

```
In [1]: import pandas as pd
import datetime
import numpy as np
import os
import sys

import statsmodels.formula.api as smf
import statsmodels.tsa.api as smt
import statsmodels.api as sm
import scipy.stats as scs
from statsmodels.stats.diagnostic import acorr_ljungbox # white noise test
from statsmodels.tsa.stattools import adfuller #stationarity

import matplotlib.pyplot as plt
import matplotlib as mpl
```

```
In [4]: #input the data from csv file
file_data = pd.read_excel('ALL-Terms.xlsx').set_index('Term').iloc[:-1,:]
file_data
```

Out[4]:

	Chiari Malformation	Arnold- Chiari Malformation	Cleland- Chiari Malformation	Cruveilhier- Cleland- Chiari Malformation	Chiari Deformity	Chiari Syndrome	Chiari Complex	Tonsillar Herniation	Chiari 0 Malformation	Chiari 0.5 Malformation	...	Chiari 3.5 Malformation	Chiari 4 Malformation	Acquired Chiari Malformation	Mali
Term															
1891-1950	2	63	0	0	28	31.0	1.0	18.0	2.0	0	...	0.0	28.0	0.0	
1951	1	11	0	0	4	3.0	1.0	2.0	0.0	0	...	0.0	0.0	0.0	
1952	0	8	0	0	6	5.0	0.0	1.0	1.0	0	...	0.0	0.0	0.0	
1953	0	16	0	0	6	6.0	1.0	3.0	0.0	0	...	0.0	1.0	0.0	
1954	0	12	0	0	3	6.0	1.0	1.0	0.0	0	...	0.0	1.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2018	736	403	0	1	50	44.0	0.0	356.0	21.0	0	...	1.0	28.0	47.0	
2019	812	336	0	0	77	55.0	0.0	447.0	25.0	1	...	0.0	15.0	41.0	
2020	832	324	0	1	77	59.0	0.0	523.0	47.0	1	...	3.0	20.0	51.0	
2021	986	403	0	0	76	59.0	0.0	507.0	38.0	1	...	0.0	24.0	57.0	
2022	1270	459	0	0	71	45.0	0.0	595.0	40.0	6	...	2.0	24.0	70.0	

73 rows × 24 columns

```
In [ ]:
```

```
In [27]: rate_result=[]
for i in range(file_data.shape[1]):
    if len(file_data[file_data.iloc[:,i]==0])!=0:
        if file_data.iloc[-1,i] != 0:
            start_num=file_data[file_data.iloc[:,i]!=0].index[0]
            temp_df =pd.DataFrame(file_data.loc[start_num:,file_data.columns[i]]).reset_index(drop=True)
            avg_increase_rate = (temp_df.iloc[-1].values[0]/temp_df.loc[0].values[0])** (1 /(len(temp_df)-1))-1

            rate_result.append(avg_increase_rate)
        else:
            rate_result.append(np.nan)
    else:
        temp_df =pd.DataFrame(file_data.loc[:,file_data.columns[i]]).reset_index(drop=True)
        avg_increase_rate = (temp_df.iloc[-1].values[0]/temp_df.loc[0].values[0])** (1 /(len(temp_df)-1))-1
        rate_result.append(avg_increase_rate)
# print(i)
rate_result
```

Out[27]: [0.09377353977728142,
0.0279660688113732,
nan,
nan,
0.01300713531116493,
0.005189464495368945,
nan,
0.04978561520607827,
0.04248511111864306,
0.11115233429704241,
0.02302333108760468,
0.07781390781997133,
0.03043463694457116,
-0.01192132321575845,
-0.015473842420183681,
-0.0021386913977885502,
0.08185356920692333,
0.014403007682450397,
nan,
0.0662774196956184,
0.008197117573228052,
0.13052820036828883,
0.044670422462421255,
0.05396388977839406]

```
In [28]: pd.DataFrame(rate_result).to_csv('da.csv')
```

```
In [ ]:
```