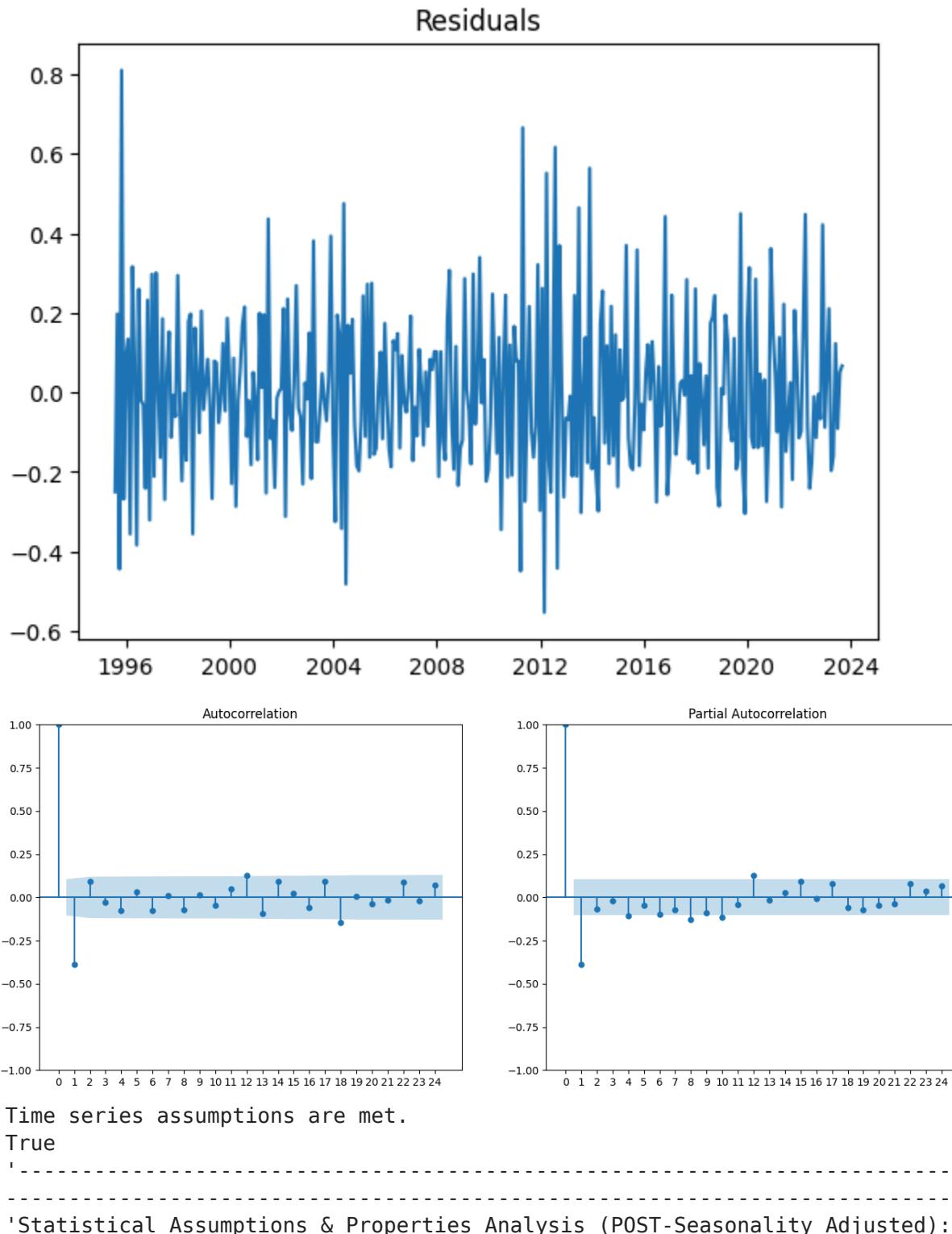
Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```







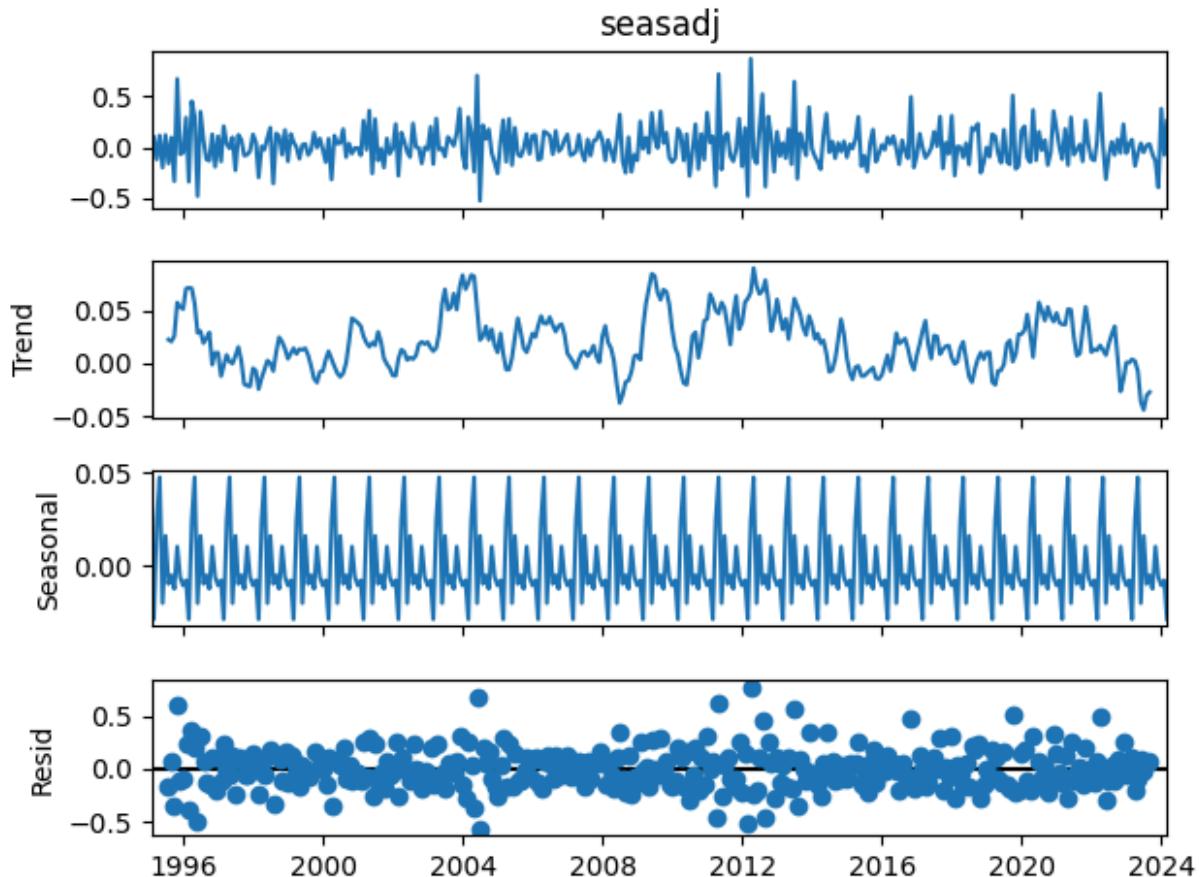
Running ADF Test for Stationarity...  
 ADF Statistic: -6.161736964862462  
 p-value: 7.146728327311864e-08  
 Critical Value 1%: -3.4499043309021955  
 Critical Value 5%: -2.870154885338925  
 Critical Value 10%: -2.5713597190254385  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.08945383324051917  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

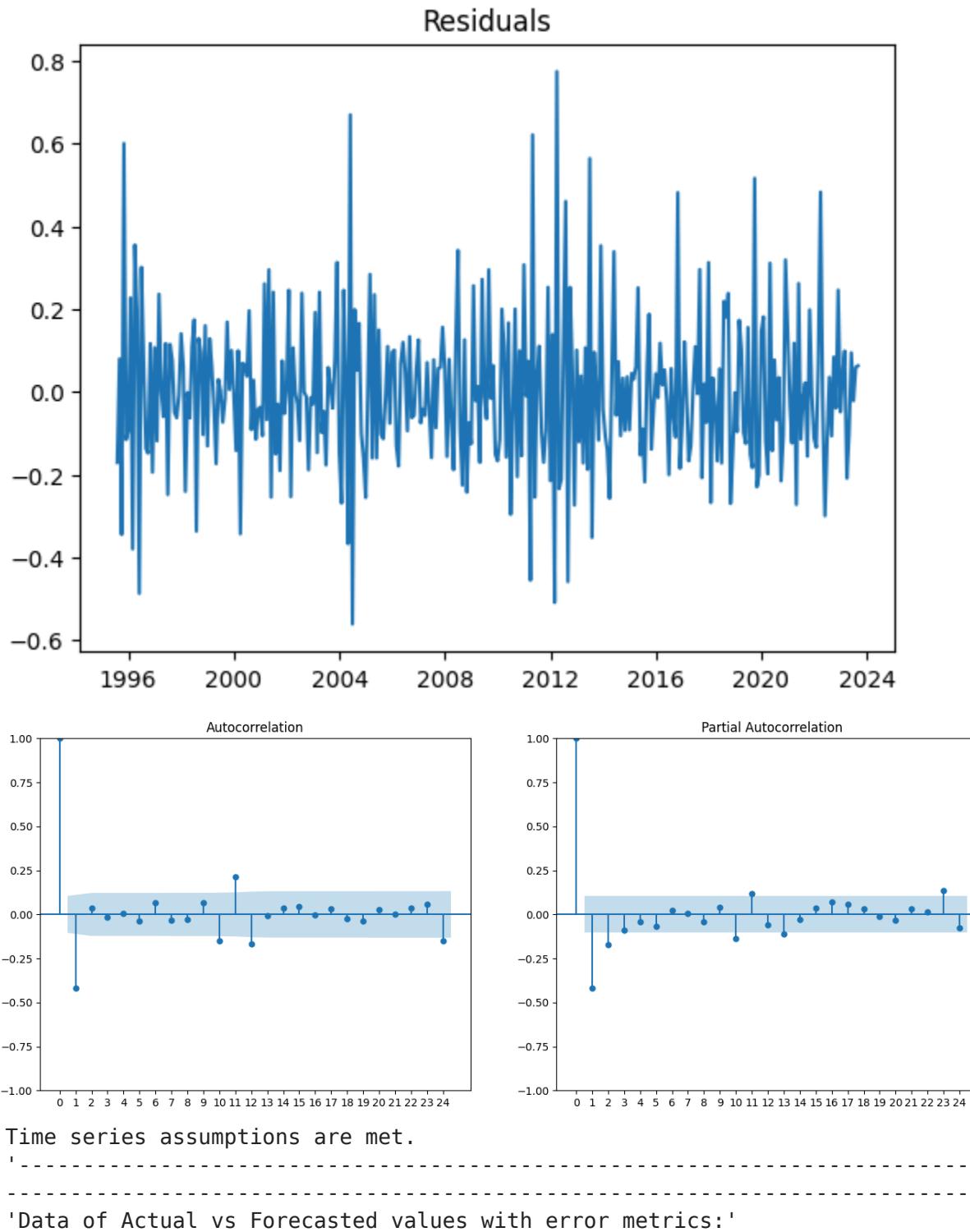
Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

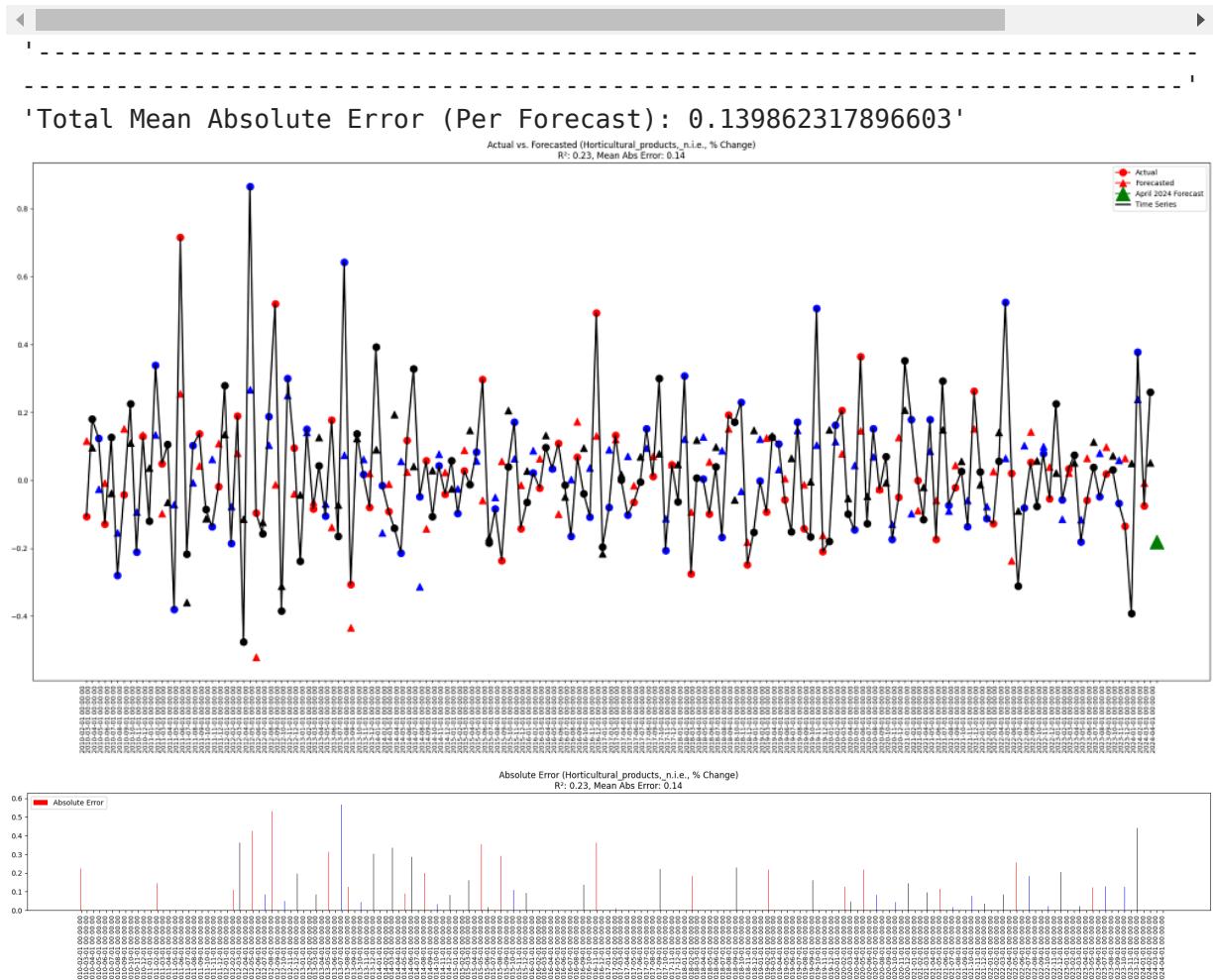


Plotting Residuals for Homoscedasticity Check...



| Horticultural_products,_n.i.e. |            | actual    | forecast  | error     | abs_error | squared_error | abs_ |
|--------------------------------|------------|-----------|-----------|-----------|-----------|---------------|------|
|                                | 2010-02-01 | -0.106685 | 0.115713  | -0.222398 | 0.222398  | 0.049461      |      |
|                                | 2010-03-01 | 0.179750  | 0.096391  | 0.083360  | 0.083360  | 0.006949      |      |
|                                | 2010-04-01 | 0.124127  | -0.026217 | 0.150344  | 0.150344  | 0.022603      |      |
|                                | 2010-05-01 | -0.128591 | -0.008212 | -0.120379 | 0.120379  | 0.014491      |      |
|                                | 2010-06-01 | 0.127080  | -0.037109 | 0.164189  | 0.164189  | 0.026958      |      |
|                                | ...        | ...       | ...       | ...       | ...       | ...           | ...  |
|                                | 2023-12-01 | -0.391910 | 0.050209  | -0.442119 | 0.442119  | 0.195469      |      |
|                                | 2024-01-01 | 0.377036  | 0.239059  | 0.137977  | 0.137977  | 0.019038      |      |
|                                | 2024-02-01 | -0.074509 | -0.008147 | -0.066362 | 0.066362  | 0.004404      |      |
|                                | 2024-03-01 | 0.259410  | 0.051454  | 0.207956  | 0.207956  | 0.043246      |      |
|                                | 2024-04-01 | NaN       | -0.181411 | NaN       | NaN       | NaN           | NaN  |

171 rows × 6 columns



## Animal\_products

```
In [25]: df = df_export_ANALYSIS.copy()
name = df.columns[6]
display(f"Component: {name}")

'Component: Animal_products'

In [26]: df_accuracy["forecast"] = cap_extreme_values(df_accuracy[["forecast"]])["for

In [27]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value.pkl') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined.pkl")
df_accuracy["forecast"] = cap_extreme_values(df_accuracy[["forecast"]])["forecast"]
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Animal_products |
|------------|-----------------|
| 1995-02-01 | 0.030157        |
| 1995-03-01 | 0.276753        |
| 1995-04-01 | -0.182274       |
| 1995-05-01 | 0.042884        |
| 1995-06-01 | -0.021916       |
| ...        | ...             |
| 2023-11-01 | -0.027080       |
| 2023-12-01 | -0.106263       |
| 2024-01-01 | 0.126101        |
| 2024-02-01 | 0.014086        |
| 2024-03-01 | 0.052249        |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -5.564124089868082
p-value: 1.5183085125402818e-06
Critical Value 1%: -3.4499043309021955
Critical Value 5%: -2.870154885338925
Critical Value 10%: -2.5713597190254385
is_stationary: True
```

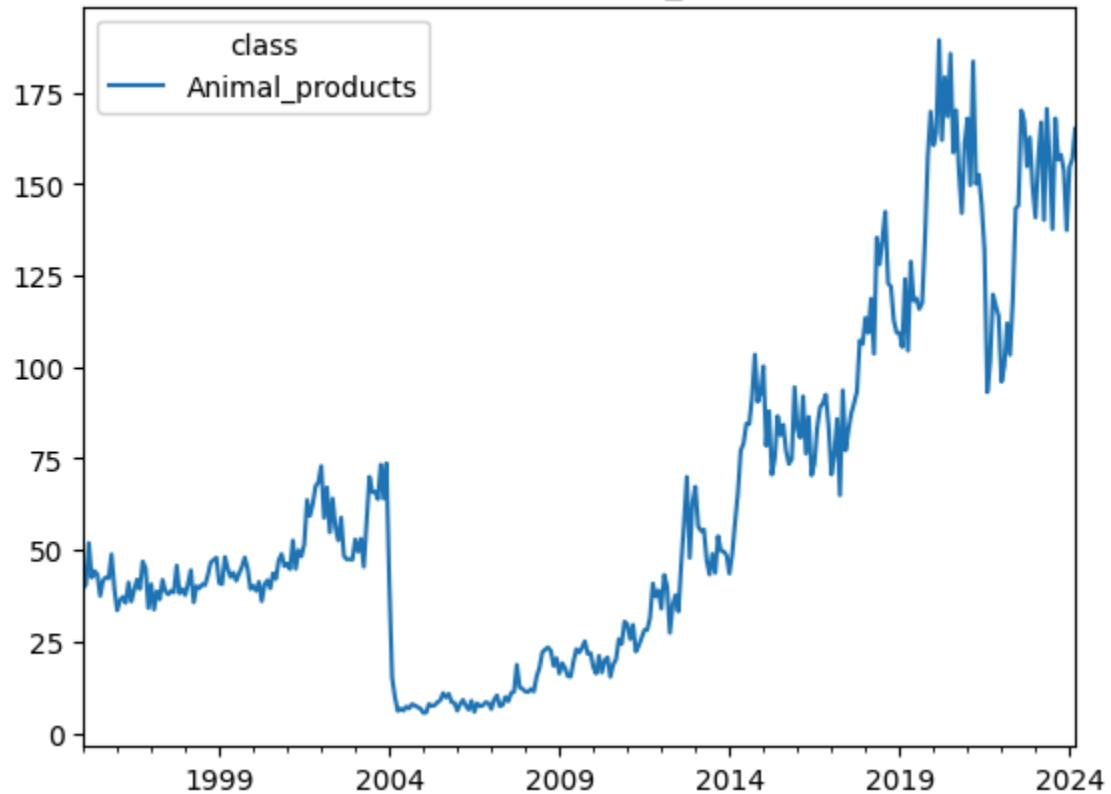
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.12278439897121088
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

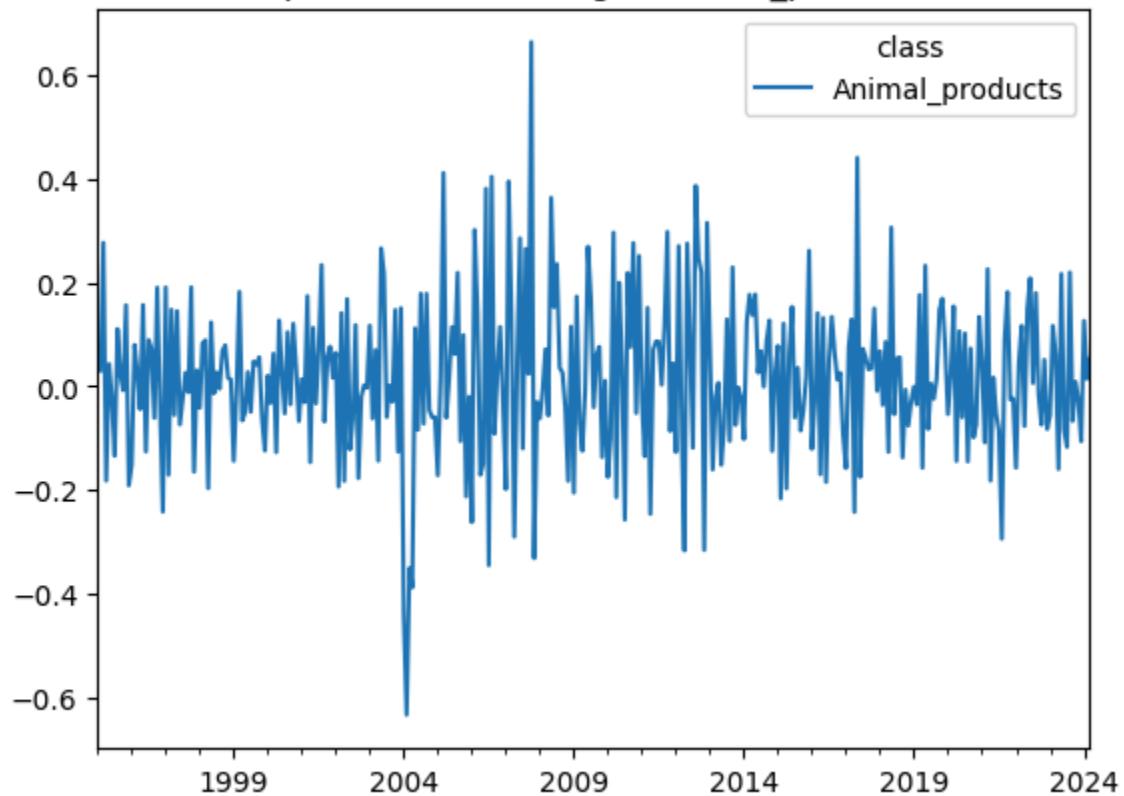
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

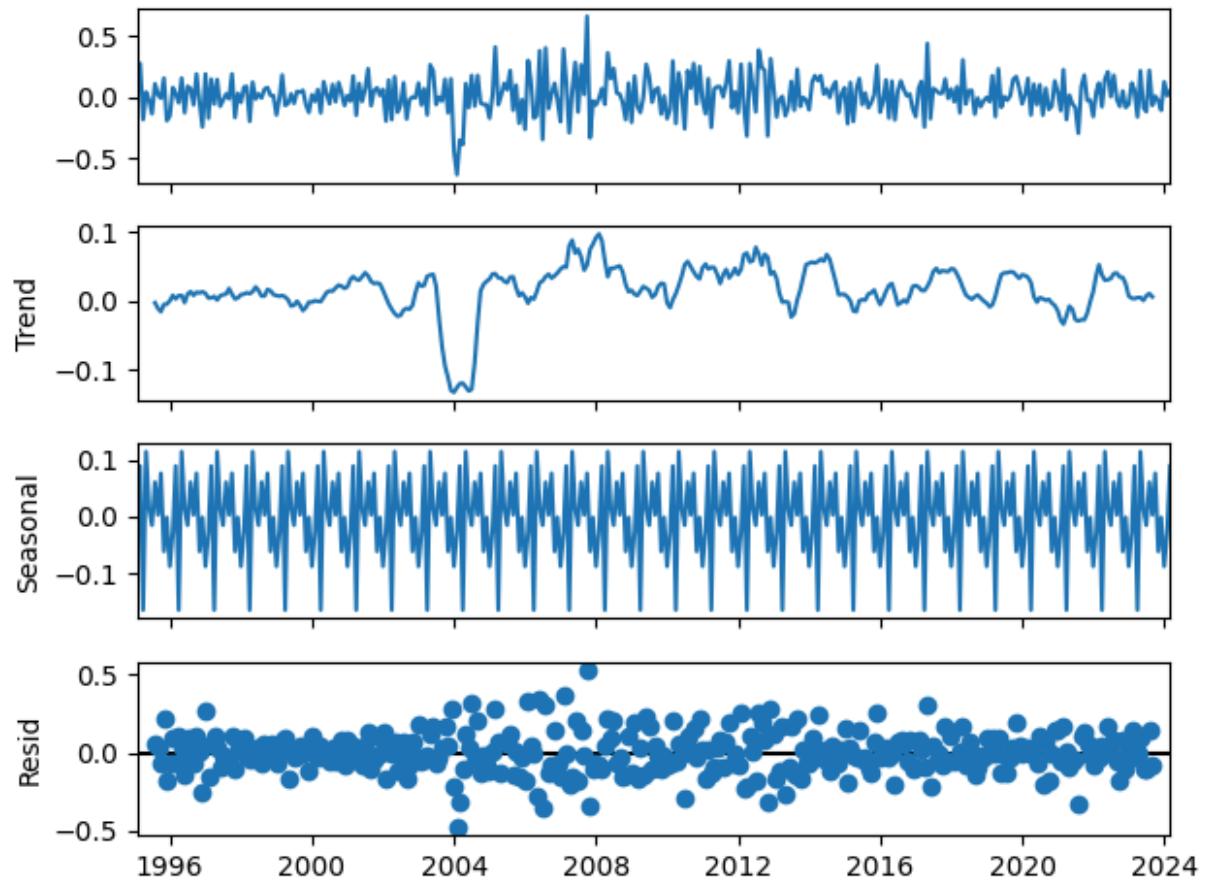
```
result = kpss(series, regression='c')
```

## Export Value (Animal\_products)



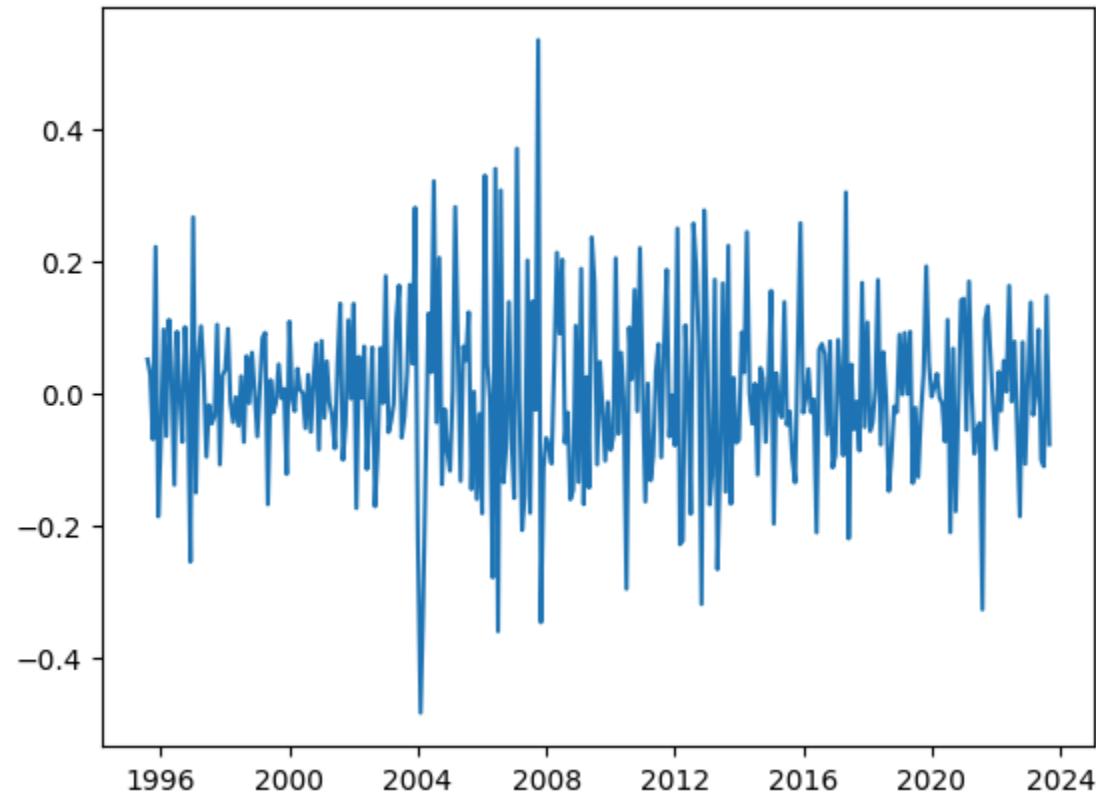
## Export Value % Change (Animal\_products)

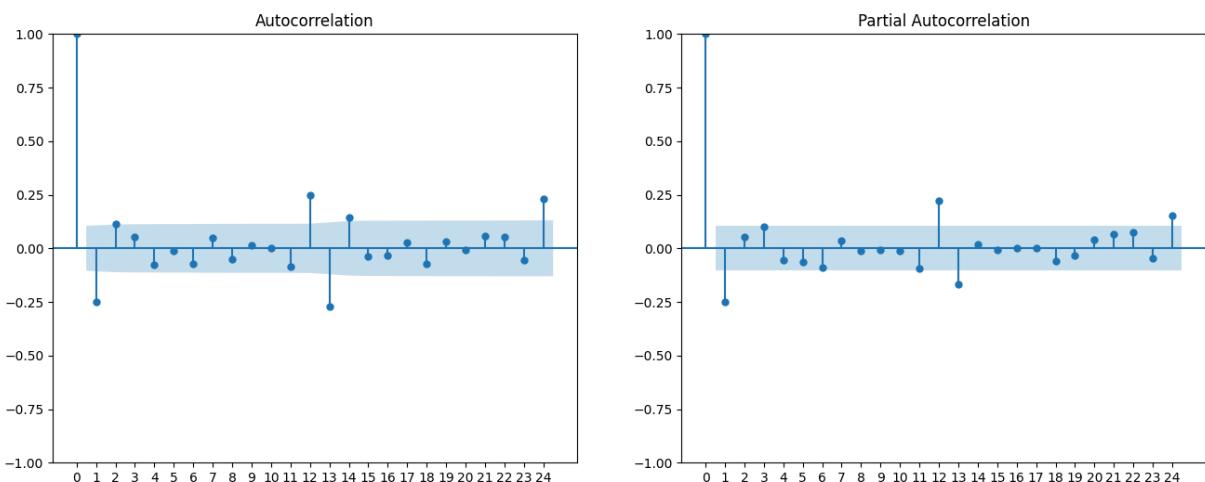




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

Running ADF Test for Stationarity...

ADF Statistic: -5.607227641769313

p-value: 1.2263618183651165e-06

Critical Value 1%: -3.449962981927952

Critical Value 5%: -2.870180642420163

Critical Value 10%: -2.5713734527352607

is\_stationary: True

Running KPSS Test for Stationarity...

KPSS Statistic: 0.15598890721132364

p-value: 0.1

Critical Value 10%: 0.347

Critical Value 5%: 0.463

Critical Value 2.5%: 0.574

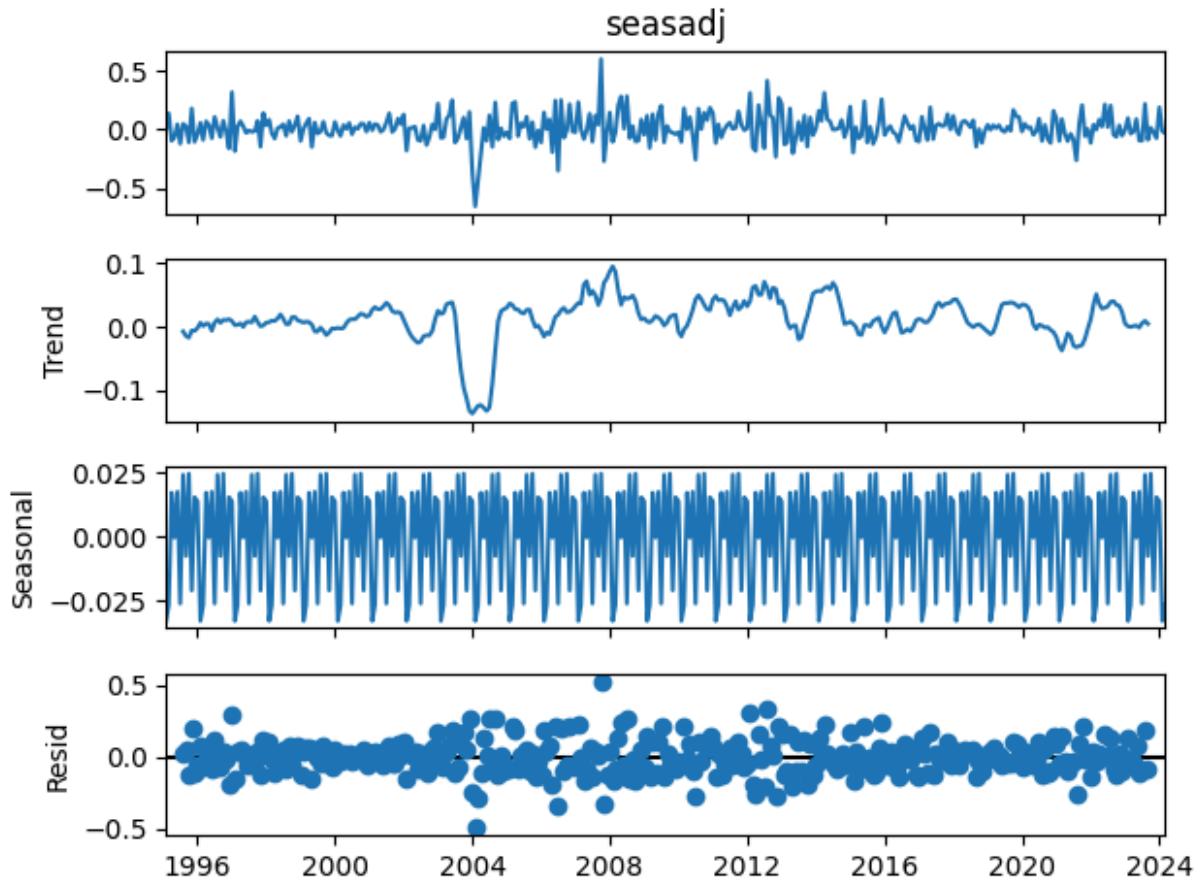
Critical Value 1%: 0.739

is\_stationary: True

Decomposing the Series...

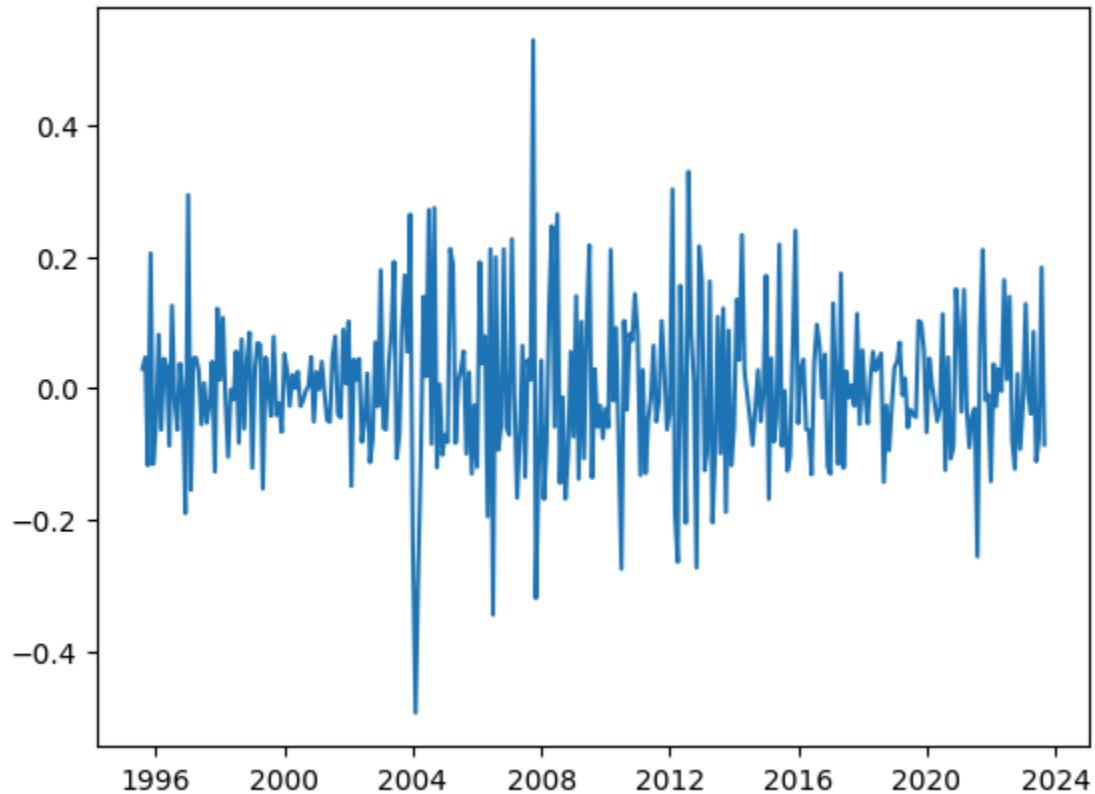
/home/wheelfredie/scripts/BoT\_Exports/helper.py:61: InterpolationWarning: The test statistic is outside of the range of p-values available in the look-up table. The actual p-value is greater than the p-value returned.

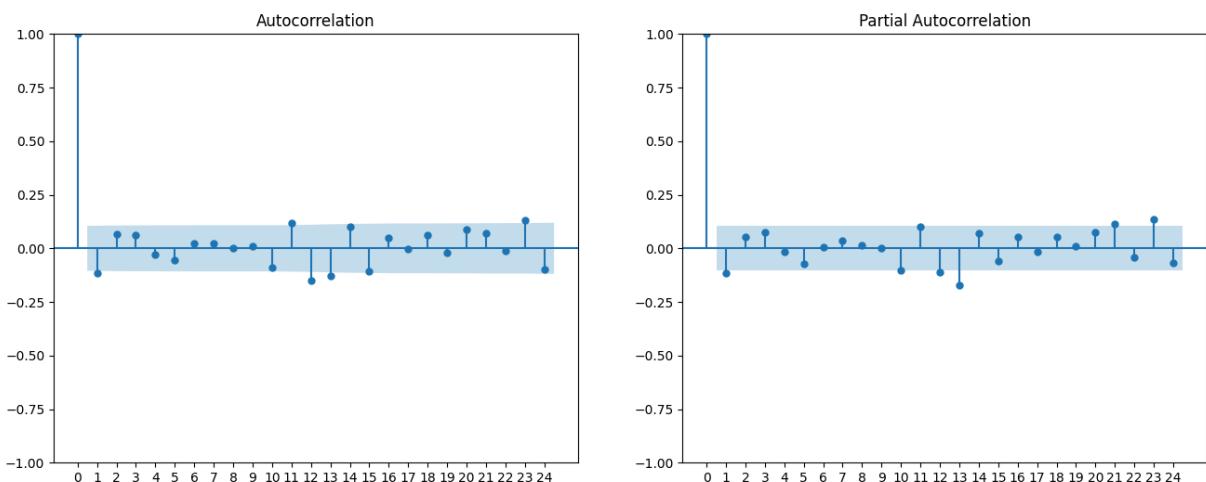
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals



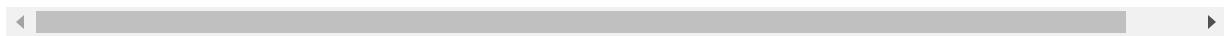


Time series assumptions are met.

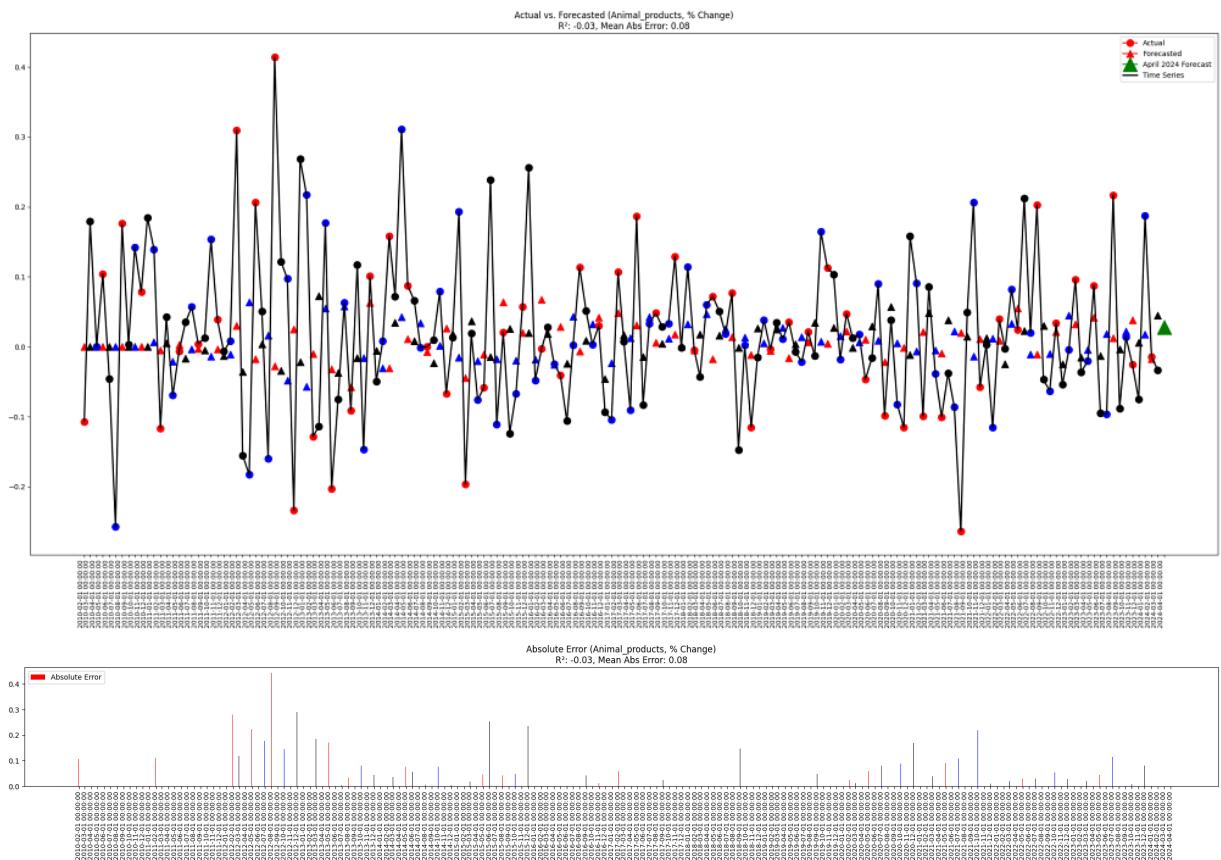
'Data of Actual vs Forecasted values with error metrics:'

| <u>Animal_products</u> | <u>actual</u> | <u>forecast</u> | <u>error</u> | <u>abs_error</u> | <u>squared_error</u> | <u>abs_percentage_</u> |
|------------------------|---------------|-----------------|--------------|------------------|----------------------|------------------------|
| <b>2010-02-01</b>      | -0.107153     | 0.000000        | -0.107153    | 0.107153         | 1.148166e-02         | 1.00                   |
| <b>2010-03-01</b>      | 0.179511      | 0.000000        | 0.179511     | 0.179511         | 3.222412e-02         | 1.00                   |
| <b>2010-04-01</b>      | -0.000333     | 0.000000        | -0.000333    | 0.000333         | 1.111411e-07         | 1.00                   |
| <b>2010-05-01</b>      | 0.103868      | 0.000000        | 0.103868     | 0.103868         | 1.078866e-02         | 1.00                   |
| <b>2010-06-01</b>      | -0.045780     | 0.000000        | -0.045780    | 0.045780         | 2.095834e-03         | 1.00                   |
| ...                    | ...           | ...             | ...          | ...              | ...                  | ...                    |
| <b>2023-12-01</b>      | -0.075139     | 0.005908        | -0.081046    | 0.081046         | 6.568517e-03         | 1.07                   |
| <b>2024-01-01</b>      | 0.187127      | 0.017227        | 0.169900     | 0.169900         | 2.886585e-02         | 0.90                   |
| <b>2024-02-01</b>      | -0.014415     | -0.017653       | 0.003239     | 0.003239         | 1.048883e-05         | 0.22                   |
| <b>2024-03-01</b>      | -0.033308     | 0.044488        | -0.077797    | 0.077797         | 6.052299e-03         | 2.33                   |
| <b>2024-04-01</b>      | NaN           | 0.027897        | NaN          | NaN              | NaN                  | NaN                    |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.08294493473108006'



## Fishery

```
In [28]: df = df_export_ANALYSIS.copy()
name = df.columns[7]
display(f"Component: {name}")

'Component: Fishery'
```

```
In [29]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value')
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
```

```
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class             | Fishery   |
|-------------------|-----------|
| <b>1995-02-01</b> | -0.157801 |
| <b>1995-03-01</b> | 0.387814  |
| <b>1995-04-01</b> | -0.177340 |
| <b>1995-05-01</b> | 0.182081  |
| <b>1995-06-01</b> | 0.081511  |
| ...               | ...       |
| <b>2023-11-01</b> | -0.065282 |
| <b>2023-12-01</b> | -0.074635 |
| <b>2024-01-01</b> | 0.065773  |
| <b>2024-02-01</b> | -0.058474 |
| <b>2024-03-01</b> | 0.219910  |

350 rows x 1 columns

'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'

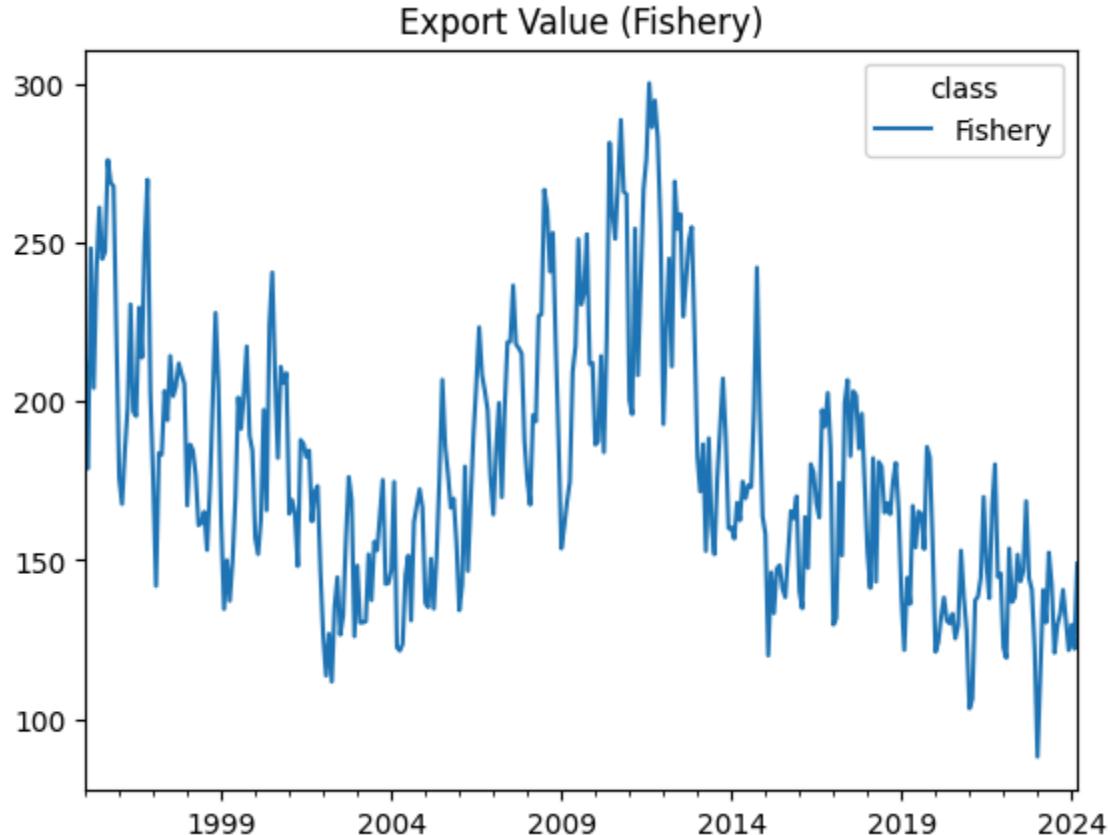
```
Running ADF Test for Stationarity...
ADF Statistic: -4.159090391460465
p-value: 0.0007722157810662703
Critical Value 1%: -3.450141065277327
Critical Value 5%: -2.870258846235788
Critical Value 10%: -2.571415151457764
is_stationary: True
```

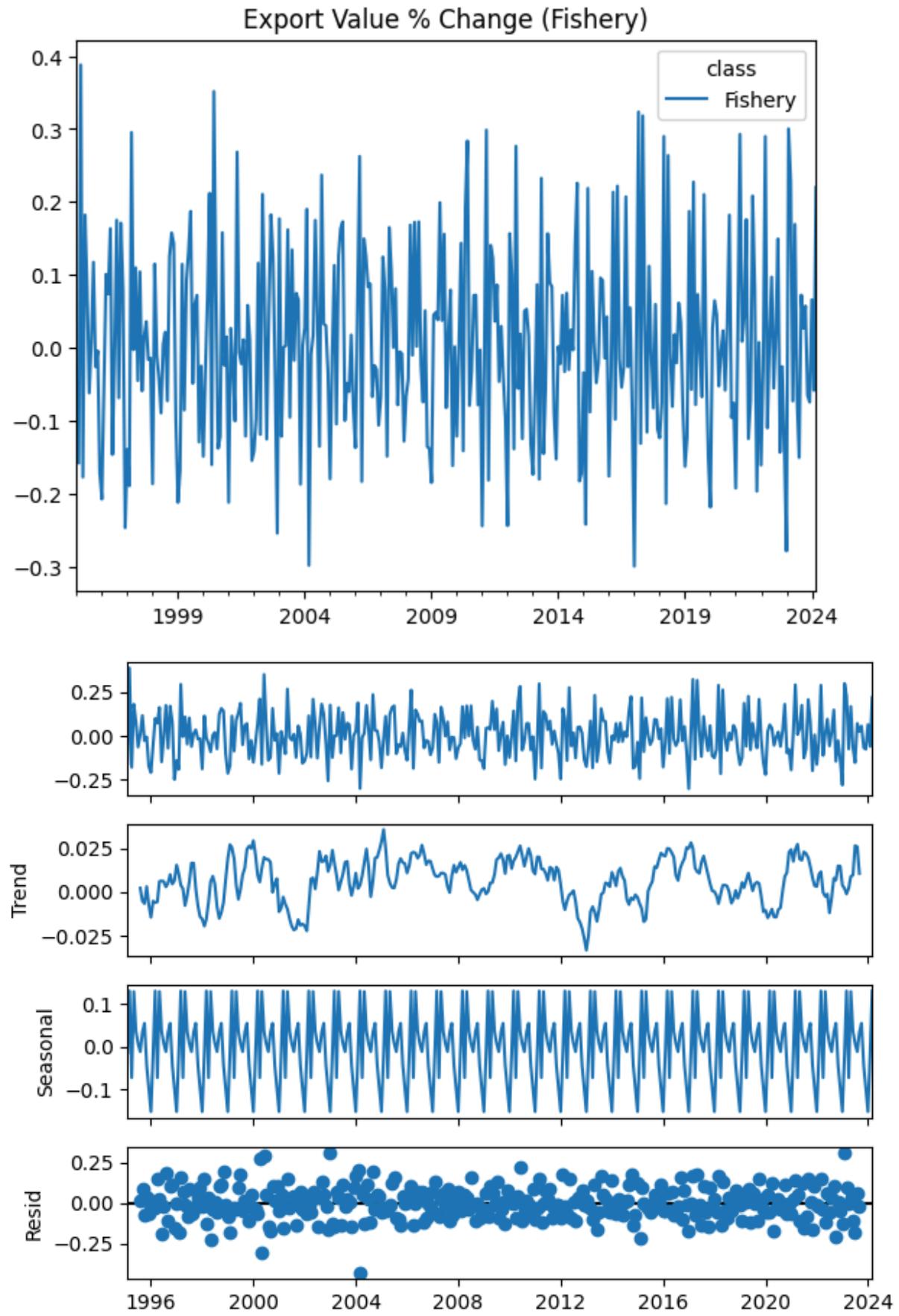
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.02731389579931995
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

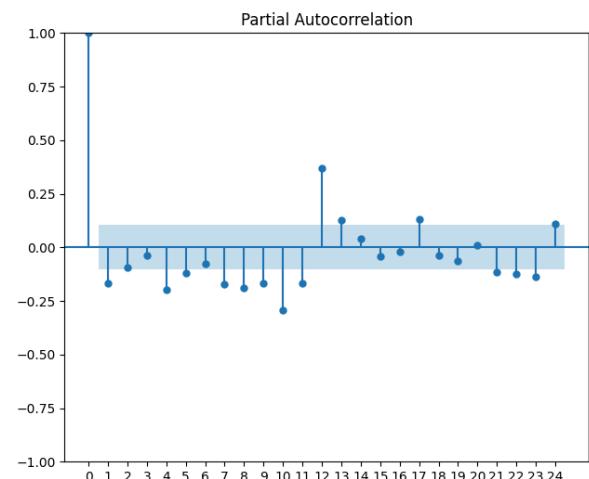
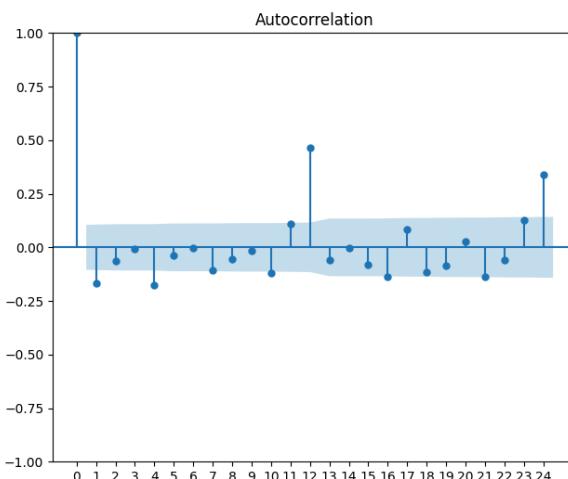
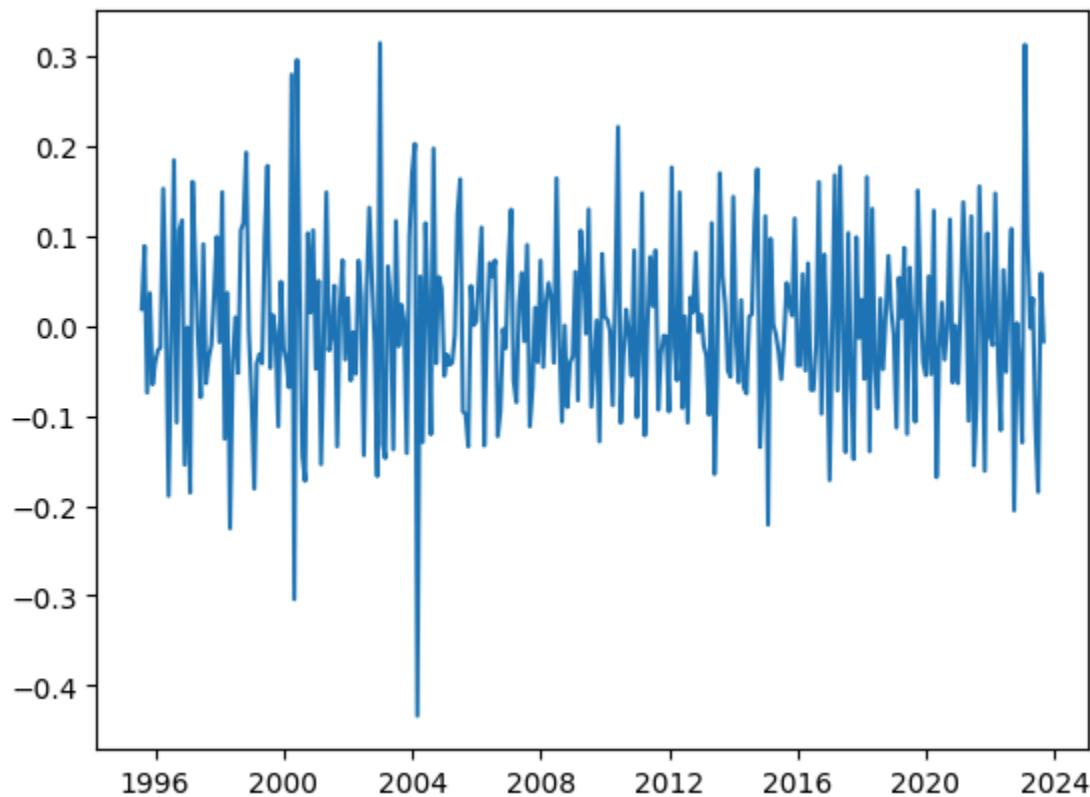
```
result = kpss(series, regression='c')
```





Plotting Residuals for Homoscedasticity Check...

### Residuals



Time series assumptions are met.

True

'-----'  
-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

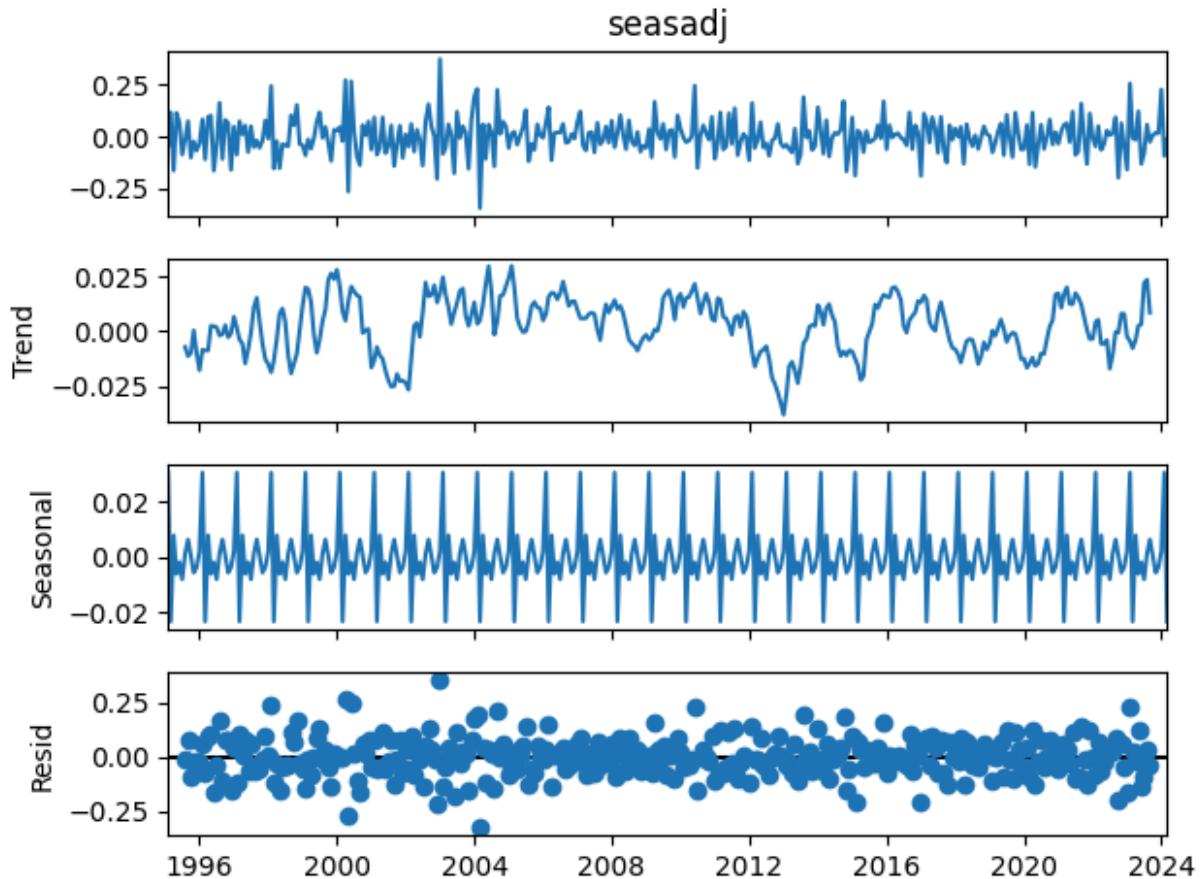
Running ADF Test for Stationarity...  
 ADF Statistic: -5.31546244079531  
 p-value: 5.085947329960103e-06  
 Critical Value 1%: -3.4497880749874628  
 Critical Value 5%: -2.870103829170425  
 Critical Value 10%: -2.571332495975496  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.03573653741963814  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

Decomposing the Series...

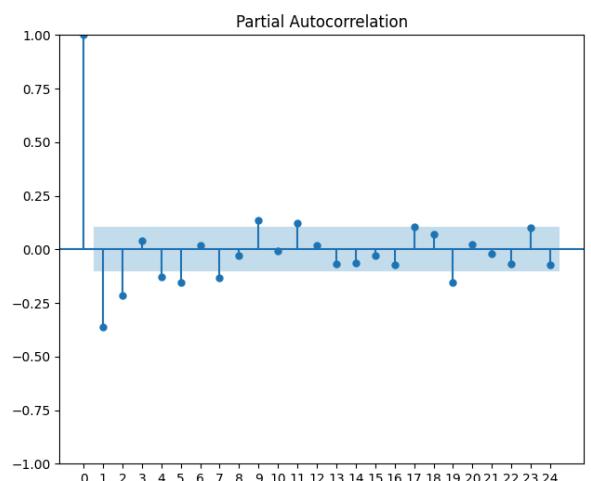
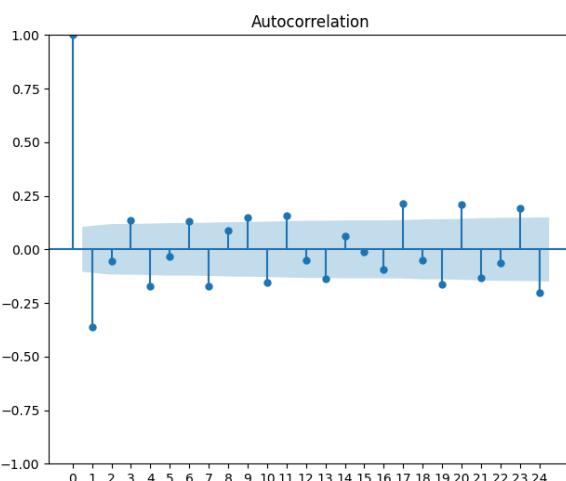
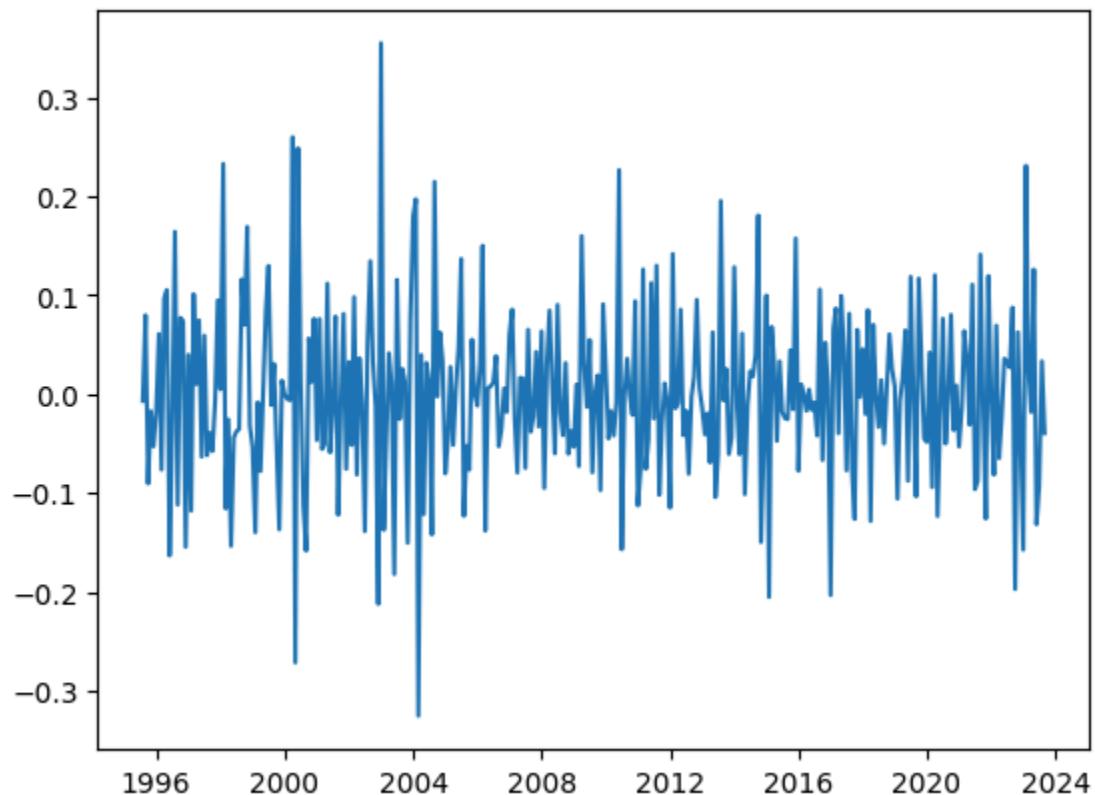
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals

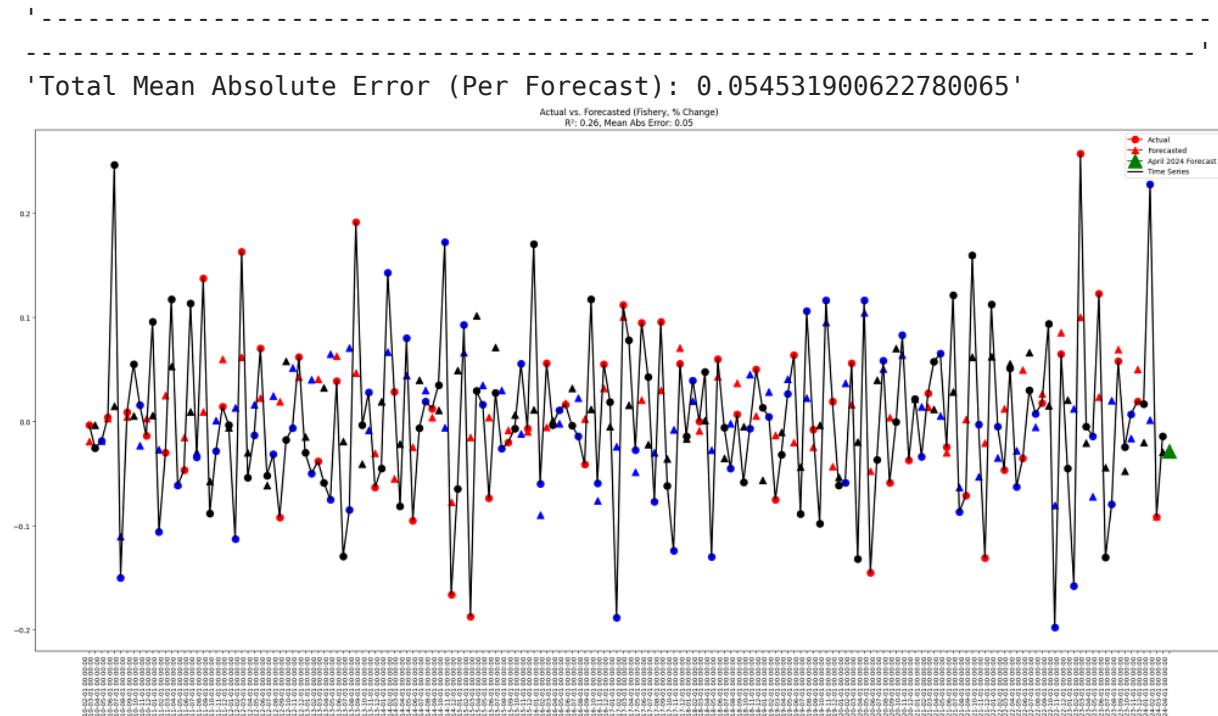


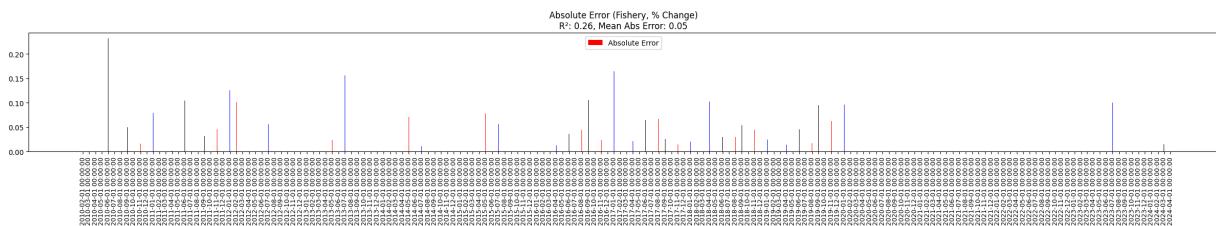
Time series assumptions are met.

'-----'  
-----'  
'Data of Actual vs Forecasted values with error metrics:'

| Fishery    | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01 | -0.003224 | -0.018818 | 0.015594  | 0.015594  | 2.431753e-04  | 4.837173             |
| 2010-03-01 | -0.025396 | -0.003122 | -0.022274 | 0.022274  | 4.961116e-04  | 0.877055             |
| 2010-04-01 | -0.018523 | -0.018748 | 0.000225  | 0.000225  | 5.048127e-08  | 0.012130             |
| 2010-05-01 | 0.003802  | 0.003036  | 0.000767  | 0.000767  | 5.879383e-07  | 0.201651             |
| 2010-06-01 | 0.246505  | 0.014925  | 0.231579  | 0.231579  | 5.362899e-02  | 0.939452             |
| ...        | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01 | 0.016574  | -0.020221 | 0.036795  | 0.036795  | 1.353868e-03  | 2.220071             |
| 2024-01-01 | 0.227966  | 0.001427  | 0.226539  | 0.226539  | 5.131977e-02  | 0.993740             |
| 2024-02-01 | -0.091556 | -0.091262 | -0.000294 | 0.000294  | 8.632802e-08  | 0.003209             |
| 2024-03-01 | -0.014015 | -0.029349 | 0.015334  | 0.015334  | 2.351302e-04  | 1.094119             |
| 2024-04-01 | NaN       | -0.028331 | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns





## Crustaceans

```
In [30]: df = df_export_ANALYSIS.copy()
name = df.columns[8]
display(f"Component: {name}")
```

'Component: Crustaceans'

```
In [31]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value.pkl', 'rb') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/{name}_accuracy.pkl")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Crustaceans |
|------------|-------------|
| 1995-02-01 | -0.205979   |
| 1995-03-01 | 0.456434    |
| 1995-04-01 | -0.166809   |
| 1995-05-01 | 0.211183    |
| 1995-06-01 | 0.085745    |
| ...        | ...         |
| 2023-11-01 | -0.011983   |
| 2023-12-01 | -0.075728   |
| 2024-01-01 | -0.078253   |
| 2024-02-01 | 0.011806    |
| 2024-03-01 | 0.133150    |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -6.3549565542307835
p-value: 2.5562651150969746e-08
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

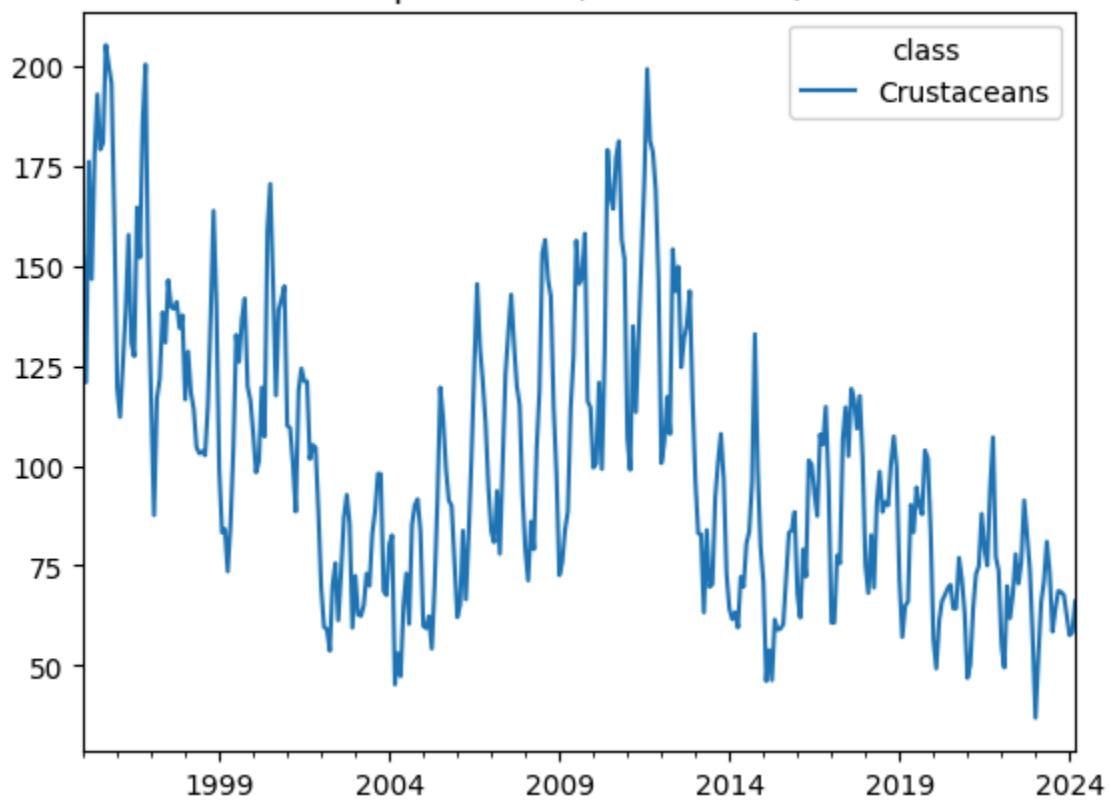
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.013817370147071744
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

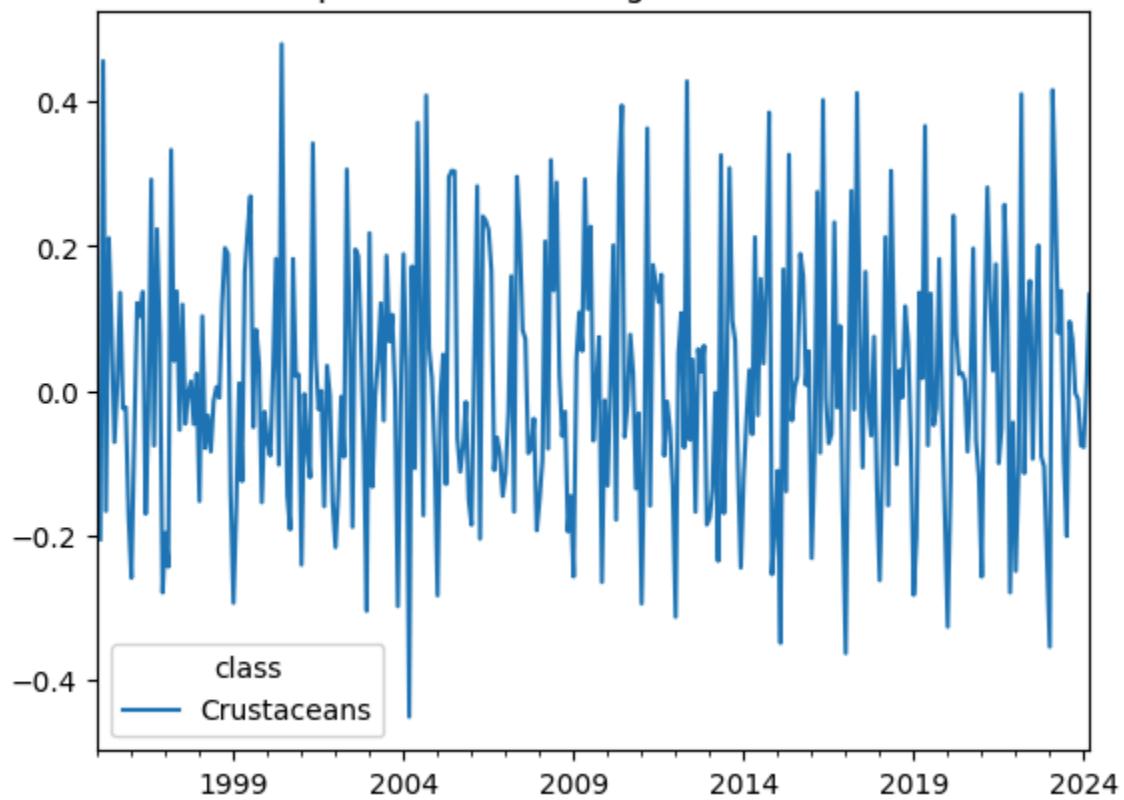
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

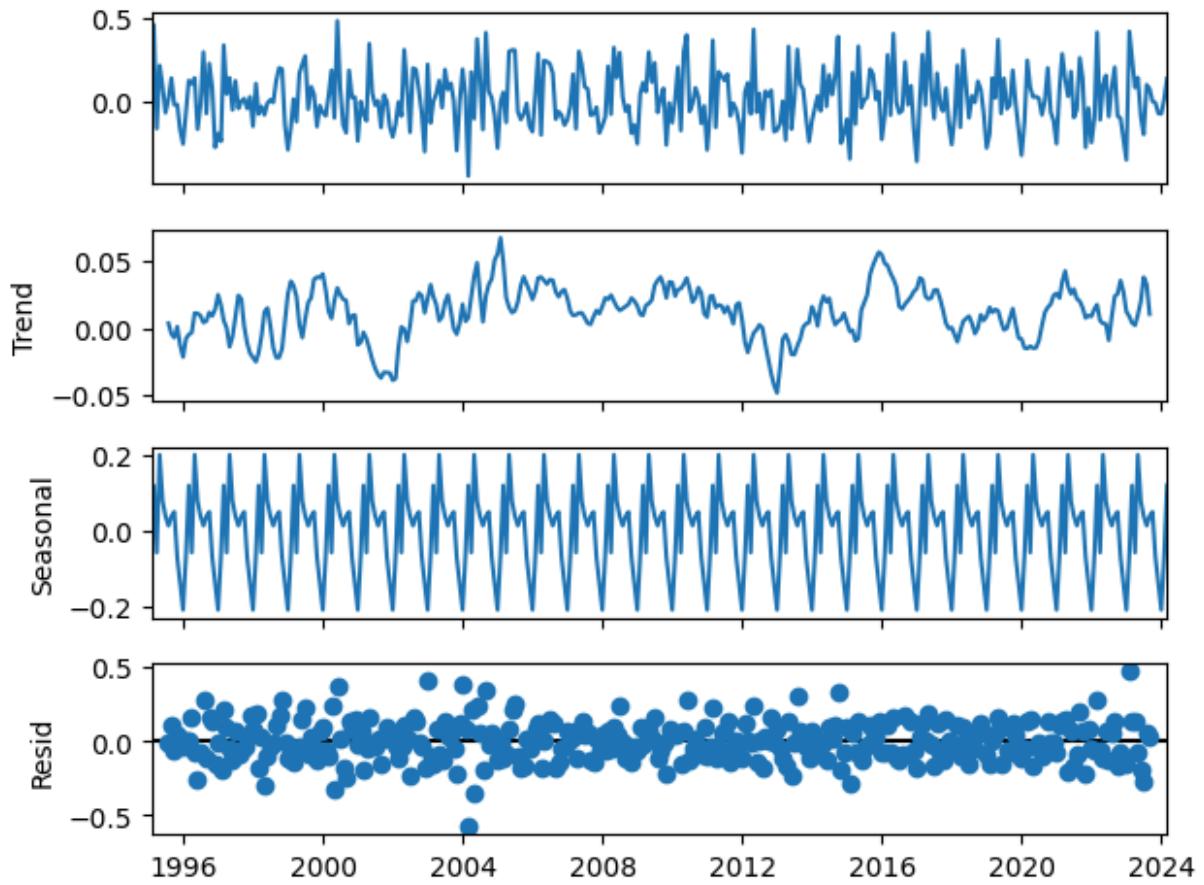
```
result = kpss(series, regression='c')
```

## Export Value (Crustaceans)

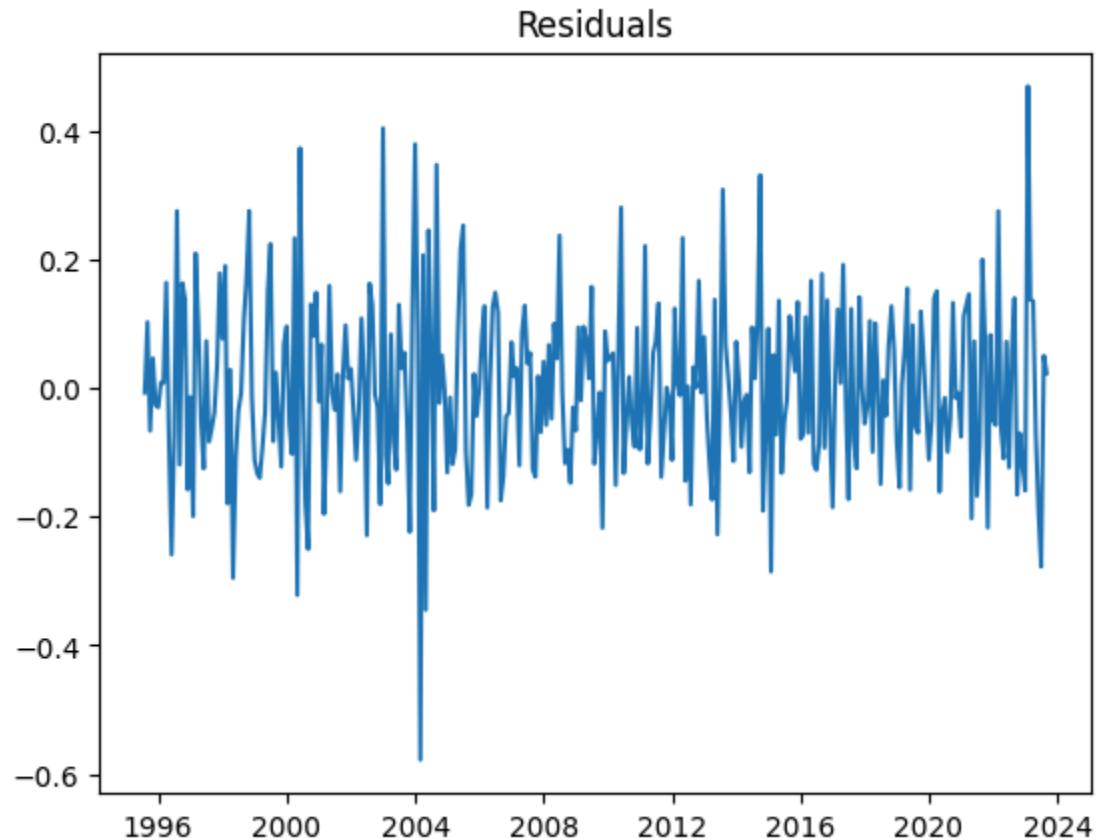


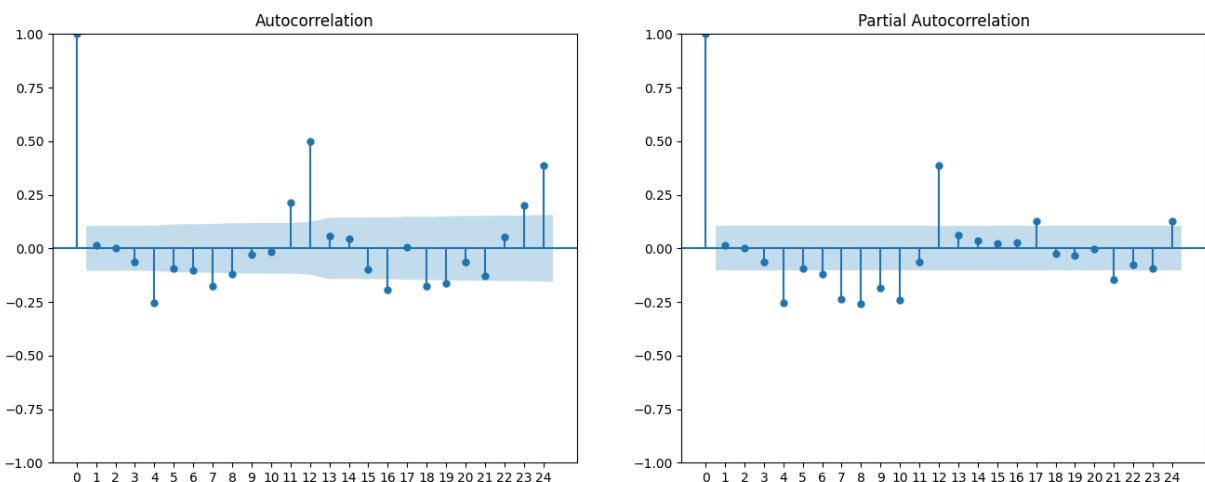
## Export Value % Change (Crustaceans)





Plotting Residuals for Homoscedasticity Check...





Time series assumptions are met.

True

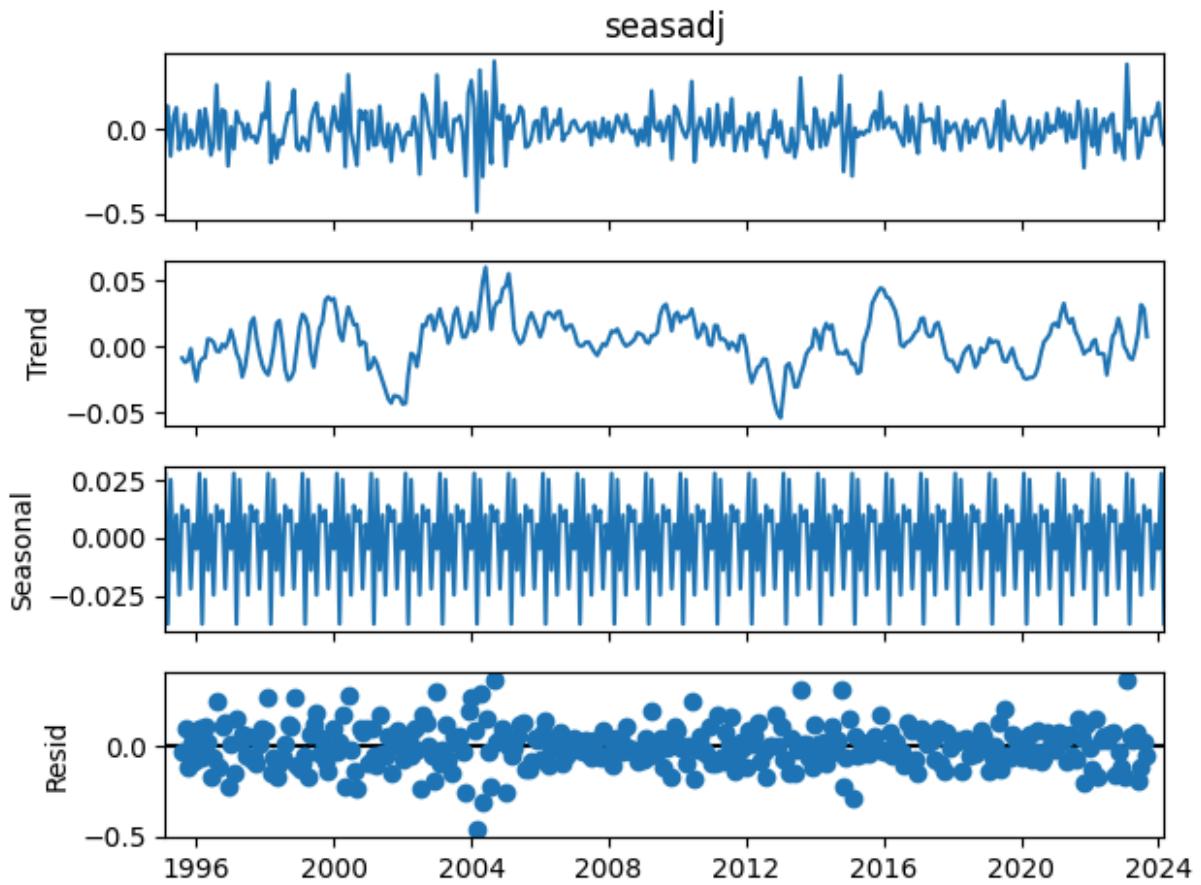
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted): '
Running ADF Test for Stationarity...
ADF Statistic: -3.9499841054732285
p-value: 0.0016962841594932894
Critical Value 1%: -3.4502011472639724
Critical Value 5%: -2.8702852297358983
Critical Value 10%: -2.5714292194077513
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.06159304469300719
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

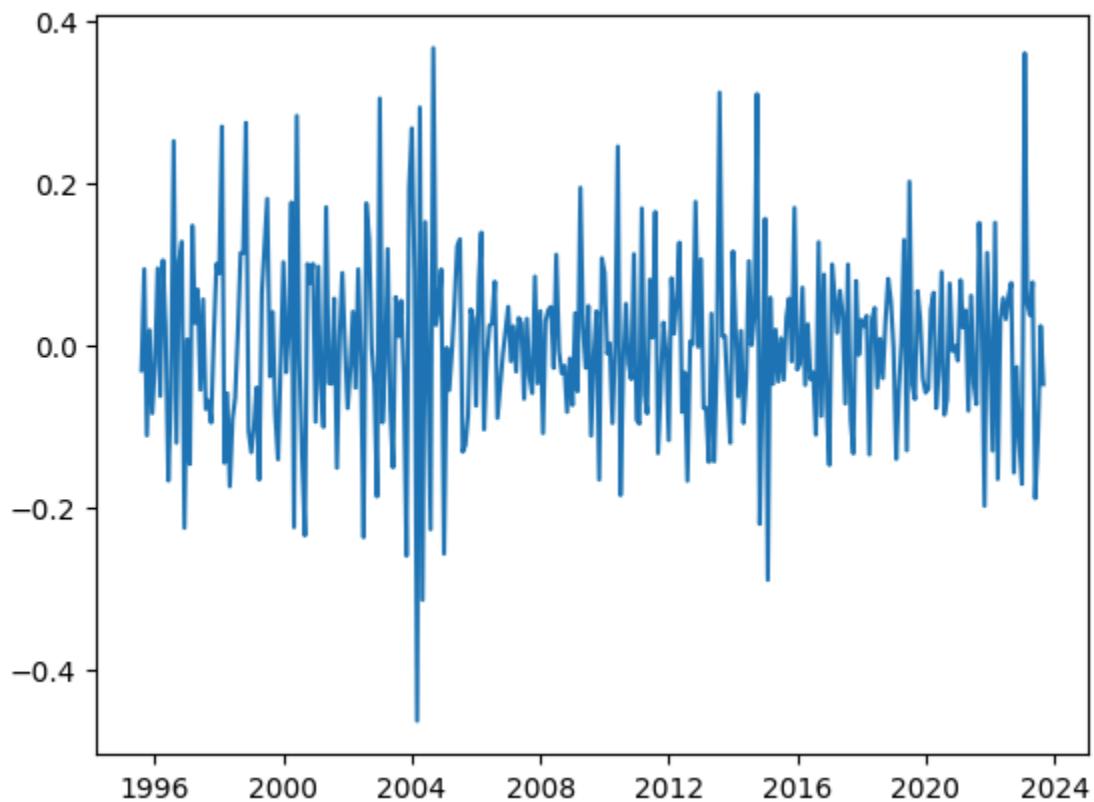
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

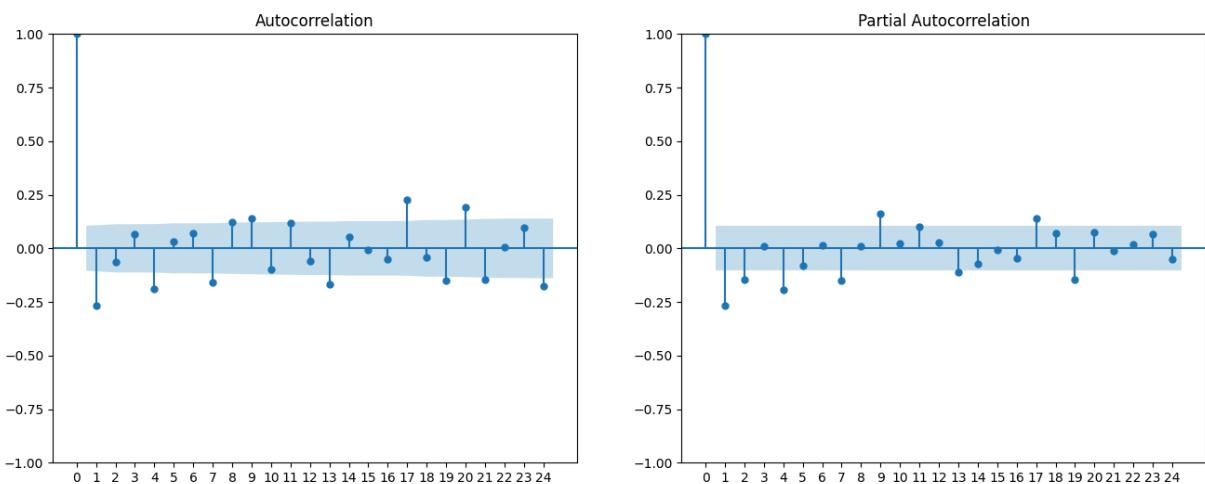
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

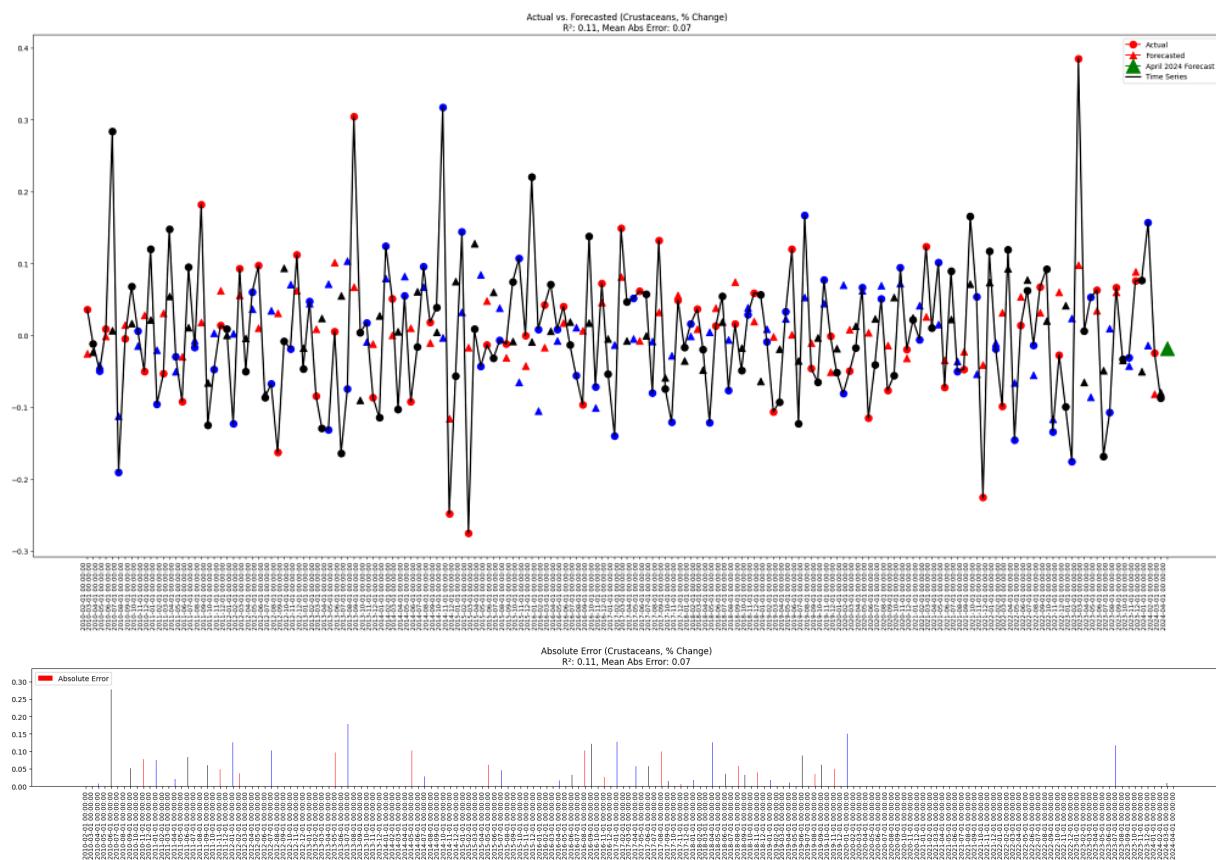
'Data of Actual vs Forecasted values with error metrics:'

| Crustaceans | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|-------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01  | 0.036319  | -0.025456 | 0.061774  | 0.061774  | 0.003816      | 1.700894             |
| 2010-03-01  | -0.011641 | -0.022825 | 0.011184  | 0.011184  | 0.000125      | 0.960703             |
| 2010-04-01  | -0.049256 | -0.041387 | -0.007869 | 0.007869  | 0.000062      | 0.159758             |
| 2010-05-01  | 0.008634  | -0.001356 | 0.009991  | 0.009991  | 0.000100      | 1.157069             |
| 2010-06-01  | 0.284011  | 0.006947  | 0.277064  | 0.277064  | 0.076764      | 0.975540             |
| ...         | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01  | 0.076918  | -0.050477 | 0.127395  | 0.127395  | 0.016229      | 1.656247             |
| 2024-01-01  | 0.157422  | -0.013545 | 0.170967  | 0.170967  | 0.029230      | 1.086042             |
| 2024-02-01  | -0.024704 | -0.081260 | 0.056556  | 0.056556  | 0.003199      | 2.289357             |
| 2024-03-01  | -0.087605 | -0.079293 | -0.008312 | 0.008312  | 0.000069      | 0.094878             |
| 2024-04-01  | NaN       | -0.018517 | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.07425169236576332'



## Fish

```
In [32]: df = df_export_ANALYSIS.copy()
name = df.columns[9]
display(f"Component: {name}")
```

'Component: Fish'

```
In [33]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

| class      | Fish      |
|------------|-----------|
| 1995-02-01 | -0.090772 |
| 1995-03-01 | 0.238017  |
| 1995-04-01 | -0.173832 |
| 1995-05-01 | 0.135100  |
| 1995-06-01 | 0.085137  |
| ...        | ...       |
| 2023-11-01 | -0.050933 |
| 2023-12-01 | -0.112646 |
| 2024-01-01 | 0.211377  |
| 2024-02-01 | -0.163618 |
| 2024-03-01 | 0.265662  |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

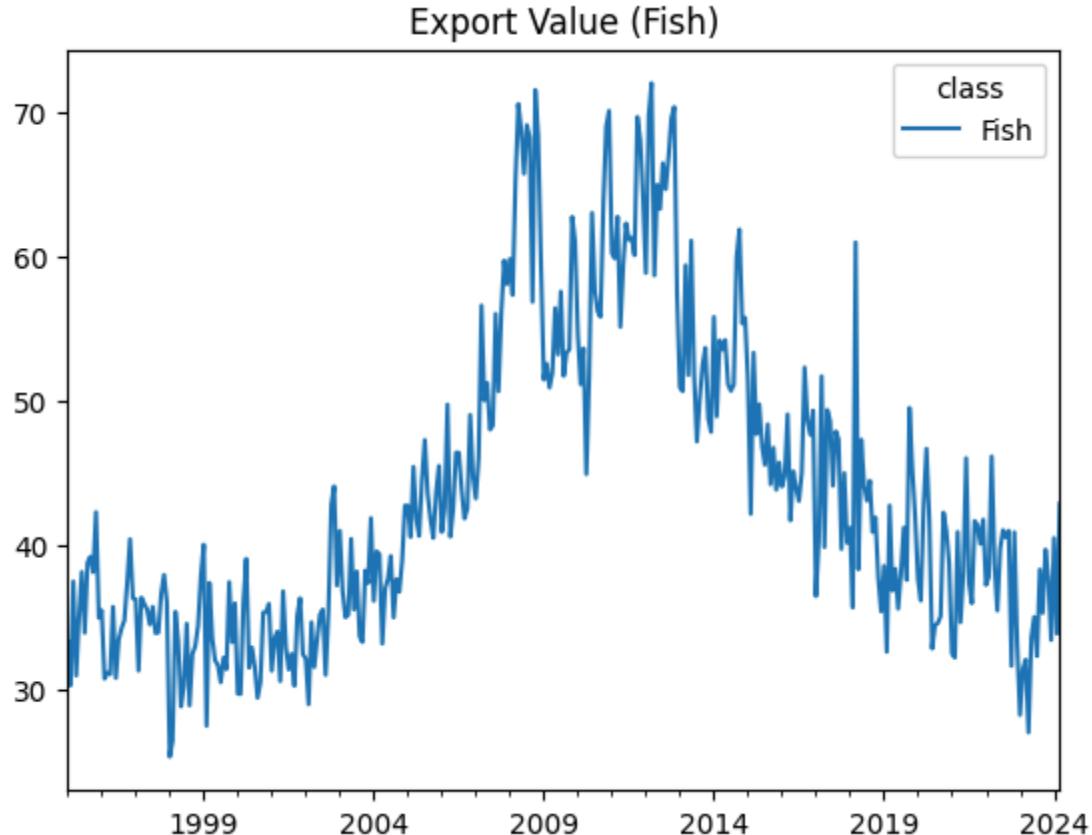
```
Running ADF Test for Stationarity...
ADF Statistic: -7.390305578435781
p-value: 8.033406251233797e-11
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

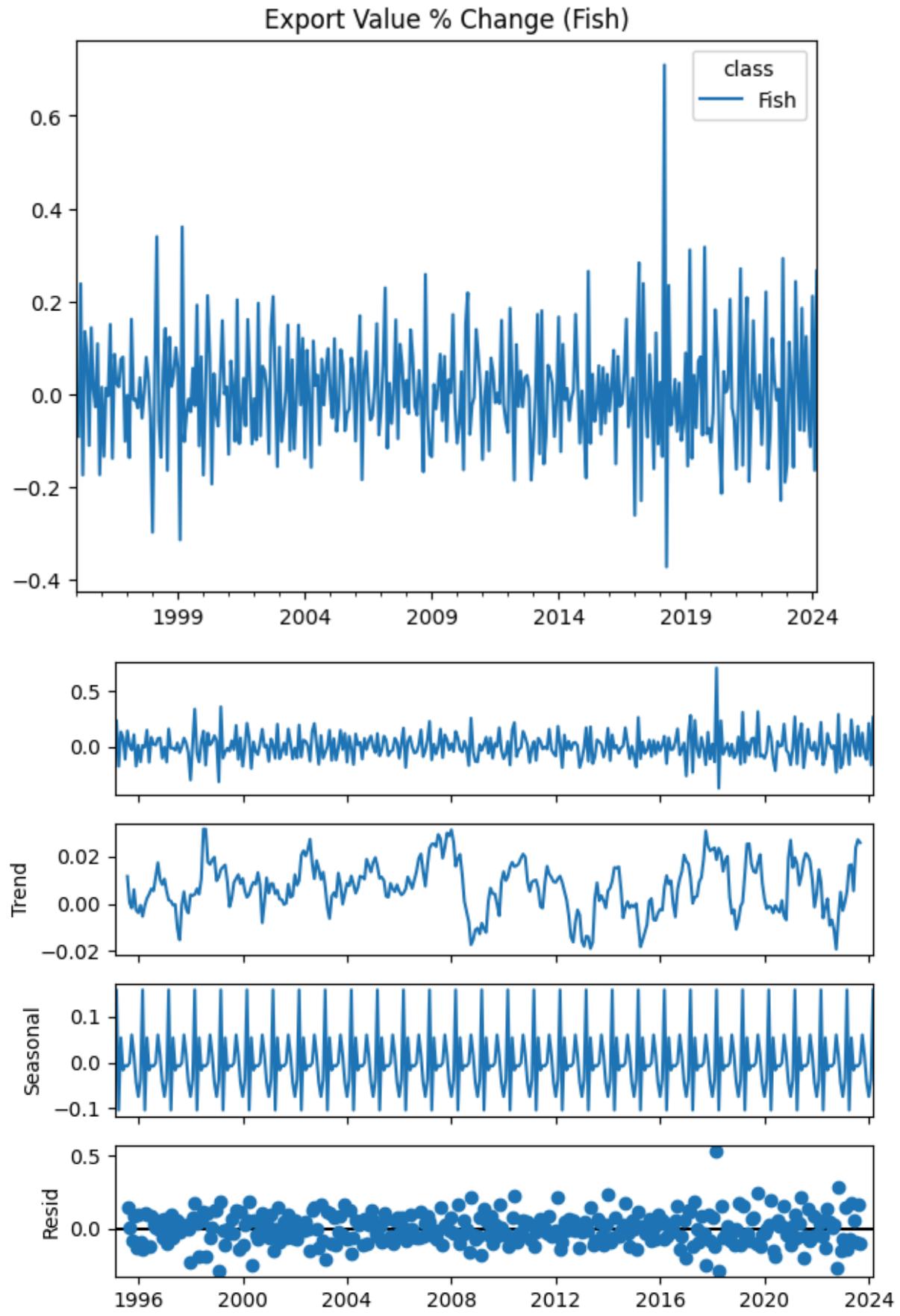
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.06789997195499971
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

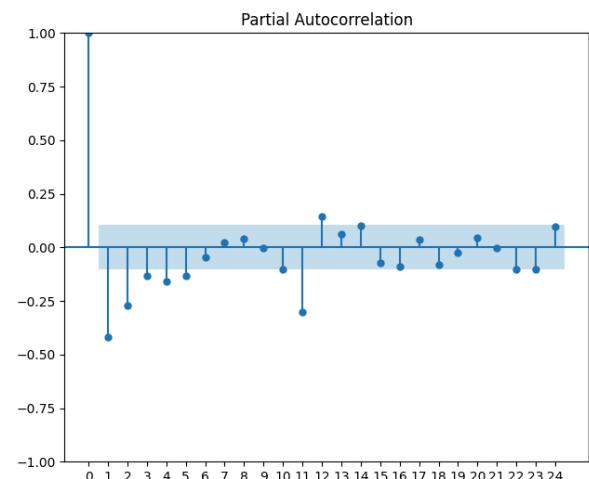
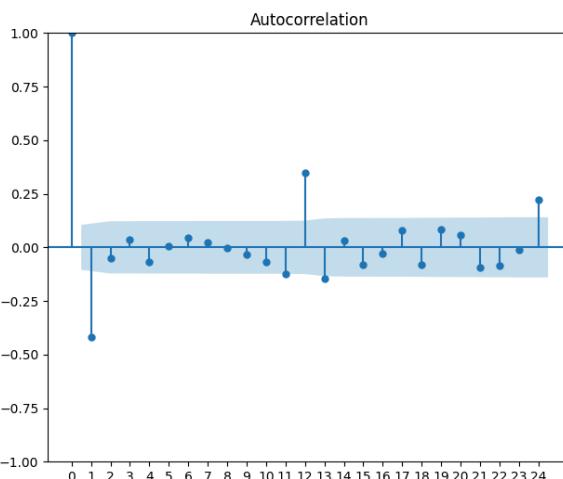
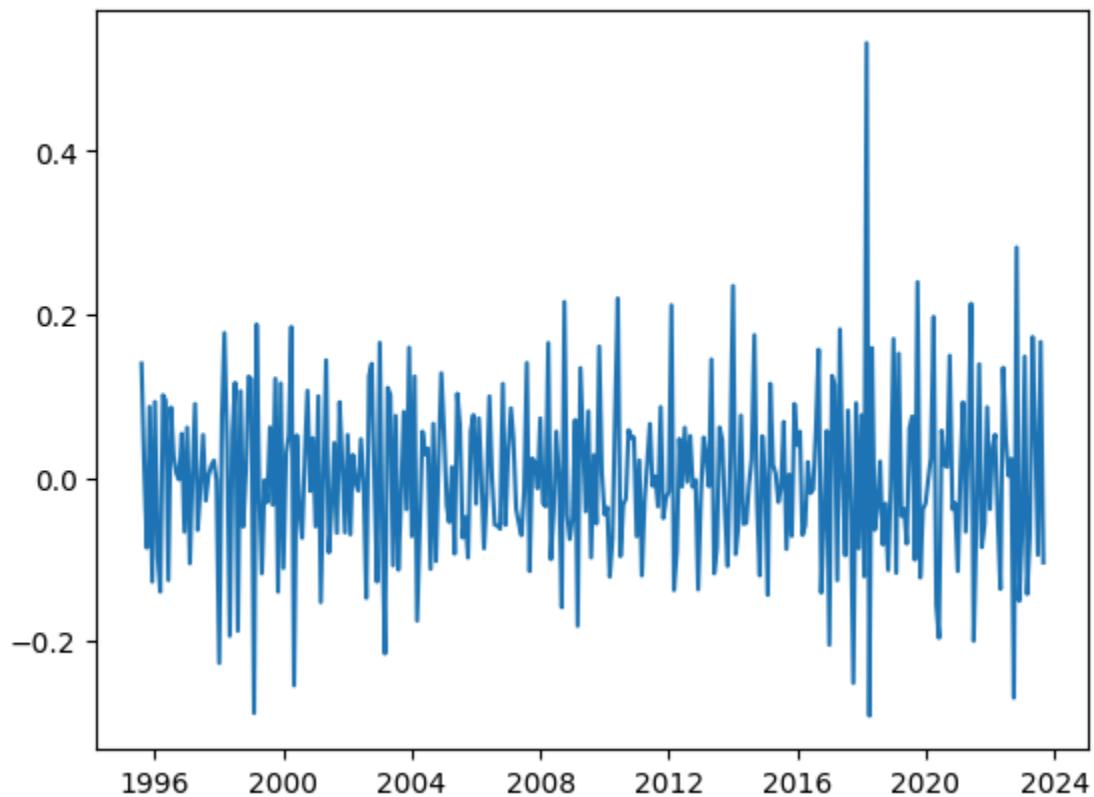
```
result = kpss(series, regression='c')
```





Plotting Residuals for Homoscedasticity Check...

### Residuals



Time series assumptions are met.

True

'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

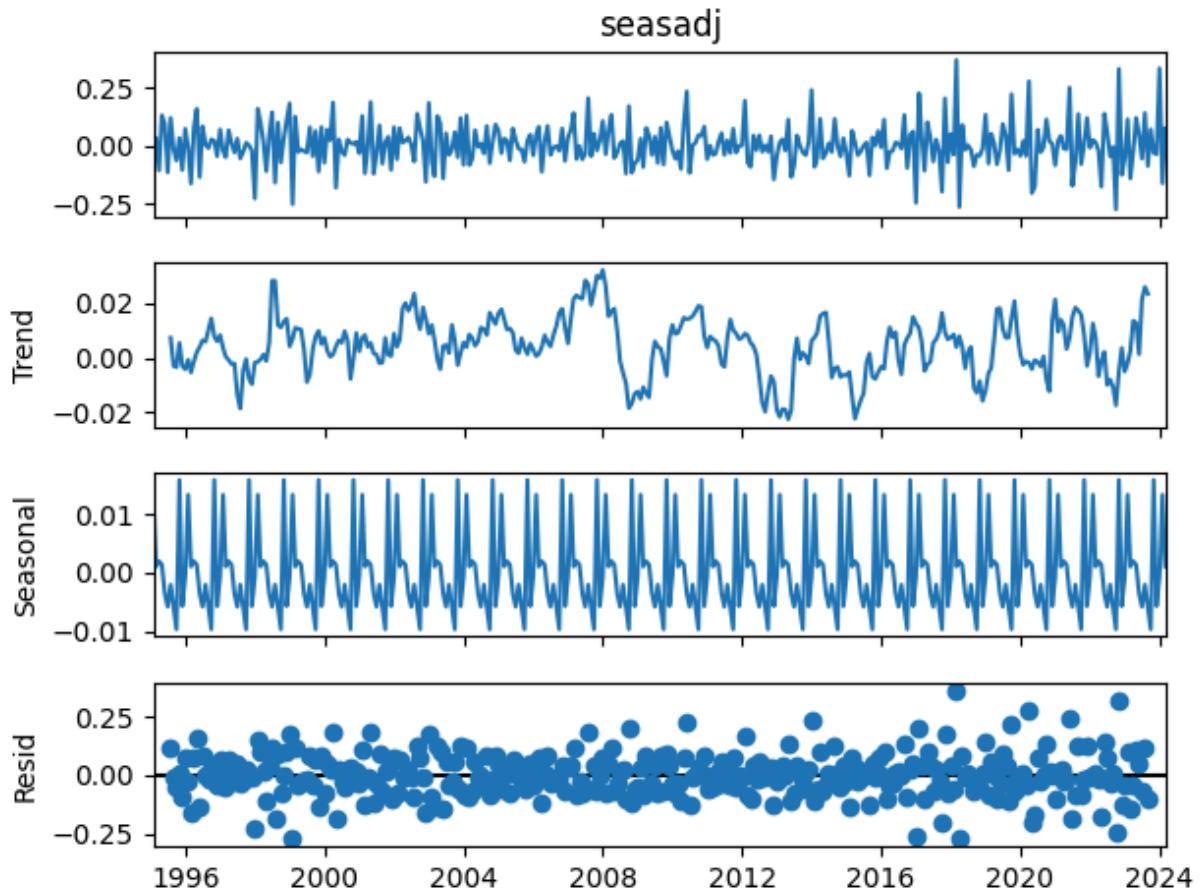
```
Running ADF Test for Stationarity...
ADF Statistic: -11.544920615708573
p-value: 3.560280465495256e-21
Critical Value 1%: -3.4494474563375737
Critical Value 5%: -2.8699542285903887
Critical Value 10%: -2.5712527305187987
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.05077578987352299
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

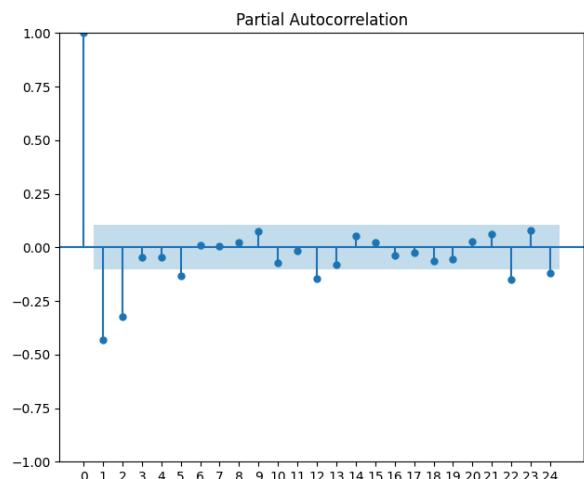
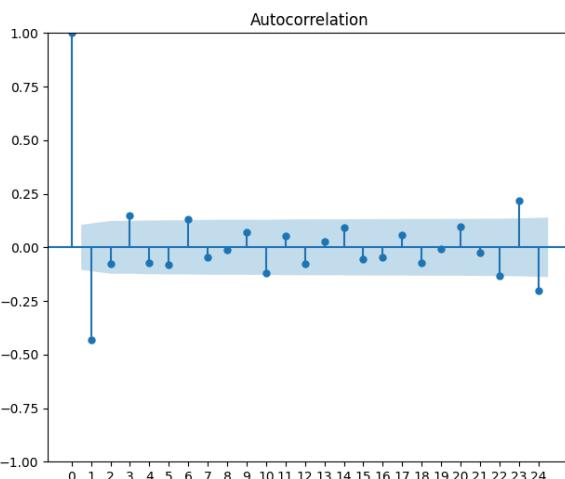
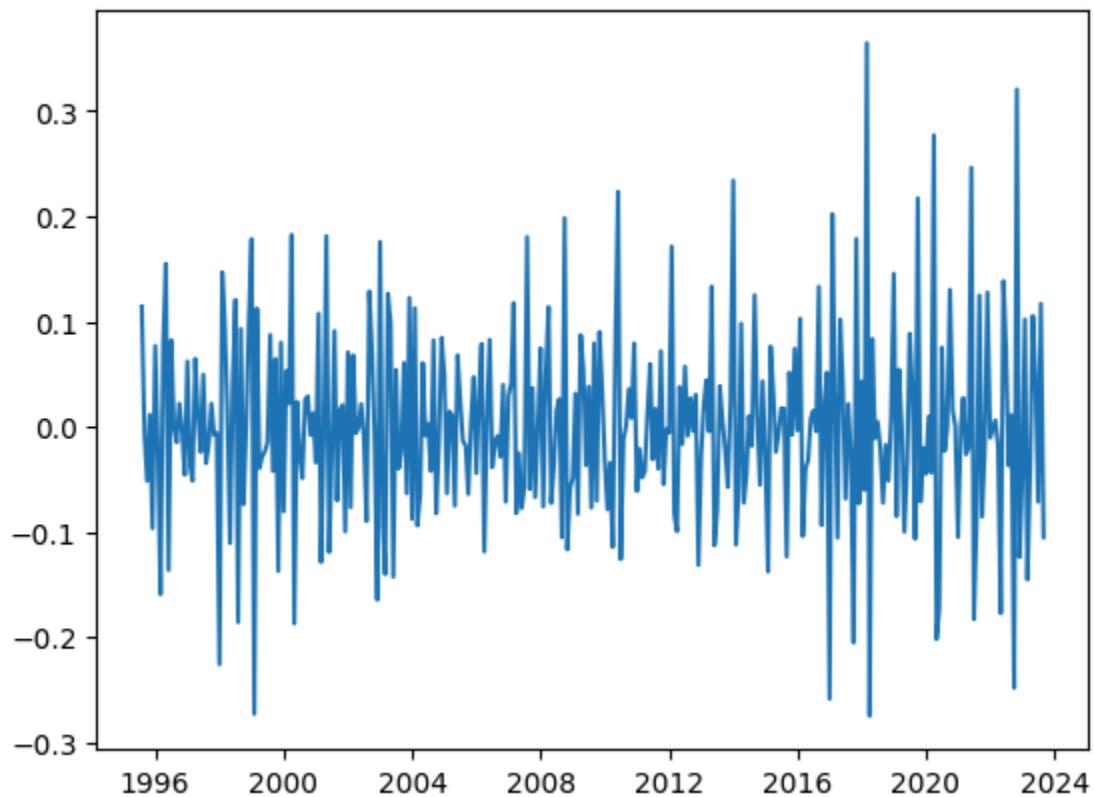
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals

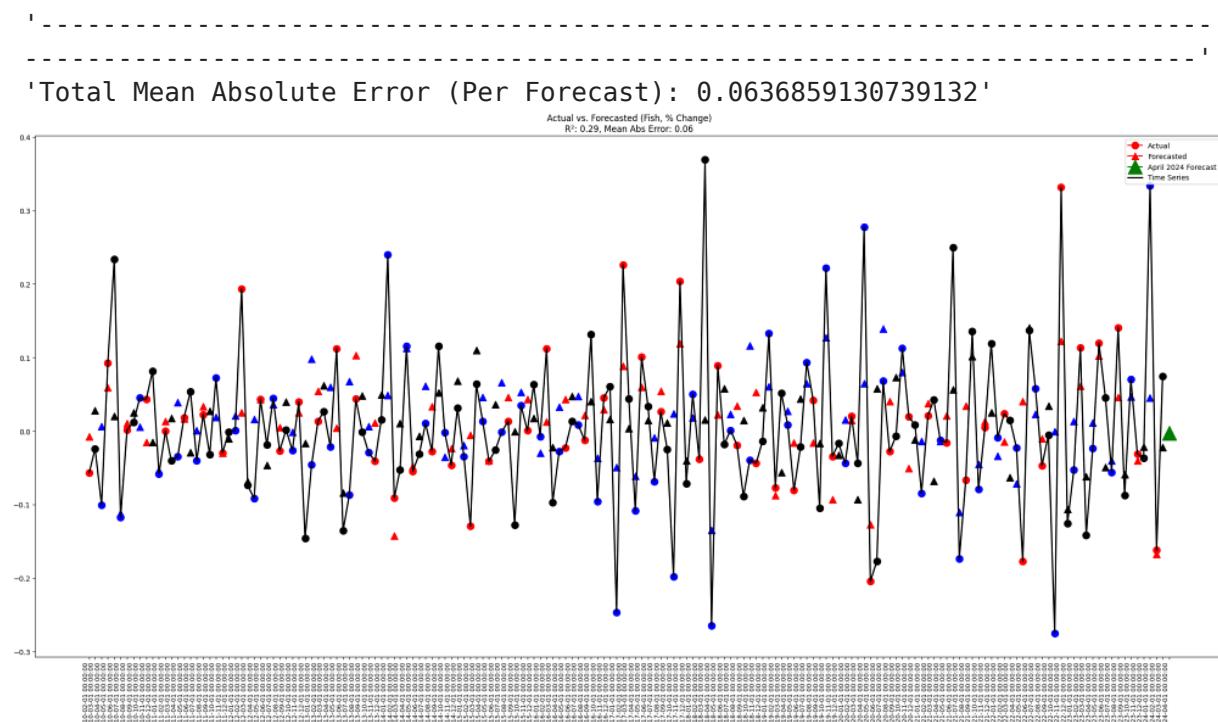


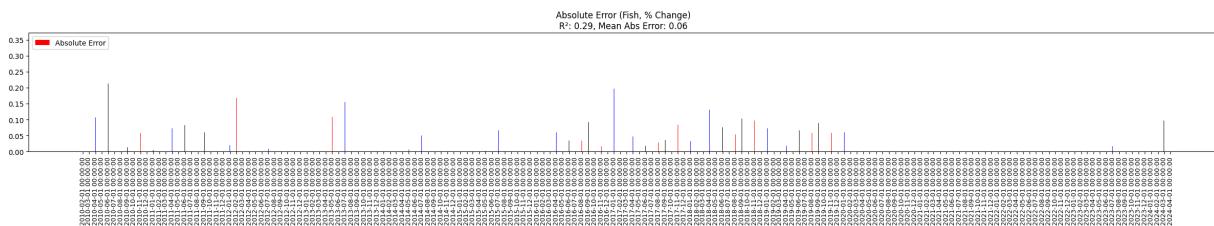
Time series assumptions are met.

'-----'  
-----'  
'Data of Actual vs Forecasted values with error metrics:'

| Fish       | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01 | -0.056565 | -0.007311 | -0.049254 | 0.049254  | 0.002426      | 0.870752             |
| 2010-03-01 | -0.024109 | 0.028270  | -0.052378 | 0.052378  | 0.002744      | 2.172606             |
| 2010-04-01 | -0.100628 | 0.006566  | -0.107194 | 0.107194  | 0.011491      | 1.065250             |
| 2010-05-01 | 0.092736  | 0.059416  | 0.033320  | 0.033320  | 0.001110      | 0.359300             |
| 2010-06-01 | 0.233805  | 0.020257  | 0.213549  | 0.213549  | 0.045603      | 0.913361             |
| ...        | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01 | -0.036721 | -0.021330 | -0.015391 | 0.015391  | 0.000237      | 0.419126             |
| 2024-01-01 | 0.333739  | 0.045307  | 0.288432  | 0.288432  | 0.083193      | 0.864243             |
| 2024-02-01 | -0.162100 | -0.167584 | 0.005484  | 0.005484  | 0.000030      | 0.033833             |
| 2024-03-01 | 0.074502  | -0.021913 | 0.096416  | 0.096416  | 0.009296      | 1.294126             |
| 2024-04-01 | NaN       | -0.002410 | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns





## Cuttlefish,\_squid,\_octopus

```
In [34]: df = df_export_ANALYSIS.copy()
name = df.columns[10]
display(f"Component: {name}")
```

'Component: Cuttlefish,\_squid,\_octopus'

```
In [35]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)
```

```
#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)
```

```
df_accuracy = pd.read_pickle(f'data/cleaned/SARIMA_RollWalkForward/combined/{name}_accuracy')
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)
```

```
#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Cuttlefish,_squid,_octopus |
|------------|----------------------------|
| 1995-02-01 | 0.037419                   |
| 1995-03-01 | 0.245854                   |
| 1995-04-01 | -0.270882                  |
| 1995-05-01 | 0.138749                   |
| 1995-06-01 | 0.063727                   |
| ...        | ...                        |
| 2023-11-01 | -0.295102                  |
| 2023-12-01 | 0.027215                   |
| 2024-01-01 | 0.360203                   |
| 2024-02-01 | -0.041442                  |
| 2024-03-01 | 0.336792                   |

350 rows × 1 columns

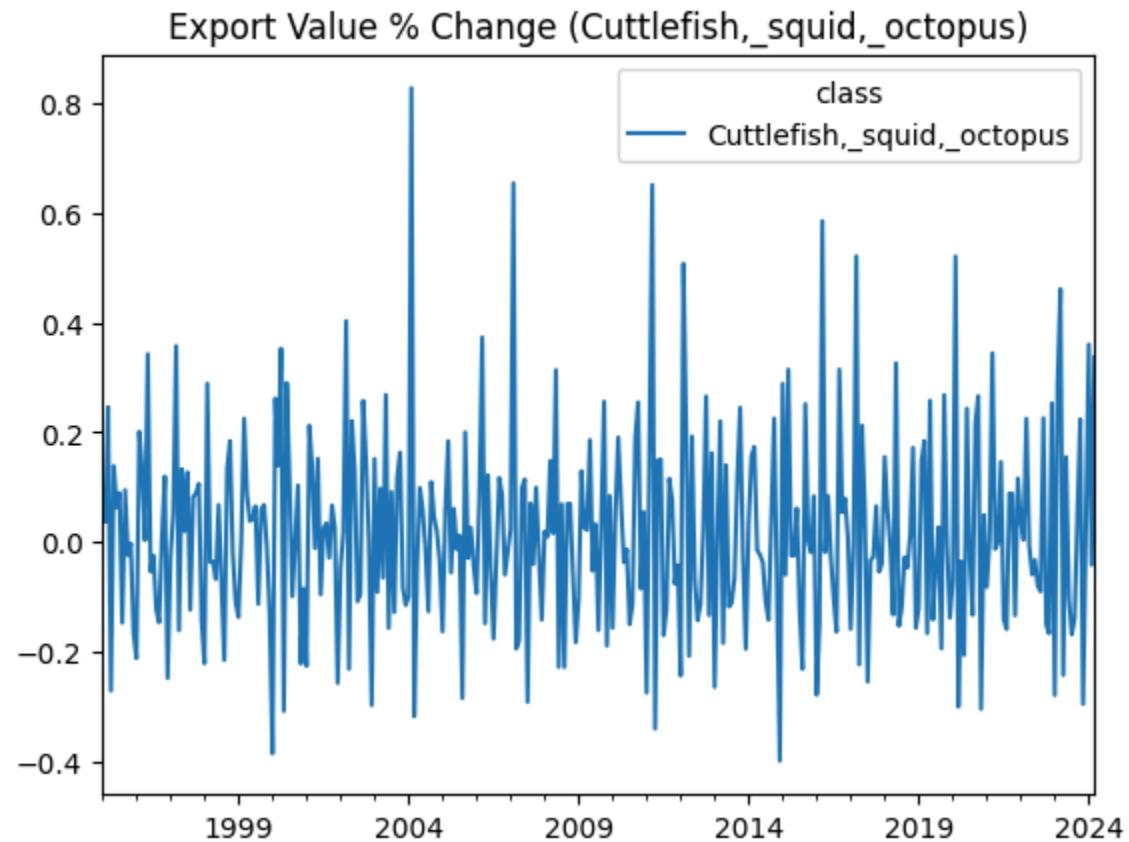
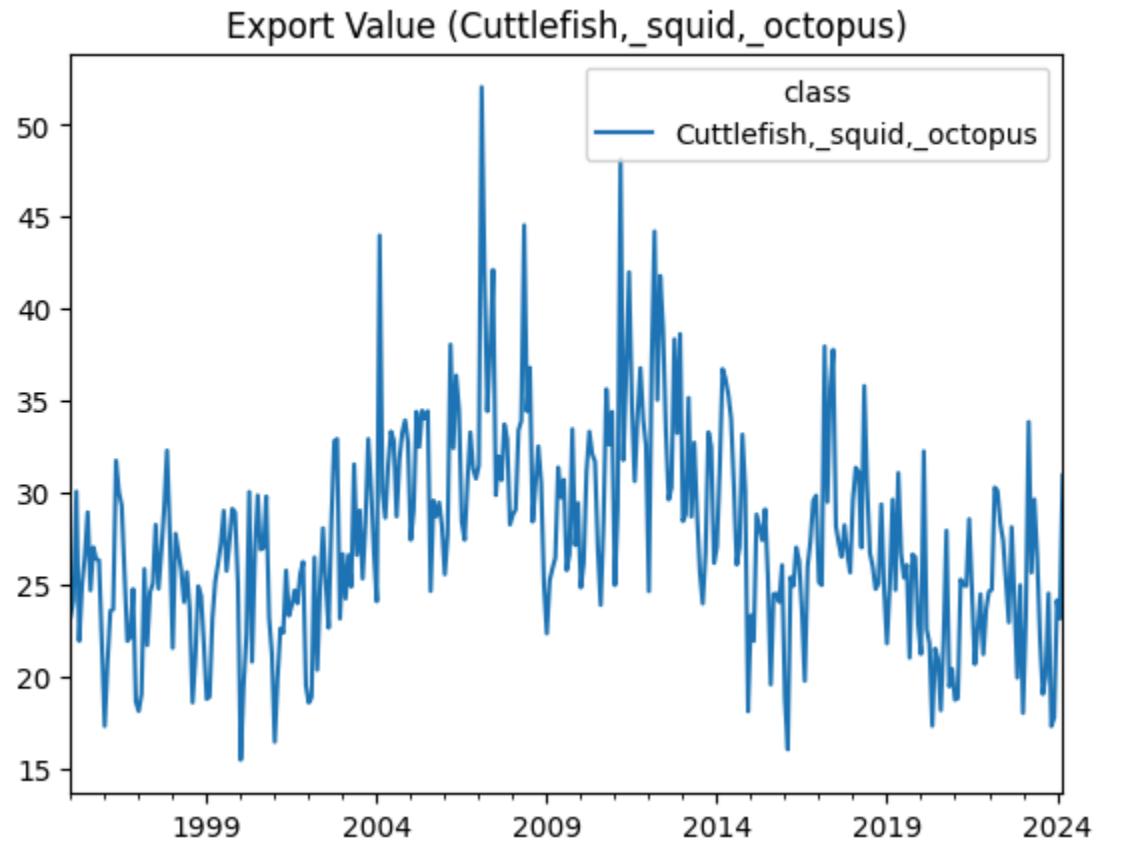
```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -4.795747973245559
p-value: 5.5413745255965345e-05
Critical Value 1%: -3.450141065277327
Critical Value 5%: -2.870258846235788
Critical Value 10%: -2.571415151457764
is_stationary: True
```

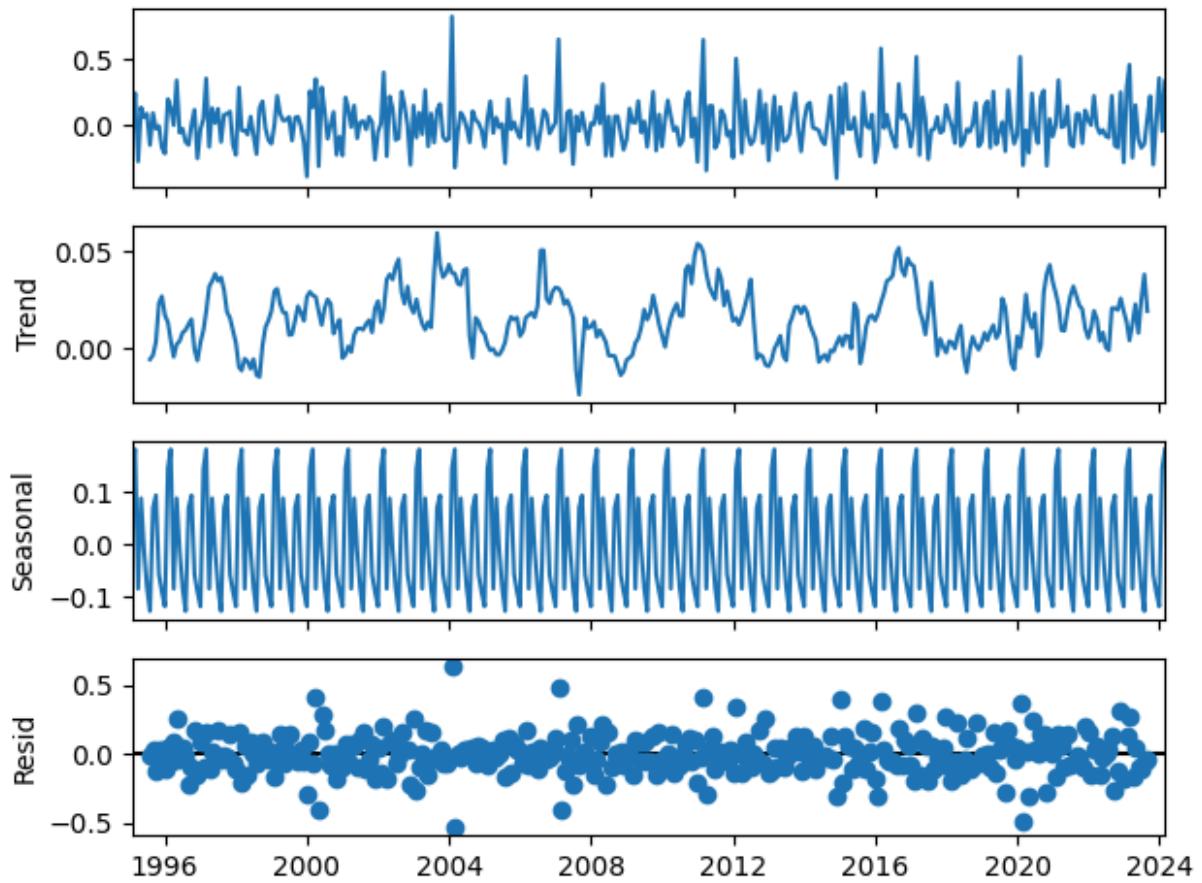
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.06761997750486669
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

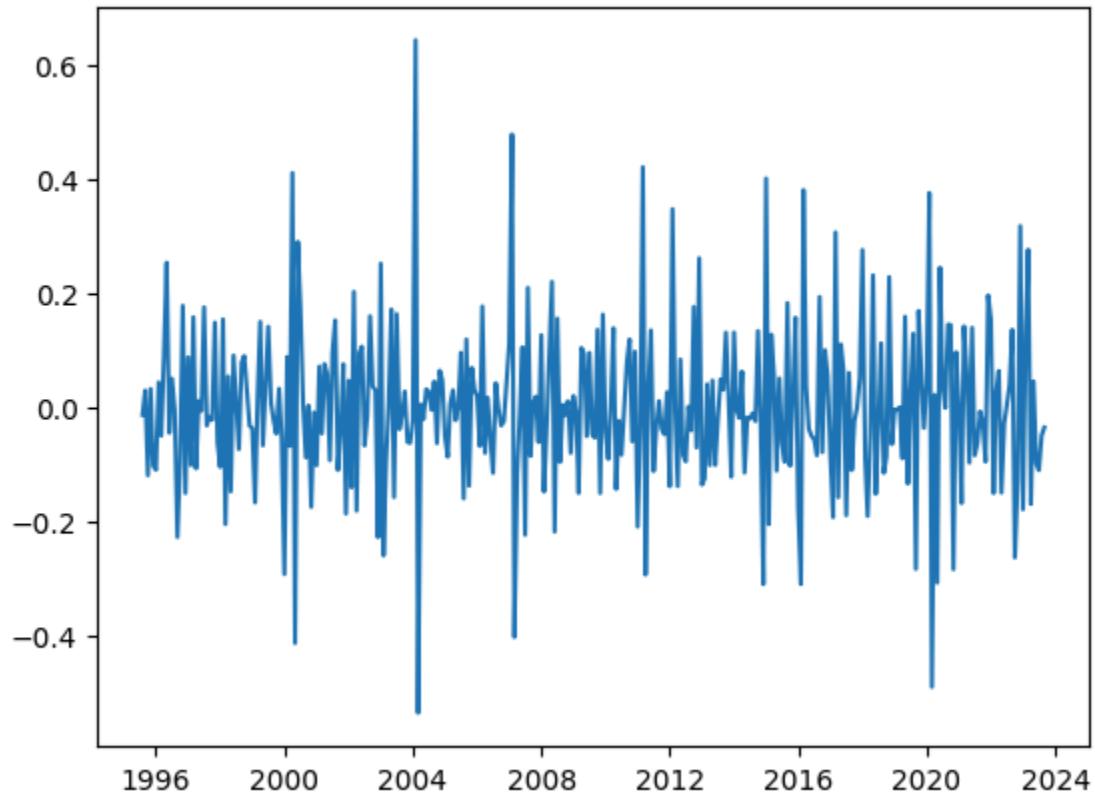
```
result = kpss(series, regression='c')
```

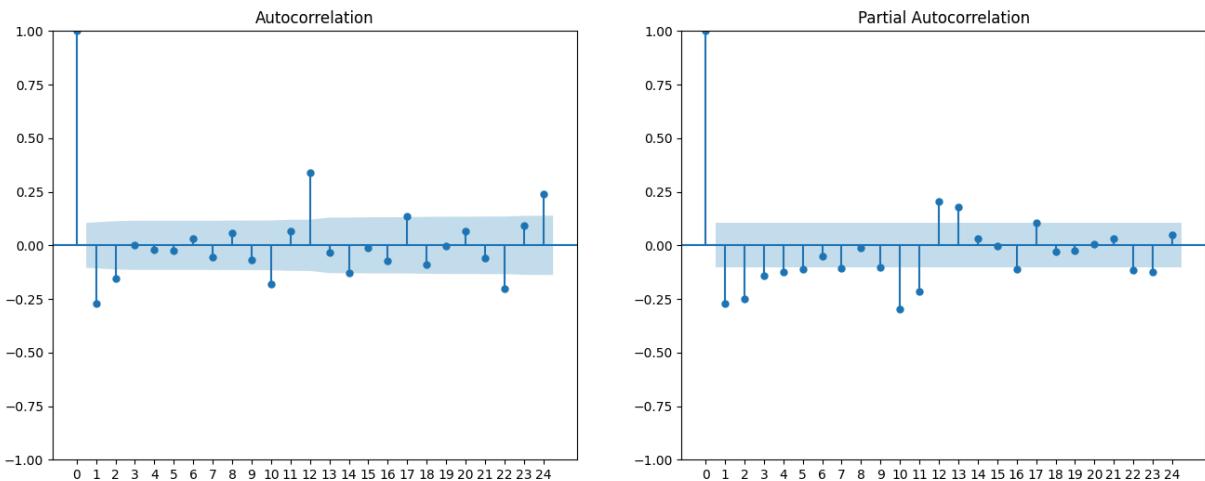




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

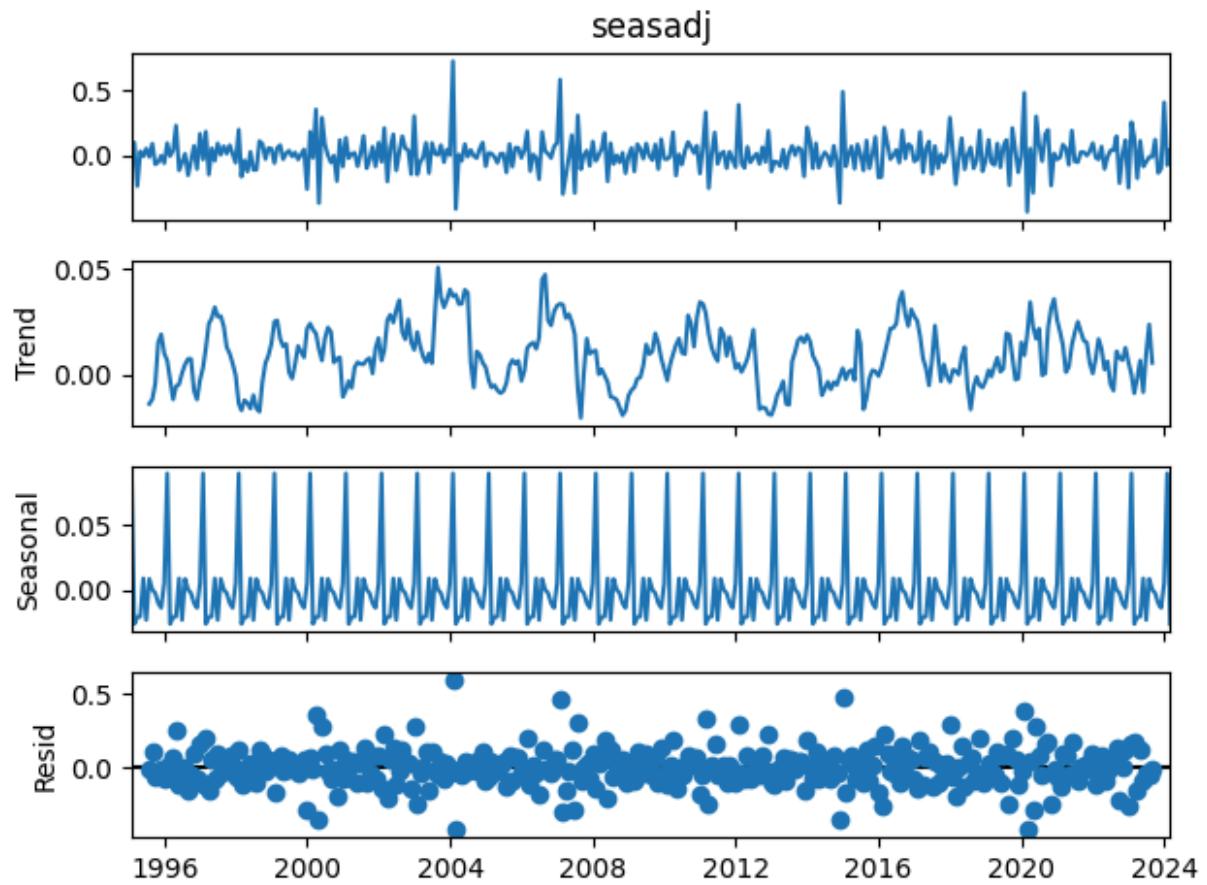
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -9.735153440499973
p-value: 8.790327711817594e-17
Critical Value 1%: -3.449559661646851
Critical Value 5%: -2.8700035112469626
Critical Value 10%: -2.5712790073013796
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.05256766076894533
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

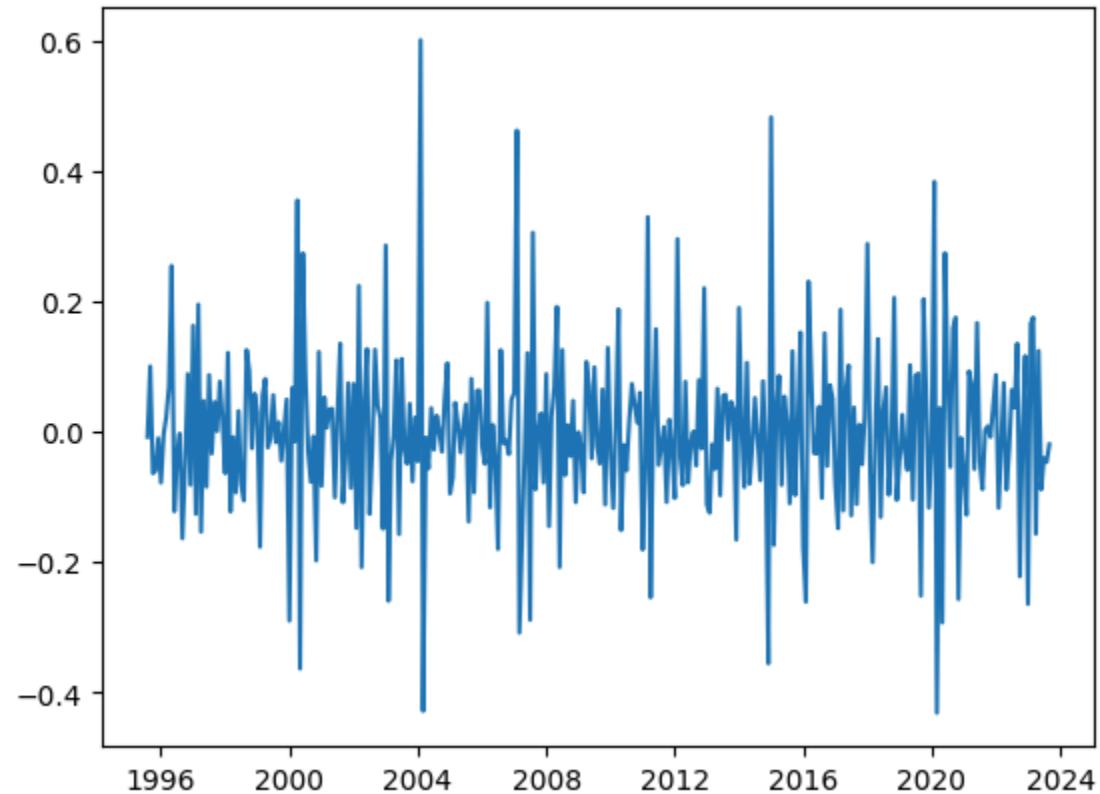
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

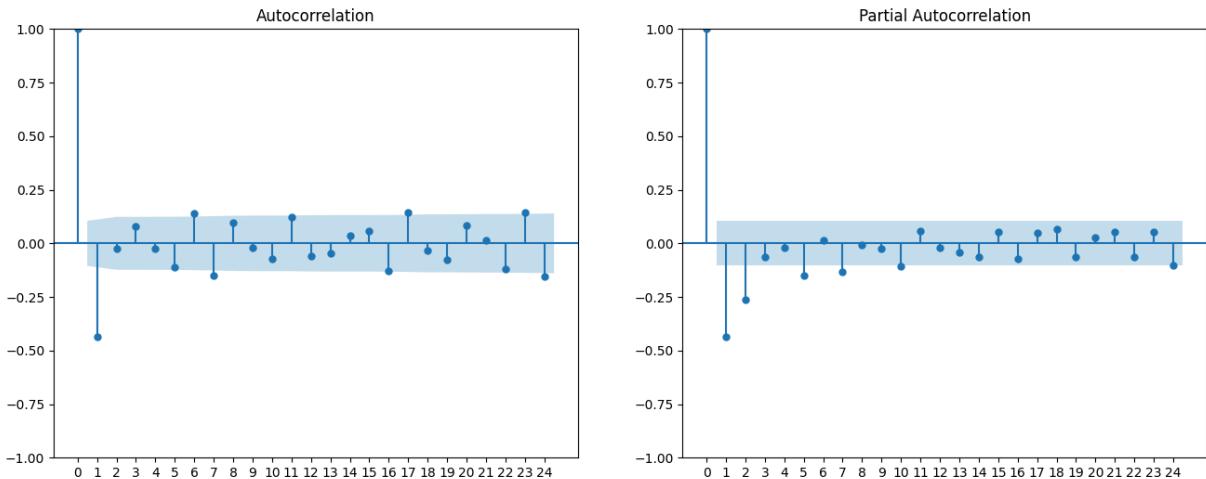
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals



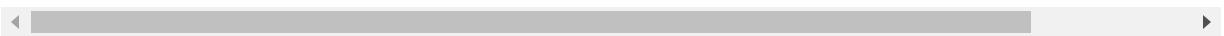


Time series assumptions are met.

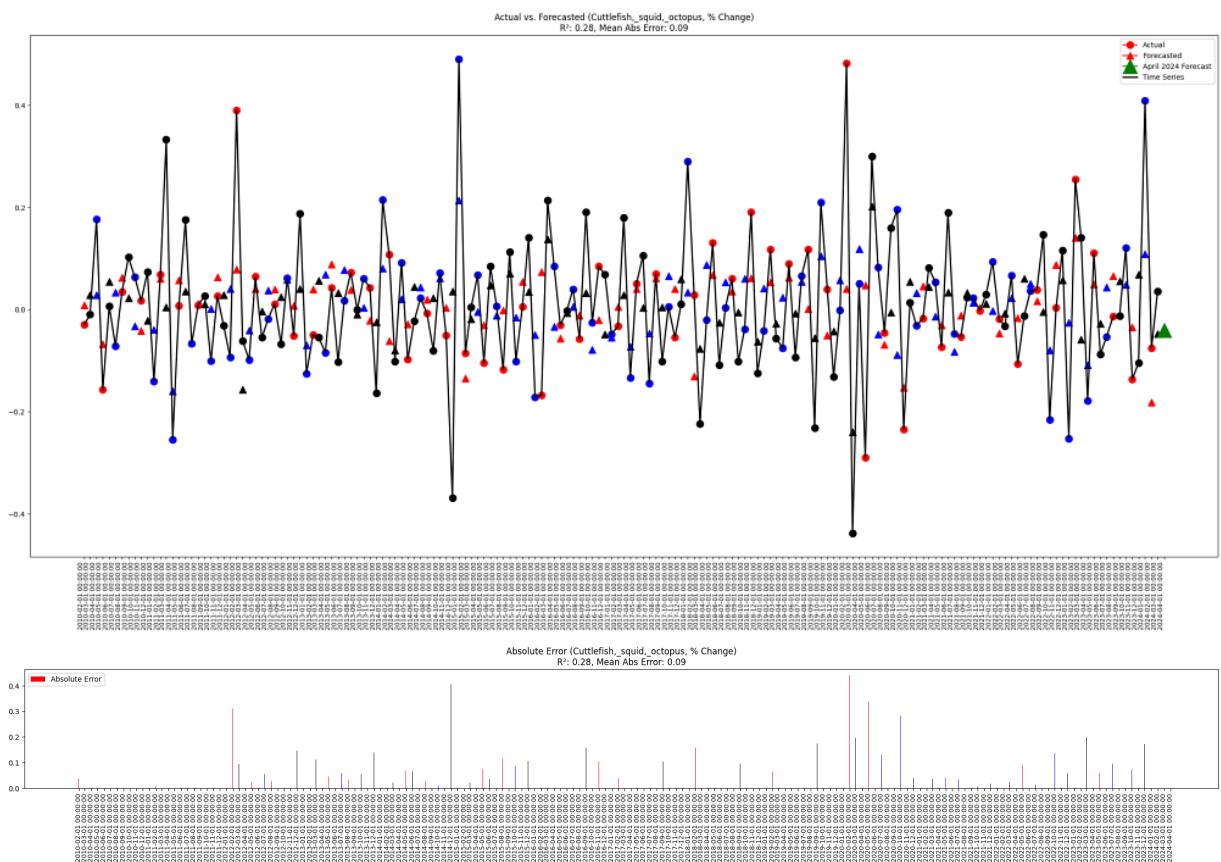
'Data of Actual vs Forecasted values with error metrics:'

| Cuttlefish,_squid,_octopus | actual    | forecast  | error     | abs_error | squared_error | abs_pe |
|----------------------------|-----------|-----------|-----------|-----------|---------------|--------|
| 2010-02-01                 | -0.029857 | 0.009036  | -0.038892 | 0.038892  | 0.001513      |        |
| 2010-03-01                 | -0.009797 | 0.028602  | -0.038399 | 0.038399  | 0.001475      |        |
| 2010-04-01                 | 0.177119  | 0.028230  | 0.148889  | 0.148889  | 0.022168      |        |
| 2010-05-01                 | -0.156944 | -0.067557 | -0.089387 | 0.089387  | 0.007990      |        |
| 2010-06-01                 | 0.006408  | 0.054832  | -0.048424 | 0.048424  | 0.002345      |        |
| ...                        | ...       | ...       | ...       | ...       | ...           | ...    |
| 2023-12-01                 | -0.104408 | 0.068434  | -0.172842 | 0.172842  | 0.029874      |        |
| 2024-01-01                 | 0.408740  | 0.109219  | 0.299521  | 0.299521  | 0.089713      |        |
| 2024-02-01                 | -0.075920 | -0.181801 | 0.105881  | 0.105881  | 0.011211      |        |
| 2024-03-01                 | 0.035896  | -0.047229 | 0.083125  | 0.083125  | 0.006910      |        |
| 2024-04-01                 | NaN       | -0.040649 | NaN       | NaN       | NaN           |        |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.08908194881866648'



## Fishery\_products,\_n.i.e.

```
In [36]: df = df_export_ANALYSIS.copy()
name = df.columns[11]
display(f"Component: {name}")

'Component: Fishery_products,_n.i.e.'
```

```
In [37]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value')
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

class Fishery\_products,\_n.i.e.

|            |           |
|------------|-----------|
| 1995-02-01 | -0.005405 |
| 1995-03-01 | 0.293478  |
| 1995-04-01 | -0.004202 |
| 1995-05-01 | -0.206751 |
| 1995-06-01 | -0.034574 |
| ...        | ...       |
| 2023-11-01 | 0.113466  |
| 2023-12-01 | -0.105263 |
| 2024-01-01 | -0.068836 |
| 2024-02-01 | -0.086022 |
| 2024-03-01 | 0.338235  |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

```
Running ADF Test for Stationarity...
ADF Statistic: -9.771406959070204
p-value: 7.119945466444371e-17
Critical Value 1%: -3.4496162602188187
Critical Value 5%: -2.870028369720798
Critical Value 10%: -2.5712922615505627
is_stationary: True
```

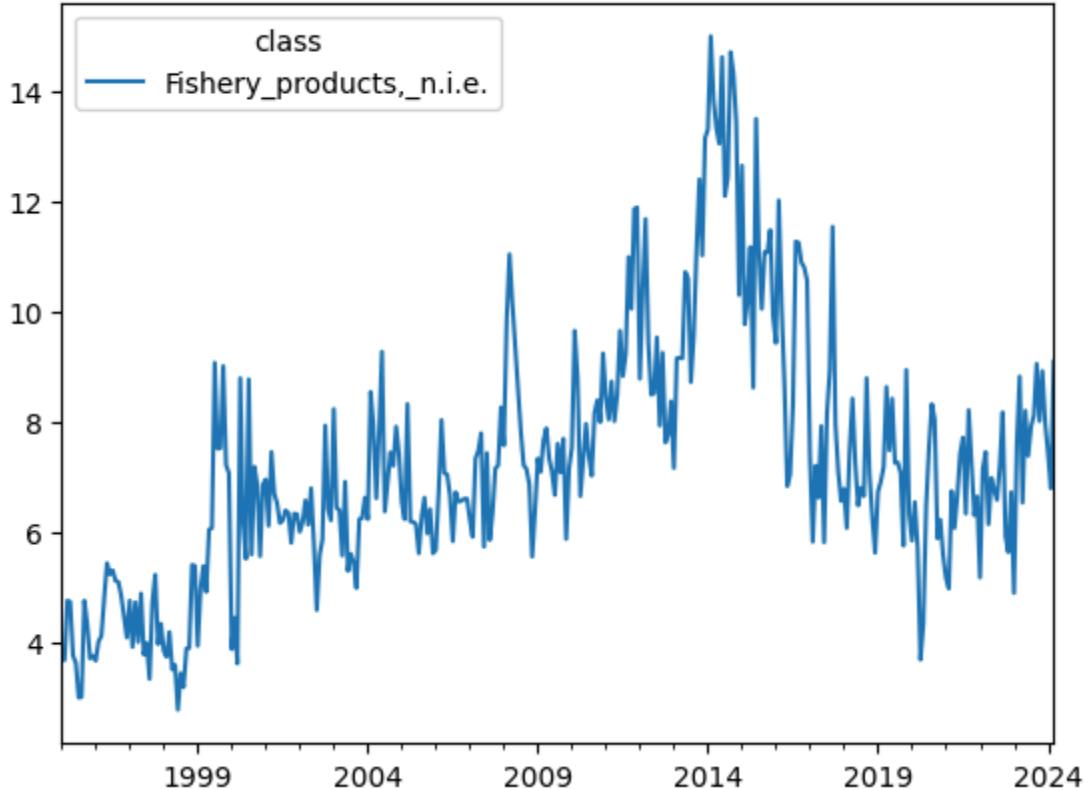
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.11672428412169358
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

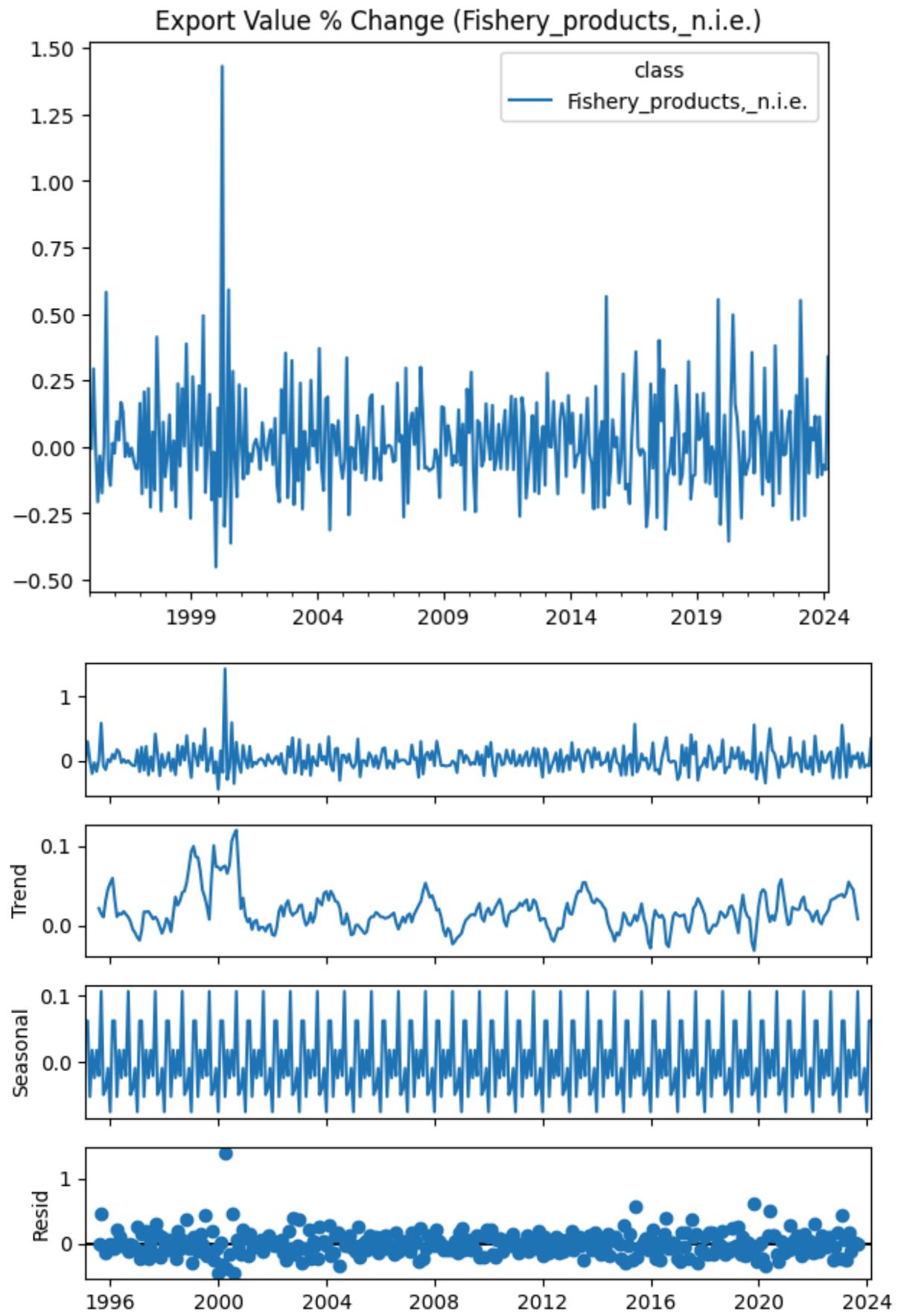
Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

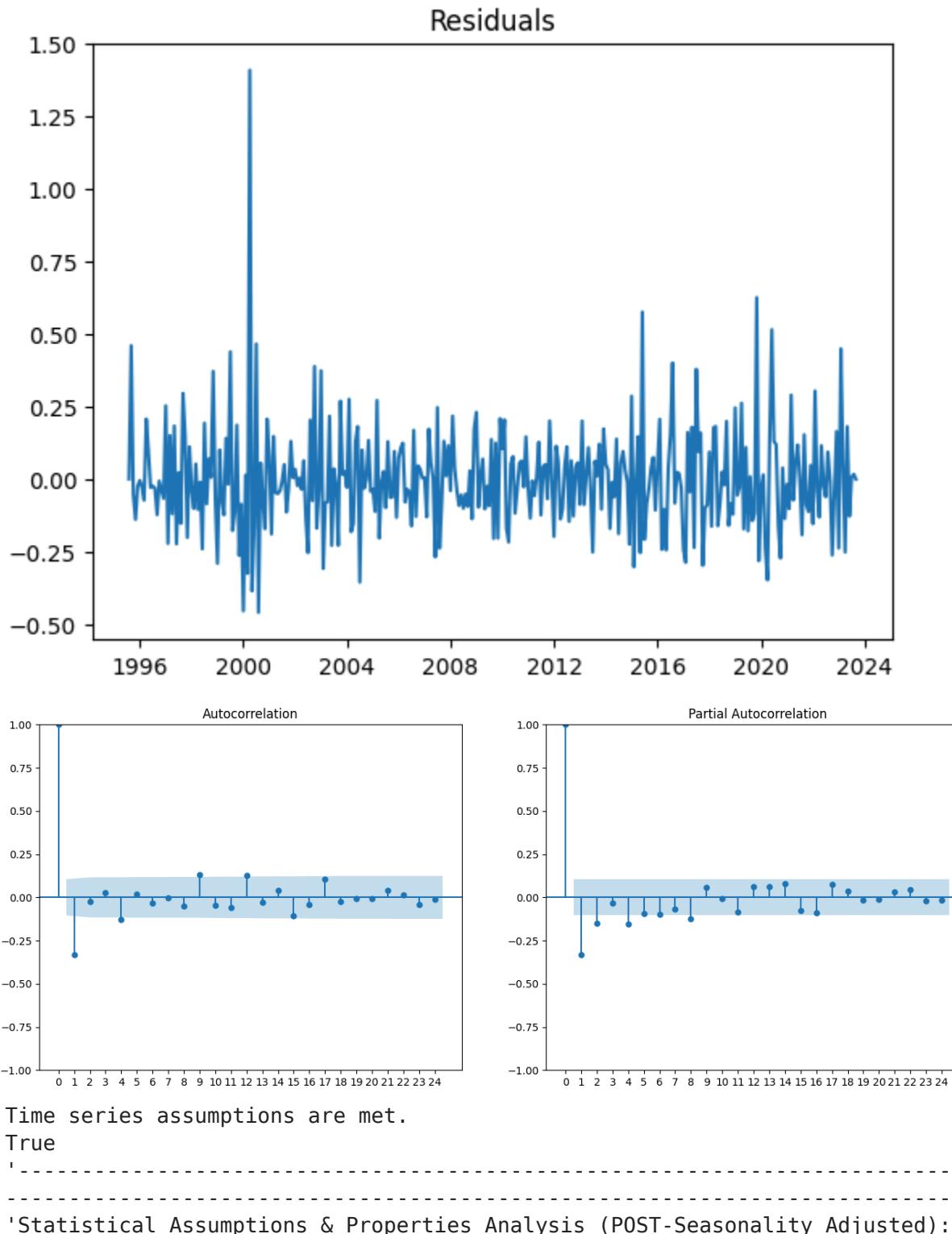
```
result = kpss(series, regression='c')
```

Export Value (Fishery\_products,\_n.i.e.)





Plotting Residuals for Homoscedasticity Check...



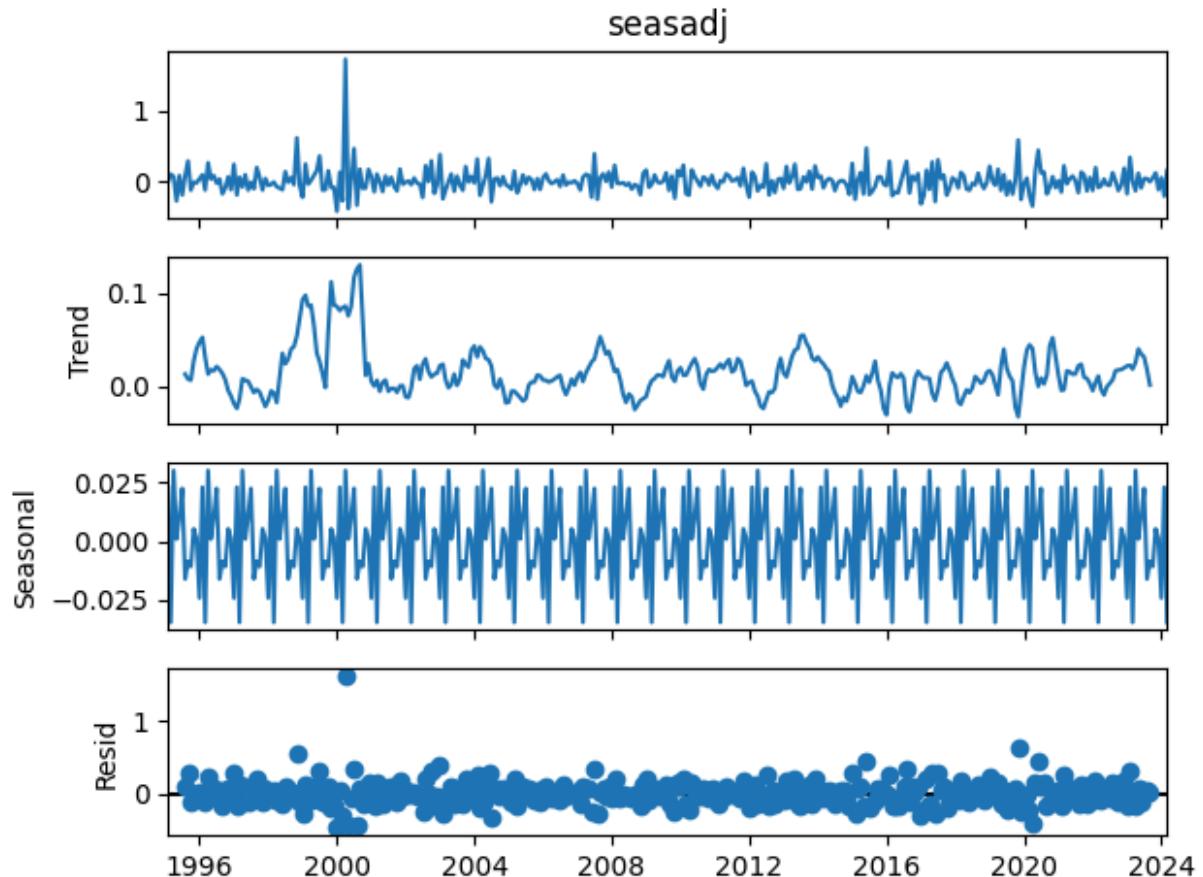
Running ADF Test for Stationarity...  
 ADF Statistic: -4.697726154550245  
 p-value: 8.498436250955315e-05  
 Critical Value 1%: -3.450141065277327  
 Critical Value 5%: -2.870258846235788  
 Critical Value 10%: -2.571415151457764  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.14842654209644202  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

Decomposing the Series...

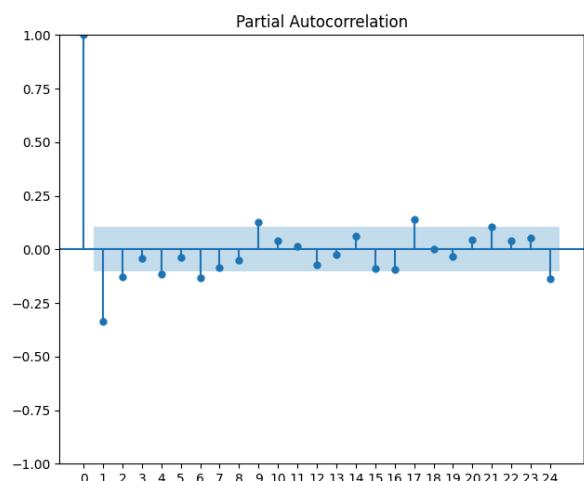
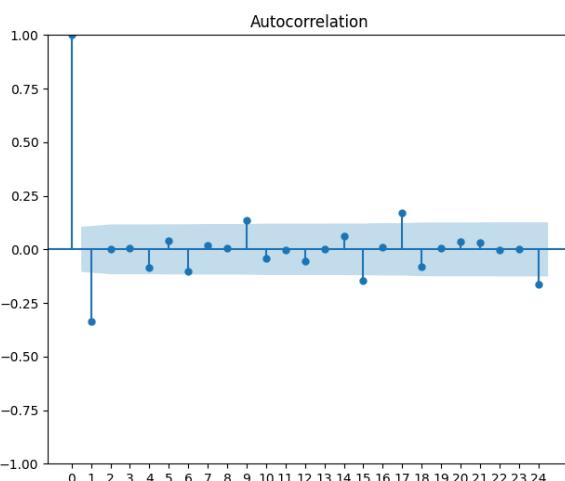
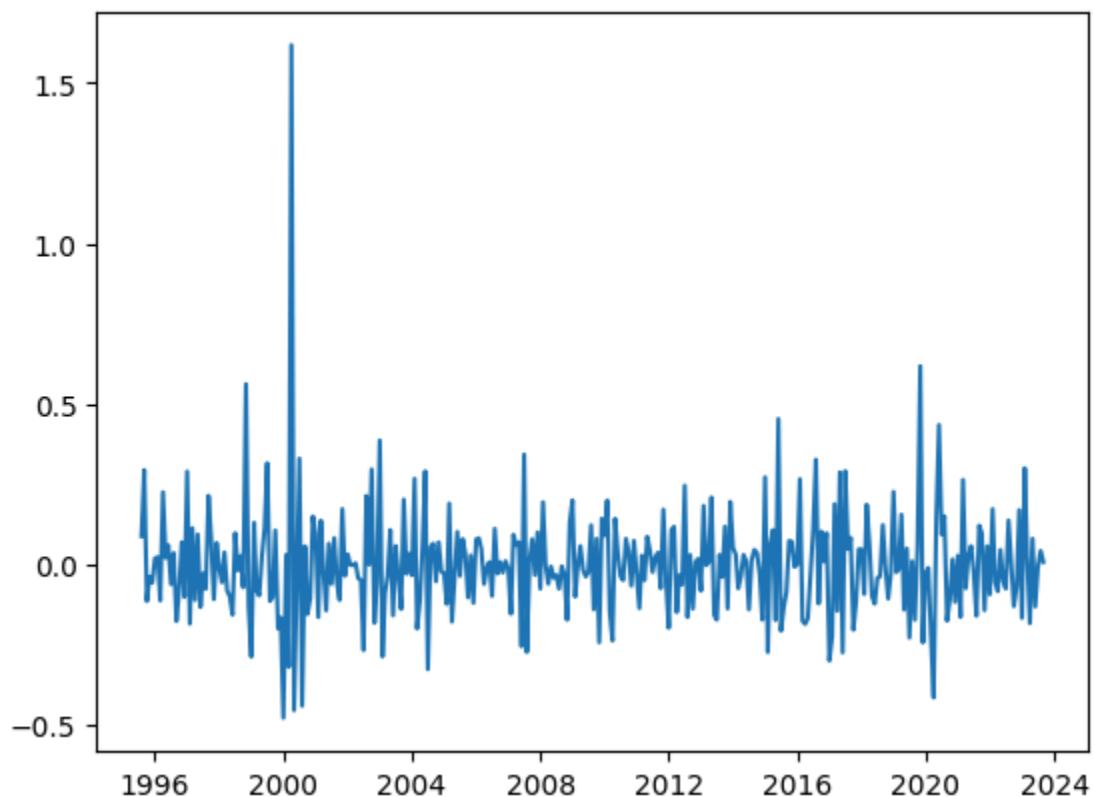
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals



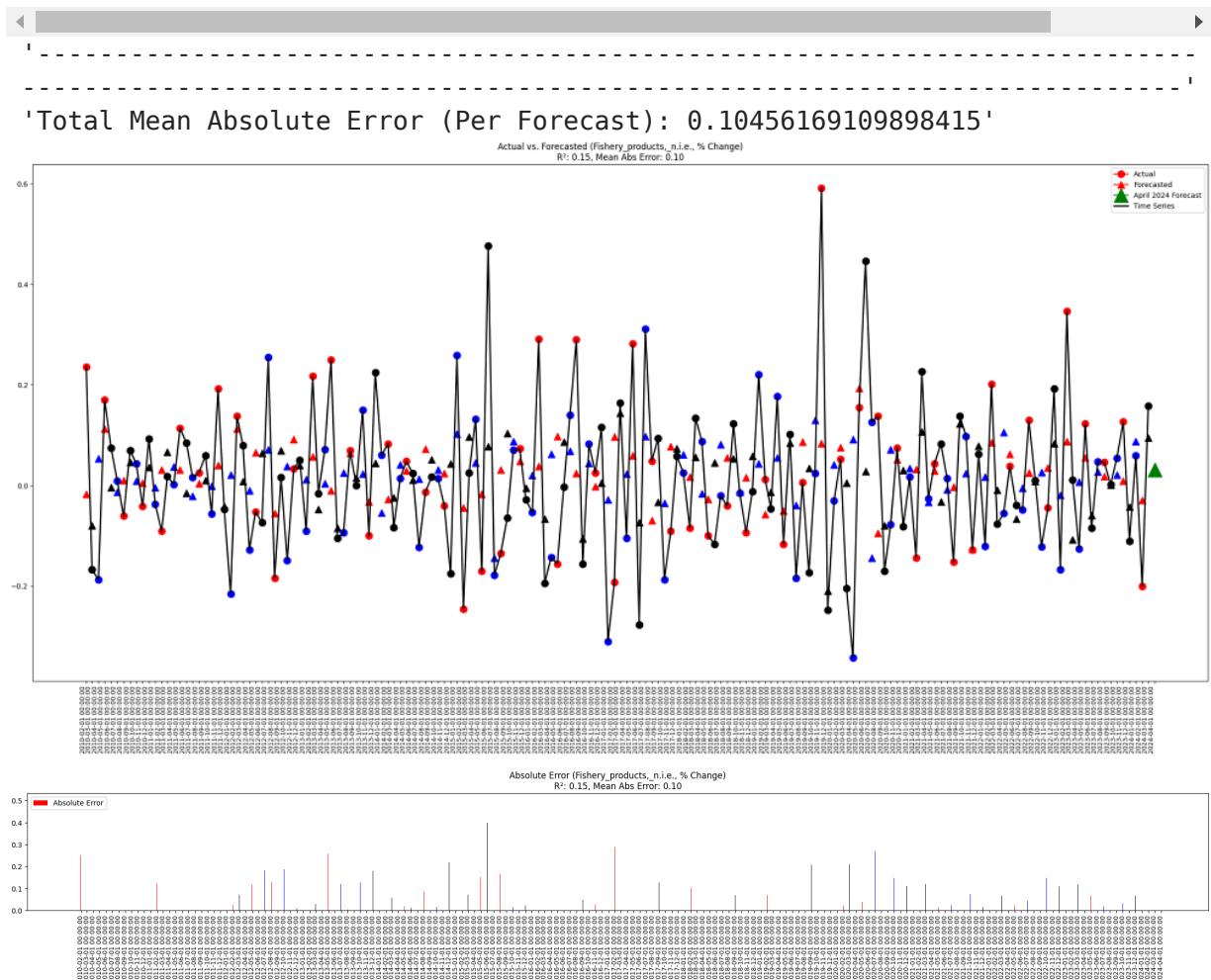
Time series assumptions are met.

'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Fishery_products,_n.i.e. | actual    | forecast  | error     | abs_error | squared_error | abs_perce |
|--------------------------|-----------|-----------|-----------|-----------|---------------|-----------|
| 2010-02-01               | 0.235853  | -0.016770 | 0.252623  | 0.252623  | 0.063818      |           |
| 2010-03-01               | -0.167285 | -0.079138 | -0.088147 | 0.088147  | 0.007770      |           |
| 2010-04-01               | -0.187089 | 0.053025  | -0.240114 | 0.240114  | 0.057655      |           |
| 2010-05-01               | 0.169996  | 0.112464  | 0.057533  | 0.057533  | 0.003310      |           |
| 2010-06-01               | 0.075012  | -0.004381 | 0.079393  | 0.079393  | 0.006303      |           |
| ...                      | ...       | ...       | ...       | ...       | ...           | ...       |
| 2023-12-01               | -0.110615 | -0.042240 | -0.068375 | 0.068375  | 0.004675      |           |
| 2024-01-01               | 0.059244  | 0.087515  | -0.028271 | 0.028271  | 0.000799      |           |
| 2024-02-01               | -0.200260 | -0.029094 | -0.171166 | 0.171166  | 0.029298      |           |
| 2024-03-01               | 0.157711  | 0.094587  | 0.063124  | 0.063124  | 0.003985      |           |
| 2024-04-01               | NaN       | 0.030929  | NaN       | NaN       | NaN           | NaN       |

171 rows × 6 columns



## Forestry

```
In [38]: df = df_export_ANALYSIS.copy()
name = df.columns[12]
display(f"Component: {name}")
```

'Component: Forestry'

```
In [39]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

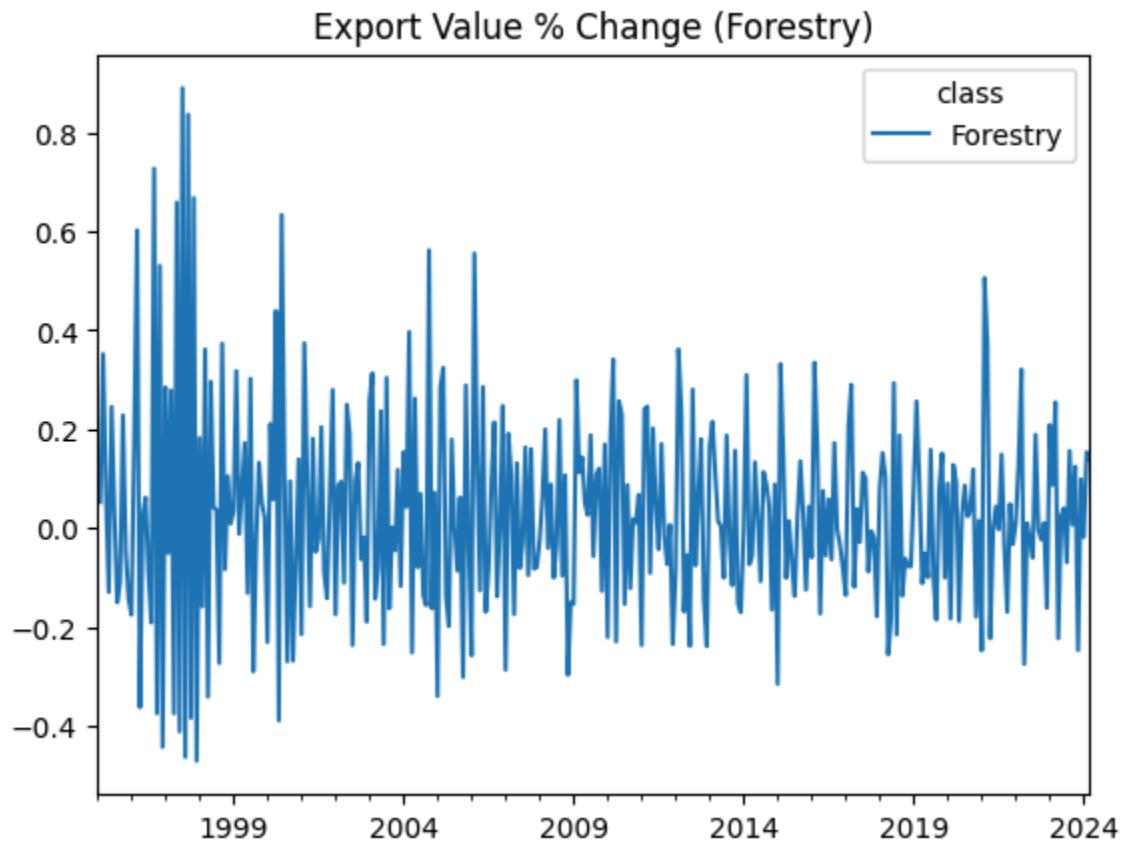
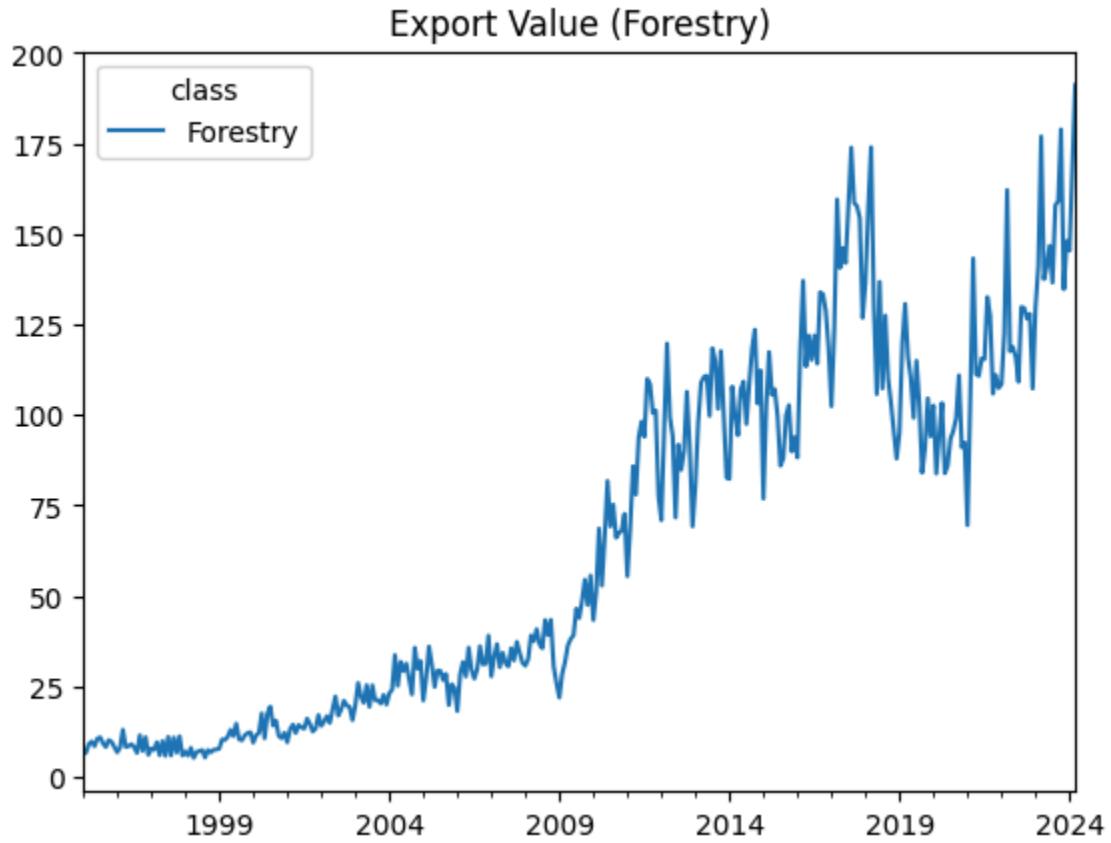
| class      | Forestry  |
|------------|-----------|
| 1995-02-01 | 0.052960  |
| 1995-03-01 | 0.352071  |
| 1995-04-01 | 0.075492  |
| 1995-05-01 | -0.129196 |
| 1995-06-01 | 0.245327  |
| ...        | ...       |
| 2023-11-01 | -0.246631 |
| 2023-12-01 | 0.098709  |
| 2024-01-01 | -0.017901 |
| 2024-02-01 | 0.153724  |
| 2024-03-01 | 0.139263  |

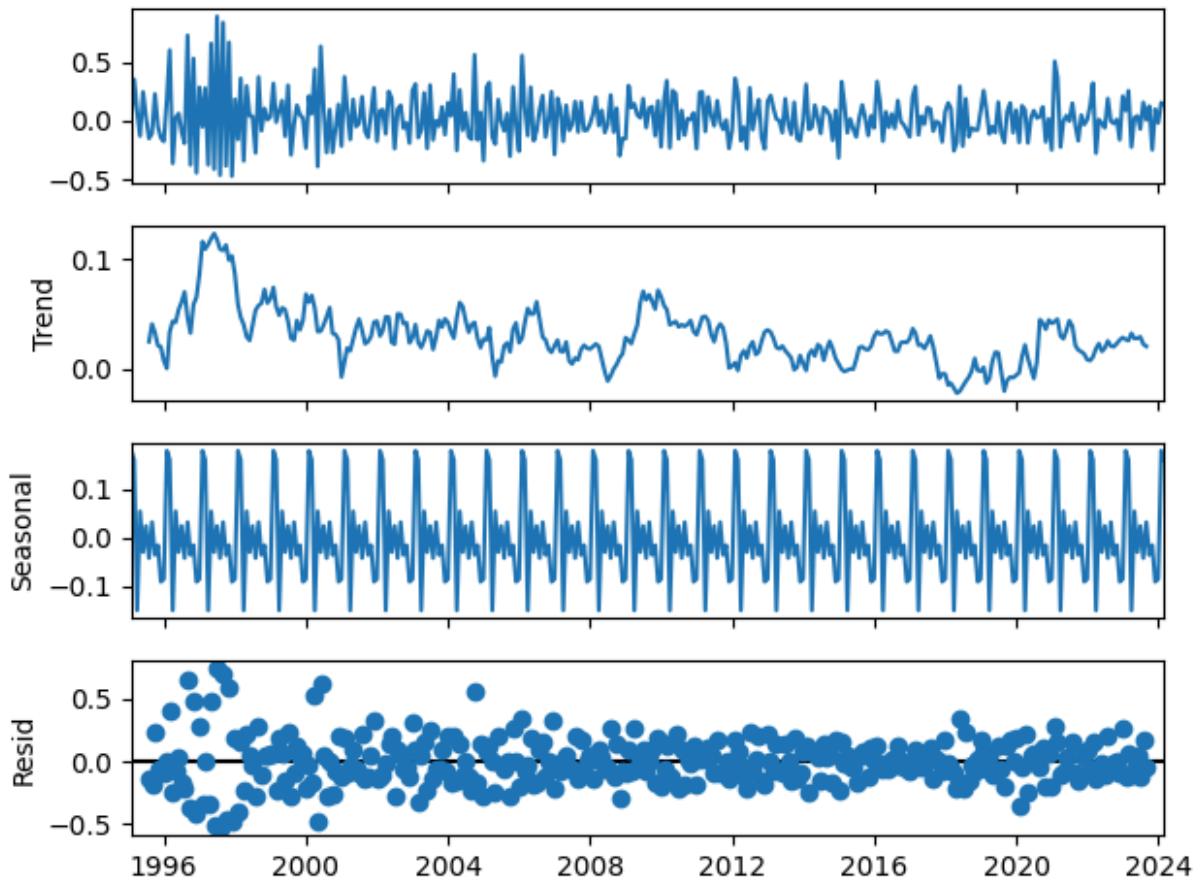
350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -4.209446499863254
p-value: 0.0006350553235352651
Critical Value 1%: -3.4499043309021955
Critical Value 5%: -2.870154885338925
Critical Value 10%: -2.5713597190254385
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.35200487953005916
p-value: 0.09784272434049174
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

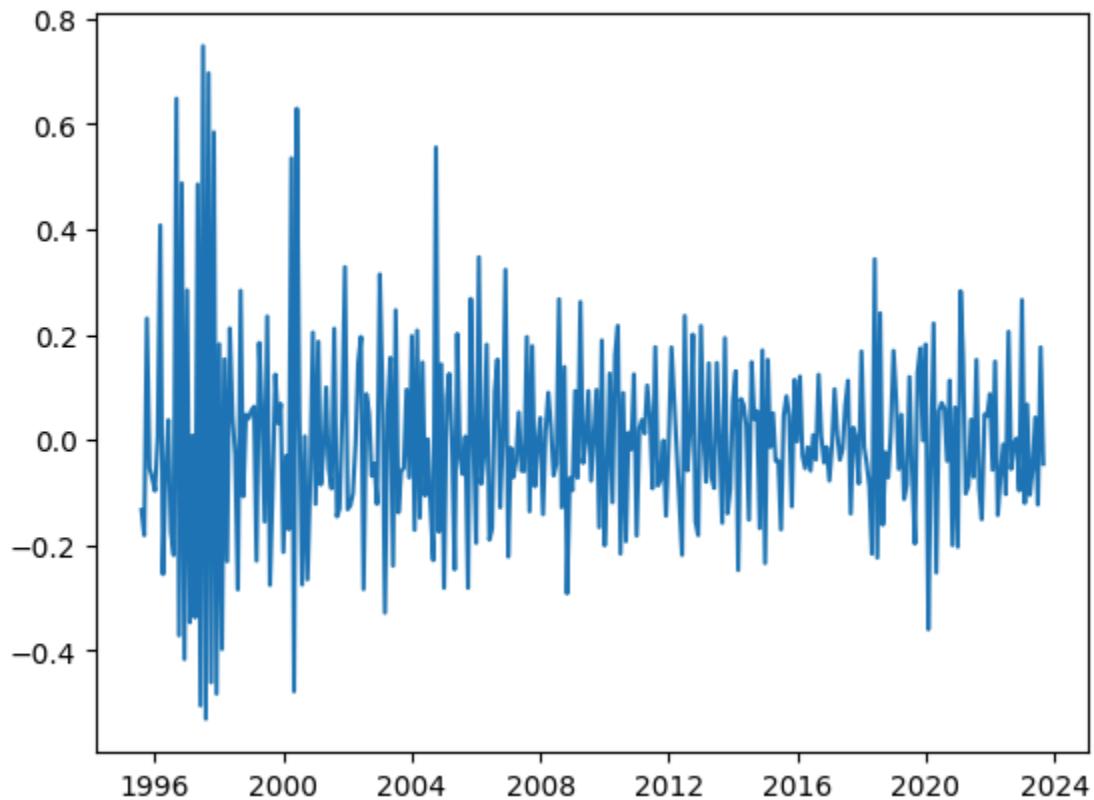
Decomposing the Series...

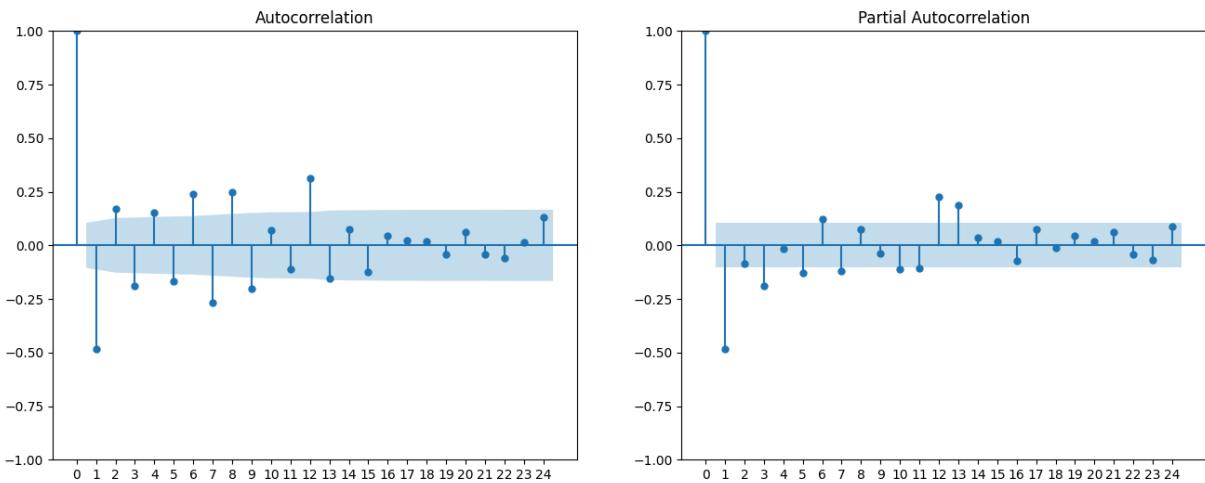




Plotting Residuals for Homoscedasticity Check...

Residuals





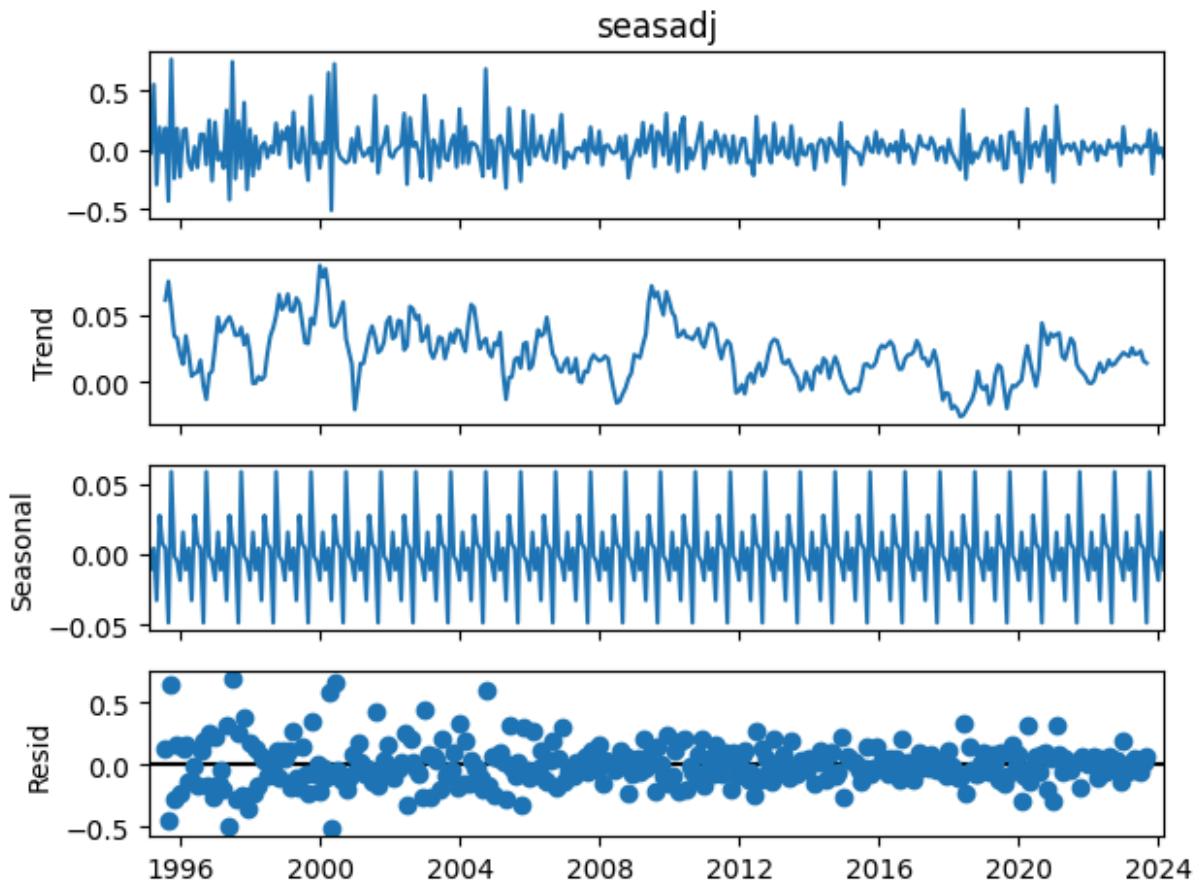
Time series assumptions are met.

True

```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -8.208757144176662
p-value: 6.909461237053829e-13
Critical Value 1%: -3.449559661646851
Critical Value 5%: -2.8700035112469626
Critical Value 10%: -2.5712790073013796
is_stationary: True
```

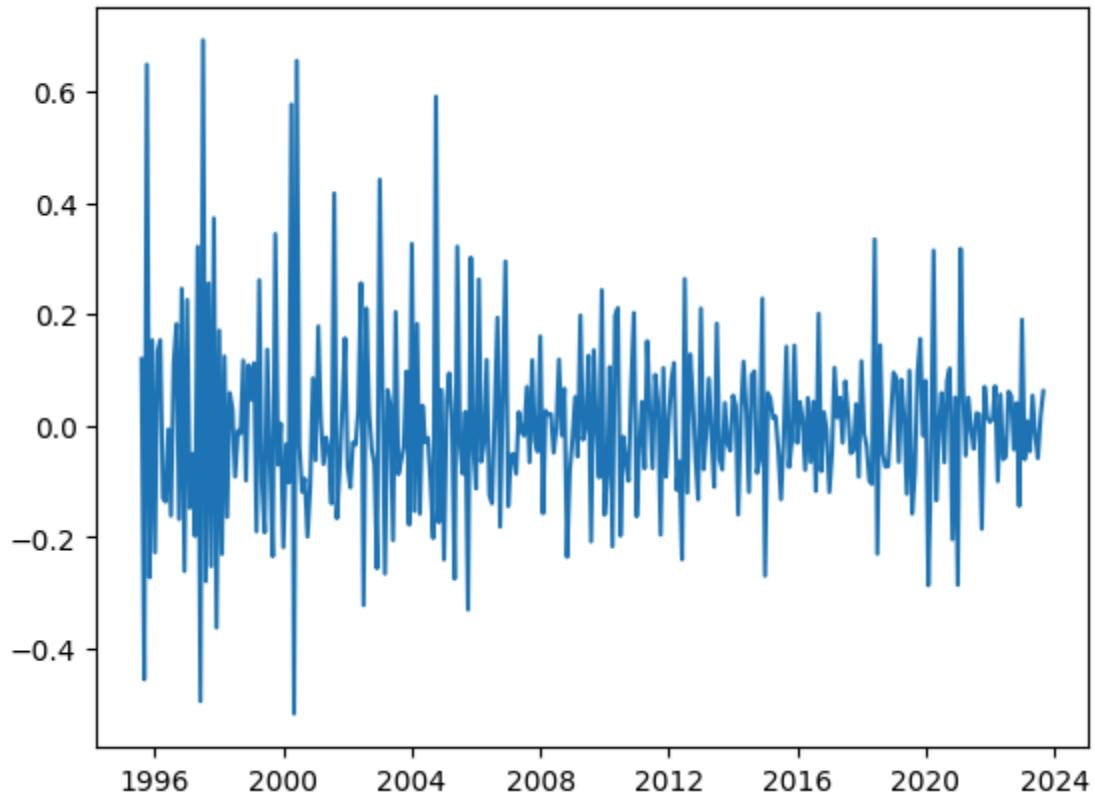
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.6477574798135671
p-value: 0.018294774562402985
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: False
```

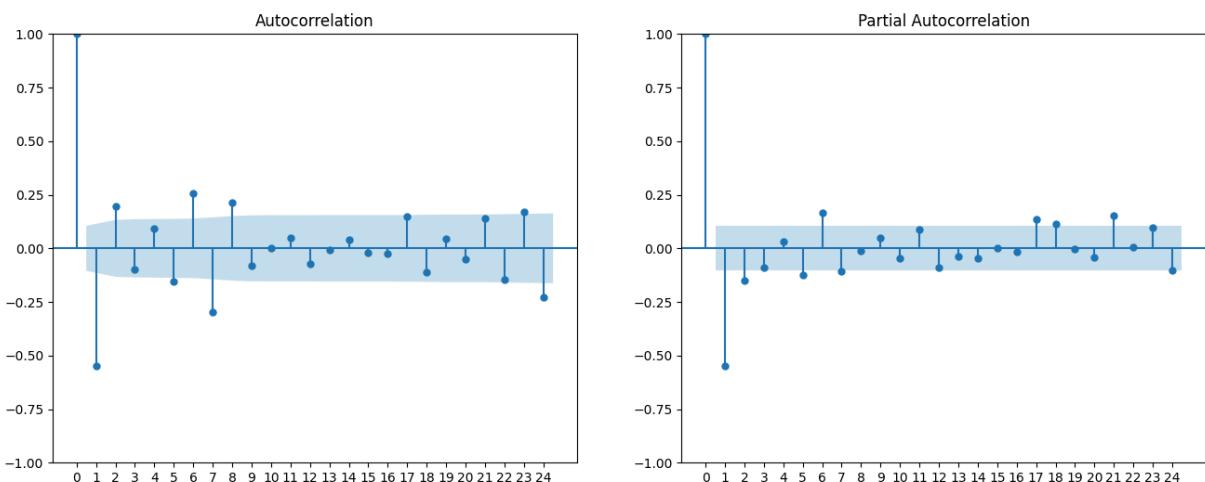
Decomposing the Series...



Plotting Residuals for Homoscedasticity Check...

Residuals





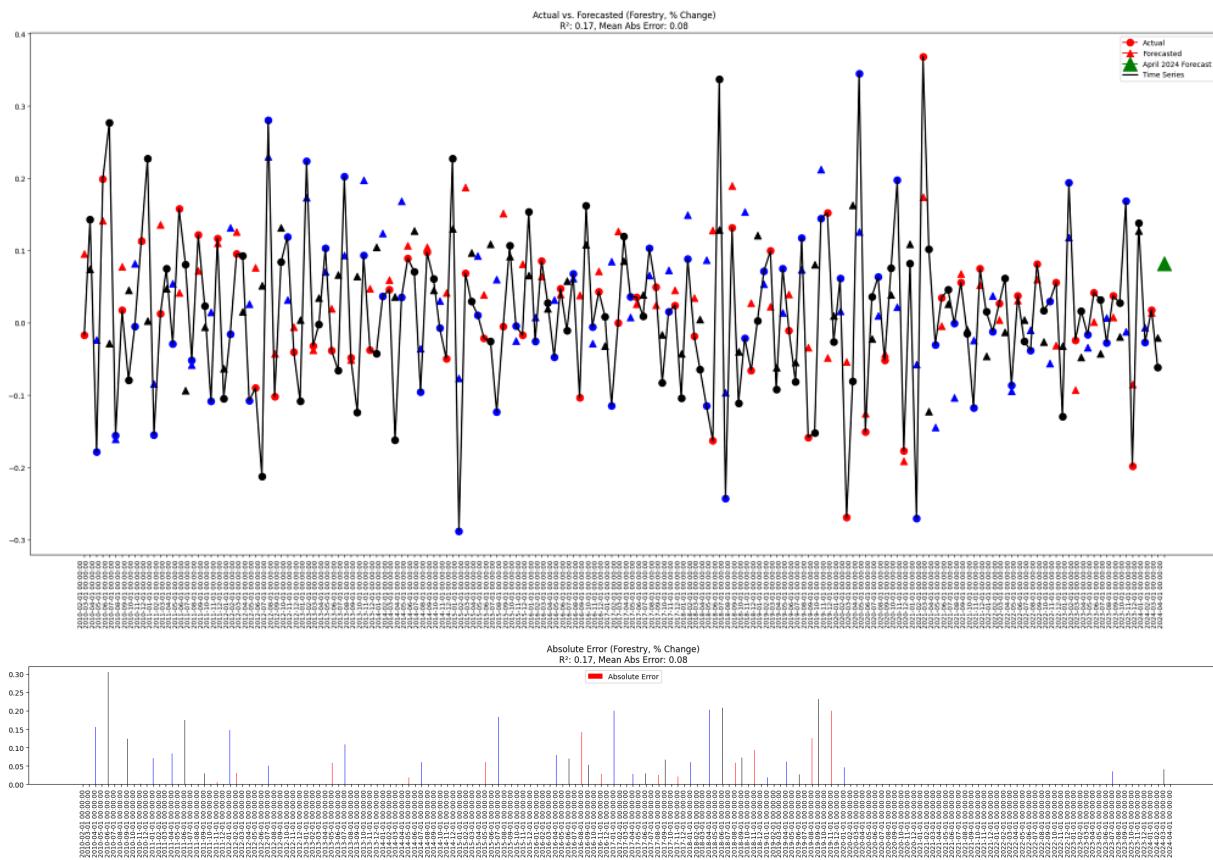
Time series assumptions are not met. Further investigation needed.

'Data of Actual vs Forecasted values with error metrics:'

| Forestry   | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01 | -0.016861 | 0.095440  | -0.112300 | 0.112300  | 0.012611      | 6.660509             |
| 2010-03-01 | 0.142911  | 0.074020  | 0.068891  | 0.068891  | 0.004746      | 0.482056             |
| 2010-04-01 | -0.178647 | -0.023354 | -0.155294 | 0.155294  | 0.024116      | 0.869276             |
| 2010-05-01 | 0.199009  | 0.141460  | 0.057549  | 0.057549  | 0.003312      | 0.289177             |
| 2010-06-01 | 0.277159  | -0.028081 | 0.305240  | 0.305240  | 0.093171      | 1.101317             |
| ...        | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01 | 0.137918  | 0.127293  | 0.010625  | 0.010625  | 0.000113      | 0.077038             |
| 2024-01-01 | -0.026927 | -0.006634 | -0.020293 | 0.020293  | 0.000412      | 0.753614             |
| 2024-02-01 | 0.017832  | 0.014420  | 0.003412  | 0.003412  | 0.000012      | 0.191345             |
| 2024-03-01 | -0.061354 | -0.020840 | -0.040514 | 0.040514  | 0.001641      | 0.660326             |
| 2024-04-01 | NaN       | 0.081986  | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns

'Total Mean Absolute Error (Per Forecast): 0.07999158350535383'



## Mining

```
In [40]: df = df_export_ANALYSIS.copy()
name = df.columns[13]
display(f"Component: {name}")
```

```
'Component: Mining'
```

```
In [41]: df = df_export_ANALYSIS[[name]]
```

```
df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

| class      | Mining    |
|------------|-----------|
| 1995-02-01 | -0.051771 |
| 1995-03-01 | -0.051724 |
| 1995-04-01 | 0.612626  |
| 1995-05-01 | -0.292828 |
| 1995-06-01 | 0.211692  |
| ...        | ...       |
| 2023-11-01 | -0.552960 |
| 2023-12-01 | 0.581095  |
| 2024-01-01 | -0.020543 |
| 2024-02-01 | 0.523124  |
| 2024-03-01 | -0.176273 |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

```
Running ADF Test for Stationarity...
ADF Statistic: -17.492079900767735
p-value: 4.431356150193511e-30
Critical Value 1%: -3.4492815848836296
Critical Value 5%: -2.8698813715275406
Critical Value 10%: -2.5712138845950587
is_stationary: True
```

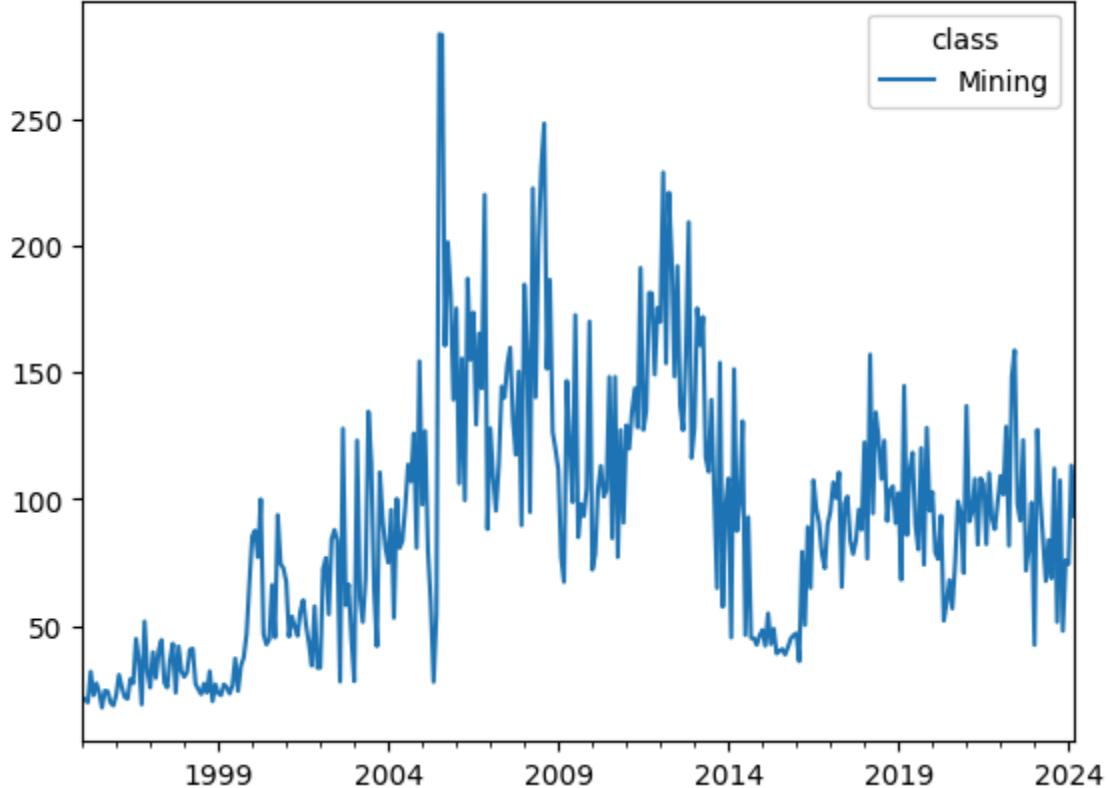
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.15113338218460284
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

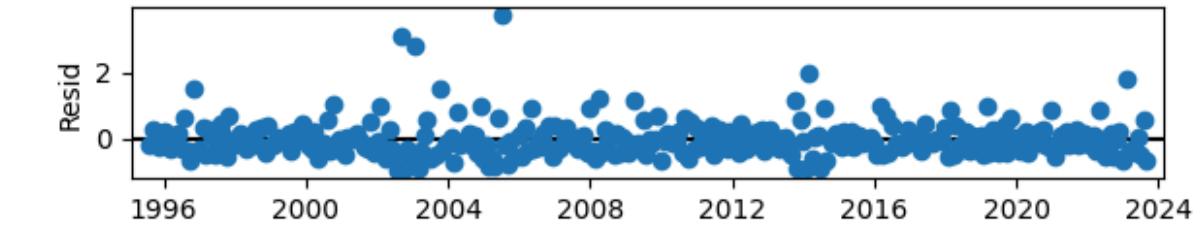
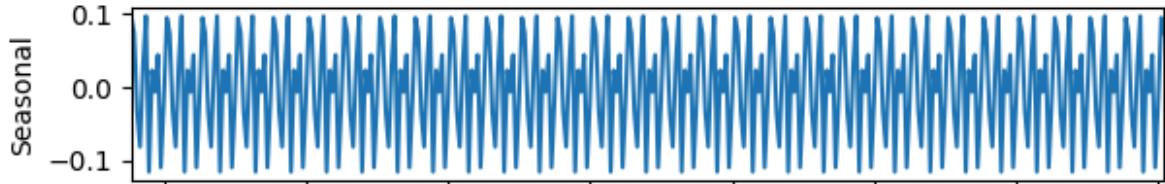
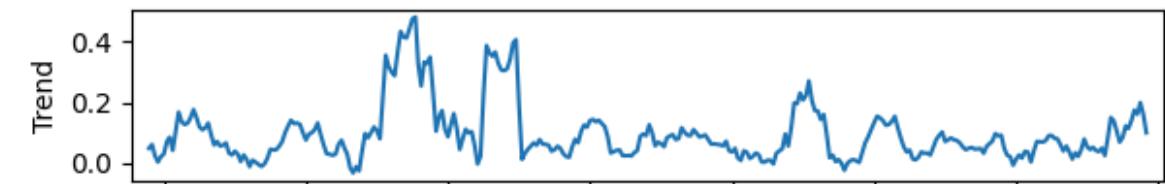
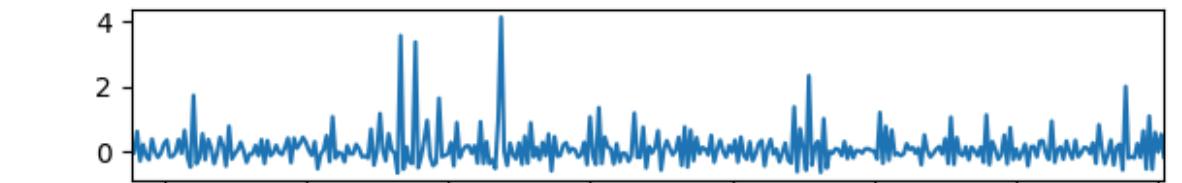
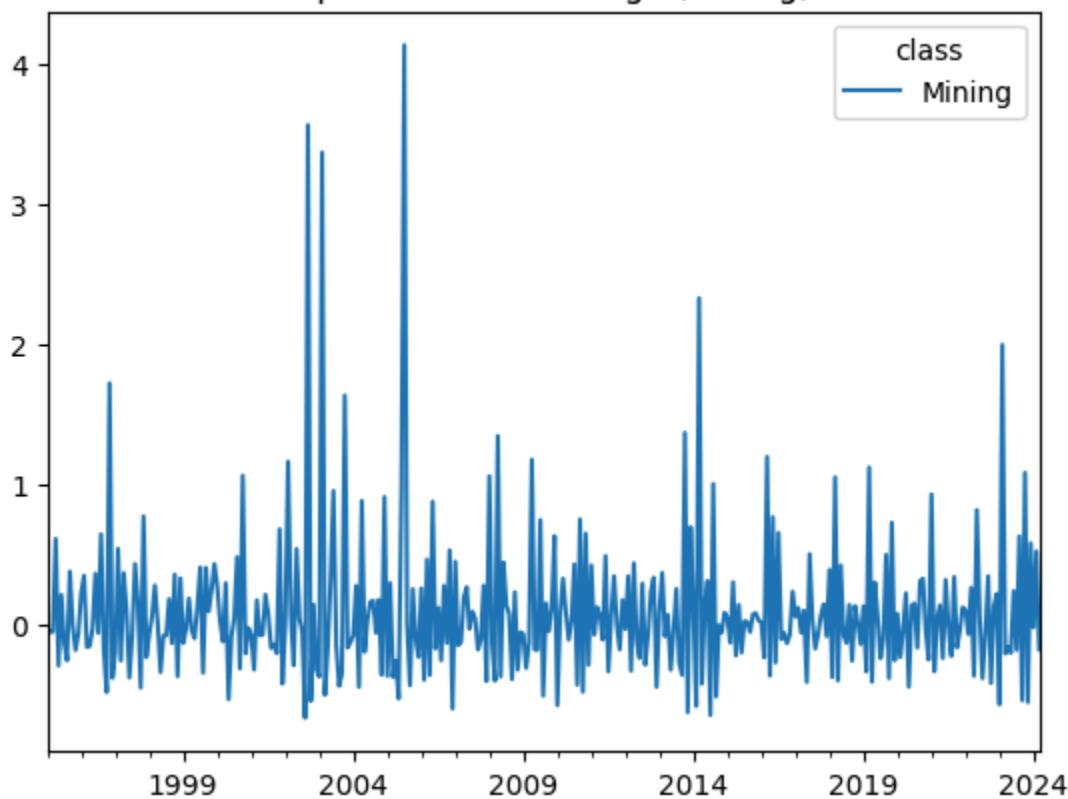
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

Export Value (Mining)

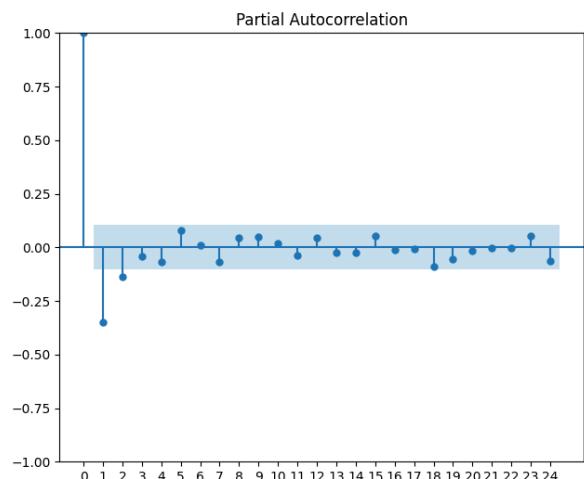
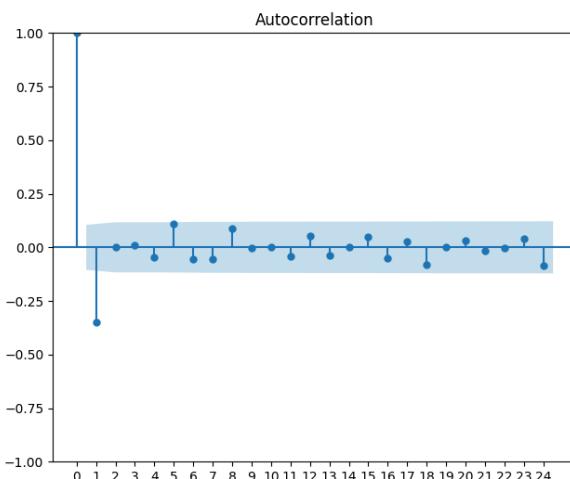
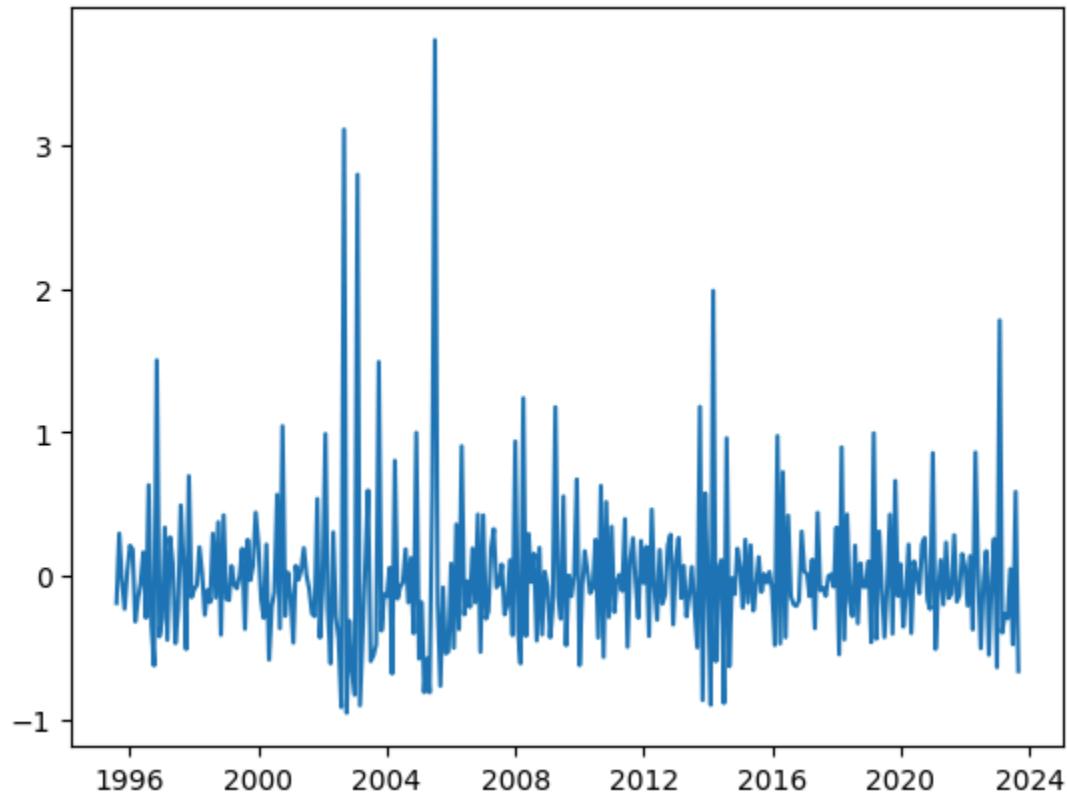


## Export Value % Change (Mining)



Plotting Residuals for Homoscedasticity Check...

### Residuals



Time series assumptions are met.

True

'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

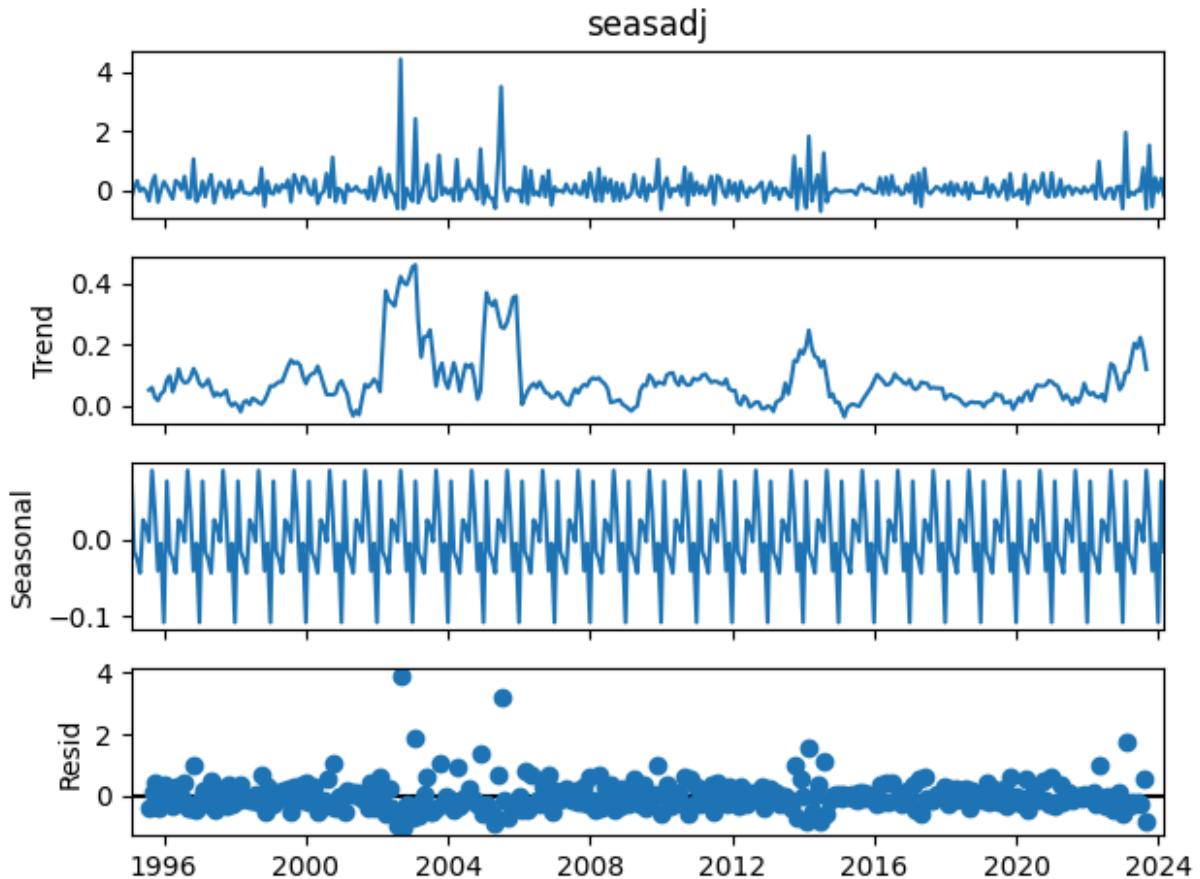
```
Running ADF Test for Stationarity...
ADF Statistic: -17.268486023474555
p-value: 5.8810299215281175e-30
Critical Value 1%: -3.4492815848836296
Critical Value 5%: -2.8698813715275406
Critical Value 10%: -2.5712138845950587
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.16488336042213542
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

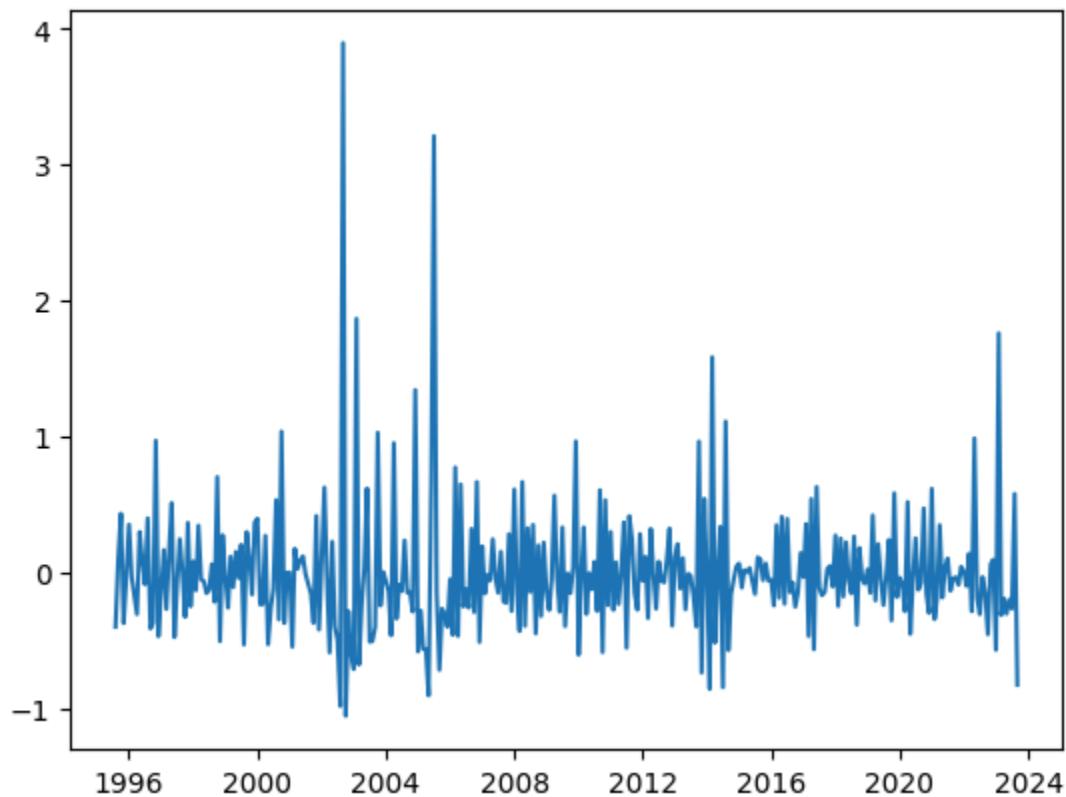
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

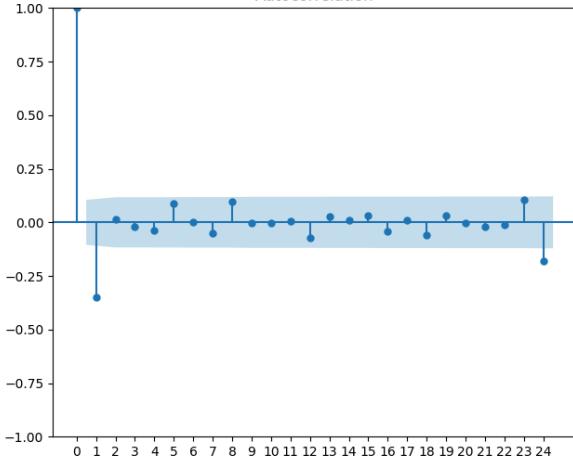


Plotting Residuals for Homoscedasticity Check...

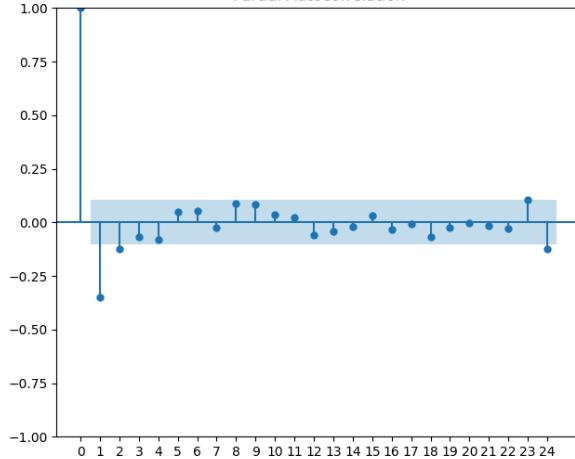
### Residuals



Autocorrelation



Partial Autocorrelation



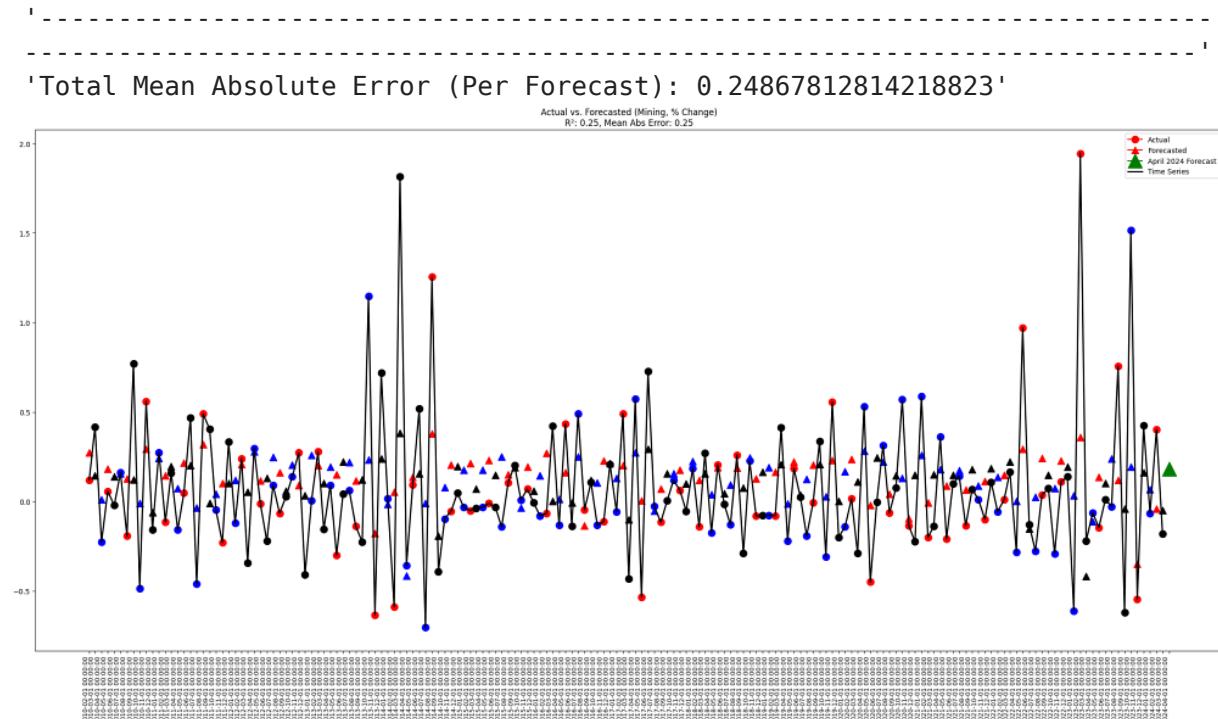
Time series assumptions are met.

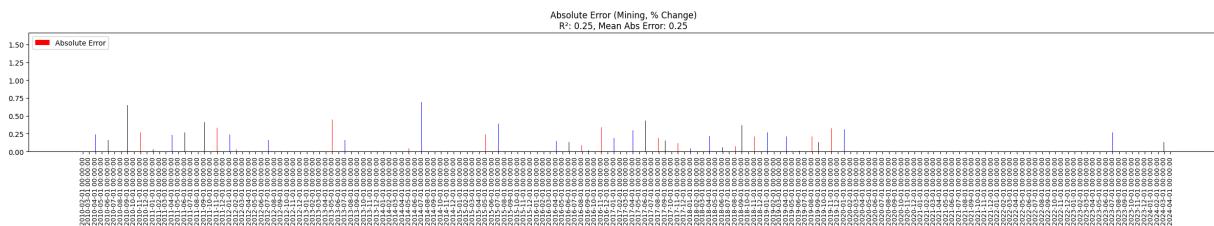
'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Mining     | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01 | 0.119365  | 0.273163  | -0.153798 | 0.153798  | 0.023654      | 1.288462             |
| 2010-03-01 | 0.415592  | 0.145709  | 0.269883  | 0.269883  | 0.072837      | 0.649395             |
| 2010-04-01 | -0.227107 | 0.011450  | -0.238556 | 0.238556  | 0.056909      | 1.050415             |
| 2010-05-01 | 0.057088  | 0.182600  | -0.125512 | 0.125512  | 0.015753      | 2.198553             |
| 2010-06-01 | -0.021650 | 0.140432  | -0.162082 | 0.162082  | 0.026271      | 7.486320             |
| ...        | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01 | 0.423529  | 0.161417  | 0.262112  | 0.262112  | 0.068703      | 0.618877             |
| 2024-01-01 | -0.064944 | 0.068019  | -0.132963 | 0.132963  | 0.017679      | 2.047348             |
| 2024-02-01 | 0.401410  | -0.041370 | 0.442780  | 0.442780  | 0.196054      | 1.103062             |
| 2024-03-01 | -0.179951 | -0.049866 | -0.130085 | 0.130085  | 0.016922      | 0.722893             |
| 2024-04-01 | NaN       | 0.182184  | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns





## Crude\_oil

```
In [42]: df = df_export_ANALYSIS.copy()
name = df.columns[14]
display(f"Component: {name}")

'Component: Crude_oil'
```

```
In [43]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.replace(0, 1e-10).ffill().bfill().pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value.pkl', 'rb') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/{name}_accuracy.pkl")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

'Timeseries of component in % form:'
```

| class      | Crude_oil     |
|------------|---------------|
| 1995-02-01 | 0.000000e+00  |
| 1995-03-01 | 0.000000e+00  |
| 1995-04-01 | 0.000000e+00  |
| 1995-05-01 | 0.000000e+00  |
| 1995-06-01 | 0.000000e+00  |
| ...        | ...           |
| 2023-11-01 | -1.000000e+00 |
| 2023-12-01 | 2.534000e+11  |
| 2024-01-01 | -5.564325e-02 |
| 2024-02-01 | 1.104471e+00  |
| 2024-03-01 | -2.776013e-01 |

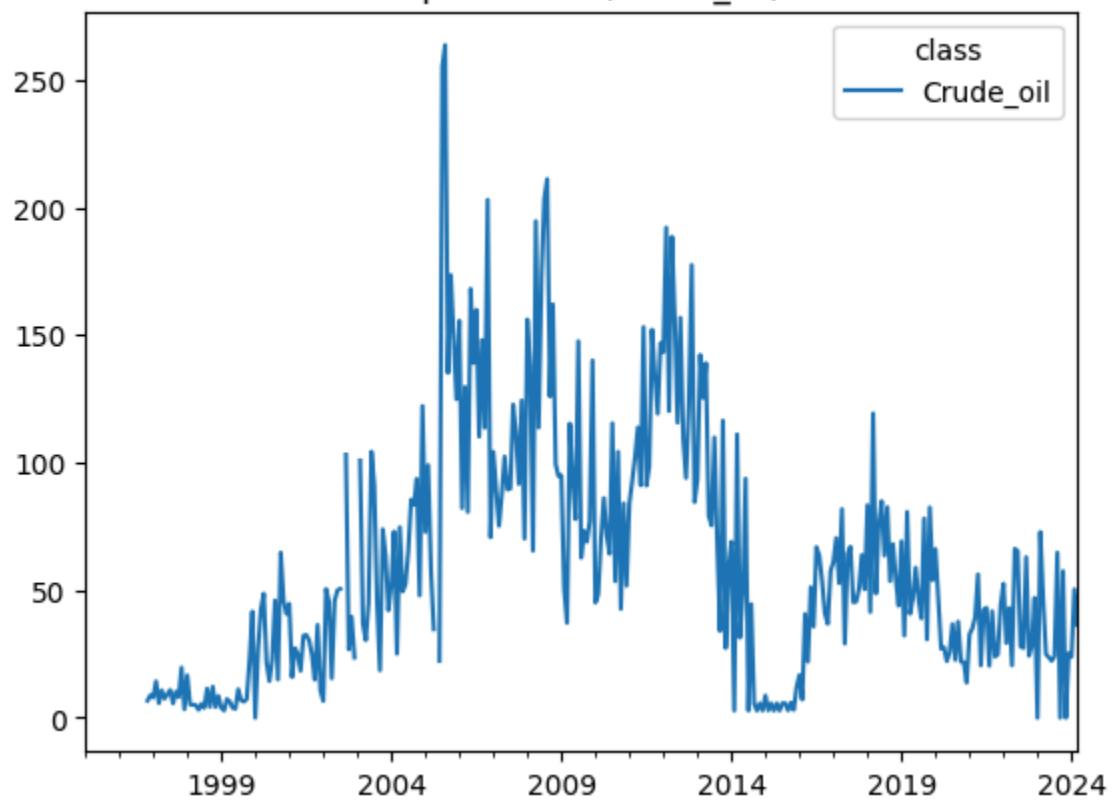
350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: 0.697860115752949
p-value: 0.9897950676701244
Critical Value 1%: -3.4497304638968043
Critical Value 5%: -2.8700785273763487
Critical Value 10%: -2.571319005190311
is_stationary: False
```

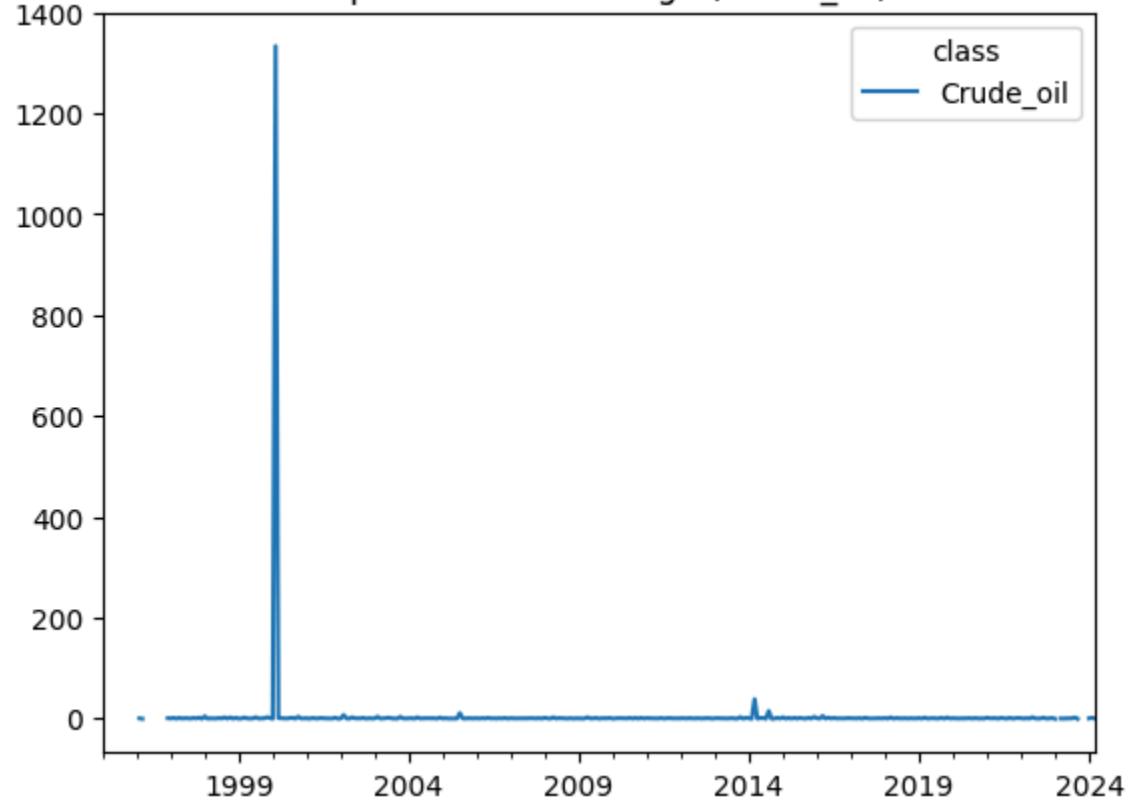
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.6549118355121443
p-value: 0.017644378589805065
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: False
```

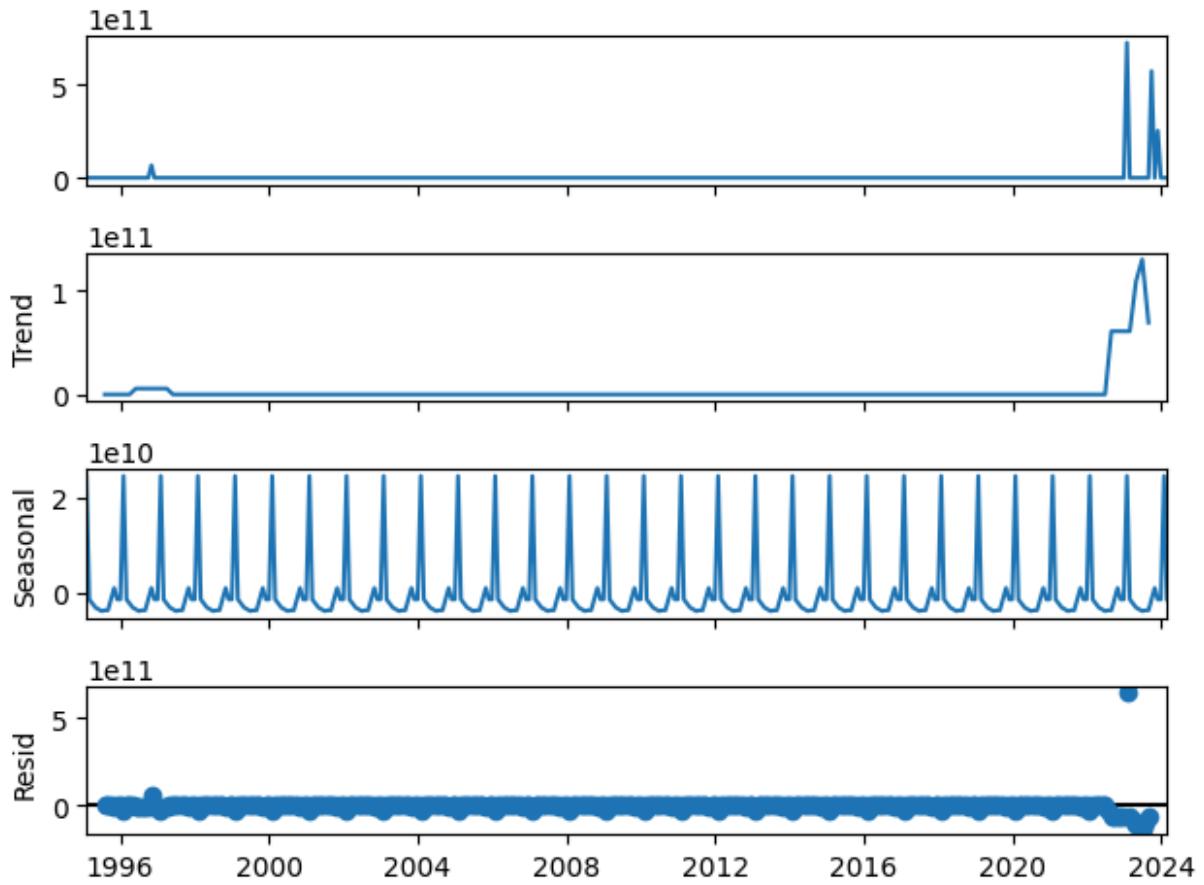
Decomposing the Series...

## Export Value (Crude\_oil)

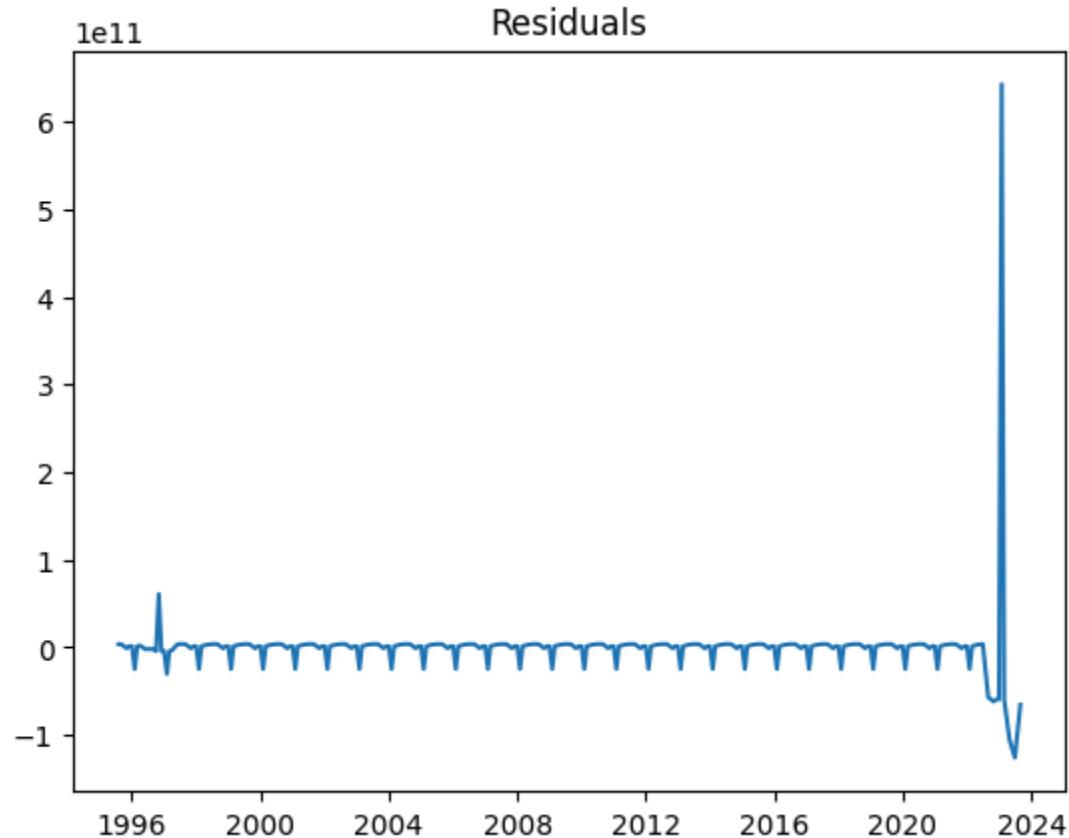


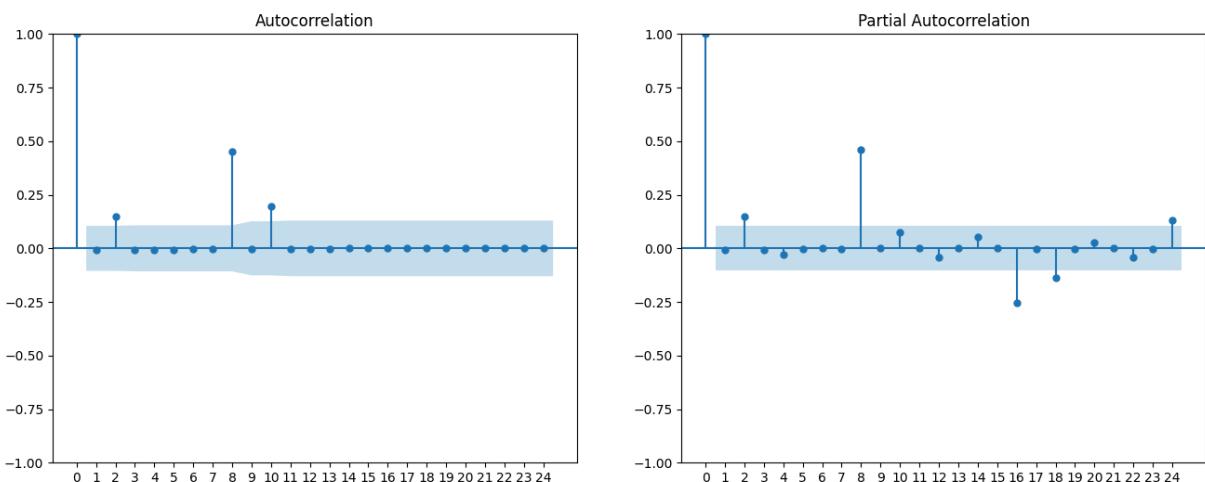
## Export Value % Change (Crude\_oil)





Plotting Residuals for Homoscedasticity Check...





Time series assumptions are not met. Further investigation needed.

False

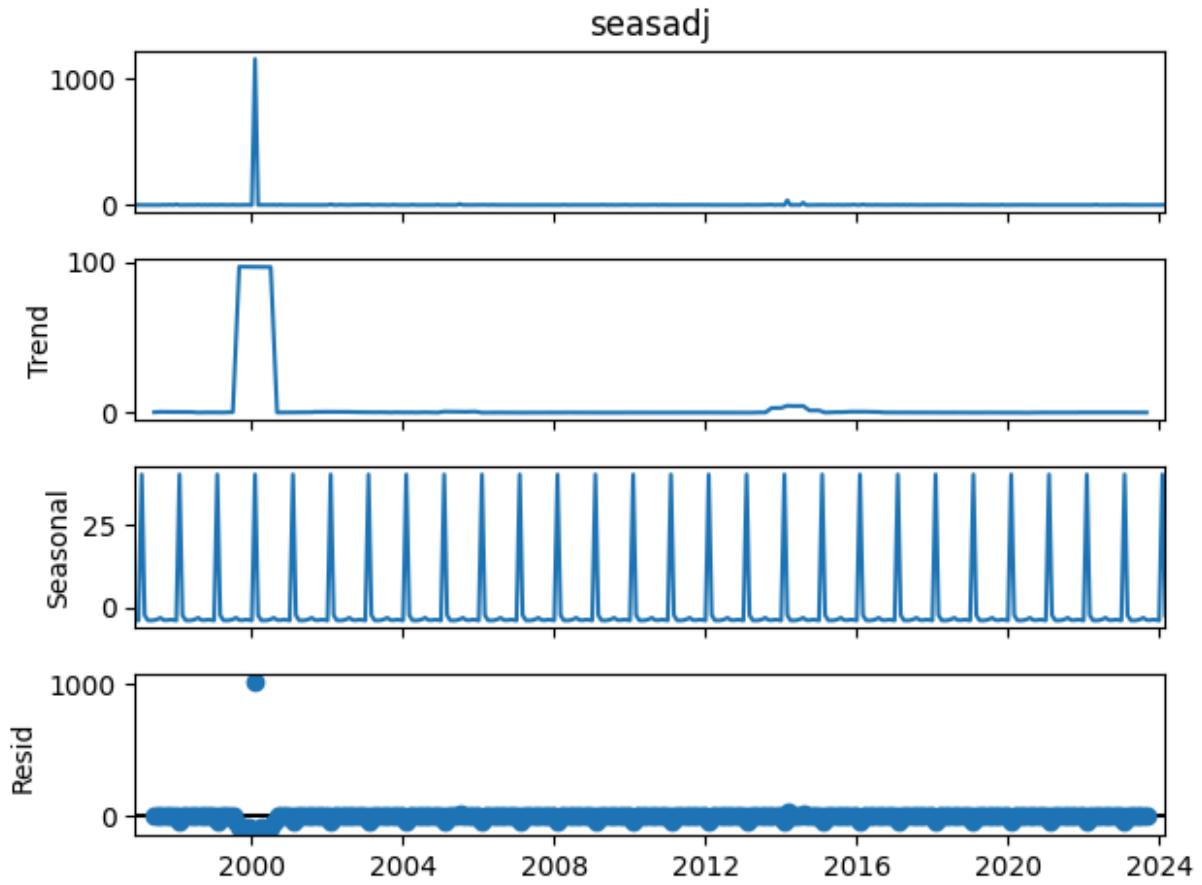
```
'-----'
-----'
Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -18.102747669937845
p-value: 2.5507924713519417e-30
Critical Value 1%: -3.45050711373316
Critical Value 5%: -2.8704195794076743
Critical Value 10%: -2.571500856923753
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.22793306398210678
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

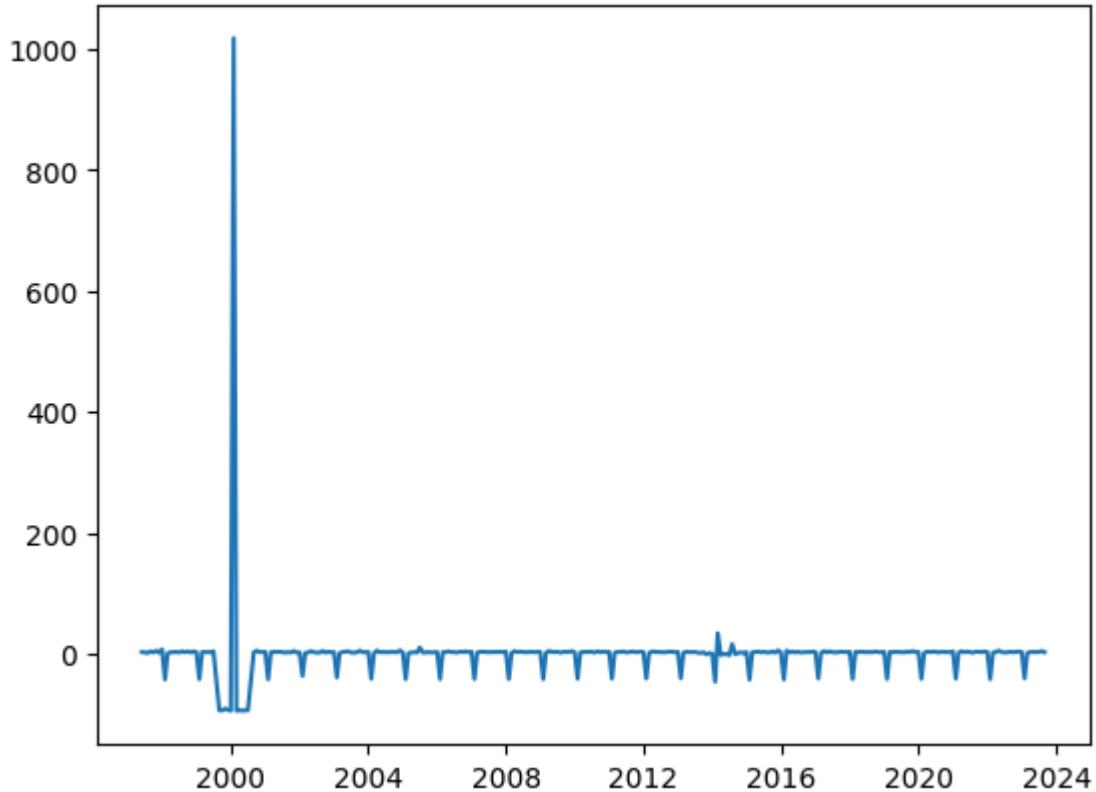
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

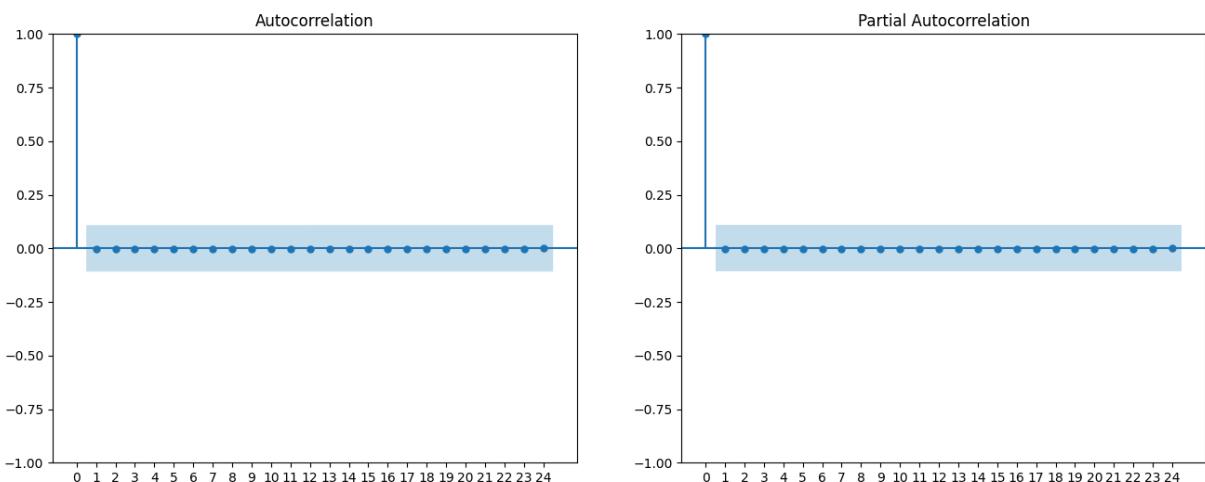
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals





Time series assumptions are met.

'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Crude_oil  | actual    | forecast | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|----------|-----------|-----------|---------------|----------------------|
| 2011-12-01 | 0.453250  | 0.0      | 0.453250  | 0.453250  | 0.205435      | 1.0                  |
| 2012-01-01 | -0.332757 | 0.0      | -0.332757 | 0.332757  | 0.110727      | 1.0                  |
| 2012-02-01 | 0.630862  | 0.0      | 0.630862  | 0.630862  | 0.397987      | 1.0                  |
| 2012-03-01 | -0.358105 | 0.0      | -0.358105 | 0.358105  | 0.128239      | 1.0                  |
| 2012-04-01 | 0.440111  | 0.0      | 0.440111  | 0.440111  | 0.193698      | 1.0                  |
| ...        | ...       | ...      | ...       | ...       | ...           | ...                  |
| 2023-12-01 | -0.569404 | 0.0      | -0.569404 | 0.569404  | 0.324221      | 1.0                  |
| 2024-01-01 | -0.200591 | 0.0      | -0.200591 | 0.200591  | 0.040237      | 1.0                  |
| 2024-02-01 | 1.297005  | 0.0      | 1.297005  | 1.297005  | 1.682223      | 1.0                  |
| 2024-03-01 | -0.299179 | 0.0      | -0.299179 | 0.299179  | 0.089508      | 1.0                  |
| 2024-04-01 | NaN       | 0.0      | NaN       | NaN       | NaN           | NaN                  |

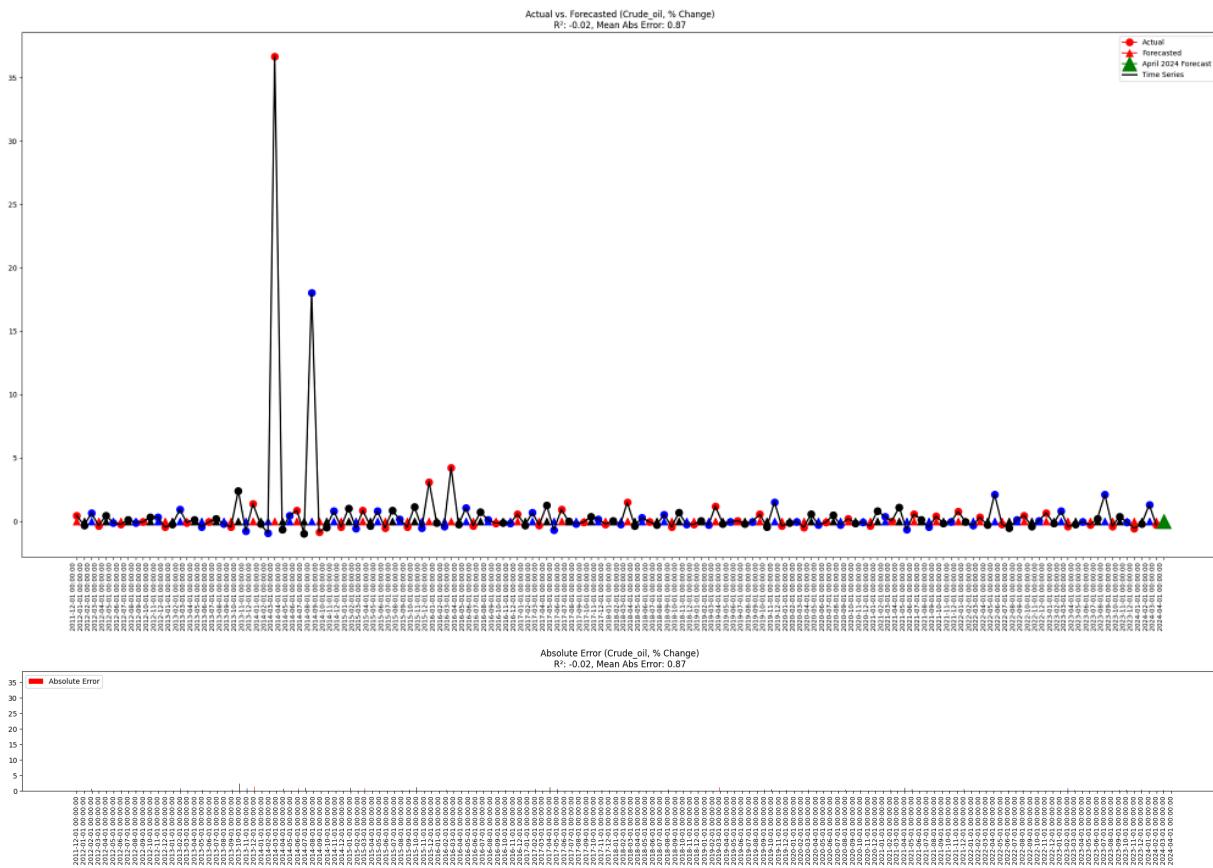
149 rows × 6 columns

'-----'

'-----'

'Total Mean Absolute Error (Per Forecast): 0.8664711038237168'

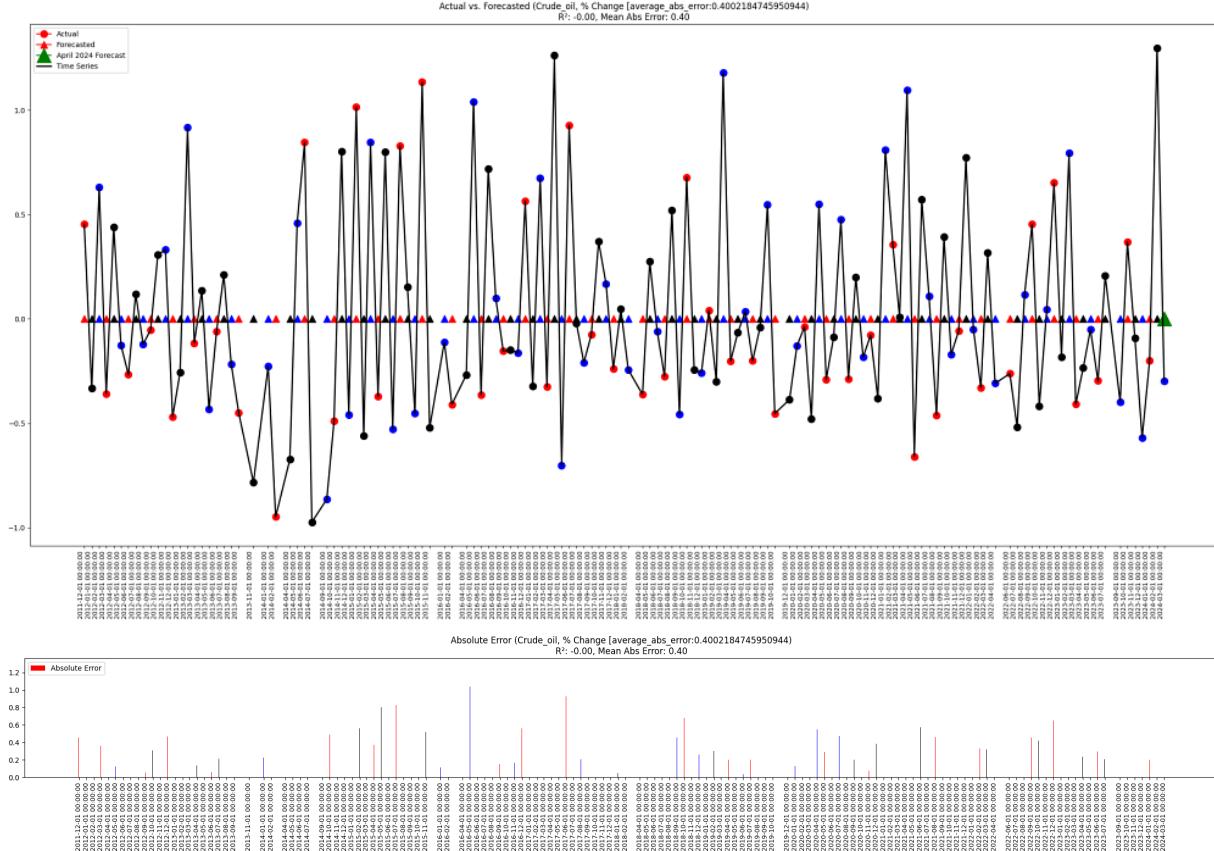
## 1\_composite\_export\_value\_analysis



```
In [44]: #it seems like the outliers miss are extremely high, which can justify the high MAE
#remove the outliers via IQR on abs error
Q1 = df_accuracy["abs_error"].quantile(0.25)
Q3 = df_accuracy["abs_error"].quantile(0.75)
IQR = Q3 - Q1
df_accuracy = df_accuracy[(df_accuracy["abs_error"] < (Q3 + 1.5 * IQR)) & (df_accuracy["abs_error"] > (Q1 - 1.5 * IQR))]
#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name, type=f"% Change [average_abs_error]")


```

'Total Mean Absolute Error (Per Forecast): 0.4002184745950944'



nothing can be done for this dataset, as the outlier jumps are shocks instead of non-stochastic trends and is similar to noise(just extreme), it might be possible to model this using non-timeseries analysis like sentiment/news/event-driven market reaction, which is not available publicly. Best course of action is to disregard crude oil and use the mining timeseries which is the parent for this that has a non-stochastic trend that can be modelled, as a single series. Furthermore, Crude oil exports are not frequent or a major weight.

Mineral\_products,\_n.i.e.

```
In [45]: df = df_export_ANALYSIS.copy()
name = df.columns[15]
display(f"Component: {name}")

'Component: Mineral_products,_n.i.e.'
```

```
In [46]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Mineral_products,_n.i.e. |
|------------|--------------------------|
| 1995-02-01 | -0.051771                |
| 1995-03-01 | -0.051724                |
| 1995-04-01 | 0.612626                 |
| 1995-05-01 | -0.292828                |
| 1995-06-01 | 0.211692                 |
| ...        | ...                      |
| 2023-11-01 | -0.038438                |
| 2023-12-01 | 0.053300                 |
| 2024-01-01 | -0.002767                |
| 2024-02-01 | 0.247374                 |
| 2024-03-01 | -0.095026                |

350 rows × 1 columns

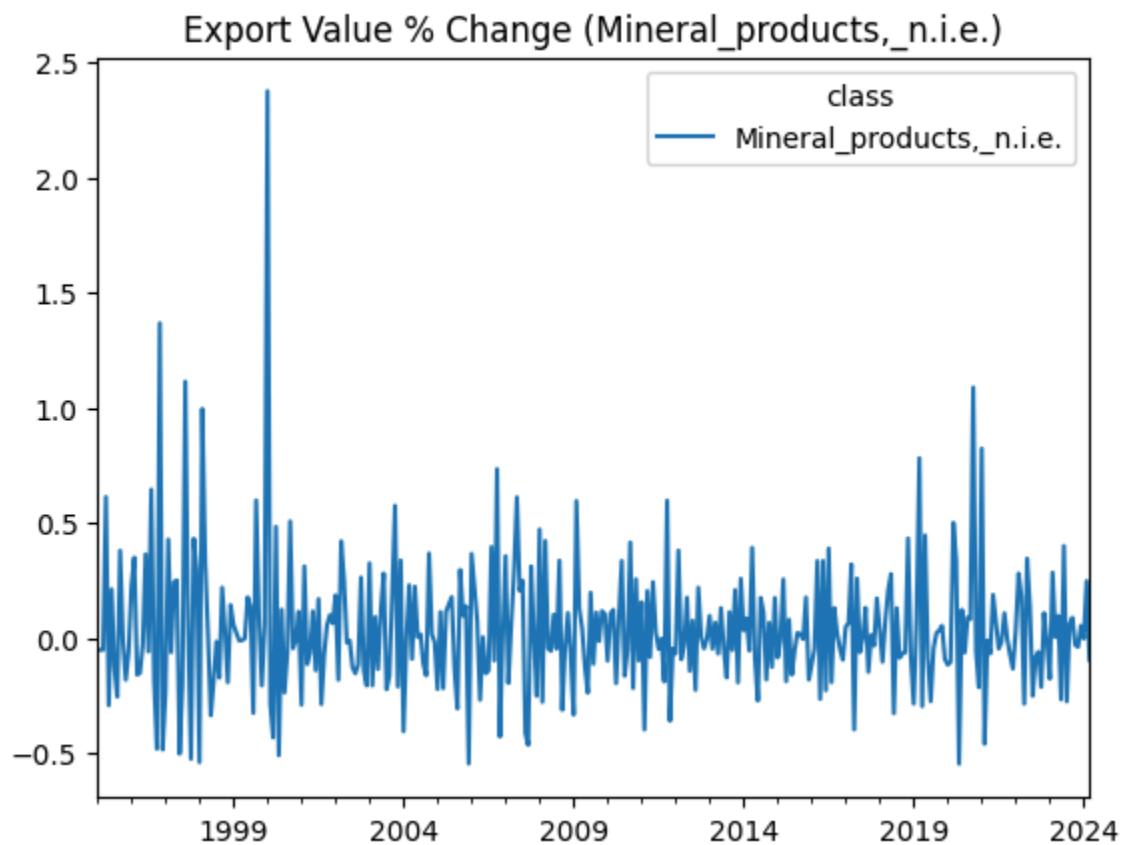
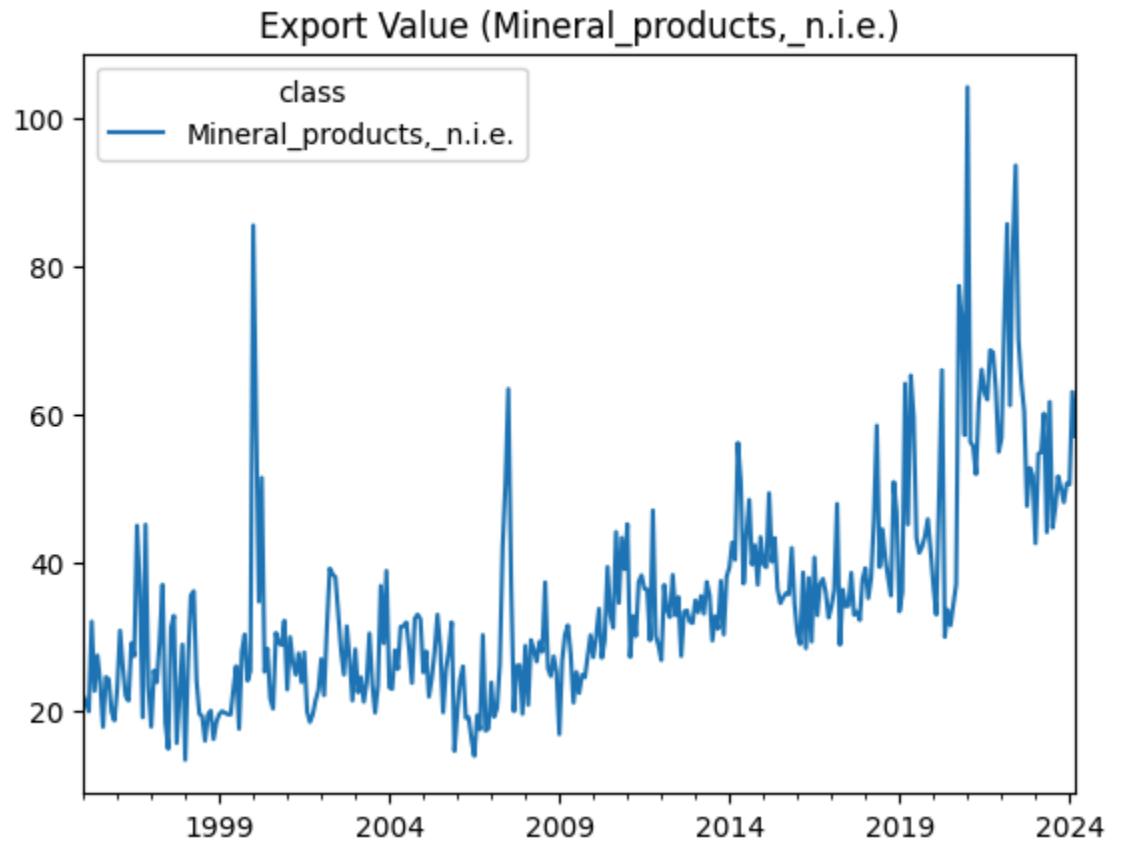
```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -19.759938584546642
p-value: 0.0
Critical Value 1%: -3.4492815848836296
Critical Value 5%: -2.8698813715275406
Critical Value 10%: -2.5712138845950587
is_stationary: True
```

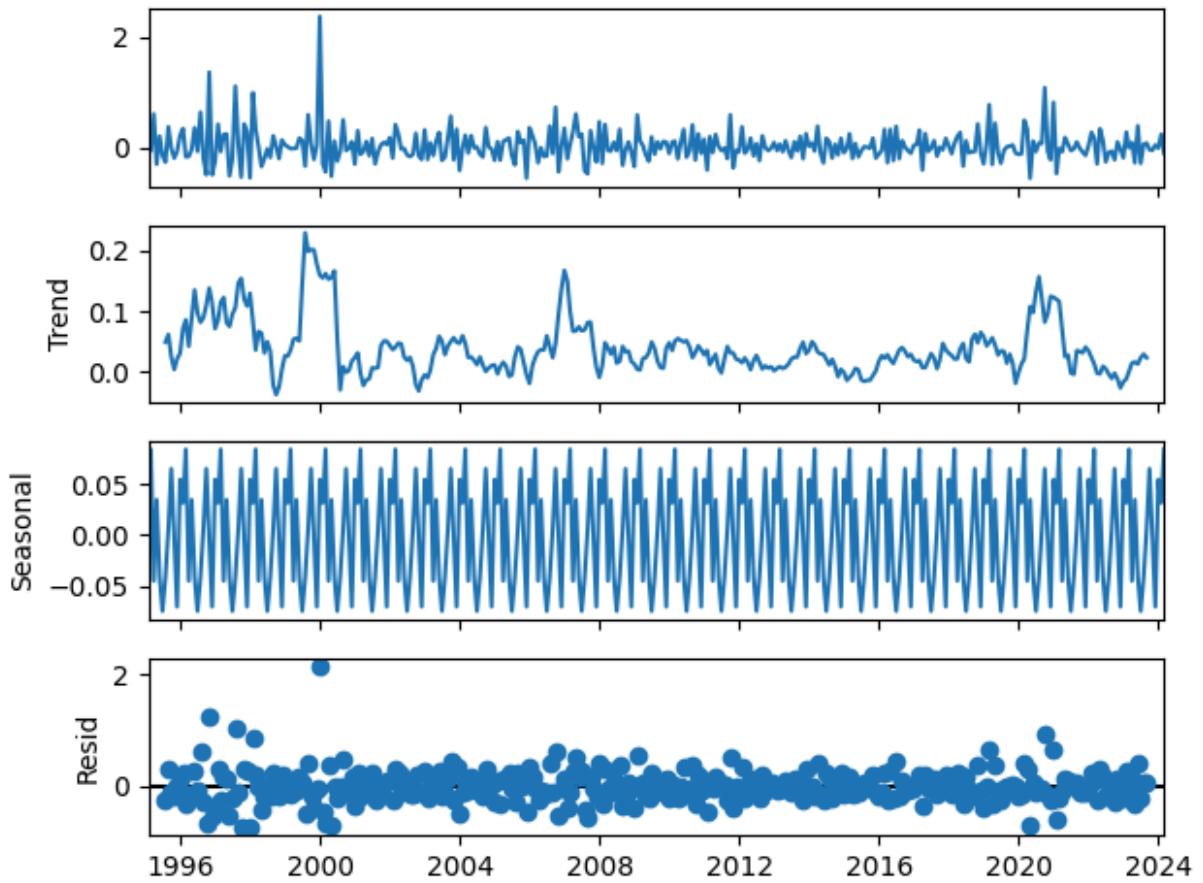
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.2521266141514137
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

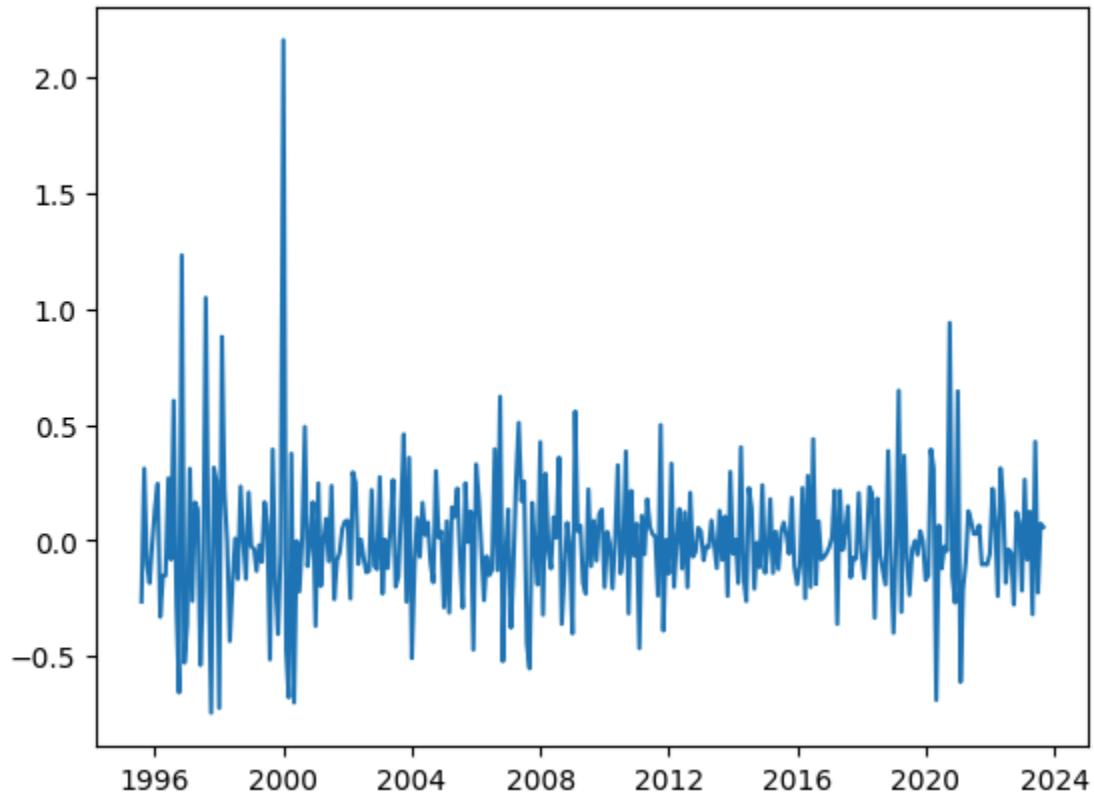
```
result = kpss(series, regression='c')
```

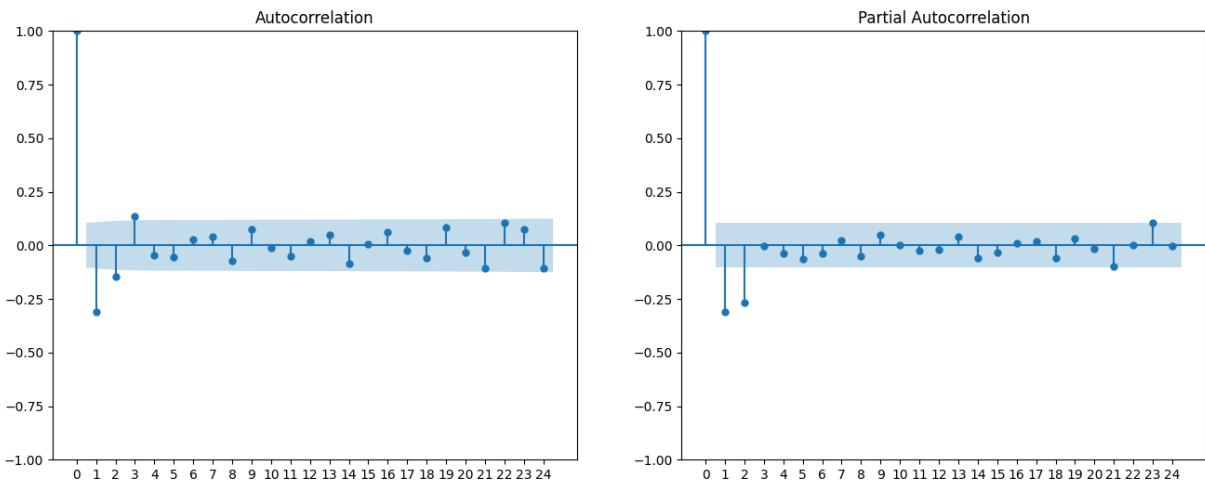




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

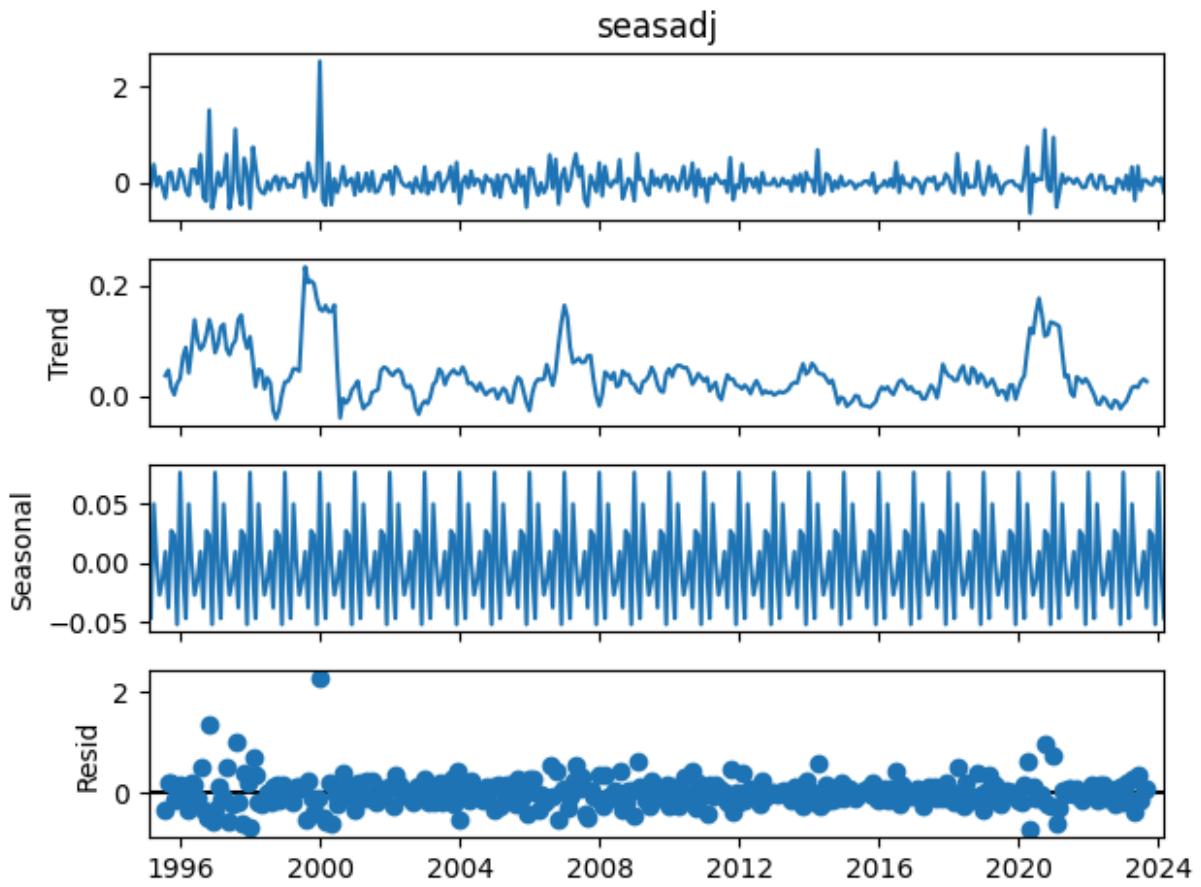
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted): '
Running ADF Test for Stationarity...
ADF Statistic: -20.25638353021514
p-value: 0.0
Critical Value 1%: -3.4492815848836296
Critical Value 5%: -2.8698813715275406
Critical Value 10%: -2.5712138845950587
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.15896342612068062
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

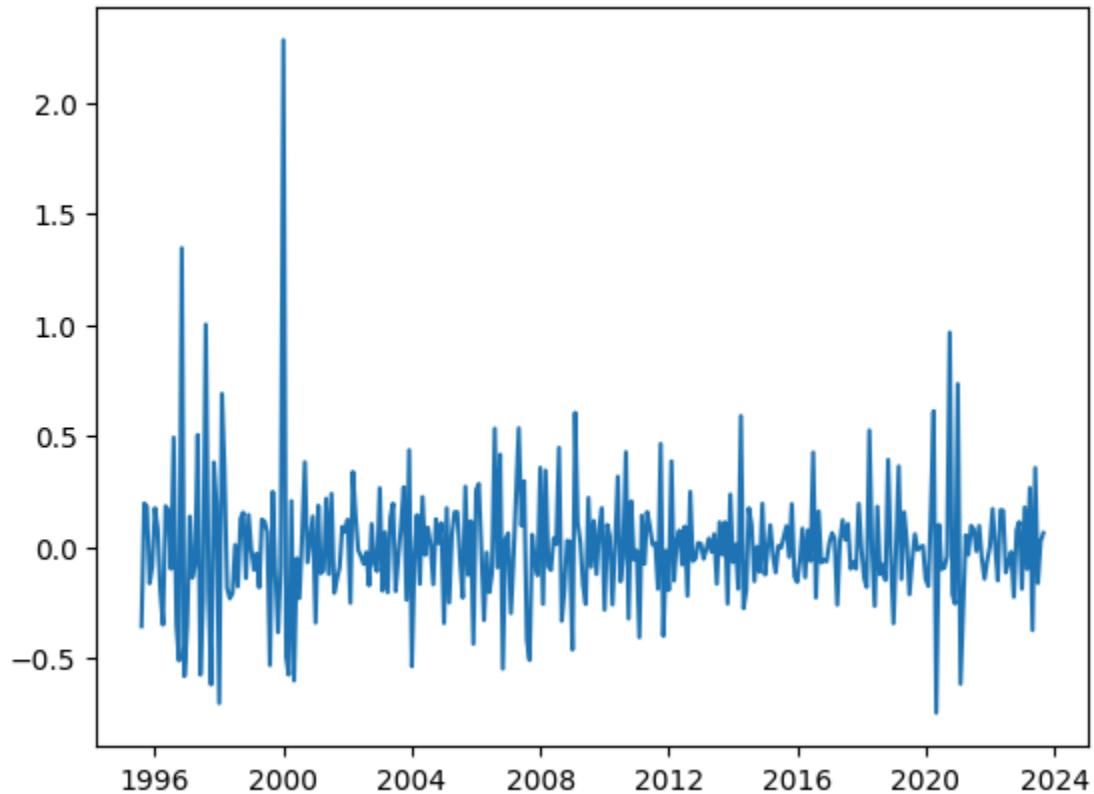
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

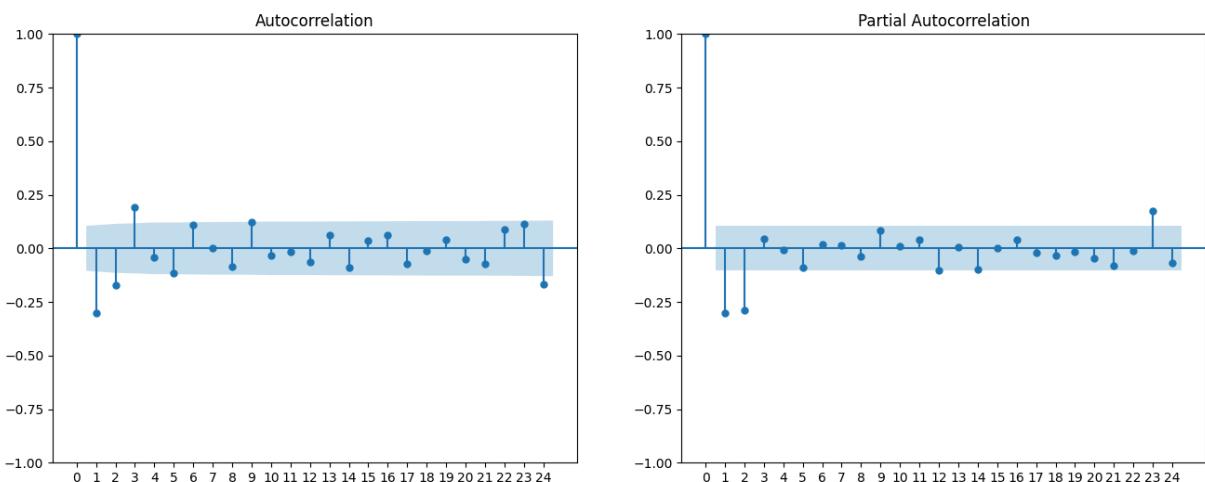
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals





Time series assumptions are met.

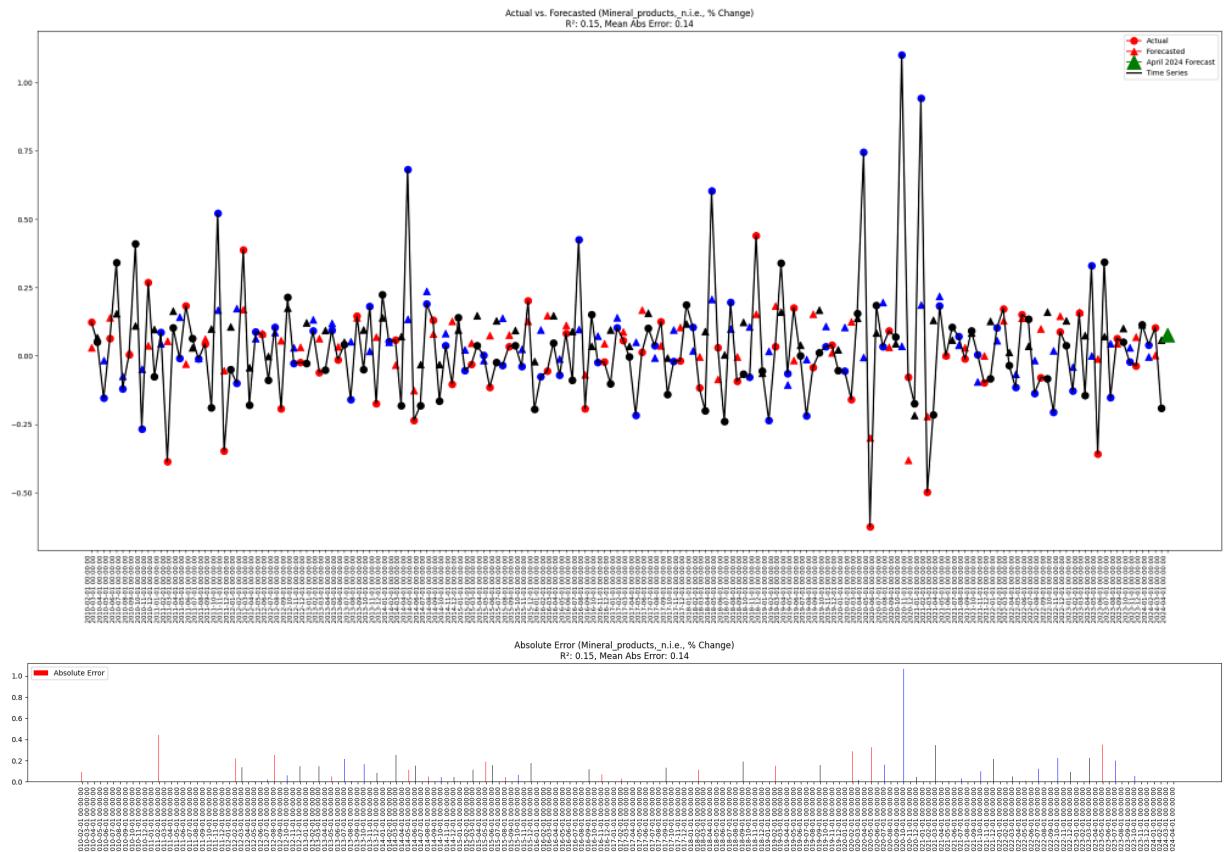
'Data of Actual vs Forecasted values with error metrics:'

| Mineral_products,_n.i.e. | actual    | forecast  | error     | abs_error | squared_error | abs_perce |
|--------------------------|-----------|-----------|-----------|-----------|---------------|-----------|
| 2010-02-01               | 0.123940  | 0.029615  | 0.094326  | 0.094326  | 8.897307e-03  |           |
| 2010-03-01               | 0.051701  | 0.067129  | -0.015427 | 0.015427  | 2.380017e-04  |           |
| 2010-04-01               | -0.154533 | -0.016077 | -0.138456 | 0.138456  | 1.917020e-02  |           |
| 2010-05-01               | 0.064468  | 0.139627  | -0.075159 | 0.075159  | 5.648805e-03  |           |
| 2010-06-01               | 0.340929  | 0.155790  | 0.185139  | 0.185139  | 3.427638e-02  |           |
| ...                      | ...       | ...       | ...       | ...       | ...           | ...       |
| 2023-12-01               | 0.113867  | 0.113383  | 0.000484  | 0.000484  | 2.338630e-07  |           |
| 2024-01-01               | 0.040300  | -0.002162 | 0.042462  | 0.042462  | 1.803016e-03  |           |
| 2024-02-01               | 0.103223  | 0.001897  | 0.101326  | 0.101326  | 1.026691e-02  |           |
| 2024-03-01               | -0.190134 | 0.058474  | -0.248608 | 0.248608  | 6.180583e-02  |           |
| 2024-04-01               | NaN       | 0.076208  | NaN       | NaN       | NaN           |           |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.14162961119665243'



## Manufacturing

```
In [47]: df = df_export_ANALYSIS.copy()
name = df.columns[16]
display(f"Component: {name}")
```

'Component: Manufacturing'

```
In [48]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

| class      | Manufacturing |
|------------|---------------|
| 1995-02-01 | 0.026971      |
| 1995-03-01 | 0.326152      |
| 1995-04-01 | -0.216692     |
| 1995-05-01 | 0.191875      |
| 1995-06-01 | 0.050954      |
| ...        | ...           |
| 2023-11-01 | 0.027194      |
| 2023-12-01 | -0.034369     |
| 2024-01-01 | -0.016421     |
| 2024-02-01 | 0.024792      |
| 2024-03-01 | 0.083343      |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

```
Running ADF Test for Stationarity...
ADF Statistic: -5.050837828510198
p-value: 1.7587027887833132e-05
Critical Value 1%: -3.4502011472639724
Critical Value 5%: -2.8702852297358983
Critical Value 10%: -2.5714292194077513
is_stationary: True
```

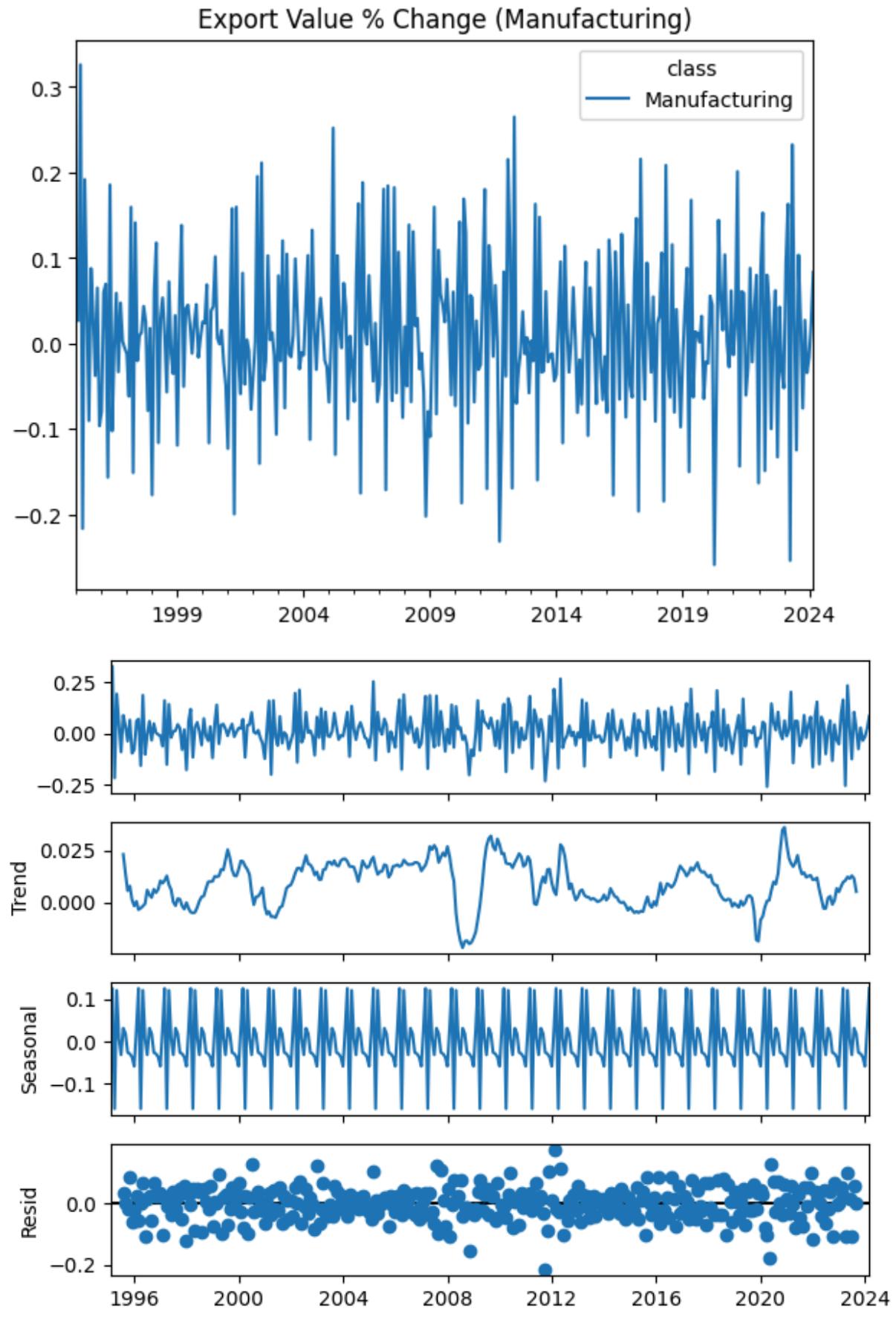
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.08037579806219414
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

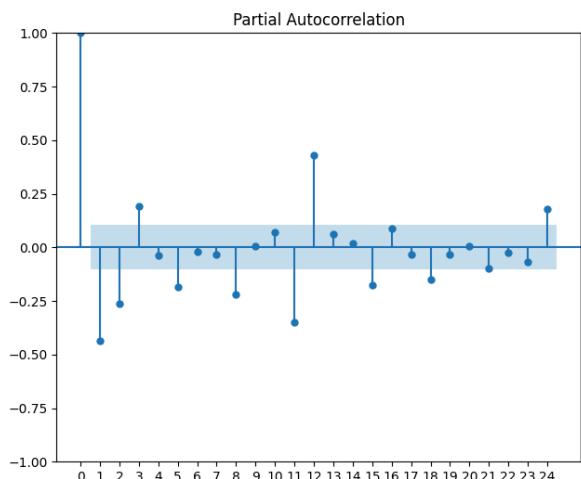
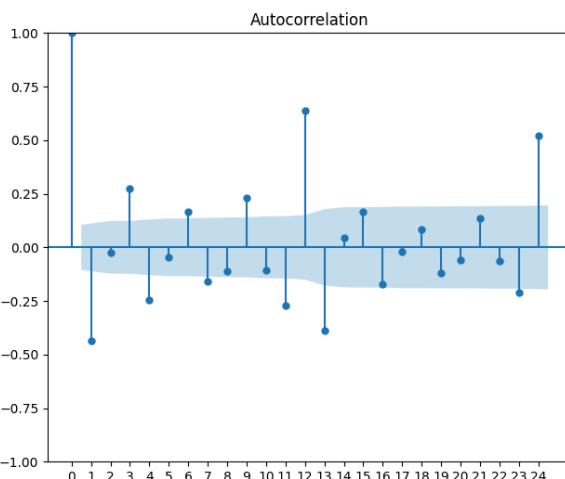
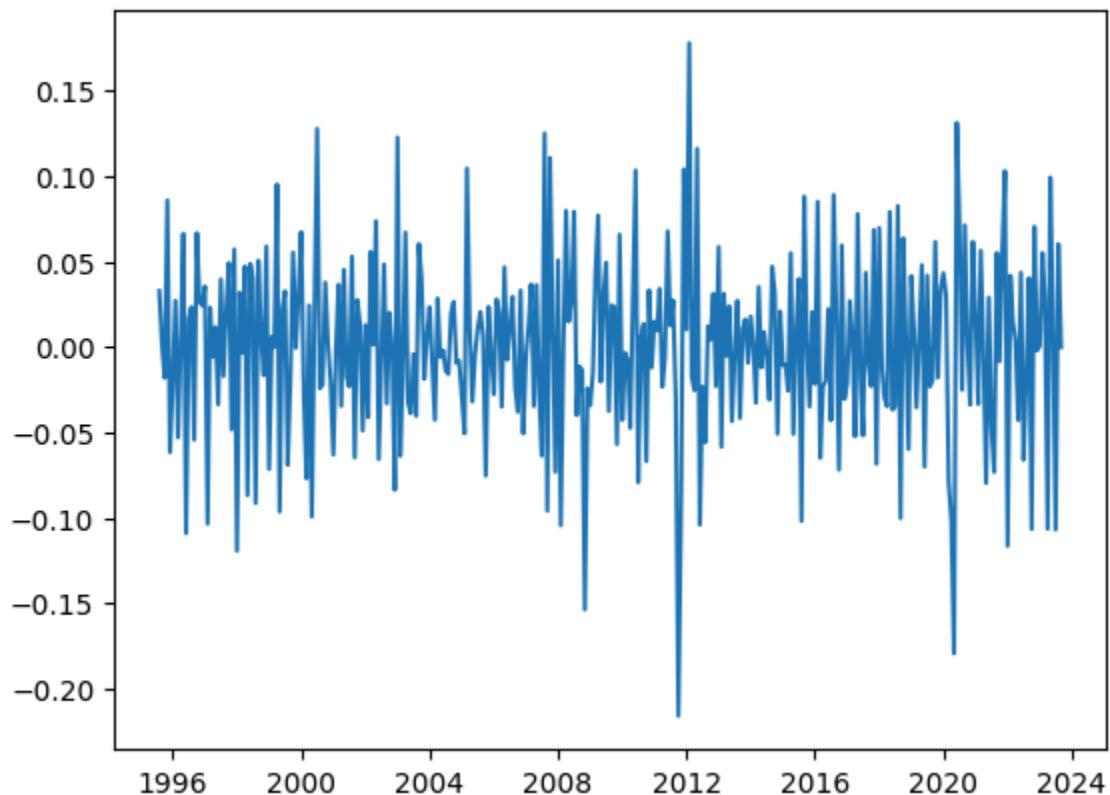
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```





### Residuals



Time series assumptions are met.

True

'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

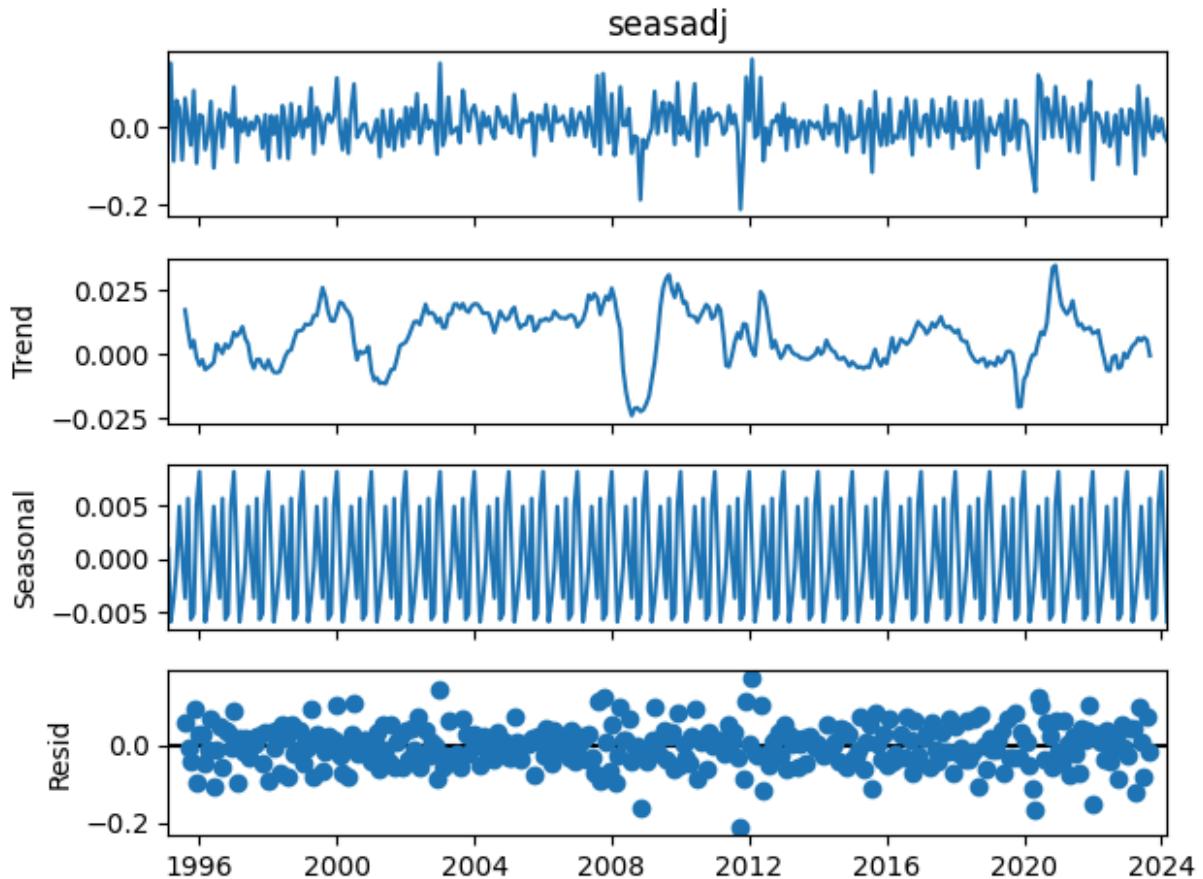
Running ADF Test for Stationarity...  
 ADF Statistic: -6.077167552411928  
 p-value: 1.1144093941294582e-07  
 Critical Value 1%: -3.449673193310592  
 Critical Value 5%: -2.8700533746644323  
 Critical Value 10%: -2.5713055939491403  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.13090623690563918  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

Decomposing the Series...

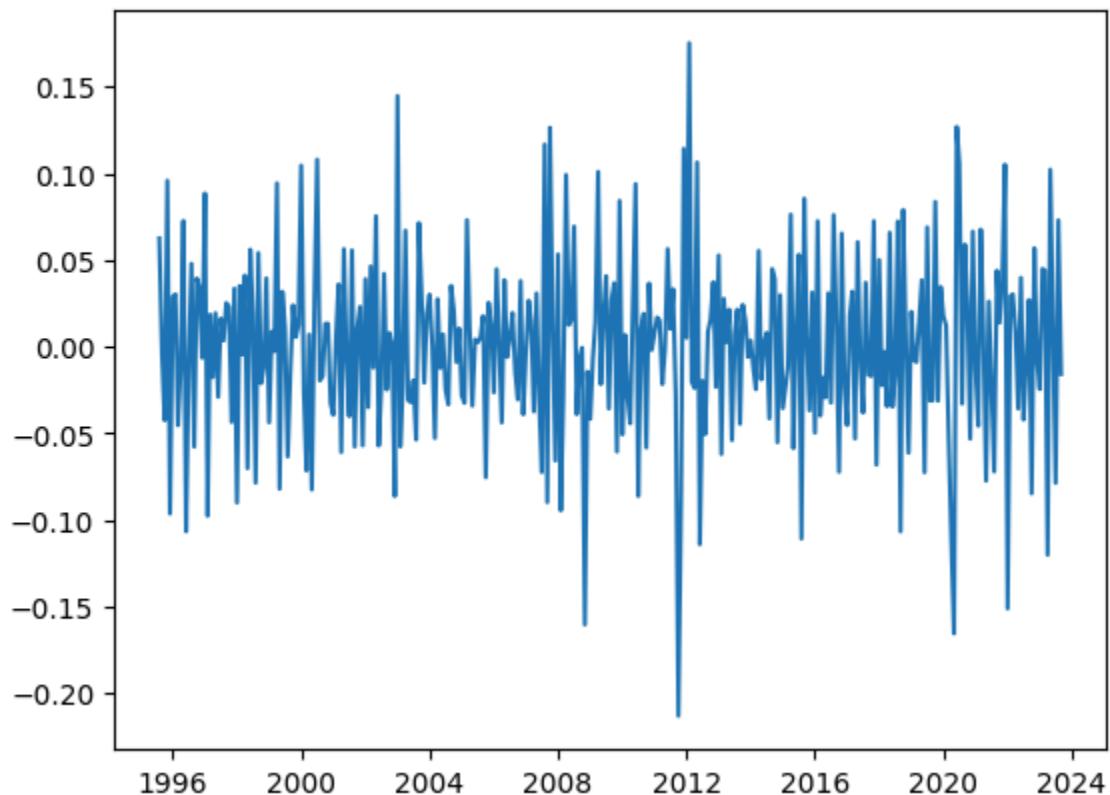
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

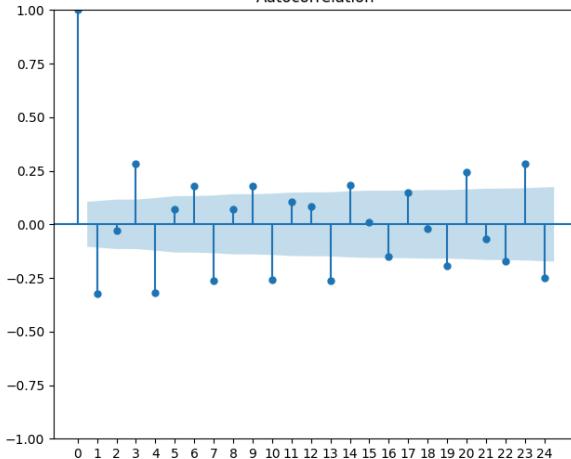


Plotting Residuals for Homoscedasticity Check...

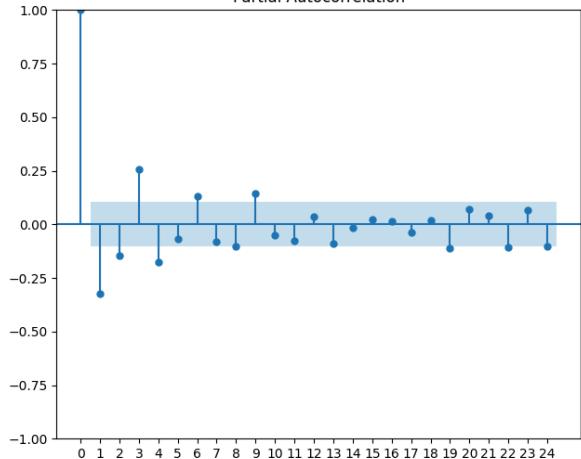
### Residuals



Autocorrelation



Partial Autocorrelation



Time series assumptions are met.

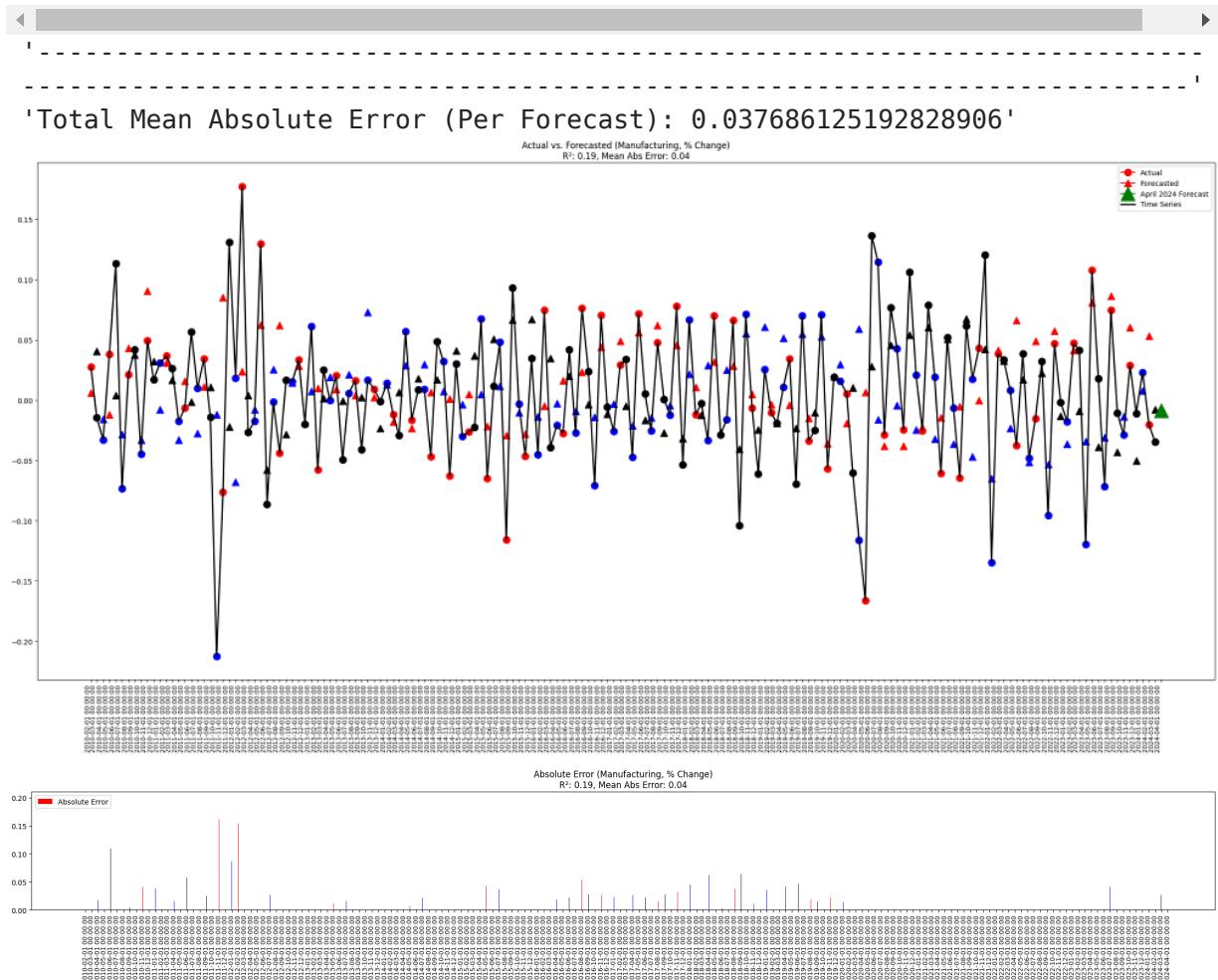
'-----'

'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Manufacturing | actual    | Forecast  | error     | abs_error | squared_error | abs_percentage_er |
|---------------|-----------|-----------|-----------|-----------|---------------|-------------------|
| 2010-02-01    | 0.027543  | 0.006181  | 0.021362  | 0.021362  | 0.000456      | 0.775%            |
| 2010-03-01    | -0.014531 | 0.040611  | -0.055142 | 0.055142  | 0.003041      | 3.794%            |
| 2010-04-01    | -0.032884 | -0.015581 | -0.017302 | 0.017302  | 0.000299      | 0.526%            |
| 2010-05-01    | 0.038053  | -0.011775 | 0.049827  | 0.049827  | 0.002483      | 1.309%            |
| 2010-06-01    | 0.113312  | 0.004087  | 0.109225  | 0.109225  | 0.011930      | 0.963%            |
| ...           | ...       | ...       | ...       | ...       | ...           | ...               |
| 2023-12-01    | -0.010933 | -0.050080 | 0.039147  | 0.039147  | 0.001532      | 3.580%            |
| 2024-01-01    | 0.023044  | 0.007639  | 0.015404  | 0.015404  | 0.000237      | 0.668%            |
| 2024-02-01    | -0.020558 | 0.053213  | -0.073771 | 0.073771  | 0.005442      | 3.588%            |
| 2024-03-01    | -0.034601 | -0.007797 | -0.026804 | 0.026804  | 0.000718      | 0.774%            |
| 2024-04-01    | NaN       | -0.008471 | NaN       | NaN       | NaN           | NaN               |

171 rows × 6 columns



## Agro-manufacturing\_Products

```
In [49]: df = df_export_ANALYSIS.copy()
name = df.columns[17]
display(f"Component: {name}")

'Component: Agro-manufacturing_Products'

In [50]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/{name}.pkl")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class Agro-manufacturing_Products |           |
|-----------------------------------|-----------|
| 1995-02-01                        | 0.077919  |
| 1995-03-01                        | 0.546396  |
| 1995-04-01                        | -0.153950 |
| 1995-05-01                        | 0.083959  |
| 1995-06-01                        | -0.075488 |
| ...                               | ...       |
| 2023-11-01                        | 0.005859  |
| 2023-12-01                        | -0.083868 |
| 2024-01-01                        | 0.013838  |
| 2024-02-01                        | 0.094663  |
| 2024-03-01                        | 0.096929  |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -4.795625334992418
p-value: 5.544366445417211e-05
Critical Value 1%: -3.449962981927952
Critical Value 5%: -2.870180642420163
Critical Value 10%: -2.5713734527352607
is_stationary: True
```

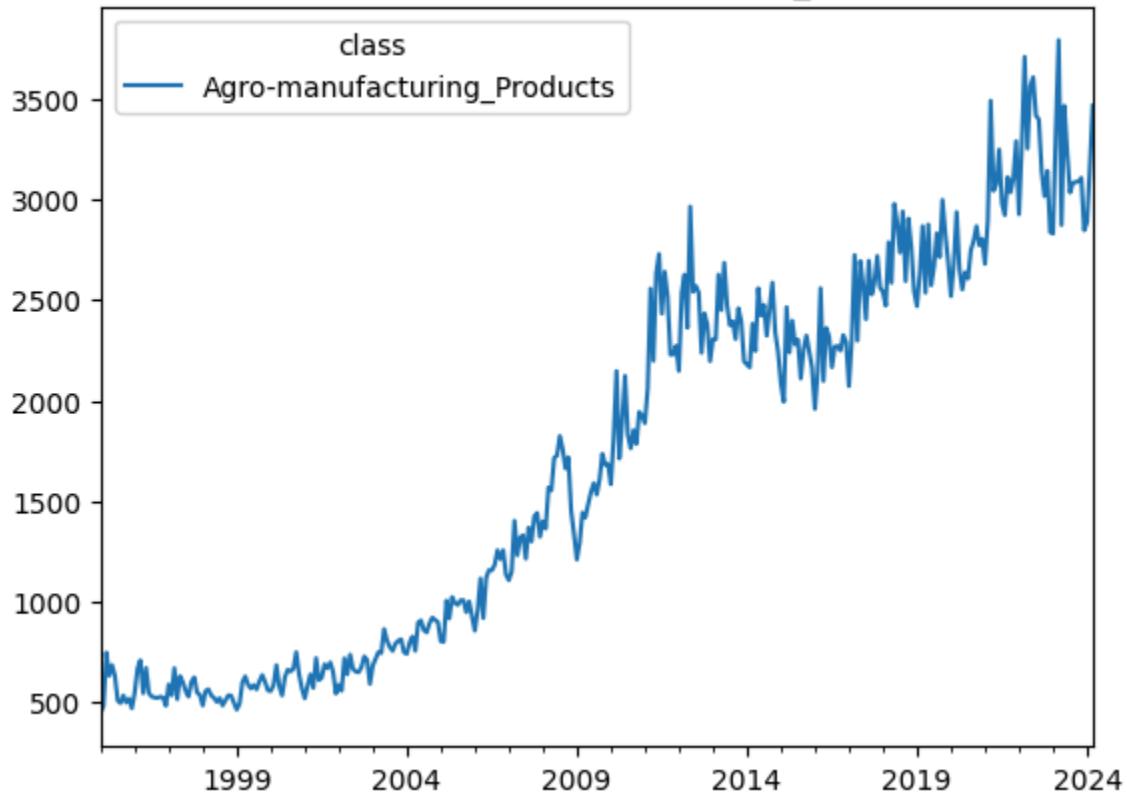
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.12843393488062868
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

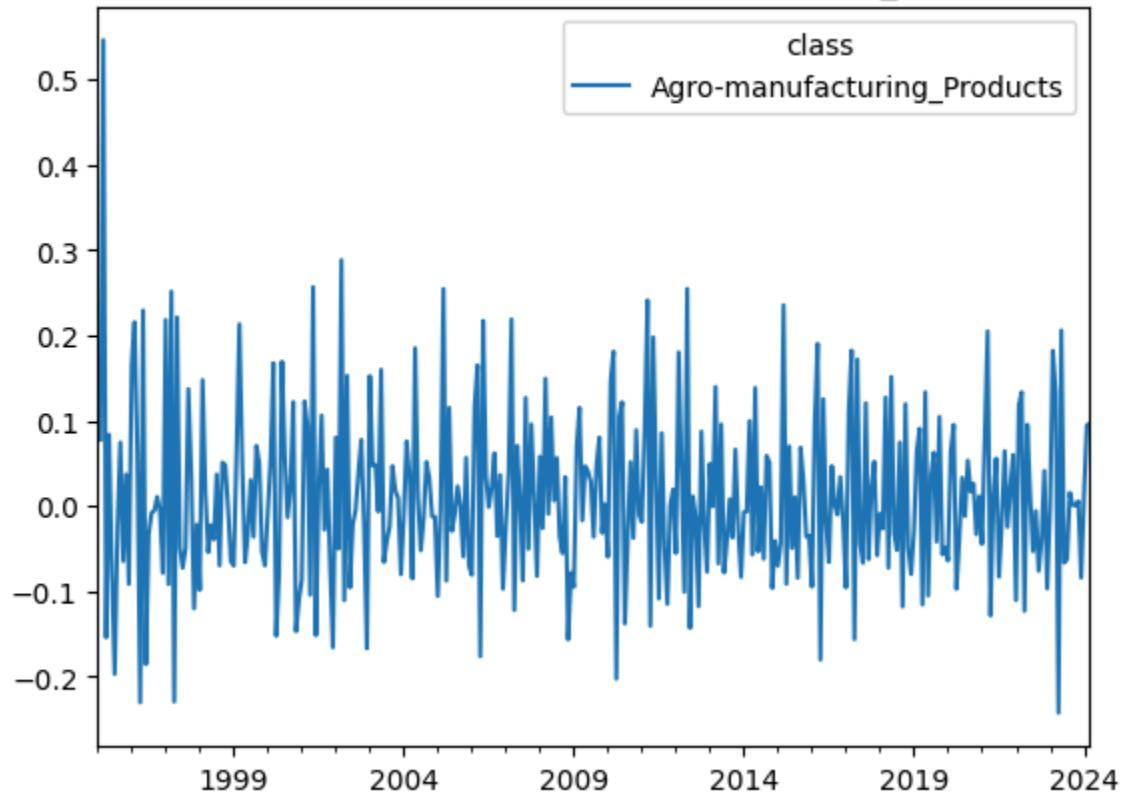
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

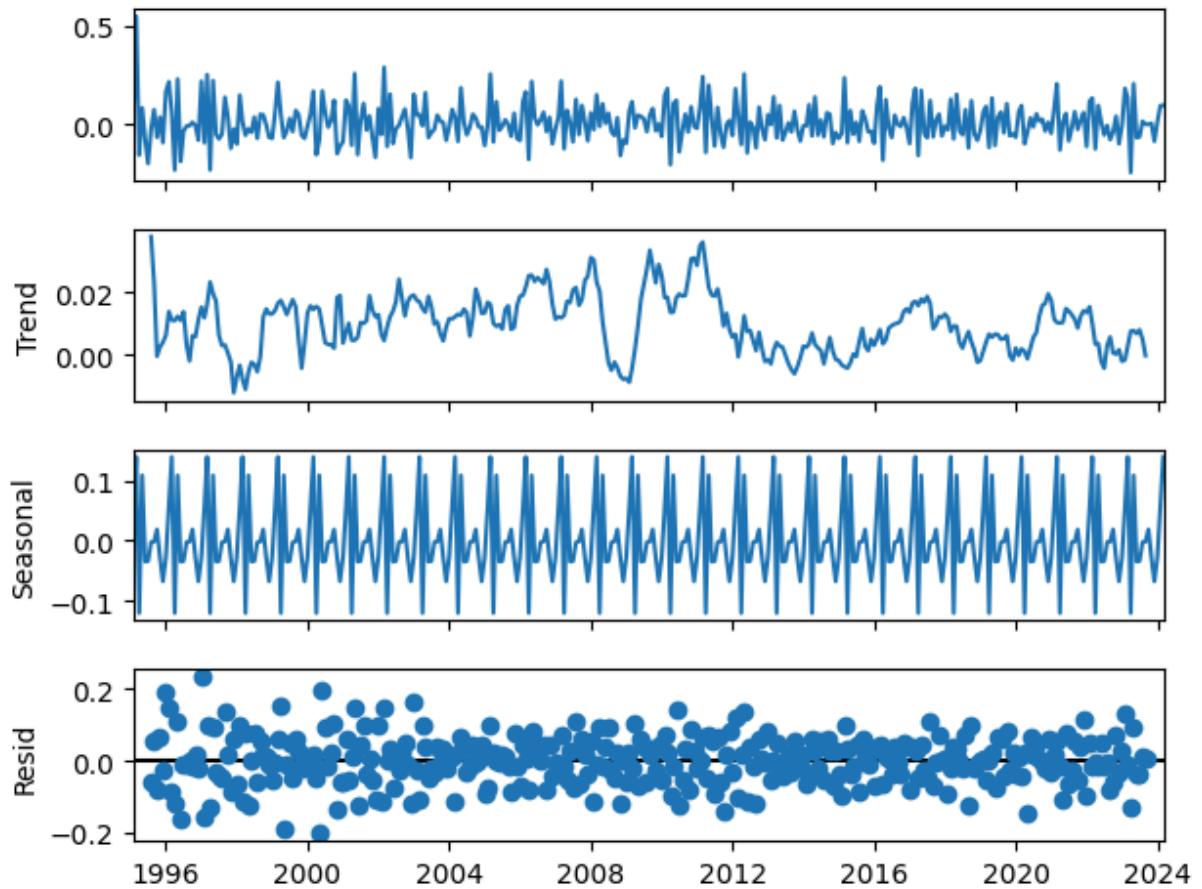
```
result = kpss(series, regression='c')
```

### Export Value (Agro-manufacturing\_Products)



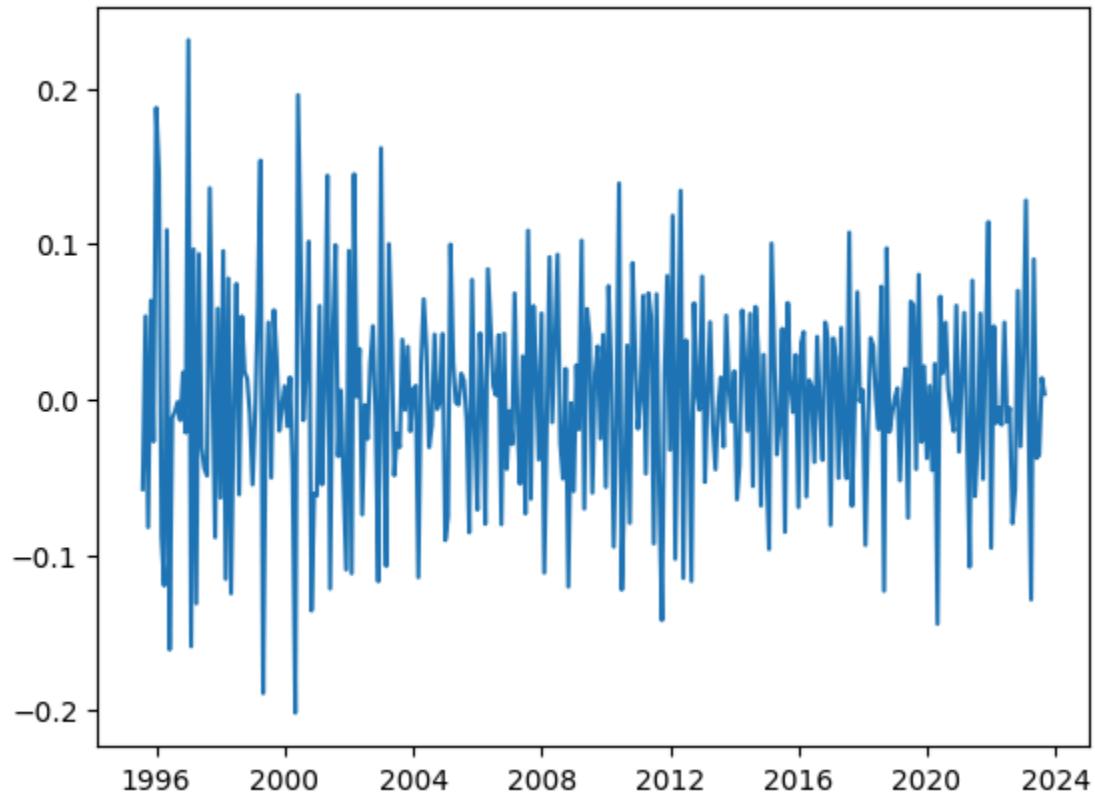
### Export Value % Change (Agro-manufacturing\_Products)

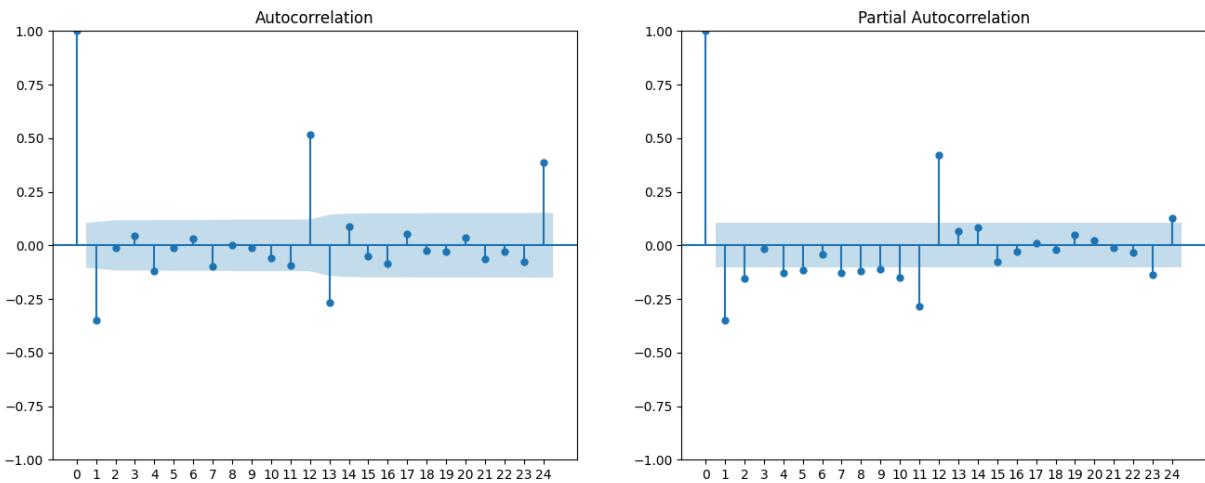




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

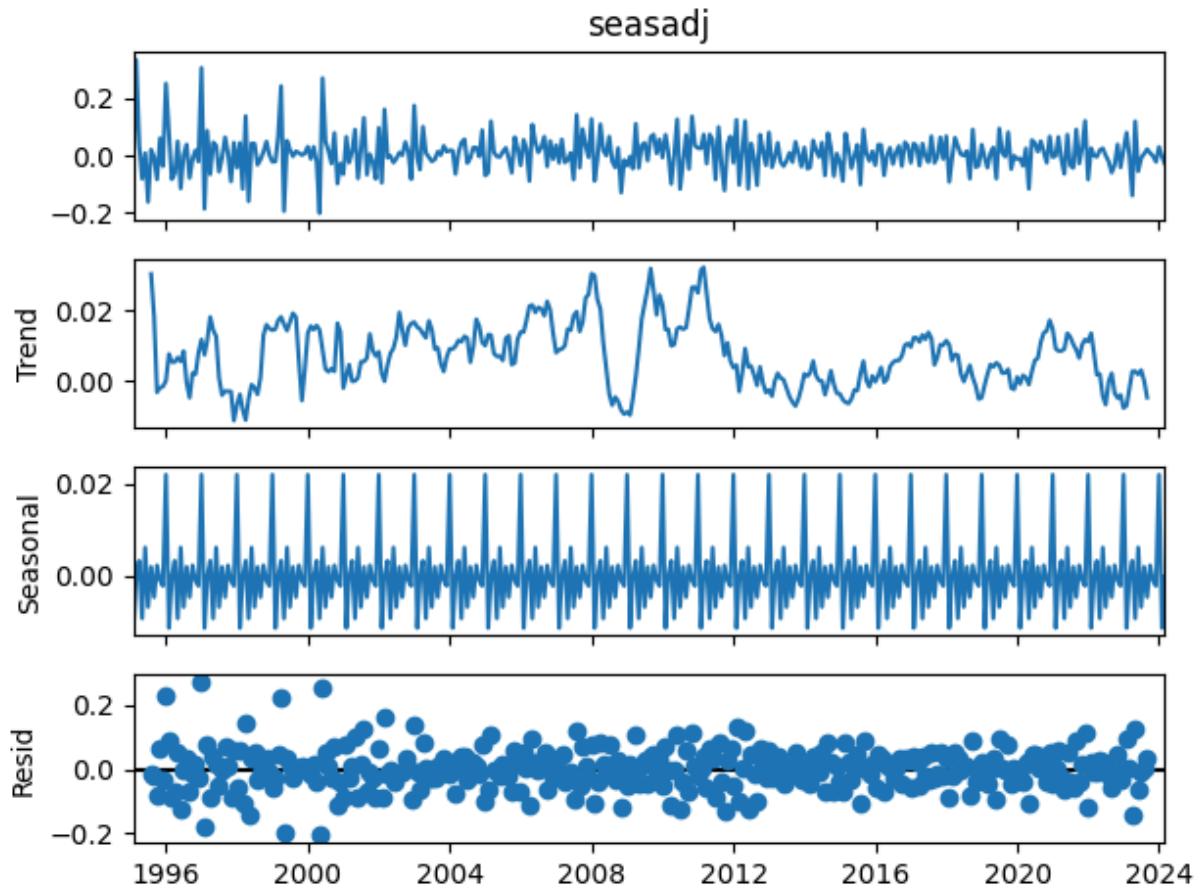
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -12.688965967388722
p-value: 1.1431670996247037e-23
Critical Value 1%: -3.449336554273722
Critical Value 5%: -2.8699055166063085
Critical Value 10%: -2.571226758215748
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.11825165935762255
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

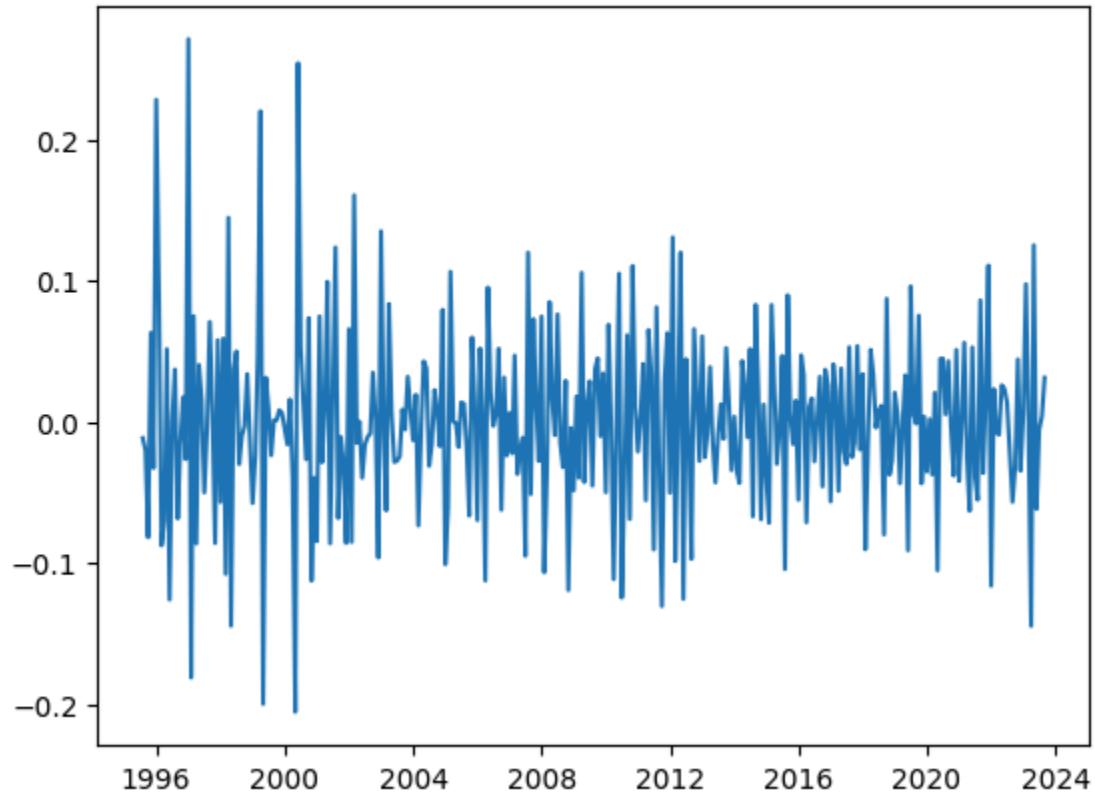
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

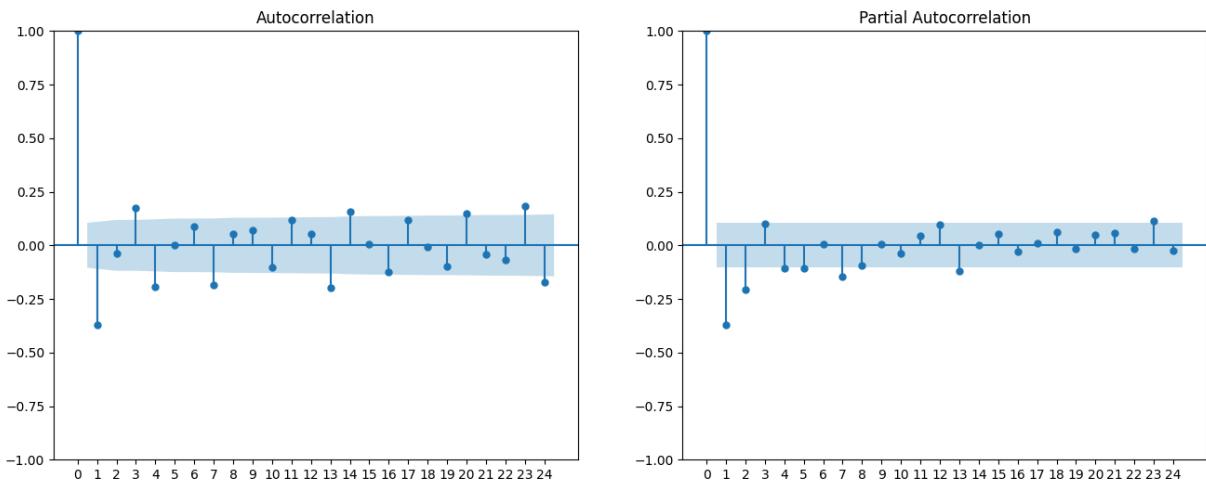
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals



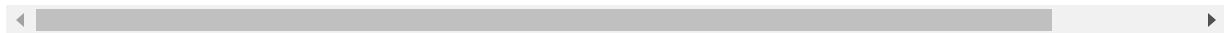


Time series assumptions are met.

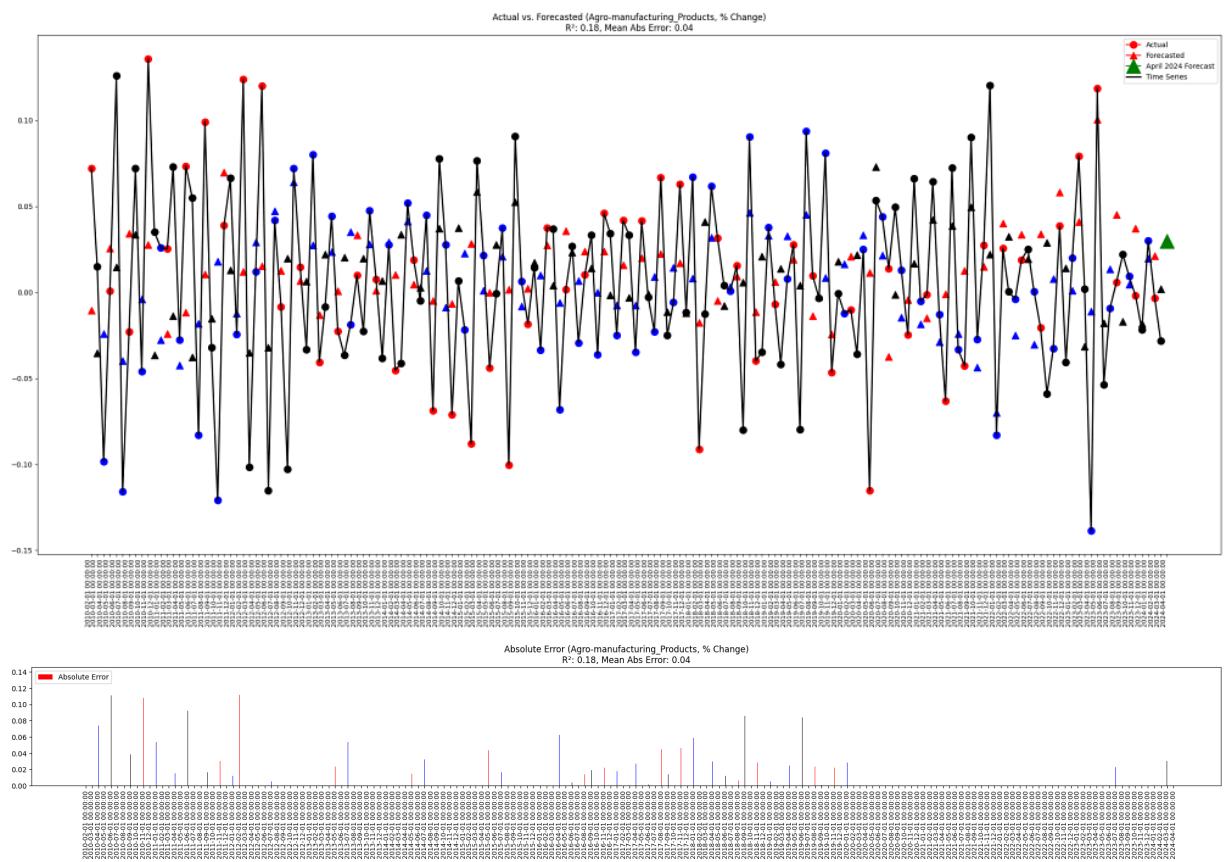
'Data of Actual vs Forecasted values with error metrics:'

| Agro-manufacturing_Products | actual    | forecast  | error     | abs_error | squared_error | abs_perc |
|-----------------------------|-----------|-----------|-----------|-----------|---------------|----------|
| 2010-02-01                  | 0.072132  | -0.010379 | 0.082512  | 0.082512  | 0.006808      |          |
| 2010-03-01                  | 0.014976  | -0.035356 | 0.050333  | 0.050333  | 0.002533      |          |
| 2010-04-01                  | -0.098247 | -0.024209 | -0.074038 | 0.074038  | 0.005482      |          |
| 2010-05-01                  | 0.000654  | 0.025574  | -0.024920 | 0.024920  | 0.000621      |          |
| 2010-06-01                  | 0.125882  | 0.014769  | 0.111113  | 0.111113  | 0.012346      |          |
| ...                         | ...       | ...       | ...       | ...       | ...           | ...      |
| 2023-12-01                  | -0.021761 | -0.018262 | -0.003499 | 0.003499  | 0.000012      |          |
| 2024-01-01                  | 0.029965  | 0.019750  | 0.010215  | 0.010215  | 0.000104      |          |
| 2024-02-01                  | -0.003369 | 0.021053  | -0.024422 | 0.024422  | 0.000596      |          |
| 2024-03-01                  | -0.028102 | 0.002008  | -0.030110 | 0.030110  | 0.000907      |          |
| 2024-04-01                  | NaN       | 0.029856  | NaN       | NaN       | NaN           |          |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.039269316077435154'



## Apparels\_and\_Textile\_Materials

```
In [51]: df = df_export_ANALYSIS.copy()
name = df.columns[18]
display(f"Component: {name}")
```

'Component: Apparels\_and\_Textile\_Materials'

```
In [52]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value.pkl', 'rb') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

class Apparels\_and\_Textile\_Materials

|            |           |
|------------|-----------|
| 1995-02-01 | -0.056700 |
| 1995-03-01 | 0.255345  |
| 1995-04-01 | -0.278537 |
| 1995-05-01 | 0.129097  |
| 1995-06-01 | 0.184119  |
| ...        | ...       |
| 2023-11-01 | -0.018454 |
| 2023-12-01 | -0.030931 |
| 2024-01-01 | -0.000262 |
| 2024-02-01 | 0.049899  |
| 2024-03-01 | 0.071320  |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'

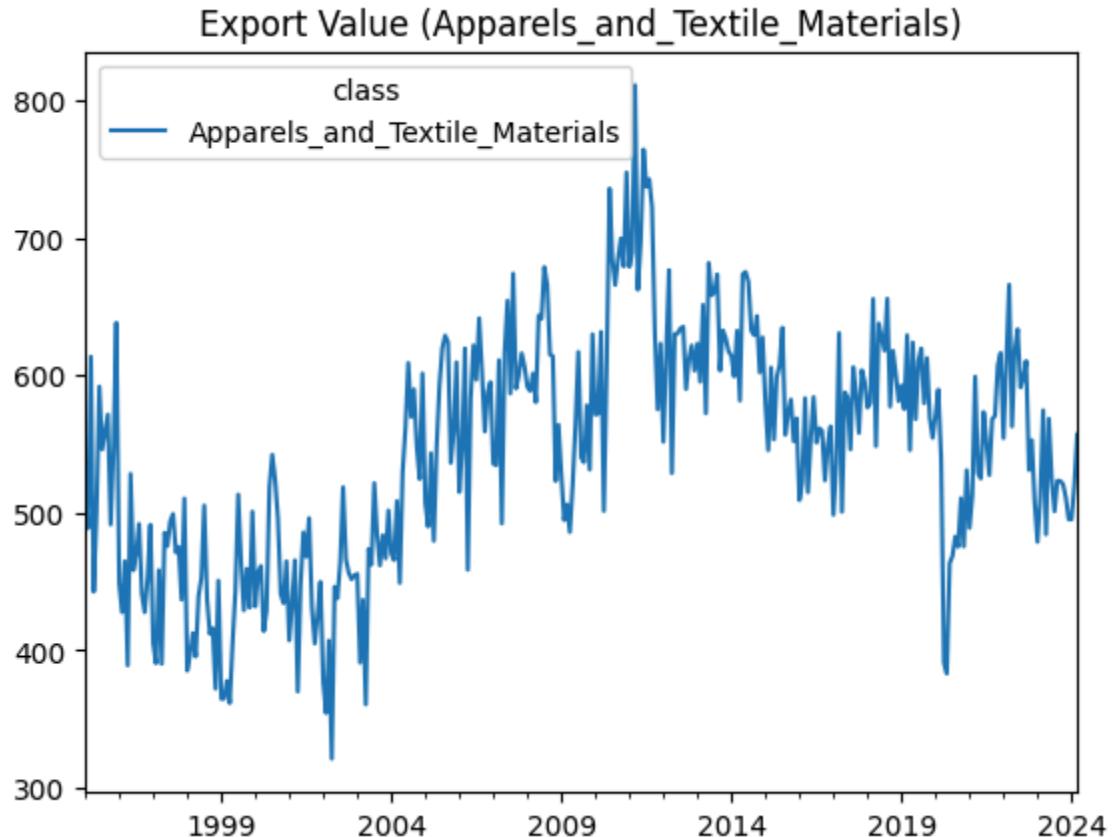
```
Running ADF Test for Stationarity...
ADF Statistic: -5.186389785489652
p-value: 9.372376904097357e-06
Critical Value 1%: -3.4502011472639724
Critical Value 5%: -2.8702852297358983
Critical Value 10%: -2.5714292194077513
is_stationary: True
```

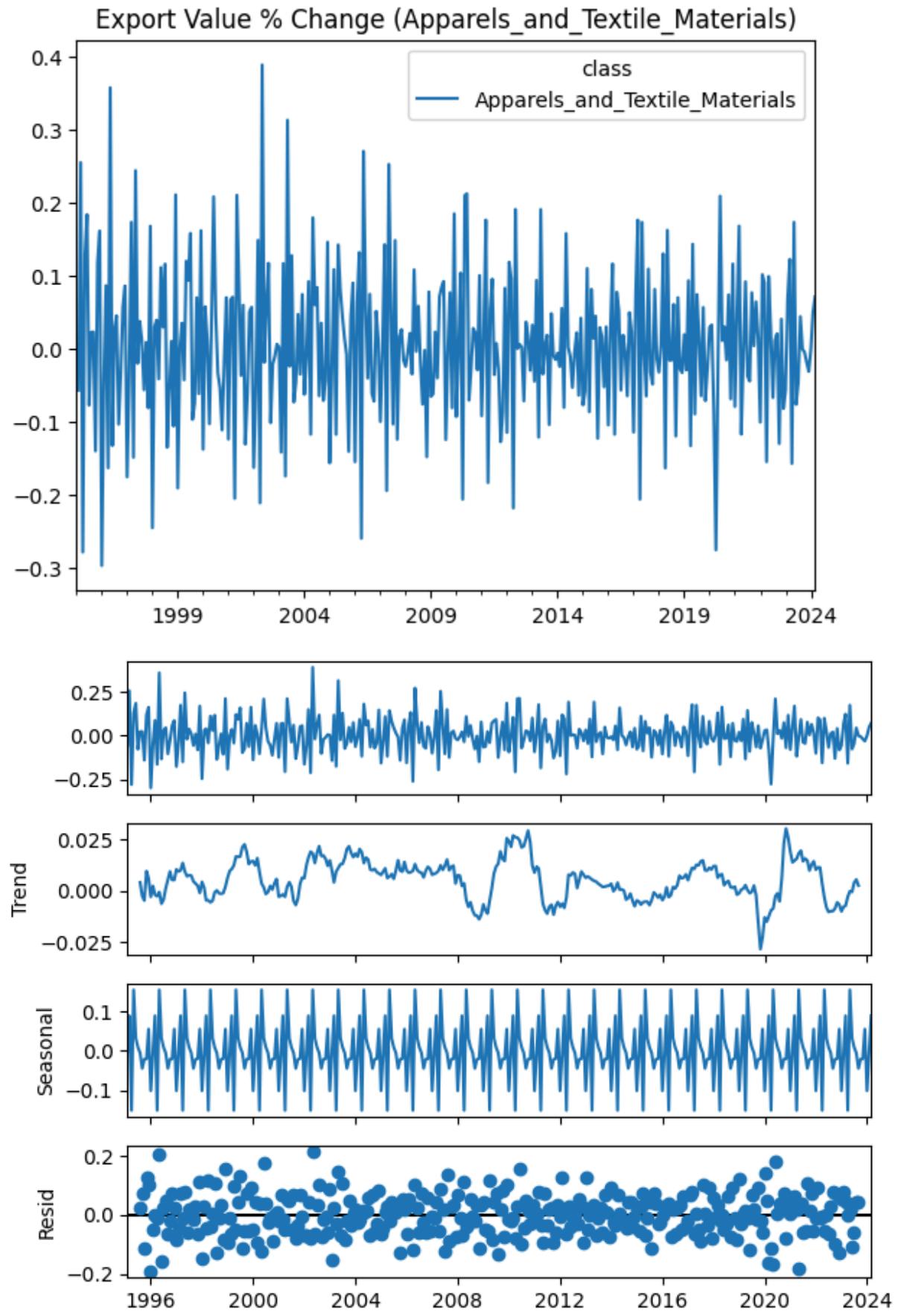
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.06690658656002833
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

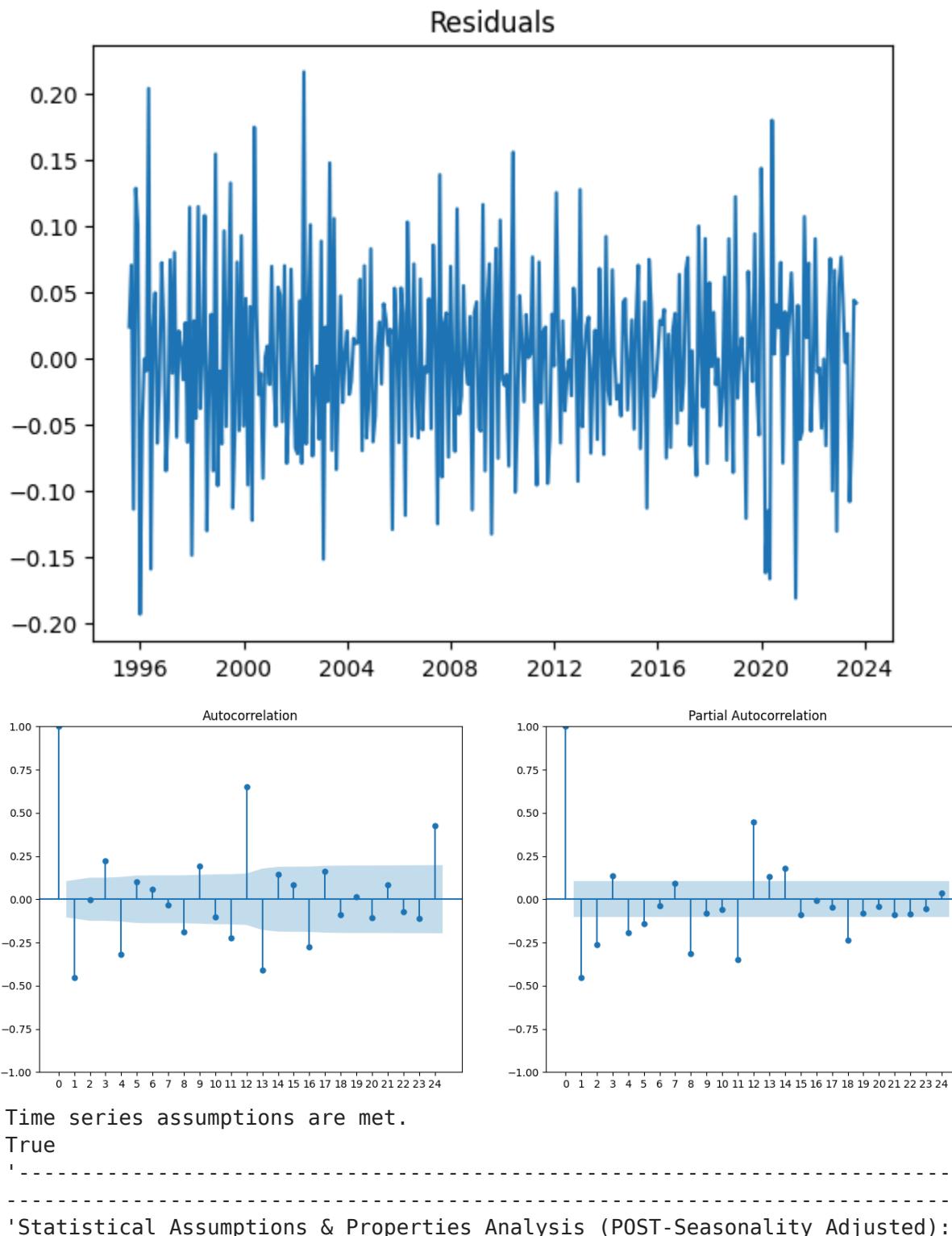
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```





Plotting Residuals for Homoscedasticity Check...



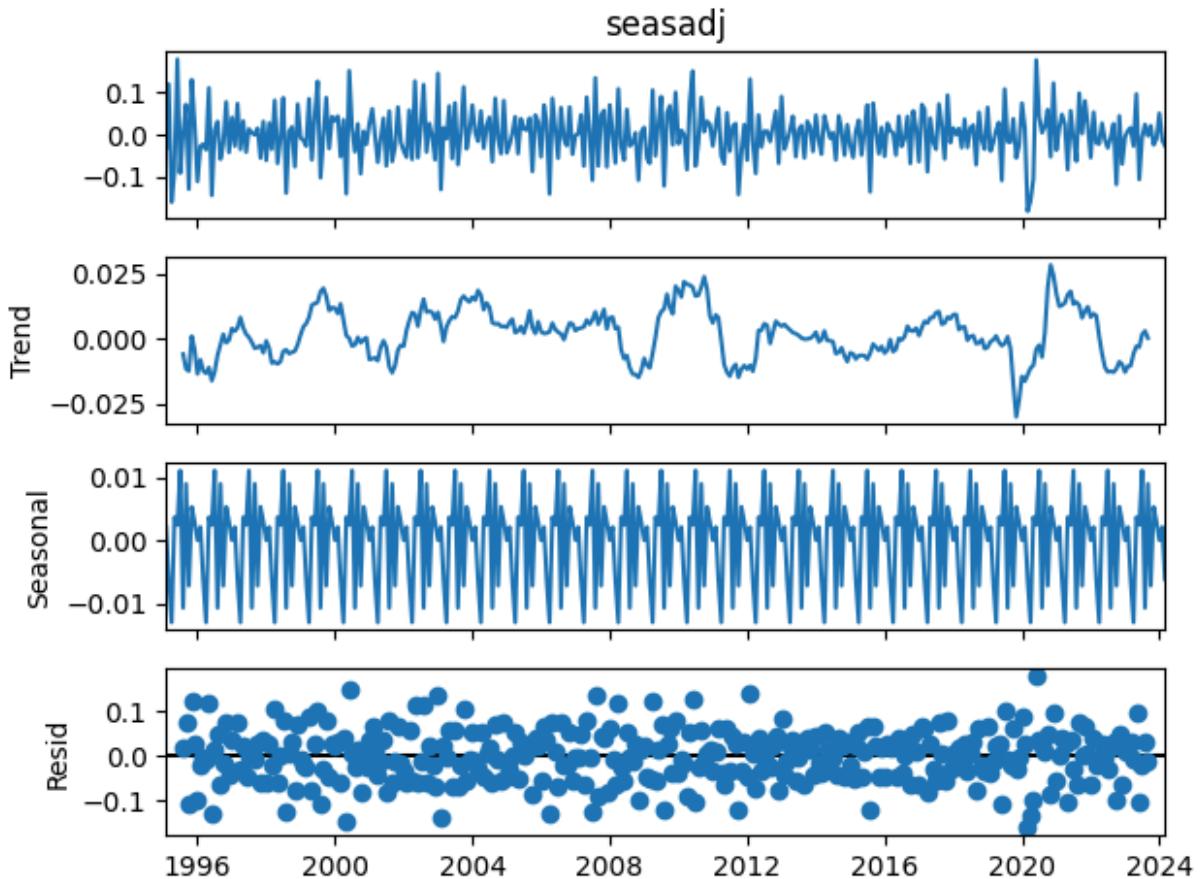
Running ADF Test for Stationarity...  
 ADF Statistic: -6.230344885375834  
 p-value: 4.971084457502175e-08  
 Critical Value 1%: -3.4499043309021955  
 Critical Value 5%: -2.870154885338925  
 Critical Value 10%: -2.5713597190254385  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.06702418059346323  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

Decomposing the Series...

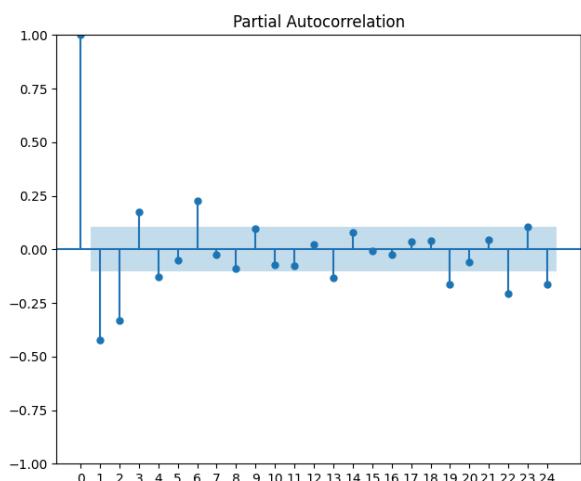
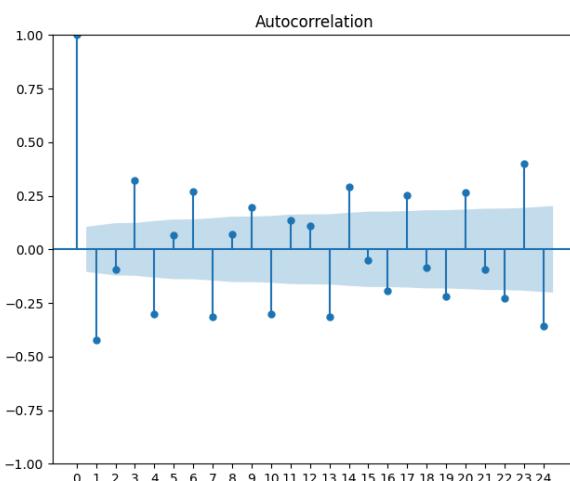
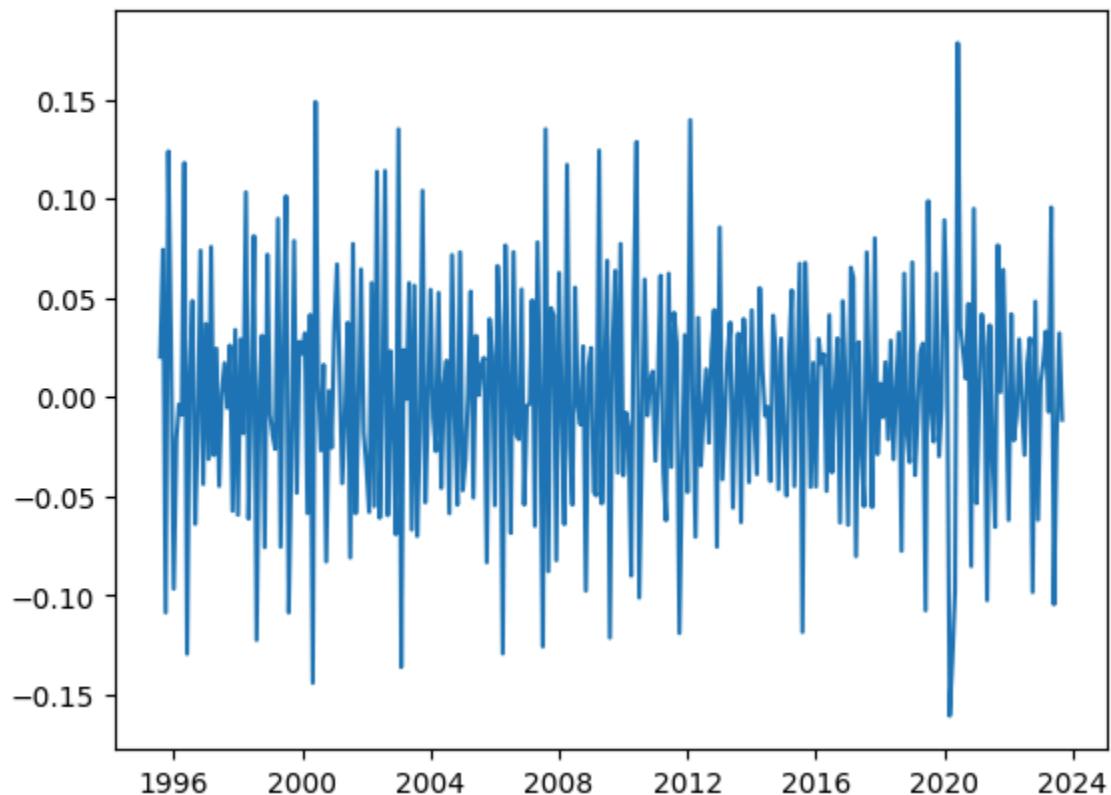
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals



Time series assumptions are met.

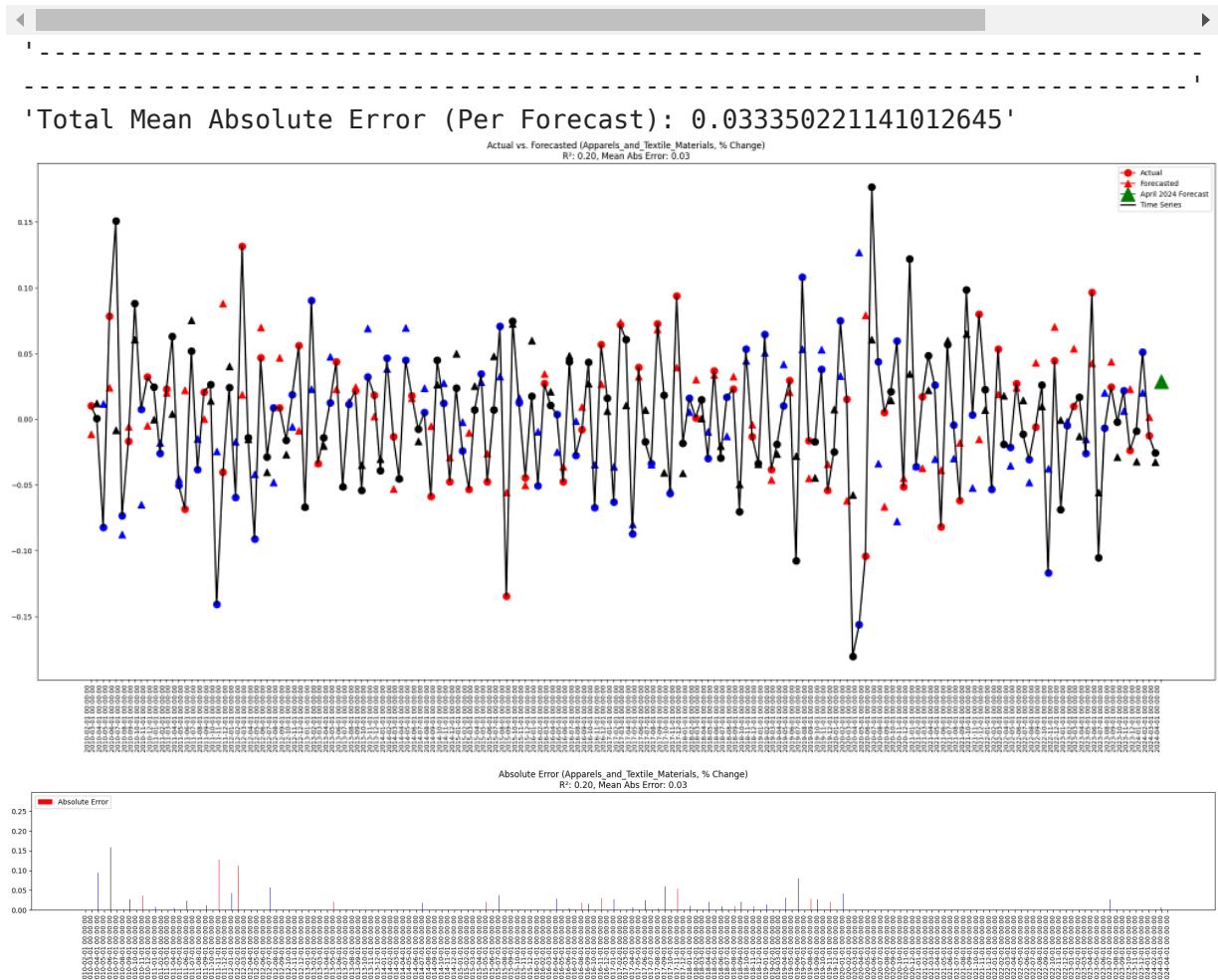
'-----'

'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Apparels_and_Textile_Materials |            | actual    | forecast  | error     | abs_error | squared_error | al  |
|--------------------------------|------------|-----------|-----------|-----------|-----------|---------------|-----|
|                                | 2010-02-01 | 0.010372  | -0.011327 | 0.021699  | 0.021699  | 0.000471      |     |
|                                | 2010-03-01 | 0.000502  | 0.012041  | -0.011539 | 0.011539  | 0.000133      |     |
|                                | 2010-04-01 | -0.082054 | 0.011758  | -0.093812 | 0.093812  | 0.008801      |     |
|                                | 2010-05-01 | 0.078445  | 0.024199  | 0.054245  | 0.054245  | 0.002943      |     |
|                                | 2010-06-01 | 0.150796  | -0.008146 | 0.158942  | 0.158942  | 0.025263      |     |
|                                | ...        | ...       | ...       | ...       | ...       | ...           | ... |
|                                | 2023-12-01 | -0.008984 | -0.032110 | 0.023127  | 0.023127  | 0.000535      |     |
|                                | 2024-01-01 | 0.050969  | 0.020371  | 0.030599  | 0.030599  | 0.000936      |     |
|                                | 2024-02-01 | -0.012636 | 0.001668  | -0.014304 | 0.014304  | 0.000205      |     |
|                                | 2024-03-01 | -0.025613 | -0.032469 | 0.006856  | 0.006856  | 0.000047      |     |
|                                | 2024-04-01 | NaN       | 0.028775  | NaN       | NaN       | NaN           | NaN |

171 rows × 6 columns



## Footware\_and\_parts

```
In [53]: df = df_export_ANALYSIS.copy()
name = df.columns[19]
display(f"Component: {name}")

'Component: Footware_and_parts'
```

```
In [54]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Footware_and_parts |
|------------|--------------------|
| 1995-02-01 | -0.000056          |
| 1995-03-01 | 0.002623           |
| 1995-04-01 | -0.245686          |
| 1995-05-01 | 0.461924           |
| 1995-06-01 | 0.113668           |
| ...        | ...                |
| 2023-11-01 | 0.118204           |
| 2023-12-01 | -0.182368          |
| 2024-01-01 | 0.054774           |
| 2024-02-01 | -0.024561          |
| 2024-03-01 | 0.012710           |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -6.764106715119824
p-value: 2.744858927803993e-09
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

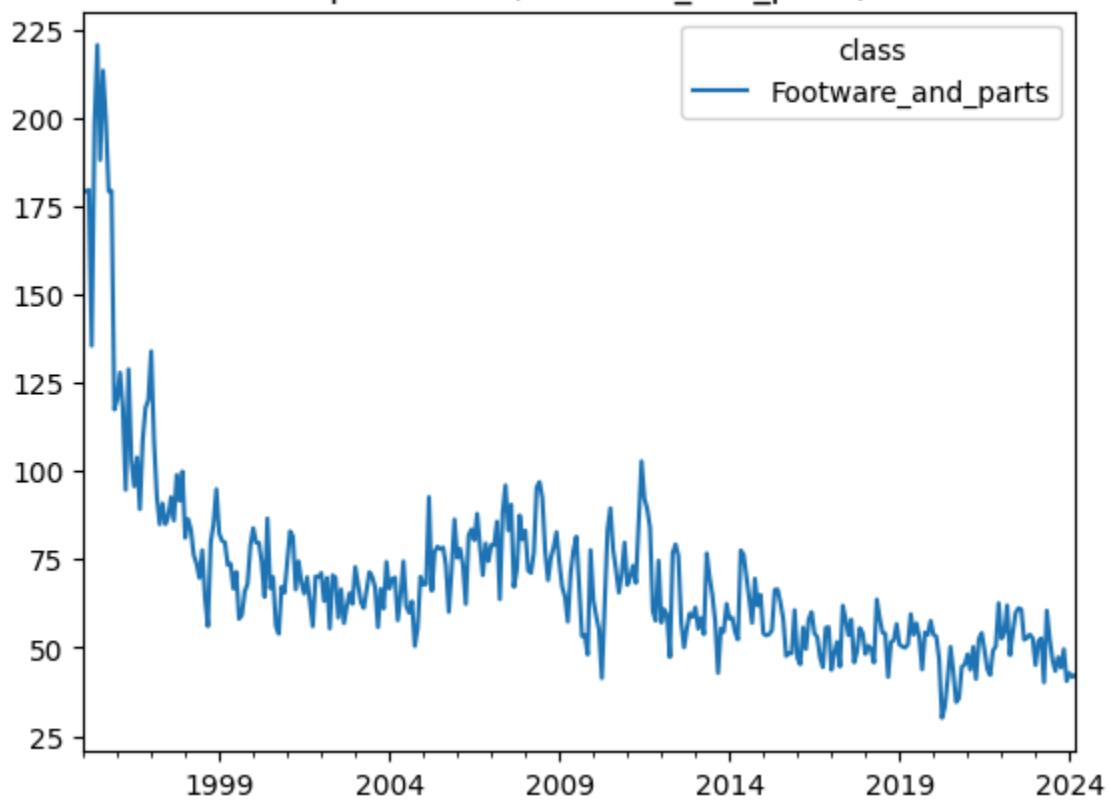
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.18203851145458144
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

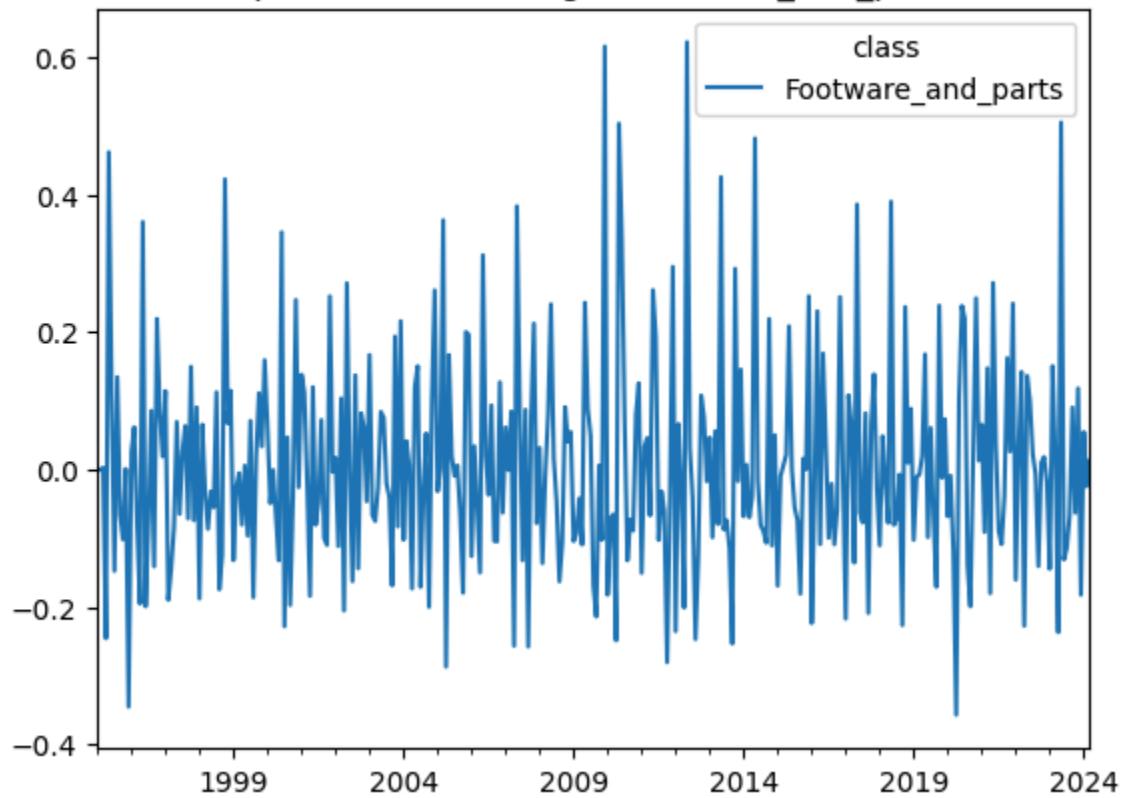
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

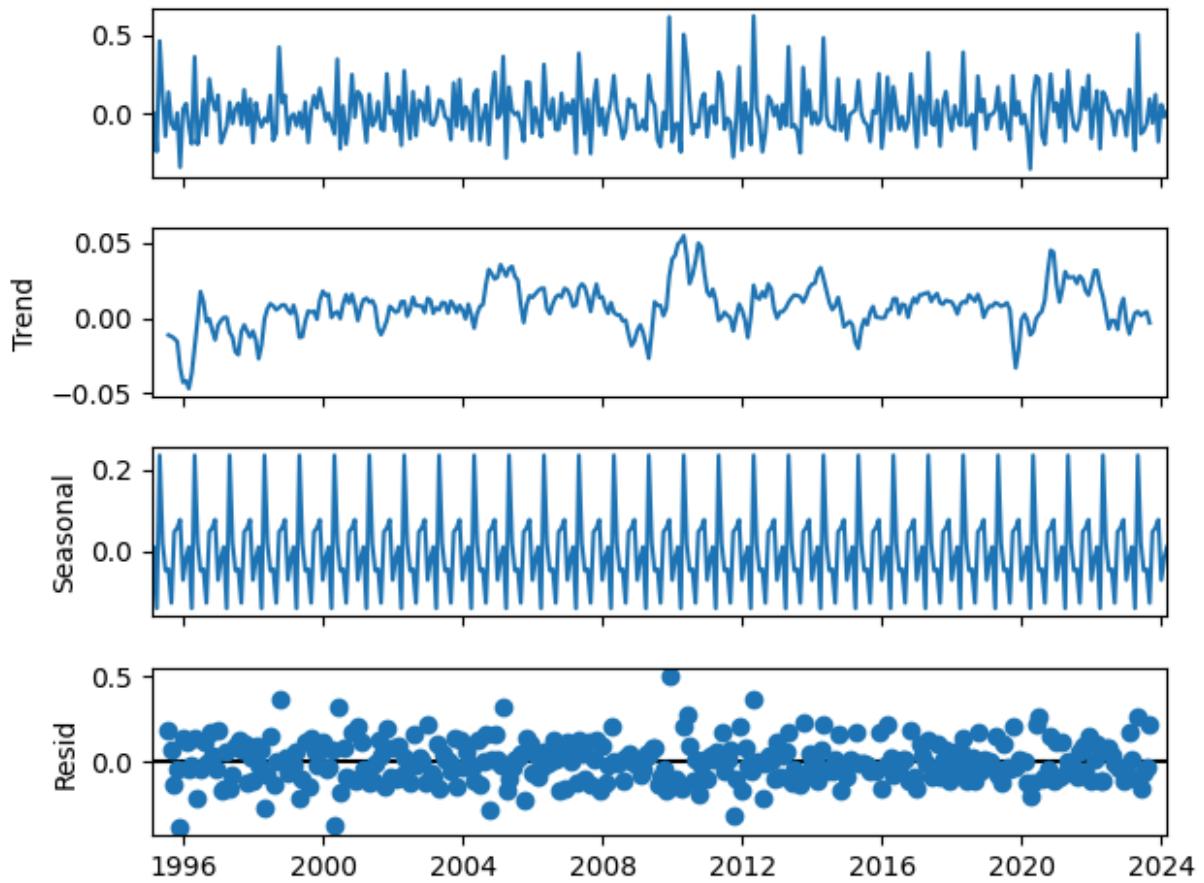
```
result = kpss(series, regression='c')
```

## Export Value (Footware\_and\_parts)

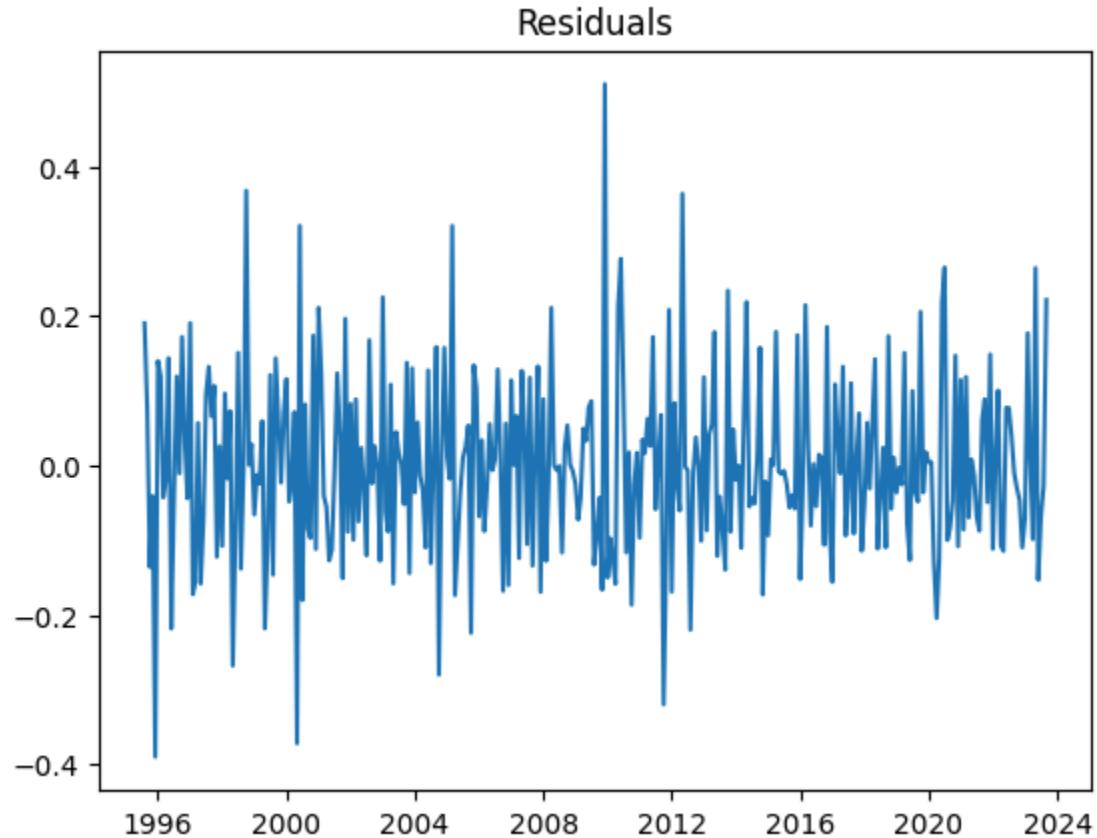


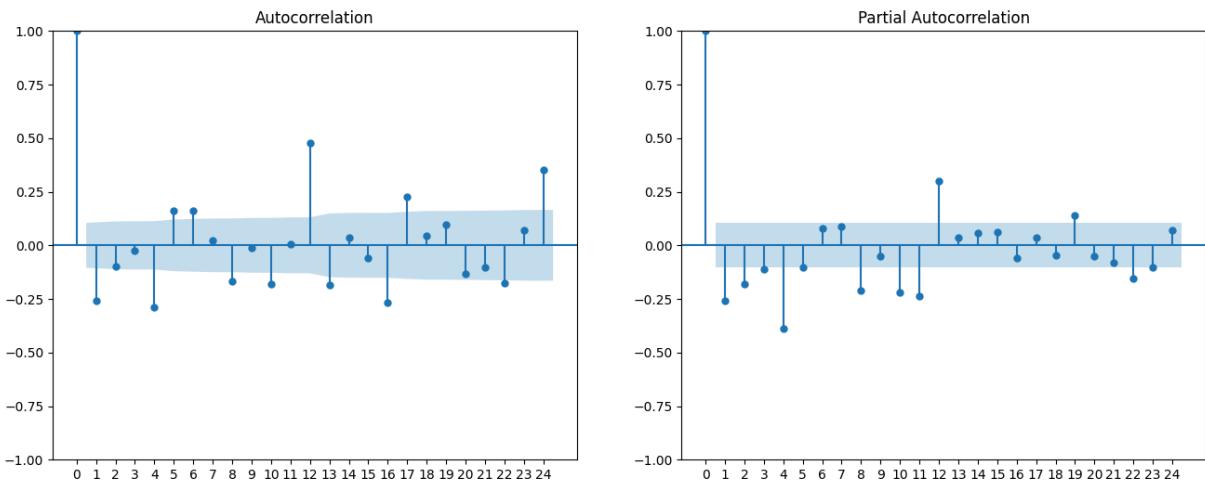
## Export Value % Change (Footware\_and\_parts)





Plotting Residuals for Homoscedasticity Check...





Time series assumptions are met.

True

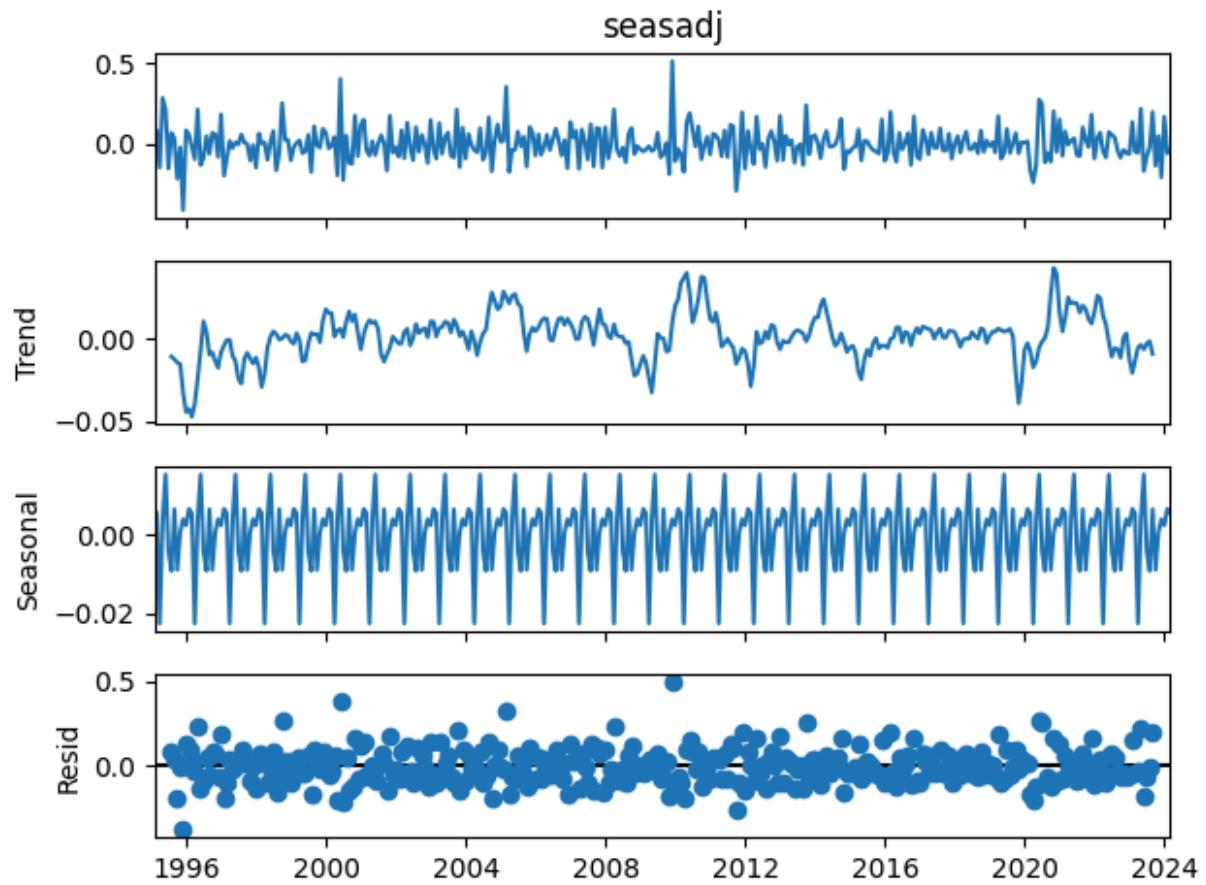
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted): '
Running ADF Test for Stationarity...
ADF Statistic: -5.450313610670317
p-value: 2.653430284197973e-06
Critical Value 1%: -3.450081345901191
Critical Value 5%: -2.870232621465807
Critical Value 10%: -2.571401168202517
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.06377310439331033
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

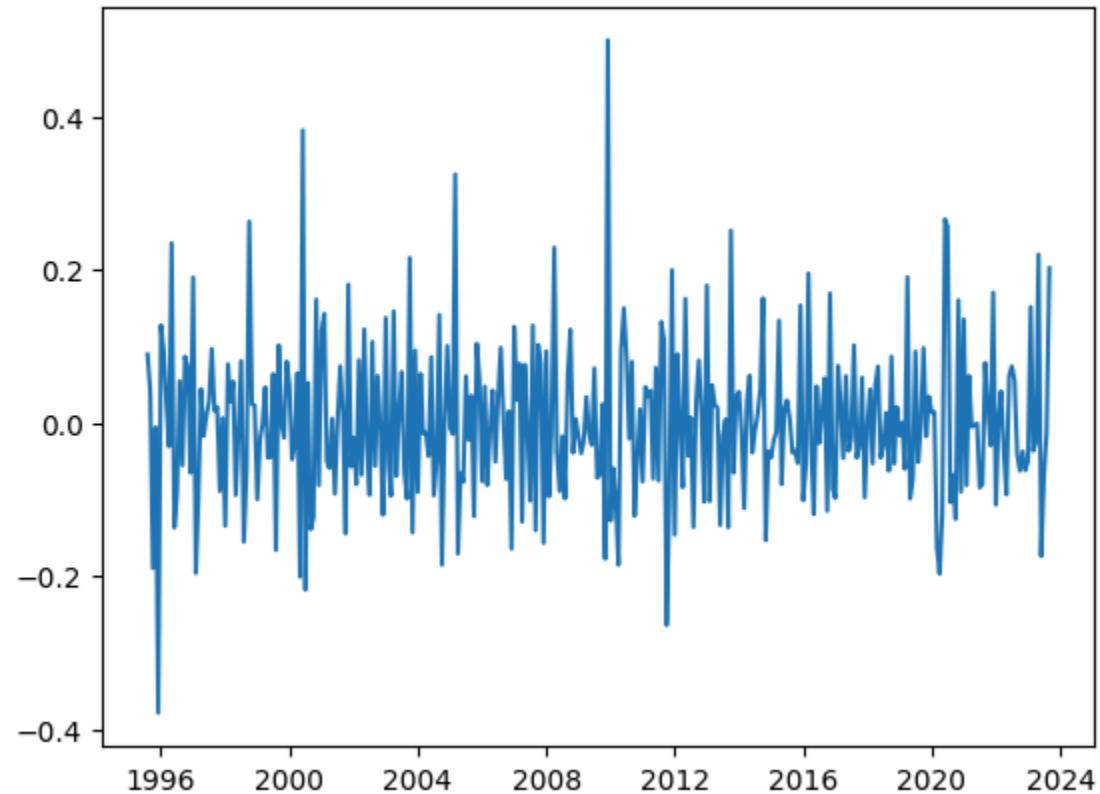
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

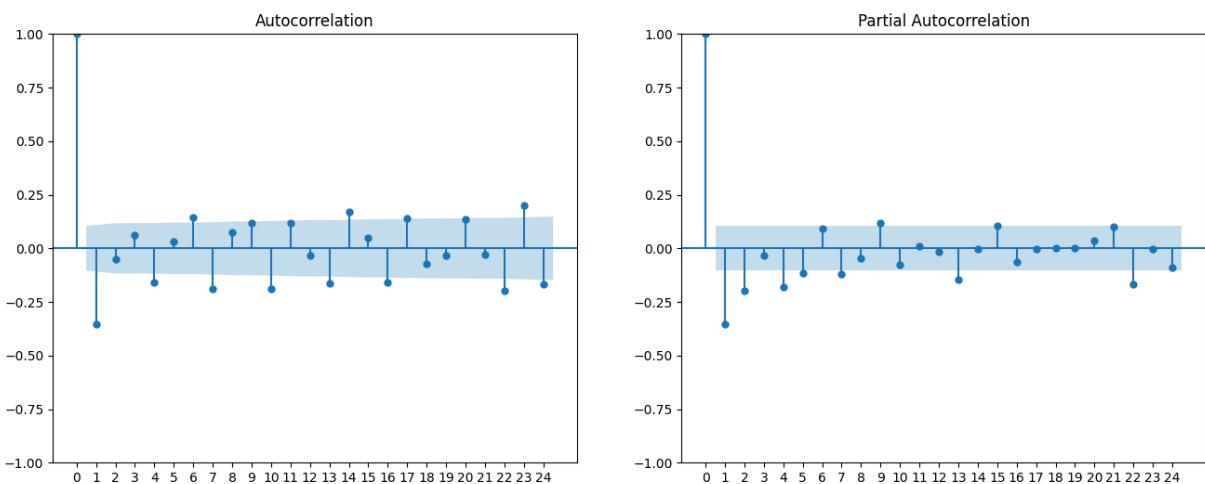
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals



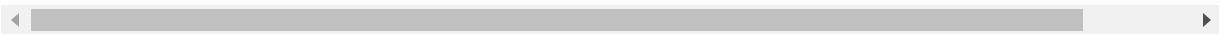


Time series assumptions are met.

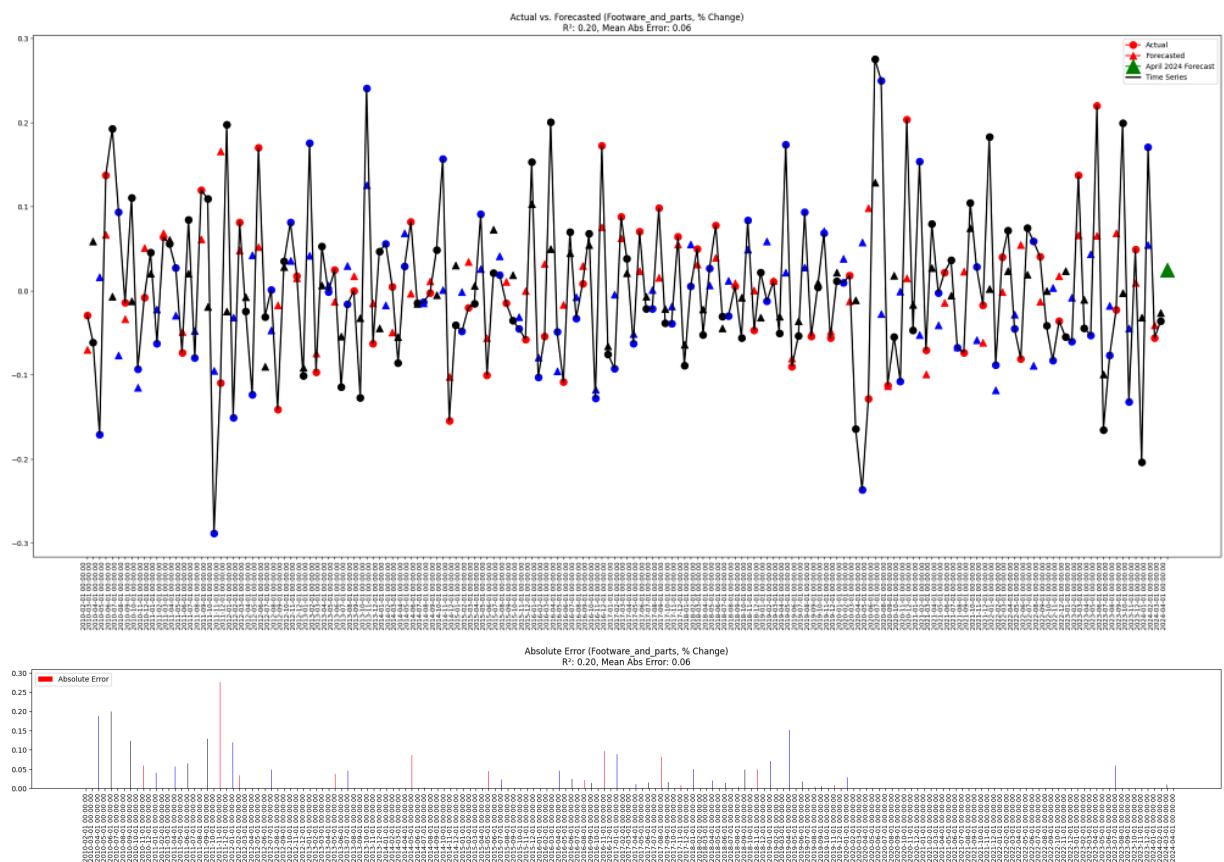
'Data of Actual vs Forecasted values with error metrics:'

| Footware_and_parts | actual    | forecast  | error     | abs_error | squared_error | abs_percenta |
|--------------------|-----------|-----------|-----------|-----------|---------------|--------------|
| 2010-02-01         | -0.029065 | -0.069726 | 0.040660  | 0.040660  | 0.001653      |              |
| 2010-03-01         | -0.061487 | 0.058625  | -0.120112 | 0.120112  | 0.014427      |              |
| 2010-04-01         | -0.171117 | 0.016186  | -0.187303 | 0.187303  | 0.035083      |              |
| 2010-05-01         | 0.137529  | 0.067008  | 0.070521  | 0.070521  | 0.004973      |              |
| 2010-06-01         | 0.192532  | -0.006885 | 0.199417  | 0.199417  | 0.039767      |              |
| ...                | ...       | ...       | ...       | ...       | ...           | ...          |
| 2023-12-01         | -0.203922 | -0.031905 | -0.172018 | 0.172018  | 0.029590      |              |
| 2024-01-01         | 0.170459  | 0.054649  | 0.115810  | 0.115810  | 0.013412      |              |
| 2024-02-01         | -0.056073 | -0.040847 | -0.015227 | 0.015227  | 0.000232      |              |
| 2024-03-01         | -0.036202 | -0.026262 | -0.009940 | 0.009940  | 0.000099      |              |
| 2024-04-01         | NaN       | 0.024702  | NaN       | NaN       | NaN           |              |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.064386609035526'



## Electronics

```
In [55]: df = df_export_ANALYSIS.copy()
name = df.columns[20]
display(f"Component: {name}")
```

'Component: Electronics'

```
In [56]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

class Electronics

|            |           |
|------------|-----------|
| 1995-02-01 | 0.035321  |
| 1995-03-01 | 0.271773  |
| 1995-04-01 | -0.144608 |
| 1995-05-01 | 0.128309  |
| 1995-06-01 | 0.038584  |
| ...        | ...       |
| 2023-11-01 | 0.104044  |
| 2023-12-01 | 0.083406  |
| 2024-01-01 | -0.138897 |
| 2024-02-01 | -0.053334 |
| 2024-03-01 | 0.206038  |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

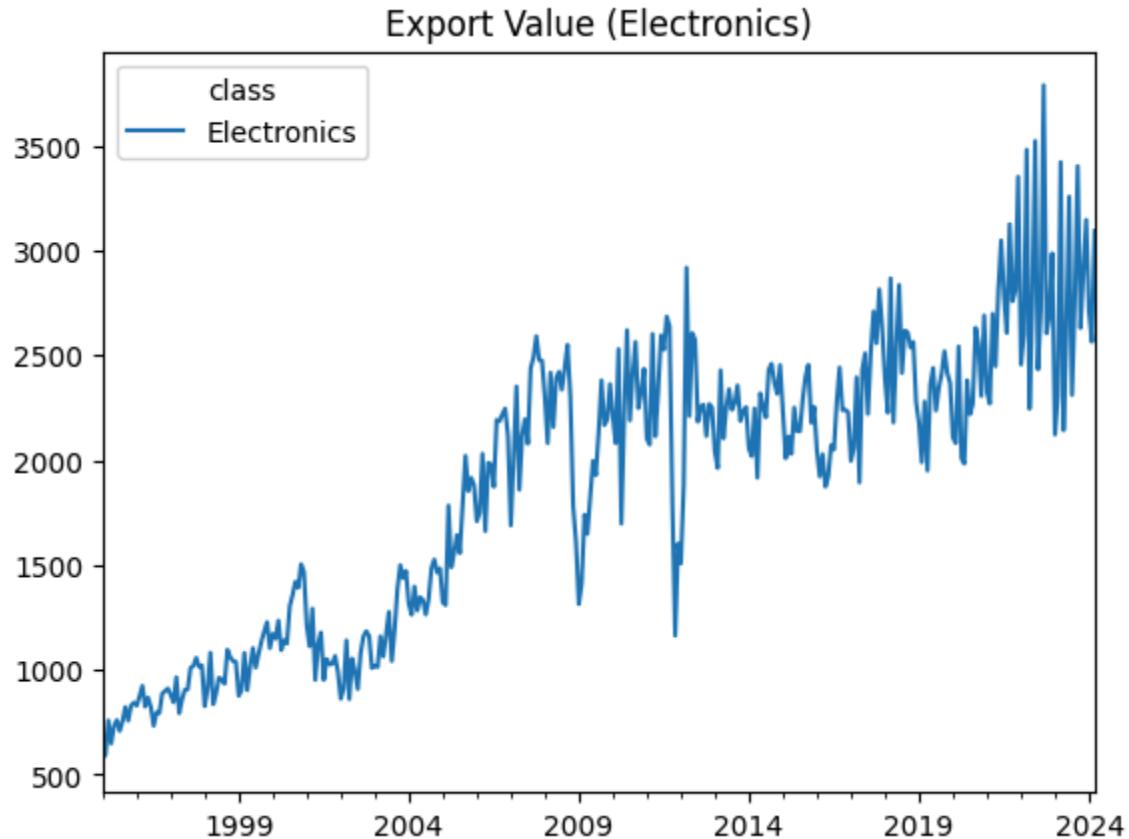
```
Running ADF Test for Stationarity...
ADF Statistic: -5.65763365900667
p-value: 9.53981132439007e-07
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

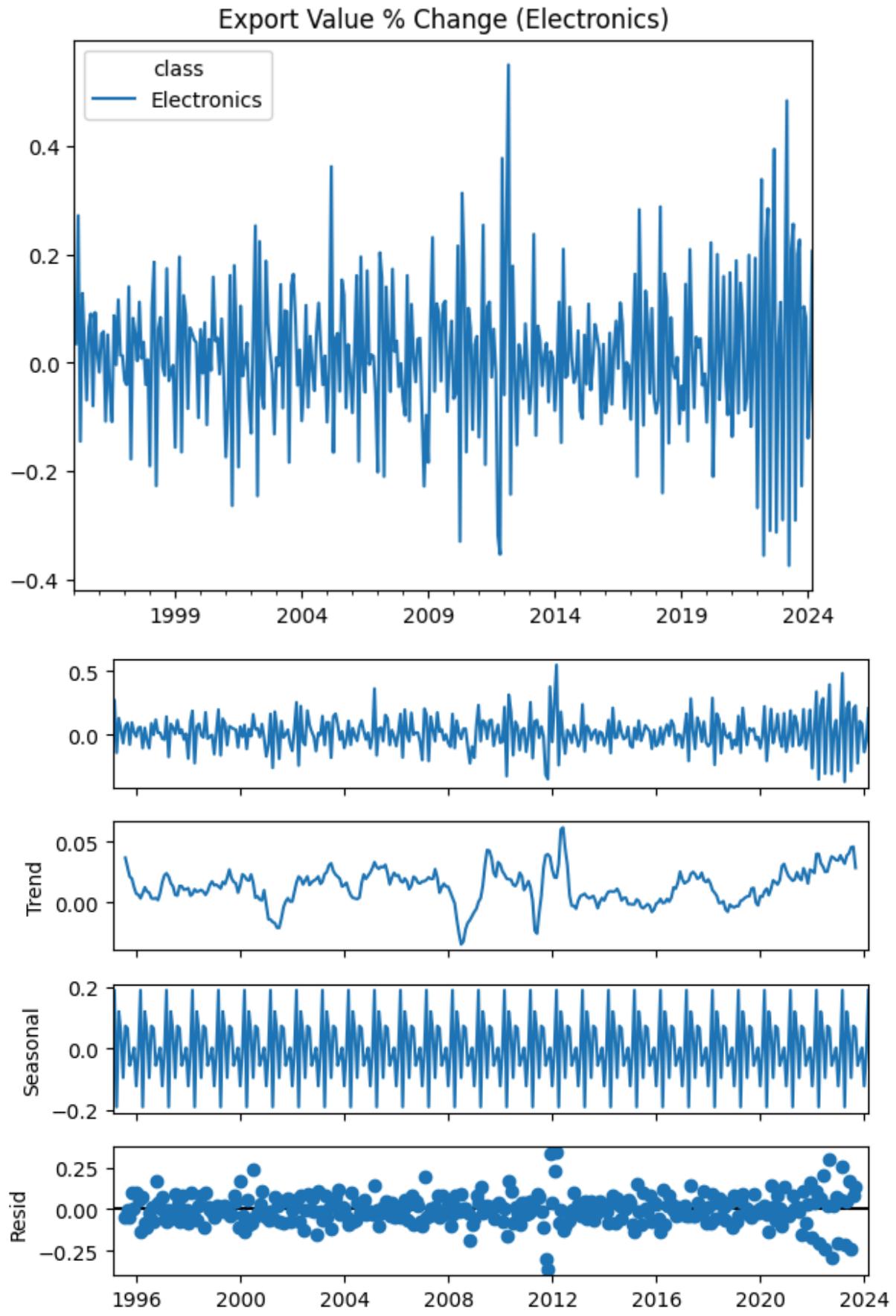
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.09276156749813826
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

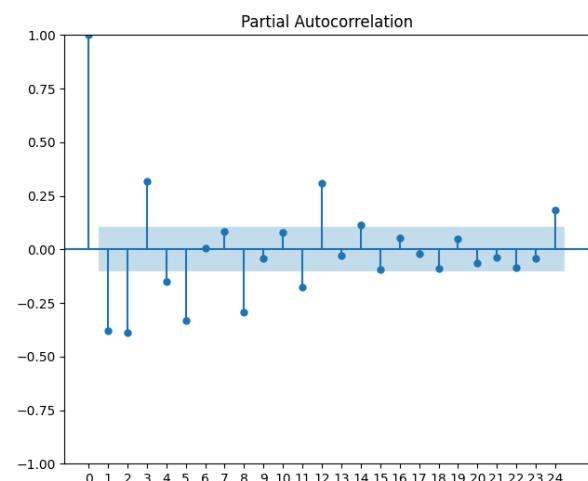
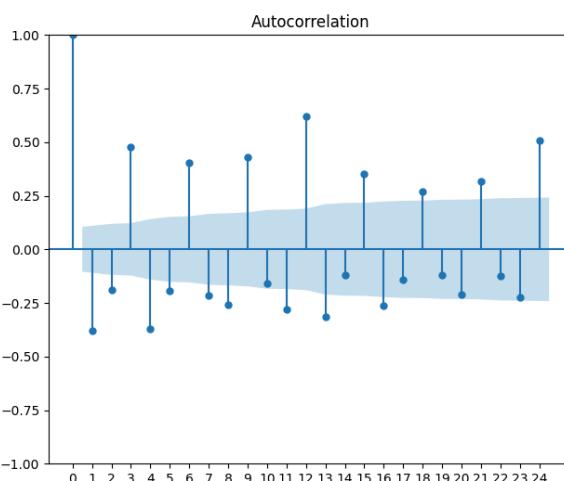
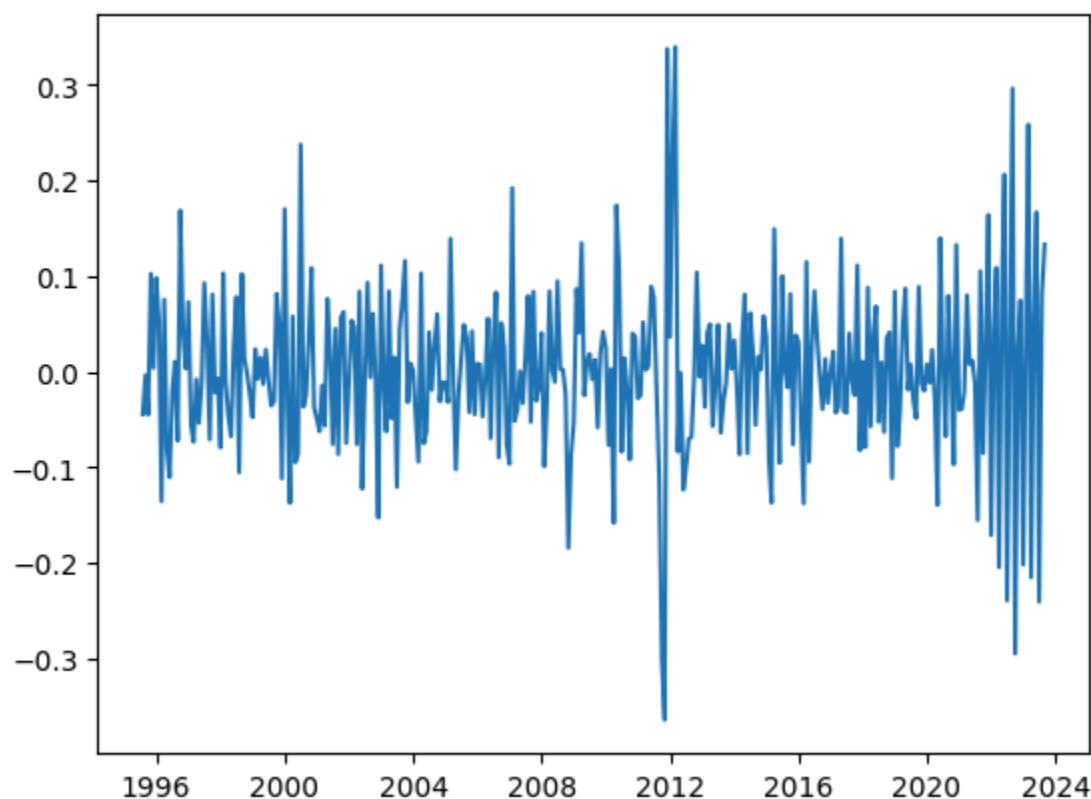
```
result = kpss(series, regression='c')
```





Plotting Residuals for Homoscedasticity Check...

### Residuals



Time series assumptions are met.

True

'-----'  
-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

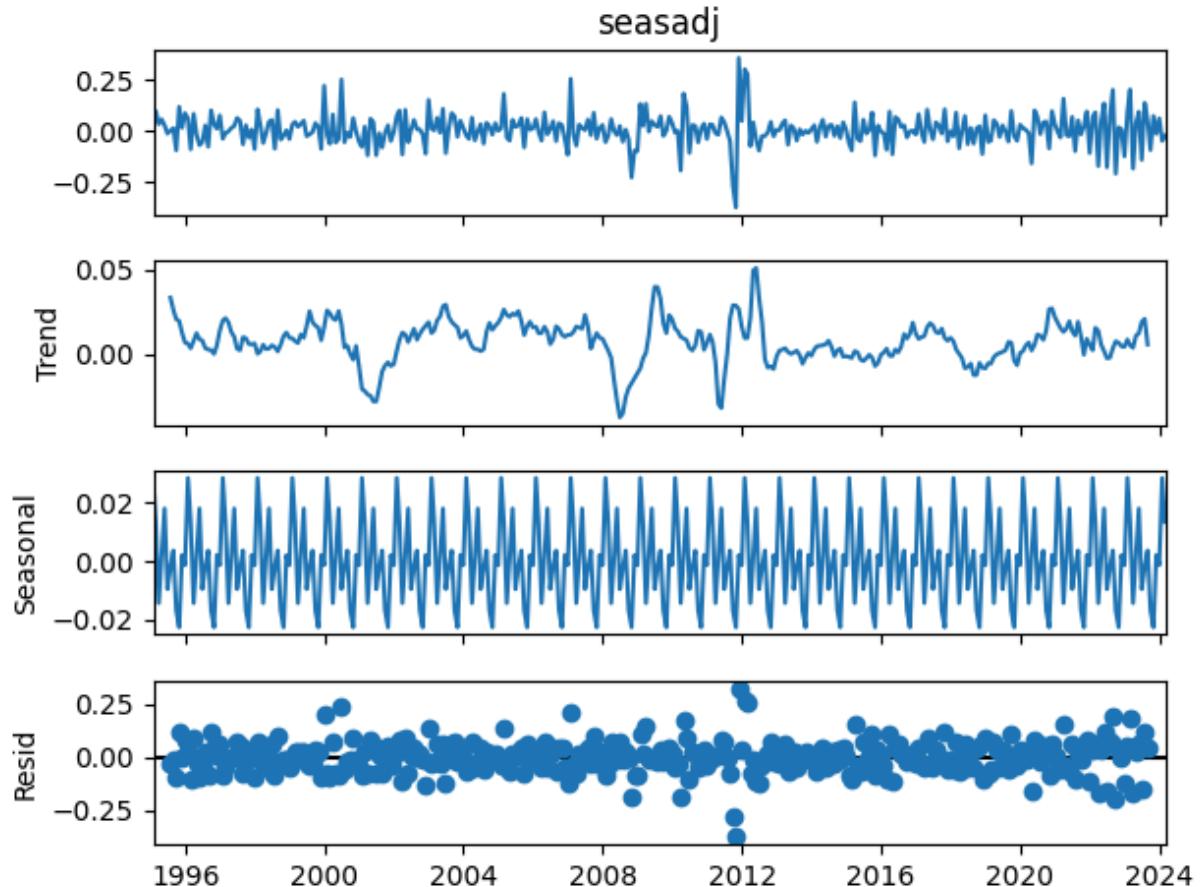
```
Running ADF Test for Stationarity...
ADF Statistic: -12.070857112772934
p-value: 2.349674281856762e-22
Critical Value 1%: -3.4494474563375737
Critical Value 5%: -2.8699542285903887
Critical Value 10%: -2.5712527305187987
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.07356333060860401
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

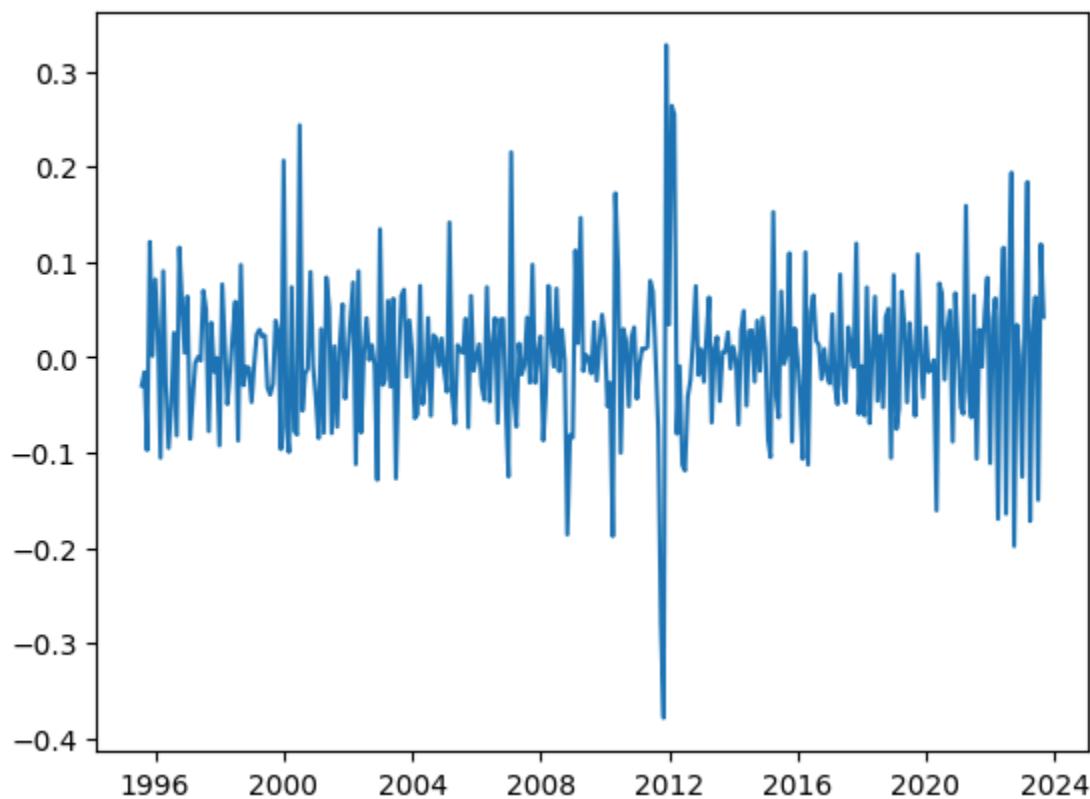
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

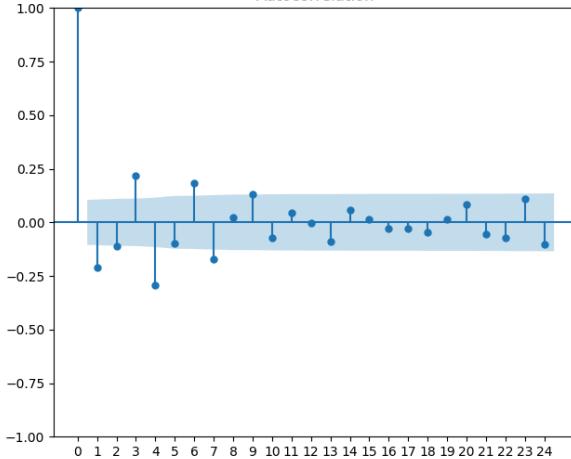


Plotting Residuals for Homoscedasticity Check...

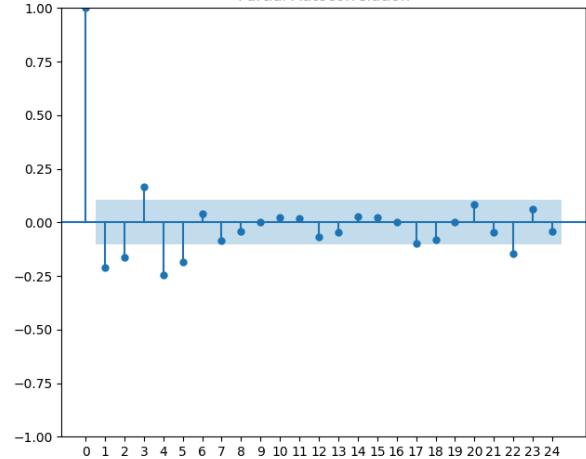
### Residuals



Autocorrelation



Partial Autocorrelation



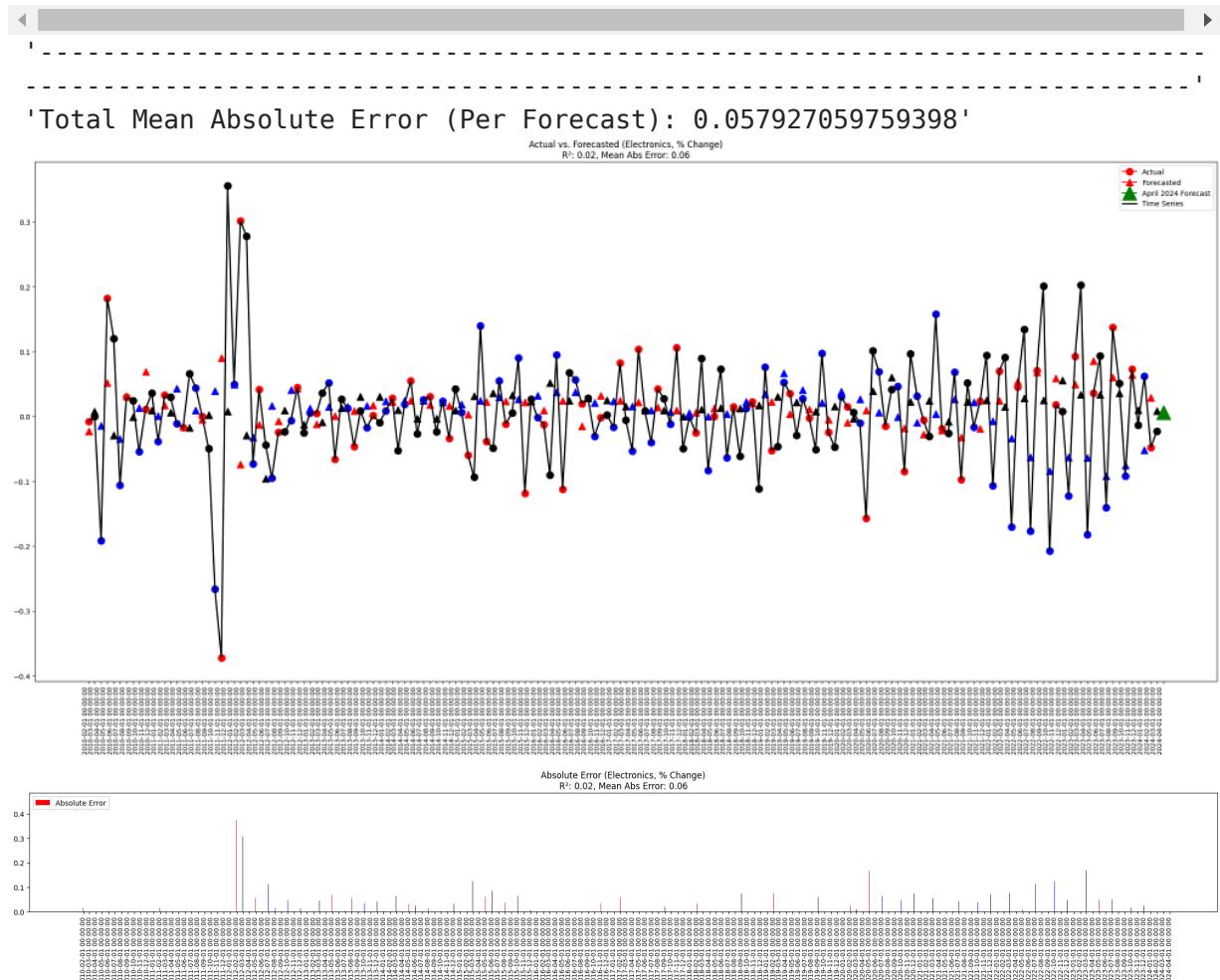
Time series assumptions are met.

'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Electronics | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|-------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01  | -0.008094 | -0.022563 | 0.014469  | 0.014469  | 0.000209      | 1.787653             |
| 2010-03-01  | -0.000143 | 0.007742  | -0.007886 | 0.007886  | 0.000062      | 55.019527            |
| 2010-04-01  | -0.192072 | -0.014231 | -0.177841 | 0.177841  | 0.031628      | 0.925908             |
| 2010-05-01  | 0.182274  | 0.051326  | 0.130947  | 0.130947  | 0.017147      | 0.718410             |
| 2010-06-01  | 0.120232  | -0.029201 | 0.149432  | 0.149432  | 0.022330      | 1.242869             |
| ...         | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01  | -0.013514 | 0.010029  | -0.023544 | 0.023544  | 0.000554      | 1.742105             |
| 2024-01-01  | 0.061625  | -0.052334 | 0.113959  | 0.113959  | 0.012987      | 1.849246             |
| 2024-02-01  | -0.047850 | 0.028690  | -0.076540 | 0.076540  | 0.005858      | 1.599587             |
| 2024-03-01  | -0.022643 | 0.008871  | -0.031514 | 0.031514  | 0.000993      | 1.391758             |
| 2024-04-01  | NaN       | 0.006330  | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns



## Electrical\_Applications

```
In [57]: df = df_export_ANALYSIS.copy()
name = df.columns[21]
display(f"Component: {name}")

'Component: Electrical_Appliances'
```

```
In [58]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/{name}.pkl")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Electrical_Appliances |
|------------|-----------------------|
| 1995-02-01 | 0.126758              |
| 1995-03-01 | 0.267712              |
| 1995-04-01 | -0.279240             |
| 1995-05-01 | 0.357869              |
| 1995-06-01 | 0.039214              |
| ...        | ...                   |
| 2023-11-01 | 0.054412              |
| 2023-12-01 | -0.045130             |
| 2024-01-01 | 0.032271              |
| 2024-02-01 | 0.082944              |
| 2024-03-01 | 0.093778              |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -5.377911805986953
p-value: 3.7684632216805727e-06
Critical Value 1%: -3.4502011472639724
Critical Value 5%: -2.8702852297358983
Critical Value 10%: -2.5714292194077513
is_stationary: True
```

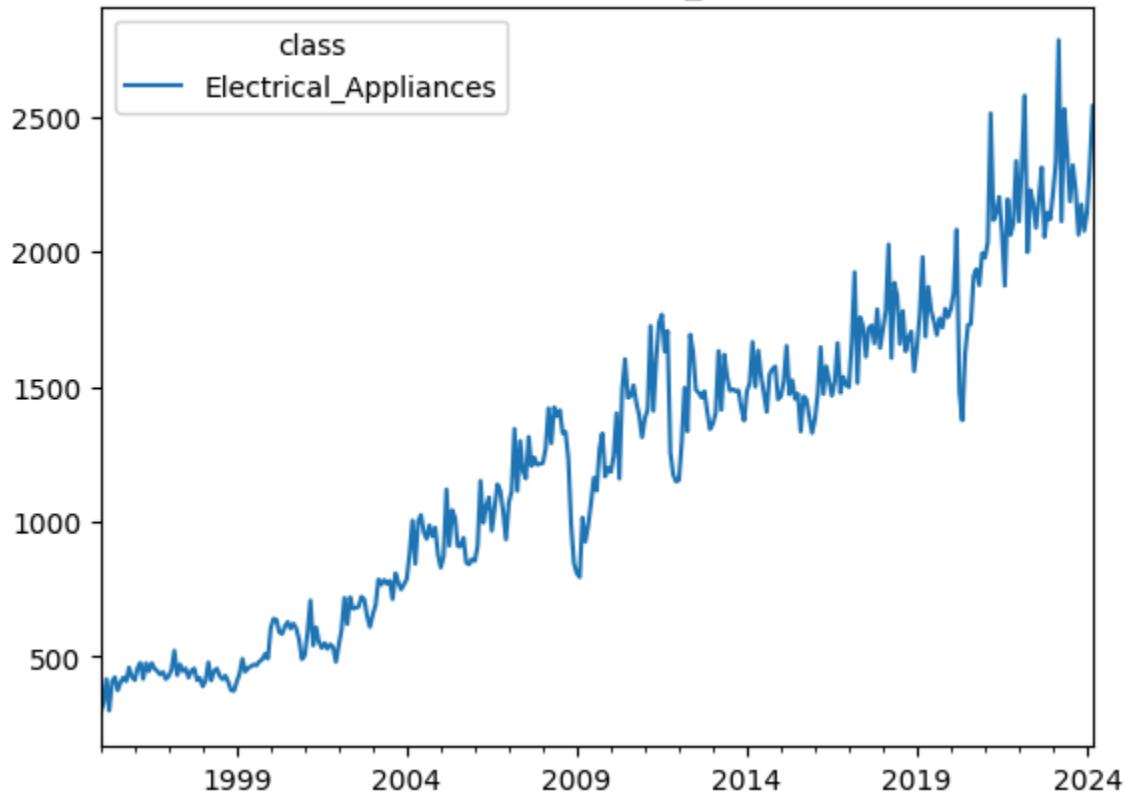
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.058172037861919296
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

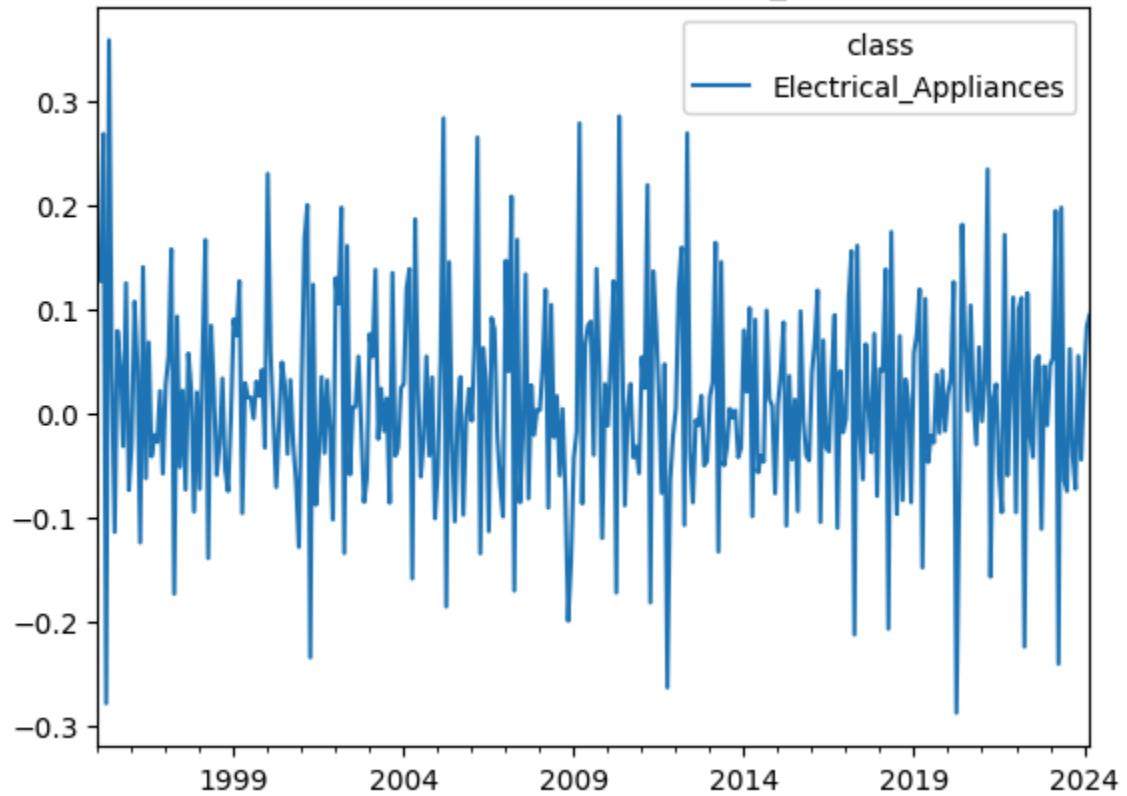
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

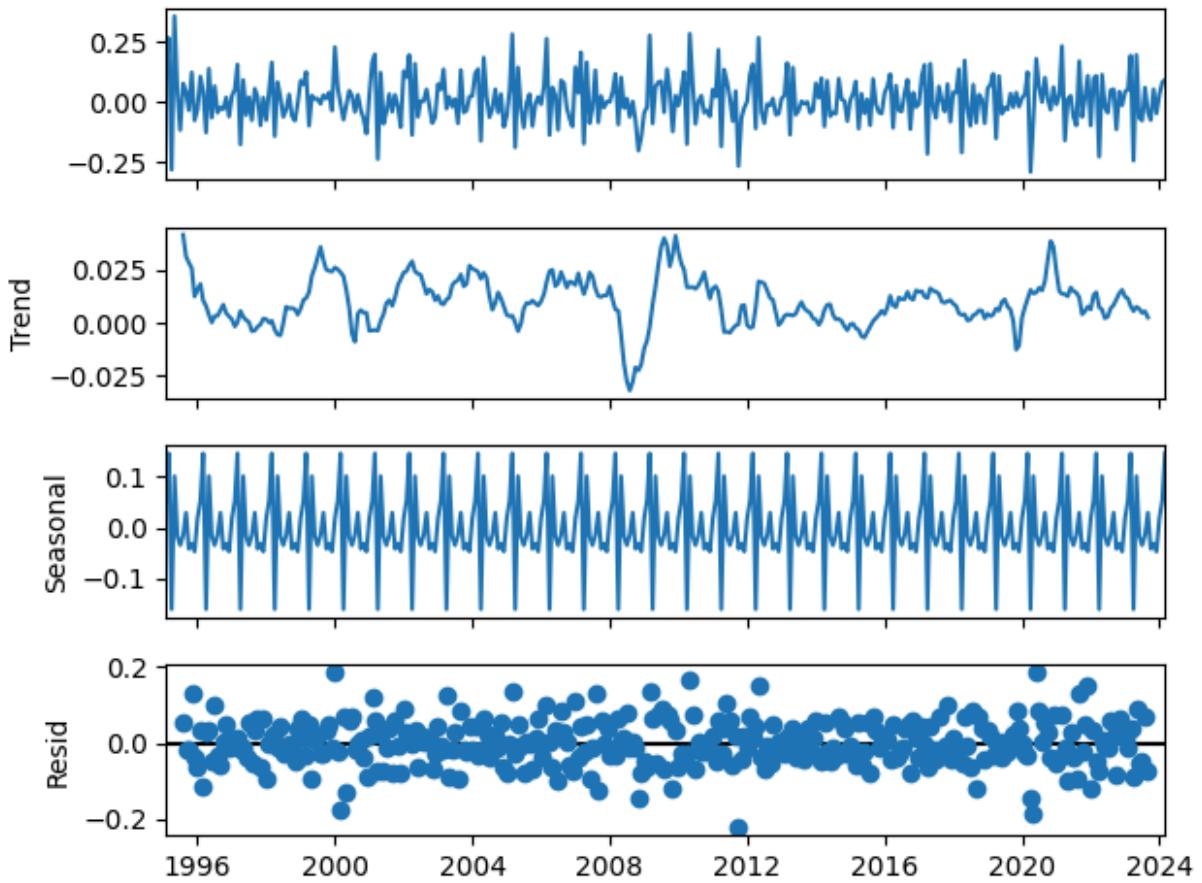
```
result = kpss(series, regression='c')
```

### Export Value (Electrical\_Appliances)

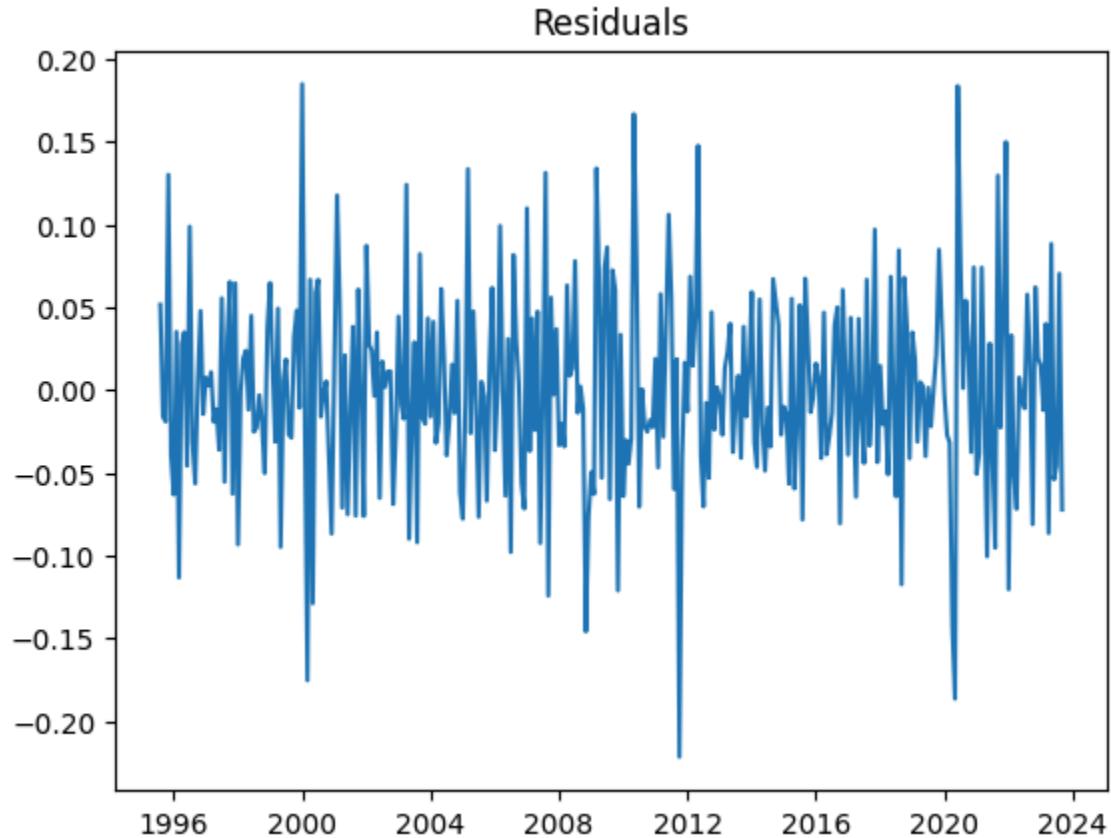


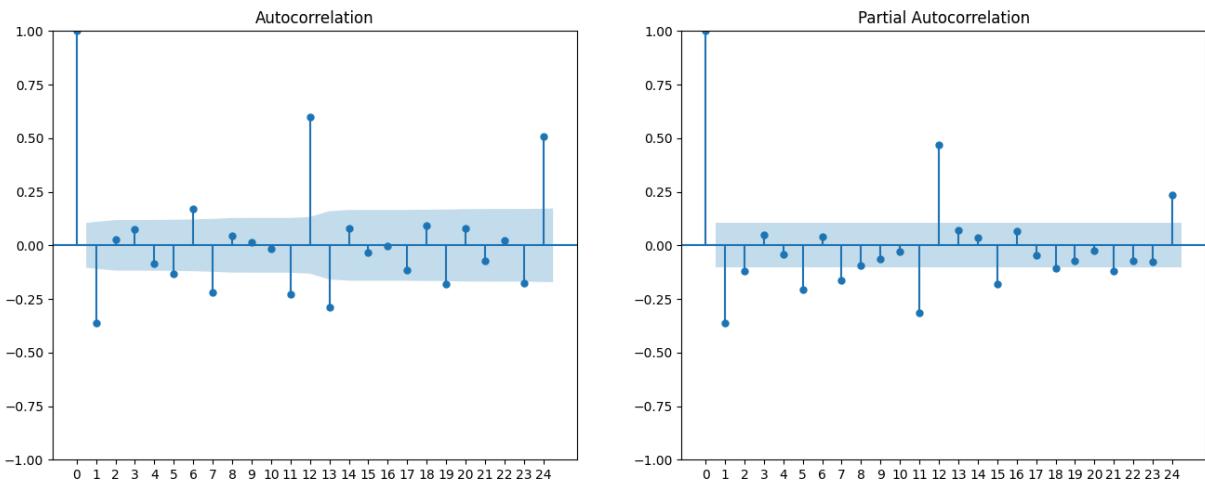
### Export Value % Change (Electrical\_Appliances)





Plotting Residuals for Homoscedasticity Check...





Time series assumptions are met.

True

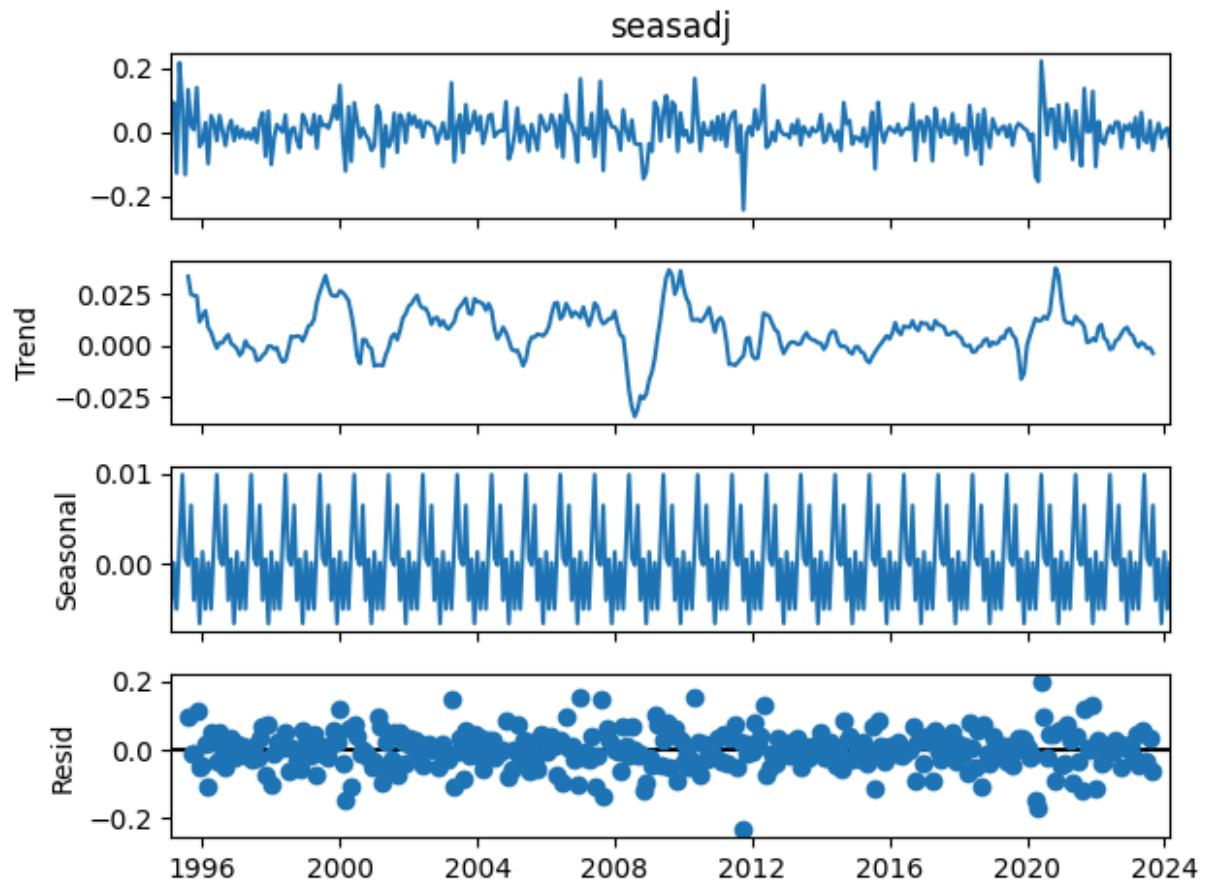
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted): '
Running ADF Test for Stationarity...
ADF Statistic: -10.612124449733983
p-value: 5.797742598104735e-19
Critical Value 1%: -3.4493918438232525
Critical Value 5%: -2.8699298018856574
Critical Value 10%: -2.5712397066390458
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.14532158329969574
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

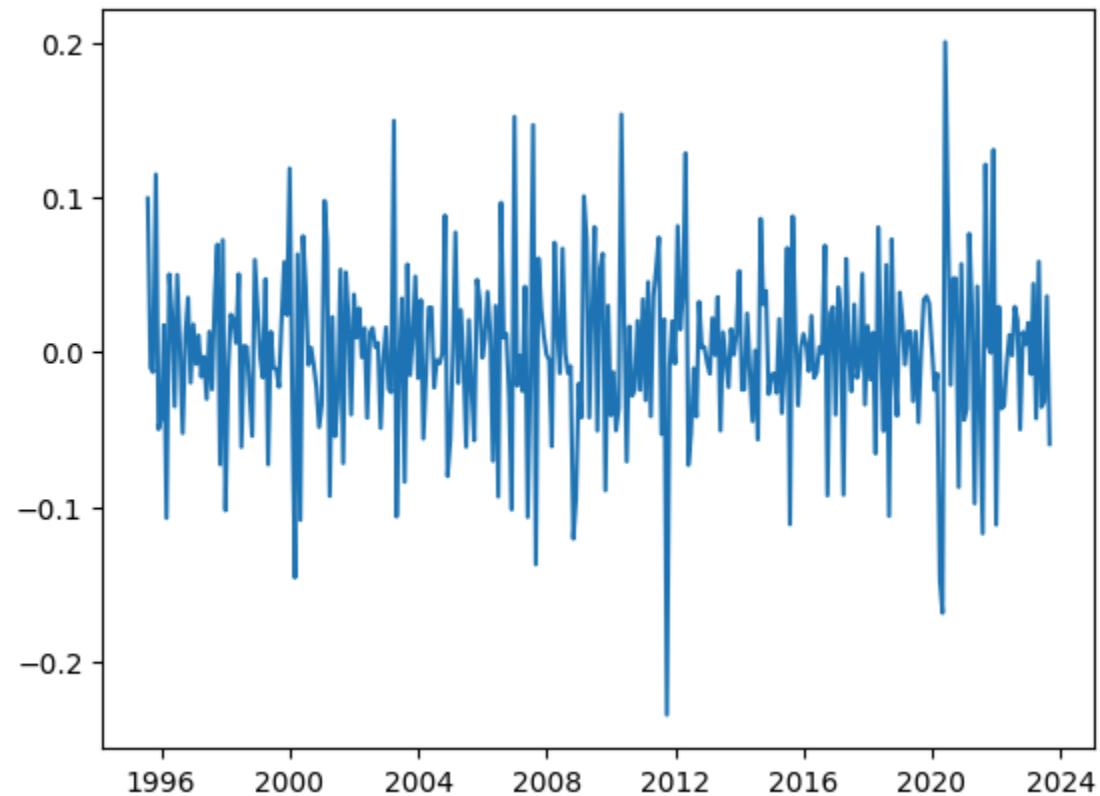
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

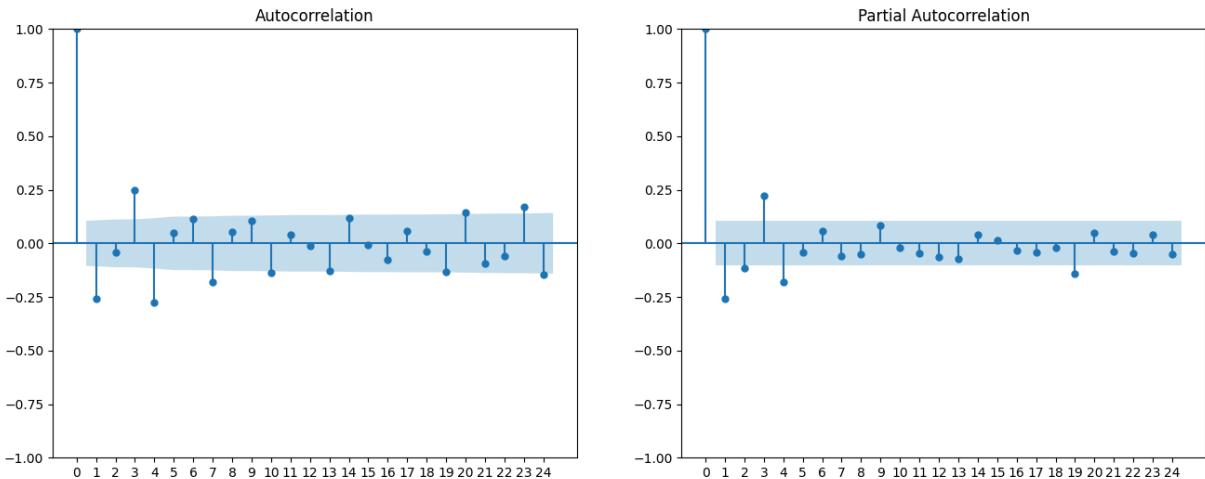
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals



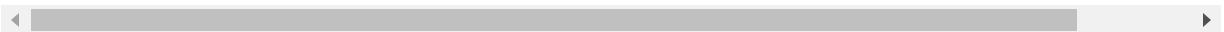


Time series assumptions are met.

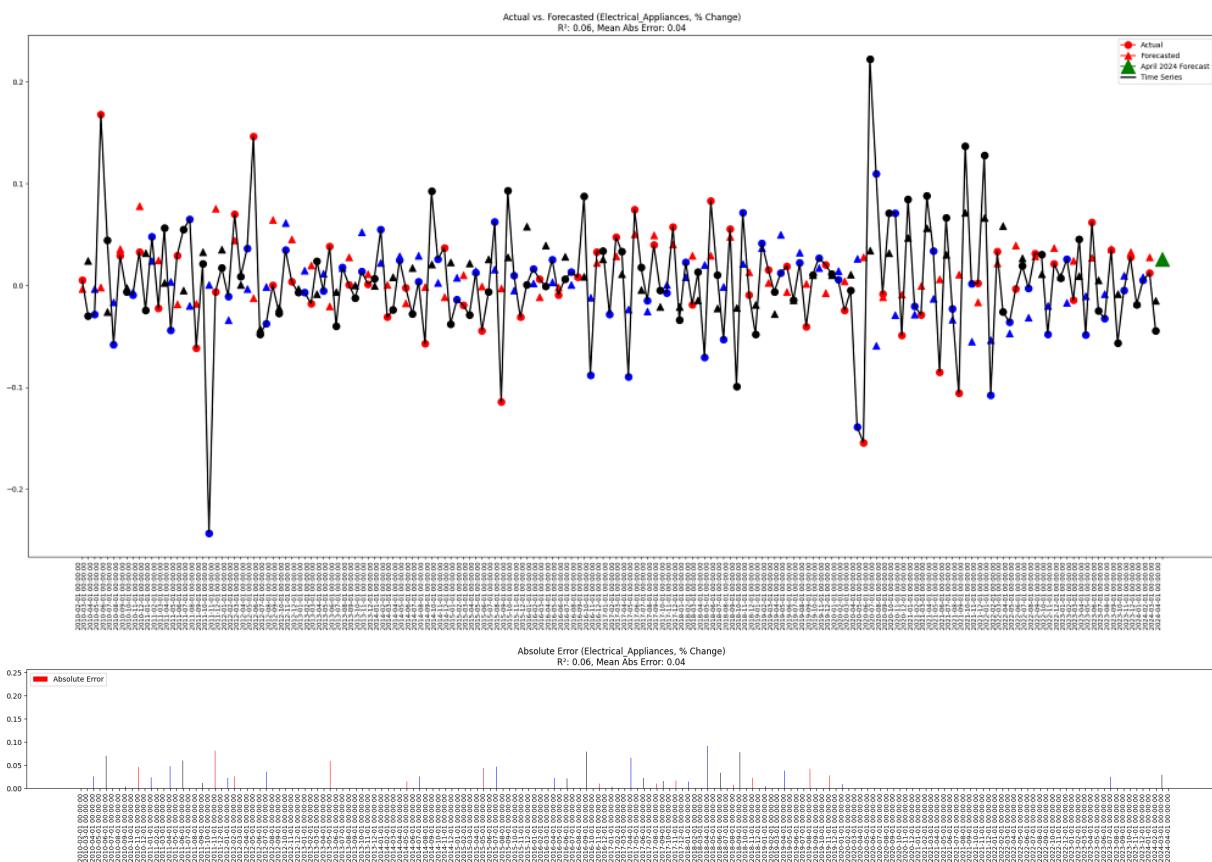
'Data of Actual vs Forecasted values with error metrics:'

| Electrical_Appliances | actual    | forecast  | error     | abs_error | squared_error | abs_percent |
|-----------------------|-----------|-----------|-----------|-----------|---------------|-------------|
| 2010-02-01            | 0.005079  | -0.003276 | 0.008356  | 0.008356  | 0.000070      |             |
| 2010-03-01            | -0.029884 | 0.024384  | -0.054268 | 0.054268  | 0.002945      |             |
| 2010-04-01            | -0.028542 | -0.003059 | -0.025483 | 0.025483  | 0.000649      |             |
| 2010-05-01            | 0.168139  | -0.001598 | 0.169737  | 0.169737  | 0.028811      |             |
| 2010-06-01            | 0.044586  | -0.025942 | 0.070527  | 0.070527  | 0.004974      |             |
| ...                   | ...       | ...       | ...       | ...       | ...           | ...         |
| 2023-12-01            | -0.018678 | -0.017168 | -0.001511 | 0.001511  | 0.000002      |             |
| 2024-01-01            | 0.005029  | 0.008329  | -0.003299 | 0.003299  | 0.000011      |             |
| 2024-02-01            | 0.012431  | 0.027645  | -0.015214 | 0.015214  | 0.000231      |             |
| 2024-03-01            | -0.044396 | -0.015029 | -0.029366 | 0.029366  | 0.000862      |             |
| 2024-04-01            | NaN       | 0.025907  | NaN       | NaN       | NaN           |             |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.0374392010650417'



## Metal\_&\_Steel

```
In [59]: df = df_export_ANALYSIS.copy()
name = df.columns[22]
display(f"Component: {name}")
```

'Component: Metal\_&\_Steel'

```
In [60]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

| class      | Metal_&_Steel |
|------------|---------------|
| 1995-02-01 | 0.064318      |
| 1995-03-01 | 0.459156      |
| 1995-04-01 | -0.281651     |
| 1995-05-01 | 0.100241      |
| 1995-06-01 | 0.026508      |
| ...        | ...           |
| 2023-11-01 | 0.226618      |
| 2023-12-01 | -0.070939     |
| 2024-01-01 | 0.328411      |
| 2024-02-01 | -0.263440     |
| 2024-03-01 | 0.022859      |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

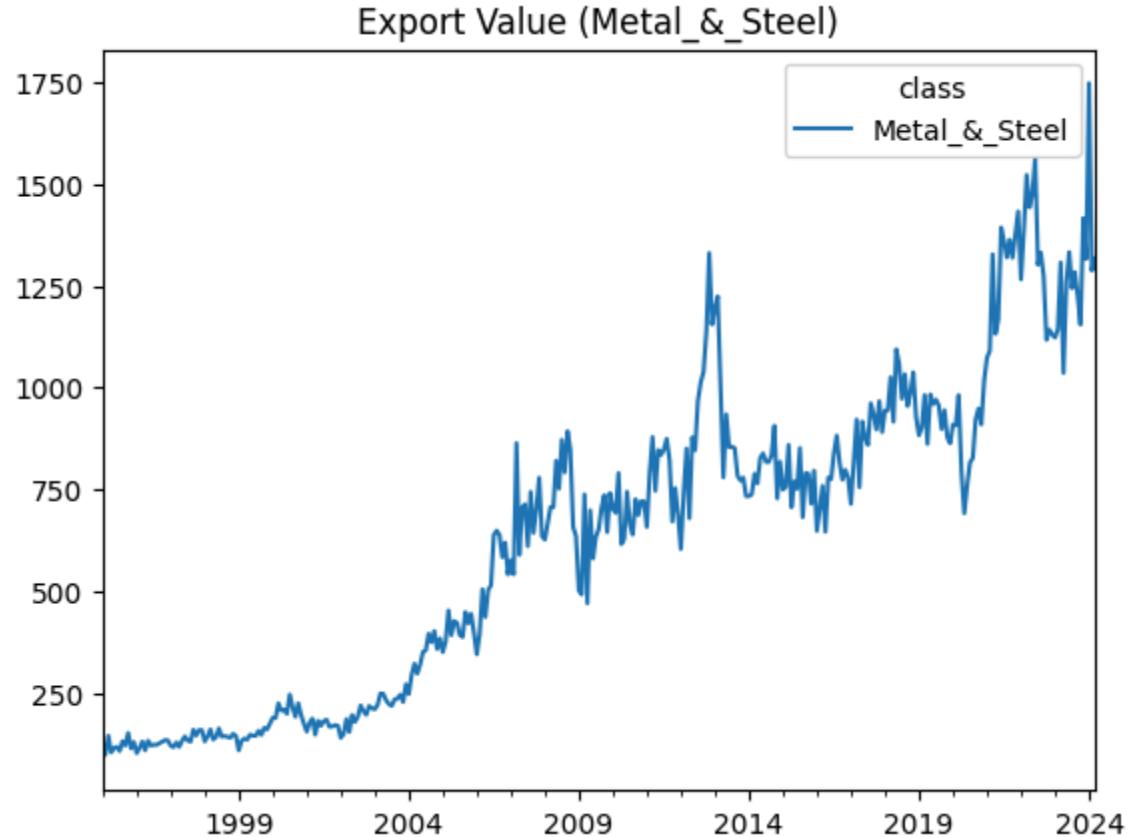
```
Running ADF Test for Stationarity...
ADF Statistic: -5.841636652333678
p-value: 3.7654382132623553e-07
Critical Value 1%: -3.4500219858626227
Critical Value 5%: -2.870206553997666
Critical Value 10%: -2.571387268879483
is_stationary: True
```

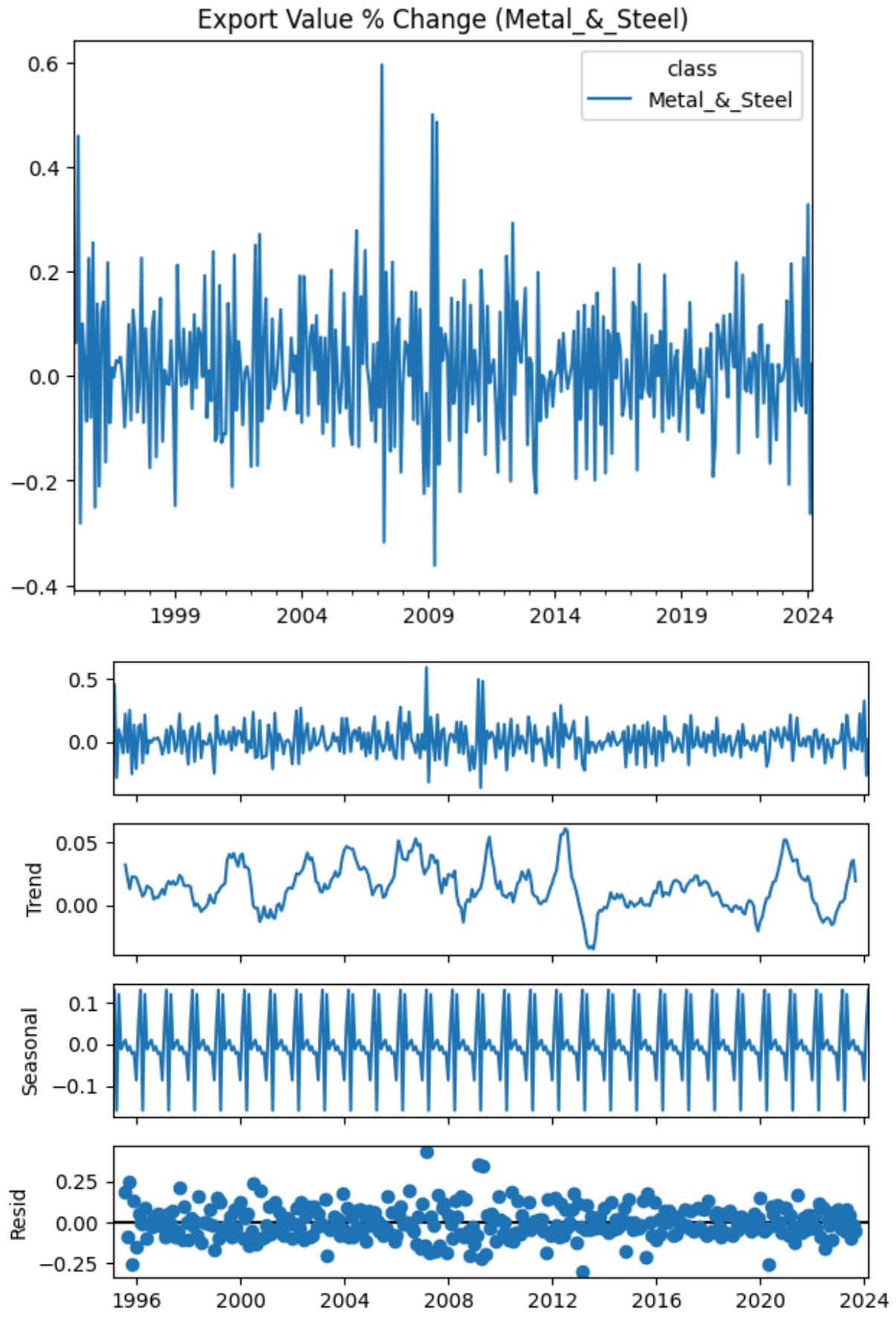
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.14452553890593434
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

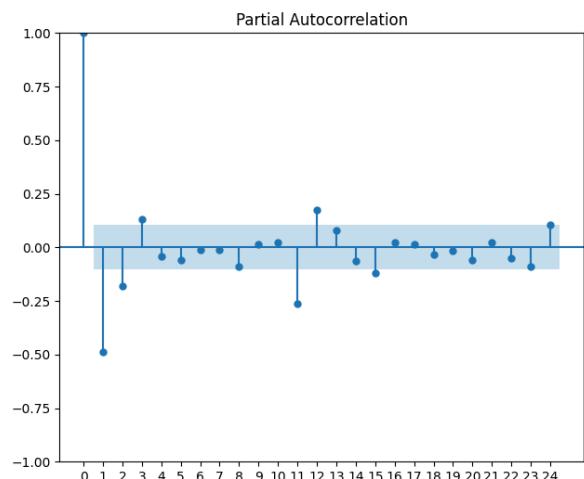
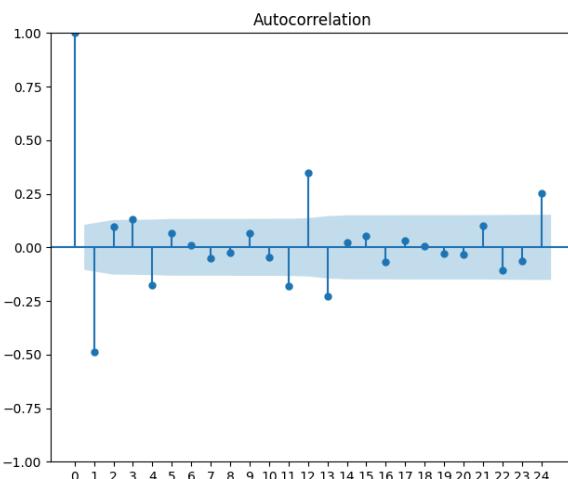
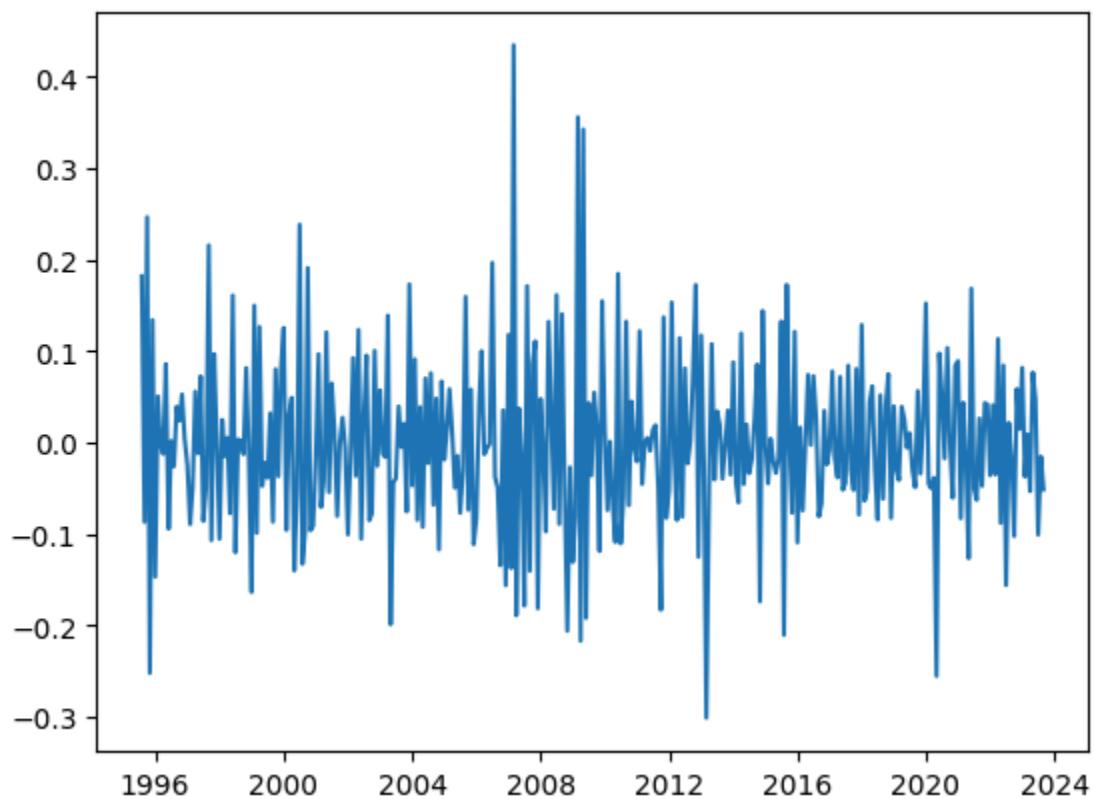
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```





### Residuals



Time series assumptions are met.

True

'-----'  
-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

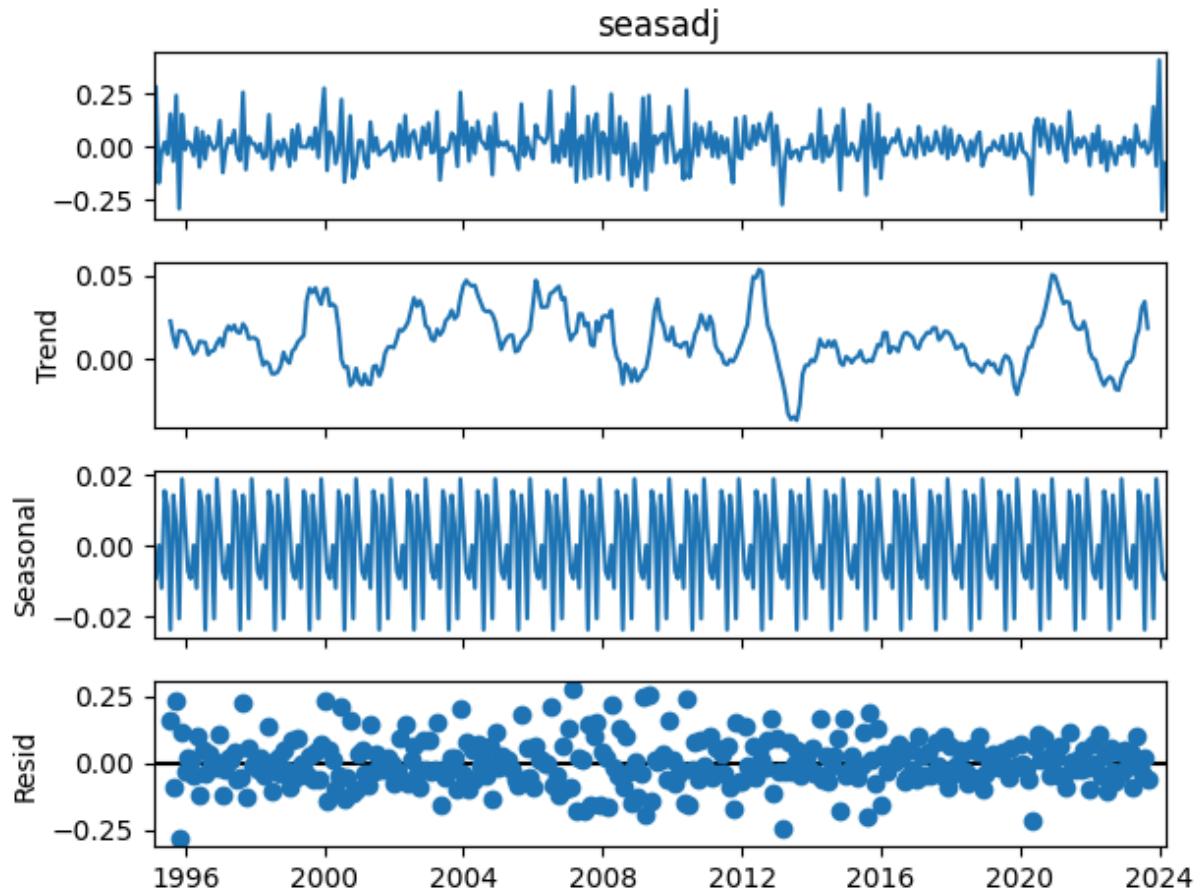
```
Running ADF Test for Stationarity...
ADF Statistic: -10.56675547842935
p-value: 7.482510308618111e-19
Critical Value 1%: -3.449336554273722
Critical Value 5%: -2.8699055166063085
Critical Value 10%: -2.571226758215748
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.1350165248570389
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

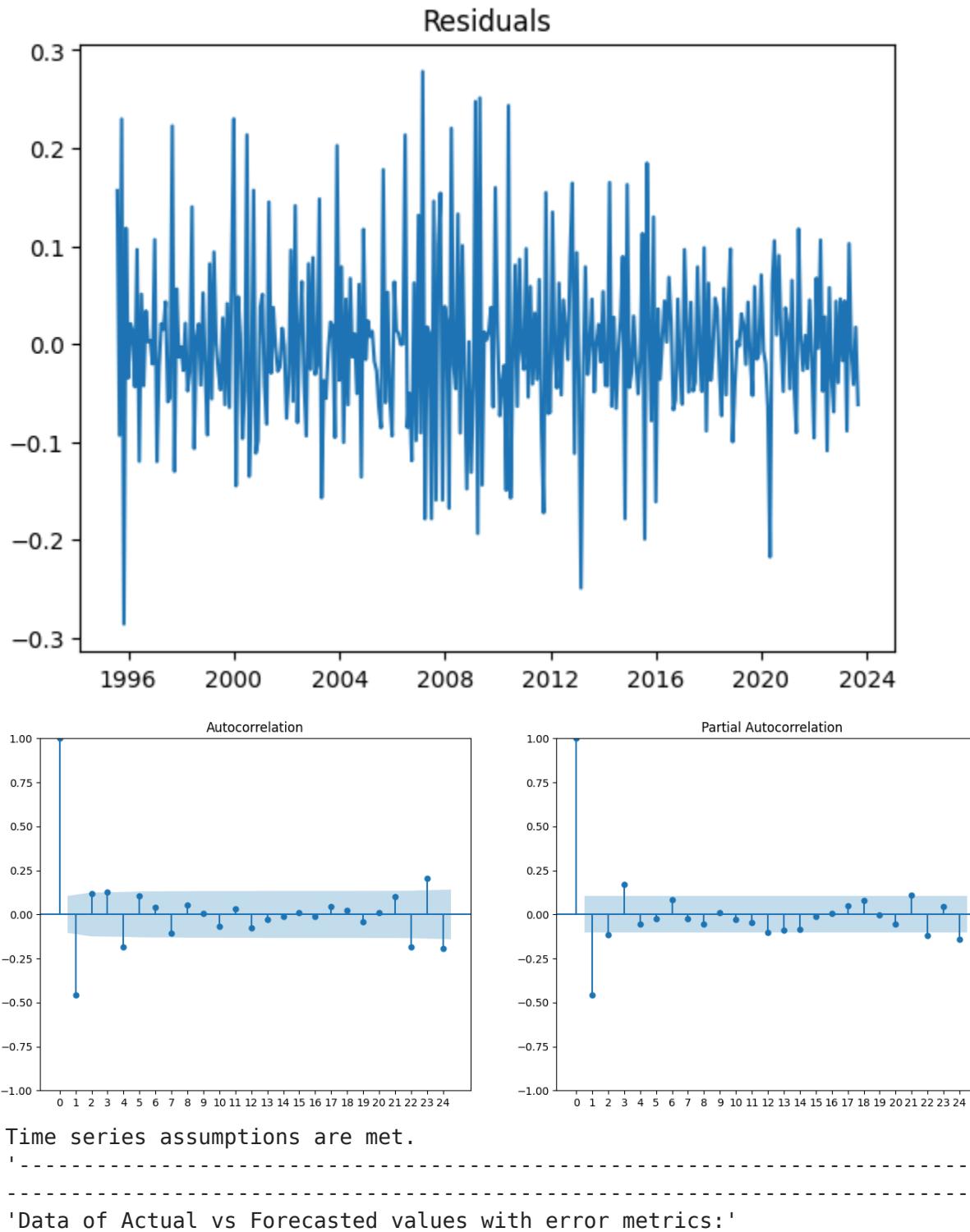
Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

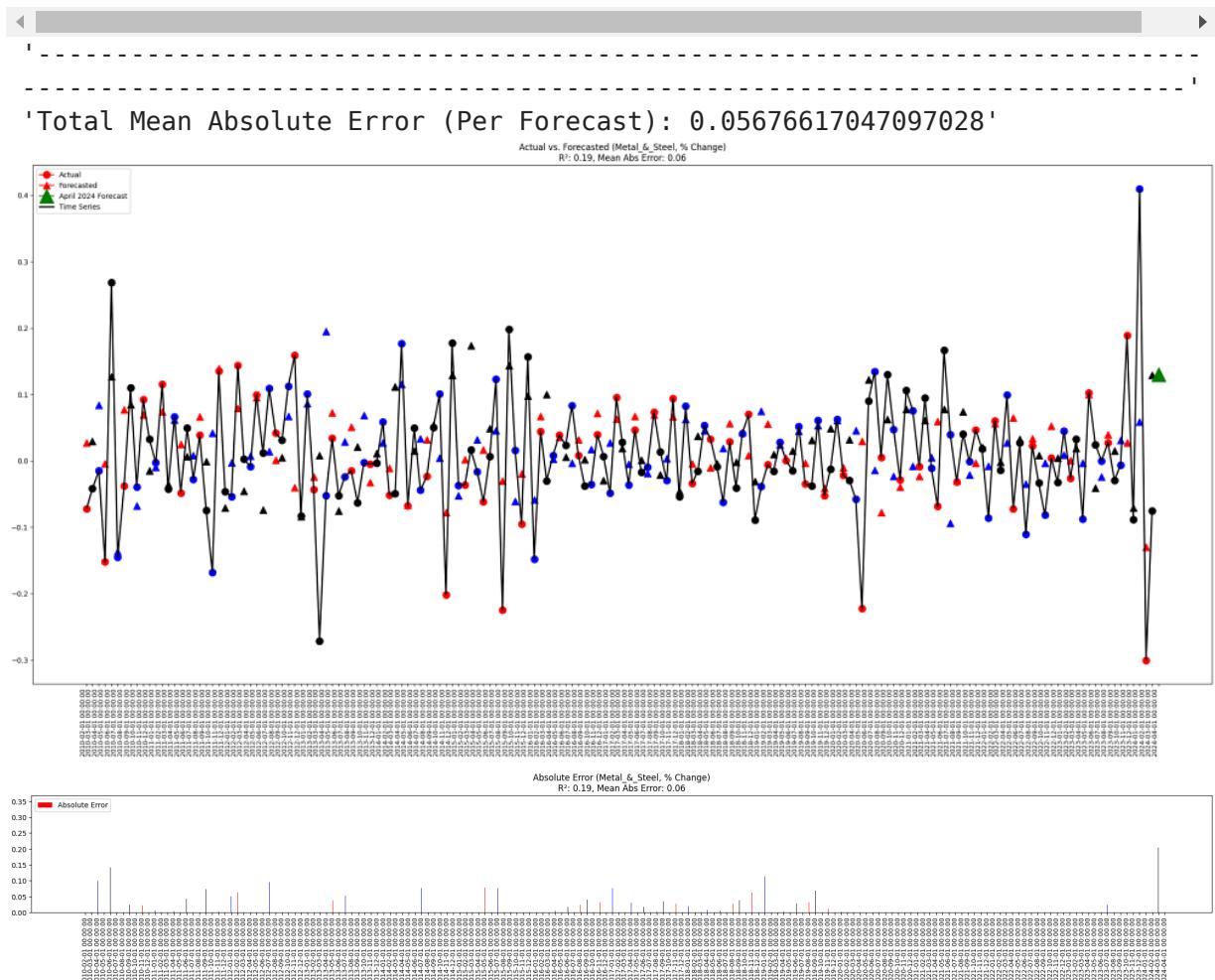


Plotting Residuals for Homoscedasticity Check...



| Metal_&_Steel | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_er |
|---------------|-----------|-----------|-----------|-----------|---------------|-------------------|
| 2010-02-01    | -0.072190 | 0.027076  | -0.099266 | 0.099266  | 0.009854      | 1.3750            |
| 2010-03-01    | -0.042031 | 0.029407  | -0.071438 | 0.071438  | 0.005103      | 1.6996            |
| 2010-04-01    | -0.015213 | 0.083807  | -0.099020 | 0.099020  | 0.009805      | 6.5089            |
| 2010-05-01    | -0.152263 | -0.004068 | -0.148195 | 0.148195  | 0.021962      | 0.9732            |
| 2010-06-01    | 0.268607  | 0.126936  | 0.141671  | 0.141671  | 0.020071      | 0.5274            |
| ...           | ...       | ...       | ...       | ...       | ...           | ...               |
| 2023-12-01    | -0.088298 | -0.070426 | -0.017871 | 0.017871  | 0.000319      | 0.2024            |
| 2024-01-01    | 0.409939  | 0.058856  | 0.351083  | 0.351083  | 0.123259      | 0.8564            |
| 2024-02-01    | -0.300484 | -0.130008 | -0.170476 | 0.170476  | 0.029062      | 0.5673            |
| 2024-03-01    | -0.075214 | 0.129002  | -0.204215 | 0.204215  | 0.041704      | 2.7151            |
| 2024-04-01    | NaN       | 0.130308  | NaN       | NaN       | NaN           | N                 |

171 rows × 6 columns



## Automotive

```
In [61]: df = df_export_ANALYSIS.copy()
name = df.columns[23]
display(f"Component: {name}")

'Component: Automotive'
```

```
In [62]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Automotive |
|------------|------------|
| 1995-02-01 | 0.256000   |
| 1995-03-01 | 0.358086   |
| 1995-04-01 | -0.165180  |
| 1995-05-01 | 0.248835   |
| 1995-06-01 | 0.062322   |
| ...        | ...        |
| 2023-11-01 | -0.032219  |
| 2023-12-01 | 0.017765   |
| 2024-01-01 | -0.046556  |
| 2024-02-01 | 0.008811   |
| 2024-03-01 | 0.128564   |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -6.059763471691257
p-value: 1.220547439364615e-07
Critical Value 1%: -3.449962981927952
Critical Value 5%: -2.870180642420163
Critical Value 10%: -2.5713734527352607
is_stationary: True
```

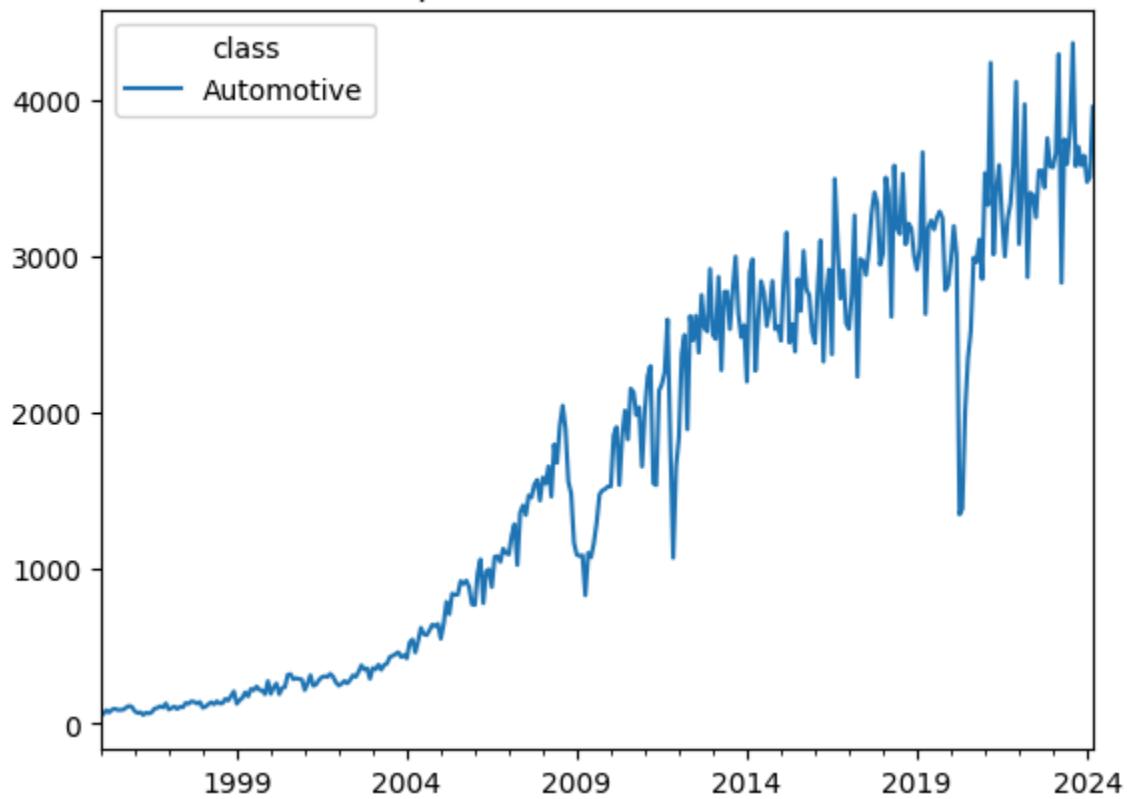
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.2146181502400707
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

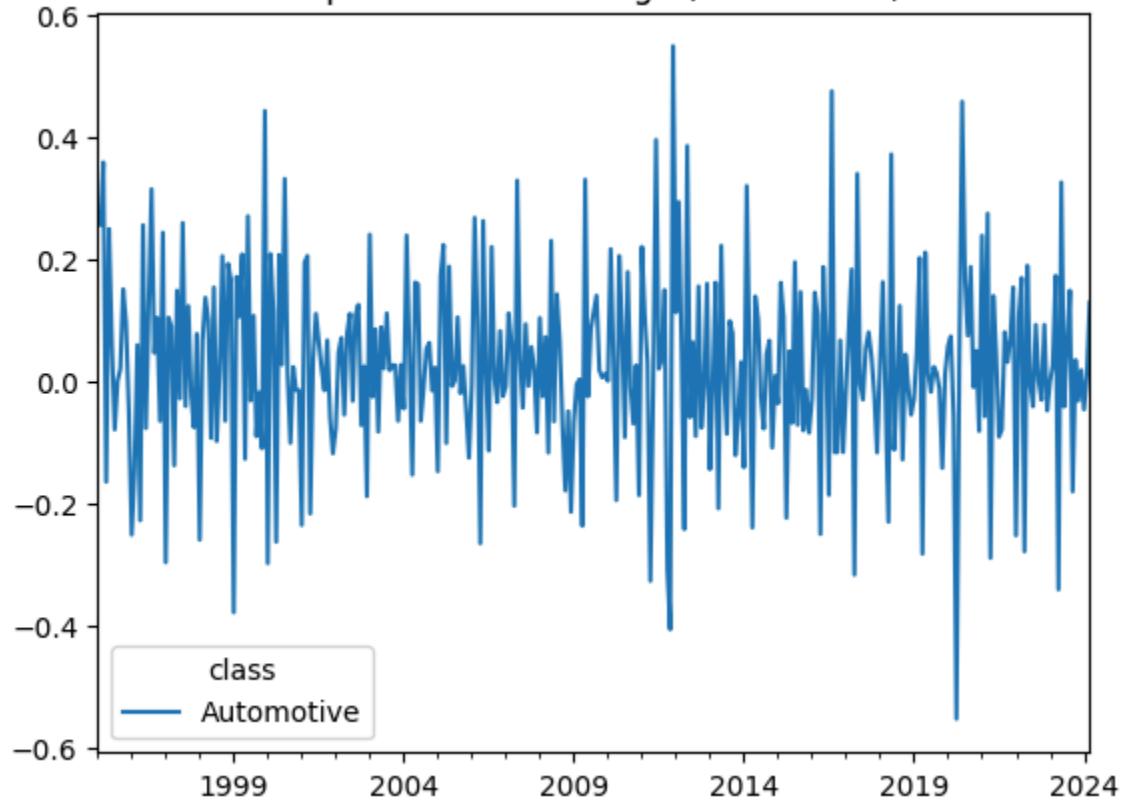
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

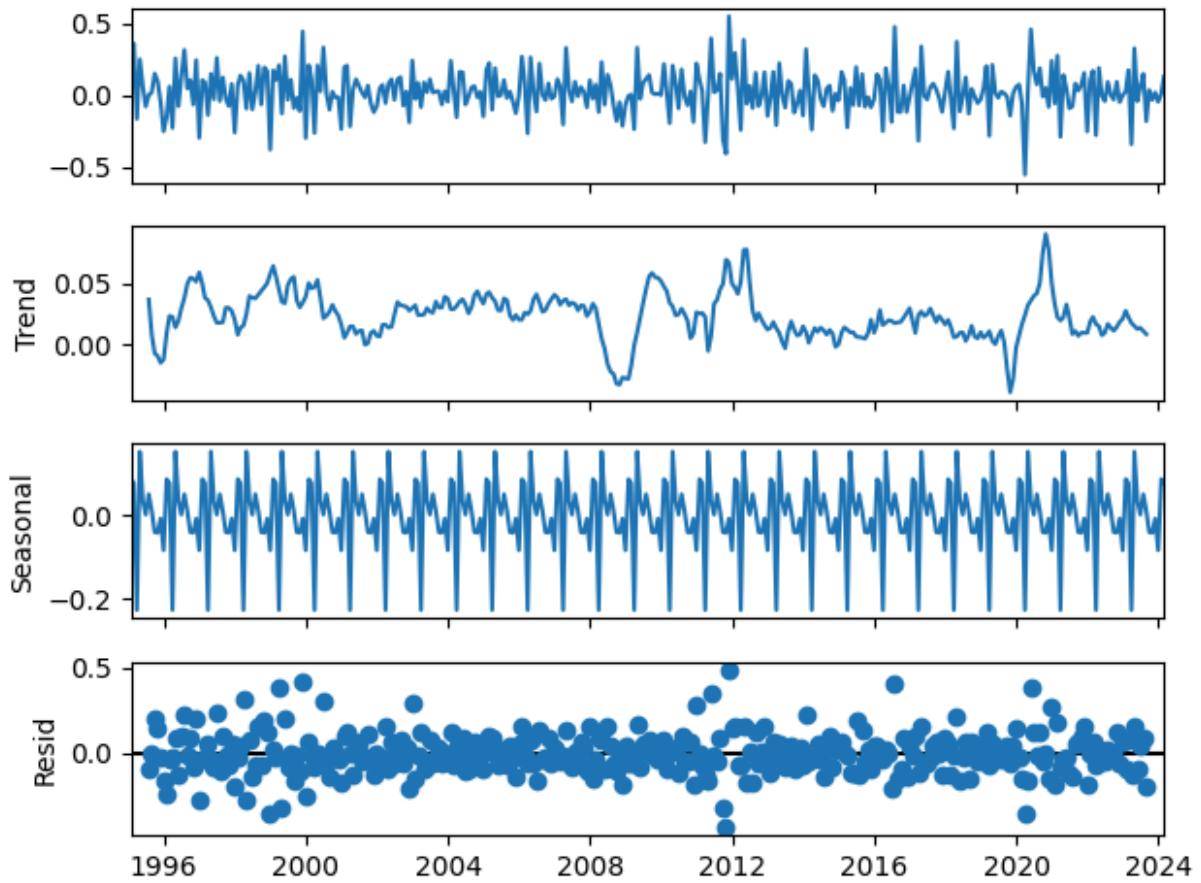
```
result = kpss(series, regression='c')
```

## Export Value (Automotive)



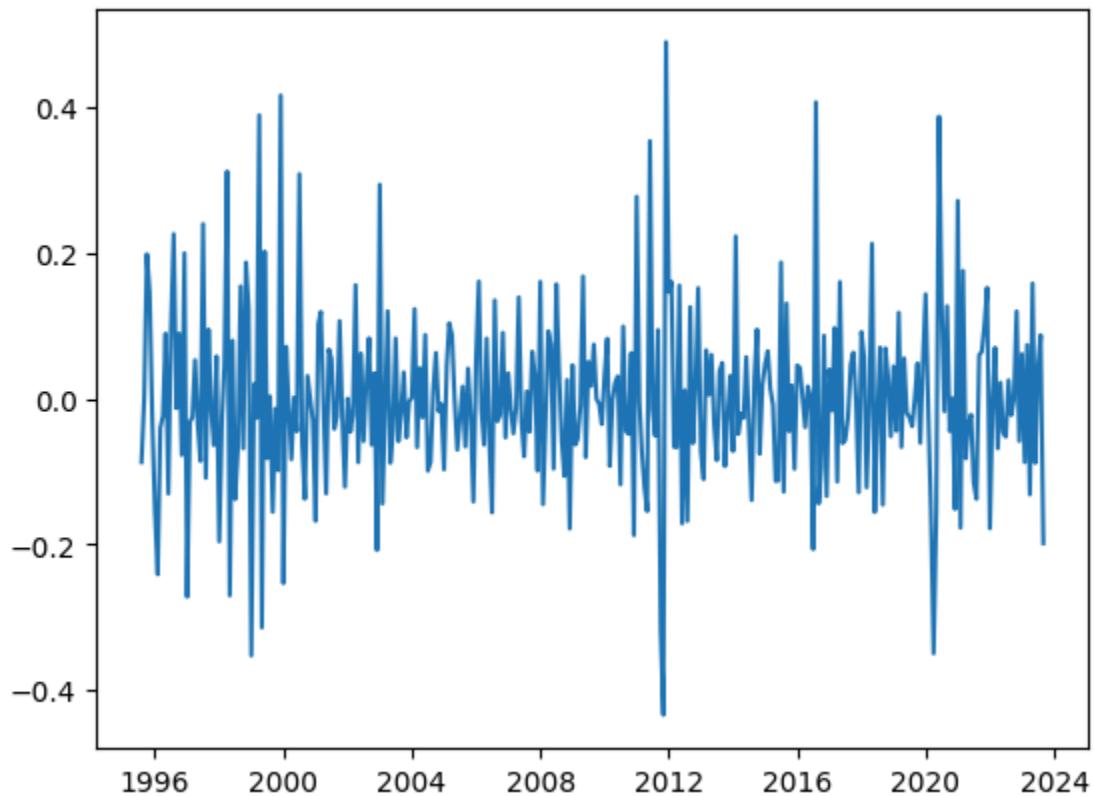
## Export Value % Change (Automotive)

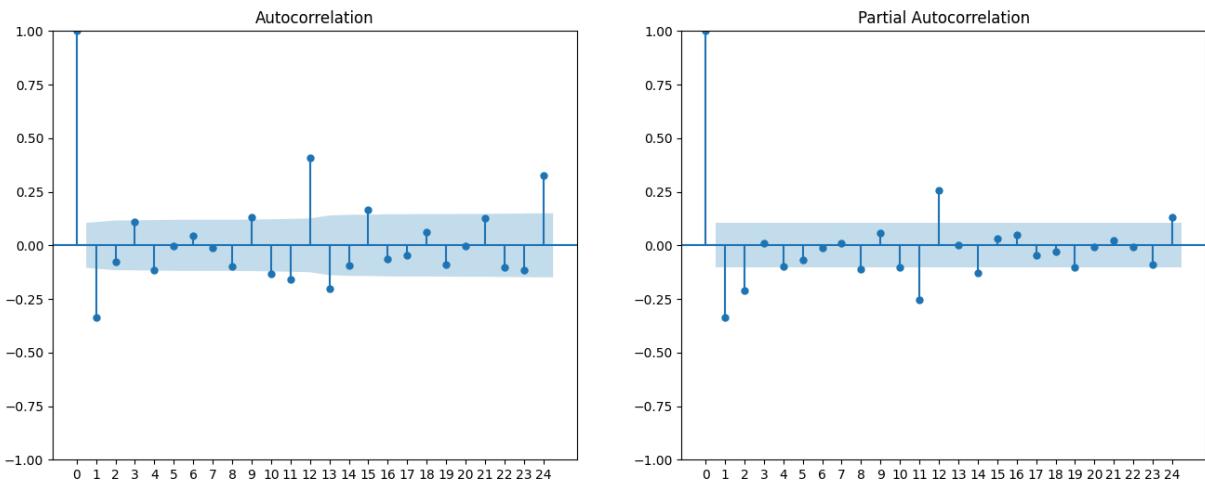




Plotting Residuals for Homoscedasticity Check...

Residuals





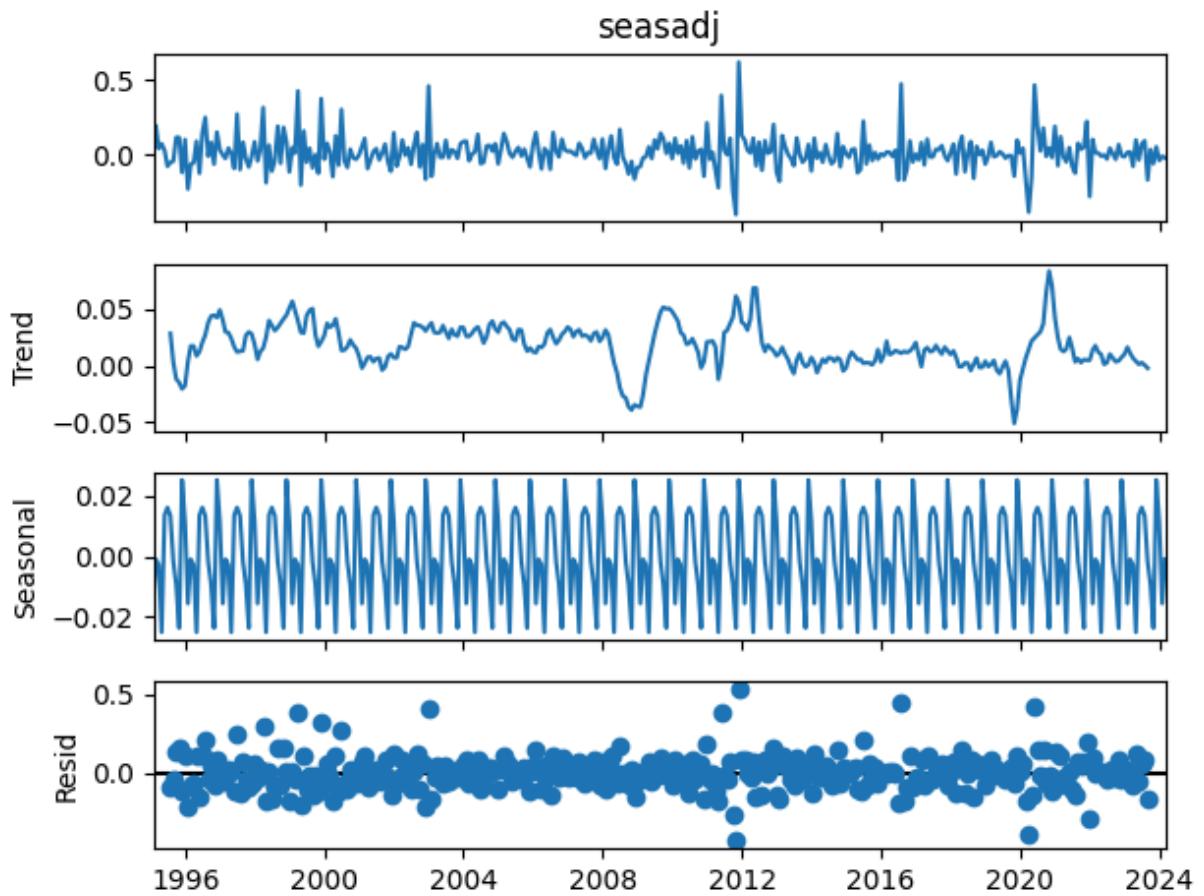
Time series assumptions are met.

True

```
'-----
-----'
'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -12.180013160350473
p-value: 1.358334652172505e-22
Critical Value 1%: -3.4493918438232525
Critical Value 5%: -2.8699298018856574
Critical Value 10%: -2.5712397066390458
is_stationary: True
```

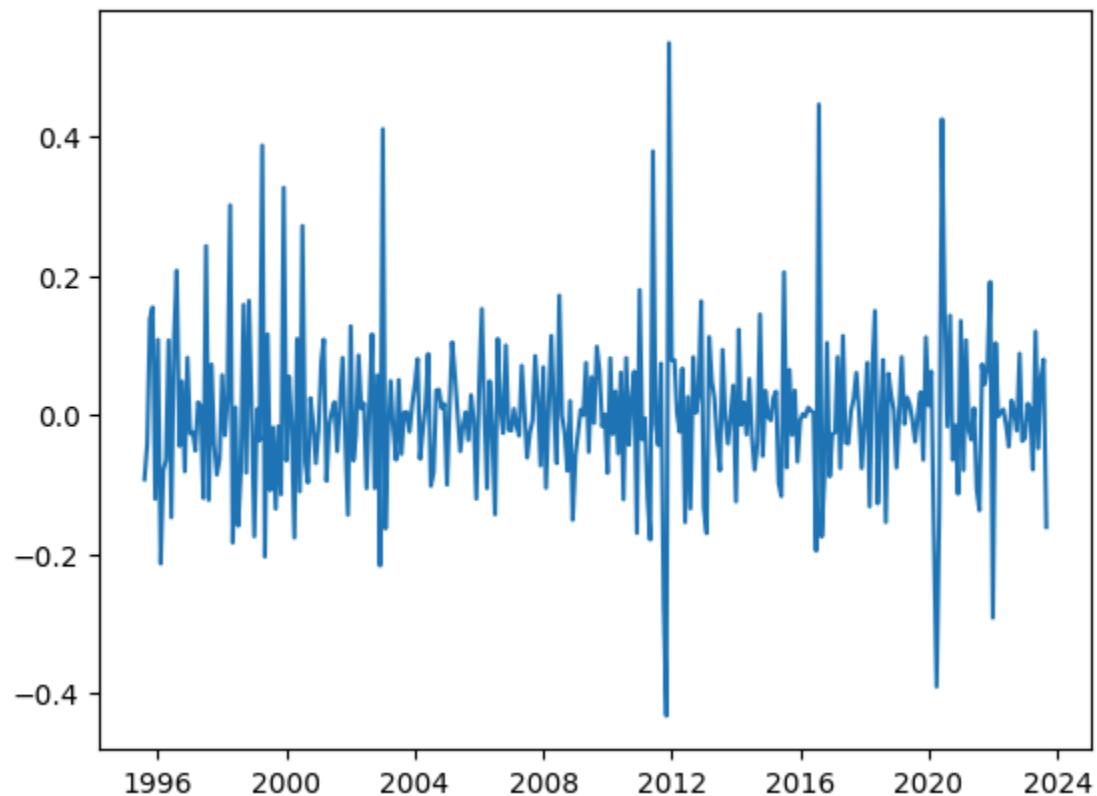
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.42409283793594005
p-value: 0.06677032847588792
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

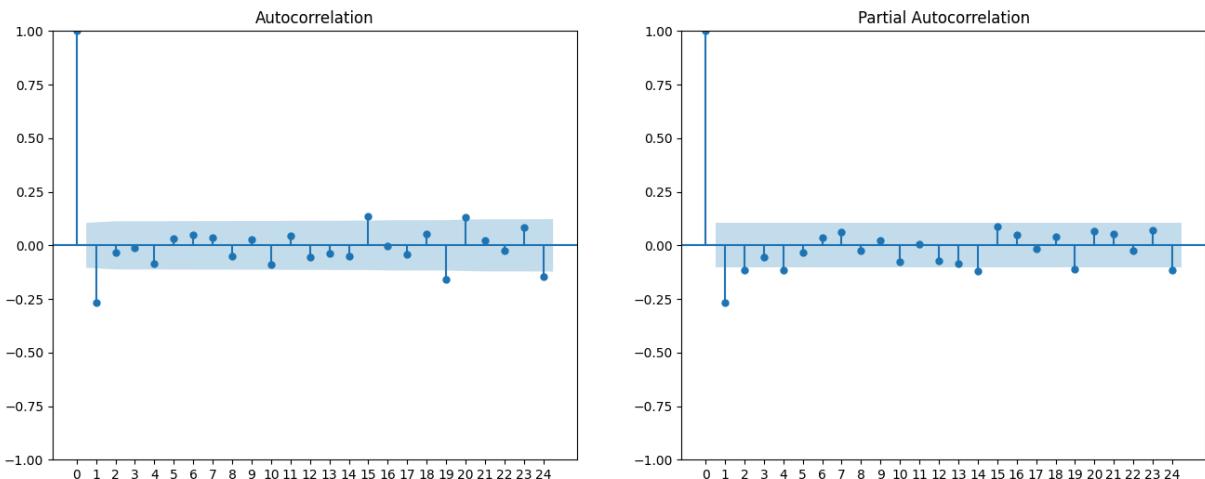
Decomposing the Series...



Plotting Residuals for Homoscedasticity Check...

Residuals



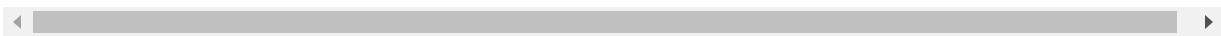


Time series assumptions are met.

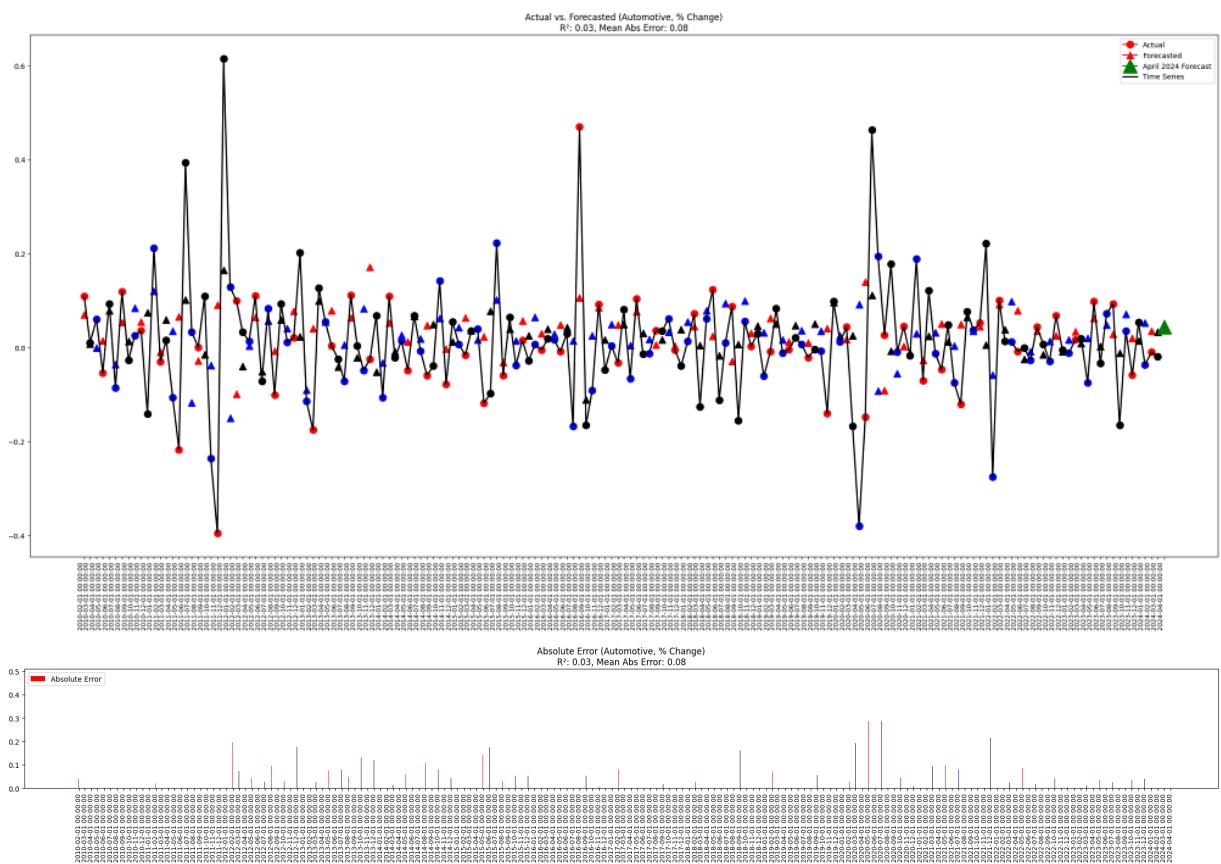
'Data of Actual vs Forecasted values with error metrics:'

| Automotive | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01 | 0.108920  | 0.070301  | 0.038618  | 0.038618  | 0.001491      | 0.354557             |
| 2010-03-01 | 0.010584  | 0.007928  | 0.002656  | 0.002656  | 0.000007      | 0.250954             |
| 2010-04-01 | 0.059690  | -0.000774 | 0.060464  | 0.060464  | 0.003656      | 1.012967             |
| 2010-05-01 | -0.054204 | 0.014101  | -0.068304 | 0.068304  | 0.004666      | 1.260144             |
| 2010-06-01 | 0.093313  | 0.079012  | 0.014301  | 0.014301  | 0.000205      | 0.153262             |
| ...        | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01 | 0.053319  | 0.013982  | 0.039337  | 0.039337  | 0.001547      | 0.737772             |
| 2024-01-01 | -0.036869 | 0.053052  | -0.089920 | 0.089920  | 0.008086      | 2.438946             |
| 2024-02-01 | -0.009765 | 0.035270  | -0.045035 | 0.045035  | 0.002028      | 4.611888             |
| 2024-03-01 | -0.019514 | 0.033162  | -0.052675 | 0.052675  | 0.002775      | 2.699384             |
| 2024-04-01 | NaN       | 0.043924  | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.08048349149774105'



## Aircrafts,\_ships,\_floating\_structures,\_and\_locomotive

```
In [63]: df = df_export_ANALYSIS.copy()
name = df.columns[24]
display(f"Component: {name}")

'Component: Aircrafts,_ships,_floating_structures,_and_locomotive'

In [64]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
```

```

check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

class Aircrafts,\_ships,\_floating\_structures,\_and\_locomotive

|            |           |
|------------|-----------|
| 1995-02-01 | 0.164388  |
| 1995-03-01 | 1.450172  |
| 1995-04-01 | -0.561828 |
| 1995-05-01 | 0.367831  |
| 1995-06-01 | 0.643136  |
| ...        | ...       |
| 2023-11-01 | 0.388672  |
| 2023-12-01 | 0.010779  |
| 2024-01-01 | -0.467876 |
| 2024-02-01 | 0.518252  |
| 2024-03-01 | 0.329107  |

350 rows × 1 columns

'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'  
'-----'

'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'

```
Running ADF Test for Stationarity...
ADF Statistic: -12.332296010858279
p-value: 6.386494076890815e-23
Critical Value 1%: -3.4492815848836296
Critical Value 5%: -2.8698813715275406
Critical Value 10%: -2.5712138845950587
is_stationary: True
```

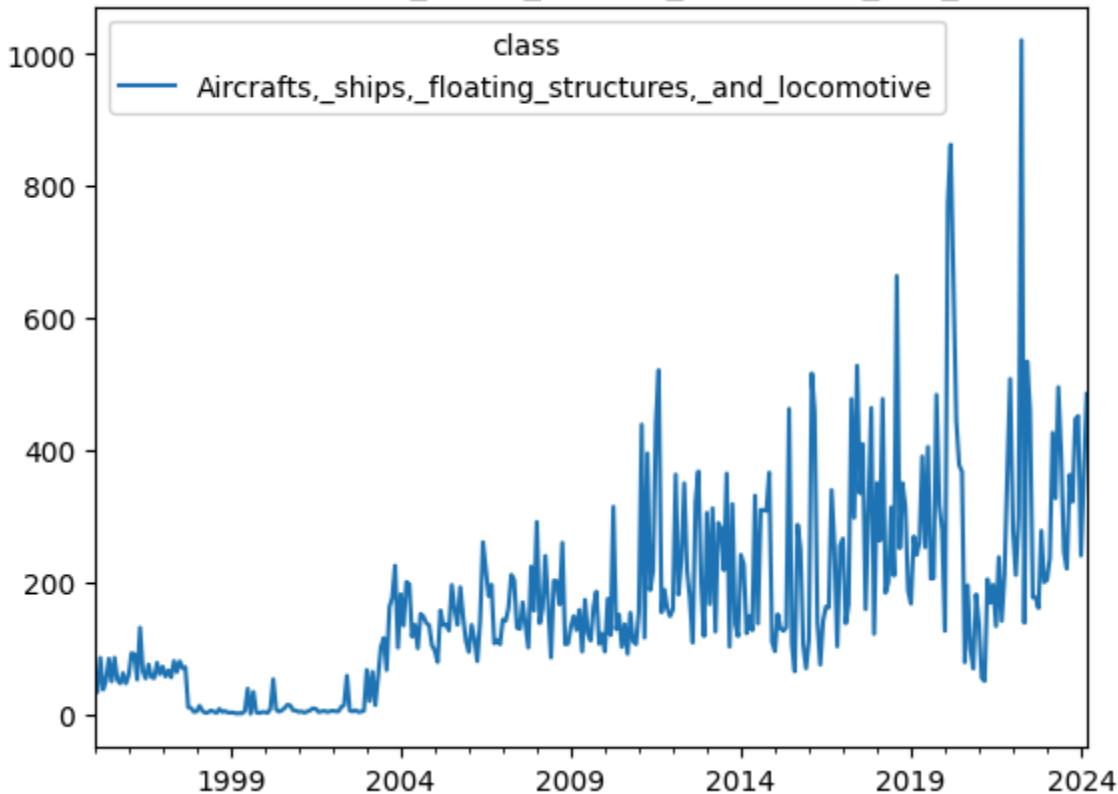
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.20582528392937968
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

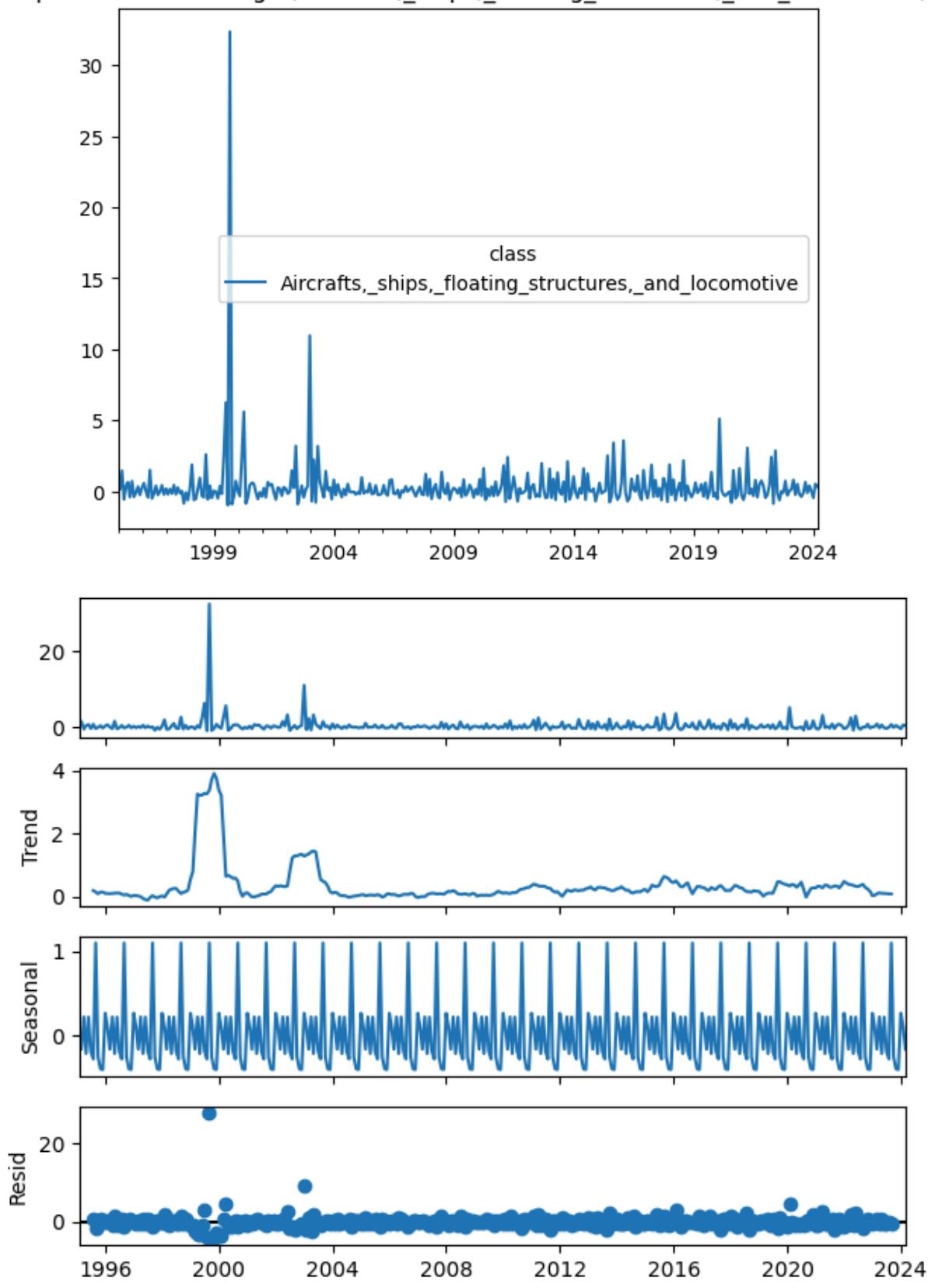
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

Export Value (Aircrafts,\_ships,\_floating\_structures,\_and\_locomotive)

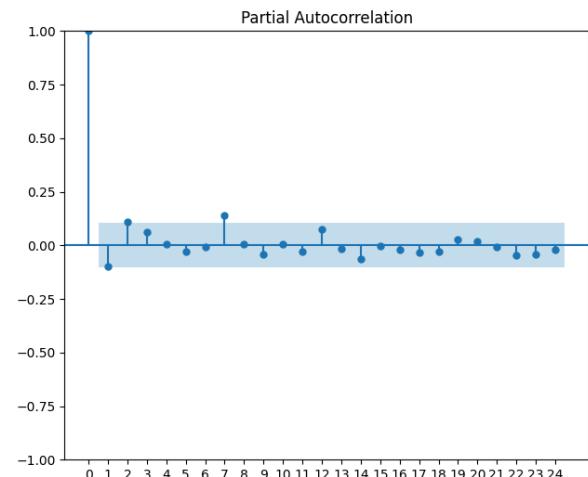
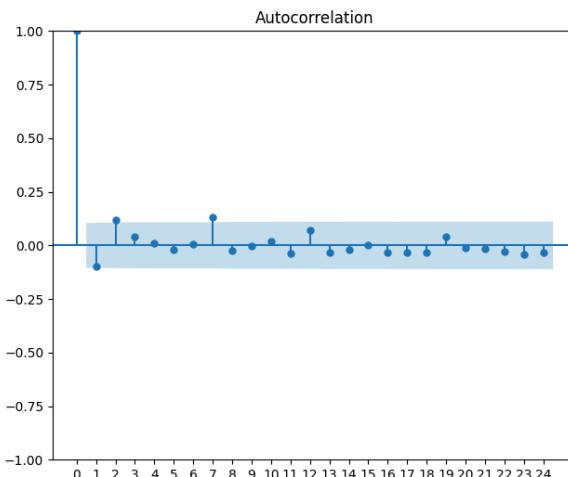
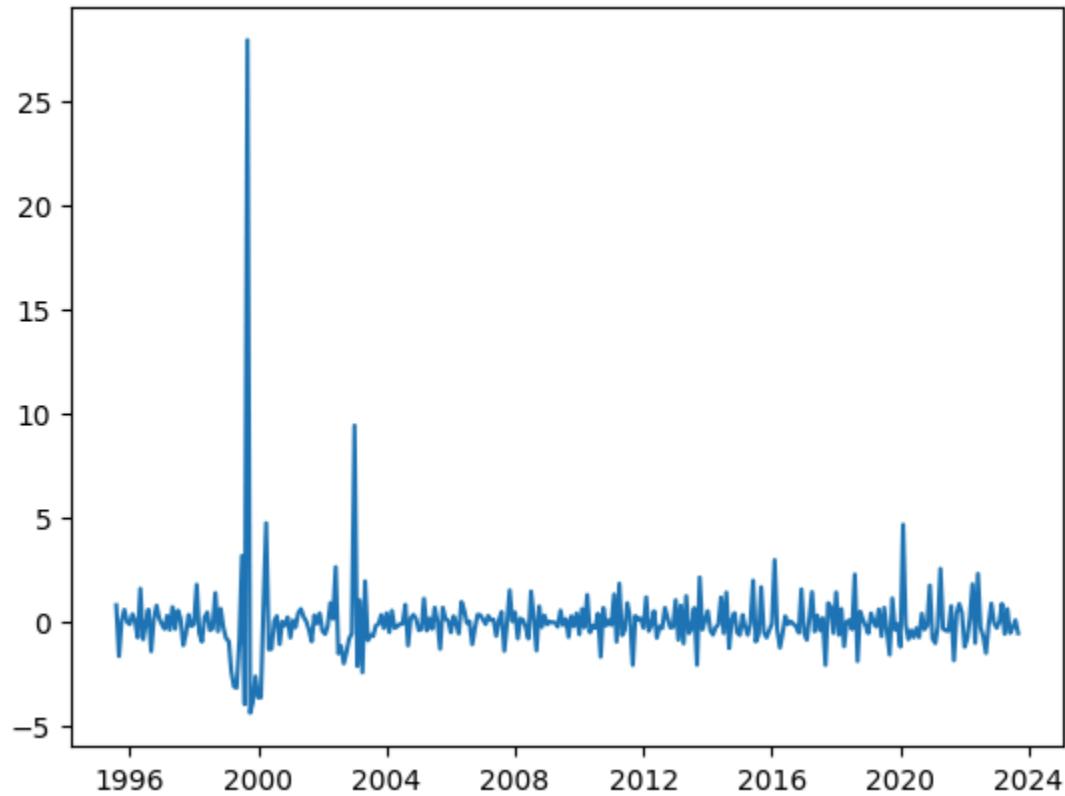


## Export Value % Change (Aircrafts,\_ships,\_floating\_structures,\_and\_locomotive)



Plotting Residuals for Homoscedasticity Check...

### Residuals



Time series assumptions are met.

True

'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

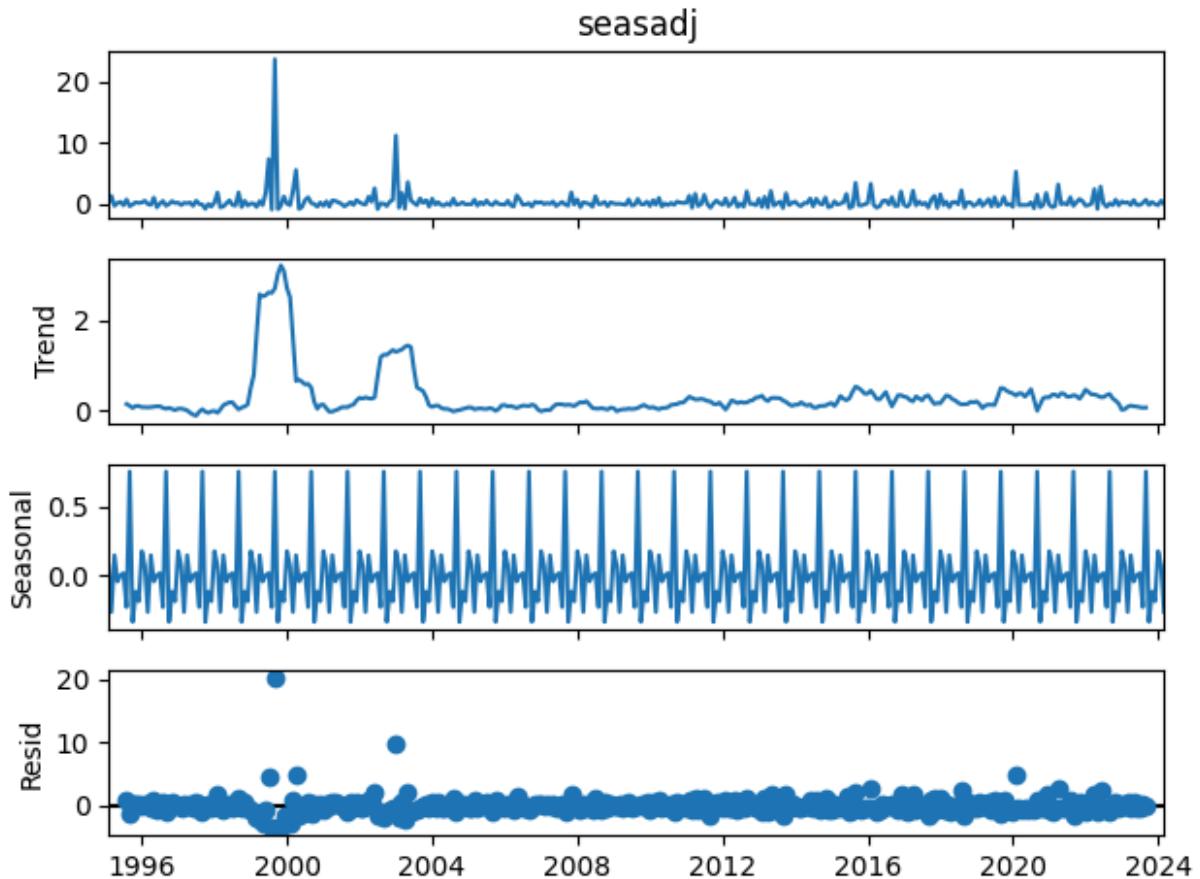
Running ADF Test for Stationarity...  
 ADF Statistic: -5.692223490536685  
 p-value: 8.022373100624624e-07  
 Critical Value 1%: -3.449559661646851  
 Critical Value 5%: -2.8700035112469626  
 Critical Value 10%: -2.5712790073013796  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.19483784078215474  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

Decomposing the Series...

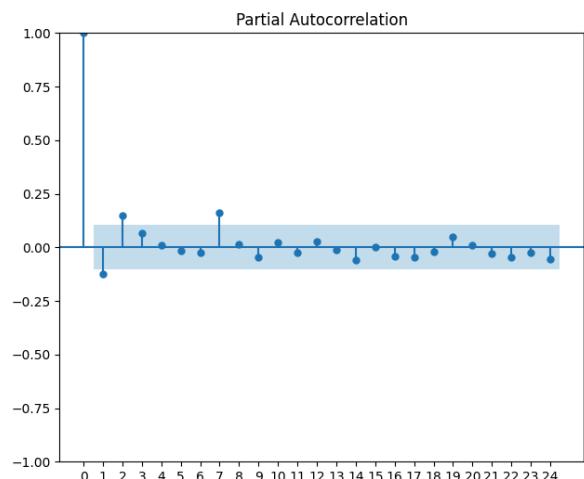
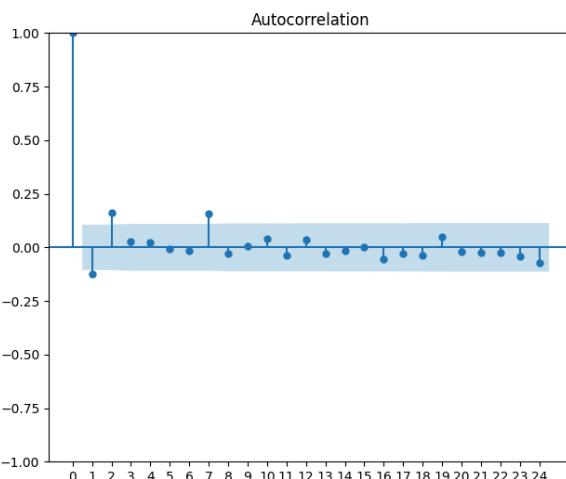
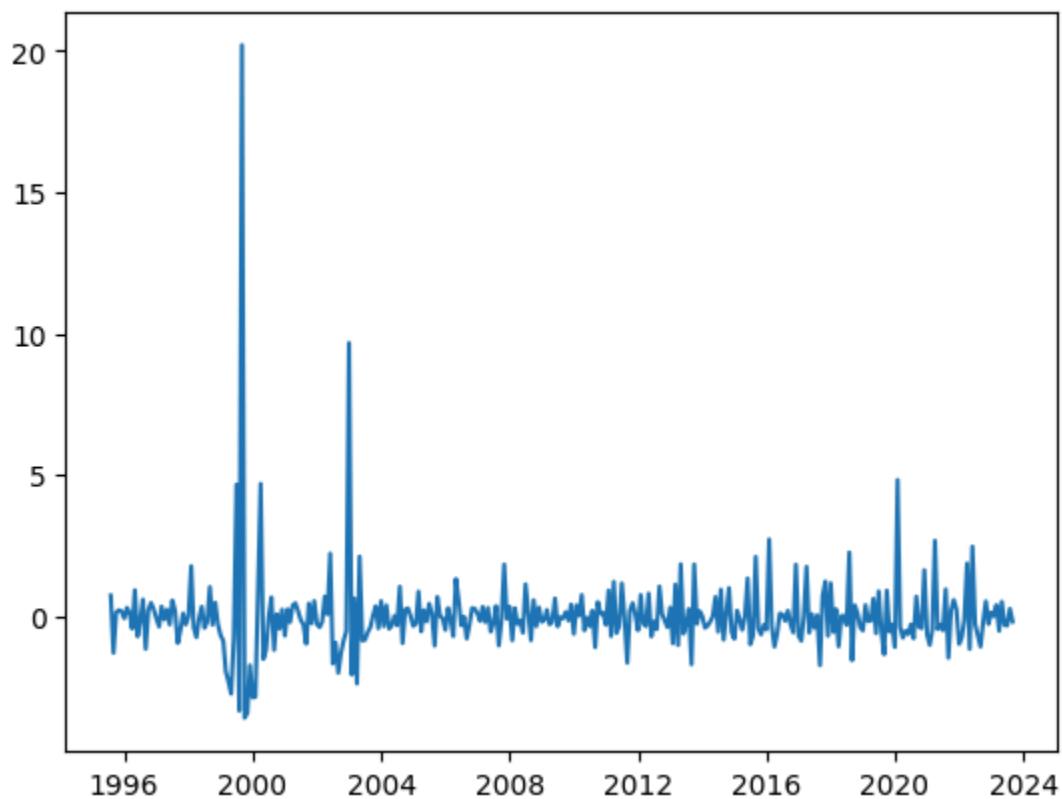
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals

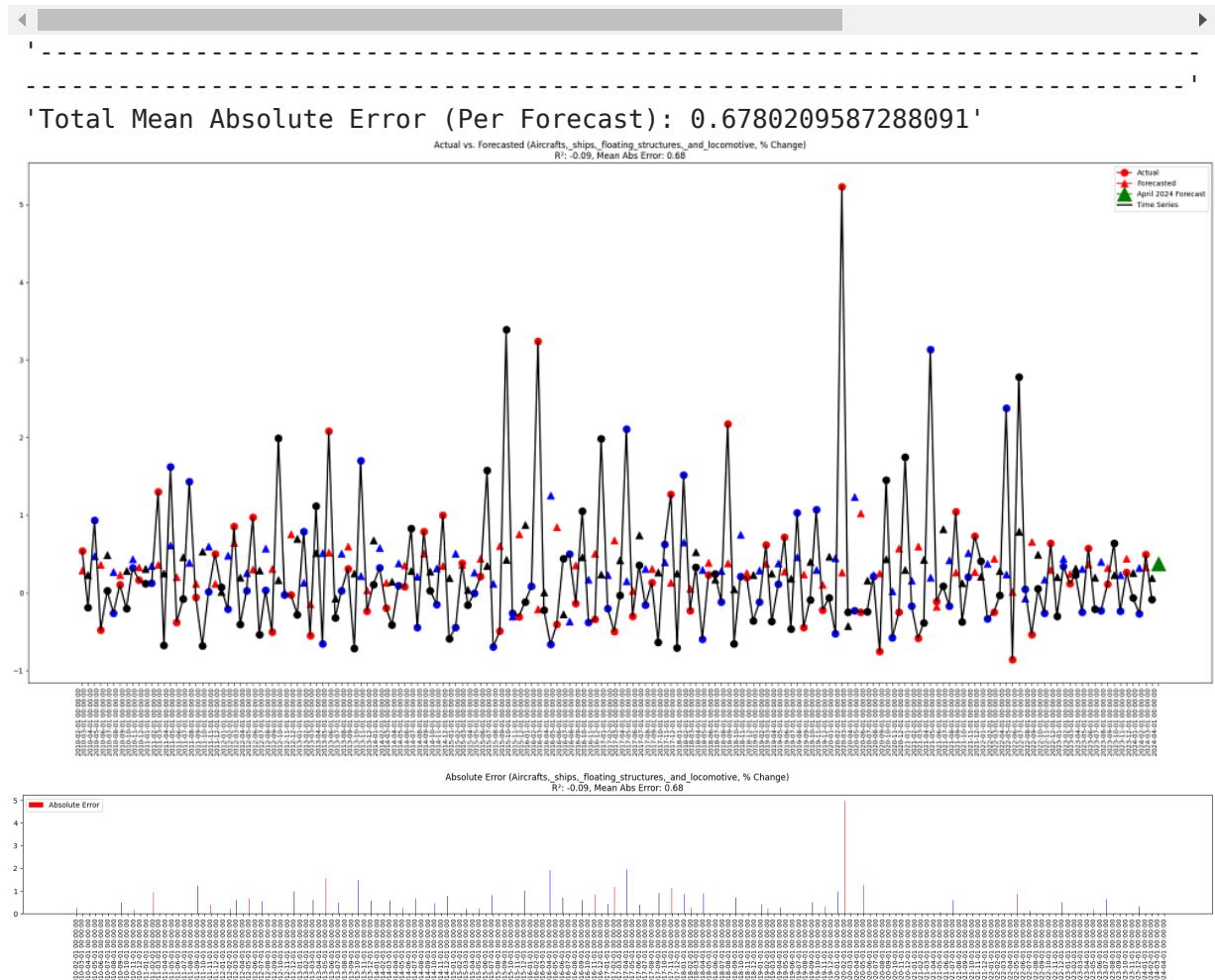


Time series assumptions are met.

'-----'  
-----'  
'Data of Actual vs Forecasted values with error metrics:'

| Aircrafts,_ships,_floating_structures,_and_locomotive | actual    | forecast | error     | abs_error |
|---|-----------|----------|-----------|-----------|
| 2010-02-01  | 0.540823  | 0.290951 | 0.249872  | 0.24987   |
| 2010-03-01  | -0.187896 | 0.228895 | -0.416791 | 0.41679   |
| 2010-04-01  | 0.933865  | 0.474655 | 0.459210  | 0.45921   |
| 2010-05-01  | -0.476219 | 0.359026 | -0.835245 | 0.83524   |
| 2010-06-01  | 0.025214  | 0.486026 | -0.460811 | 0.46081   |
| ...   | ...       | ...      | ...       | ...       |
| 2023-12-01  | -0.066537 | 0.255122 | -0.321659 | 0.32165   |
| 2024-01-01  | -0.266207 | 0.320834 | -0.587041 | 0.58704   |
| 2024-02-01  | 0.492755  | 0.327727 | 0.165028  | 0.16502   |
| 2024-03-01  | -0.083545 | 0.188960 | -0.272505 | 0.27250   |
| 2024-04-01  | NaN       | 0.372016 | NaN       | NaN       |

171 rows × 6 columns



## Machinery\_&\_Equipment

```
In [65]: df = df_export_ANALYSIS.copy()
name = df.columns[25]
display(f"Component: {name}")

'Component: Machinery_&_Equipment'
```

```
In [66]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Machinery_&_Equipment |
|------------|-----------------------|
| 1995-02-01 | 0.007865              |
| 1995-03-01 | 0.200336              |
| 1995-04-01 | -0.152429             |
| 1995-05-01 | 0.175925              |
| 1995-06-01 | 0.107978              |
| ...        | ...                   |
| 2023-11-01 | 0.053624              |
| 2023-12-01 | -0.090947             |
| 2024-01-01 | 0.004479              |
| 2024-02-01 | 0.082038              |
| 2024-03-01 | 0.051000              |

350 rows × 1 columns

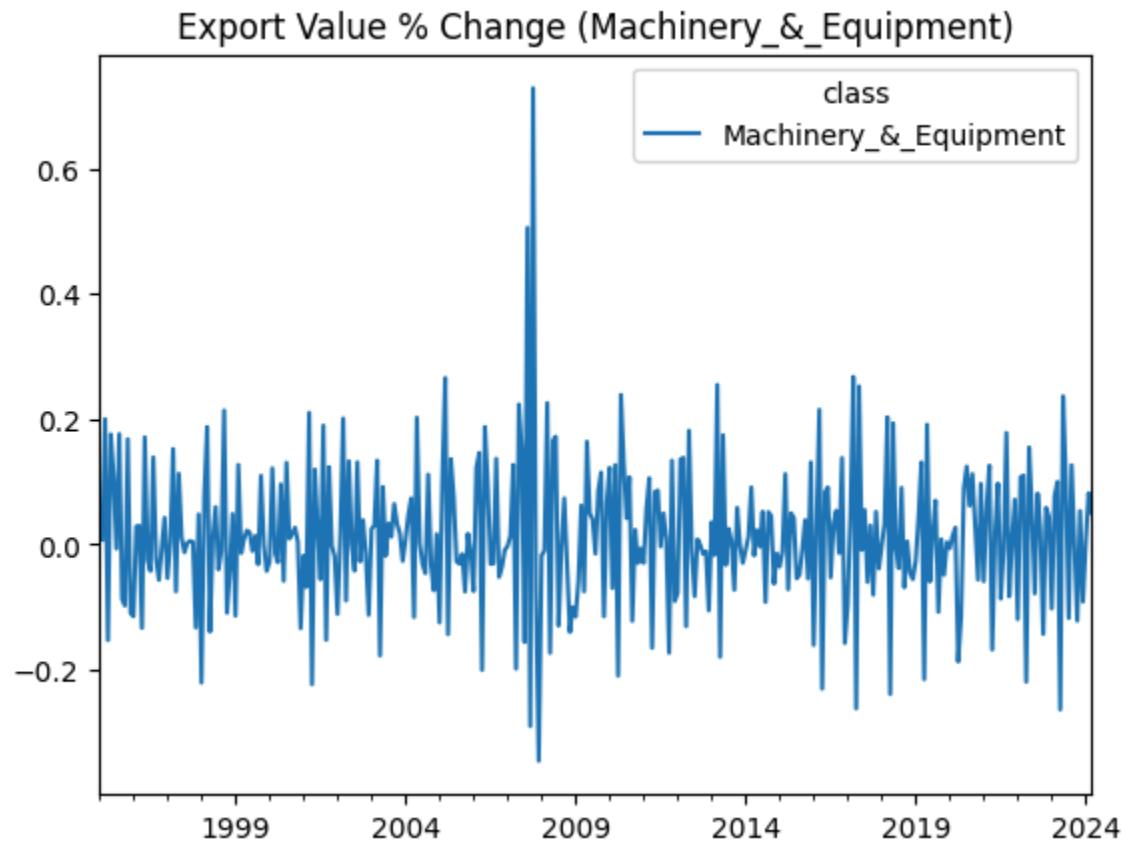
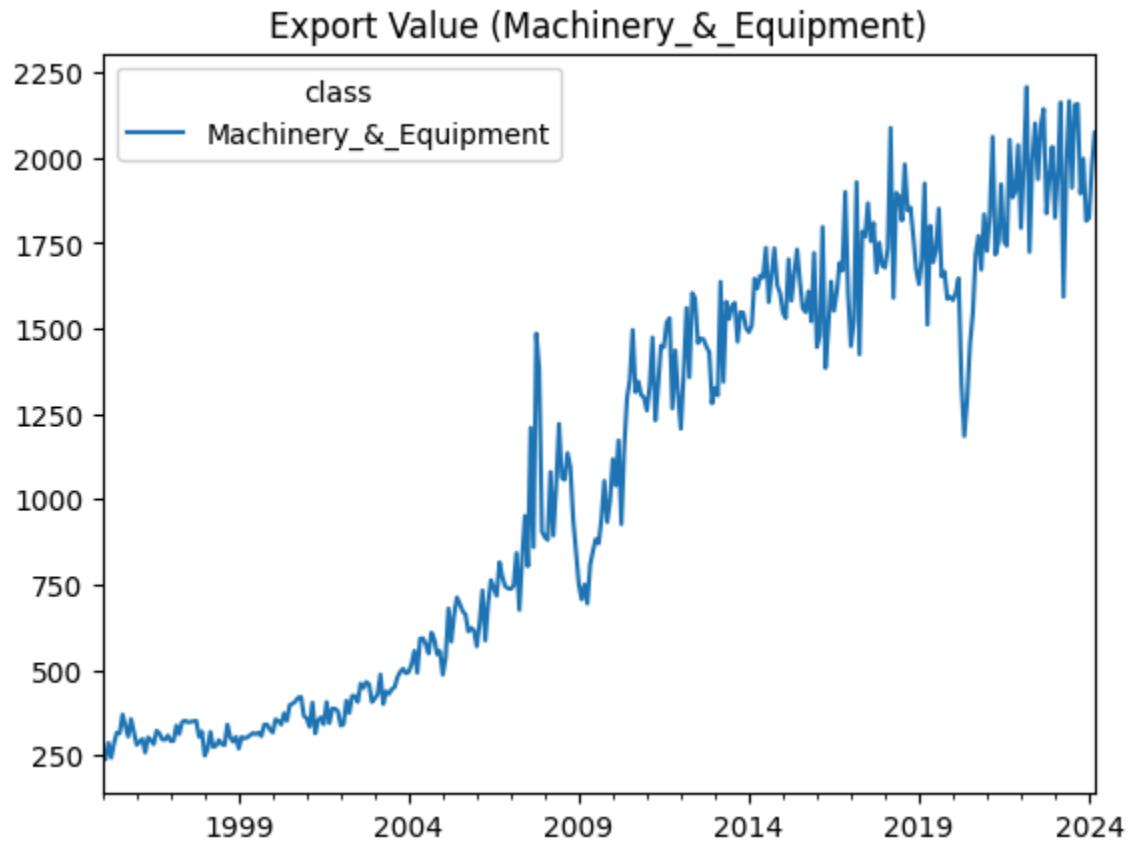
```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -4.769996604383572
p-value: 6.204812010425586e-05
Critical Value 1%: -3.4499043309021955
Critical Value 5%: -2.870154885338925
Critical Value 10%: -2.5713597190254385
is_stationary: True
```

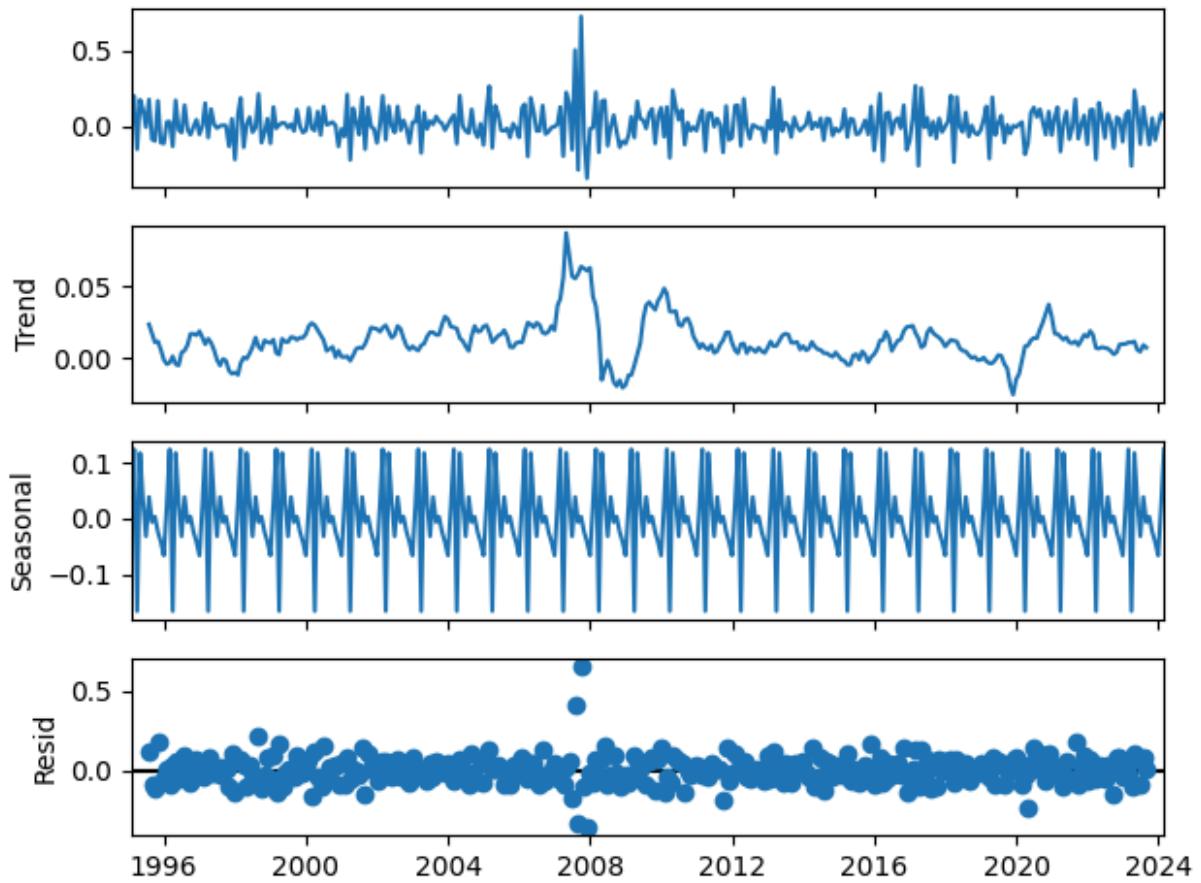
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.07181604777890156
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

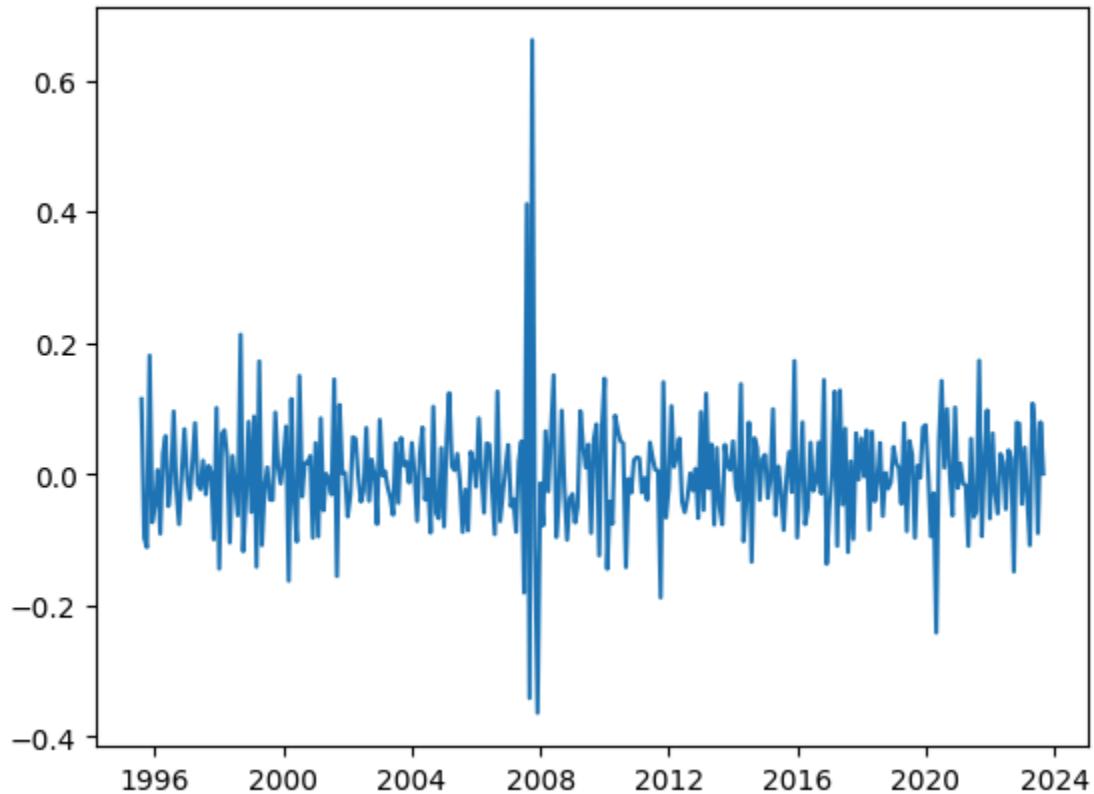
```
result = kpss(series, regression='c')
```

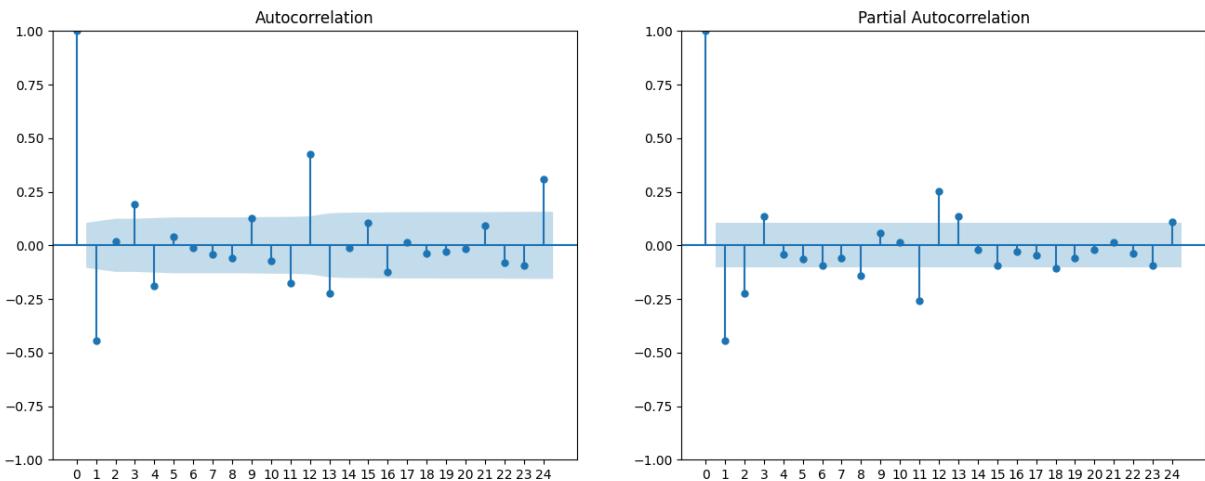




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

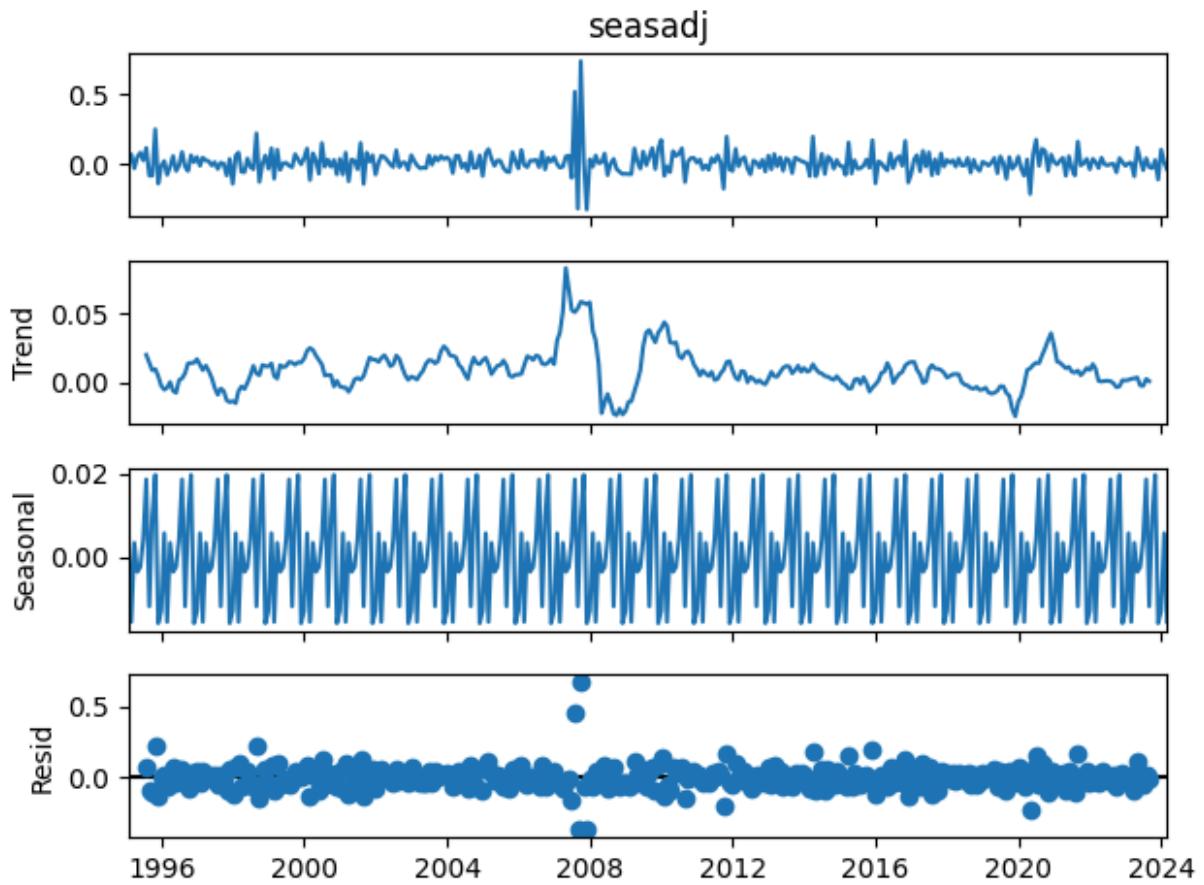
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted): '
Running ADF Test for Stationarity...
ADF Statistic: -10.205677184695563
p-value: 5.813164609753272e-18
Critical Value 1%: -3.4493918438232525
Critical Value 5%: -2.8699298018856574
Critical Value 10%: -2.5712397066390458
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.13527414324388737
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

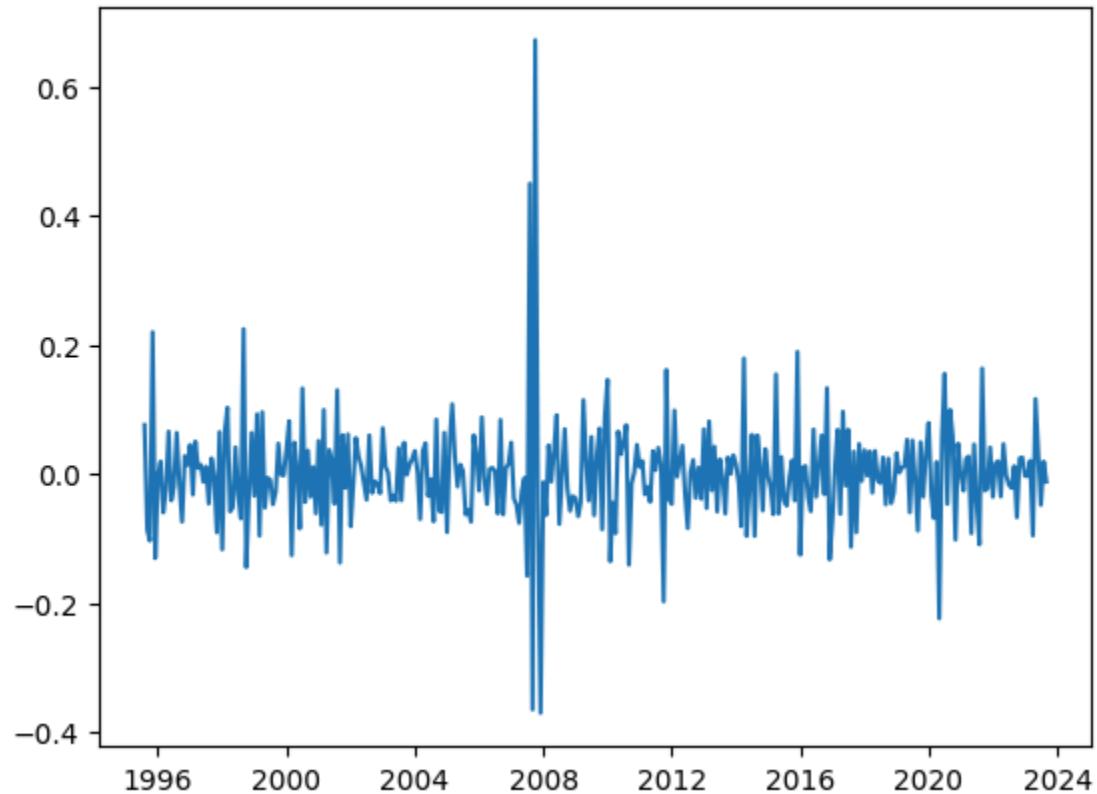
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

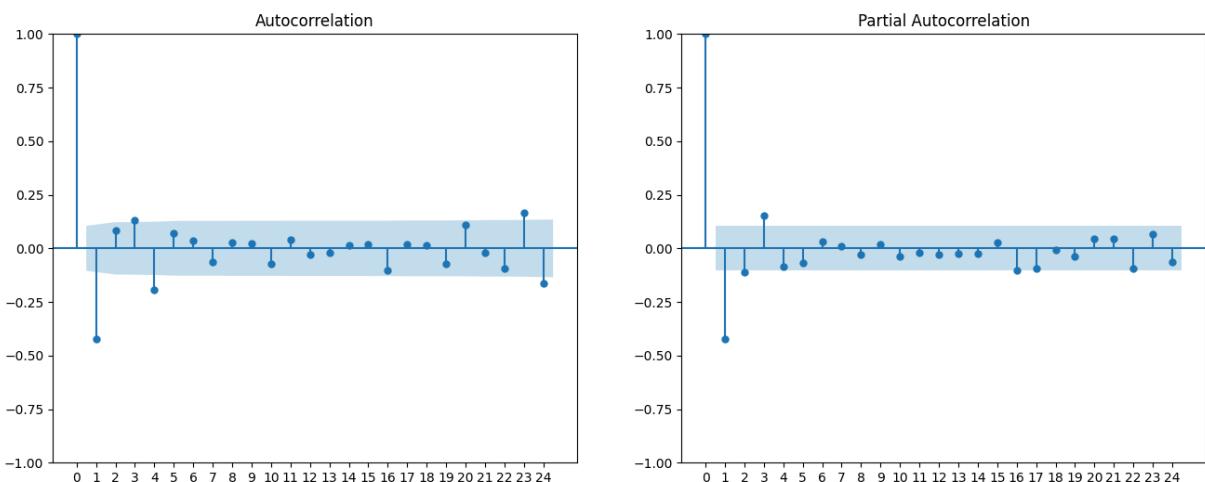
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

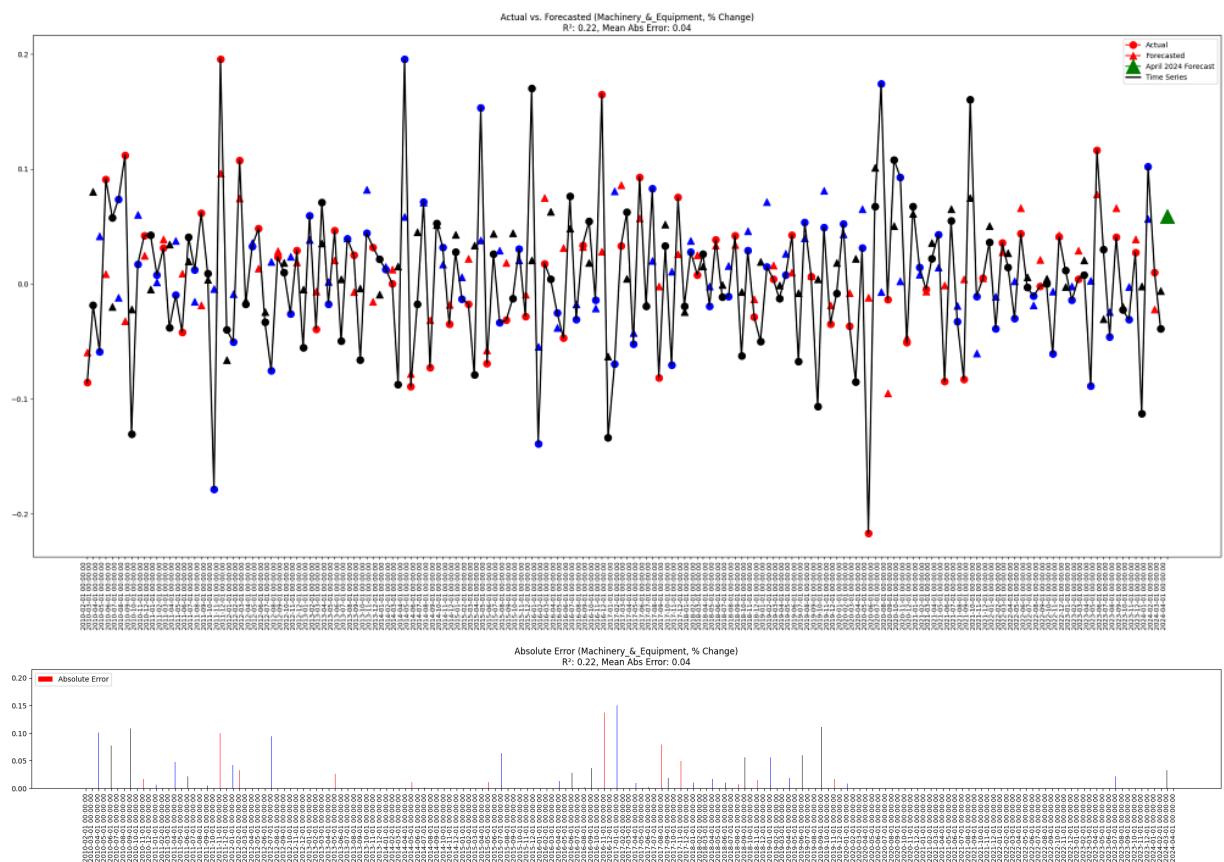
'Data of Actual vs Forecasted values with error metrics:'

| Machinery_&_Equipment | actual    | forecast  | error     | abs_error | squared_error | abs_perc |
|-----------------------|-----------|-----------|-----------|-----------|---------------|----------|
| 2010-02-01            | -0.085667 | -0.059230 | -0.026436 | 0.026436  | 0.000699      |          |
| 2010-03-01            | -0.018548 | 0.080240  | -0.098788 | 0.098788  | 0.009759      |          |
| 2010-04-01            | -0.058910 | 0.041582  | -0.100492 | 0.100492  | 0.010099      |          |
| 2010-05-01            | 0.091145  | 0.008844  | 0.082301  | 0.082301  | 0.006773      |          |
| 2010-06-01            | 0.057591  | -0.019682 | 0.077274  | 0.077274  | 0.005971      |          |
| ...                   | ...       | ...       | ...       | ...       | ...           | ...      |
| 2023-12-01            | -0.112859 | -0.001825 | -0.111034 | 0.111034  | 0.012329      |          |
| 2024-01-01            | 0.102267  | 0.056543  | 0.045724  | 0.045724  | 0.002091      |          |
| 2024-02-01            | 0.009883  | -0.022210 | 0.032092  | 0.032092  | 0.001030      |          |
| 2024-03-01            | -0.038750 | -0.006105 | -0.032645 | 0.032645  | 0.001066      |          |
| 2024-04-01            | NaN       | 0.059133  | NaN       | NaN       | NaN           |          |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.04190734467741312'



## Jewellery

```
In [67]: df = df_export_ANALYSIS.copy()
name = df.columns[26]
display(f"Component: {name}")
```

'Component: Jewellery'

```
In [68]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

class Jewellery

| 1995-02-01 | -0.222918 |
|------------|-----------|
| 1995-03-01 | 0.654083  |
| 1995-04-01 | -0.273469 |
| 1995-05-01 | 0.124966  |
| 1995-06-01 | 0.104088  |
| ...        | ...       |
| 2023-11-01 | -0.057911 |
| 2023-12-01 | -0.220355 |
| 2024-01-01 | 0.260761  |
| 2024-02-01 | 0.634558  |
| 2024-03-01 | -0.389364 |

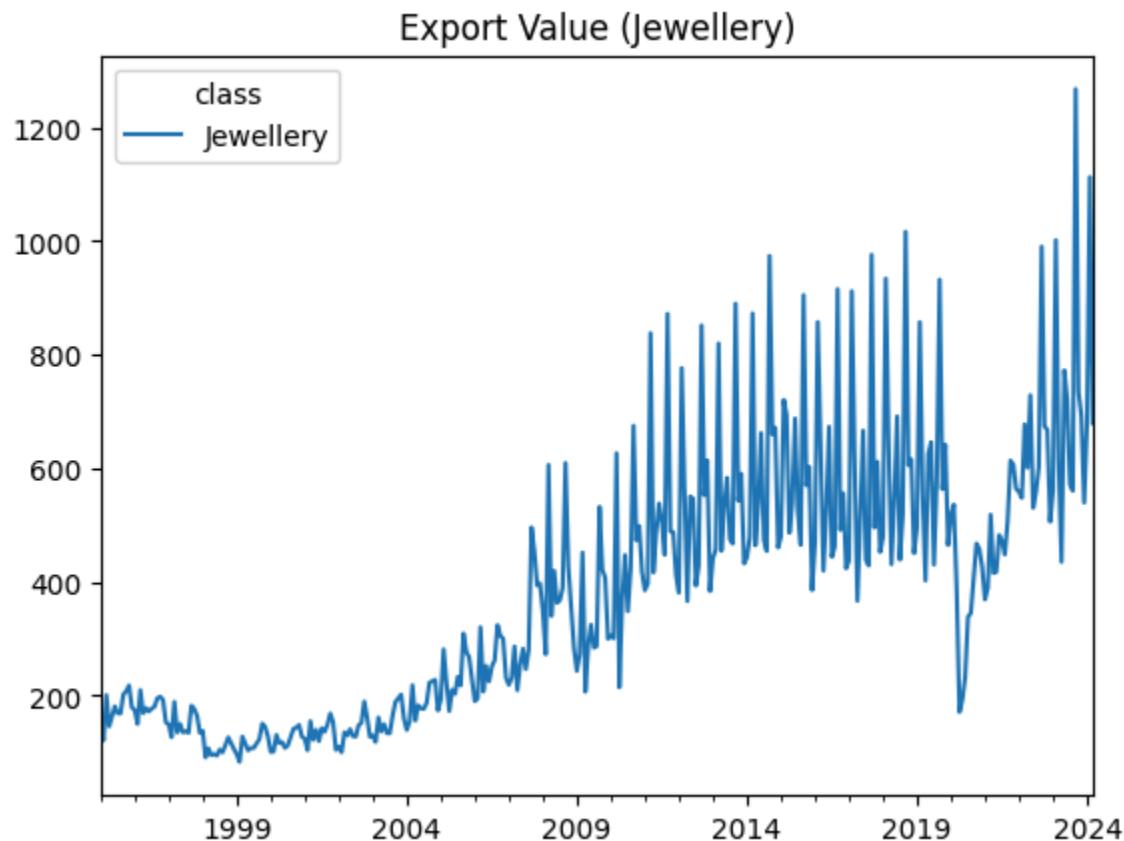
350 rows × 1 columns

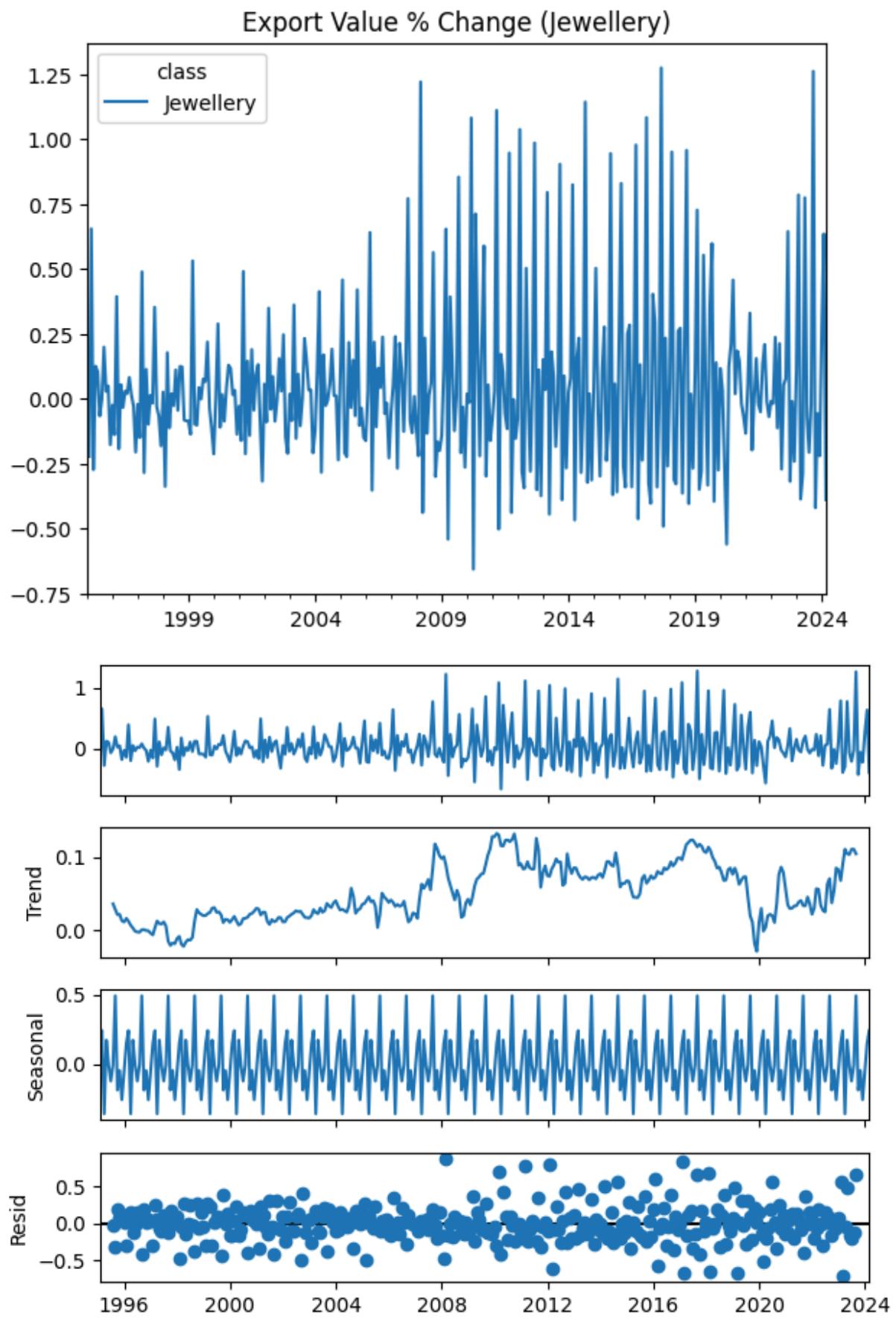
'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

```
Running ADF Test for Stationarity...
ADF Statistic: -2.511627126403834
p-value: 0.11267088930593122
Critical Value 1%: -3.449962981927952
Critical Value 5%: -2.870180642420163
Critical Value 10%: -2.5713734527352607
is_stationary: False
```

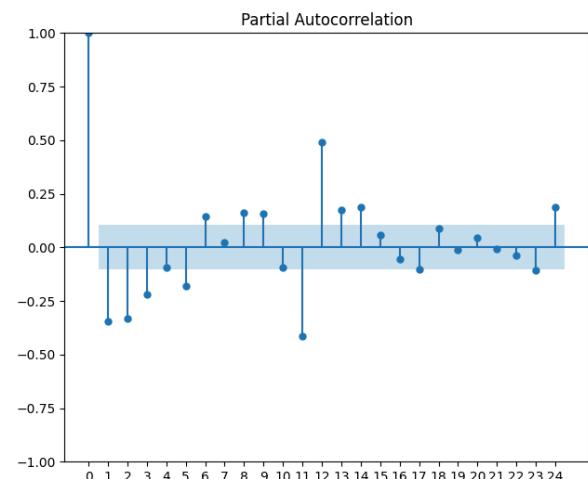
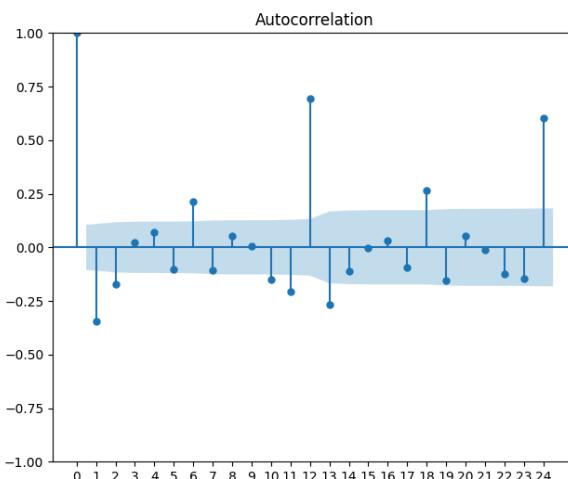
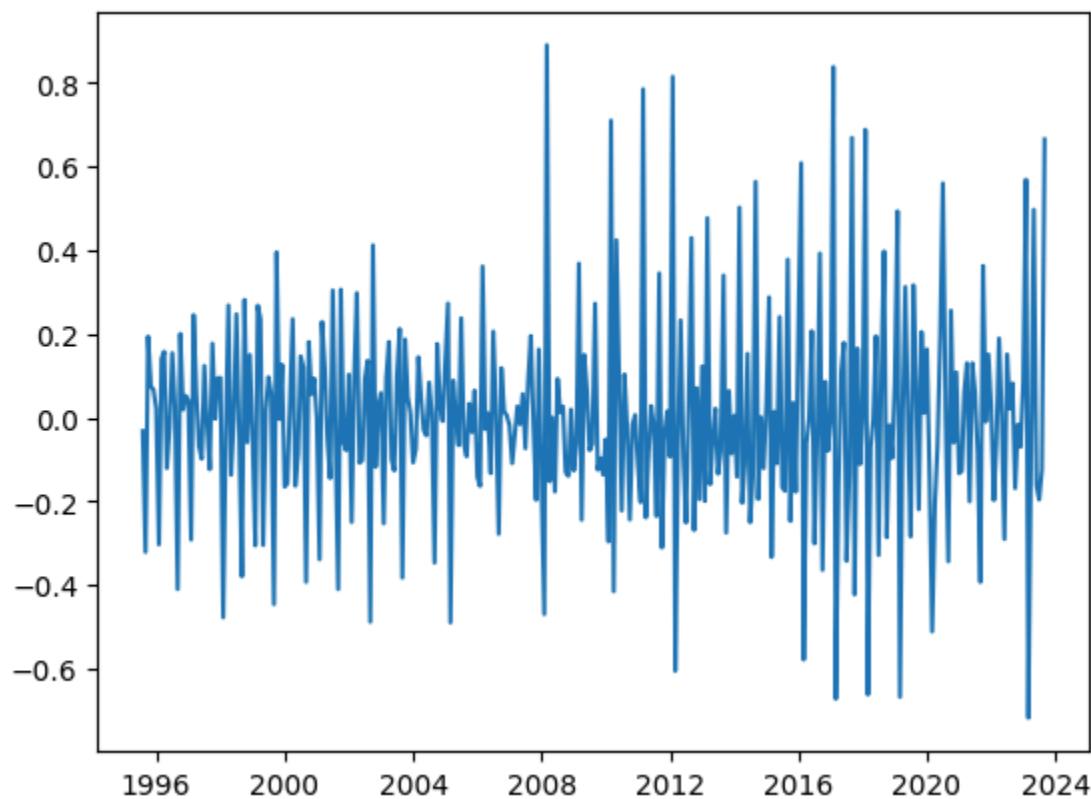
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.35782335776750956
p-value: 0.09533475958297001
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...





### Residuals



Time series assumptions are not met. Further investigation needed.

False

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

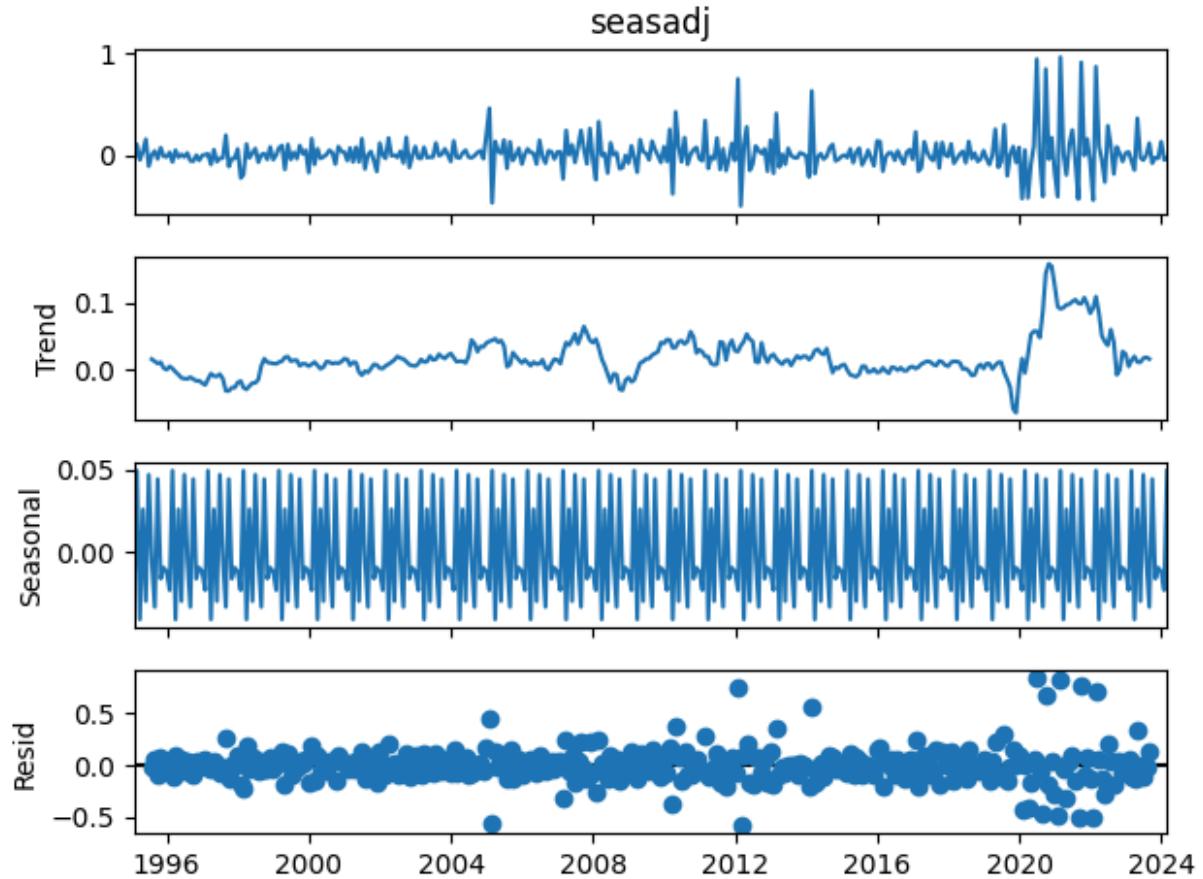
```
Running ADF Test for Stationarity...
ADF Statistic: -7.8555683377077115
p-value: 5.454572693881563e-12
Critical Value 1%: -3.4494474563375737
Critical Value 5%: -2.8699542285903887
Critical Value 10%: -2.5712527305187987
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.33889979395690045
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

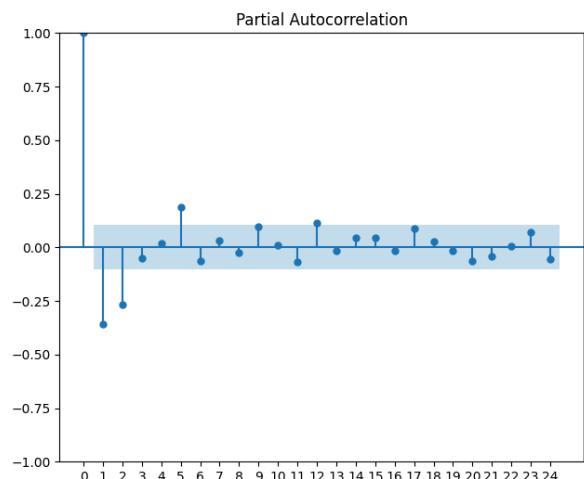
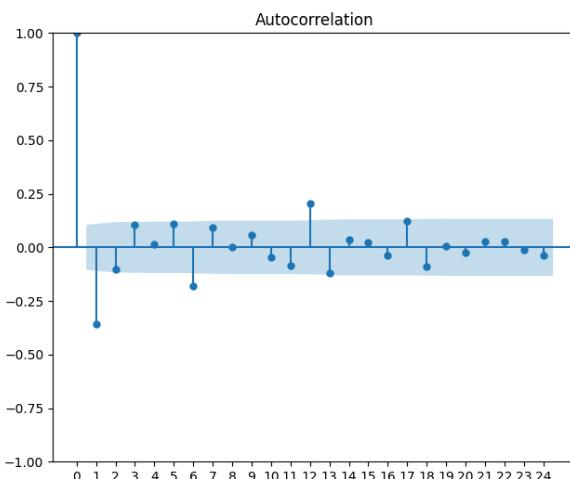
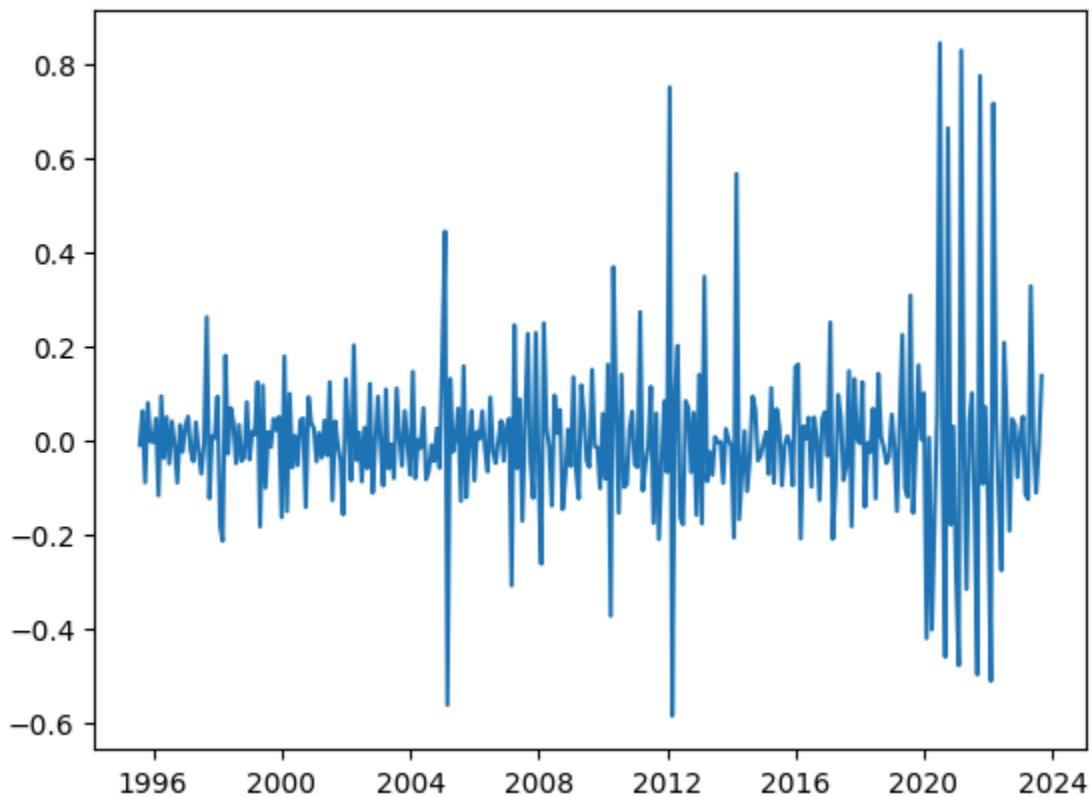
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals



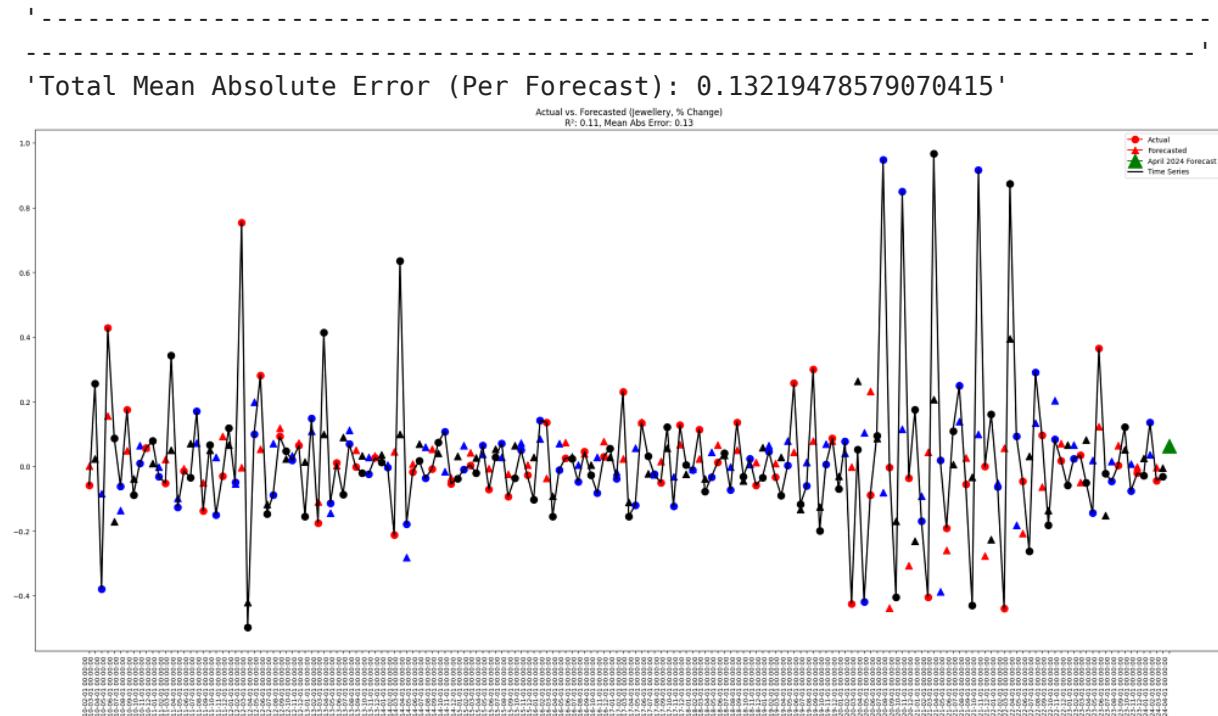
Time series assumptions are met.

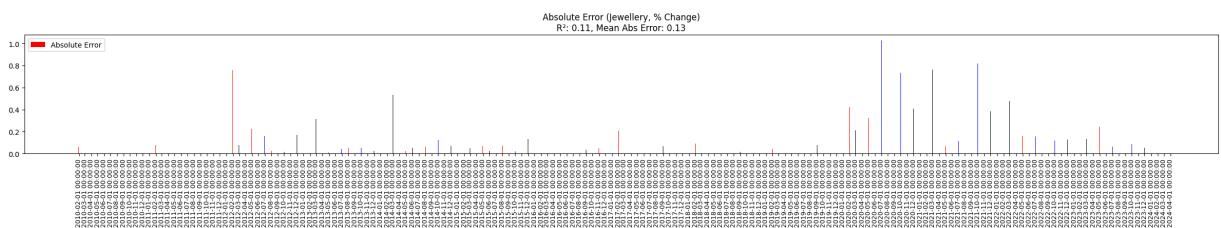
'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Jewellery  | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_error |
|------------|-----------|-----------|-----------|-----------|---------------|----------------------|
| 2010-02-01 | -0.058634 | 0.001552  | -0.060186 | 0.060186  | 0.003622      | 1.026472             |
| 2010-03-01 | 0.255059  | 0.022881  | 0.232178  | 0.232178  | 0.053907      | 0.910292             |
| 2010-04-01 | -0.379065 | -0.083780 | -0.295285 | 0.295285  | 0.087193      | 0.778982             |
| 2010-05-01 | 0.427789  | 0.155730  | 0.272059  | 0.272059  | 0.074016      | 0.635966             |
| 2010-06-01 | 0.087046  | -0.171788 | 0.258834  | 0.258834  | 0.066995      | 2.973538             |
| ...        | ...       | ...       | ...       | ...       | ...           | ...                  |
| 2023-12-01 | -0.028825 | 0.024274  | -0.053100 | 0.053100  | 0.002820      | 1.842109             |
| 2024-01-01 | 0.135504  | 0.035202  | 0.100301  | 0.100301  | 0.010060      | 0.740210             |
| 2024-02-01 | -0.044451 | -0.003733 | -0.040718 | 0.040718  | 0.001658      | 0.916018             |
| 2024-03-01 | -0.032116 | -0.005585 | -0.026531 | 0.026531  | 0.000704      | 0.826114             |
| 2024-04-01 | NaN       | 0.062304  | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns





## Chemicals\_&\_Petro-chemical\_Products

```
In [69]: df = df_export_ANALYSIS.copy()
name = df.columns[27]
display(f"Component: {name}")
```

'Component: Chemicals\_&\_Petro-chemical\_Products'

```
In [70]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value.pkl', 'rb') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/{name}_accuracy.pkl")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Chemicals_&_Petro-chemical_Products |
|------------|-------------------------------------|
| 1995-02-01 | -0.013421                           |
| 1995-03-01 | 0.288744                            |
| 1995-04-01 | -0.038704                           |
| 1995-05-01 | 0.469981                            |
| 1995-06-01 | -0.195582                           |
| ...        | ...                                 |
| 2023-11-01 | 0.006685                            |
| 2023-12-01 | -0.097435                           |
| 2024-01-01 | 0.033378                            |
| 2024-02-01 | -0.007009                           |
| 2024-03-01 | 0.091555                            |

350 rows × 1 columns

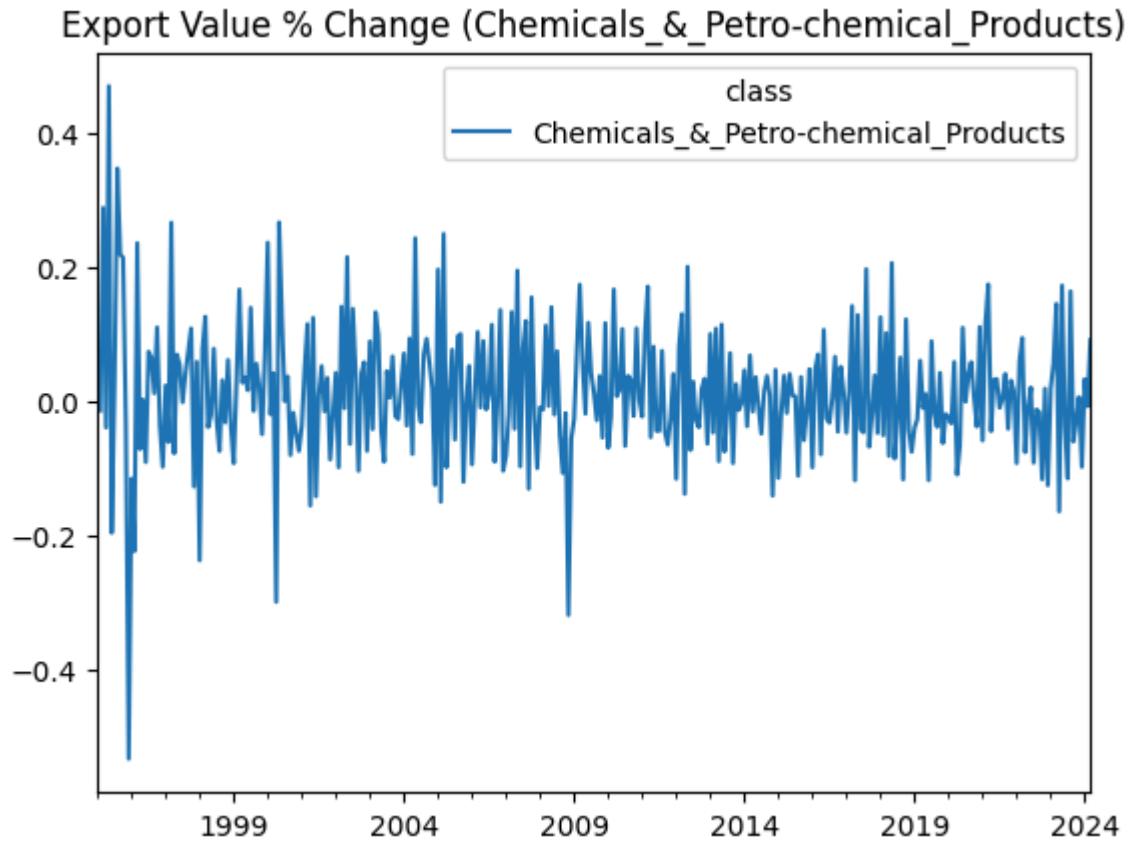
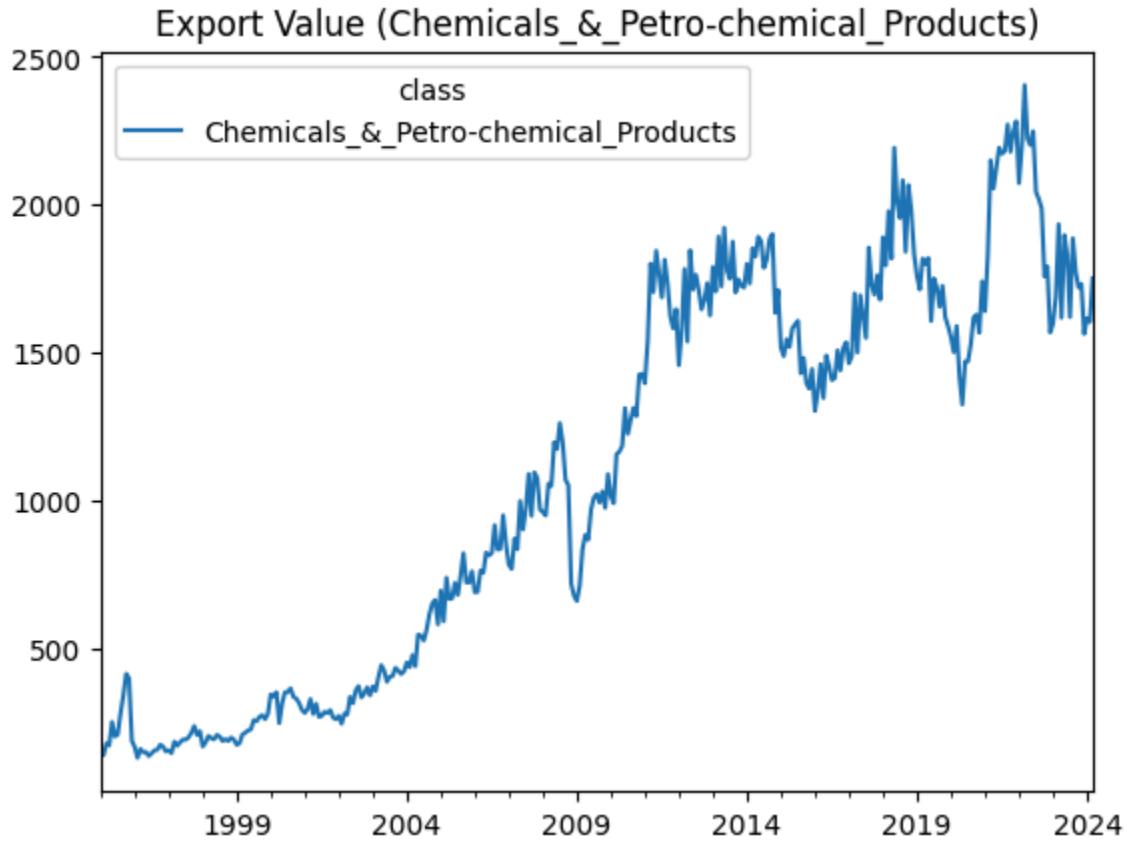
```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -5.330488619537796
p-value: 4.7330976913727265e-06
Critical Value 1%: -3.450141065277327
Critical Value 5%: -2.870258846235788
Critical Value 10%: -2.571415151457764
is_stationary: True
```

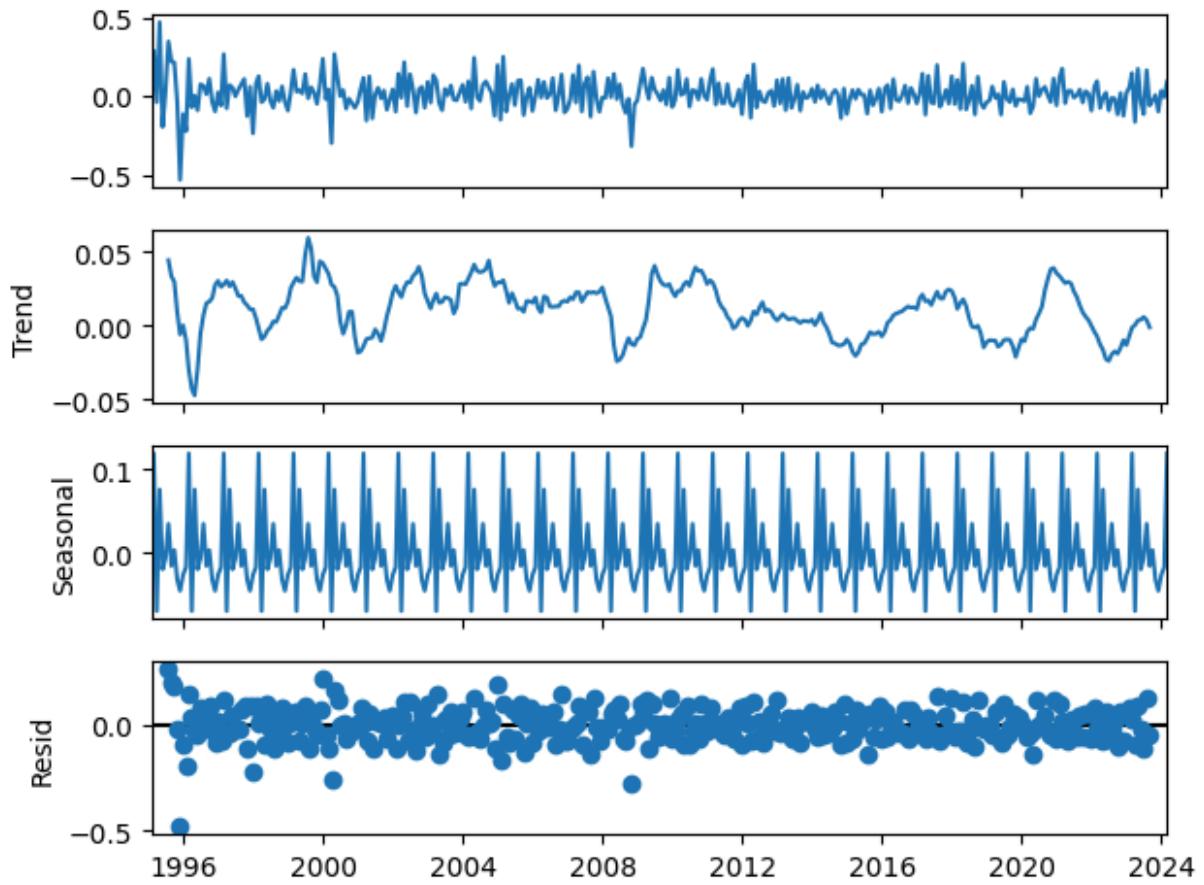
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.2888272765743553
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

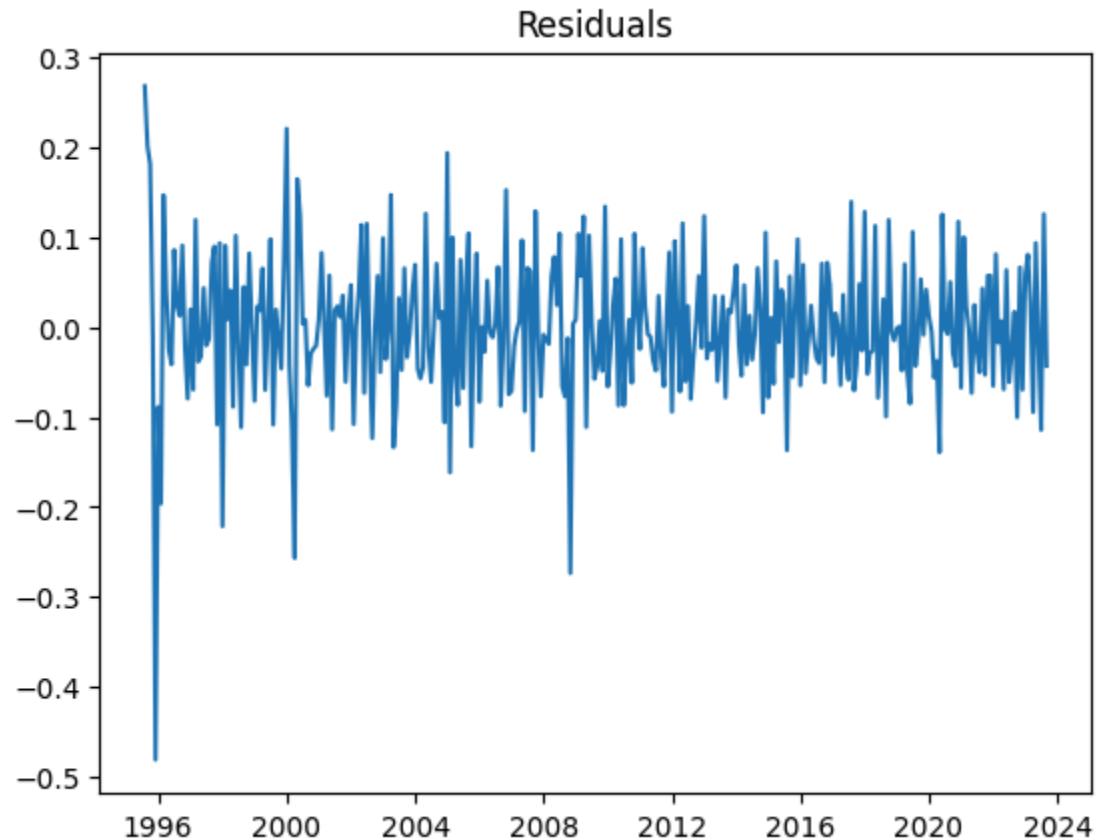
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

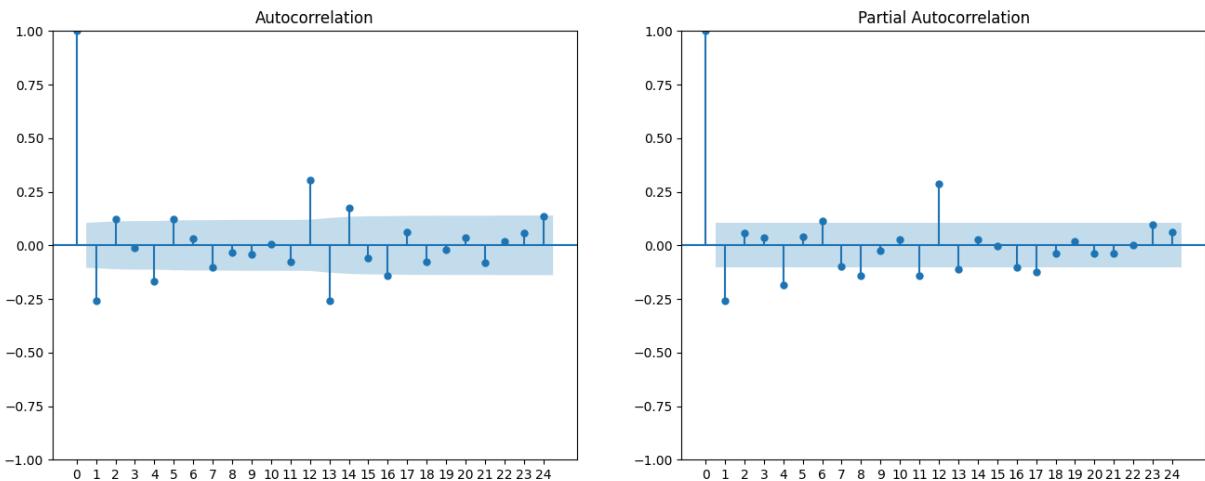
```
result = kpss(series, regression='c')
```





Plotting Residuals for Homoscedasticity Check...





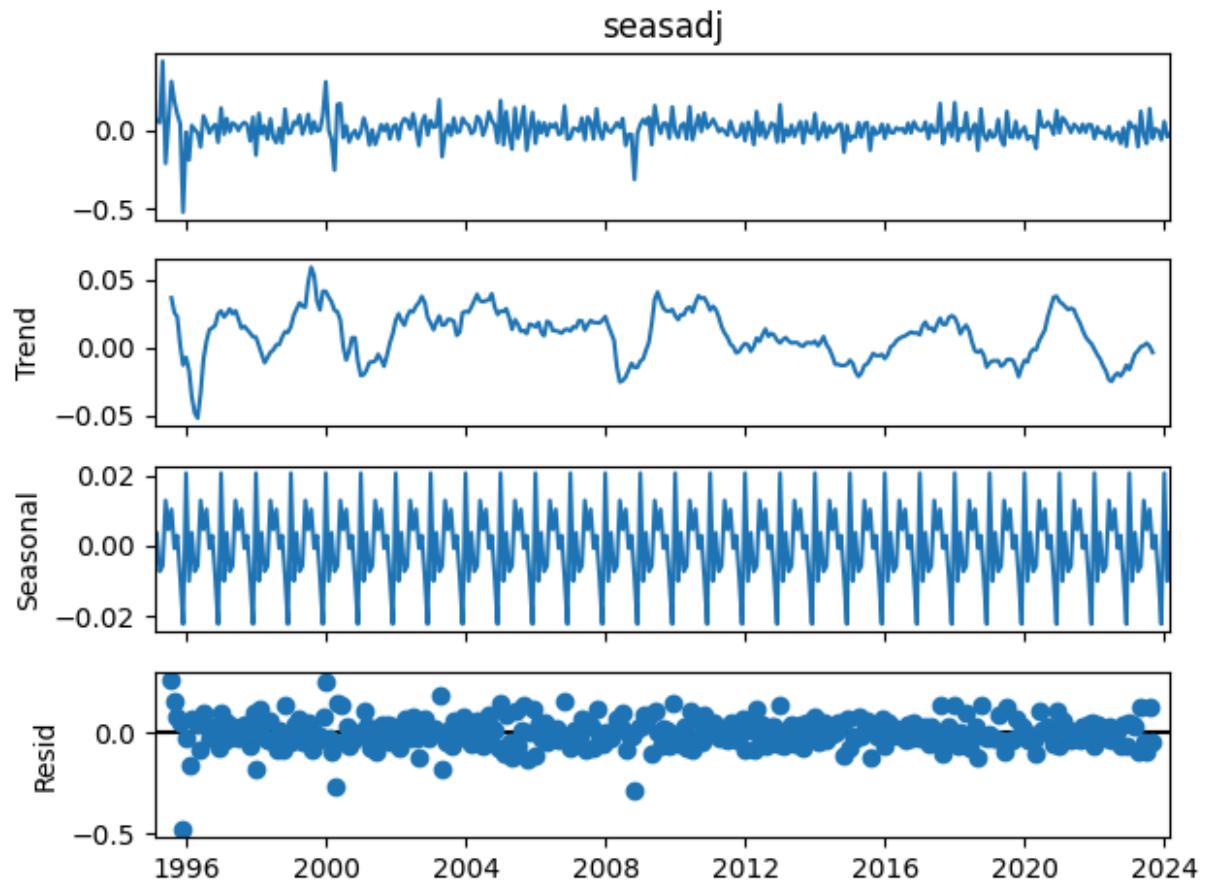
Time series assumptions are met.

True

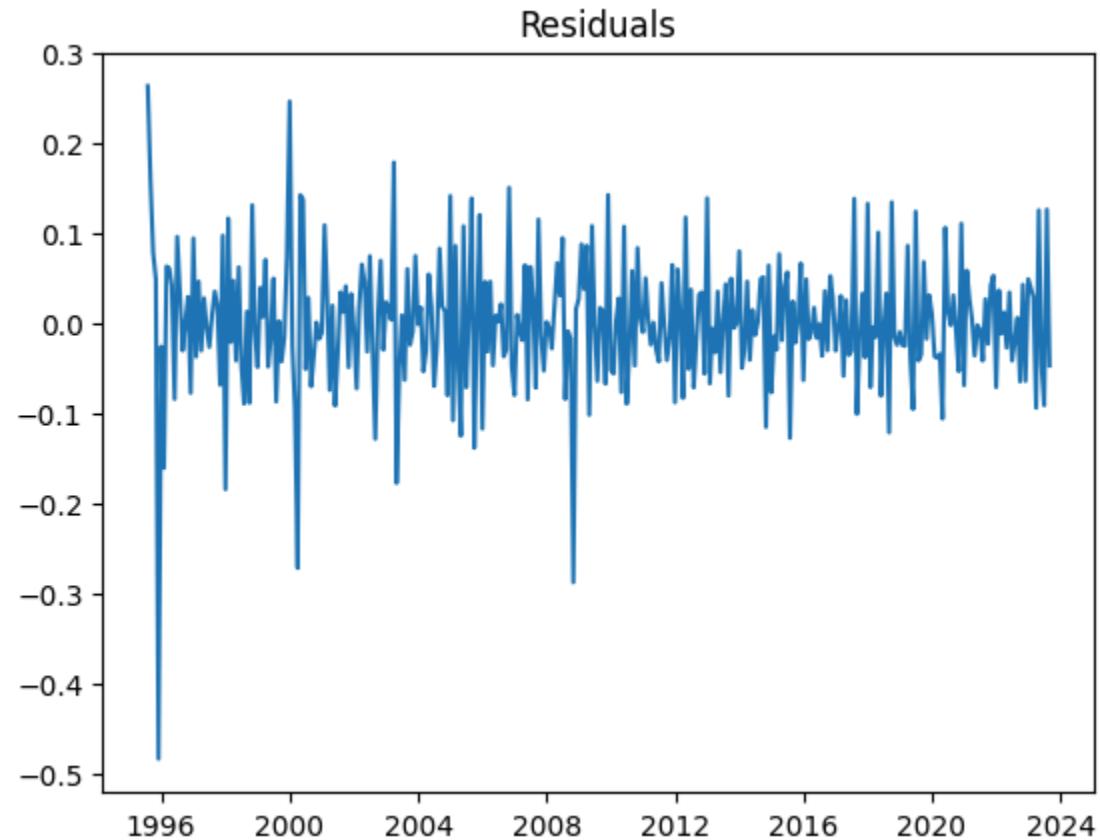
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -5.779666305341256
p-value: 5.161042274379734e-07
Critical Value 1%: -3.4499043309021955
Critical Value 5%: -2.870154885338925
Critical Value 10%: -2.5713597190254385
is_stationary: True
```

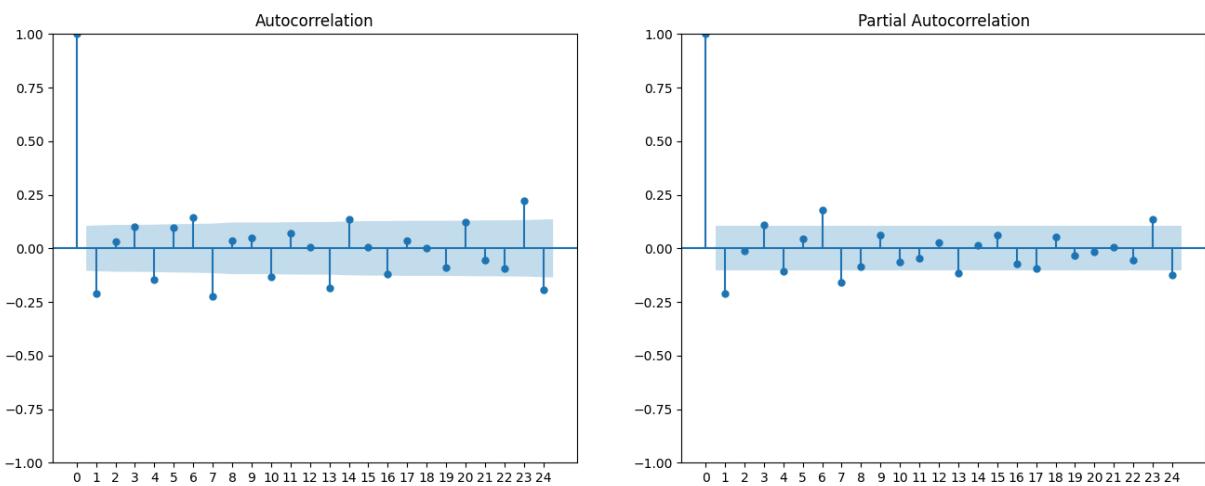
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.3648000618538905
p-value: 0.09232755954573685
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...



Plotting Residuals for Homoscedasticity Check...





Time series assumptions are met.

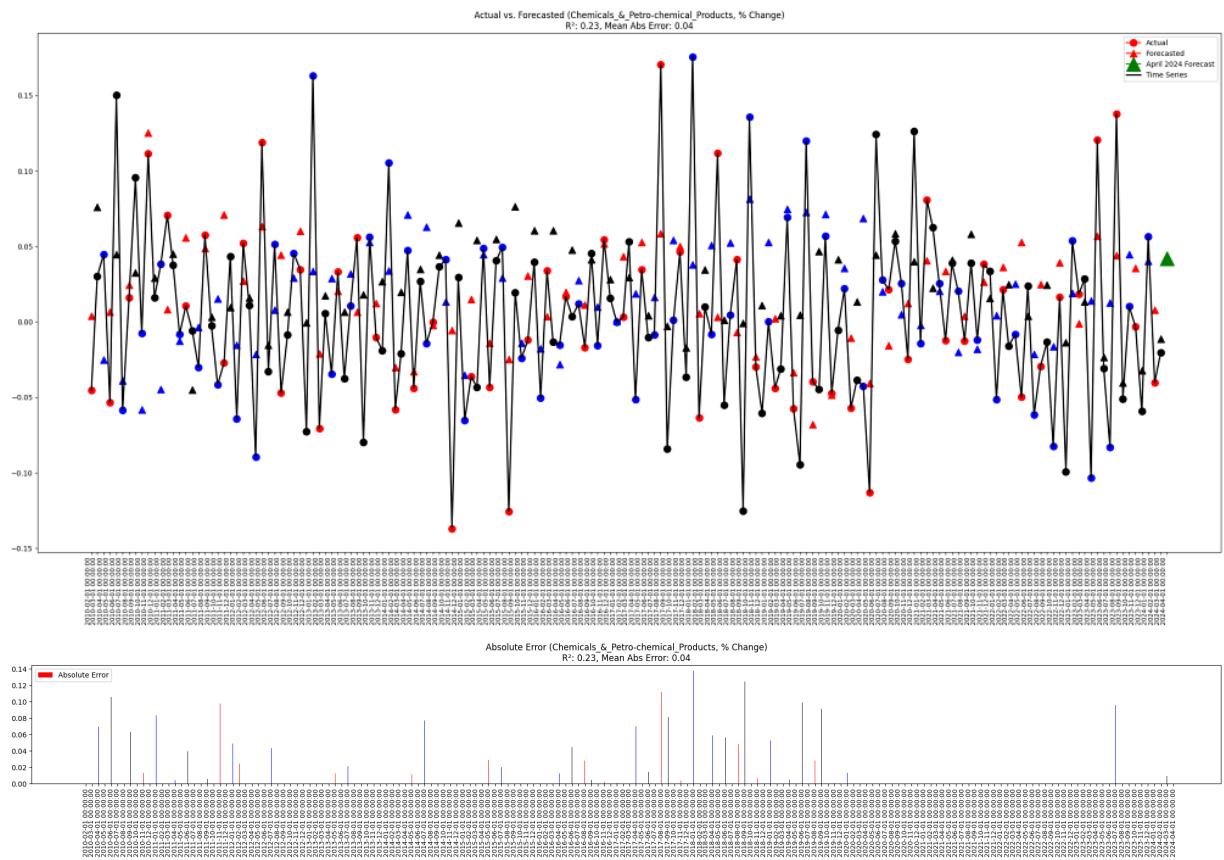
'Data of Actual vs Forecasted values with error metrics:'

| Chemicals_&_Petro-chemical_Products | actual    | forecast  | error     | abs_error | squared_error | abs_percenta |
|-------------------------------------|-----------|-----------|-----------|-----------|---------------|--------------|
| 2010-02-01                          | -0.045251 | 0.004026  | -0.049277 | 0.049277  | 0.002428      |              |
| 2010-03-01                          | 0.030032  | 0.076069  | -0.046038 | 0.046038  | 0.002119      |              |
| 2010-04-01                          | 0.044690  | -0.024999 | 0.069690  | 0.069690  | 0.004857      |              |
| 2010-05-01                          | -0.053401 | 0.006717  | -0.060118 | 0.060118  | 0.003614      |              |
| 2010-06-01                          | 0.150295  | 0.044742  | 0.105553  | 0.105553  | 0.011141      | (            |
| ...                                 | ...       | ...       | ...       | ...       | ...           |              |
| 2023-12-01                          | -0.059319 | -0.032050 | -0.027268 | 0.027268  | 0.000744      | )            |
| 2024-01-01                          | 0.056399  | 0.040303  | 0.016097  | 0.016097  | 0.000259      | (            |
| 2024-02-01                          | -0.040201 | 0.007912  | -0.048113 | 0.048113  | 0.002315      |              |
| 2024-03-01                          | -0.020327 | -0.011144 | -0.009184 | 0.009184  | 0.000084      | )            |
| 2024-04-01                          | NaN       | 0.041863  | NaN       | NaN       | NaN           |              |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.04134805673981834'



## Petroleum\_products

```
In [71]: df = df_export_ANALYSIS.copy()
name = df.columns[28]
display(f"Component: {name}")
```

'Component: Petroleum\_products'

```
In [72]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

class Petroleum\_products

|            |           |
|------------|-----------|
| 1995-02-01 | 1.739015  |
| 1995-03-01 | 0.096014  |
| 1995-04-01 | -0.487913 |
| 1995-05-01 | -0.146384 |
| 1995-06-01 | 0.098935  |
| ...        | ...       |
| 2023-11-01 | -0.152933 |
| 2023-12-01 | 0.058325  |
| 2024-01-01 | -0.246006 |
| 2024-02-01 | -0.171409 |
| 2024-03-01 | 0.142808  |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

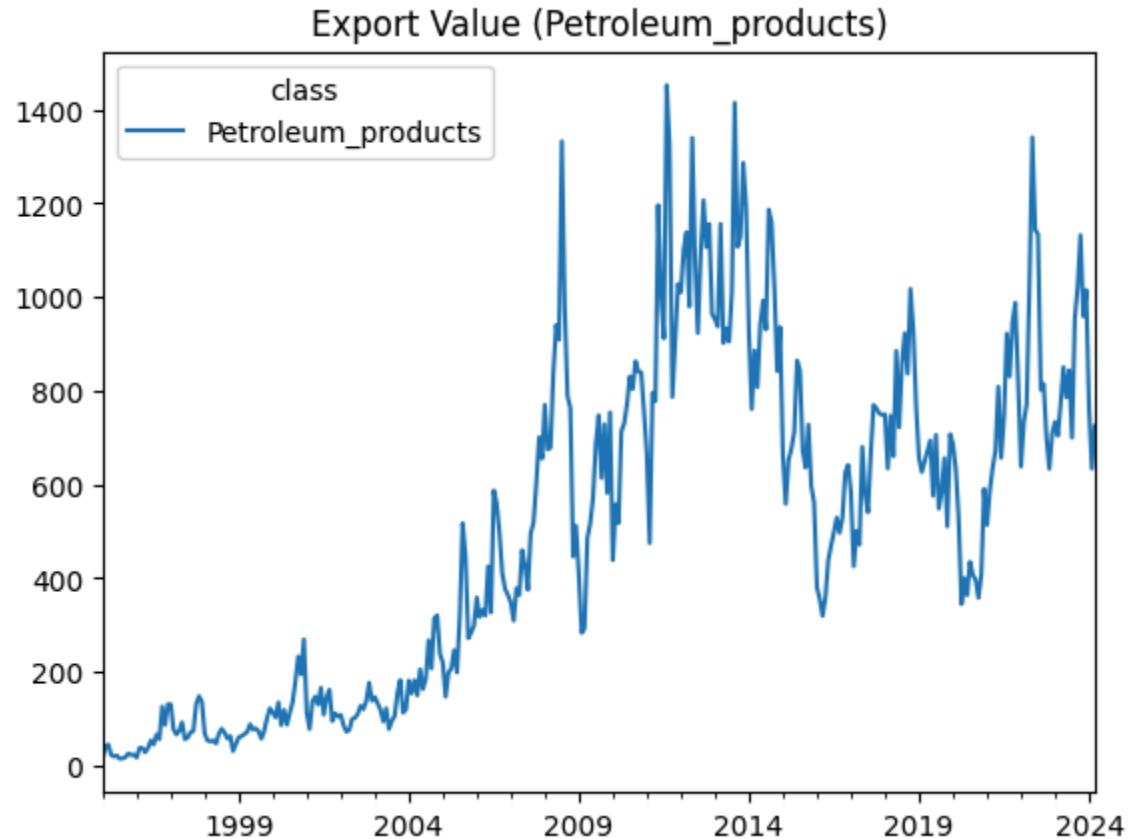
```
Running ADF Test for Stationarity...
ADF Statistic: -4.551095018074174
p-value: 0.0001588073220471269
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

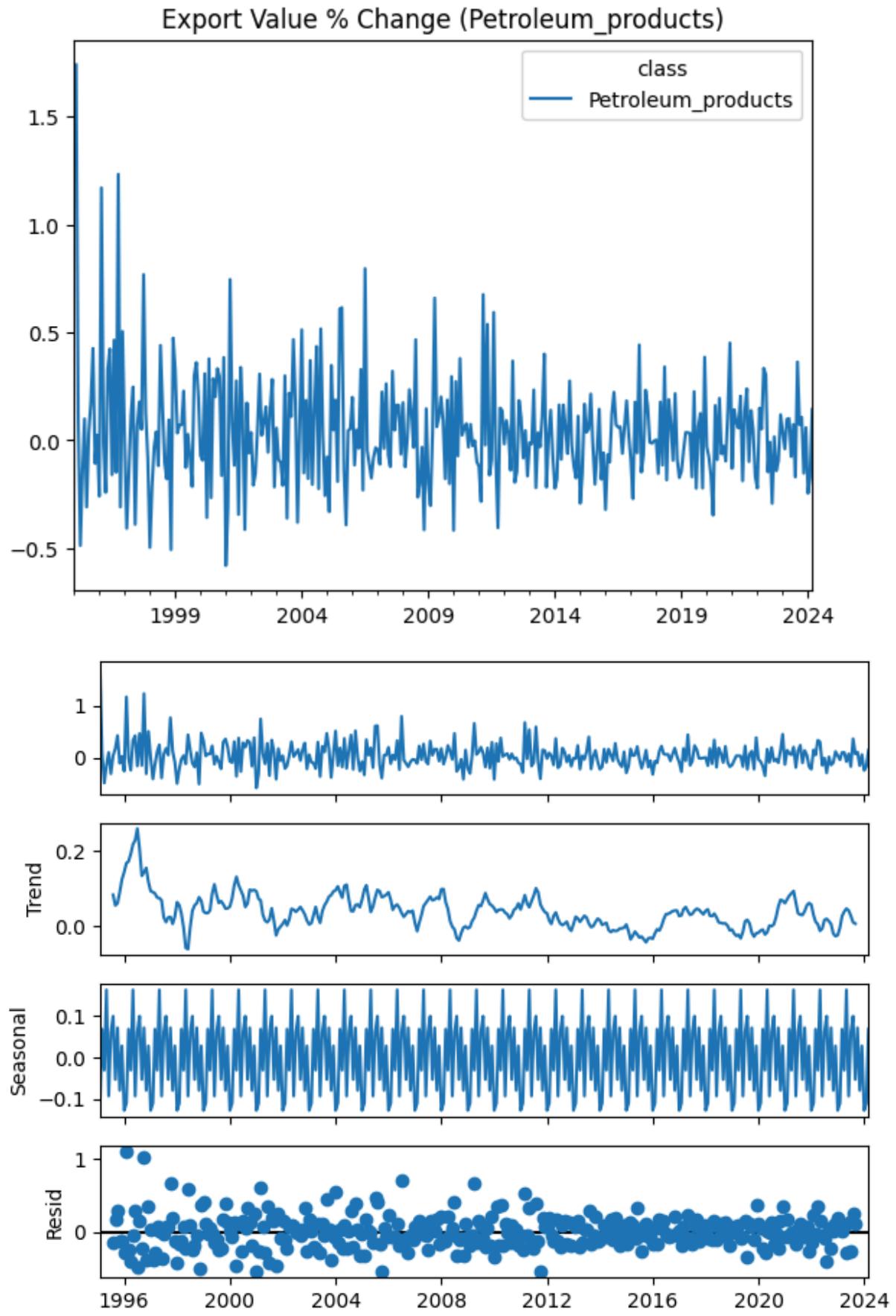
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.8999848419258791
p-value: 0.01
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: False
```

Decomposing the Series...

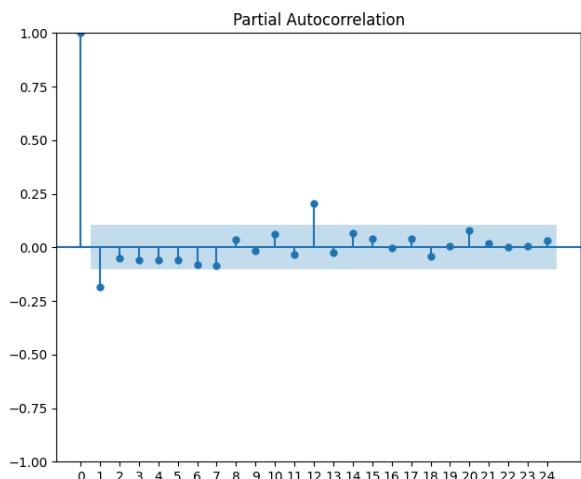
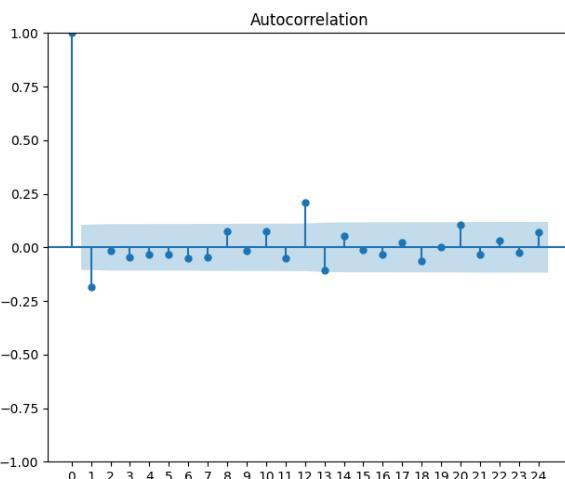
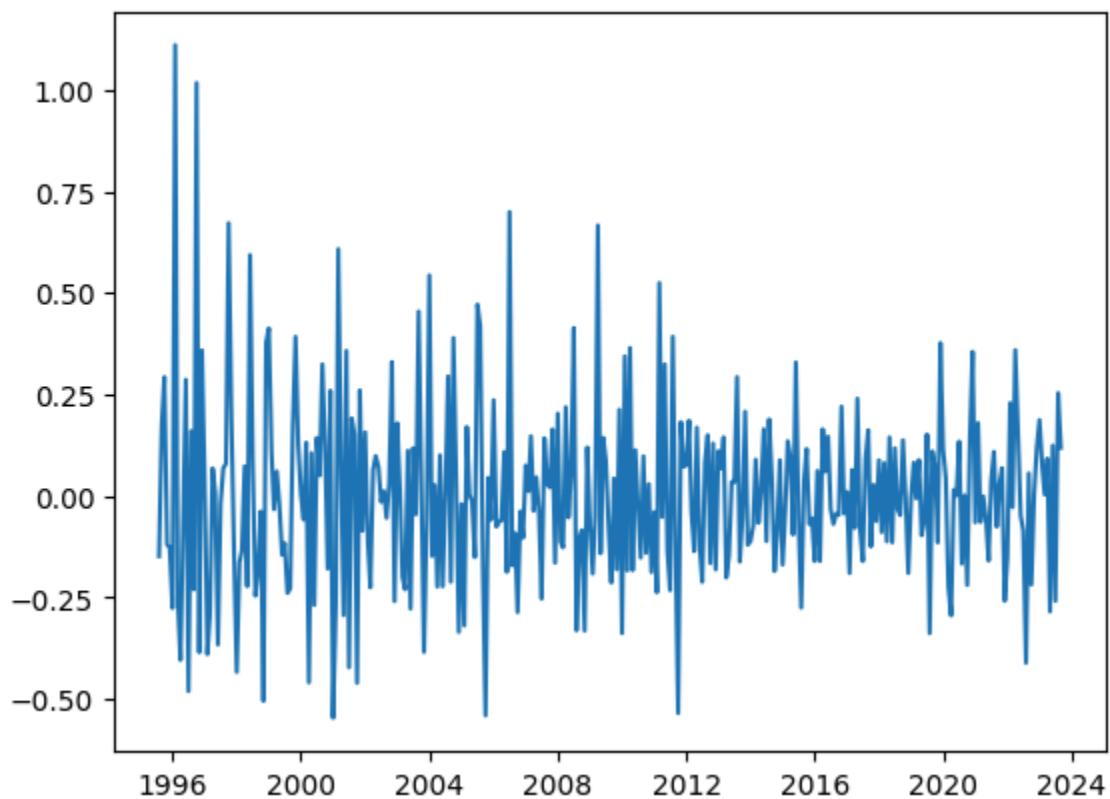
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.
```

```
result = kpss(series, regression='c')
```





## Residuals



Time series assumptions are not met. Further investigation needed.

False

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

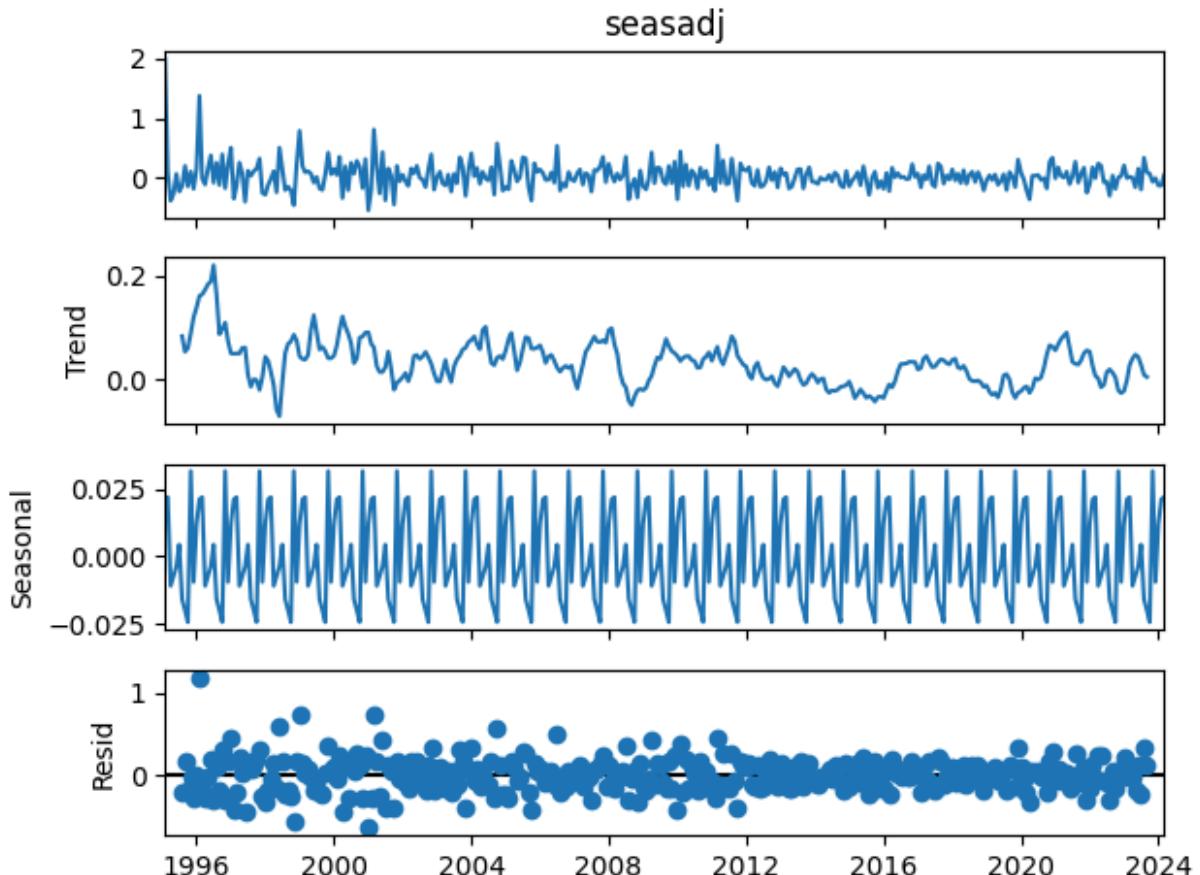
Running ADF Test for Stationarity...  
 ADF Statistic: -4.839394025705354  
 p-value: 4.569455927936921e-05  
 Critical Value 1%: -3.449846029628477  
 Critical Value 5%: -2.870129281376164  
 Critical Value 10%: -2.5713460670144603  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.8389443829970102  
 p-value: 0.01  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: False

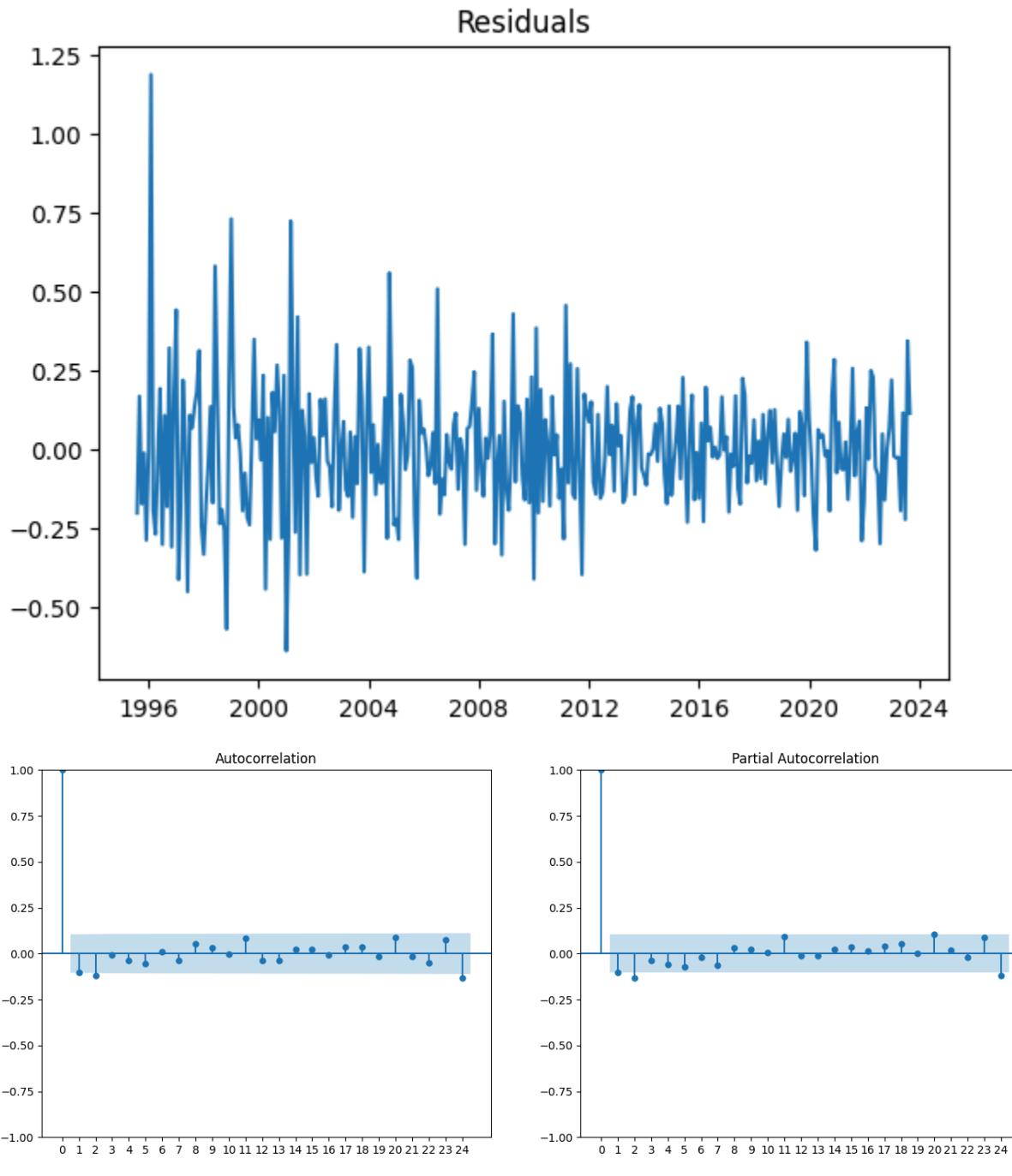
Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...



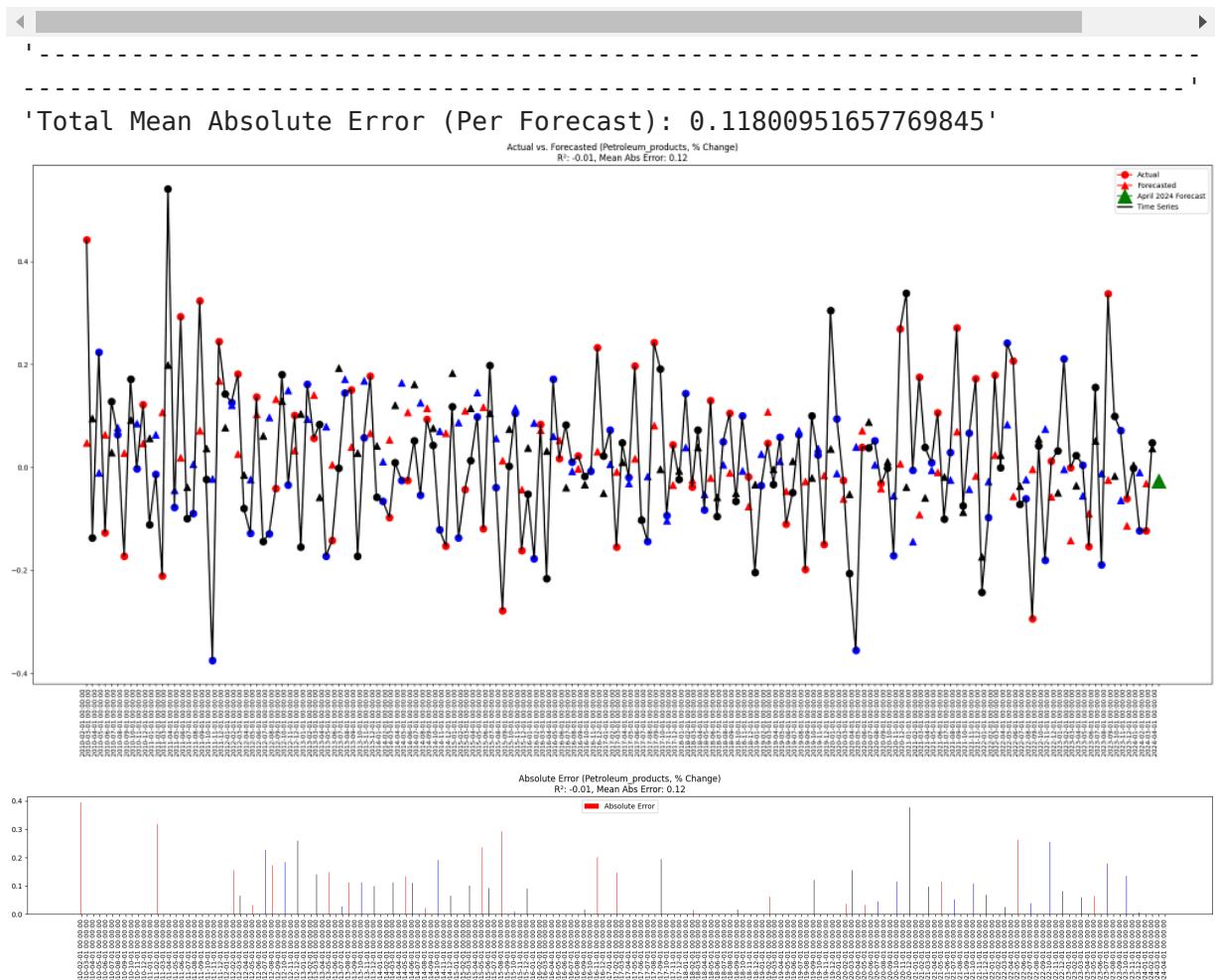
Time series assumptions are not met. Further investigation needed.

'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Petroleum_products | actual    | forecast  | error     | abs_error | squared_error | abs_percenta |
|--------------------|-----------|-----------|-----------|-----------|---------------|--------------|
| 2010-02-01         | 0.441760  | 0.048162  | 0.393597  | 0.393597  | 0.154919      |              |
| 2010-03-01         | -0.136632 | 0.095097  | -0.231729 | 0.231729  | 0.053698      |              |
| 2010-04-01         | 0.224017  | -0.010698 | 0.234715  | 0.234715  | 0.055091      |              |
| 2010-05-01         | -0.126839 | 0.063360  | -0.190200 | 0.190200  | 0.036176      |              |
| 2010-06-01         | 0.128215  | 0.028602  | 0.099613  | 0.099613  | 0.009923      |              |
| ...                | ...       | ...       | ...       | ...       | ...           | ...          |
| 2023-12-01         | -0.002006 | 0.004542  | -0.006548 | 0.006548  | 0.000043      |              |
| 2024-01-01         | -0.123391 | -0.009933 | -0.113458 | 0.113458  | 0.012873      |              |
| 2024-02-01         | -0.123407 | -0.031408 | -0.092000 | 0.092000  | 0.008464      |              |
| 2024-03-01         | 0.047704  | 0.037028  | 0.010677  | 0.010677  | 0.000114      |              |
| 2024-04-01         | NaN       | -0.026820 | NaN       | NaN       | NaN           |              |

171 rows × 6 columns



## Photographic\_&cinematographic\_instruments&\_suppl

```
In [73]: df = df_export_ANALYSIS.copy()
name = df.columns[29]
display(f"Component: {name}")

'Component: Photographic_&cinematographic_instruments_&_supplies'

In [74]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

'Timeseries of component in % form:'
```

| class Photographic_& cinematographic_instruments_& supplies |           |
|---|-----------|
| 1995-02-01  | 0.373116  |
| 1995-03-01  | -0.059774 |
| 1995-04-01  | -0.095037 |
| 1995-05-01  | -0.015054 |
| 1995-06-01  | 0.092431  |
| ...   | ...       |
| 2023-11-01  | -0.039328 |
| 2023-12-01  | -0.184898 |
| 2024-01-01  | 0.075788  |
| 2024-02-01  | -0.139791 |
| 2024-03-01  | 0.149853  |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -9.183476131408236
p-value: 2.2195621030071635e-15
Critical Value 1%: -3.4497304638968043
Critical Value 5%: -2.8700785273763487
Critical Value 10%: -2.571319005190311
is_stationary: True
```

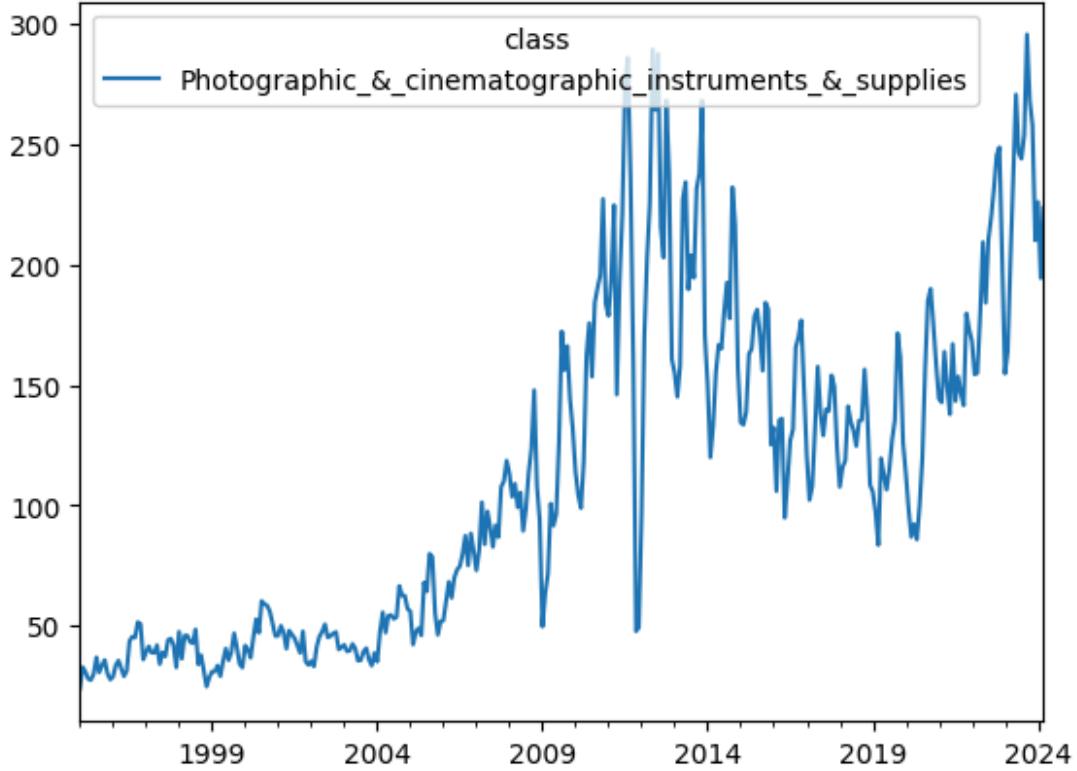
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.07129275264917881
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

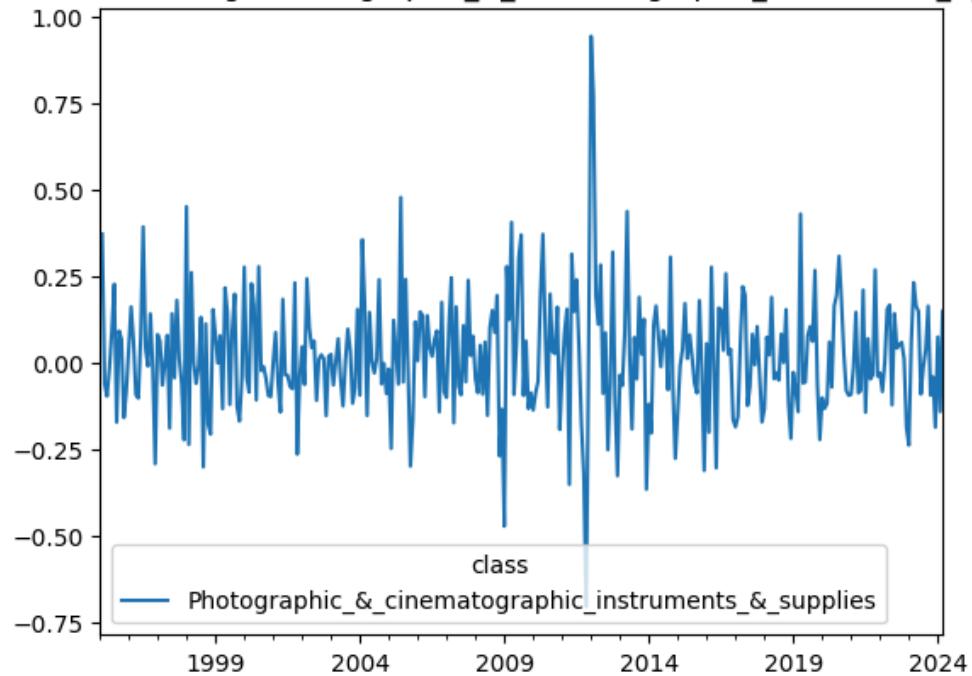
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

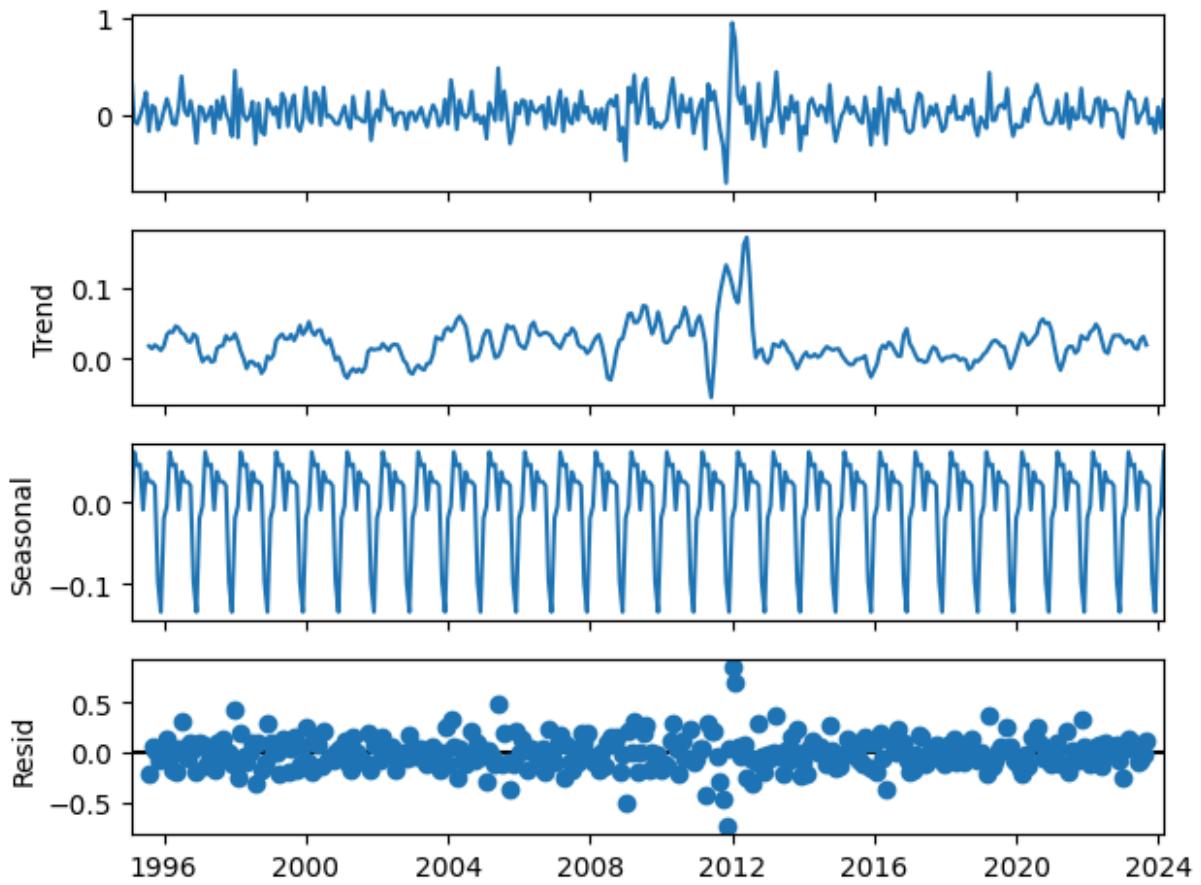
```
result = kpss(series, regression='c')
```

### Export Value (Photographic\_&\_cinematographic\_instruments\_&\_supplies)



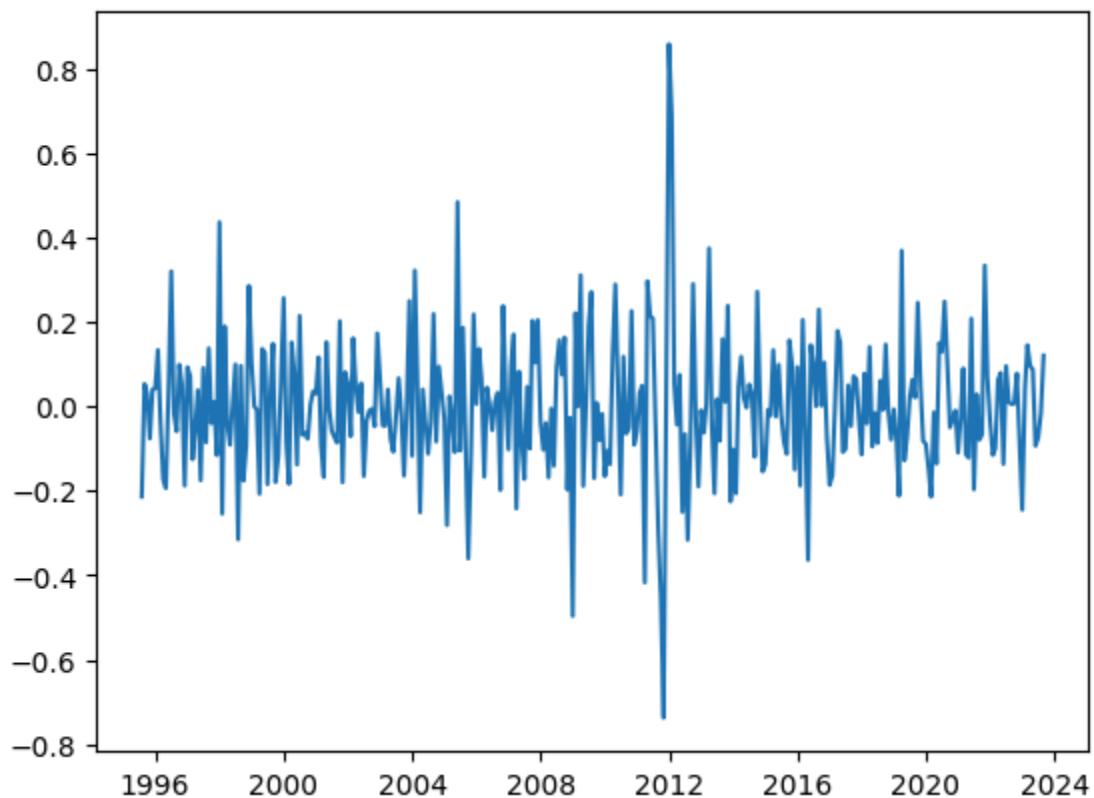
### Export Value % Change (Photographic\_&\_cinematographic\_instruments\_&\_supplies)

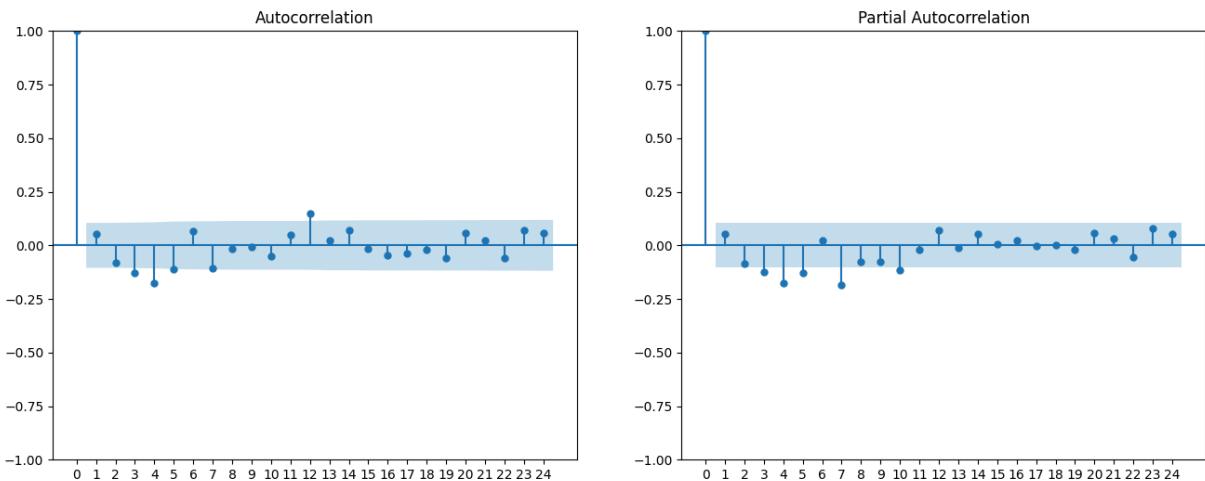




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

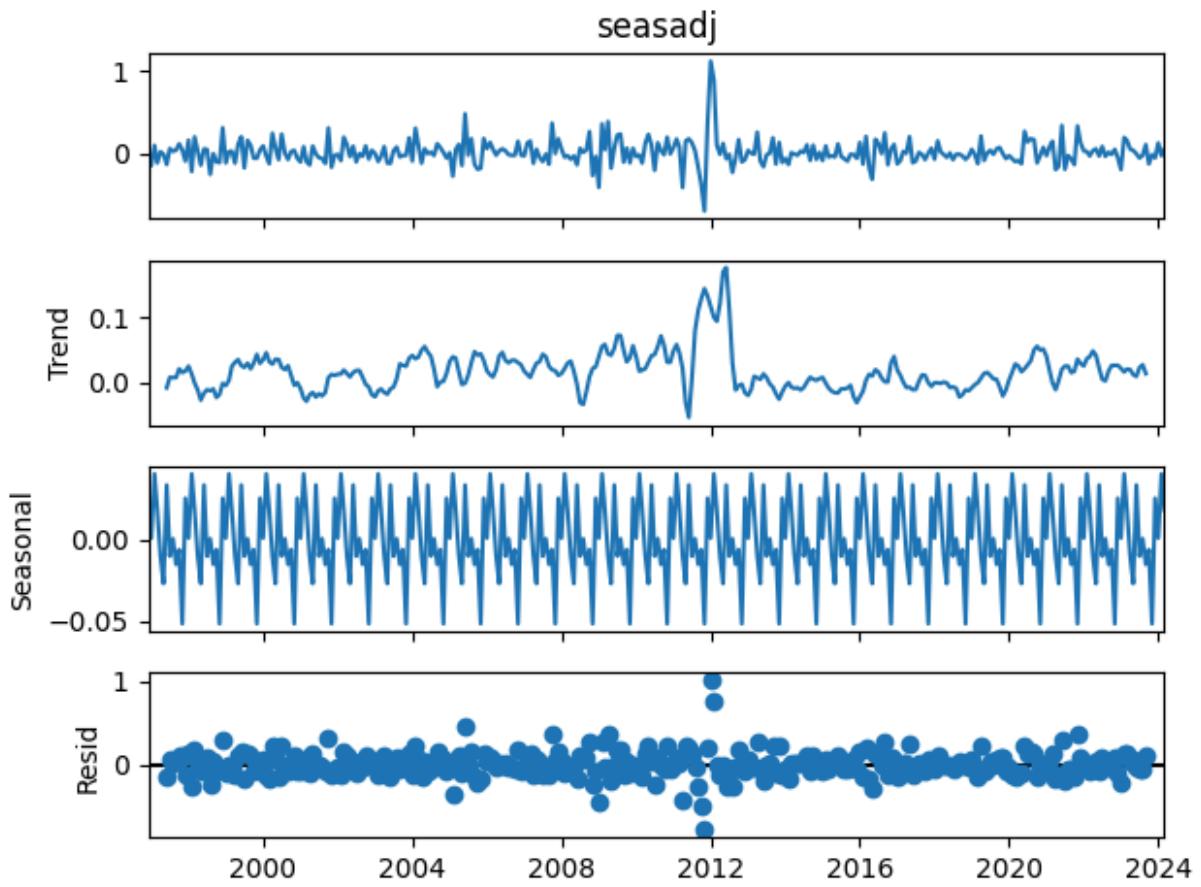
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -12.002703748305692
p-value: 3.3179957880425834e-22
Critical Value 1%: -3.450695263332383
Critical Value 5%: -2.87050218926466
Critical Value 10%: -2.5715449066453284
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.10020457897846362
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

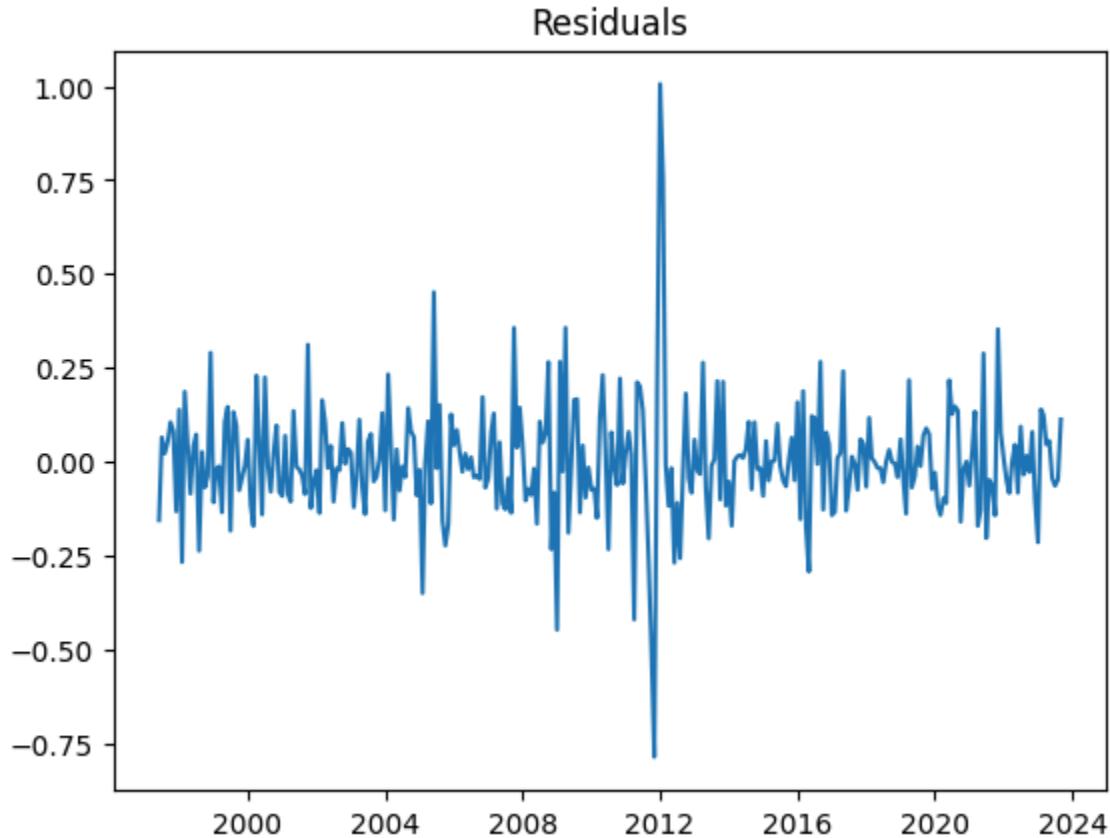
Decomposing the Series...

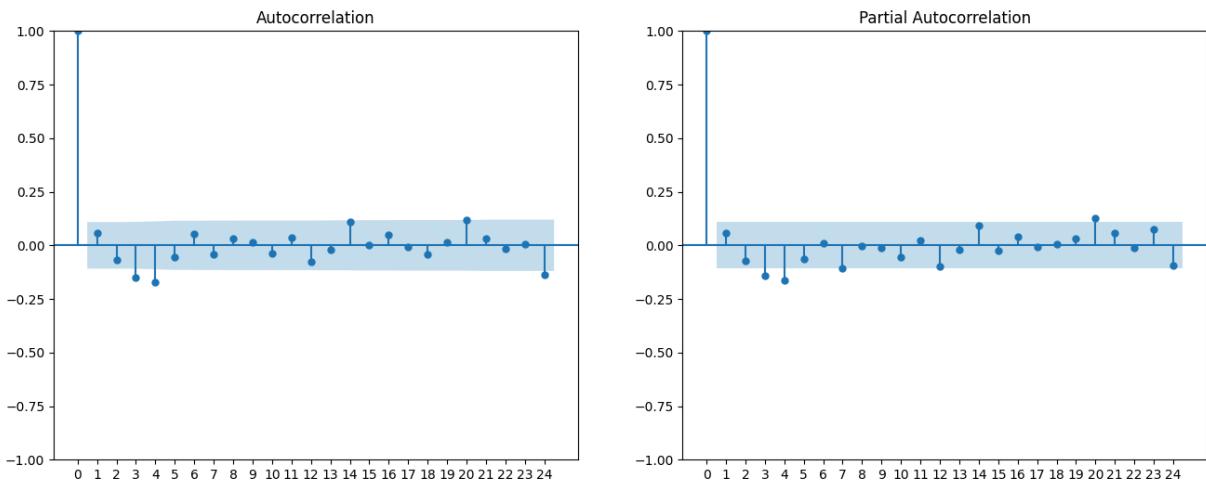
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...





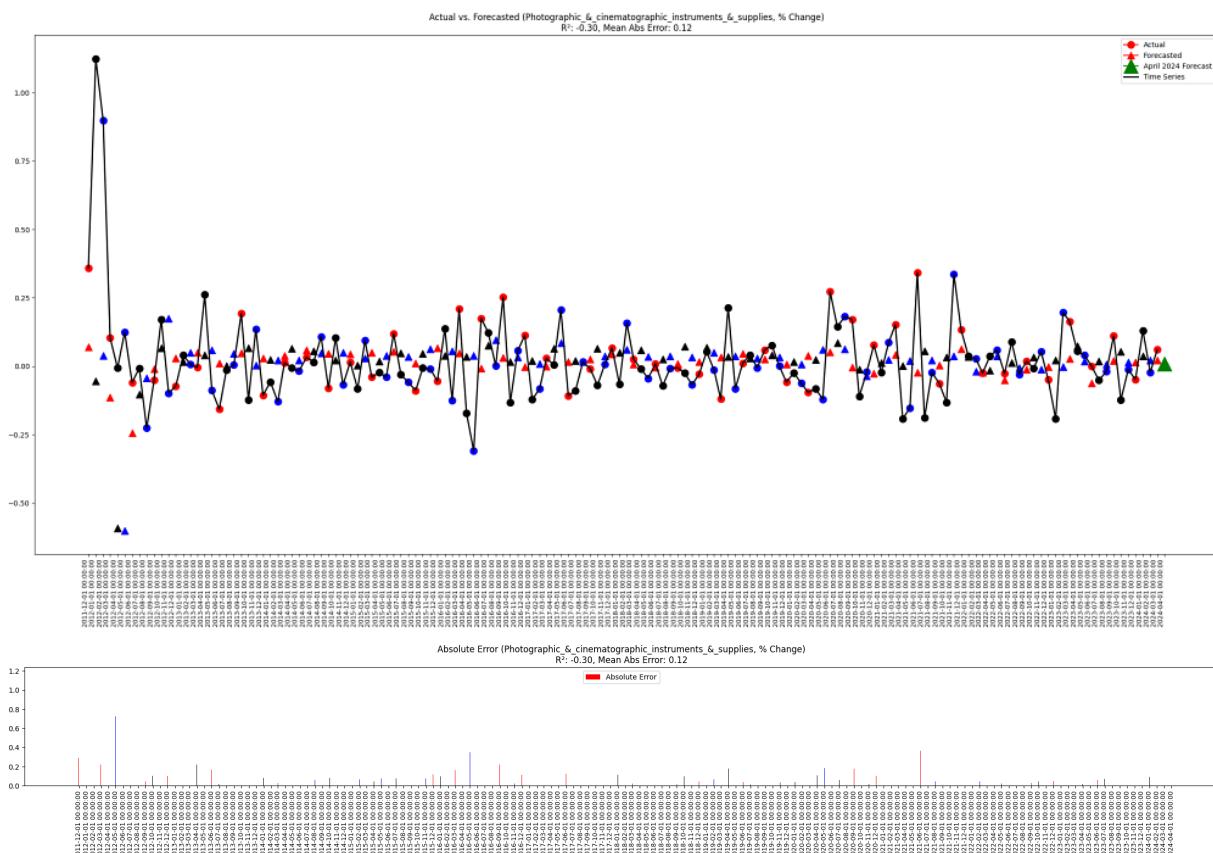
Time series assumptions are met.

'Data of Actual vs Forecasted values with error metrics:'

| Photographic_& cinematographic_instruments_& supplies | actual    | forecast  | error     | abs_error |
|---|-----------|-----------|-----------|-----------|
| 2011-12-01  | 0.359001  | 0.069880  | 0.289122  | 0.289122  |
| 2012-01-01  | 1.123780  | -0.054937 | 1.178717  | 1.178717  |
| 2012-02-01  | 0.897644  | 0.038666  | 0.858977  | 0.858977  |
| 2012-03-01  | 0.103778  | -0.114810 | 0.218588  | 0.218588  |
| 2012-04-01  | -0.006821 | -0.592559 | 0.585738  | 0.585738  |
| ...   |           |           |           |           |
| 2023-12-01  | -0.049701 | 0.014464  | -0.064165 | 0.064165  |
| 2024-01-01  | 0.130163  | 0.035688  | 0.094475  | 0.094475  |
| 2024-02-01  | -0.022415 | 0.022100  | -0.044515 | 0.044515  |
| 2024-03-01  | 0.060068  | 0.021142  | 0.038926  | 0.038926  |
| 2024-04-01  | NaN       | 0.008773  | NaN       | NaN       |

149 rows × 6 columns

'Total Mean Absolute Error (Per Forecast): 0.11566691315482246'



## Medicinal\_and\_surgical\_equipment\_and\_supplies

```
In [75]: df = df_export_ANALYSIS.copy()
name = df.columns[30]
display(f"Component: {name}")
```

'Component: Medicinal\_and\_surgical\_equipment\_and\_supplies'

```
In [76]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

class Medicinal\_and\_surgical\_equipment\_and\_supplies

|            |           |
|------------|-----------|
| 1995-02-01 | -0.065259 |
| 1995-03-01 | 0.124743  |
| 1995-04-01 | -0.217252 |
| 1995-05-01 | 0.287464  |
| 1995-06-01 | 0.009058  |
| ...        | ...       |
| 2023-11-01 | -0.021785 |
| 2023-12-01 | -0.007343 |
| 2024-01-01 | 0.109326  |
| 2024-02-01 | -0.051489 |
| 2024-03-01 | 0.061373  |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

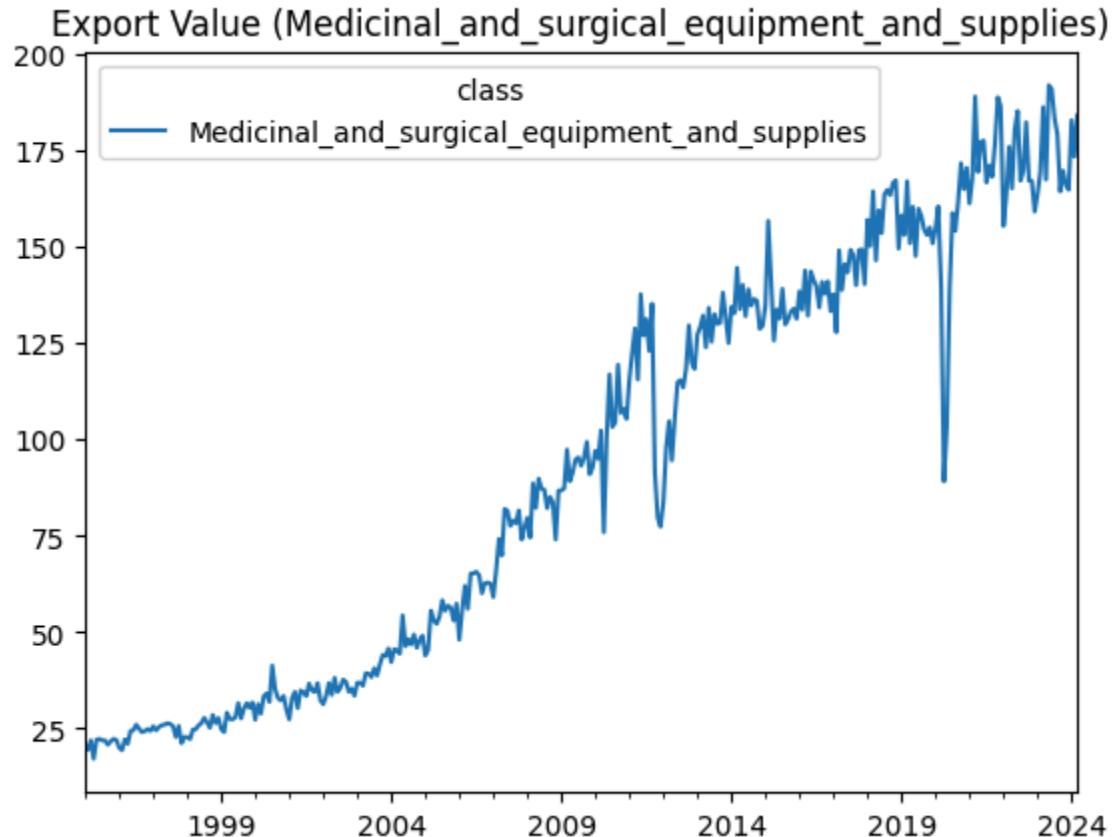
```
Running ADF Test for Stationarity...
ADF Statistic: -6.422414374457231
p-value: 1.778066905345242e-08
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.1650885778322806
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

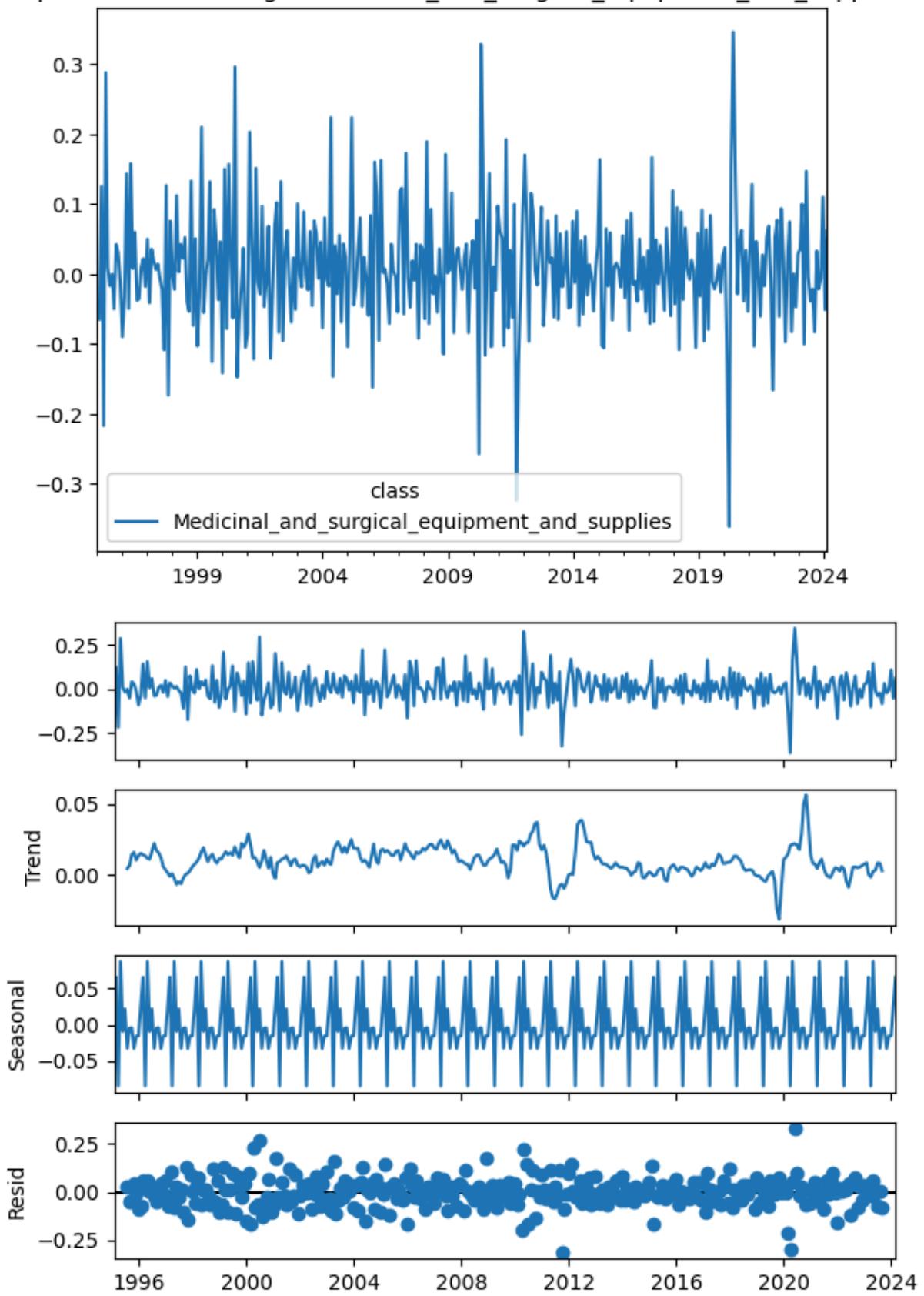
Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

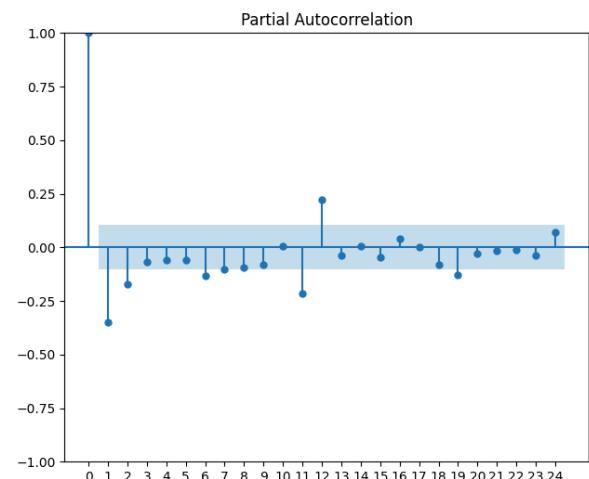
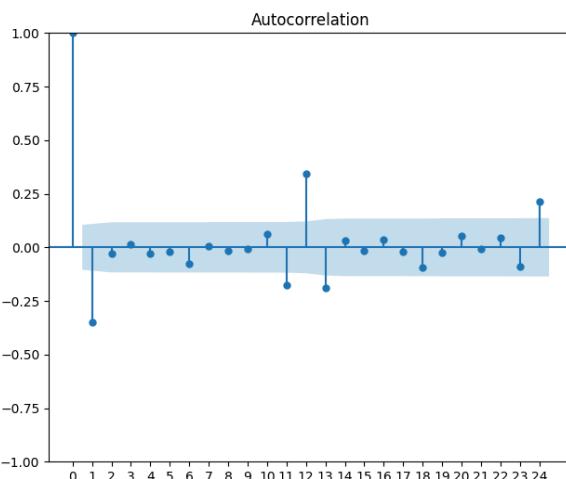
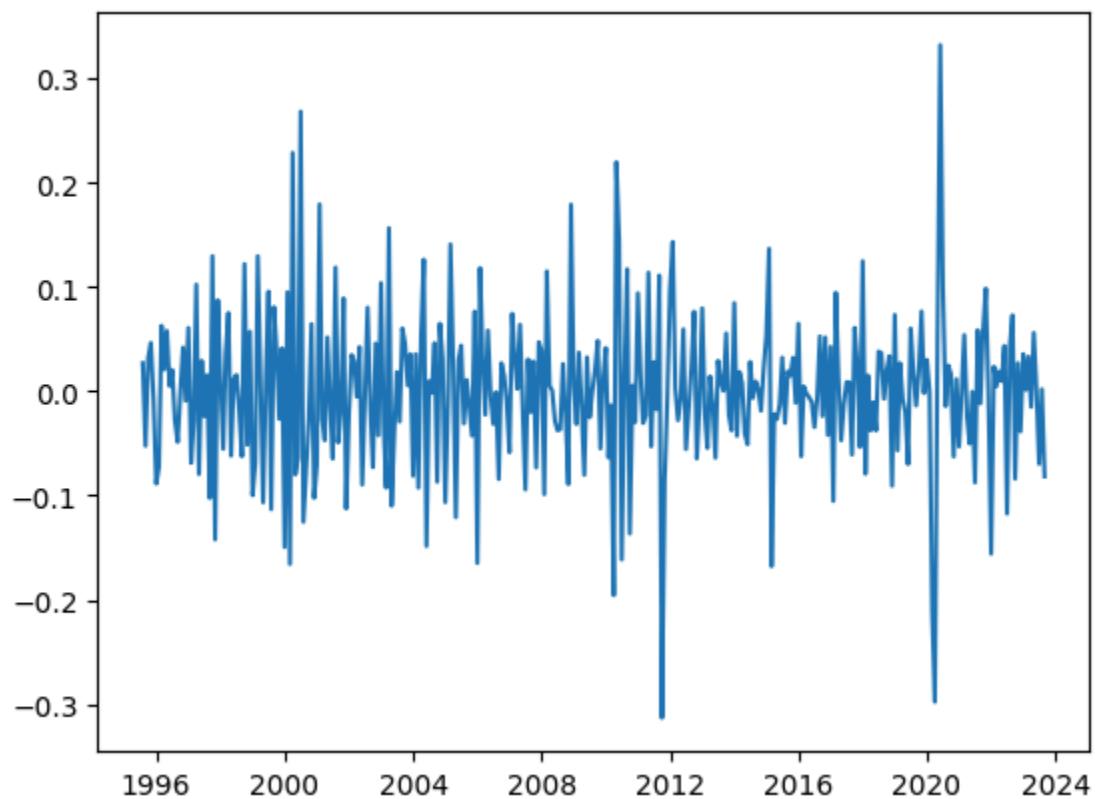


## Export Value % Change (Medicinal\_and\_surgical\_equipment\_and\_supplies)



Plotting Residuals for Homoscedasticity Check...

### Residuals



Time series assumptions are met.

True

'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

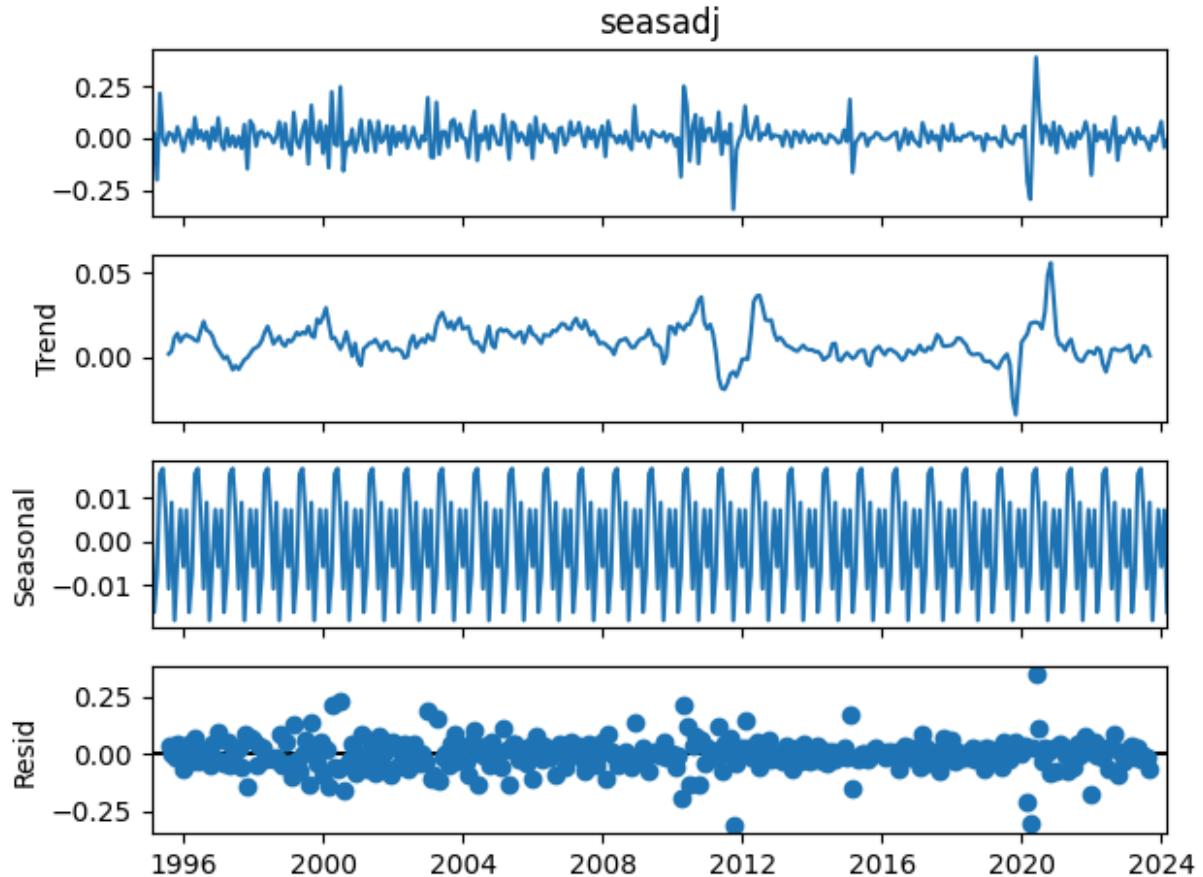
Running ADF Test for Stationarity...  
 ADF Statistic: -11.065529070163342  
 p-value: 4.692168020413442e-20  
 Critical Value 1%: -3.4494474563375737  
 Critical Value 5%: -2.8699542285903887  
 Critical Value 10%: -2.5712527305187987  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.1651244245198311  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

Decomposing the Series...

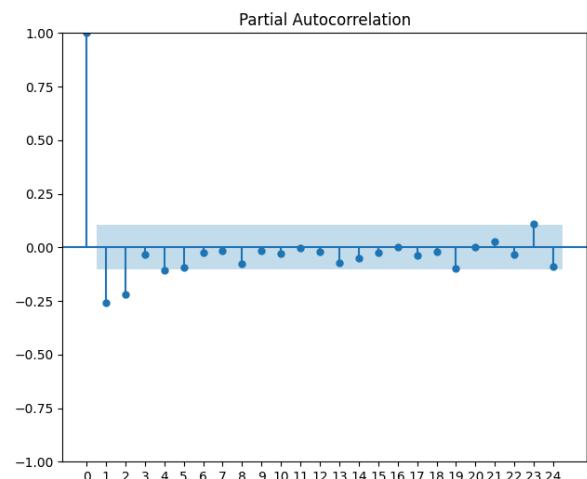
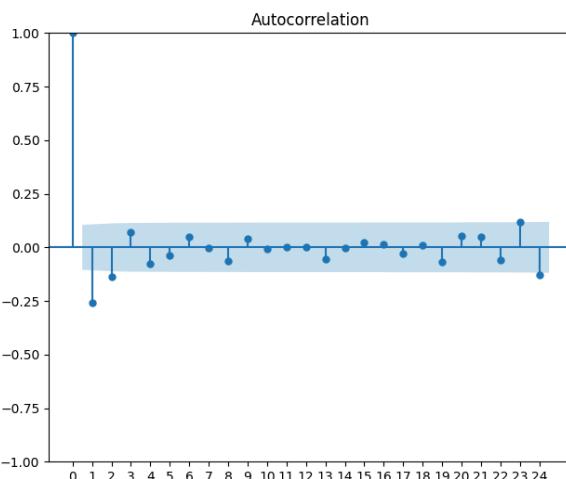
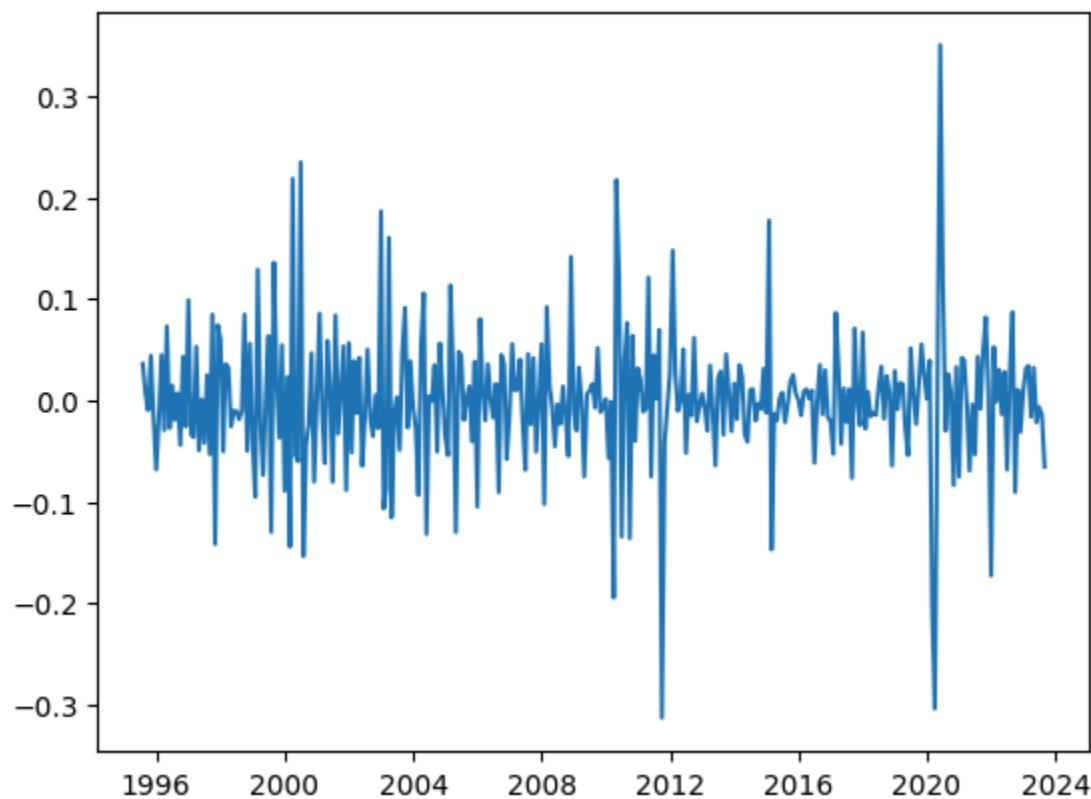
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals



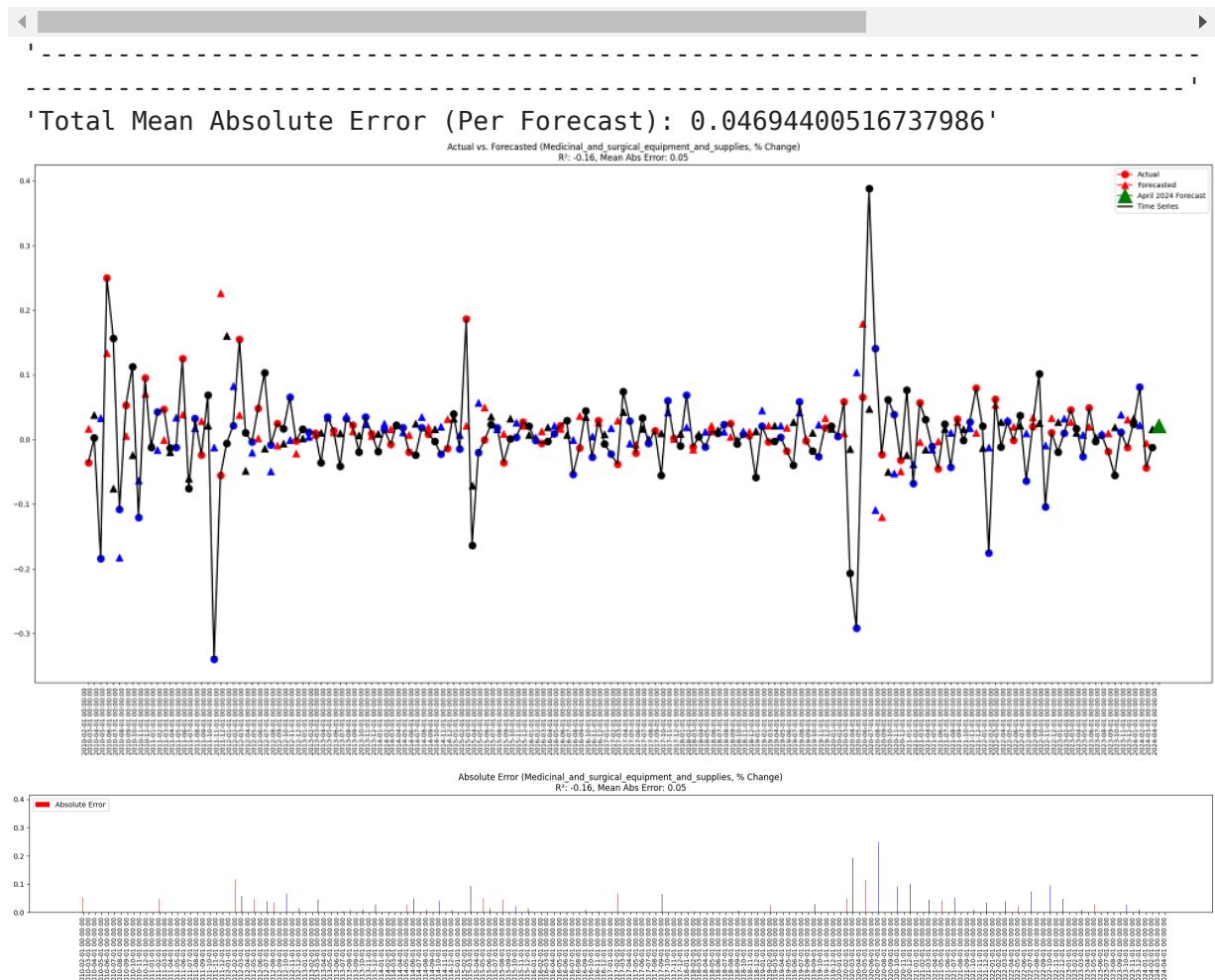
Time series assumptions are met.

'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Medicinal_and_surgical_equipment_and_supplies |            | actual    | forecast  | error     | abs_error | s   |
|---|------------|-----------|-----------|-----------|-----------|-----|
|   | 2010-02-01 | -0.035700 | 0.016194  | -0.051894 | 0.051894  |     |
|   | 2010-03-01 | 0.002127  | 0.037635  | -0.035508 | 0.035508  |     |
|   | 2010-04-01 | -0.184225 | 0.032988  | -0.217213 | 0.217213  |     |
|   | 2010-05-01 | 0.249380  | 0.133704  | 0.115675  | 0.115675  |     |
|   | 2010-06-01 | 0.156345  | -0.075854 | 0.232199  | 0.232199  |     |
|   | ...        | ...       | ...       | ...       | ...       | ... |
|   | 2023-12-01 | 0.023525  | 0.032437  | -0.008912 | 0.008912  |     |
|   | 2024-01-01 | 0.080564  | 0.021761  | 0.058803  | 0.058803  |     |
|   | 2024-02-01 | -0.043753 | -0.005171 | -0.038582 | 0.038582  |     |
|   | 2024-03-01 | -0.012377 | 0.015508  | -0.027885 | 0.027885  |     |
|   | 2024-04-01 | NaN       | 0.021001  | NaN       | NaN       |     |

171 rows × 6 columns



## Toiletries\_and\_cosmetics

```
In [77]: df = df_export_ANALYSIS.copy()
name = df.columns[31]
display(f"Component: {name}")

'Component: Toiletries_and_cosmetics'

In [78]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/{name}.pkl")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Toiletries_and_cosmetics |
|------------|--------------------------|
| 1995-02-01 | 0.067757                 |
| 1995-03-01 | 0.571116                 |
| 1995-04-01 | -0.245822                |
| 1995-05-01 | 0.076639                 |
| 1995-06-01 | 0.055746                 |
| ...        | ...                      |
| 2023-11-01 | 0.025501                 |
| 2023-12-01 | -0.011259                |
| 2024-01-01 | 0.017130                 |
| 2024-02-01 | 0.036279                 |
| 2024-03-01 | 0.010897                 |

350 rows × 1 columns

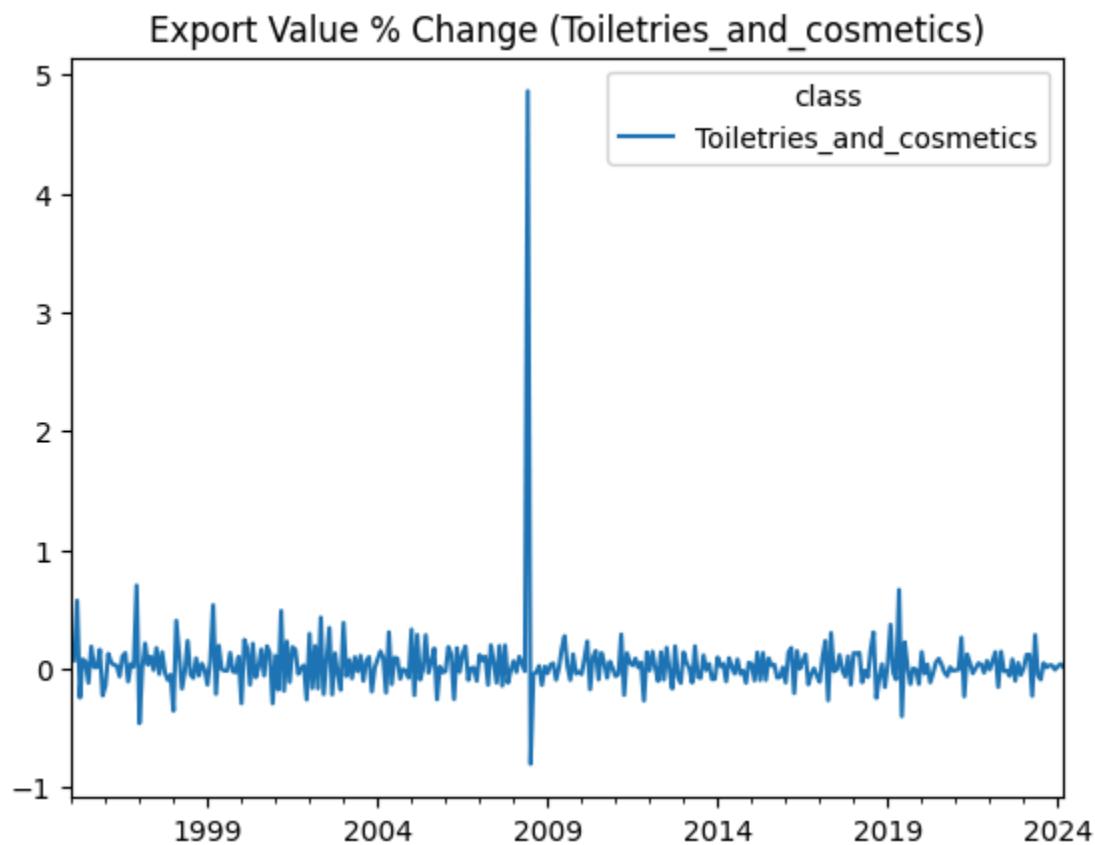
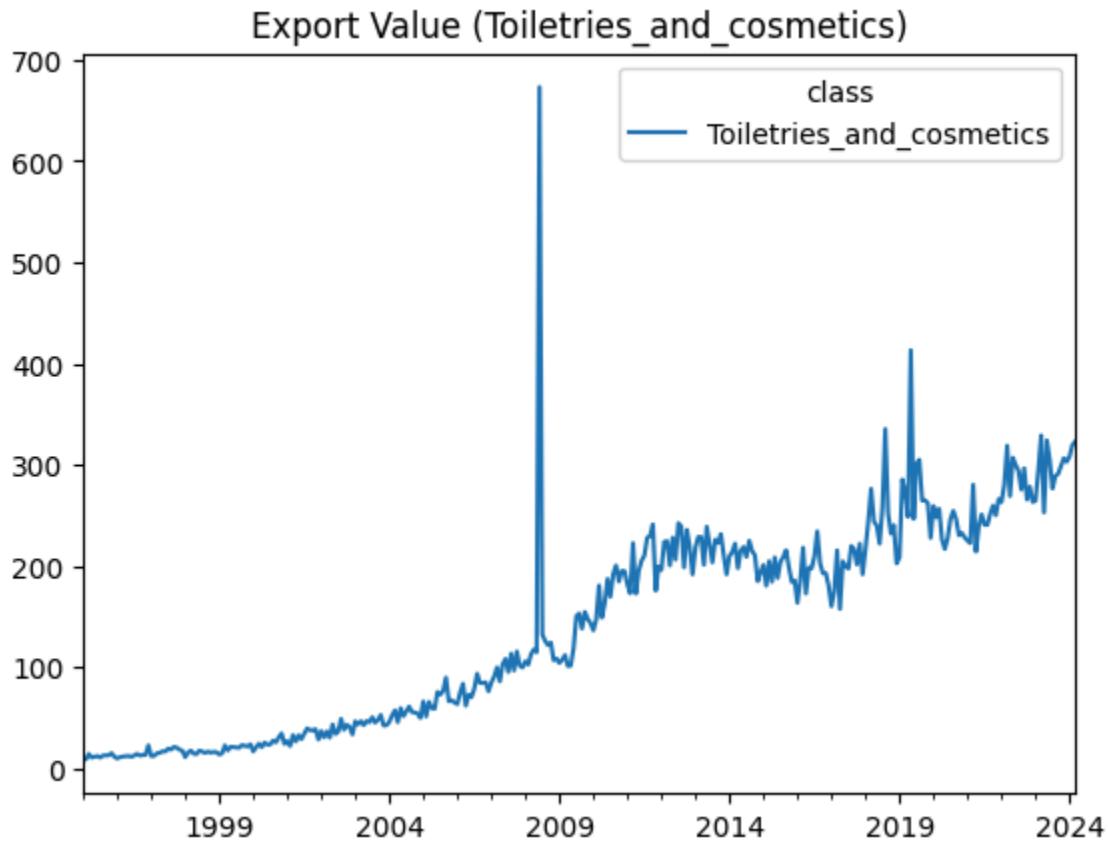
```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -24.040223456792255
p-value: 0.0
Critical Value 1%: -3.449226932880019
Critical Value 5%: -2.869857365438656
Critical Value 10%: -2.571201085130664
is_stationary: True
```

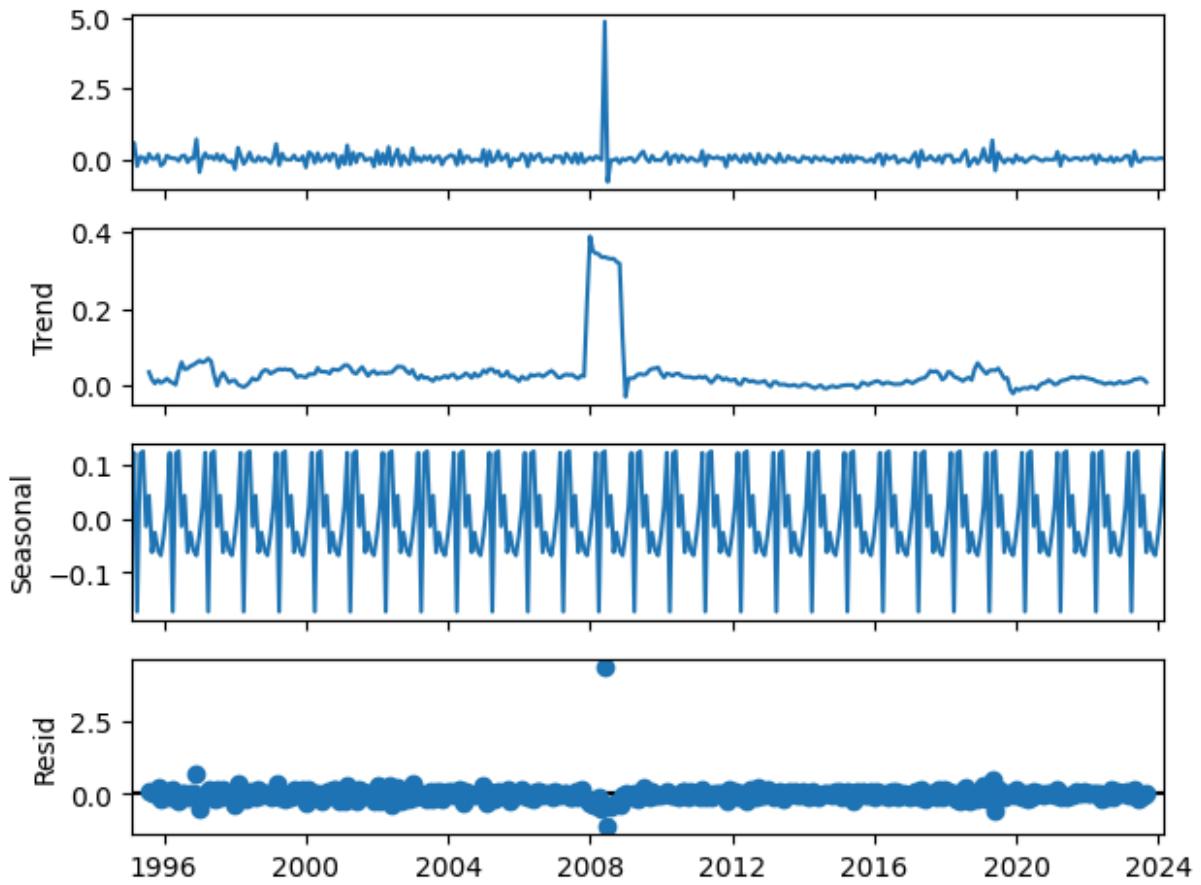
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.14248430654400962
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

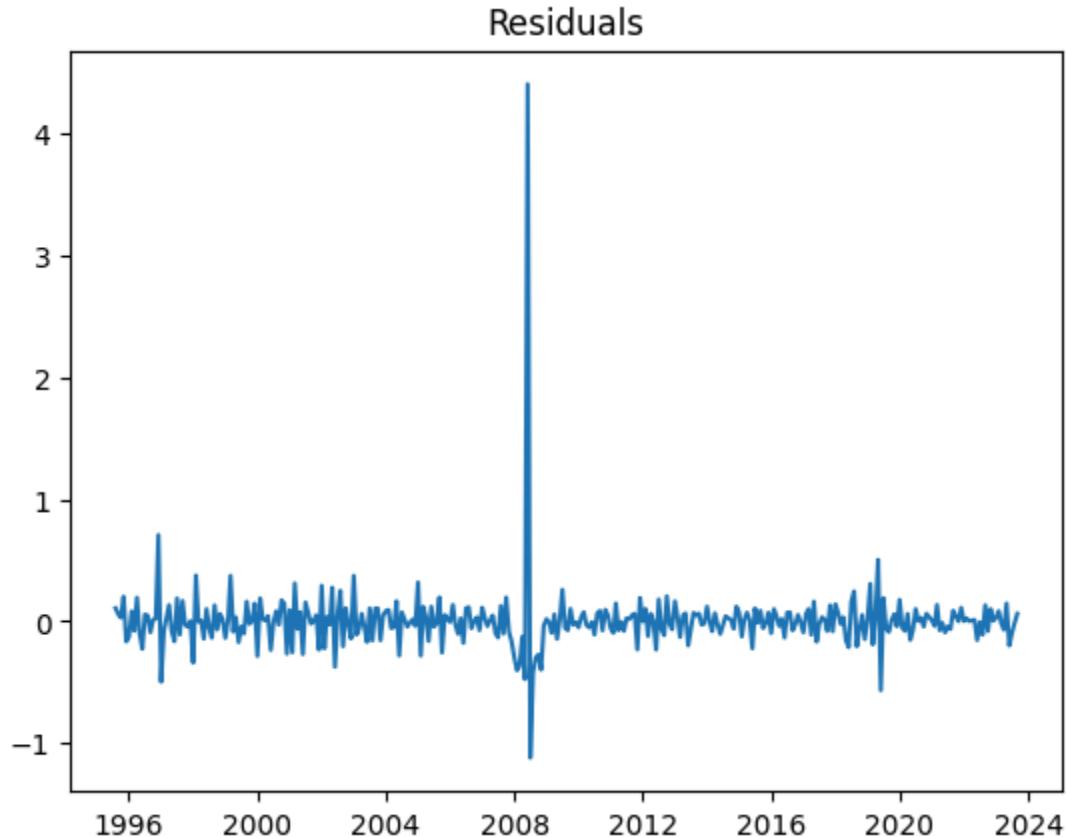
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

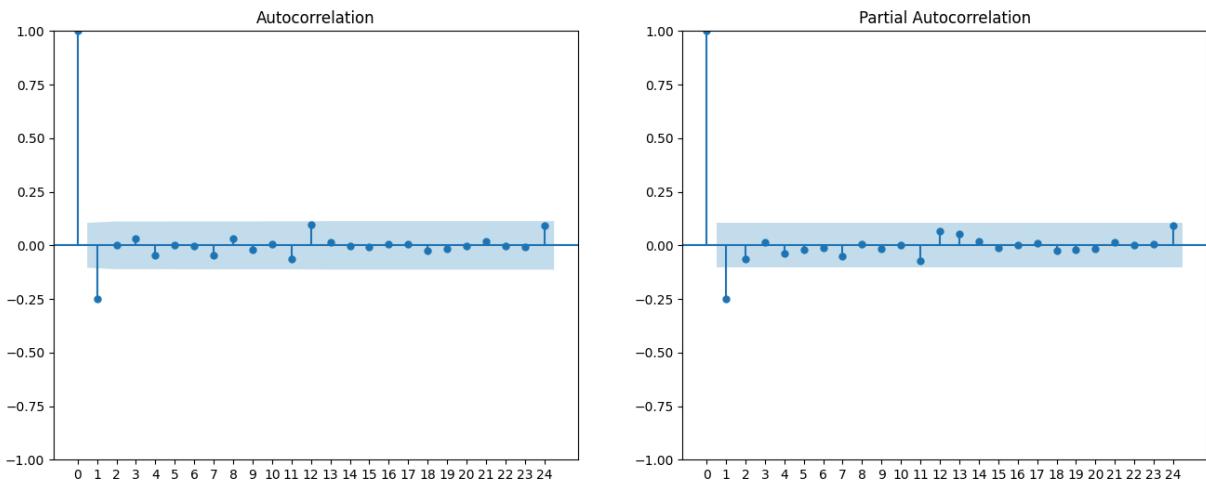
```
result = kpss(series, regression='c')
```





Plotting Residuals for Homoscedasticity Check...





Time series assumptions are met.

True

'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

Running ADF Test for Stationarity...

ADF Statistic: -23.888194878927028

p-value: 0.0

Critical Value 1%: -3.449226932880019

Critical Value 5%: -2.869857365438656

Critical Value 10%: -2.571201085130664

is\_stationary: True

Running KPSS Test for Stationarity...

KPSS Statistic: 0.14286465702875714

p-value: 0.1

Critical Value 10%: 0.347

Critical Value 5%: 0.463

Critical Value 2.5%: 0.574

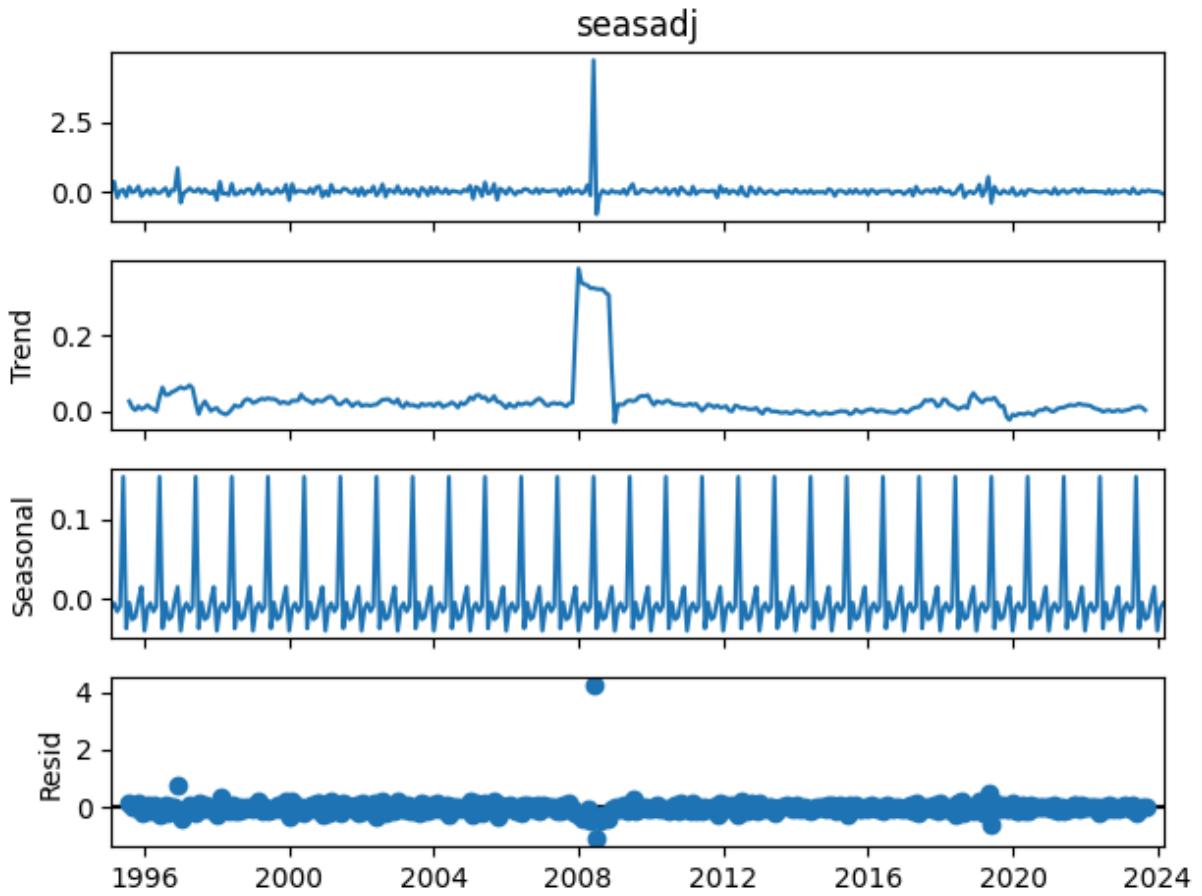
Critical Value 1%: 0.739

is\_stationary: True

Decomposing the Series...

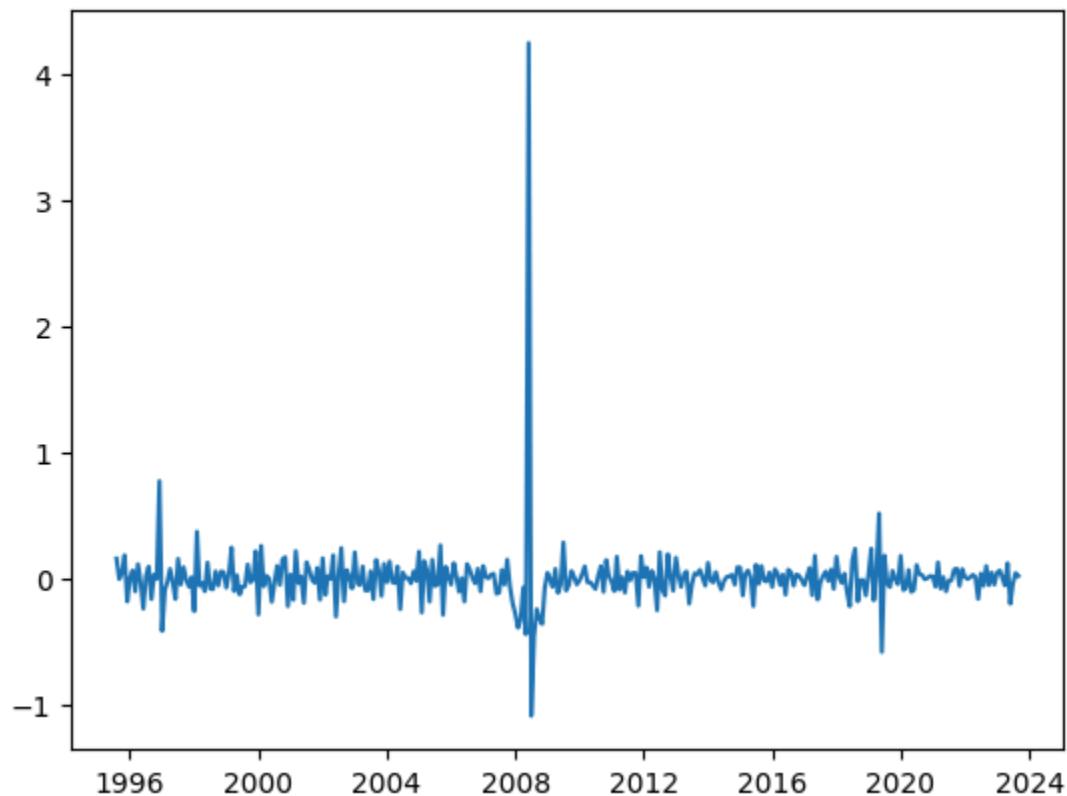
/home/wheelfredie/scripts/BoT\_Exports/helper.py:61: InterpolationWarning: The test statistic is outside of the range of p-values available in the look-up table. The actual p-value is greater than the p-value returned.

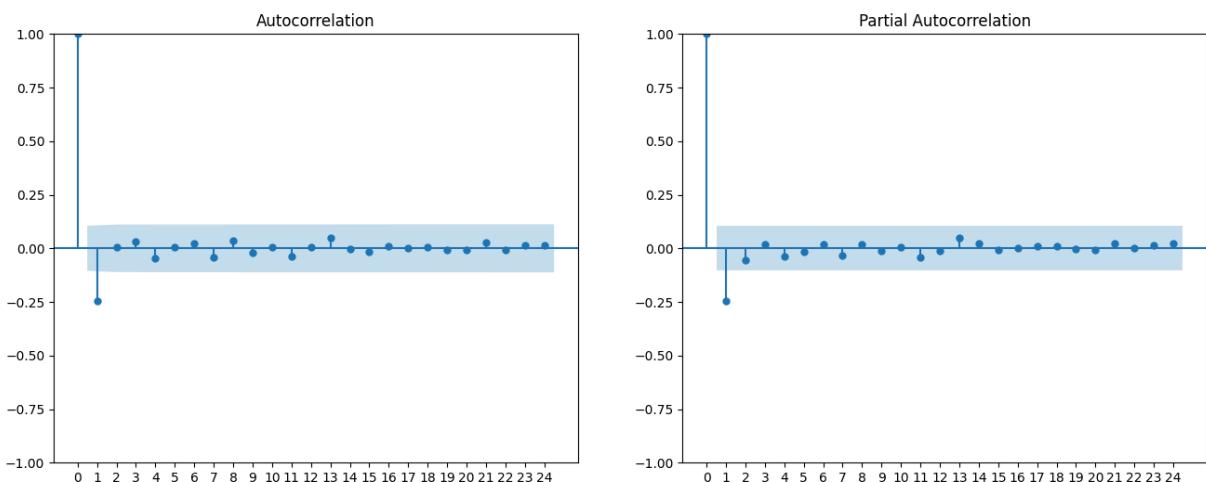
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

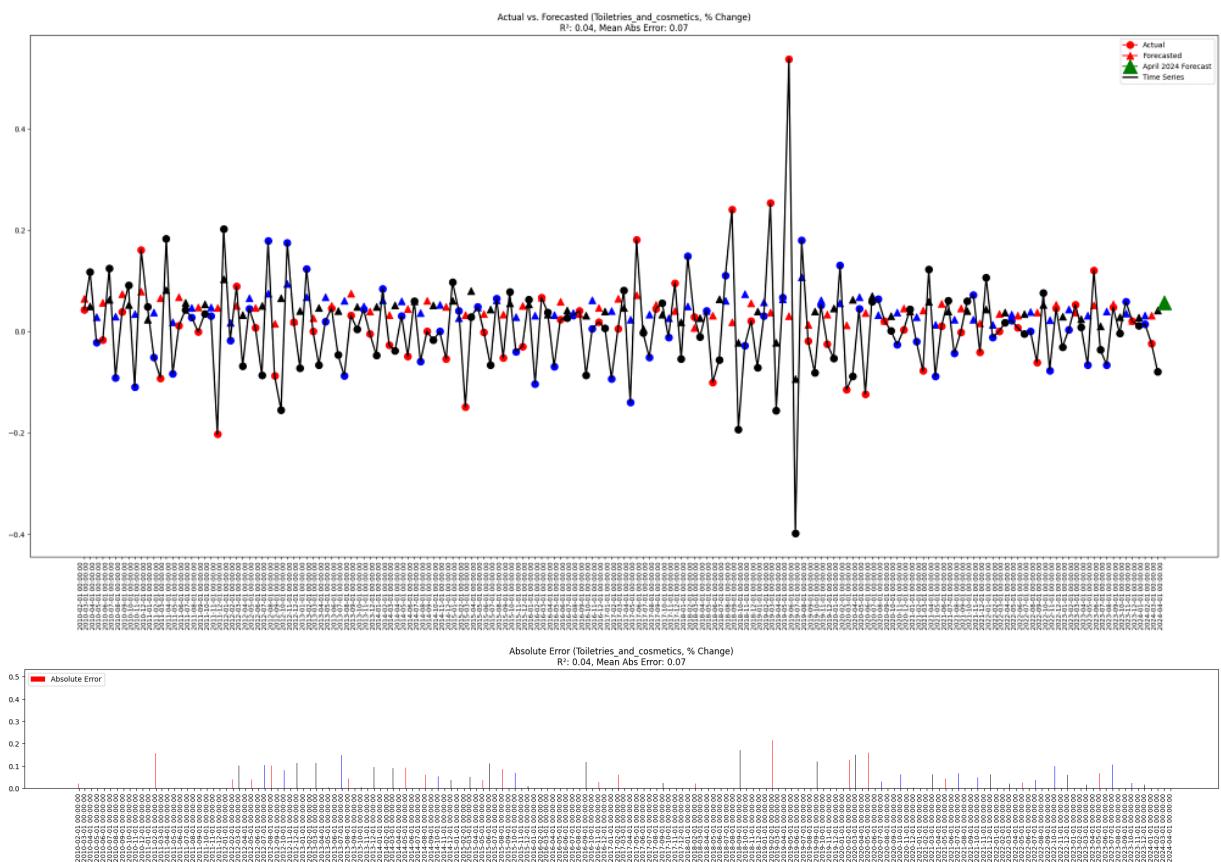
'Data of Actual vs Forecasted values with error metrics:'

| Toiletries_and_cosmetics | actual    | forecast | error     | abs_error | squared_error | abs_perc |
|--------------------------|-----------|----------|-----------|-----------|---------------|----------|
| 2010-02-01               | 0.042508  | 0.064736 | -0.022228 | 0.022228  | 0.000494      |          |
| 2010-03-01               | 0.117423  | 0.050245 | 0.067178  | 0.067178  | 0.004513      |          |
| 2010-04-01               | -0.021967 | 0.028634 | -0.050602 | 0.050602  | 0.002561      |          |
| 2010-05-01               | -0.016491 | 0.057271 | -0.073762 | 0.073762  | 0.005441      |          |
| 2010-06-01               | 0.124996  | 0.062635 | 0.062361  | 0.062361  | 0.003889      |          |
| ...                      | ...       | ...      | ...       | ...       | ...           | ...      |
| 2023-12-01               | 0.011681  | 0.027950 | -0.016269 | 0.016269  | 0.000265      |          |
| 2024-01-01               | 0.014604  | 0.031651 | -0.017046 | 0.017046  | 0.000291      |          |
| 2024-02-01               | -0.023743 | 0.031810 | -0.055553 | 0.055553  | 0.003086      |          |
| 2024-03-01               | -0.078751 | 0.041393 | -0.120144 | 0.120144  | 0.014435      |          |
| 2024-04-01               | NaN       | 0.057378 | NaN       | NaN       | NaN           |          |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.06687355062851026'



## Furniture\_and\_parts

```
In [79]: df = df_export_ANALYSIS.copy()
name = df.columns[32]
display(f"Component: {name}")

'Component: Furniture_and_parts'
```

```
In [80]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

| class Furniture_and_parts |           |
|---------------------------|-----------|
| 1995-02-01                | -0.069798 |
| 1995-03-01                | 0.316330  |
| 1995-04-01                | -0.276418 |
| 1995-05-01                | 0.267410  |
| 1995-06-01                | 0.041534  |
| ...                       | ...       |
| 2023-11-01                | 0.073885  |
| 2023-12-01                | -0.050655 |
| 2024-01-01                | -0.125461 |
| 2024-02-01                | 0.068776  |
| 2024-03-01                | 0.068851  |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

```
Running ADF Test for Stationarity...
ADF Statistic: -6.212313976989752
p-value: 5.4699252243022124e-08
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

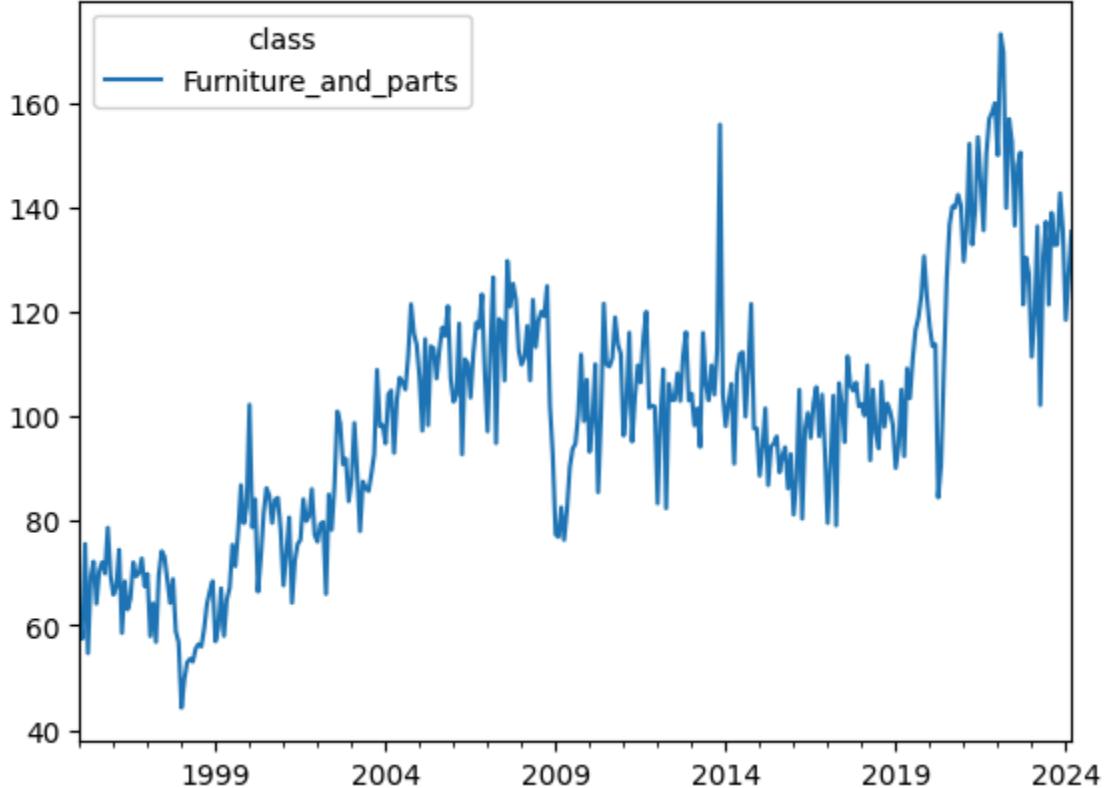
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.01954624829859106
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

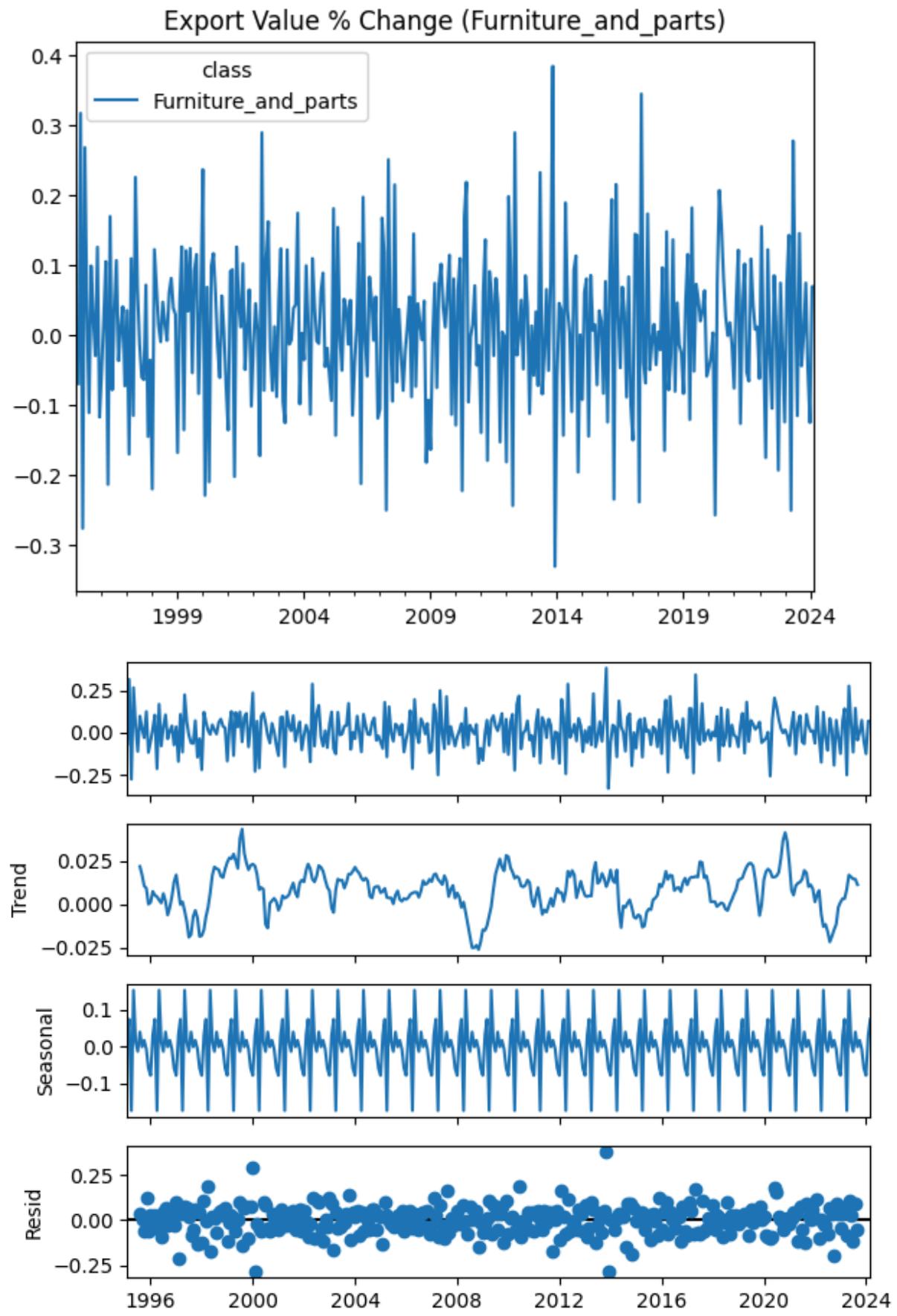
Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

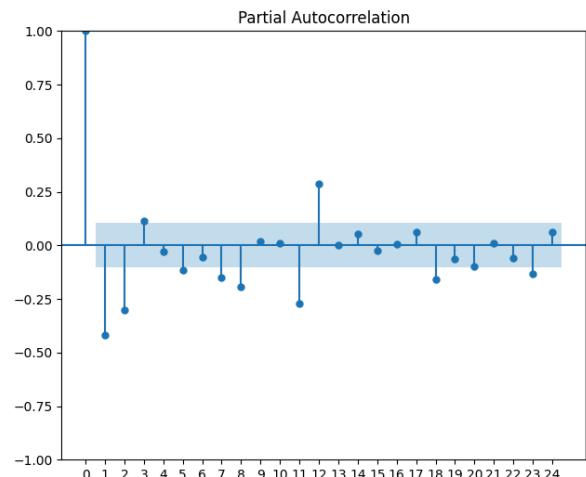
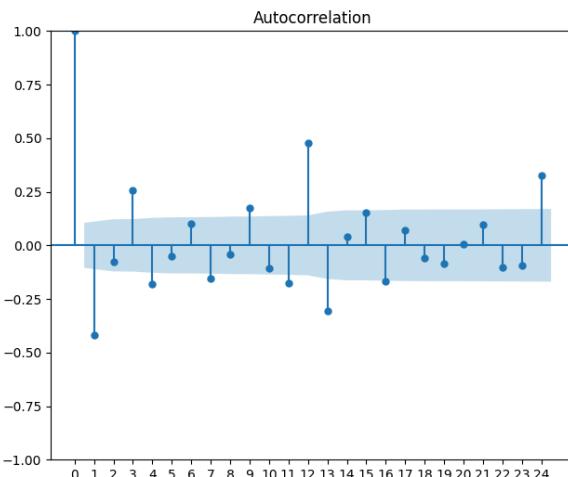
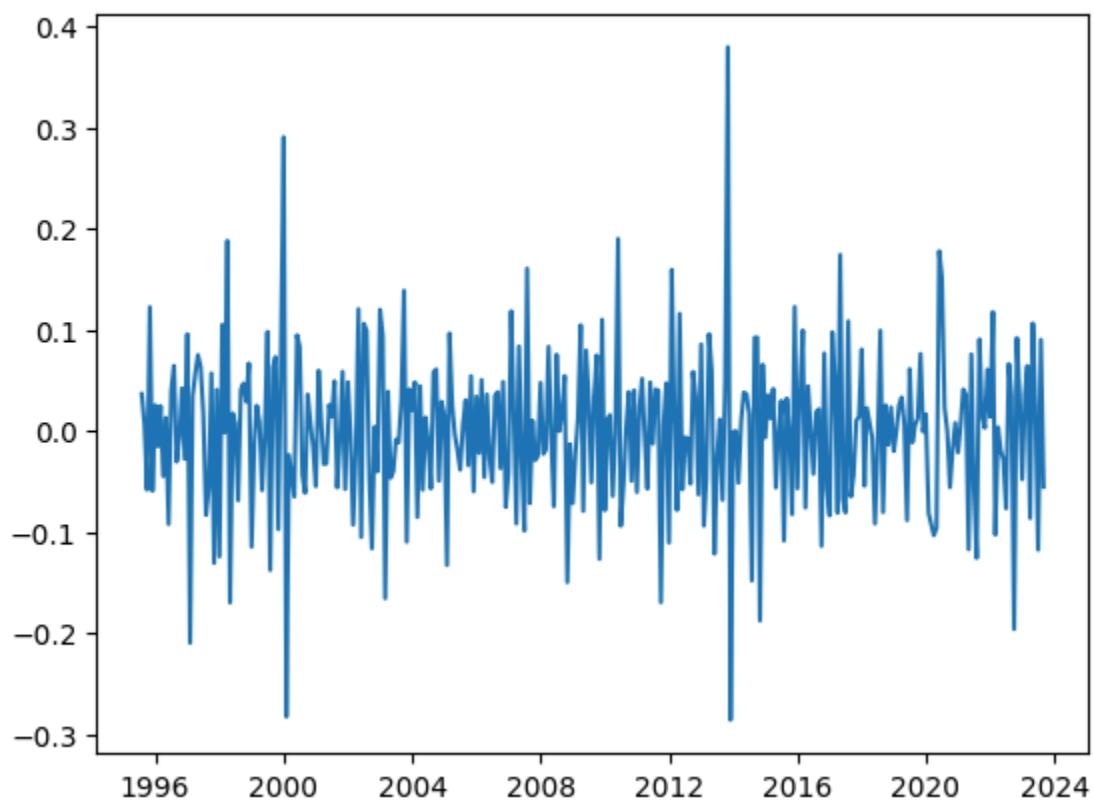
Export Value (Furniture\_and\_parts)





Plotting Residuals for Homoscedasticity Check...

### Residuals



Time series assumptions are met.

True

'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

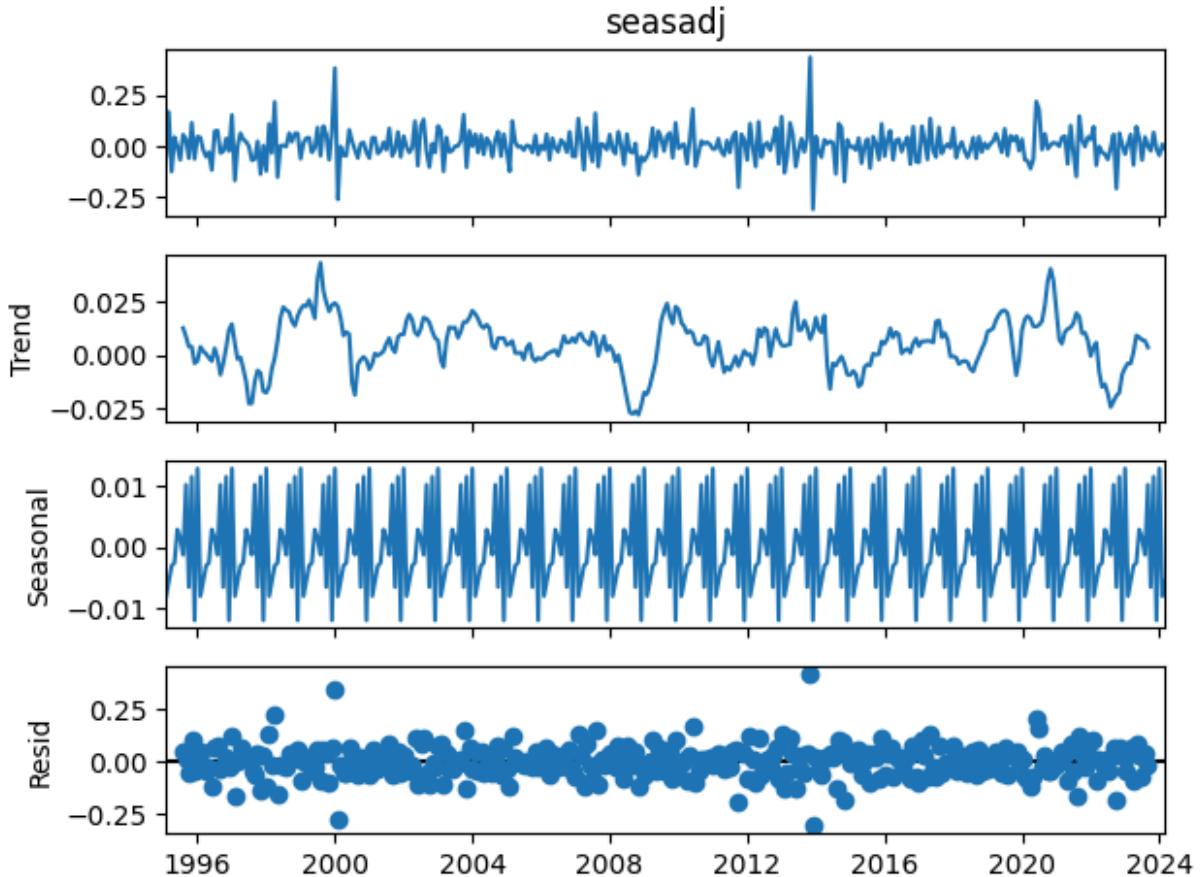
Running ADF Test for Stationarity...  
 ADF Statistic: -4.956342948092415  
 p-value: 2.705908239158204e-05  
 Critical Value 1%: -3.450141065277327  
 Critical Value 5%: -2.870258846235788  
 Critical Value 10%: -2.571415151457764  
 is\_stationary: True

Running KPSS Test for Stationarity...  
 KPSS Statistic: 0.025334961231282756  
 p-value: 0.1  
 Critical Value 10%: 0.347  
 Critical Value 5%: 0.463  
 Critical Value 2.5%: 0.574  
 Critical Value 1%: 0.739  
 is\_stationary: True

Decomposing the Series...

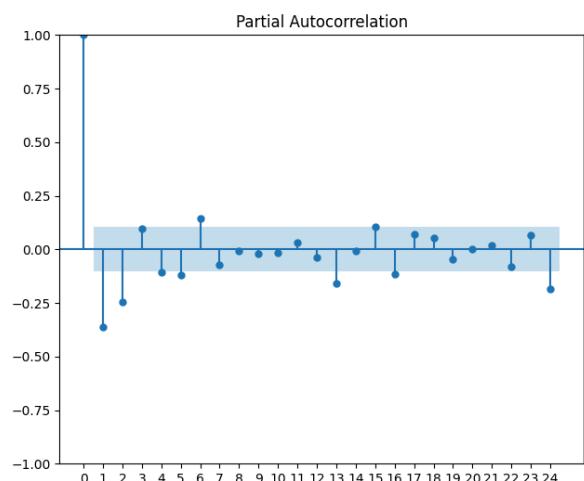
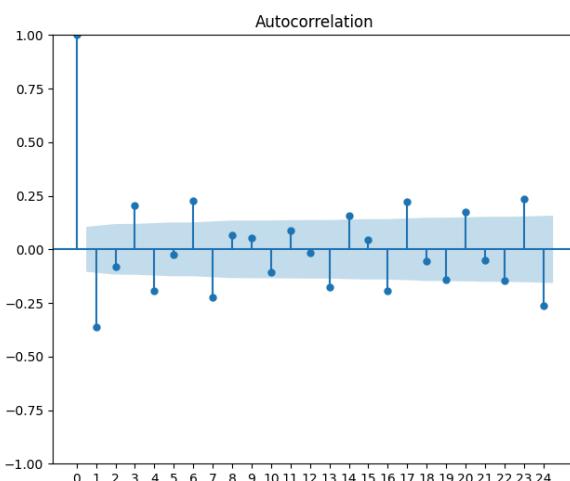
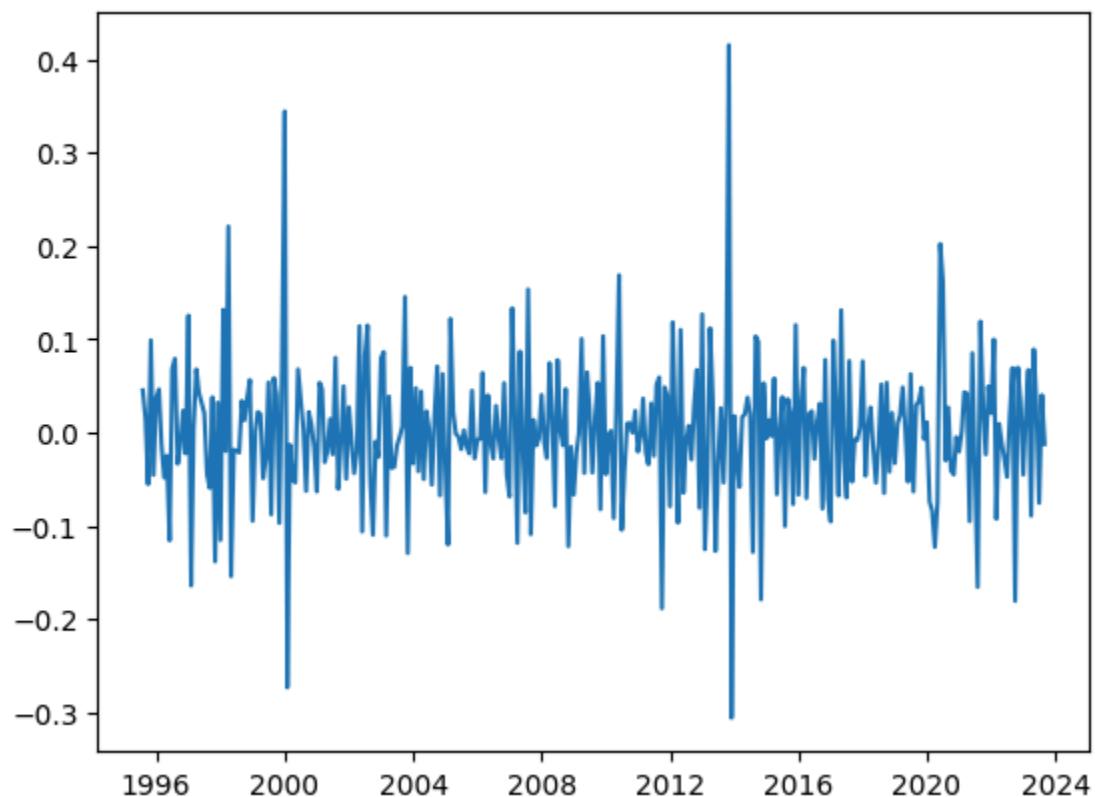
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals

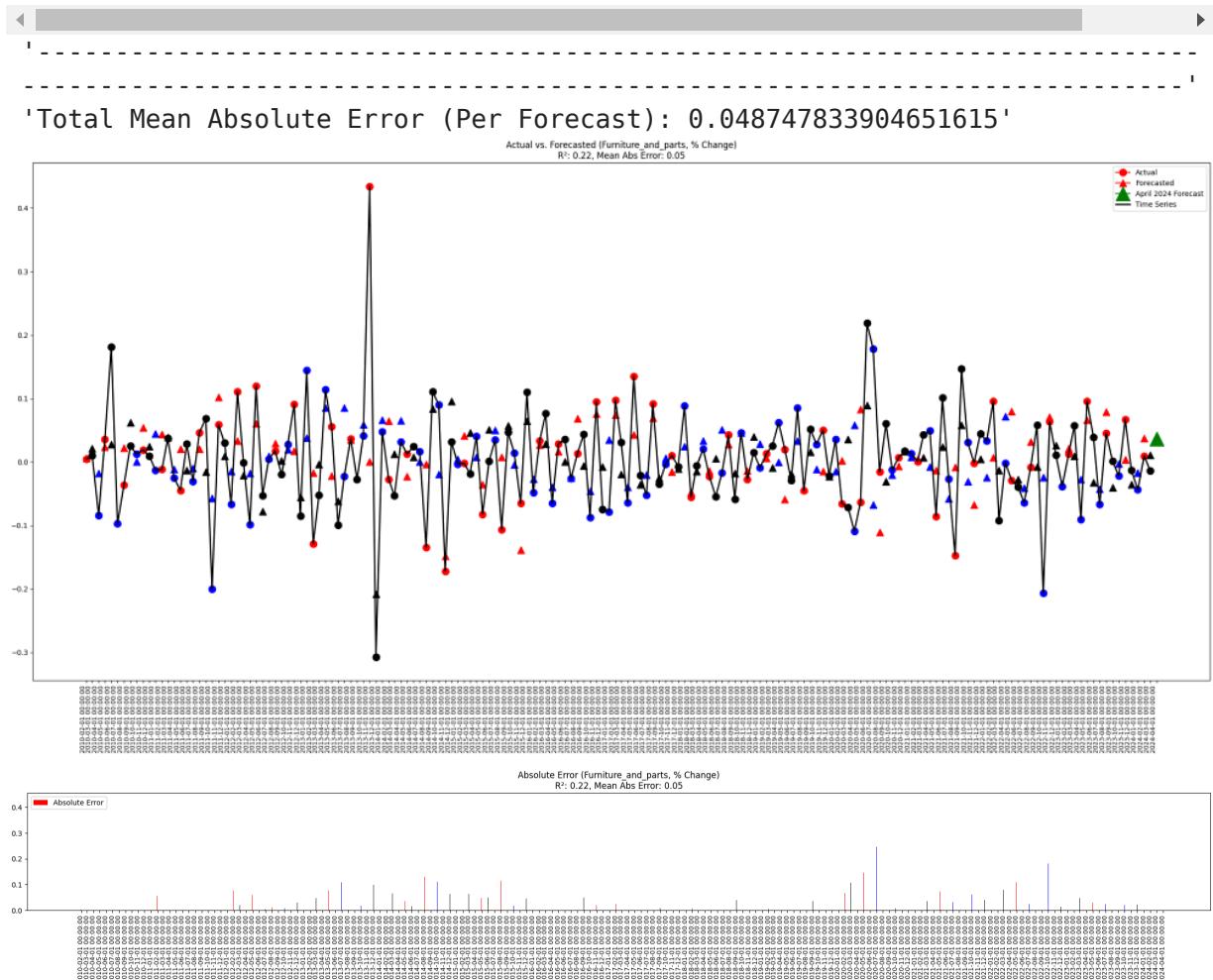


Time series assumptions are met.

'-----'  
-----'  
'Data of Actual vs Forecasted values with error metrics:'

| Furniture_and_parts | actual    | forecast  | error     | abs_error | squared_error | abs_percenta |
|---------------------|-----------|-----------|-----------|-----------|---------------|--------------|
| 2010-02-01          | 0.004458  | 0.006113  | -0.001654 | 0.001654  | 0.000003      |              |
| 2010-03-01          | 0.009731  | 0.021423  | -0.011692 | 0.011692  | 0.000137      |              |
| 2010-04-01          | -0.084678 | -0.018039 | -0.066639 | 0.066639  | 0.004441      |              |
| 2010-05-01          | 0.035323  | 0.023379  | 0.011945  | 0.011945  | 0.000143      |              |
| 2010-06-01          | 0.181589  | 0.027996  | 0.153593  | 0.153593  | 0.023591      |              |
| ...                 | ...       | ...       | ...       | ...       | ...           | ...          |
| 2023-12-01          | -0.013130 | -0.035898 | 0.022768  | 0.022768  | 0.000518      |              |
| 2024-01-01          | -0.043846 | -0.017129 | -0.026717 | 0.026717  | 0.000714      |              |
| 2024-02-01          | 0.009530  | 0.037567  | -0.028037 | 0.028037  | 0.000786      |              |
| 2024-03-01          | -0.013875 | 0.010647  | -0.024522 | 0.024522  | 0.000601      |              |
| 2024-04-01          | NaN       | 0.036714  | NaN       | NaN       | NaN           | NaN          |

171 rows × 6 columns



## Other\_Manufacturing\_products

```
In [81]: df = df_export_ANALYSIS.copy()
name = df.columns[33]
display(f"Component: {name}")

'Component: Other_Manufacturing_products'
```

```
In [82]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/{name}.pkl")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Other_Manufacturing_products |
|------------|------------------------------|
| 1995-02-01 | 0.007269                     |
| 1995-03-01 | 0.336890                     |
| 1995-04-01 | -0.282515                    |
| 1995-05-01 | 0.292957                     |
| 1995-06-01 | 0.140076                     |
| ...        | ...                          |
| 2023-11-01 | -0.004233                    |
| 2023-12-01 | -0.059999                    |
| 2024-01-01 | -0.041570                    |
| 2024-02-01 | 0.075350                     |
| 2024-03-01 | 0.097300                     |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -5.907699394769142
p-value: 2.684227855477888e-07
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

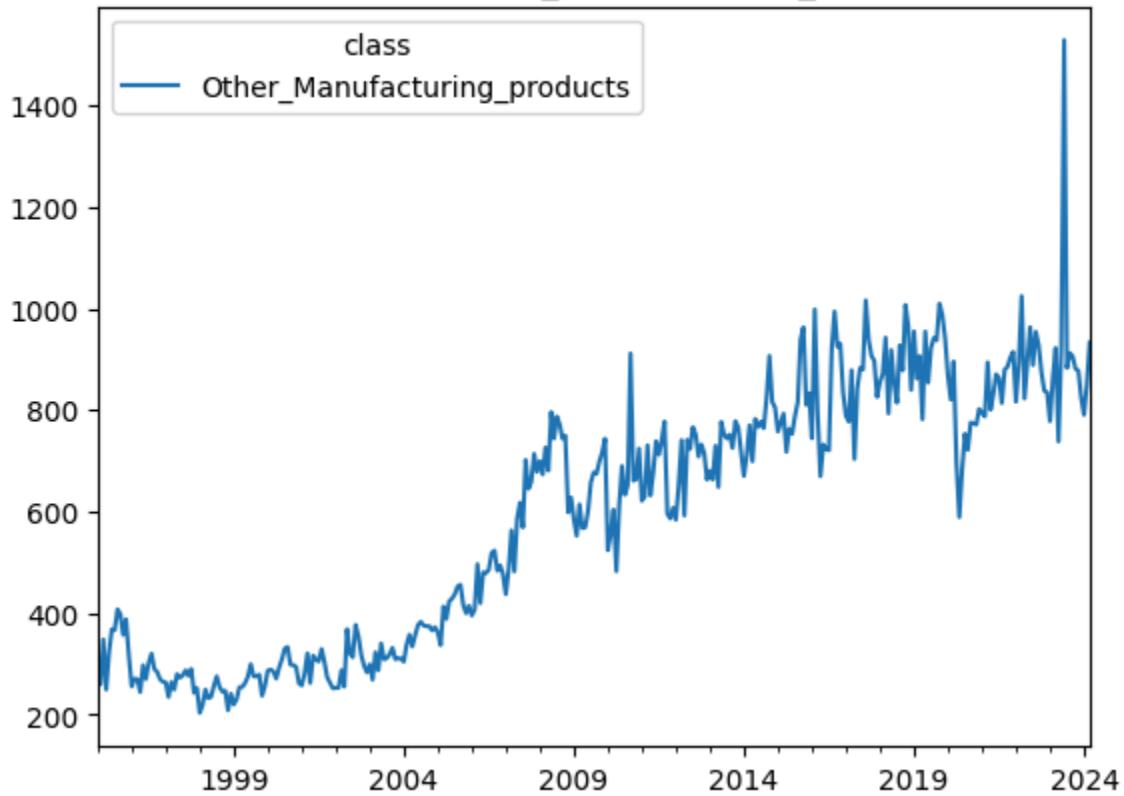
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.019179652565776997
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

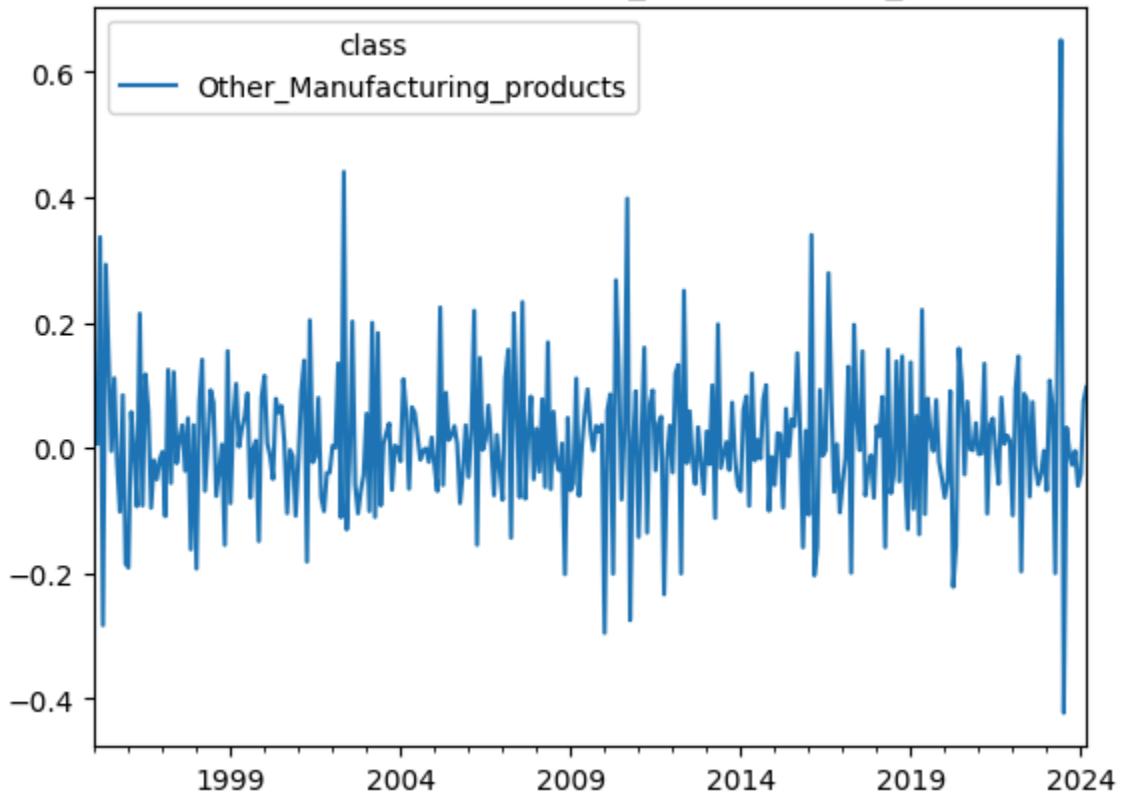
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

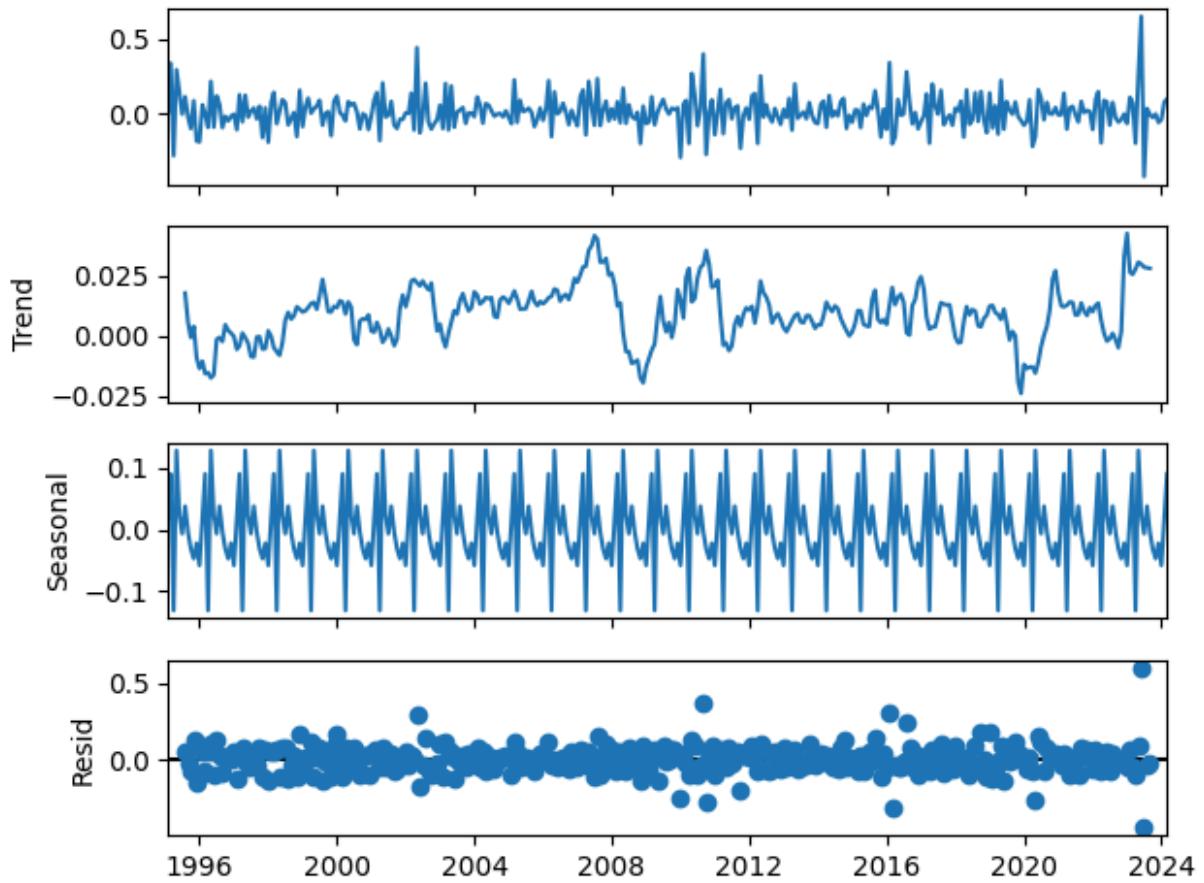
```
result = kpss(series, regression='c')
```

### Export Value (Other\_Manufacturing\_products)



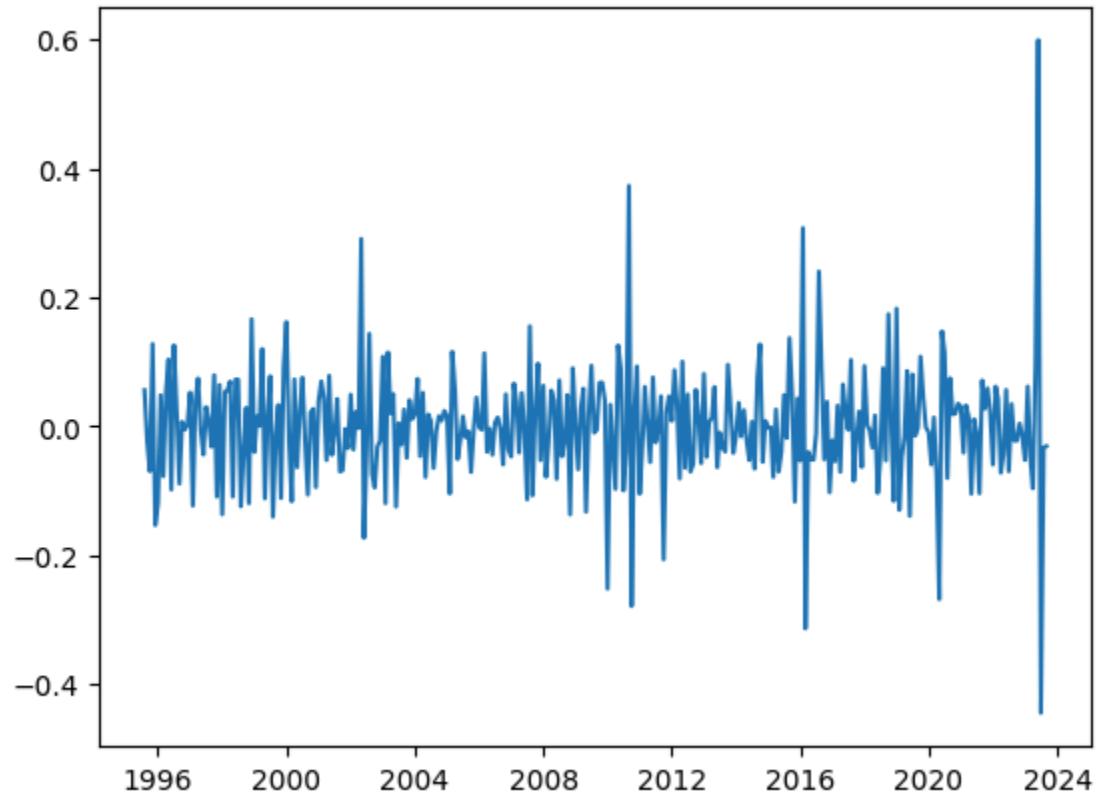
### Export Value % Change (Other\_Manufacturing\_products)

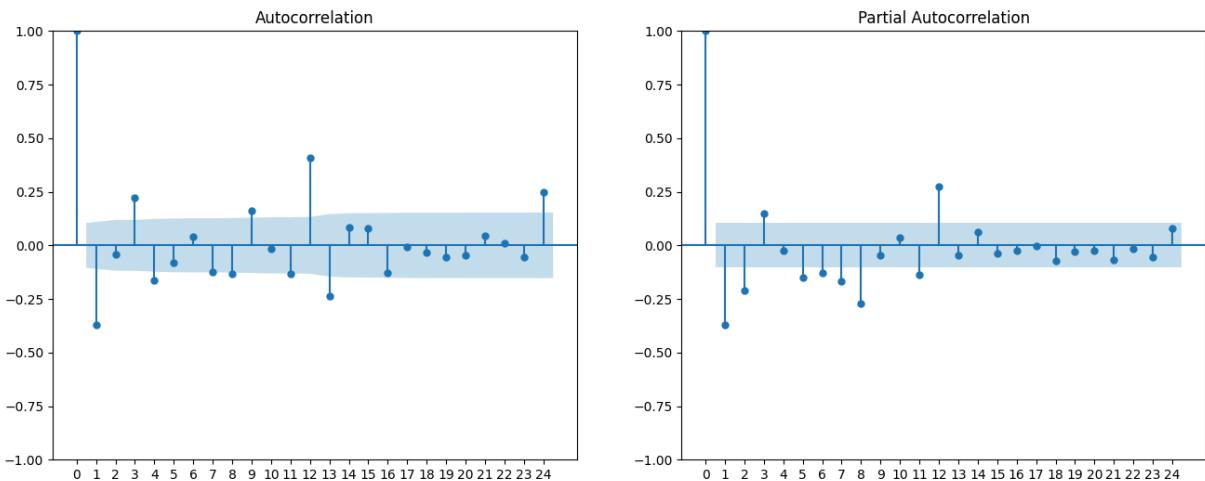




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

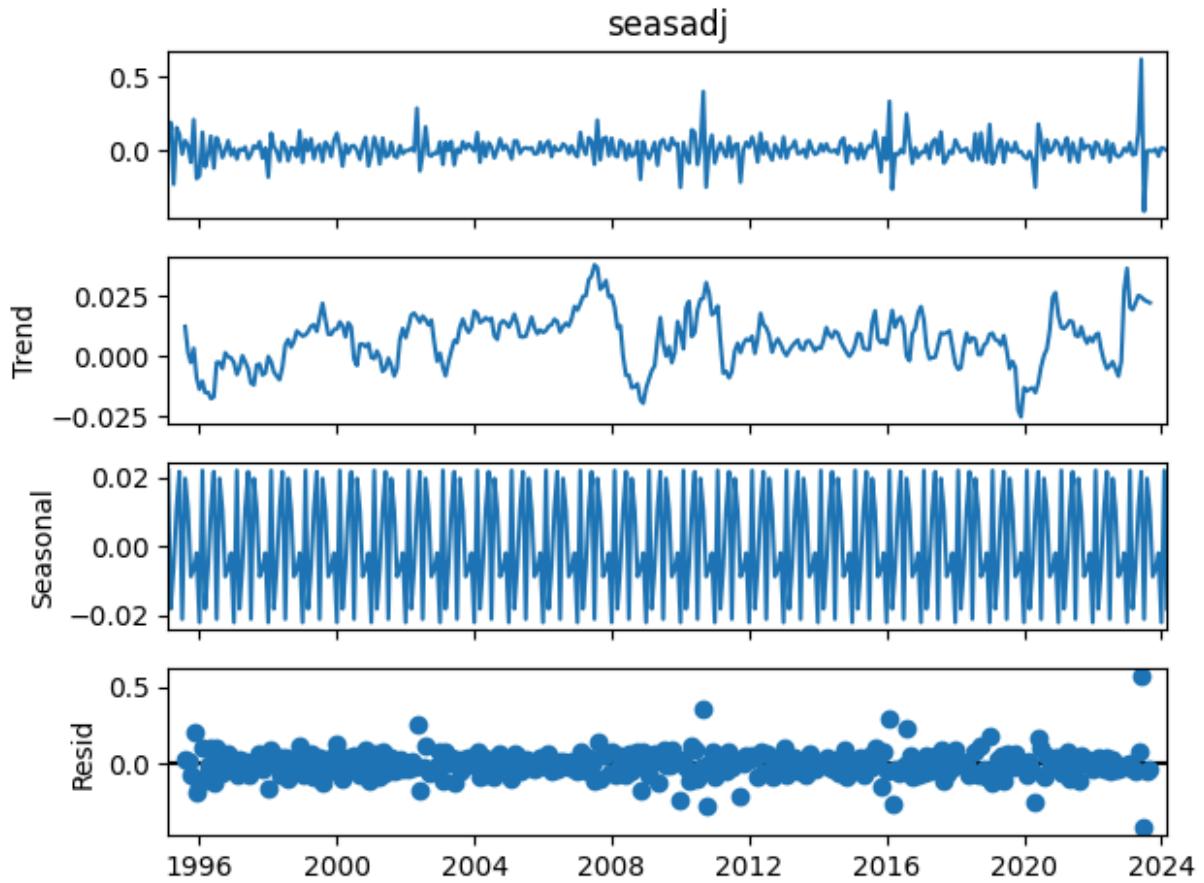
```
'-----
-----'
'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -20.9510003211164
p-value: 0.0
Critical Value 1%: -3.4492815848836296
Critical Value 5%: -2.8698813715275406
Critical Value 10%: -2.5712138845950587
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.024312346586279135
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

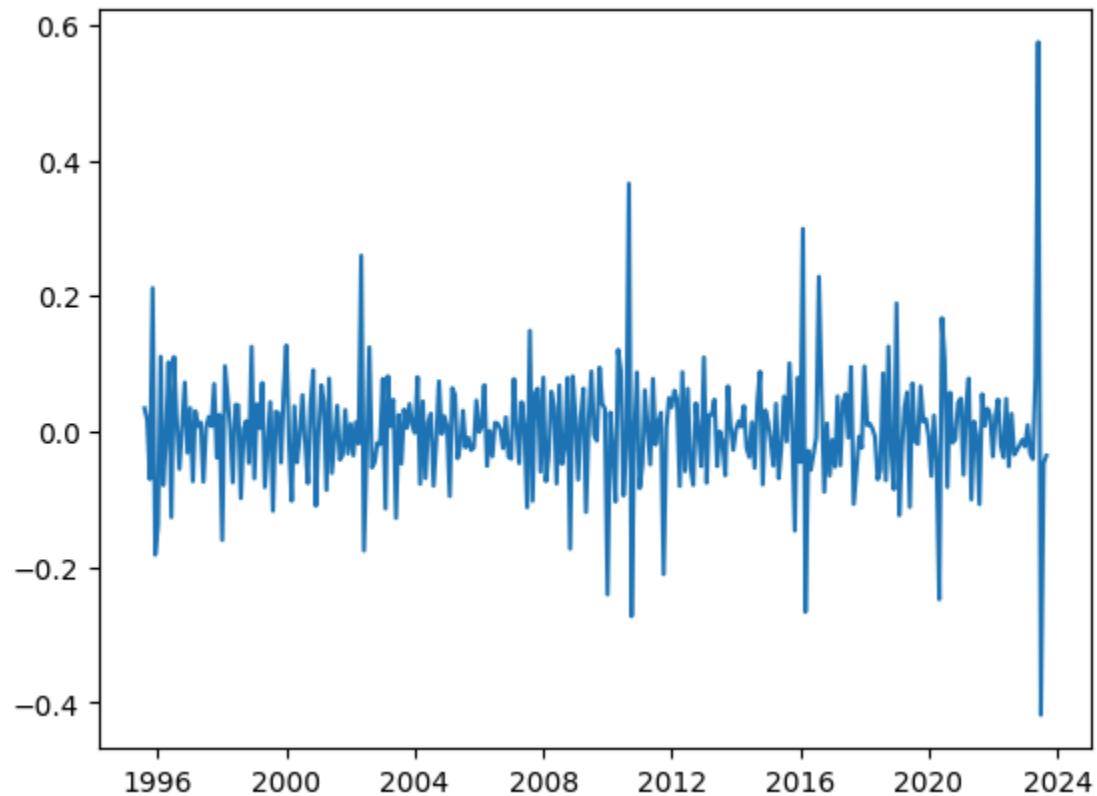
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

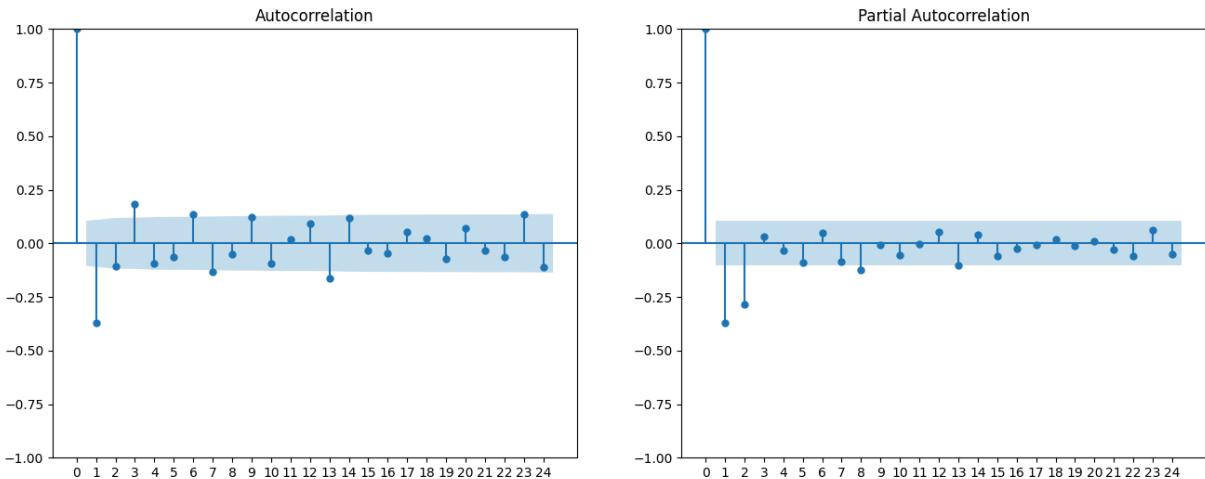
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals





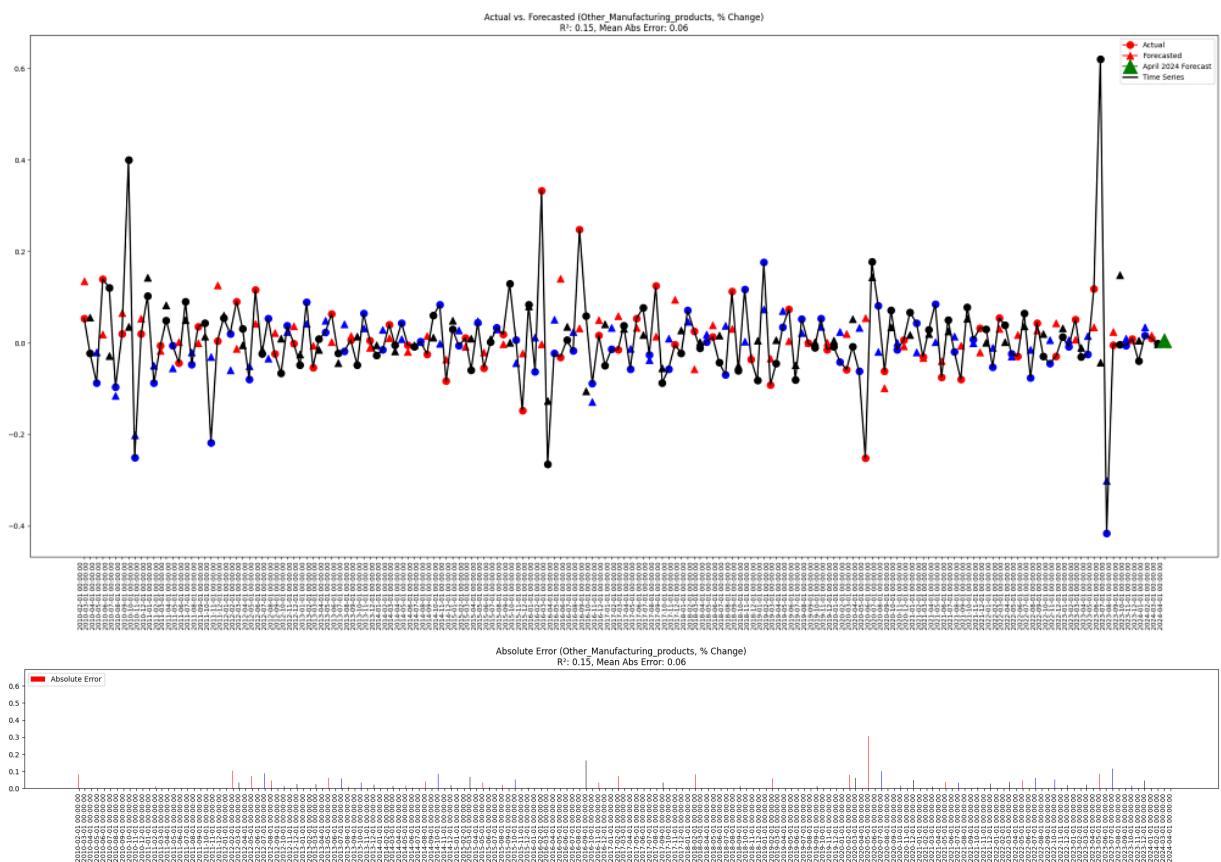
Time series assumptions are met.

'Data of Actual vs Forecasted values with error metrics:'

| Other_Manufacturing_products | actual    | forecast  | error     | abs_error | squared_error | at  |
|------------------------------|-----------|-----------|-----------|-----------|---------------|-----|
| 2010-02-01                   | 0.053193  | 0.135239  | -0.082046 | 0.082046  | 6.731513e-03  |     |
| 2010-03-01                   | -0.022773 | 0.055580  | -0.078353 | 0.078353  | 6.139172e-03  |     |
| 2010-04-01                   | -0.086872 | -0.020352 | -0.066520 | 0.066520  | 4.424905e-03  |     |
| 2010-05-01                   | 0.139705  | 0.019072  | 0.120633  | 0.120633  | 1.455242e-02  |     |
| 2010-06-01                   | 0.120194  | -0.028040 | 0.148234  | 0.148234  | 2.197323e-02  |     |
| ...                          | ...       | ...       | ...       | ...       | ...           | ... |
| 2023-12-01                   | -0.039889 | 0.004855  | -0.044744 | 0.044744  | 2.001995e-03  |     |
| 2024-01-01                   | 0.016909  | 0.034271  | -0.017362 | 0.017362  | 3.014247e-04  |     |
| 2024-02-01                   | 0.009855  | 0.015986  | -0.006131 | 0.006131  | 3.758524e-05  |     |
| 2024-03-01                   | -0.001325 | -0.002076 | 0.000751  | 0.000751  | 5.645446e-07  |     |
| 2024-04-01                   | NaN       | 0.005598  | NaN       | NaN       | NaN           |     |

171 rows × 6 columns

'Total Mean Absolute Error (Per Forecast): 0.057222612566236945'



## Other\_Exports

```
In [83]: df = df_export_ANALYSIS.copy()
name = df.columns[34]
display(f"Component: {name}")

'Component: Other_Exports'
```

```
In [84]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value')
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
check_time_series_assumptions(adj_ts, freq=12)
```

```

display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

| class      | Other_Exports |
|------------|---------------|
| 1995-02-01 | -0.196895     |
| 1995-03-01 | 0.644483      |
| 1995-04-01 | -0.353816     |
| 1995-05-01 | 0.433571      |
| 1995-06-01 | -0.200128     |
| ...        | ...           |
| 2023-11-01 | -0.686372     |
| 2023-12-01 | 0.705487      |
| 2024-01-01 | 0.087158      |
| 2024-02-01 | 0.562042      |
| 2024-03-01 | -0.399582     |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

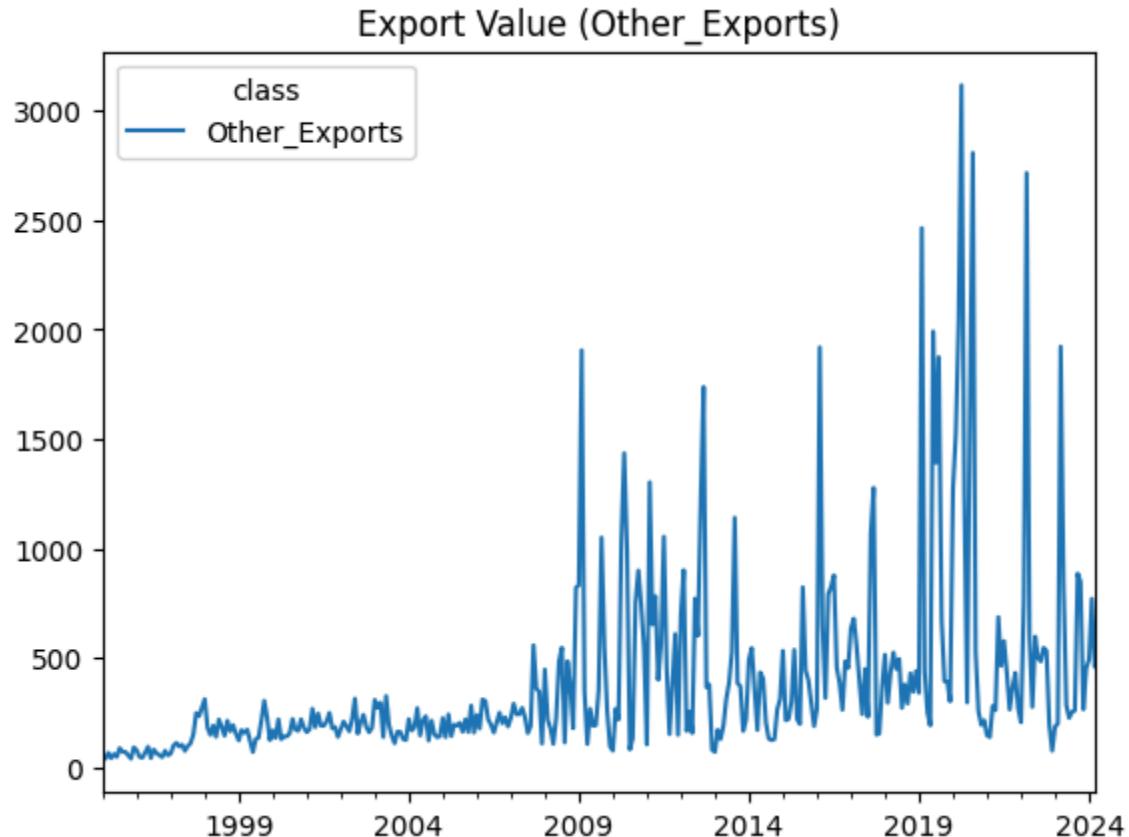
```
Running ADF Test for Stationarity...
ADF Statistic: -13.401065447200093
p-value: 4.565275274095409e-25
Critical Value 1%: -3.449336554273722
Critical Value 5%: -2.8699055166063085
Critical Value 10%: -2.571226758215748
is_stationary: True
```

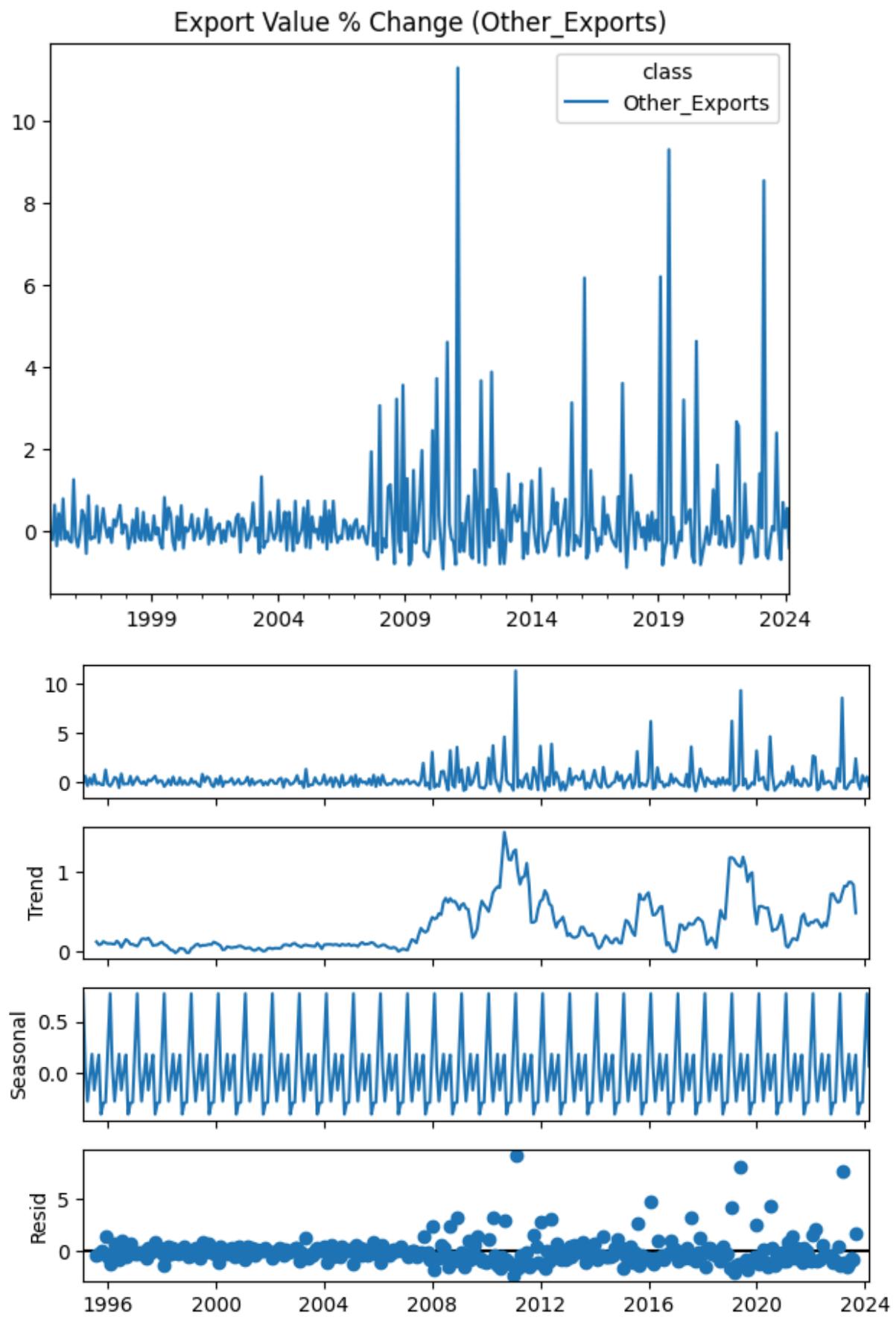
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.7424604725450522
p-value: 0.01
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: False
```

Decomposing the Series...

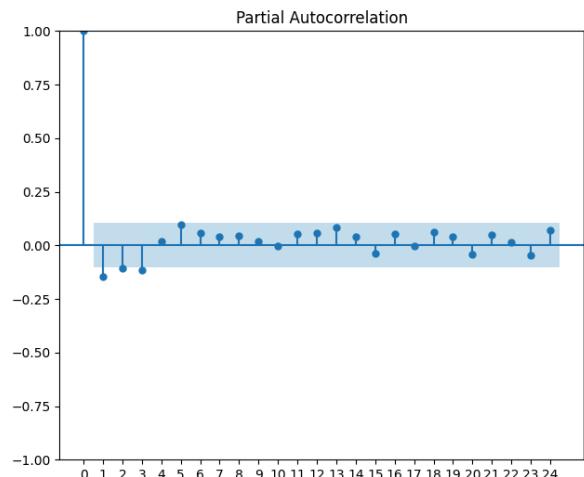
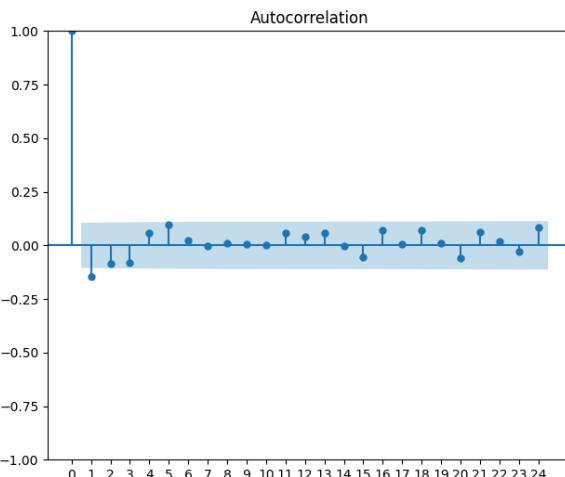
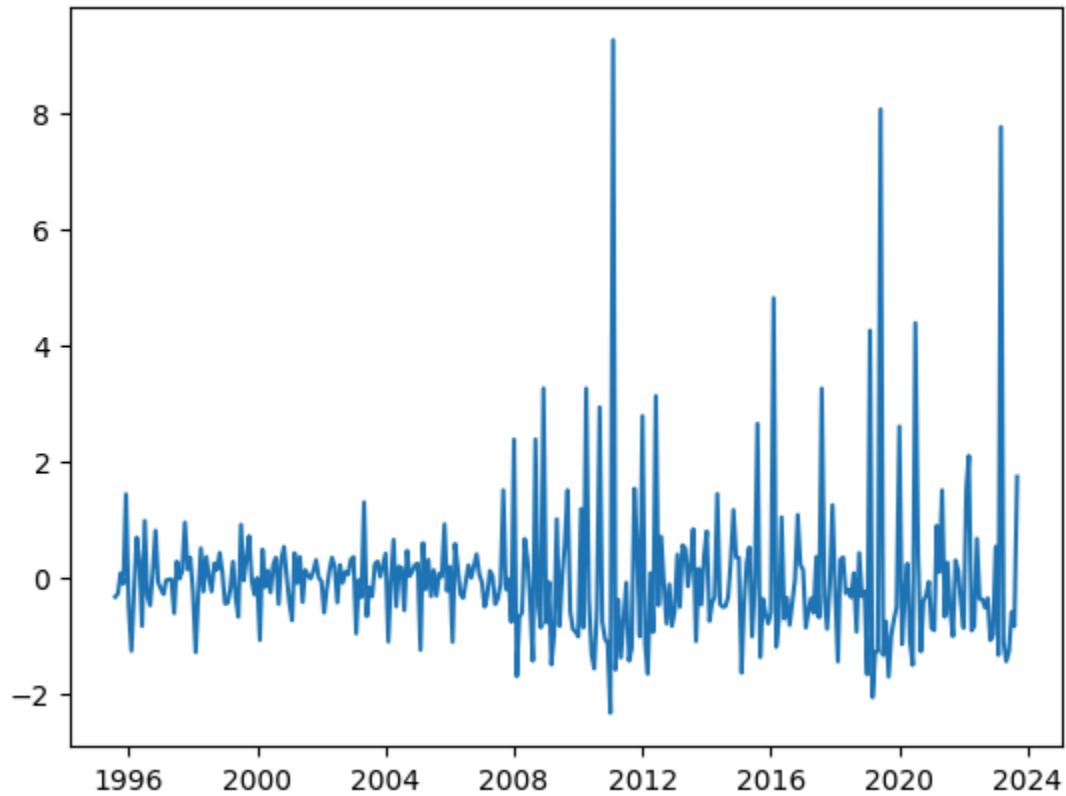
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.
```

```
result = kpss(series, regression='c')
```





### Residuals



Time series assumptions are not met. Further investigation needed.

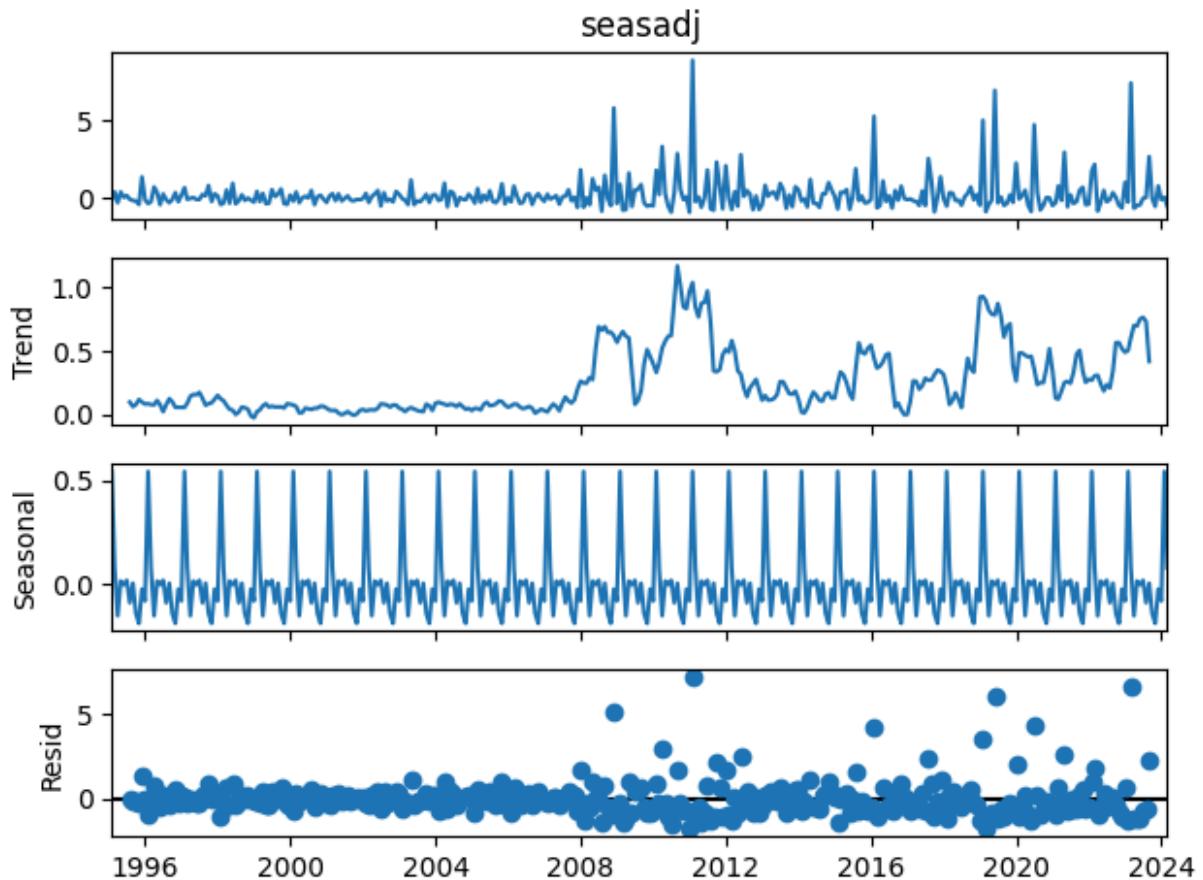
False

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

```
Running ADF Test for Stationarity...
ADF Statistic: -13.48290223972115
p-value: 3.21533330380634e-25
Critical Value 1%: -3.449336554273722
Critical Value 5%: -2.8699055166063085
Critical Value 10%: -2.571226758215748
is_stationary: True
```

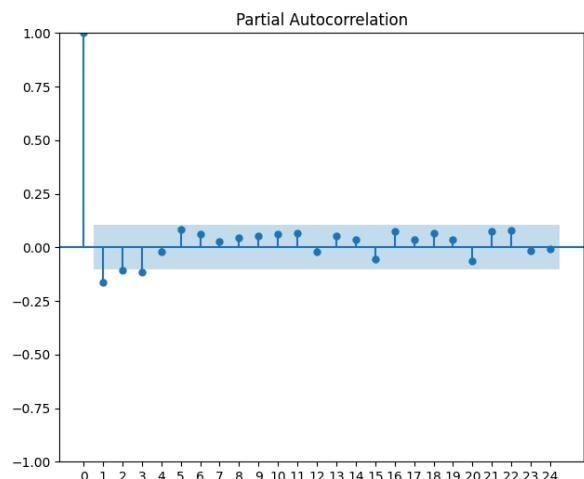
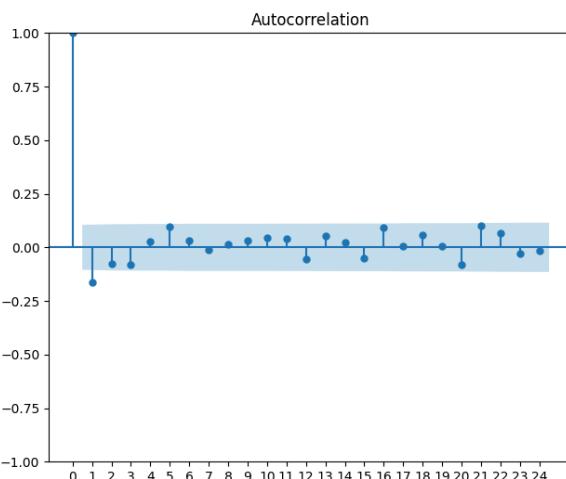
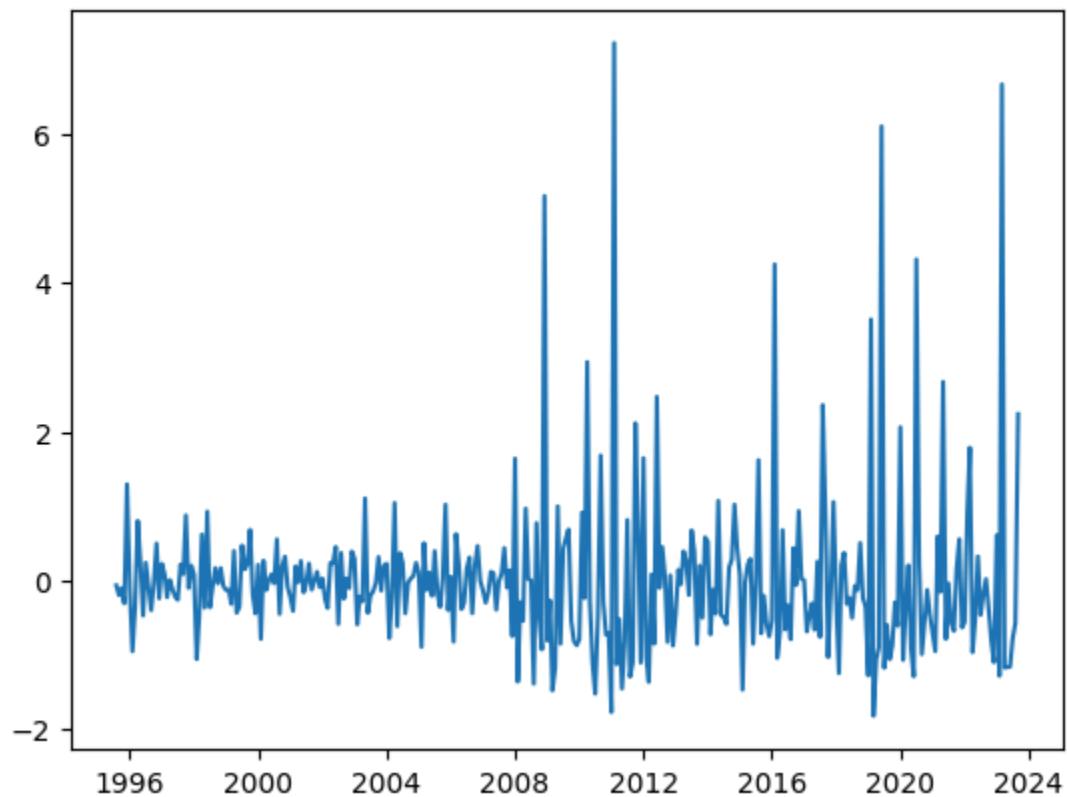
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.7371503022645324
p-value: 0.010168154339587966
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: False
```

Decomposing the Series...



Plotting Residuals for Homoscedasticity Check...

### Residuals



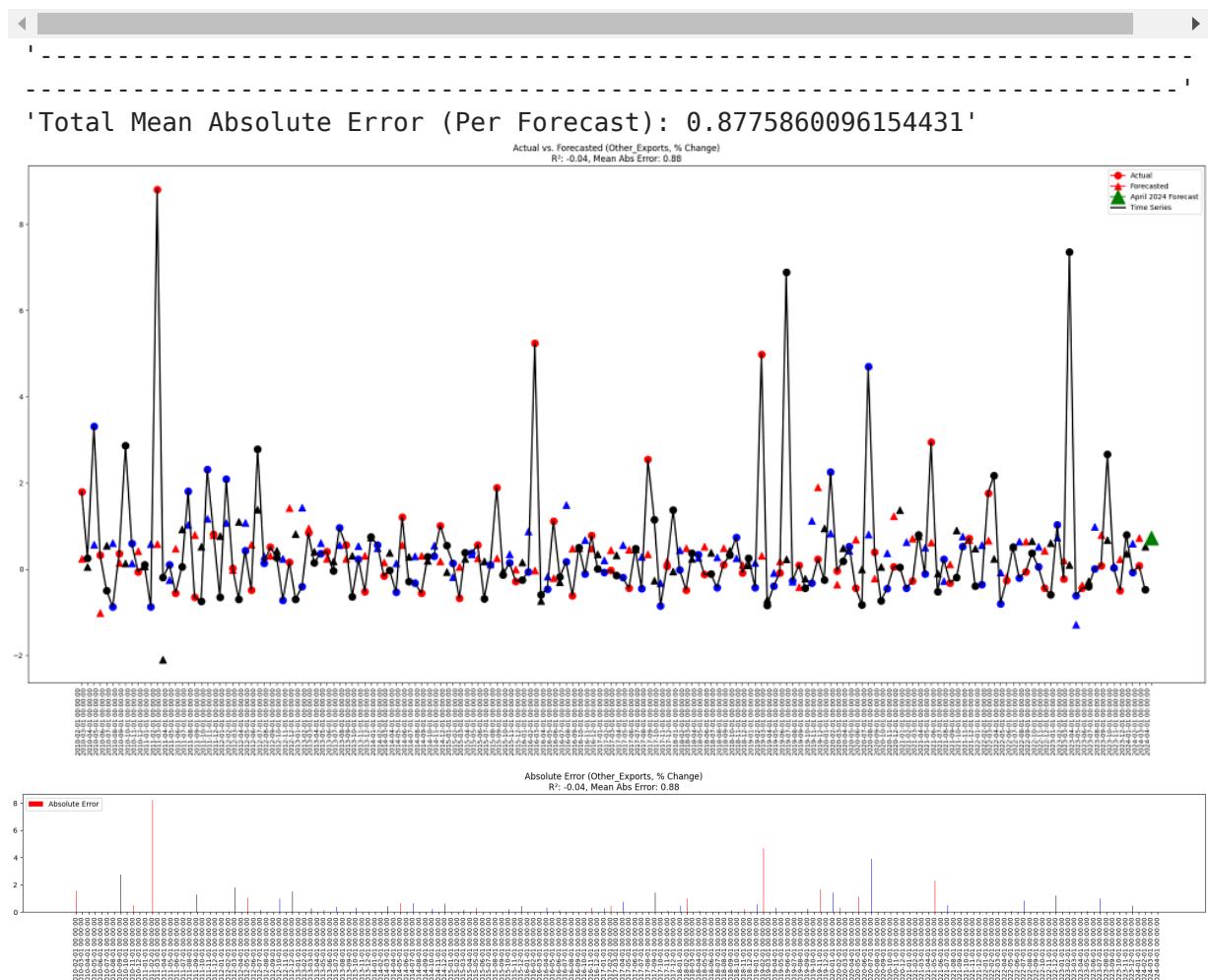
Time series assumptions are not met. Further investigation needed.

'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Other_Exports | actual    | forecast  | error     | abs_error | squared_error | abs_percentage_er |
|---------------|-----------|-----------|-----------|-----------|---------------|-------------------|
| 2010-02-01    | 1.791389  | 0.240574  | 1.550814  | 1.550814  | 2.405025      | 0.865             |
| 2010-03-01    | 0.250659  | 0.056105  | 0.194554  | 0.194554  | 0.037851      | 0.776             |
| 2010-04-01    | 3.309913  | 0.567173  | 2.742740  | 2.742740  | 7.522622      | 0.828             |
| 2010-05-01    | 0.326369  | -1.018965 | 1.345334  | 1.345334  | 1.809923      | 4.122             |
| 2010-06-01    | -0.495064 | 0.552673  | -1.047737 | 1.047737  | 1.097754      | 2.116             |
| ...           | ...       | ...       | ...       | ...       | ...           | ...               |
| 2023-12-01    | 0.798963  | 0.362539  | 0.436423  | 0.436423  | 0.190465      | 0.546             |
| 2024-01-01    | -0.072353 | 0.590413  | -0.662765 | 0.662765  | 0.439258      | 9.160             |
| 2024-02-01    | 0.077346  | 0.720935  | -0.643589 | 0.643589  | 0.414206      | 8.320             |
| 2024-03-01    | -0.478766 | 0.519218  | -0.997984 | 0.997984  | 0.995972      | 2.084             |
| 2024-04-01    | NaN       | 0.727948  | NaN       | NaN       | NaN           | NaN               |

171 rows × 6 columns



## Re-Exports\_1

```
In [85]: df = df_export_ANALYSIS.copy()
name = df.columns[35]
display(f"Component: {name}")

'Component: Re-Exports_1'
```

```
In [86]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.replace([np.inf, -np.inf], np.nan).replace(0, 1e-6).ffill().bfill()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

| class      | Re-Exports_1 |
|------------|--------------|
| 1995-02-01 | -0.580420    |
| 1995-03-01 | 0.300000     |
| 1995-04-01 | 0.139423     |
| 1995-05-01 | -0.094233    |
| 1995-06-01 | 0.031056     |
| ...        | ...          |
| 2023-11-01 | 0.692355     |
| 2023-12-01 | -0.583607    |
| 2024-01-01 | 0.902887     |
| 2024-02-01 | -0.383908    |
| 2024-03-01 | -0.429104    |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -18.68154826795235
p-value: 2.041775341822564e-30
Critical Value 1%: -3.449226932880019
Critical Value 5%: -2.869857365438656
Critical Value 10%: -2.571201085130664
is_stationary: True
```

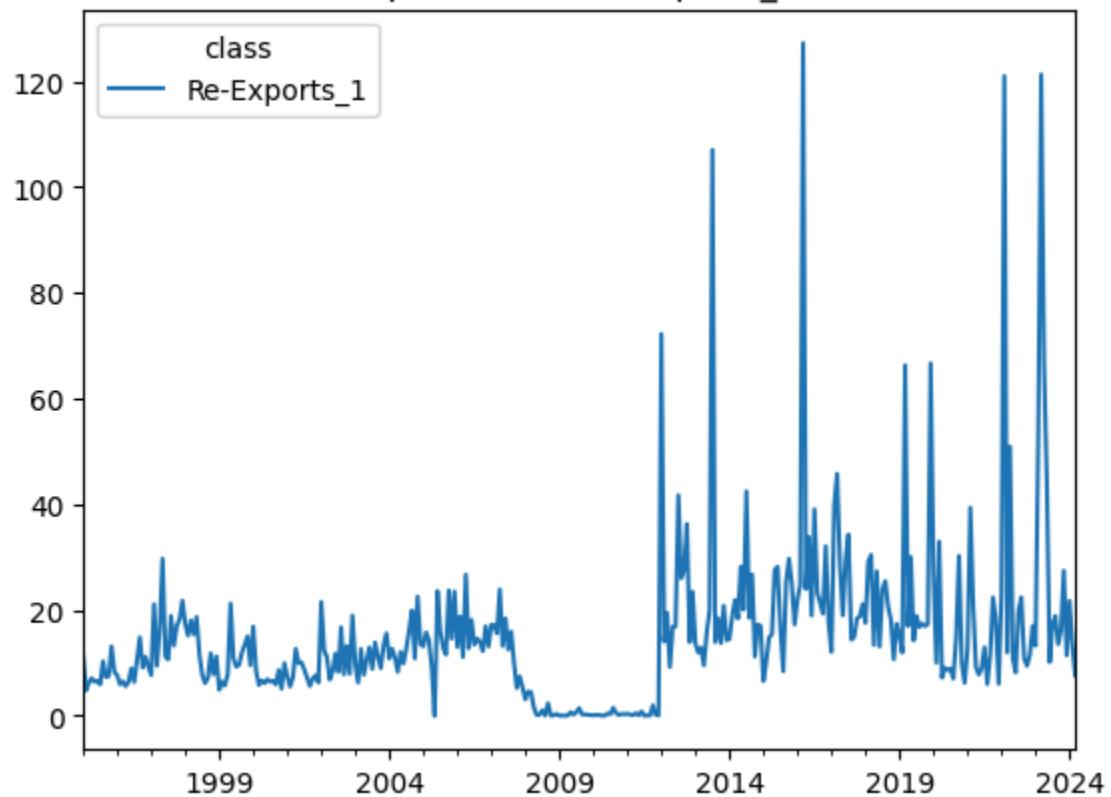
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.10444799720805127
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

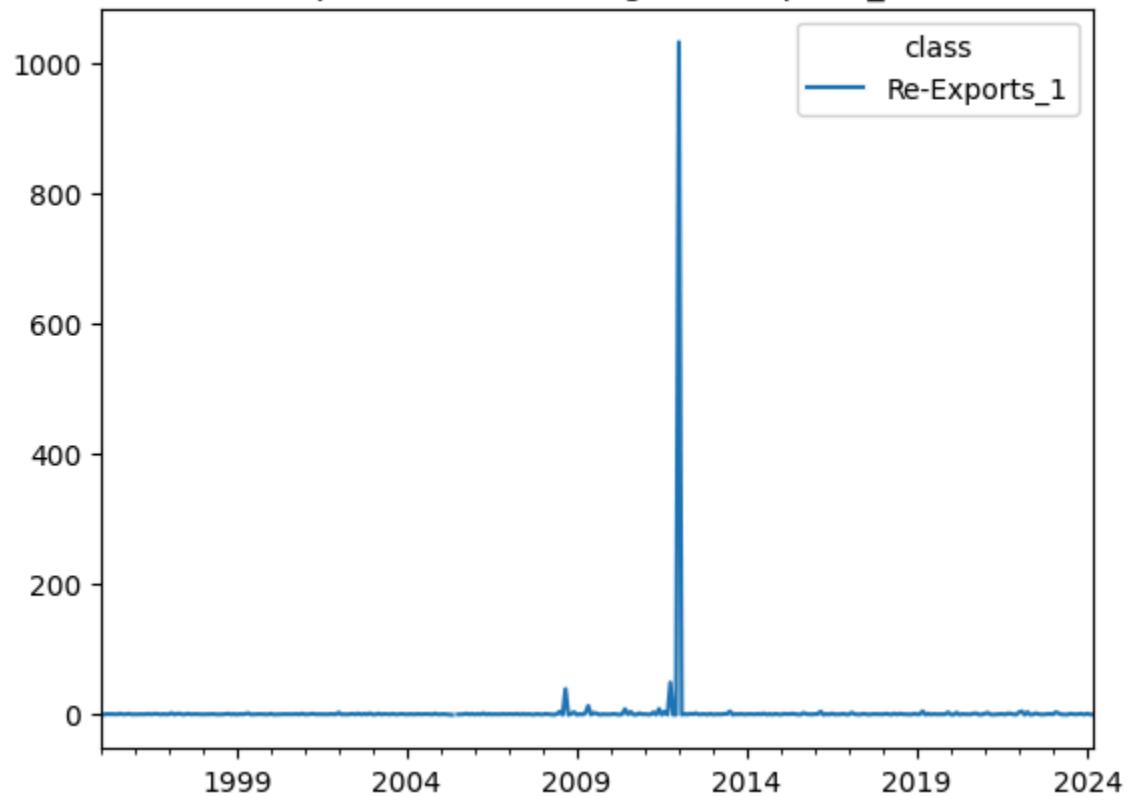
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

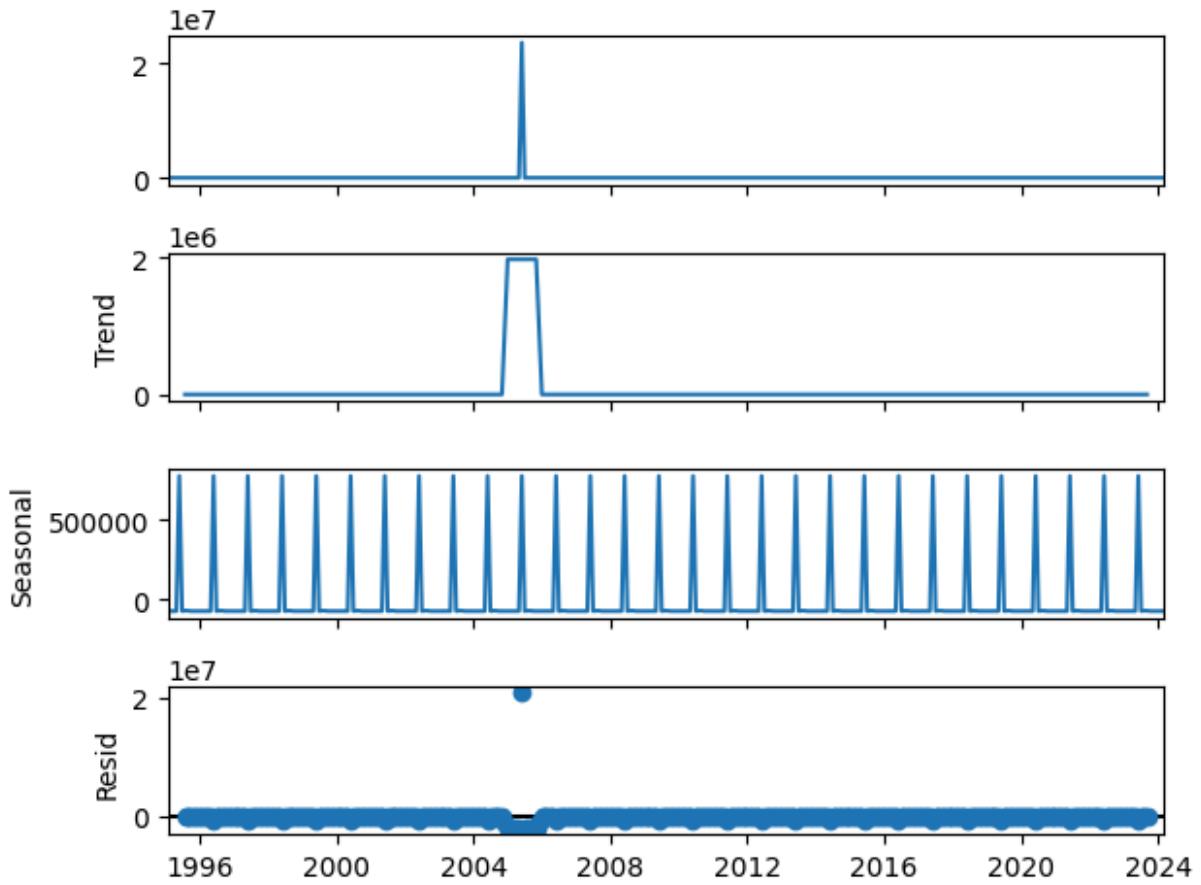
```
result = kpss(series, regression='c')
```

## Export Value (Re-Exports\_1)

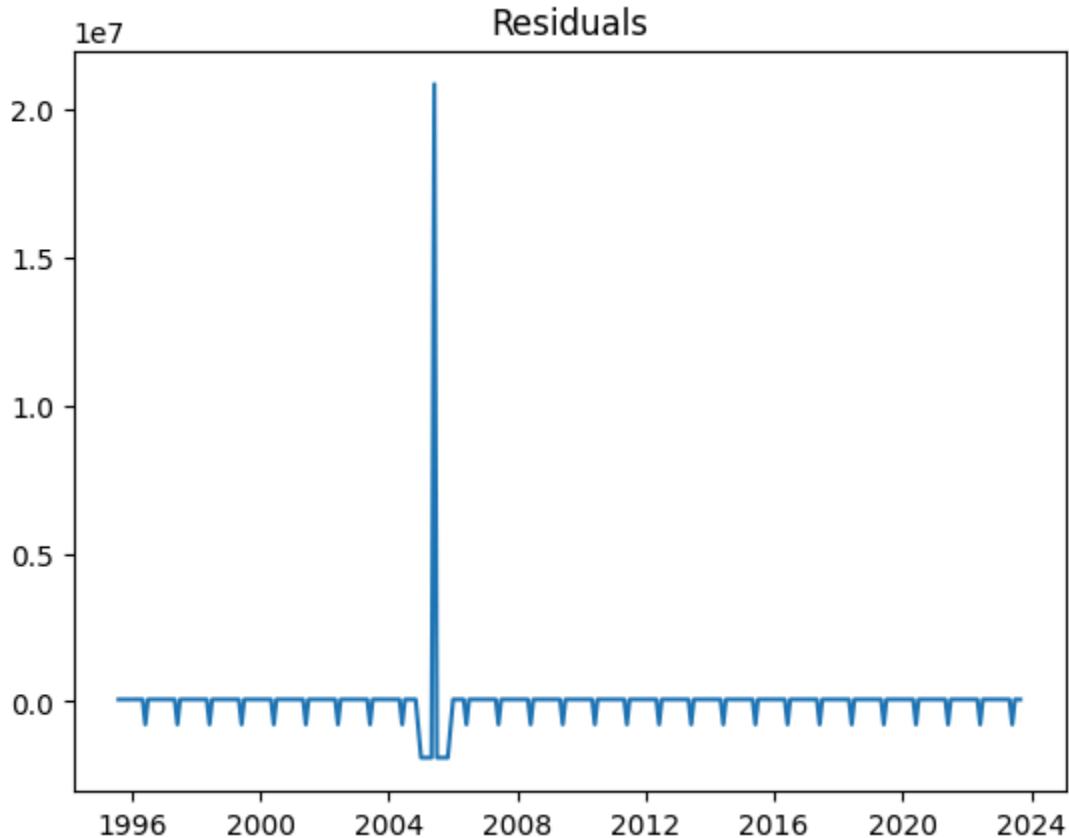


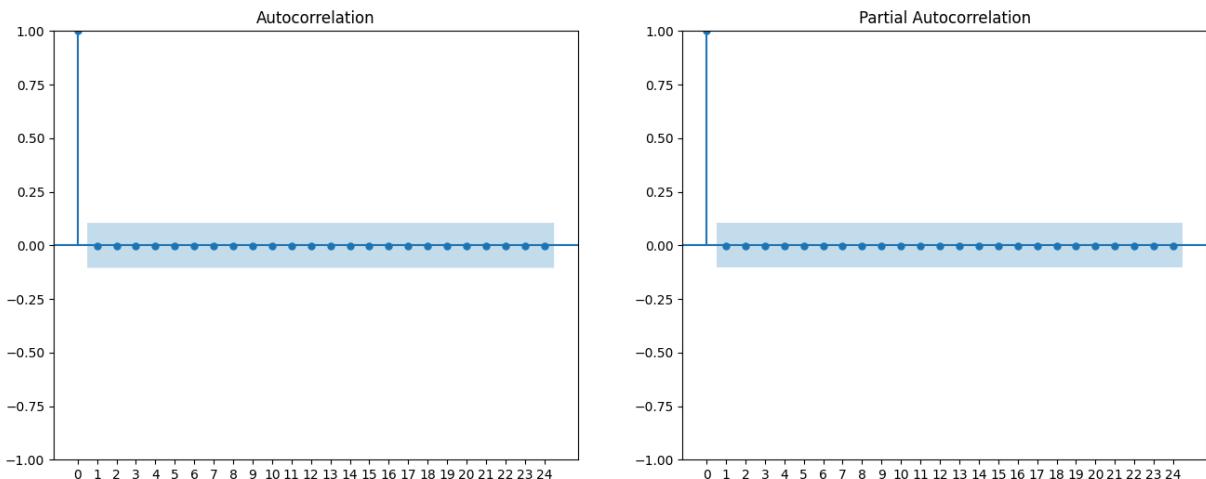
## Export Value % Change (Re-Exports\_1)





Plotting Residuals for Homoscedasticity Check...





Time series assumptions are met.

True

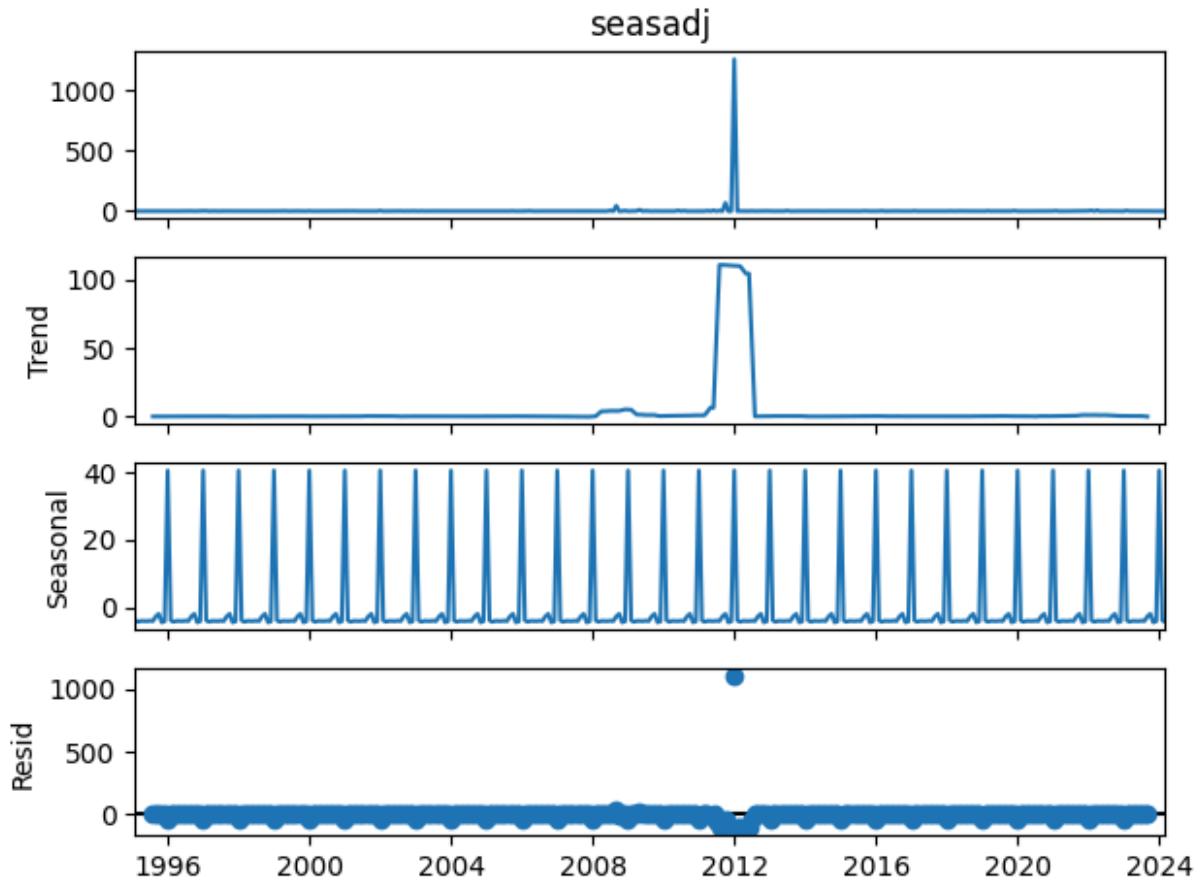
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -18.72356949487054
p-value: 2.032193647609522e-30
Critical Value 1%: -3.449226932880019
Critical Value 5%: -2.869857365438656
Critical Value 10%: -2.571201085130664
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.10370071594865692
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

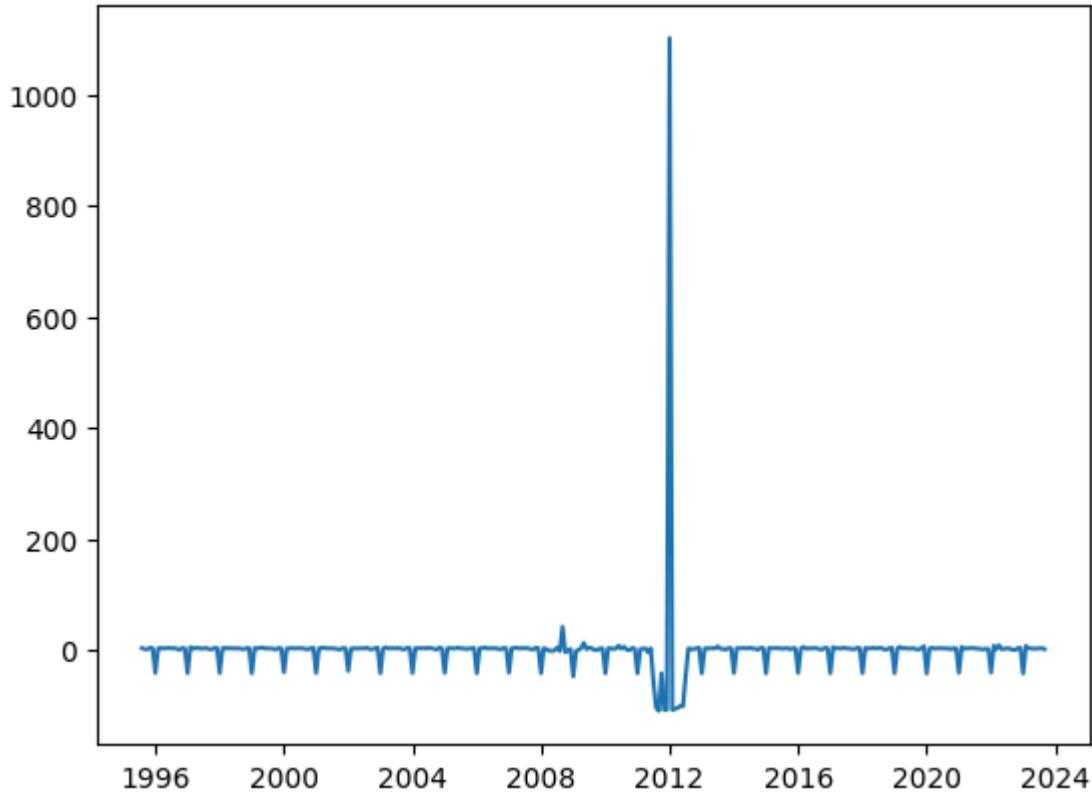
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

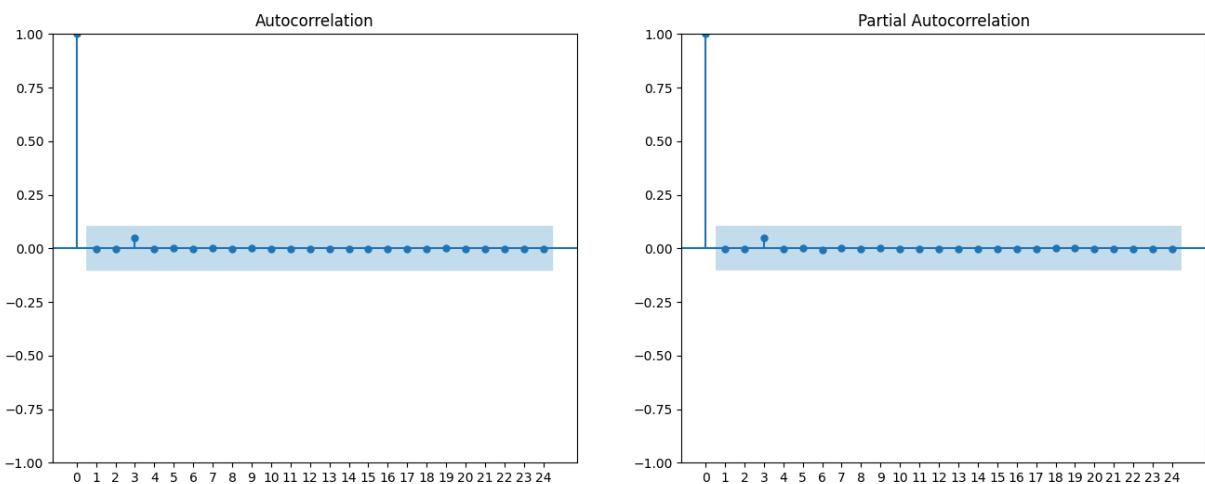
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals





Time series assumptions are met.

'-----'

'Data of Actual vs Forecasted values with error metrics:'

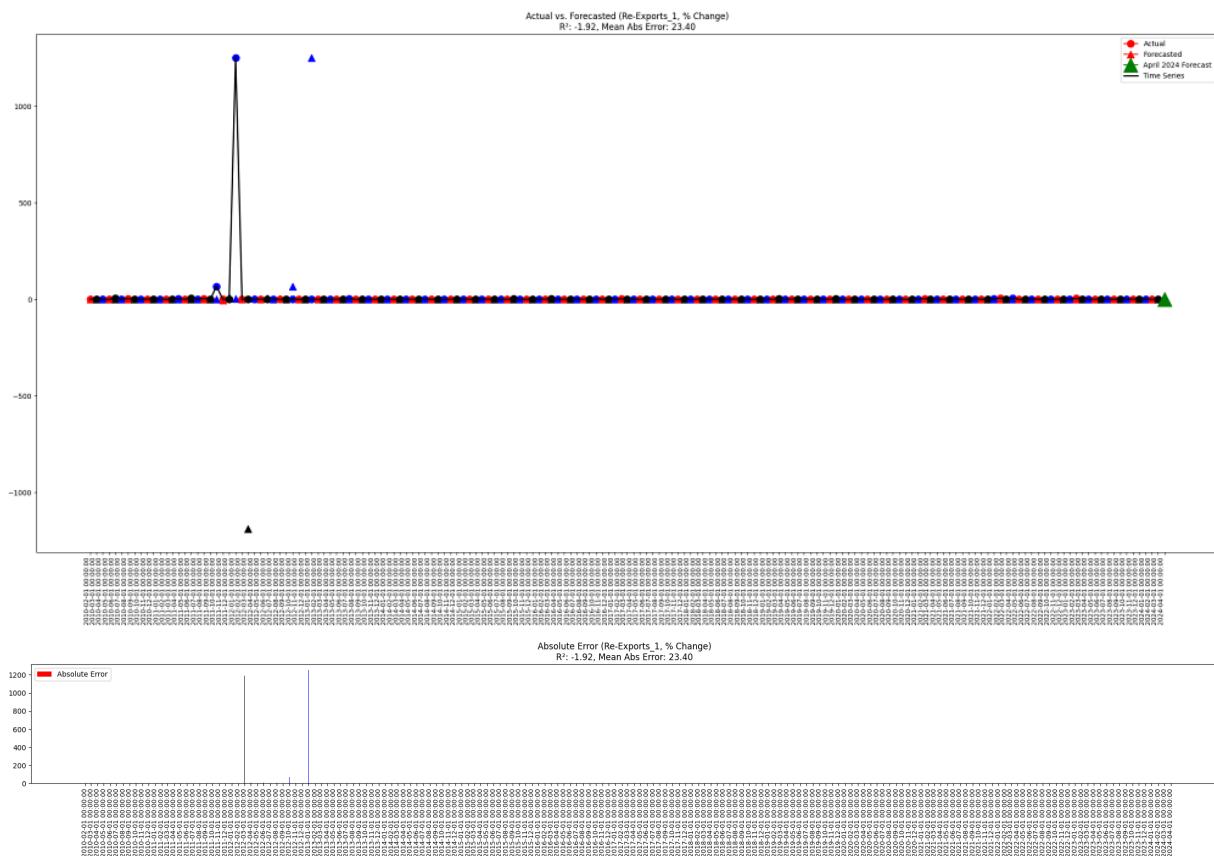
| Re-Exports_1 | actual    | forecast | error     | abs_error | squared_error | abs_percentage_error |
|--------------|-----------|----------|-----------|-----------|---------------|----------------------|
| 2010-02-01   | 0.366390  | 0.423256 | -0.056866 | 0.056866  | 0.003234      | 0.155206             |
| 2010-03-01   | 0.121134  | 0.422942 | -0.301809 | 0.301809  | 0.091088      | 2.491537             |
| 2010-04-01   | -0.658208 | 0.421284 | -1.079492 | 1.079492  | 1.165302      | 1.640047             |
| 2010-05-01   | -0.178775 | 0.415385 | -0.594160 | 0.594160  | 0.353026      | 3.323512             |
| 2010-06-01   | 5.207109  | 0.412156 | 4.794954  | 4.794954  | 22.991579     | 0.920847             |
| ...          | ...       | ...      | ...       | ...       | ...           | ...                  |
| 2023-12-01   | -0.459824 | 0.000000 | -0.459824 | 0.459824  | 0.211438      | 1.000000             |
| 2024-01-01   | 0.331094  | 0.000000 | 0.331094  | 0.331094  | 0.109623      | 1.000000             |
| 2024-02-01   | -0.149089 | 0.000000 | -0.149089 | 0.149089  | 0.022228      | 1.000000             |
| 2024-03-01   | -0.615727 | 0.000000 | -0.615727 | 0.615727  | 0.379119      | 1.000000             |
| 2024-04-01   | NaN       | 0.000000 | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns

'-----'

'-----'

'Total Mean Absolute Error (Per Forecast): 23.398866459506547'



Forecast to actual is quite tightly bound except for some forecast which are extreme outlier, as such, this model will benefit from percentile regularization of forecast

```
In [87]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.replace([np.inf, -np.inf], np.nan).replace(0, 1e-6).ffill().bfill()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

# read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted)")
display(check_time_series_assumptions(adj_ts, freq=12))
display("-" * 150)
```

```

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined")
df_accuracy.columns.name = name
df_accuracy["forecast"] = cap_extreme_values(df_accuracy[["forecast"]])["forecast"]
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

```

'Timeseries of component in % form:'

class Re-Exports\_1

|                   |           |
|-------------------|-----------|
| <b>1995-02-01</b> | -0.580420 |
| <b>1995-03-01</b> | 0.300000  |
| <b>1995-04-01</b> | 0.139423  |
| <b>1995-05-01</b> | -0.094233 |
| <b>1995-06-01</b> | 0.031056  |
| ...               | ...       |
| <b>2023-11-01</b> | 0.692355  |
| <b>2023-12-01</b> | -0.583607 |
| <b>2024-01-01</b> | 0.902887  |
| <b>2024-02-01</b> | -0.383908 |
| <b>2024-03-01</b> | -0.429104 |

350 rows × 1 columns

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted)':'

```
Running ADF Test for Stationarity...
ADF Statistic: -18.68154826795235
p-value: 2.041775341822564e-30
Critical Value 1%: -3.449226932880019
Critical Value 5%: -2.869857365438656
Critical Value 10%: -2.571201085130664
is_stationary: True
```

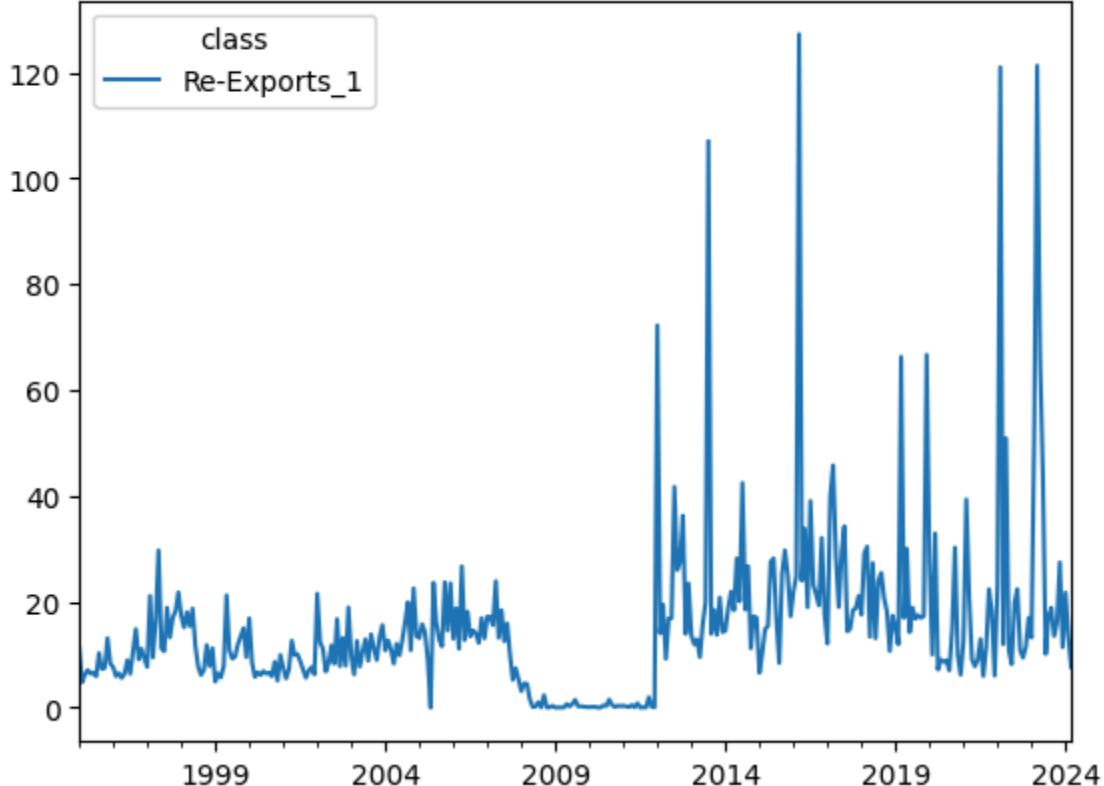
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.10444799720805127
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

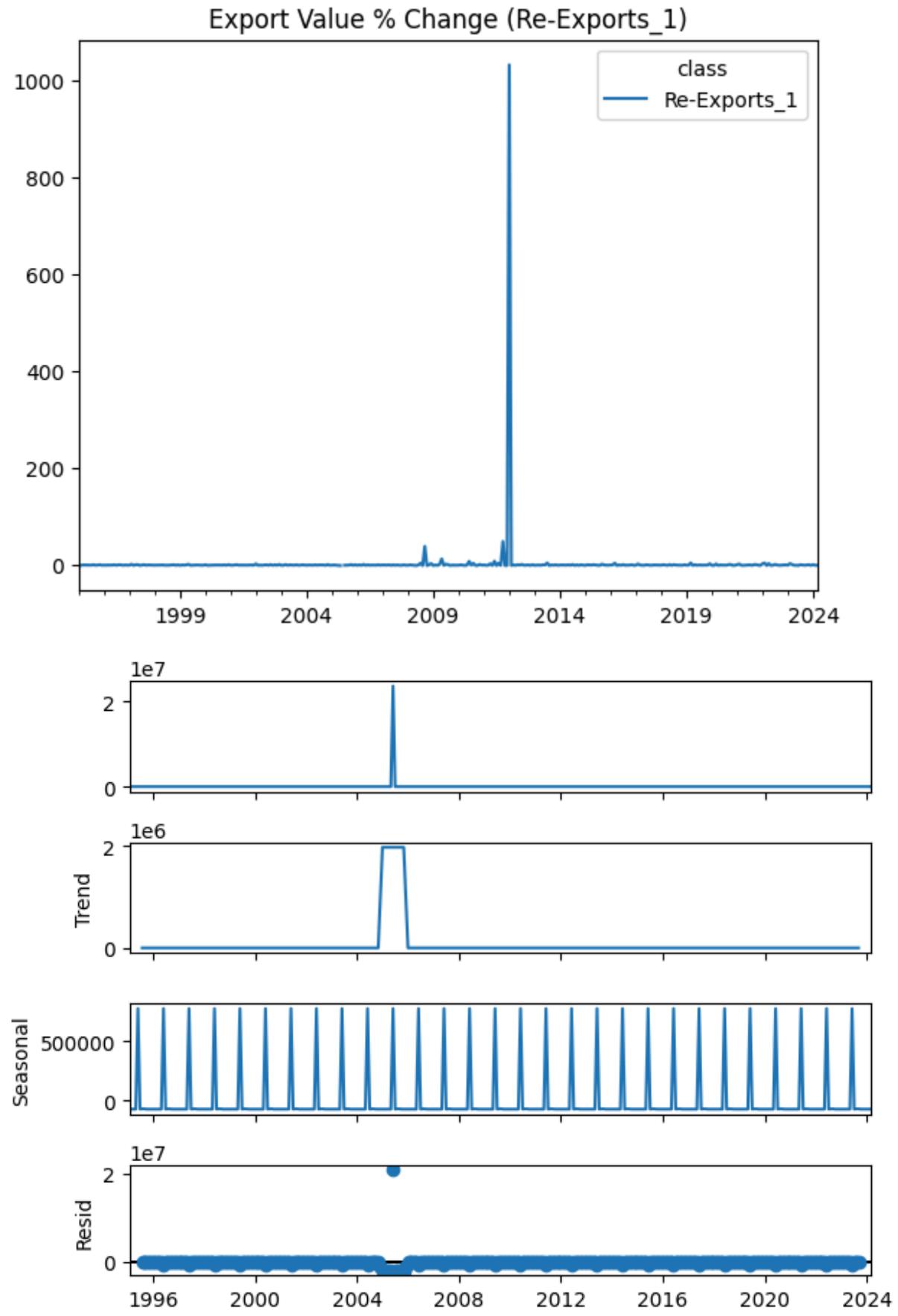
Decomposing the Series...

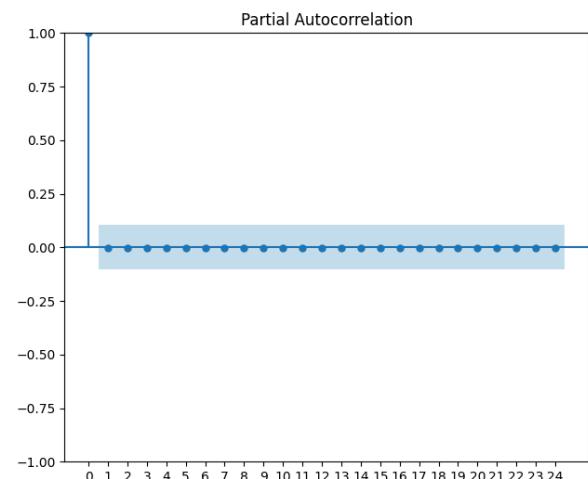
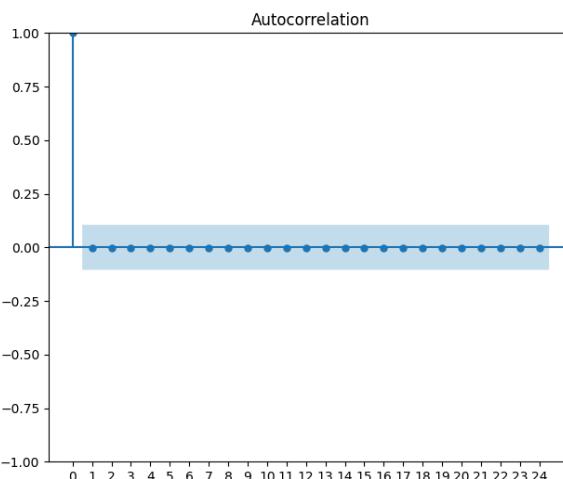
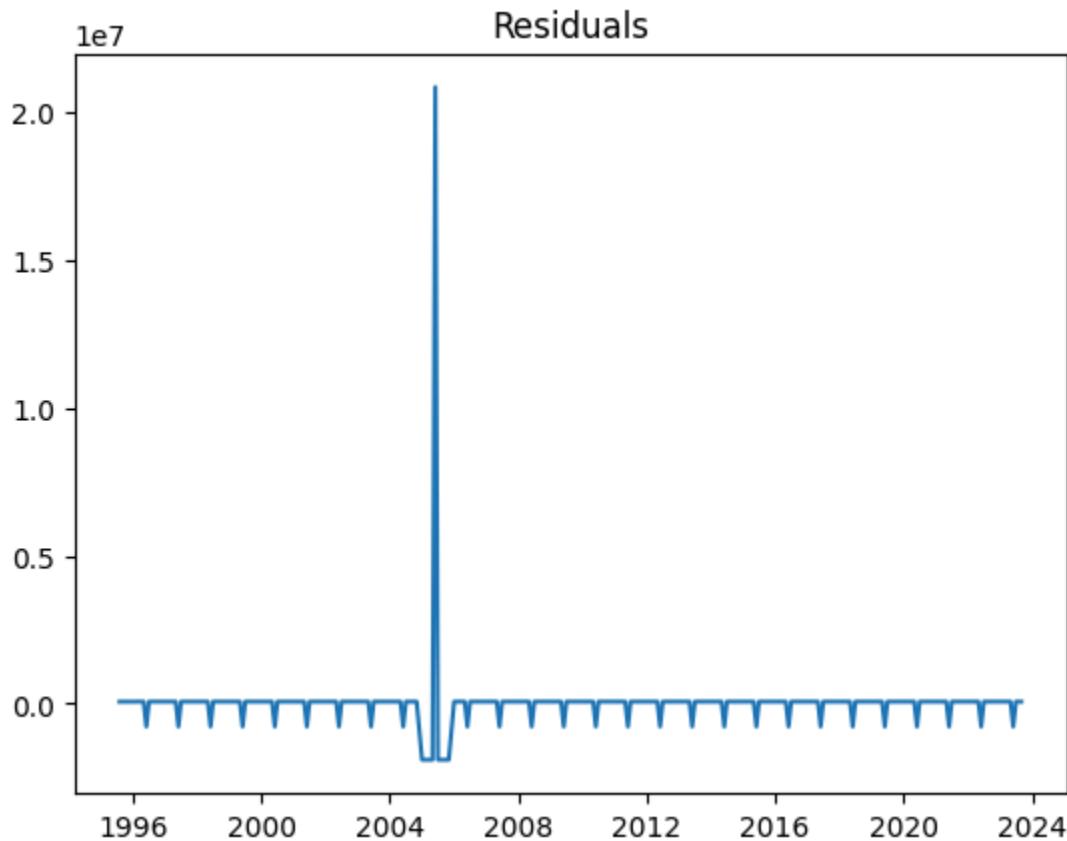
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```

Export Value (Re-Exports\_1)







Time series assumptions are met.

True

'-----'  
-----'  
'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

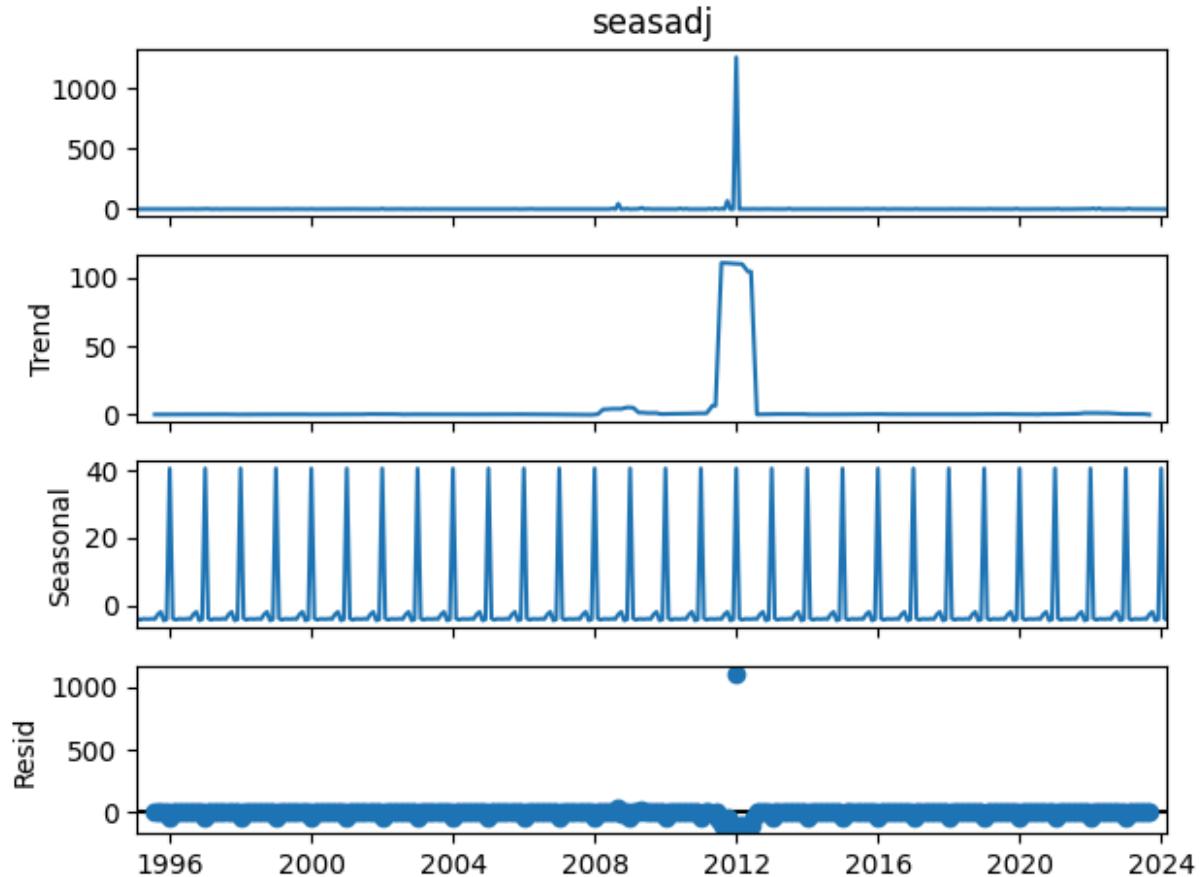
```
Running ADF Test for Stationarity...
ADF Statistic: -18.72356949487054
p-value: 2.032193647609522e-30
Critical Value 1%: -3.449226932880019
Critical Value 5%: -2.869857365438656
Critical Value 10%: -2.571201085130664
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.10370071594865692
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

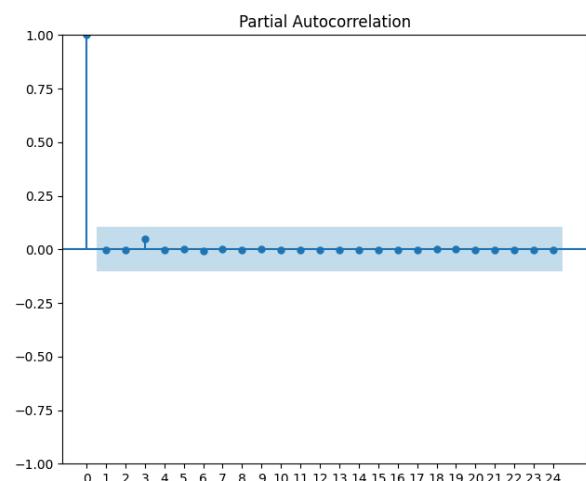
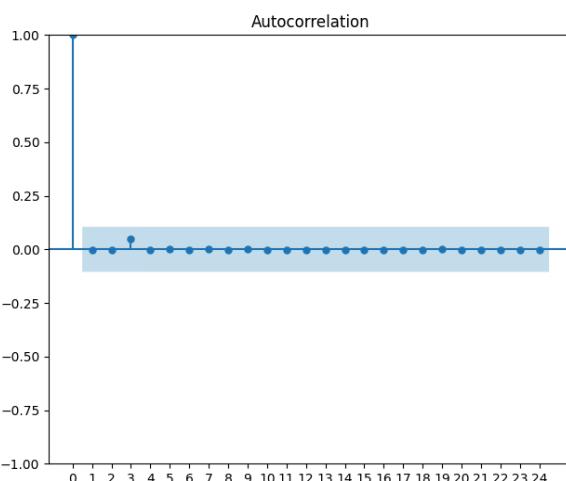
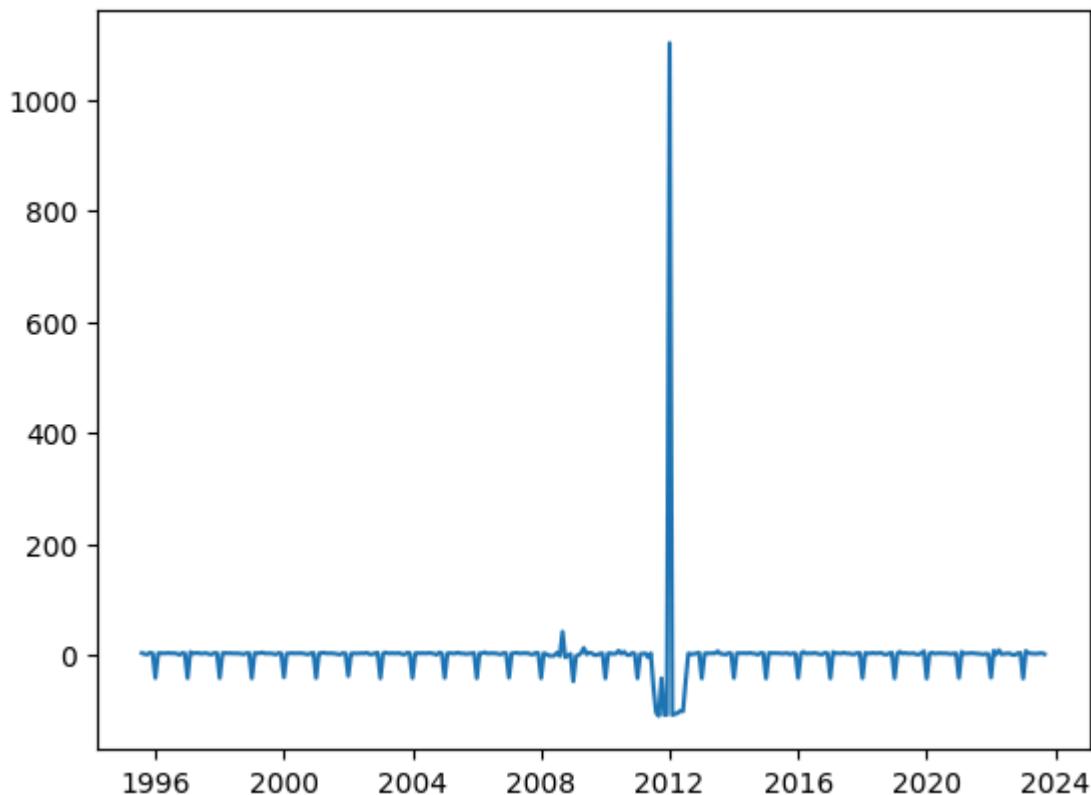
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

### Residuals



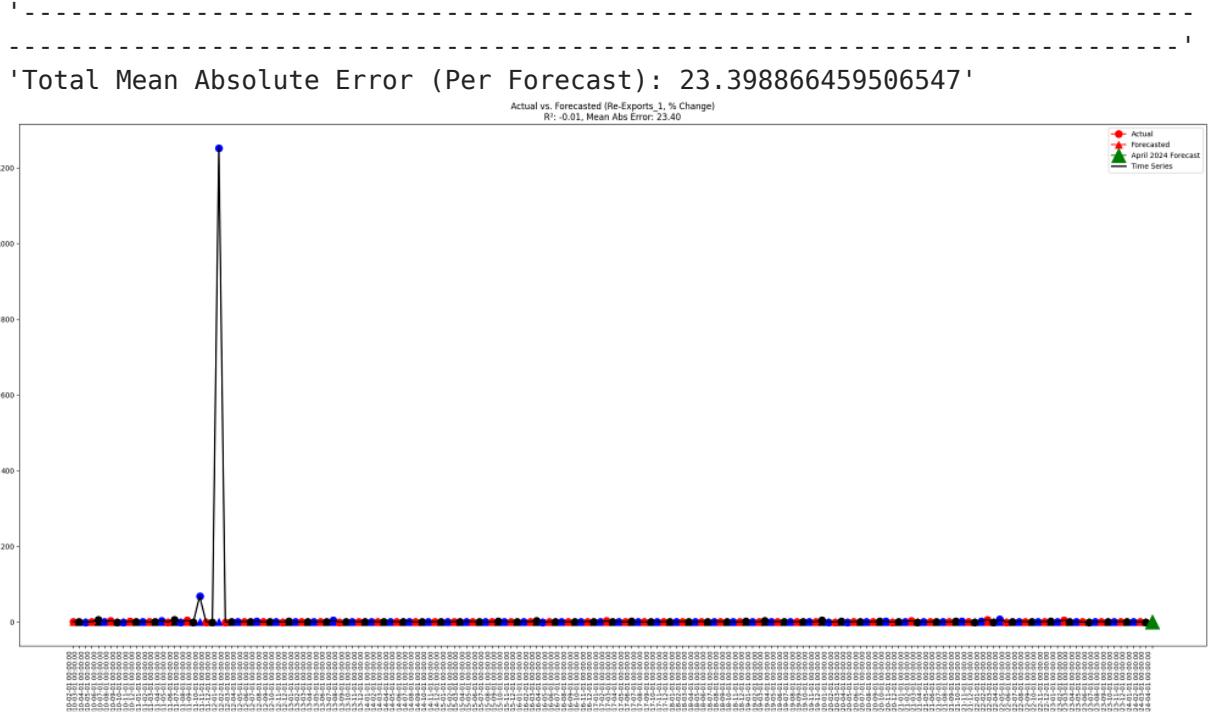
Time series assumptions are met.

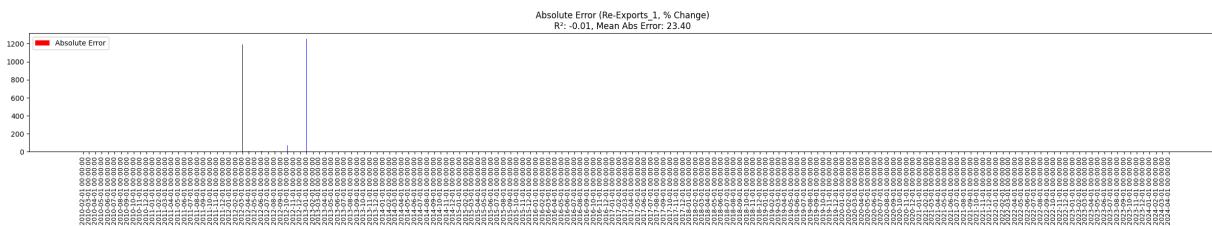
'-----'

'Data of Actual vs Forecasted values with error metrics:'

| Re-Exports_1 | actual    | forecast | error     | abs_error | squared_error | abs_percentage_error |
|--------------|-----------|----------|-----------|-----------|---------------|----------------------|
| 2010-02-01   | 0.366390  | 0.0      | -0.056866 | 0.056866  | 0.003234      | 0.155206             |
| 2010-03-01   | 0.121134  | 0.0      | -0.301809 | 0.301809  | 0.091088      | 2.491537             |
| 2010-04-01   | -0.658208 | 0.0      | -1.079492 | 1.079492  | 1.165302      | 1.640047             |
| 2010-05-01   | -0.178775 | 0.0      | -0.594160 | 0.594160  | 0.353026      | 3.323512             |
| 2010-06-01   | 5.207109  | 0.0      | 4.794954  | 4.794954  | 22.991579     | 0.920847             |
| ...          | ...       | ...      | ...       | ...       | ...           | ...                  |
| 2023-12-01   | -0.459824 | 0.0      | -0.459824 | 0.459824  | 0.211438      | 1.000000             |
| 2024-01-01   | 0.331094  | 0.0      | 0.331094  | 0.331094  | 0.109623      | 1.000000             |
| 2024-02-01   | -0.149089 | 0.0      | -0.149089 | 0.149089  | 0.022228      | 1.000000             |
| 2024-03-01   | -0.615727 | 0.0      | -0.615727 | 0.615727  | 0.379119      | 1.000000             |
| 2024-04-01   | NaN       | 0.0      | NaN       | NaN       | NaN           | NaN                  |

171 rows × 6 columns





## Total\_Exports\_(Customs\_basis)

```
In [88]: df = df_export_ANALYSIS.copy()
name = df.columns[36]
display(f"Component: {name}")

'Component: Total_Exports_(Customs_basis)'
```

```
In [89]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value.pkl', 'rb') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/df_accuracy.pkl")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

'Timeseries of component in % form:'
```

| class      | Total_Exports_(Customs_basis) |
|------------|-------------------------------|
| 1995-02-01 | -0.000100                     |
| 1995-03-01 | 0.298815                      |
| 1995-04-01 | -0.216444                     |
| 1995-05-01 | 0.200359                      |
| 1995-06-01 | 0.029700                      |
| ...        | ...                           |
| 2023-11-01 | -0.004204                     |
| 2023-12-01 | -0.029309                     |
| 2024-01-01 | -0.006216                     |
| 2024-02-01 | 0.032450                      |
| 2024-03-01 | 0.067381                      |

350 rows × 1 columns

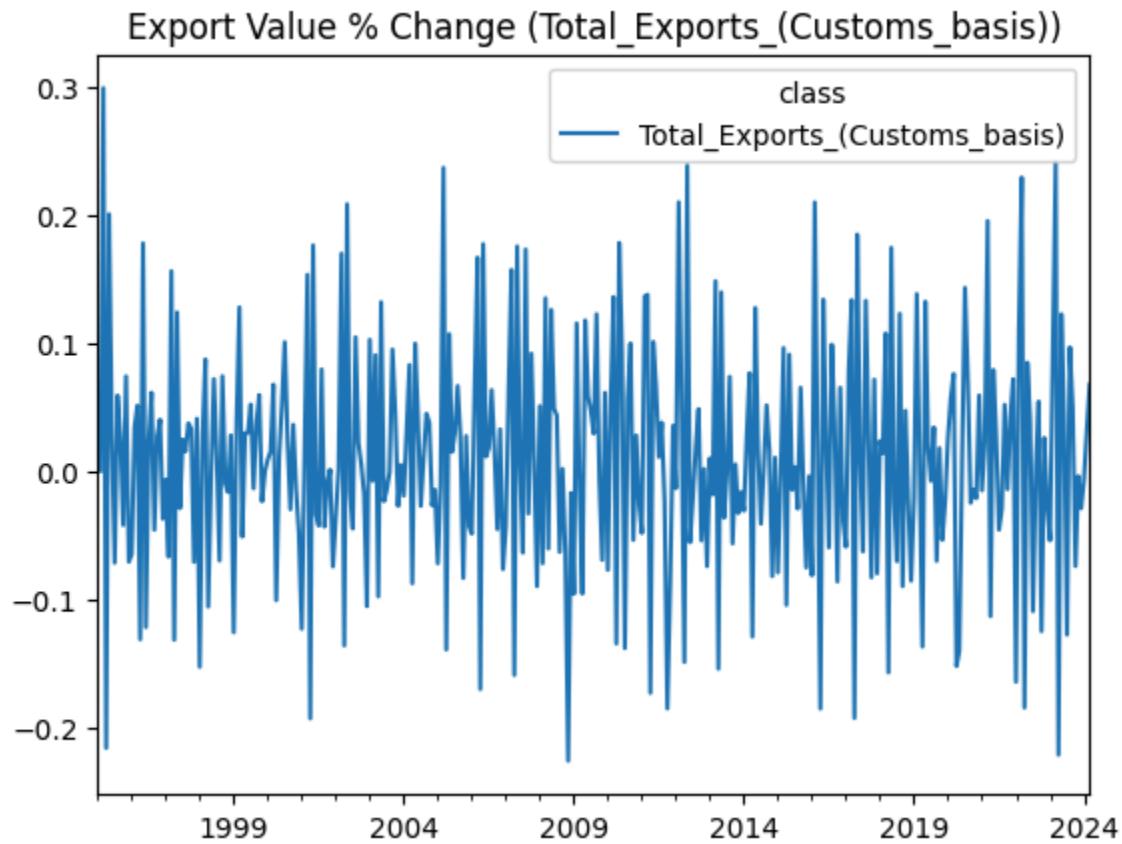
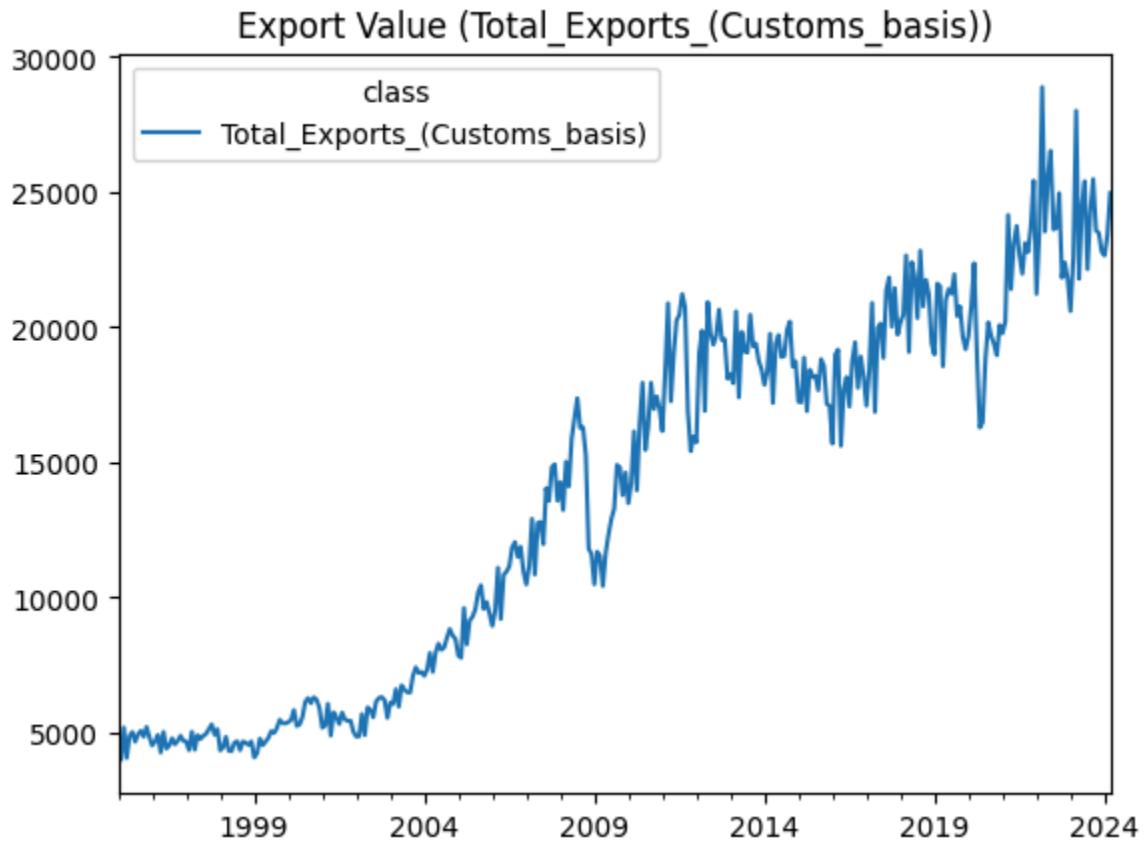
```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -5.462761935574315
p-value: 2.497258809301555e-06
Critical Value 1%: -3.449846029628477
Critical Value 5%: -2.870129281376164
Critical Value 10%: -2.5713460670144603
is_stationary: True
```

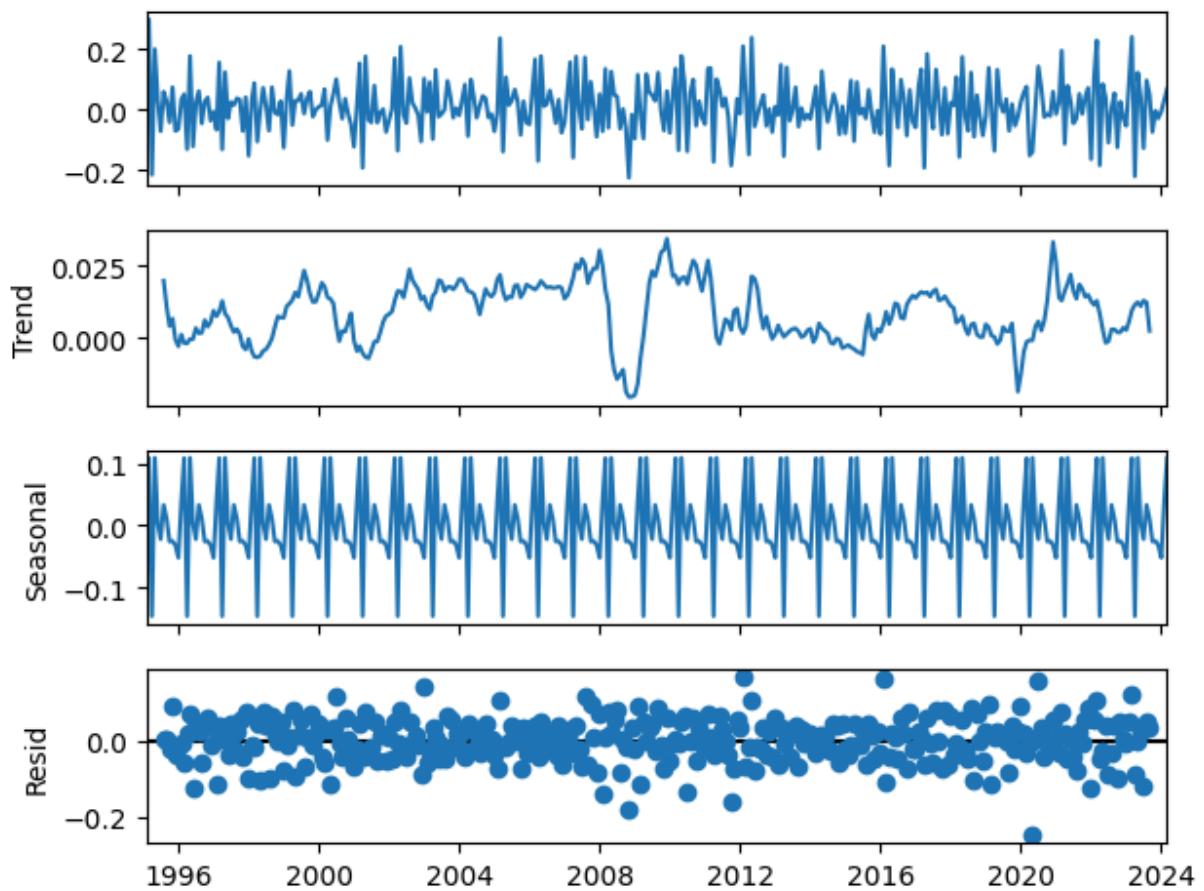
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.05742973139965504
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

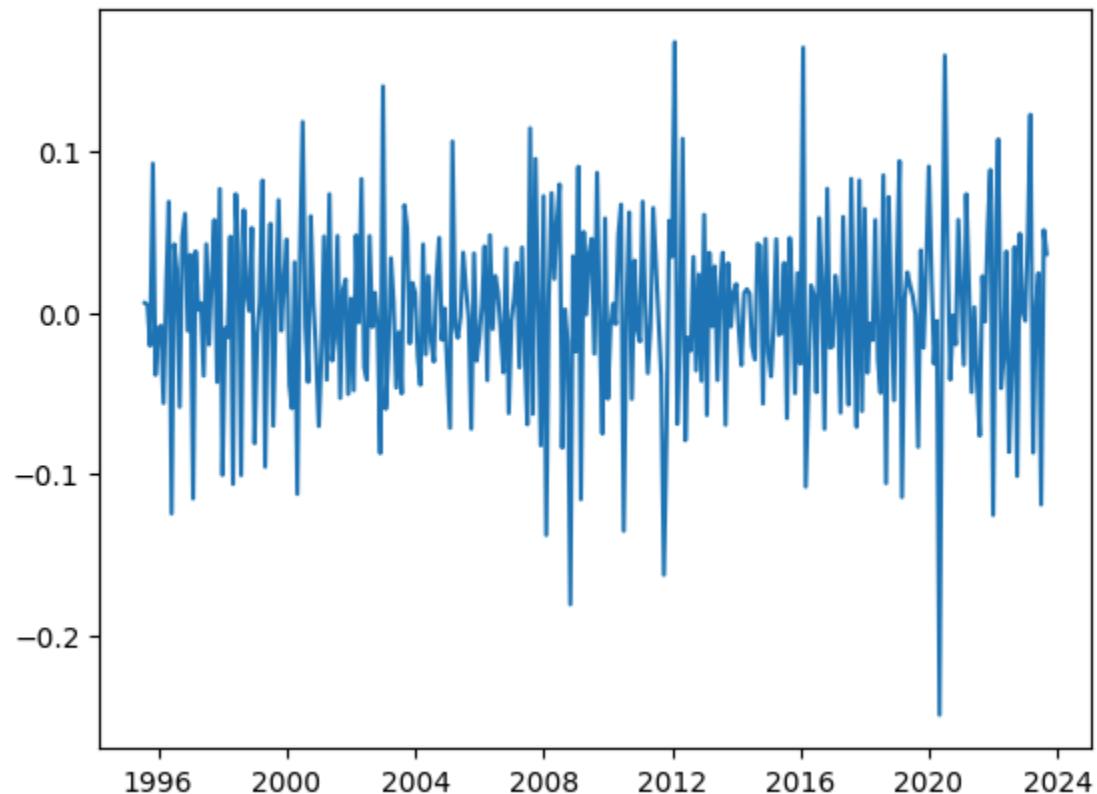
```
result = kpss(series, regression='c')
```

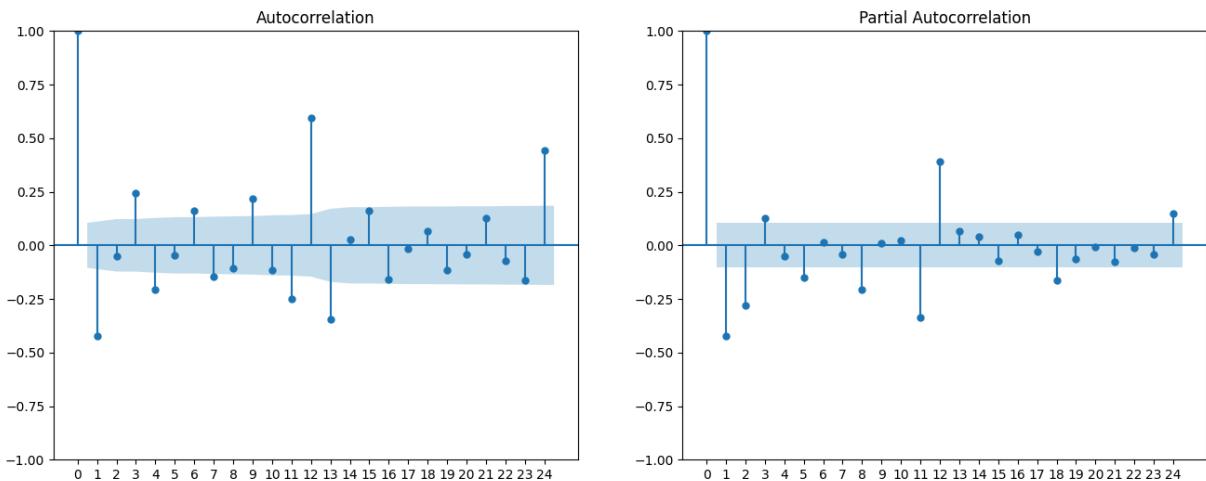




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

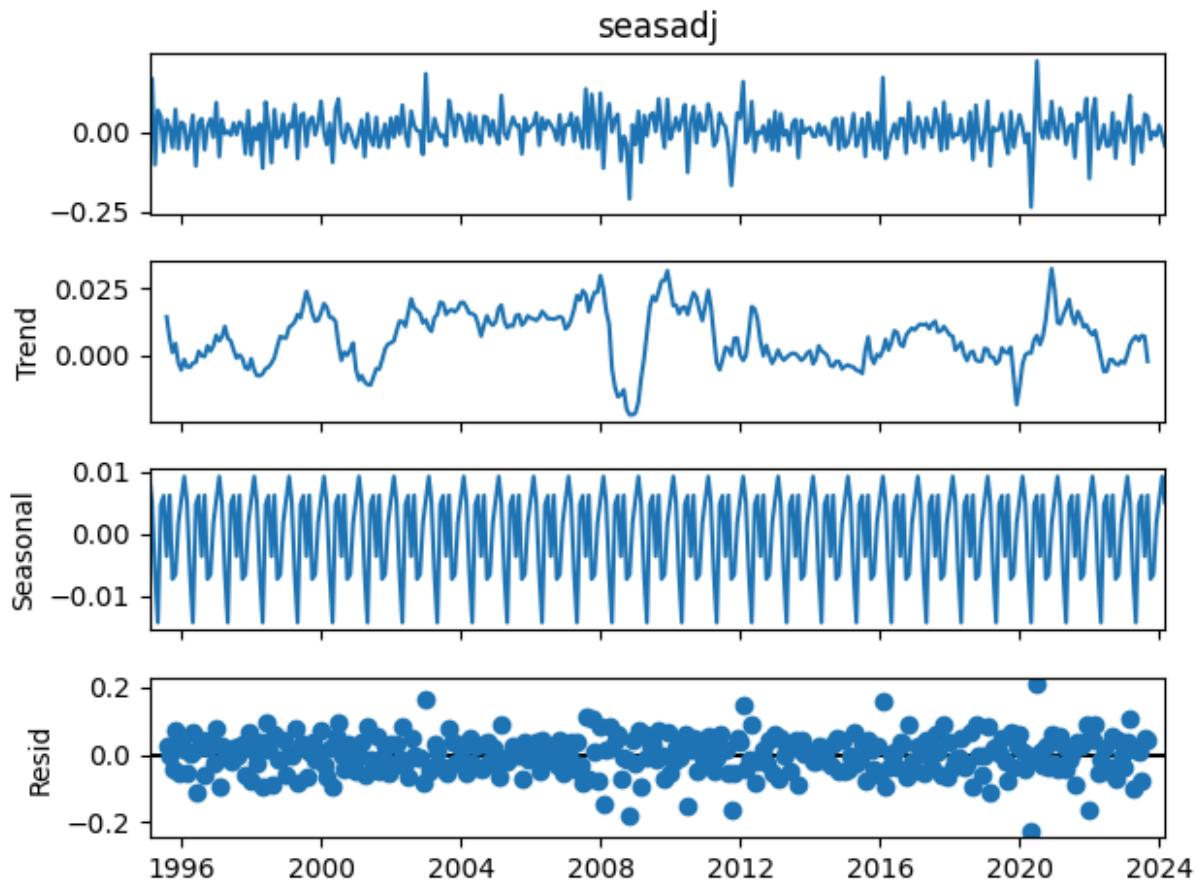
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted): '
Running ADF Test for Stationarity...
ADF Statistic: -6.164918773971001
p-value: 7.027775472688554e-08
Critical Value 1%: -3.4497304638968043
Critical Value 5%: -2.8700785273763487
Critical Value 10%: -2.571319005190311
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.12172401106495867
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

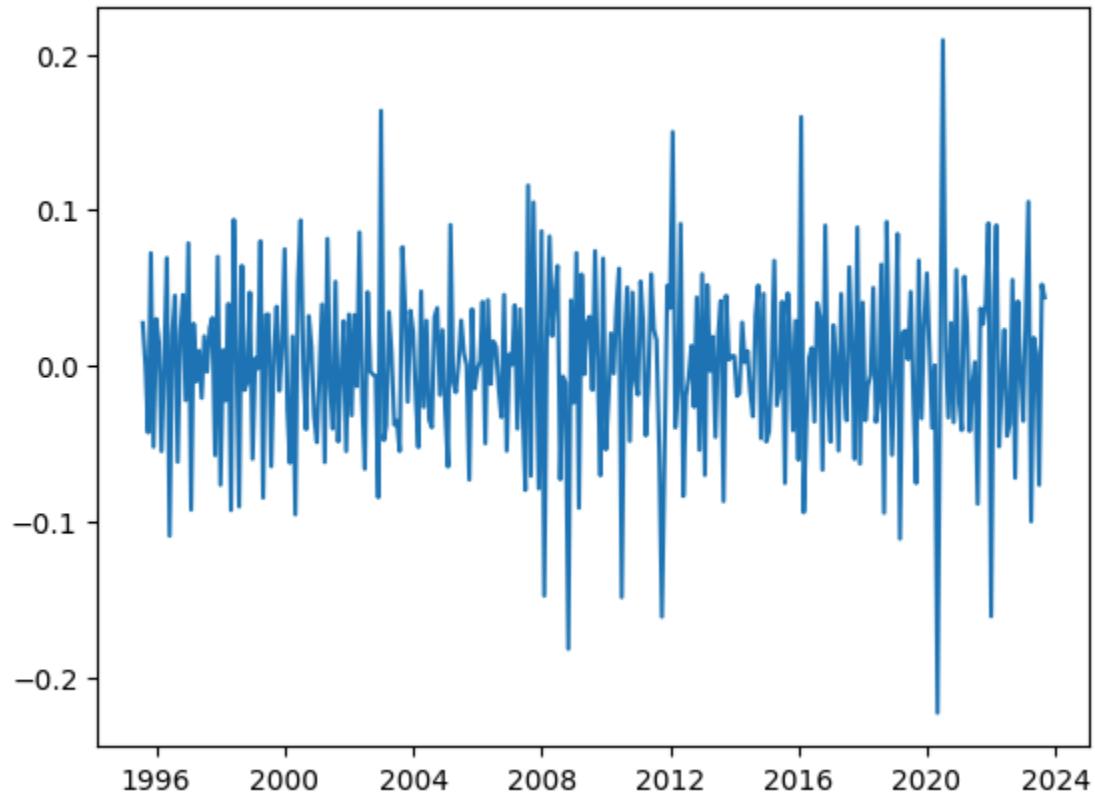
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

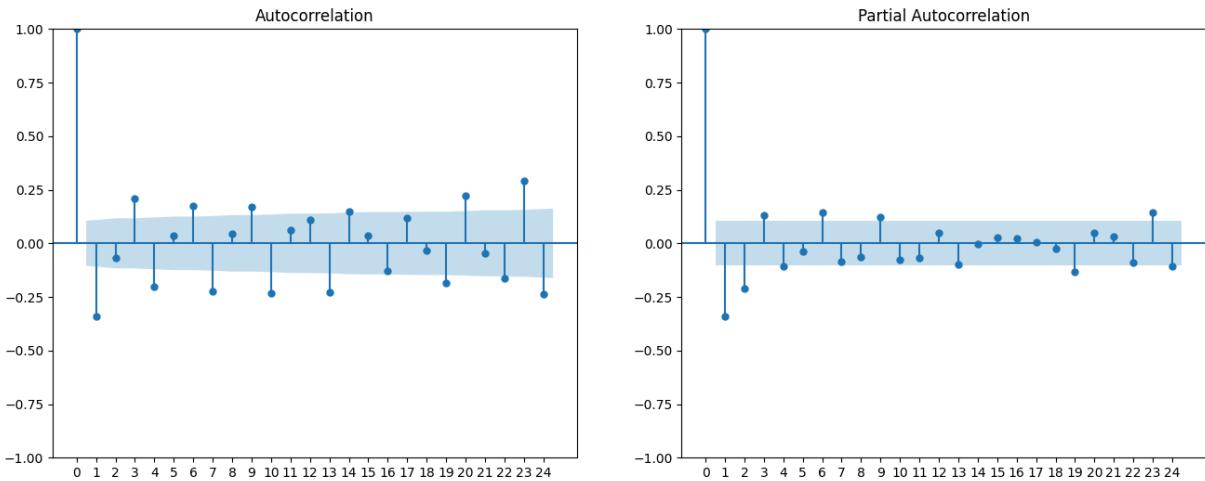
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

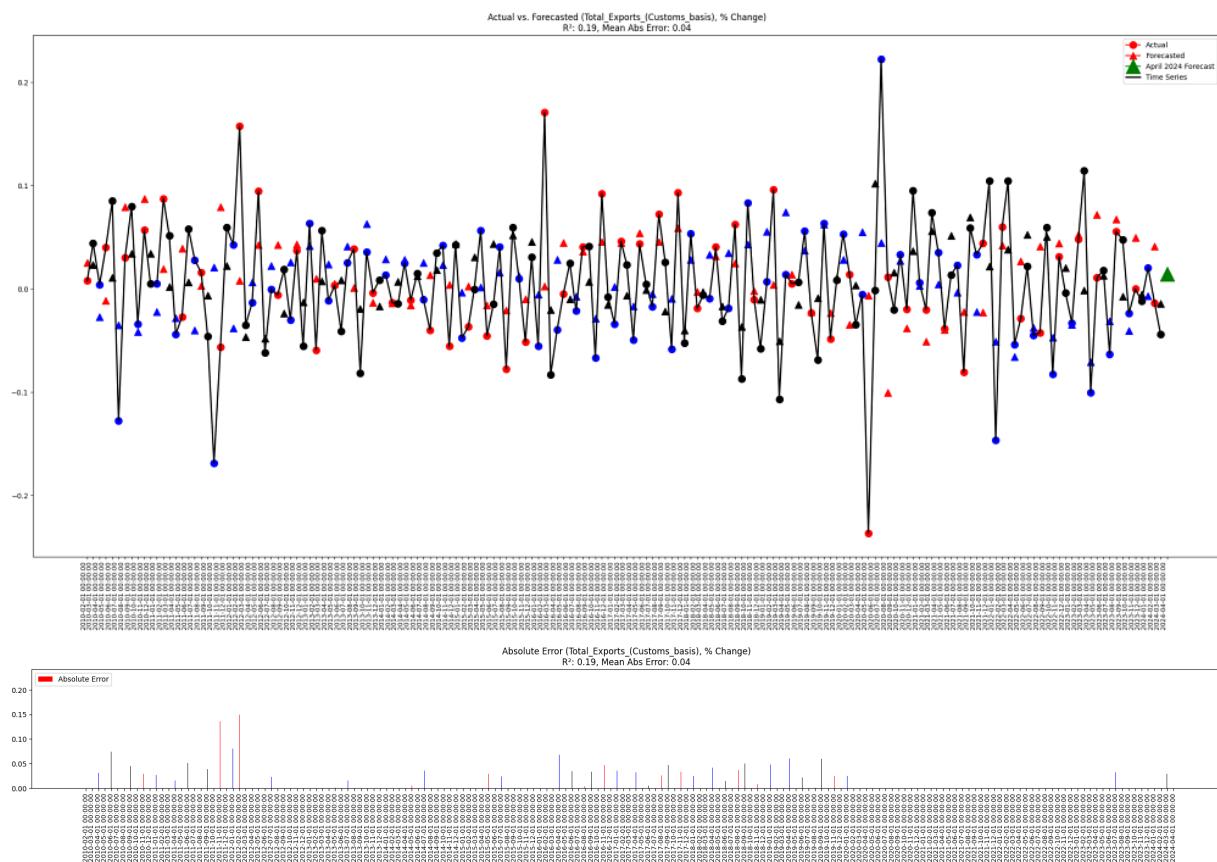
'Data of Actual vs Forecasted values with error metrics:'

| Total_Exports_(Customs_basis) | actual    | forecast  | error     | abs_error | squared_error | ab  |
|-------------------------------|-----------|-----------|-----------|-----------|---------------|-----|
| 2010-02-01                    | 0.007756  | 0.025254  | -0.017497 | 0.017497  | 0.000306      |     |
| 2010-03-01                    | 0.044331  | 0.023303  | 0.021028  | 0.021028  | 0.000442      |     |
| 2010-04-01                    | 0.004051  | -0.027061 | 0.031113  | 0.031113  | 0.000968      |     |
| 2010-05-01                    | 0.039930  | -0.011500 | 0.051430  | 0.051430  | 0.002645      |     |
| 2010-06-01                    | 0.084999  | 0.010934  | 0.074065  | 0.074065  | 0.005486      |     |
| ...                           | ...       | ...       | ...       | ...       | ...           | ... |
| 2023-12-01                    | -0.011903 | -0.005668 | -0.006236 | 0.006236  | 0.000039      |     |
| 2024-01-01                    | 0.020506  | -0.006954 | 0.027460  | 0.027460  | 0.000754      |     |
| 2024-02-01                    | -0.013592 | 0.041115  | -0.054708 | 0.054708  | 0.002993      |     |
| 2024-03-01                    | -0.044153 | -0.014255 | -0.029898 | 0.029898  | 0.000894      |     |
| 2024-04-01                    | NaN       | 0.014593  | NaN       | NaN       | NaN           |     |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.03944135928483341'



## Coverage\_Adjustment

```
In [90]: df = df_export_ANALYSIS.copy()
name = df.columns[37]
display(f"Component: {name}")
```

'Component: Coverage\_Adjustment'

```
In [91]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value'):
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment")
check_time_series_assumptions(adj_ts, freq=12)
```

```
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined")
df_accuracy.columns.name = name
df_accuracy["forecast"] = cap_extreme_values(df_accuracy[["forecast"]])["forecast"]
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)
```

'Timeseries of component in % form:'

## class Coverage\_Adjustment

|            |           |
|------------|-----------|
| 1995-02-01 | -0.316990 |
| 1995-03-01 | 1.296899  |
| 1995-04-01 | -0.425928 |
| 1995-05-01 | 0.503618  |
| 1995-06-01 | -0.098994 |
| ...        | ...       |
| 2023-11-01 | 0.604792  |
| 2023-12-01 | 0.082835  |
| 2024-01-01 | 0.550565  |
| 2024-02-01 | -0.453620 |
| 2024-03-01 | 0.196033  |

350 rows x 1 columns

'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'

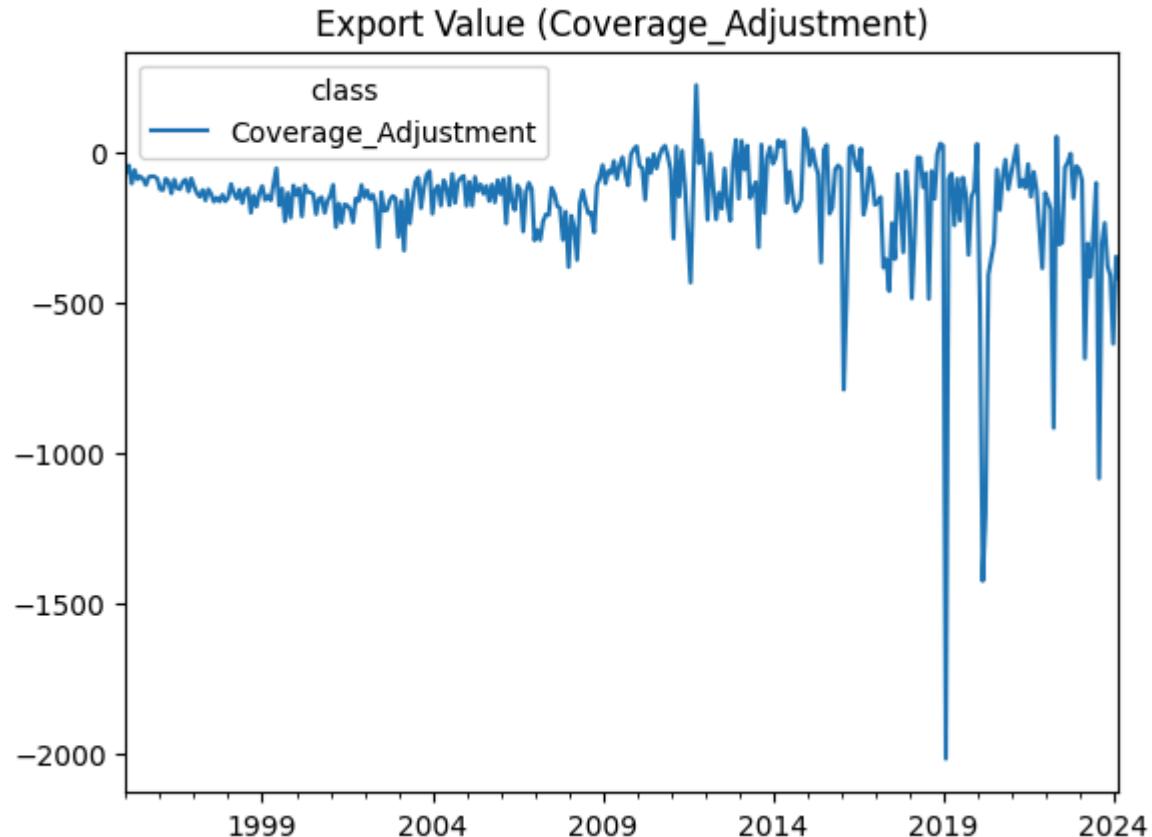
```
Running ADF Test for Stationarity...
ADF Statistic: -18.61738404728031
p-value: 2.062661680538337e-30
Critical Value 1%: -3.449226932880019
Critical Value 5%: -2.869857365438656
Critical Value 10%: -2.571201085130664
is_stationary: True
```

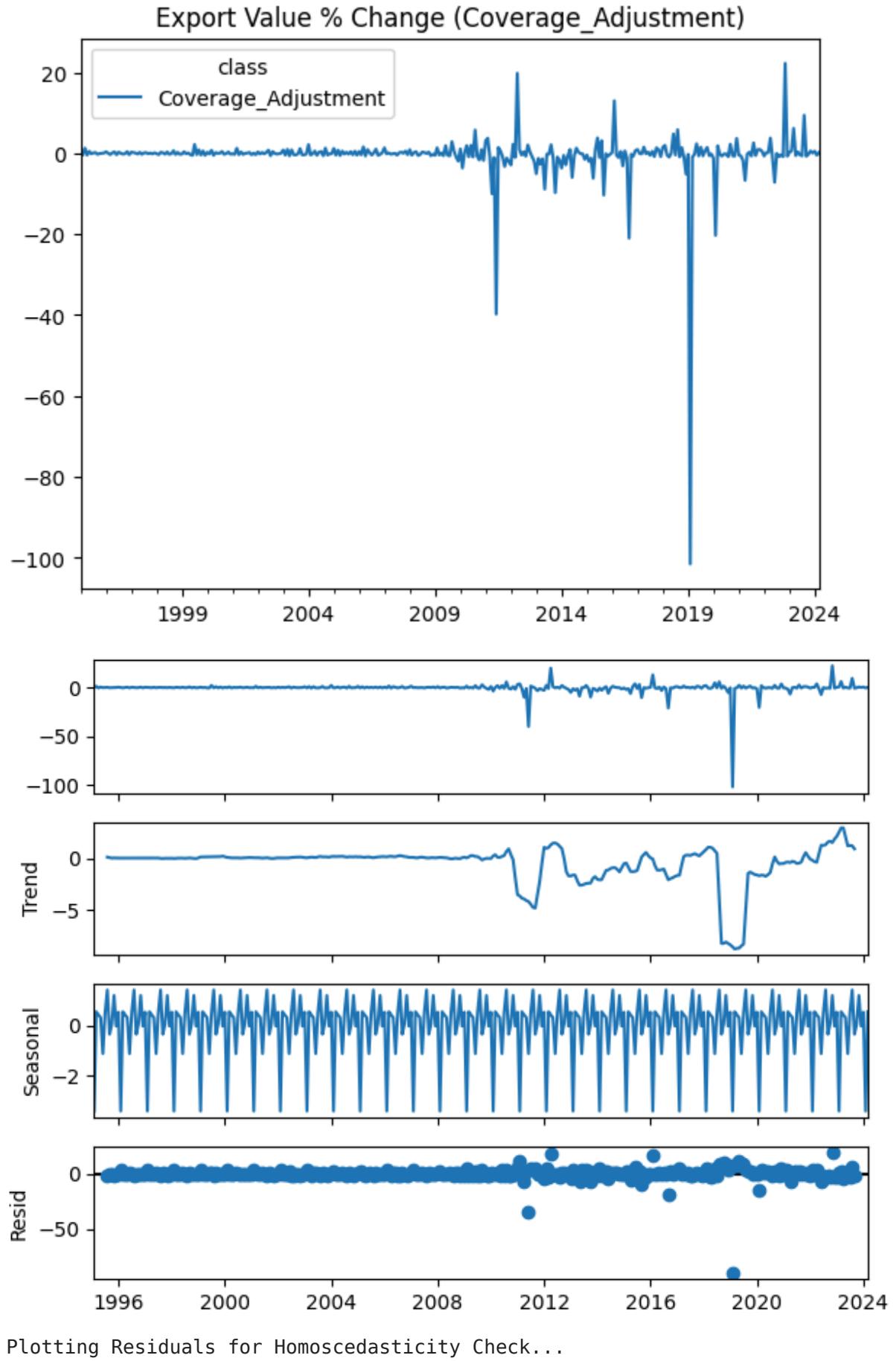
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.2308992478719636
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

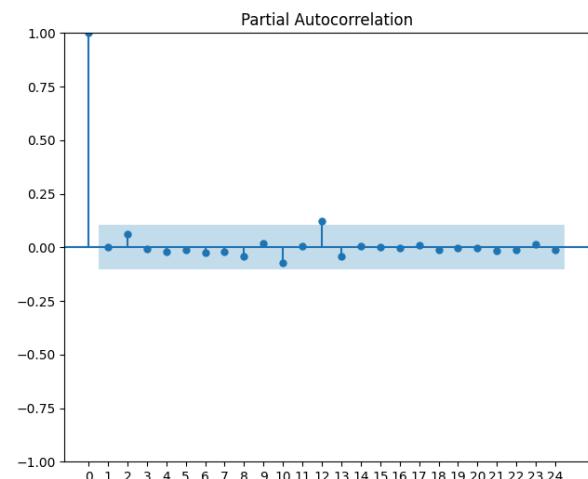
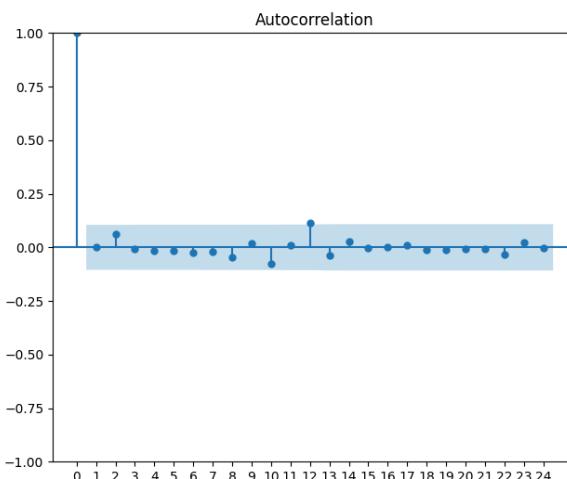
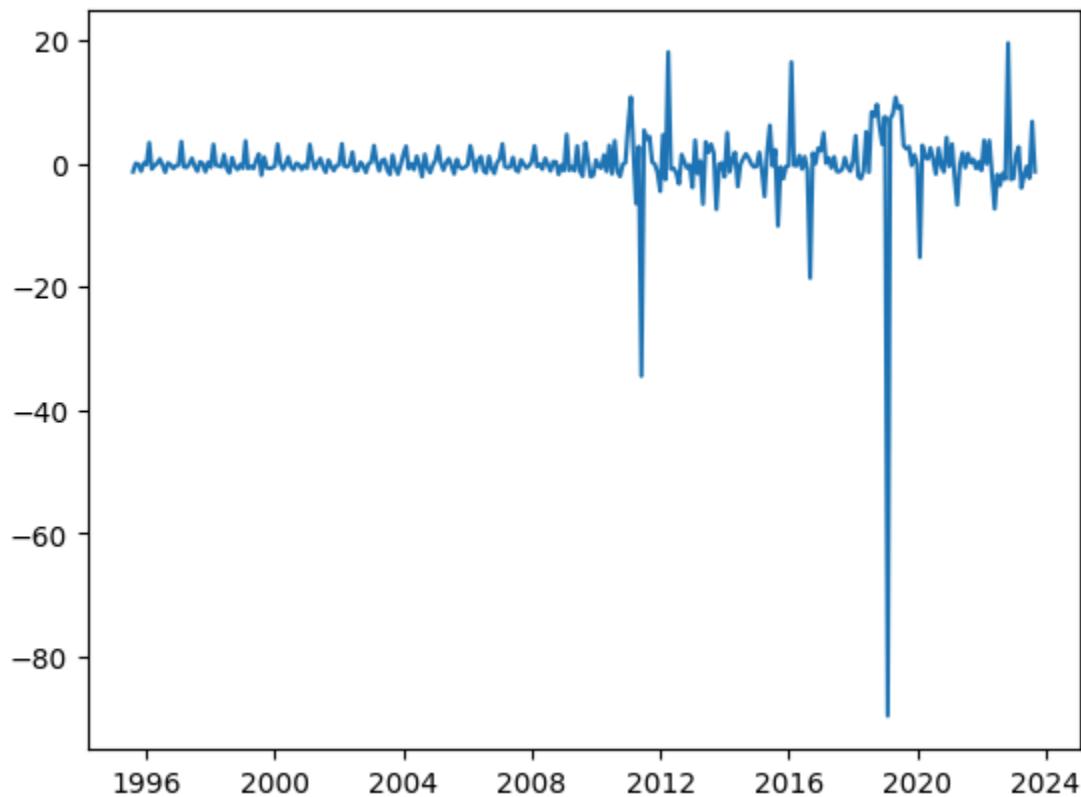
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
result = kpss(series, regression='c')
```





### Residuals



Time series assumptions are met.

True

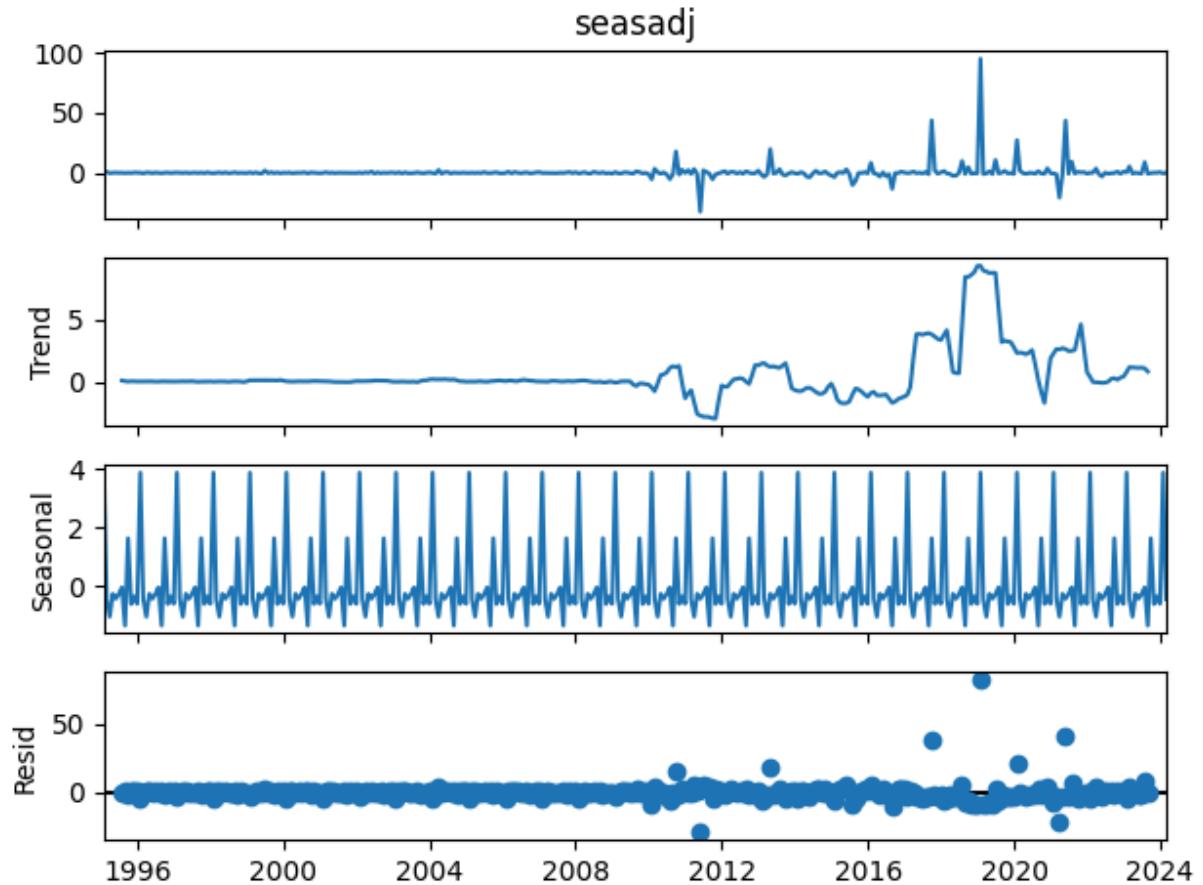
'-----'

'Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted):'

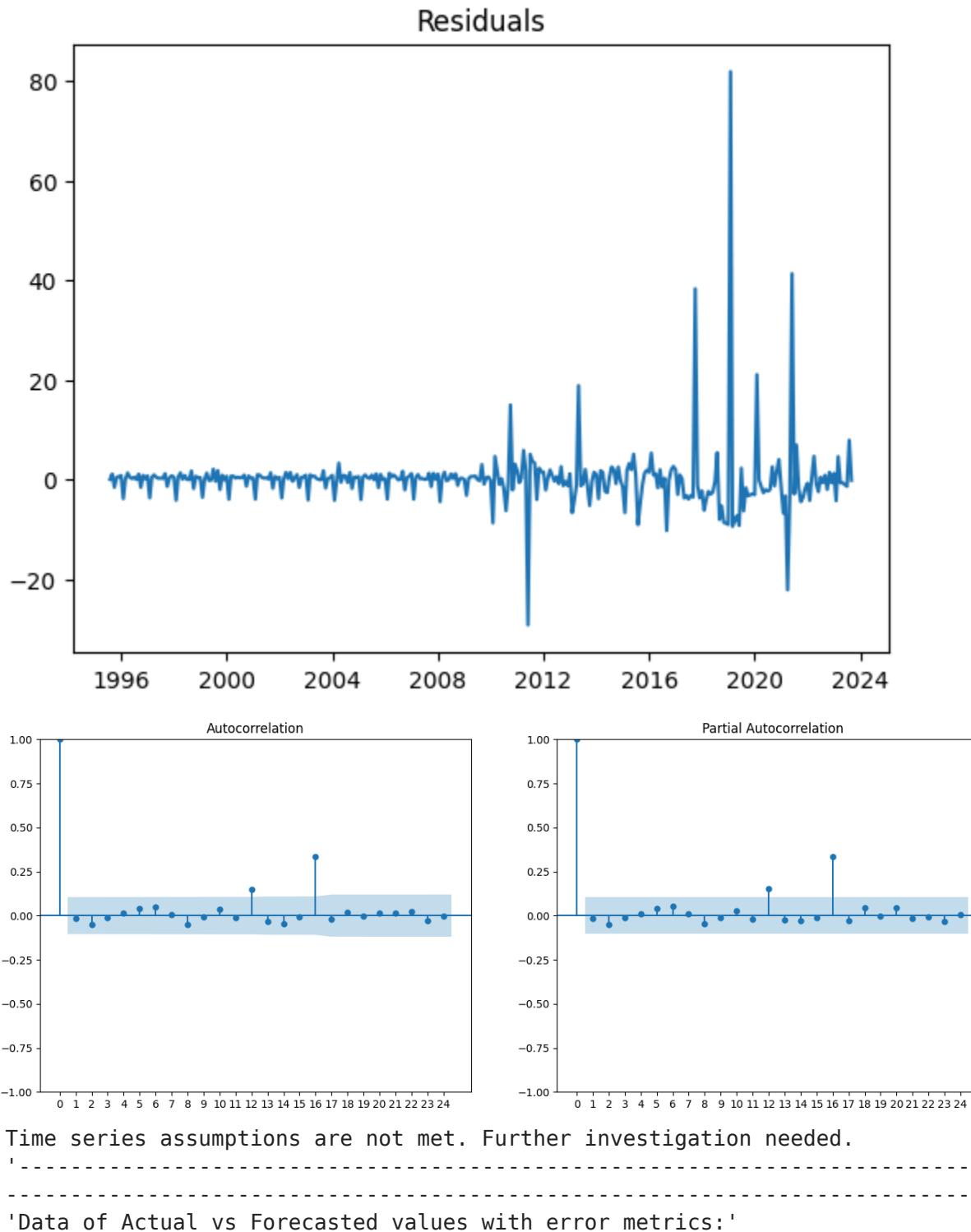
Running ADF Test for Stationarity...  
ADF Statistic: -2.942814871532678  
p-value: 0.040590954217201825  
Critical Value 1%: -3.450081345901191  
Critical Value 5%: -2.870232621465807  
Critical Value 10%: -2.571401168202517  
is\_stationary: True

Running KPSS Test for Stationarity...  
KPSS Statistic: 0.48846664309019094  
p-value: 0.044264269574281326  
Critical Value 10%: 0.347  
Critical Value 5%: 0.463  
Critical Value 2.5%: 0.574  
Critical Value 1%: 0.739  
is\_stationary: False

Decomposing the Series...

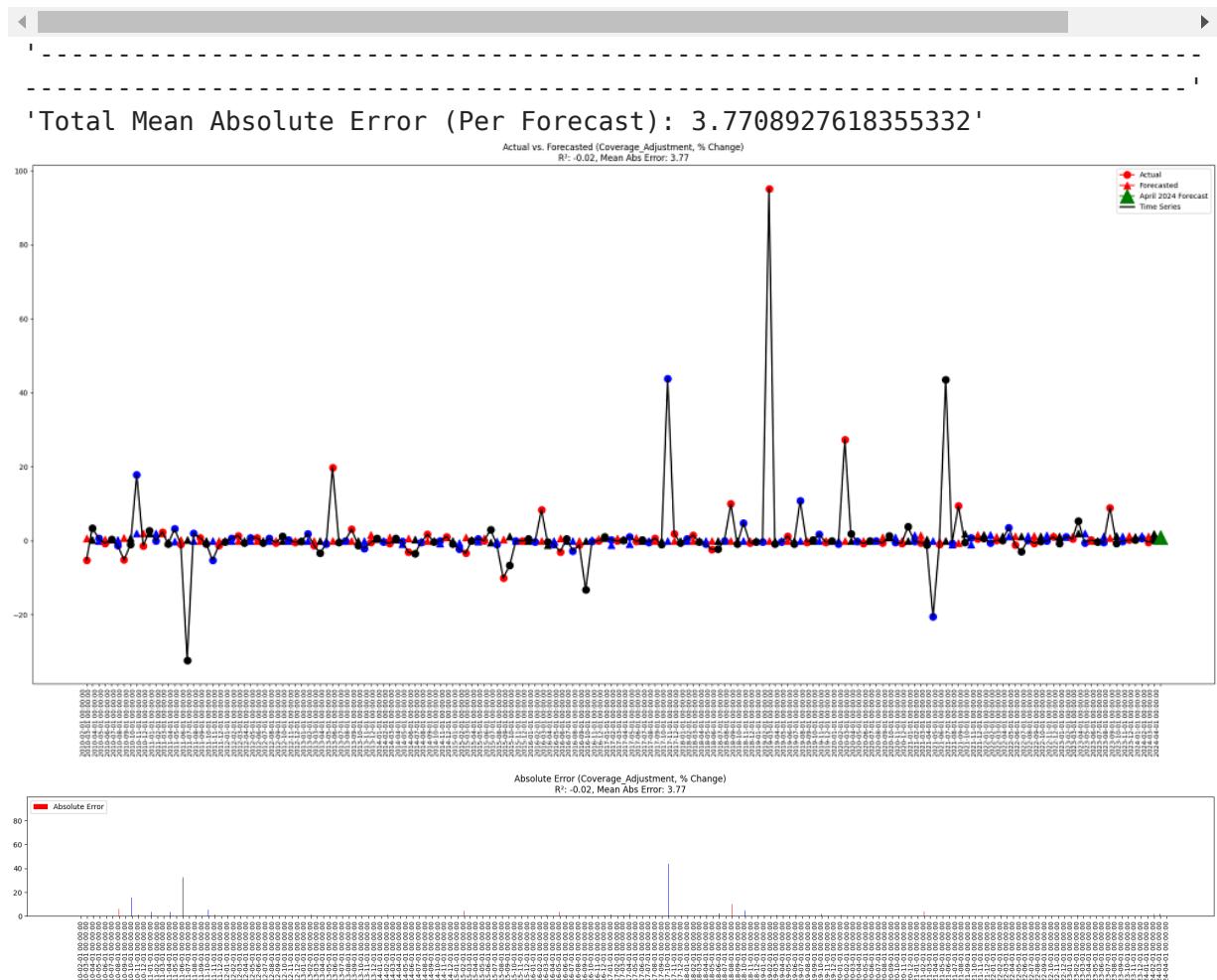


Plotting Residuals for Homoscedasticity Check...



| Coverage_Adjustment | actual    | forecast  | error     | abs_error | squared_error | abs_percen |
|---------------------|-----------|-----------|-----------|-----------|---------------|------------|
| 2010-02-01          | -5.297214 | 0.604674  | -5.901888 | 5.901888  | 34.832284     |            |
| 2010-03-01          | 3.429791  | 0.251969  | 3.177822  | 3.177822  | 10.098553     |            |
| 2010-04-01          | 0.573724  | -0.108074 | 0.681798  | 0.681798  | 0.464849      |            |
| 2010-05-01          | -0.732752 | 0.298221  | -1.030973 | 1.030973  | 1.062904      |            |
| 2010-06-01          | 0.362550  | 0.345022  | 0.017527  | 0.017527  | 0.000307      |            |
| ...                 | ...       | ...       | ...       | ...       | ...           | ...        |
| 2023-12-01          | 0.152881  | 1.084302  | -0.931421 | 0.931421  | 0.867544      |            |
| 2024-01-01          | 0.615977  | 1.339454  | -0.723478 | 0.723478  | 0.523420      |            |
| 2024-02-01          | -0.451317 | 1.213539  | -1.664857 | 1.664857  | 2.771748      |            |
| 2024-03-01          | 0.135495  | 1.917972  | -1.782477 | 1.782477  | 3.177223      |            |
| 2024-04-01          | NaN       | 0.871509  | NaN       | NaN       | NaN           | NaN        |

171 rows × 6 columns



**Timing\_Adjustment** has too few significant AR points with random shocks, would be better to omit and no-change at 0

## Exports,f.o.b.(BOP\_basis)

```
In [92]: df = df_export_ANALYSIS.copy()
name = df.columns[39]
display(f"Component: {name}")

'Component: Exports,_f.o.b._(BOP_basis)'

In [93]: df = df_export_ANALYSIS[[name]]

df.plot(title=f"Export Value ({name})", label=f"{name}")
df.pct_change().plot(title=f"Export Value % Change ({name})", label=f"{name}")
ts = df.pct_change().dropna()
display("Timeseries of component in % form:")
display(ts)
display("-" * 150)

### check if the data timeseries & statistical assumptions validity (stationarity)
# freq 12 is used since the data is monthly, so the seasonality is 12 months
display("Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjustment)")
display(check_time_series_assumptions(ts, freq=12))
display("-" * 150)

#read the pickle file
with open(f'data/cleaned/deseasonalised_x13/update/dict_deseasonalized_value') as handle:
    dict_deseasonalized_value = pickle.load(handle)[name]
adj_ts = dict_deseasonalized_value['seasadj'].pct_change().dropna()
display("Statistical Assumptions & Properties Analysis (POST-Seasonality Adjustment)")
check_time_series_assumptions(adj_ts, freq=12)
display("-" * 150)

df_accuracy = pd.read_pickle(f"data/cleaned/SARIMA_RollWalkForward/combined/")
df_accuracy.columns.name = name
#Forecast are all Out-Of-Sample on a rolling basis
display("Data of Actual vs Forecasted values with error metrics:")
display(df_accuracy)
display("-" * 150)

#Plot Actual vs Forecasted values with error metrics
average_abs_error = df_accuracy["abs_error"].mean()
display(f"Total Mean Absolute Error (Per Forecast): {average_abs_error}")
plot_actual_vs_forecast(df_accuracy, name)

'Timeseries of component in % form:'
```

| class      | Exports,_f.o.b._(BOP_basis) |
|------------|-----------------------------|
| 1995-02-01 | 0.005346                    |
| 1995-03-01 | 0.287161                    |
| 1995-04-01 | -0.212079                   |
| 1995-05-01 | 0.195755                    |
| 1995-06-01 | 0.032157                    |
| ...        | ...                         |
| 2023-11-01 | -0.010378                   |
| 2023-12-01 | -0.031152                   |
| 2024-01-01 | -0.016448                   |
| 2024-02-01 | 0.046531                    |
| 2024-03-01 | 0.065435                    |

350 rows × 1 columns

```
'-----'
-----'
'Statistical Assumptions & Properties Analysis (Pre-Seasonality Adjusted):'
Running ADF Test for Stationarity...
ADF Statistic: -4.878580924094646
p-value: 3.8381529130956075e-05
Critical Value 1%: -3.4502011472639724
Critical Value 5%: -2.8702852297358983
Critical Value 10%: -2.5714292194077513
is_stationary: True
```

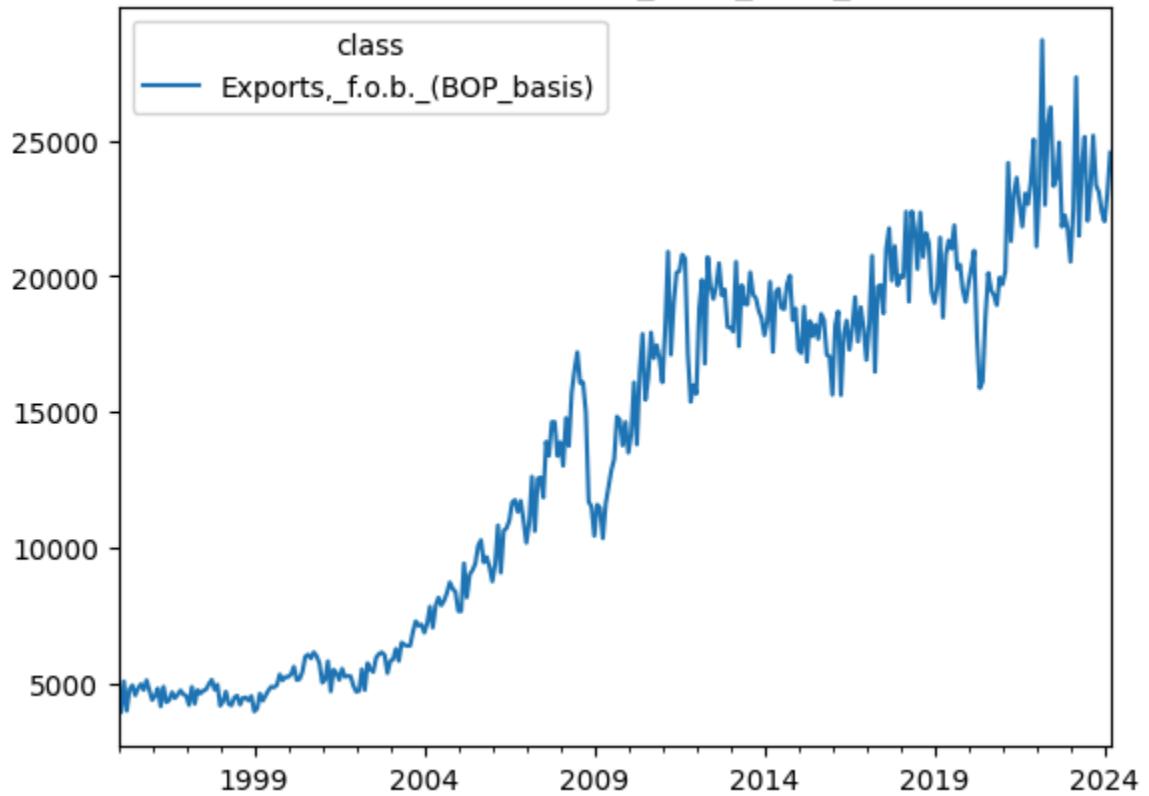
```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.06865483610896644
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

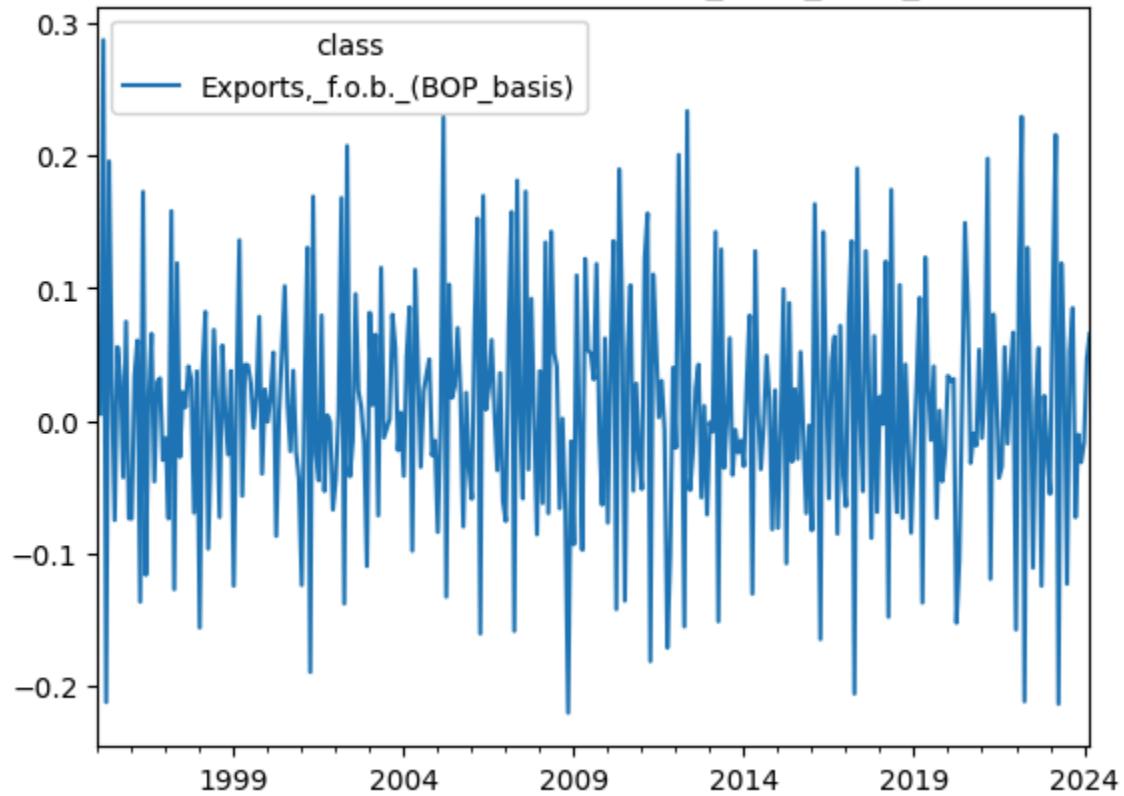
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

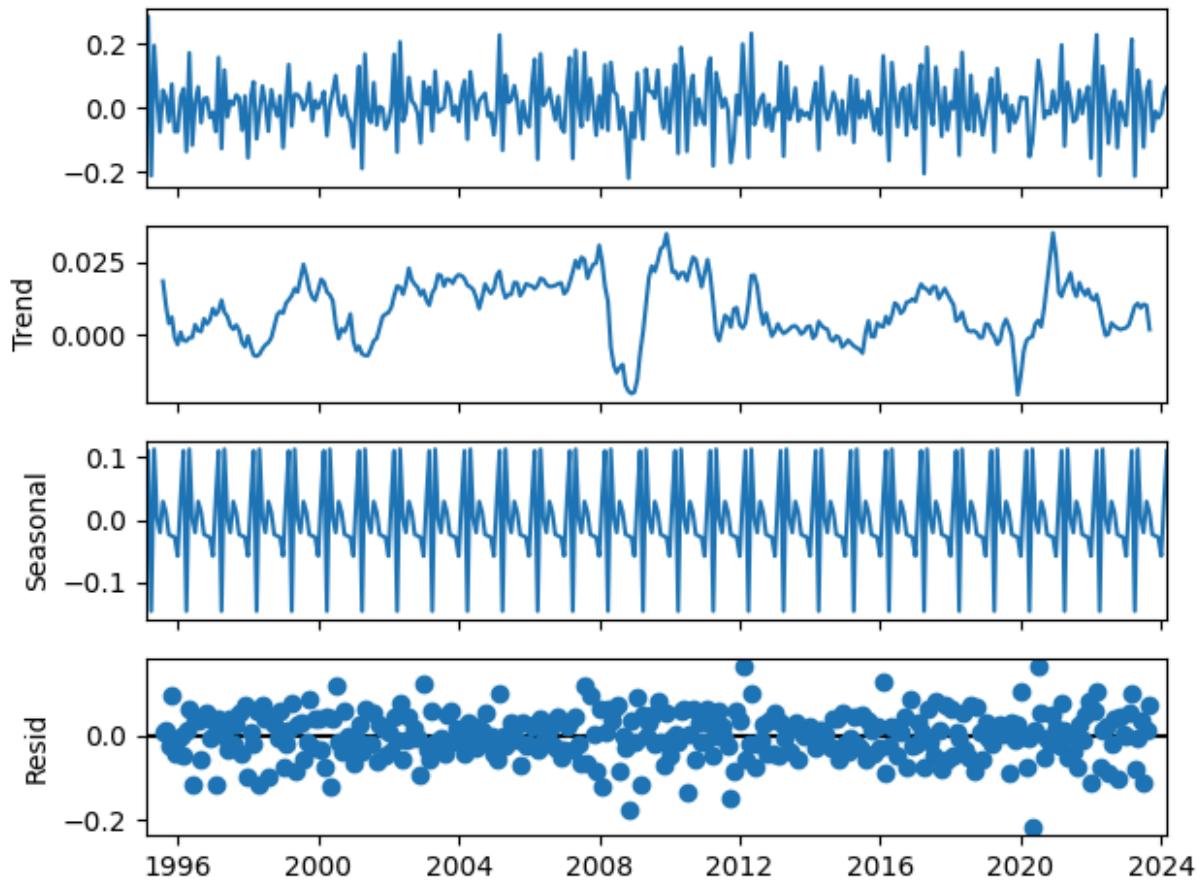
```
result = kpss(series, regression='c')
```

### Export Value (Exports,\_f.o.b.\_(BOP\_basis))



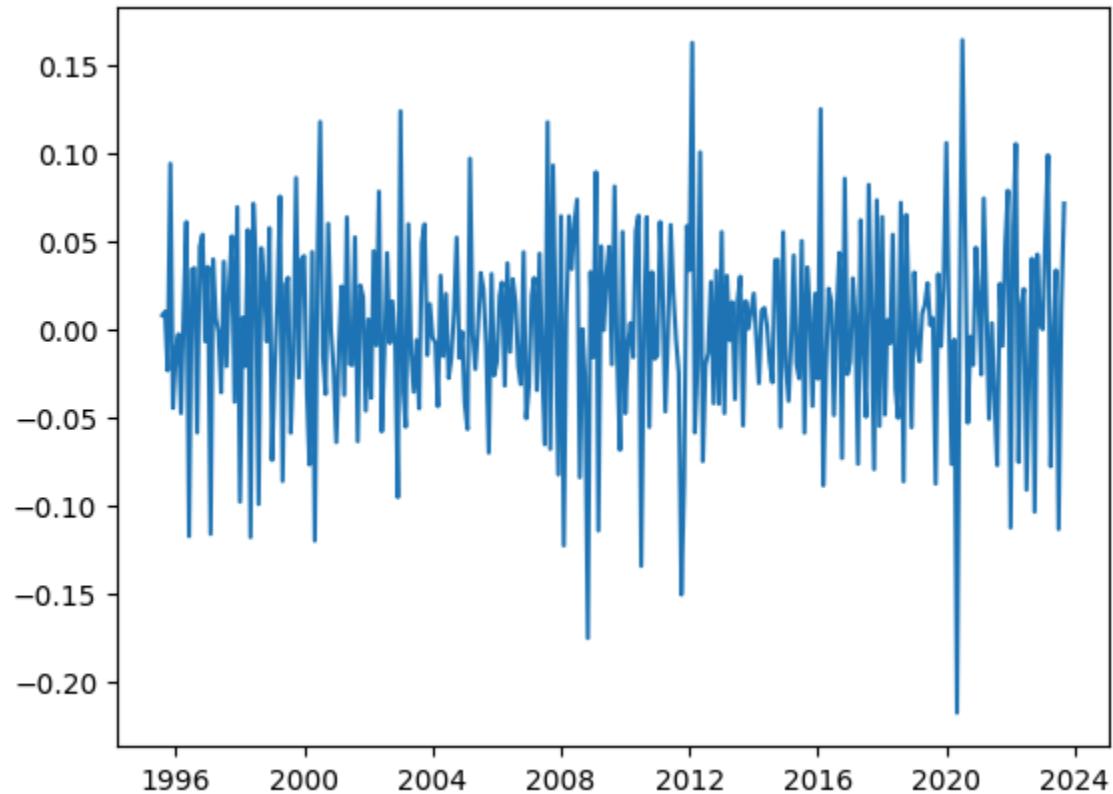
### Export Value % Change (Exports,\_f.o.b.\_(BOP\_basis))

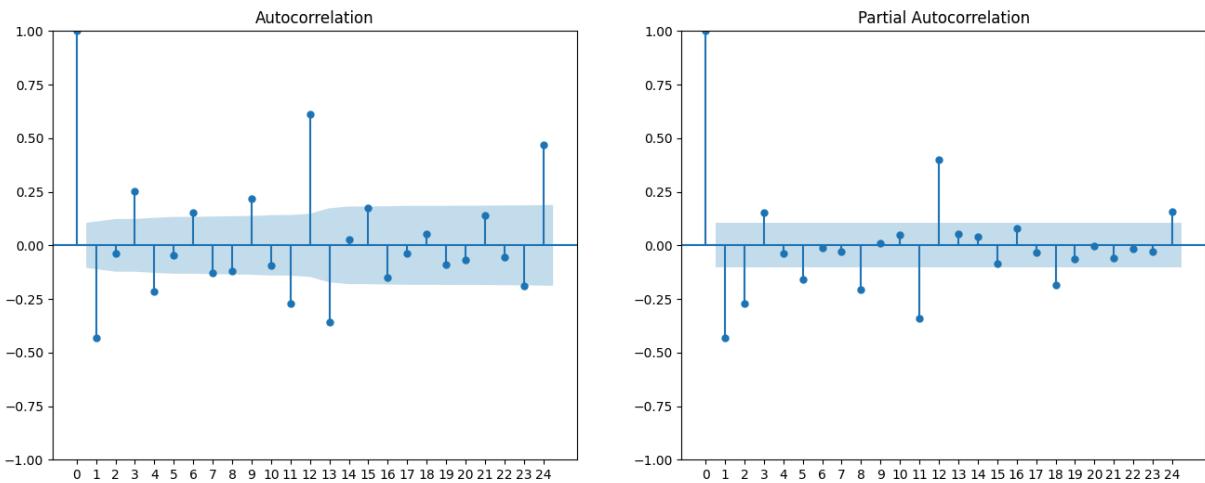




Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

True

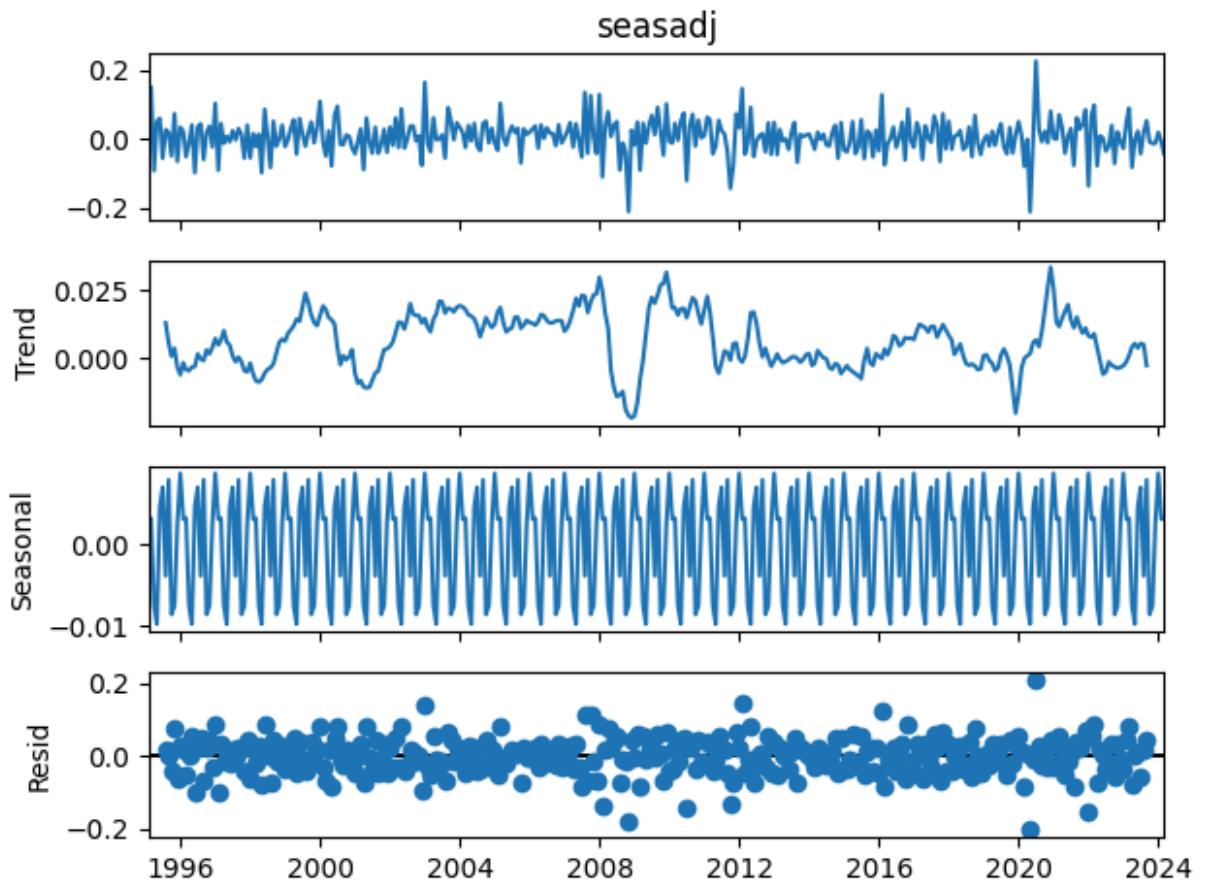
```
'-----
'----- Statistical Assumptions & Properties Analysis (POST-Seasonality Adjusted): '
Running ADF Test for Stationarity...
ADF Statistic: -7.1806422891222725
p-value: 2.655476784300986e-10
Critical Value 1%: -3.4495033946549123
Critical Value 5%: -2.8699787979596136
Critical Value 10%: -2.5712658305164955
is_stationary: True
```

```
Running KPSS Test for Stationarity...
KPSS Statistic: 0.1468459558118012
p-value: 0.1
Critical Value 10%: 0.347
Critical Value 5%: 0.463
Critical Value 2.5%: 0.574
Critical Value 1%: 0.739
is_stationary: True
```

Decomposing the Series...

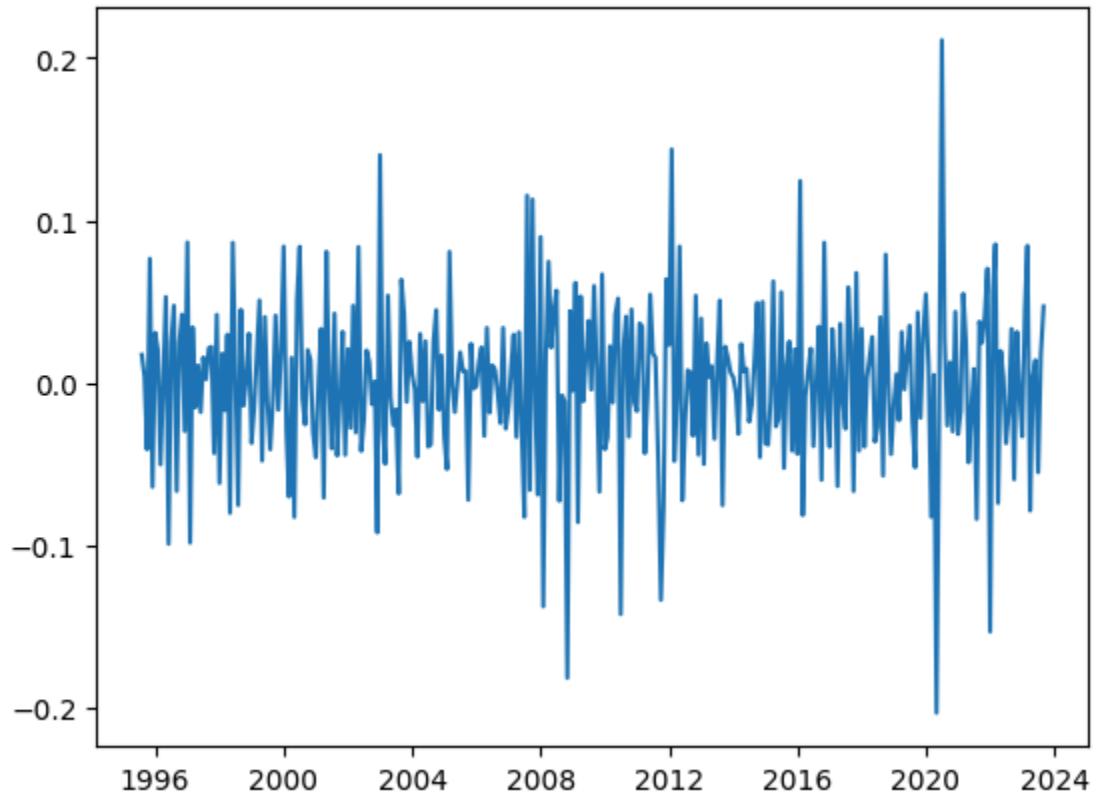
```
/home/wheelfredie/scripts/BoT_Exports/helper.py:61: InterpolationWarning: Th
e test statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

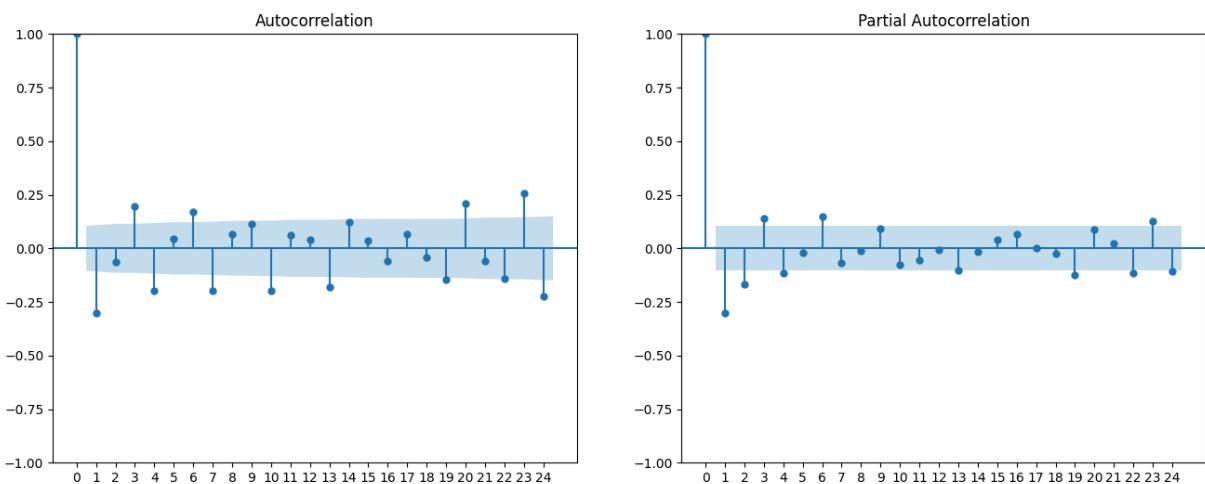
```
result = kpss(series, regression='c')
```



Plotting Residuals for Homoscedasticity Check...

Residuals





Time series assumptions are met.

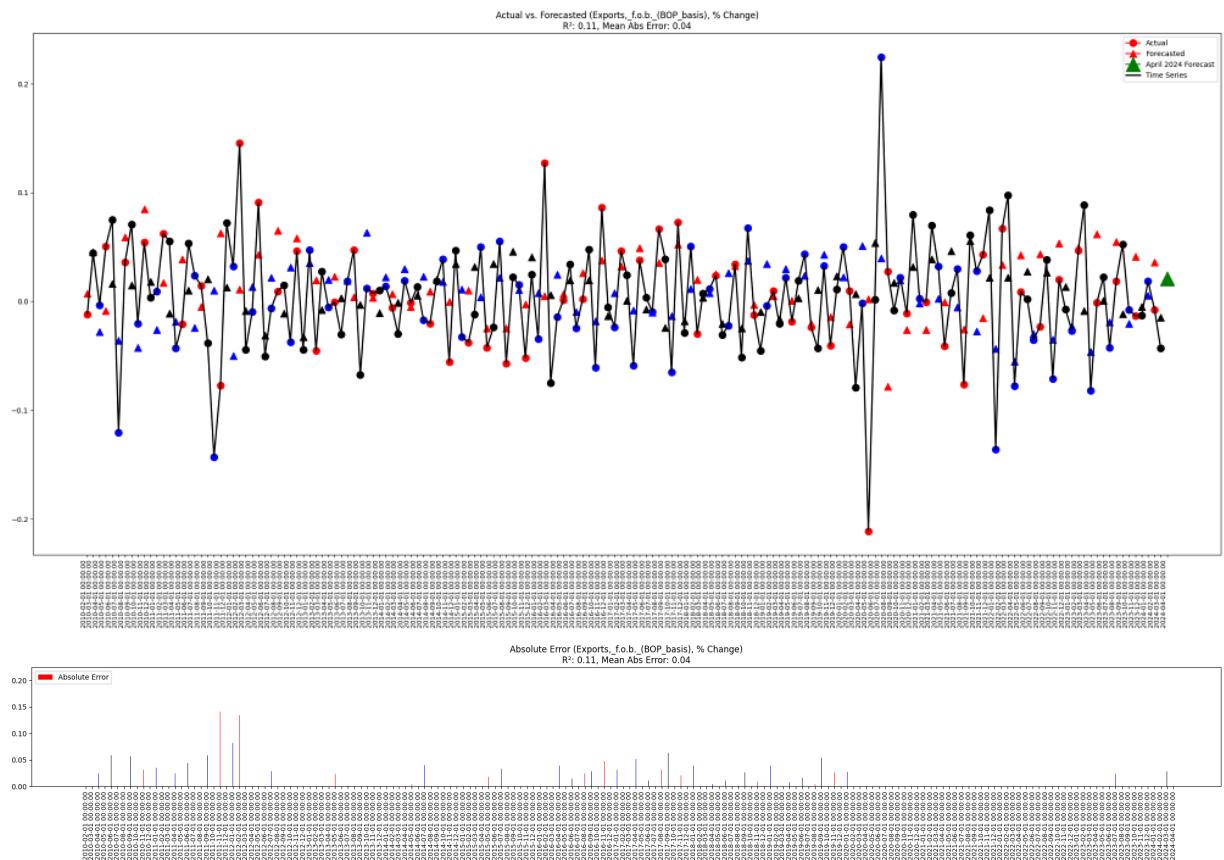
'Data of Actual vs Forecasted values with error metrics:'

| Exports,_F.o.b._(BOP_basis) | actual    | forecast  | error     | abs_error | squared_error | abs_pe |
|-----------------------------|-----------|-----------|-----------|-----------|---------------|--------|
| 2010-02-01                  | -0.012069 | 0.007016  | -0.019085 | 0.019085  | 0.000364      |        |
| 2010-03-01                  | 0.044460  | 0.045477  | -0.001017 | 0.001017  | 0.000001      |        |
| 2010-04-01                  | -0.003778 | -0.028084 | 0.024306  | 0.024306  | 0.000591      |        |
| 2010-05-01                  | 0.050561  | -0.008684 | 0.059245  | 0.059245  | 0.003510      |        |
| 2010-06-01                  | 0.074909  | 0.016149  | 0.058760  | 0.058760  | 0.003453      |        |
| ...                         | ...       | ...       | ...       | ...       | ...           | ...    |
| 2023-12-01                  | -0.013067 | -0.004826 | -0.008241 | 0.008241  | 0.000068      |        |
| 2024-01-01                  | 0.018430  | 0.005392  | 0.013038  | 0.013038  | 0.000170      |        |
| 2024-02-01                  | -0.007643 | 0.036119  | -0.043762 | 0.043762  | 0.001915      |        |
| 2024-03-01                  | -0.042899 | -0.015008 | -0.027891 | 0.027891  | 0.000778      |        |
| 2024-04-01                  | NaN       | 0.020950  | NaN       | NaN       | NaN           |        |

171 rows × 6 columns



'Total Mean Absolute Error (Per Forecast): 0.036641612307191554'



**Upcoming Notebook**  
**2\_composite\_weight\_analysis.ipynb:**  
 check if weights change drastically across months and Out of Sample forecast of next month(t) using last release(t-1,...,t-ar\_lag)

**Upcoming Notebook**  
**3\_stitching\_series.ipynb:**  
 Restitching the Components into the Total Export Series(BoP) with Error Optimisation